Robert Winter
J. Leon Zhao
Stephan Aier (Eds.)

LNCS 6105

# Global Perspectives on Design Science Research

5th International Conference, DESRIST 2010
St. Gallen, Switzerland, June 2010
Proceedings

Springer

# Lecture Notes in Computer Science 6105

Robert Winter   J. Leon Zhao   Stephan Aier (Eds.)

# Global Perspectives on Design Science Research

5th International Conference, DESRIST 2010
St. Gallen, Switzerland, June 4-5, 2010
Proceedings

Springer

Volume Editors

Robert Winter
Institute of Information Management
University of St. Gallen
St. Gallen, Switzerland
E-mail: robert.winter@unisg.ch

J. Leon Zhao
Department of Information Systems
City University of Hong Kong
Kowloon, Hong Kong, China
E-mail: jlzhao@cityu.edu.hk

Stephan Aier
Institute of Information Management
University of St. Gallen
St. Gallen, Switzerland
E-mail: stephan.aier@unisg.ch

Library of Congress Control Number: 2010927760

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

| | |
|---|---|
| ISSN | 0302-9743 |
| ISBN-10 | 3-642-13334-7 Springer Berlin Heidelberg New York |
| ISBN-13 | 978-3-642-13334-3 Springer Berlin Heidelberg New York |



springer.com


Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

# Preface

With four inspiring conferences that took place in Claremont, Pasadena, Atlanta and Philadelphia, the International Conference on Design Science Research in Information Systems and Technology (DESRIST) has developed into a premier conference for design-oriented research in Information Systems. Becoming a truly global conference series, DESRIST was held in St. Gallen, Switzerland, in 2010. DESRIST 2010 brought together researchers and practitioners engaged in design science research from all over the world to provide global perspectives on both design science and design research in the broadest sense.

The design science research paradigm has been discussed thoroughly in recent years and is now gaining ground for both building knowledge and improving practice in information systems and several related disciplines. As opposed to natural and social research, design research does not crave ultimate truths, grand theories or general laws, but seeks to identify and understand real-world problems and propose appropriate, useful solutions. It is commonly believed that design research involves building, investigating and evaluating innovative artefacts such as constructs, frameworks, models, methods, and information system instantiations in order to solve practical problems. Moreover, the study of methods, behaviours, and best practices related to the problem analysis and artefact development process are encompassed – commonly referred to as design science. An ongoing debate related to the nature, scope and dominant ideologies of design science research, however, shows that the paradigm is still emerging. Its core, its boundaries and its interplay with other research approaches are increasingly being revealed and defined.

The topical theme of DESRIST 2010 is "Global Perspectives on Design Science Research". Once more, the DESRIST conference successfully serves as a forum for raising and discussing new ideas in the area of design science research. Among others, the papers submitted to DESRIST 2010 contribute to a better understanding of the interplay between design and organisation, design and information, design and behaviour, and design and collaboration. A number of contributions present design research exemplars, while others illuminate design research techniques or design research organisation. All papers were reviewed by at least two reviewers and the selection process was competitive. In total, 80 papers were submitted, out of which 35 were selected as full research papers (acceptance rate of 44 %). Furthermore, ten submissions were accepted as short papers and presented as posters. The submissions came from authors located in 29 different countries, geographically distributed as follows: 59 % of the authors are located in Europe, 32.5 % in the Americas and 4.25 % each in Asia and Australia.

In conjunction with the main conference, DESRIST 2010 hosted three workshops on design, enterprise architecture management, and enterprise engineering.

Papers accepted for CIAO! (one of these workshops) have been published in a separate volume of Springer's *Lecture Notes in Business Information Processing* series.

In addition, three invited keynoters and four panels stimulated the discussions on new and emerging issues in line with the conference topics. The panels addressed the following topics: innovation in design science research, design for use, publishing design science research, and organising design science research. We are thankful for the fruitful and inspiring discussions and the interesting impulses for future relevant work in the field of design science research.

We wish to thank all the people who submitted papers to the DESRIST 2010 conference for having shared their work with us. We sincerely hope that you find the papers as interesting and inspiring as we did. Moreover, we owe special thanks to all members of the programme committee of DESRIST 2010 as well as all reviewers for their work. We are also very appreciative to the many people who were involved in the organisation of the DESRIST conference and its accompanying events. We believe that DESRIST 2010 provided detailed insights into the current state of the art, set directions for fruitful further research initiatives and truly contributed to the transfer of academic knowledge for practical problem-solving.

June 2010

Robert Winter
J. Leon Zhao
Stephan Aier

# Organisation

DESRIST 2010 was organised by the Institute of Information Management, University of St. Gallen in cooperation with ACM, ACM/SIGCAS, ACM/SIGMIS, ACM/SIGCHI, and ACM/SIGDAA.

## General Chairs

| | |
|---|---|
| Sandeep Purao | Penn State University, USA |
| Kalle Lyytinen | Case Western Reserve University, USA |
| Il-Yeol Song | Drexel University, USA |

## Programme Chairs

| | |
|---|---|
| Robert Winter | University of St. Gallen, Switzerland |
| J. Leon Zhao | City University of Hong Kong, Hong Kong SAR, China |

## Doctoral Consortium Chairs

| | |
|---|---|
| Ulrich Frank | University of Duisburg-Essen, Germany |
| Shirley Gregor | Australian National University, Australia |

## Proceedings

| | |
|---|---|
| Stephan Aier | University of St. Gallen, Switzerland |

## Publicity

| | |
|---|---|
| Anne Cleven | University of St. Gallen, Switzerland |
| Mark Toleman | University of Southern Queensland, Australia |
| Karthikeyan Umapathy | University of North Florida, USA |

## Local Arrangements

| | |
|---|---|
| Philipp Gubler | University of St. Gallen, Switzerland |

## Programme Committee

| | |
|---|---|
| Antonia Albani | Technical University of Delft, The Netherlands |
| Guy-Alain Amoussou | Humboldt State University, USA |

| | |
|---|---|
| Richard Baskerville | Georgia State University, USA |
| Jörg Becker | University of Münster, Germany |
| Udo Bub | Deutsche Telekom Laboratories, Germany |
| Tobias Bucher | PHOENIX Group, Germany |
| Sven Carlsson | Lund University, Sweden |
| Samir Chatterjee | Claremont Graduate School, USA |
| Guoqing Chen | Tsinghua University, China |
| Anindya Datta | National University of Singapore, Singapore |
| Anke Gericke | Munich Re, Germany |
| Mike Goul | Arizona State University, USA |
| Alan Hevner | University of South Florida, USA |
| Kunihiko Higa | Tokyo Institute of Technology, Japan |
| Lakshmi Iyer | University of North Carolina at Greensboro, USA |
| Pontus Johnson | KTH Royal Institute of Technology, Sweden |
| Dimitris Karagiannis | University of Vienna, Austria |
| Helmut Krcmar | Technical University of Munich, Germany |
| William Kuechler Jr. | University of Nevada, USA |
| Susanne Leist | University of Regensburg, Germany |
| Qing Li | City University of Hong Kong, Hong Kong SAR, China |
| Ting-Peng Liang | City University of Hong Kong, Hong Kong SAR, China; National Sun Yat-Sen University, Taiwan |
| Ishwar Murthy | Indian Institute of Management Bangalore, India |
| Björn Niehaves | University of Münster, Germany |
| Jinsoo Park | Seoul National University, Republic of Korea |
| Jeffrey Parsons | Memorial University of Newfoundland, Canada |
| Erik Proper | Radboud University Nijmegen, The Netherlands |
| Matti Rossi | University of Helsinki, Finland |
| Cecilia Rossignoli | University of Verona, Italy |
| Henk Sol | University of Groningen, The Netherlands |
| José Tribolet | INESC and University of Lisbon, Portugal |
| Vijay Vaishnavi | Georgia State University, USA |
| Jan Verelst | University of Antwerp, Belgium |
| Jan vom Brocke | University of Liechtenstein, Liechtenstein |
| Chih-Ping Wei | National Tsing Hua University, Taiwan |
| George Widmeyer | New Jersey Institute of Technology, USA |
| Roel Wieringa | University of Twente, The Netherlands |

## Financial Support

McKinsey & Company, Inc.
http://www.mckinsey.com

Deutsche Telekom AG Laboratories
http://www.laboratories.telekom.com

SAP AG
http://www.sap.com

University of St. Gallen
http://www.unisg.ch

## Publication Partner

Business & Information Systems Engineering (BISE) – The International Journal of WIRTSCHAFTSINFORMATIK
http://www.bise-journal.org

# Table of Contents

## Organising Design Research



## Reflecting Design Science Research

## Design Research Techniques



## Design and Context



## Design and Organisation



## Design and Information

## Design Research Exemplars



## Design and Behaviour

## Designing Collaboration



## Design and Requirements Engineering



## Short Papers

# When Designers Are Not in Control – Experiences from Using Action Research to Improve Researcher-Developer Collaboration in Design Science Research

Anders Hjalmarsson[1,2], Daniel Rudmark[3], and Mikael Lind[1,2]

[1] University of Borås, School of Business & Informatics, Sweden
[2] Viktoria institute, Sweden
[3] University of Borås, InnovationLab, Sweden
{Anders.Hjalmarsson,Daniel.Rudmark}@hb.se,
Mikael.Lind@viktoria.se

**Abstract.** Design science research (DSR) has received much attention in the past few years from the field of information systems. This paper argues that control in researcher-developer collaboration during artefact development has not yet received enough attention in design science research even though control is necessary for successful artefact instantiation. Experiences are presented from improving researcher-developer collaboration during DSR by using action research (AR) as means. These experiences are driven from the need to achieve meta-design control throughout the development of artefacts by non-researching system developers when DSR is performed in an authentic setting. The paper shows that the use of AR to both diagnose uncertainty and actively improve building activities may lead to sustainable improvements in researcher-developer collaboration, and hence achieve meta-design control in DSR performed in authentic development environments, as well as enhance progress in DSR methodology development.

**Keywords:** Design science research, action research, researcher-developer collaboration, meta-design control, authentic setting.

## 1 Introduction

In 50 years of research within the field of information systems (IS) a large amount of work has been dedicated to actual system construction. Recently, more and more attention has been paid to a theory of the artefact as such [1] and it has been speculated that researchers should actively engage in the actual design of the systems they study through Design Science Research (DSR) [2]. Within IS DSR a number of argumentations have been posed on the potential relationship between action research (AR) and DSR and whether these two methods of inquiry may benefit from joint uses [3, 4, 5]. This debate has so far concerned the environment in which the artefact is operating within, and to what extent the artefact should intervene (through evaluation) and be informed by (through design propositions) a particular environment and what potential role this plays for the final design theory. However, engaging in such research regarding artefacts, be it by focusing on technical principles [6] or on the construction of more socio-technical systems [7], requires complex construction activities.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 1–15, 2010.

Researchers pursuing DSR are constructing artefacts in the domain of their study. This does not necessarily mean that the researchers are the sole responsible for the implementation of the artefact: in fact, they may cooperate with other specialists in order to construct a design. Hence, particular actions taken as a result of the interplay between researchers and developers could have an impact on the final design theory [7]. Still, DSR methodology literature has yet to articulate how to make the best of such collaborative construction settings. Hence, this paper aims at contributing to the research activity build (construction, artefact development) in DSR and provide experiences how to control the relationship between researcher-developer when artefacts are constructed. More specifically, we do this by exploring Action Research (AR) as a means to develop meta-design control during artefact development in an authentic DSR setting. AR is here used as a structure in a DSR project concerned with platforms for architectural innovation using collaborative artefact construction. Experiences and the constituents of this use of AR are framed in accordance with canonical action research (CAR) [8]. This research is motivated, as discussed above, by the lack of knowledge concerning the interplay between researchers and developers engaged in construction in DSR projects; as such, the research question in this paper is *how to obtain meta-design control in researcher-developer collaboration during artefact construction?*

The paper is organized as follows: first, we present a review of DSR literature focusing on artefact construction and the need of meta design-control in design research. This is followed by a short overview of AR and some of its currently established relationships with DSR. Next, we detail our experiences from collaborative construction and how AR may be used to obtain control in researcher-developer collaborations. We conclude with a discussion of the experiences we made and how this use of AR relates to previous uses of 1) AR in DSR, 2) meta-design control and 3) AR in one's own organization.

## 2 Artefact Development, Authentic Settings and Meta-design Control in DSR

In DSR, knowledge about and understanding of a problem domain and its artefact-based solutions are achieved through a scientifically grounded implementation and evaluation of artefacts [9]. At the core of DSR thus lies the creation of artefacts solving problems that so far have not been solved. In DSR, the formulation of a problem typically proceeds the construction of artefacts [10], but it has been argued that the opposite also holds true [11]. Hence, it might be argued that the research problem and the artefact typically co-evolve over time [12] requiring sufficient method support to effectively deal with these changes.

A closer investigation of widely used frameworks in this line of research reveals similar although not identical structures. The most cited framework by Hevner et al. [9] argues that IS researchers in DSR are concerned with two major activities: develop / build (theories and artefacts) and evaluate (through a variety of methods ranging from simulations to case studies to field studies). These activities are influenced both by the environment (expressed as business needs) and by the existing body of knowledge (theories). Nunamaker et al. presented a framework [11] which was later extended by Venable [13]. This latter extended version places solution technology invention as the central activity in a DSR project, surrounded by theory building and evaluation (either naturalistic evaluation or artificial evaluation).

Considering the importance of the artefact construction process to DSR argued for in the frameworks we reviewed, the literature has so far paid insufficient attention both to DSR in an authentic setting and to the challenges that could occur during researcher-developer interplay in a development environment such as that. On a more general, problem-solving level, guidance to this process has been offered [10] and recently extended [14], but when it comes to the day-to-day handling of artefact construction an insufficient volume of investigations has been performed. In DSR these activities may not be considered to be merely instrumental on behalf of the researcher. Rather, as shown for example in Markus et al. [7], reflections based on the interplay between researchers and non-researching developers had a direct influence on principle no. 6 in DSR [7 p. 202].

Engaging in exploratory research ("*knowing through building*" [15]) requires a different set of skills and practices than other types of research approaches where the researcher is potentially more in control of the inquiring instruments (i.e. experiments, interviews, surveys). Since the actual construction activities for a certain artefact may lie outside the scope of the core competency of the researcher(s), these skills must be obtained elsewhere, for example by engaging professional developers and end users in an authentic development setting. Since the research is then concerned with exploring different options to achieve a desired future state, uncertainty and unanticipated changes are given and must be dealt with on a regular basis.

This emphasizes a need for alternative forms of control during DSR. Meta-design [16] is an emerging framework aimed at defining and creating social and technical infrastructures in which new forms of collaborative design can take place. It expands beyond the traditional notion of system development by allowing users as well as developers to become co-designers [17]. Meta-design is grounded in the idea that future uses and problems cannot be completely anticipated at the time of the design, when an artefact is developed: Similarly, we argue that problems during DSR cannot be completely anticipated in advance. In order to obtain control in the DSR process challenges must be identified and managed by the researchers together with the developers by using a jointly established meta-design framework. The aim of this paper is to explore AR constituting such a meta-design framework when DSR is performed.

## 3 AR as a Means to Improve Practice

AR is a methodology which intends to contribute both to practical concerns in problematic situations and to the scientific body of knowledge. Its origins can be traced back to Lewin [18] and one of his descendants, Blum, states that "*AR includes diagnosing social problems with a view of helping to improve the situation*" [19]. This is also evident in the work carried out by Susman and Evered, CAR, [8] which prescribes more structure to AR than Lewin and Blum's original work did, and, by doing so, adds more scientific rigour to the research process [20, 21]. In CAR, five canonical phases are described in a cyclic process: 1) diagnosing, 2) action planning, 3) action taking, 4) evaluating, and 5) specifying learning. Before iterating a research environment labelled, the client-system infrastructure has to be established. Susman has later described the phases in greater detail, based on a pragmatic perspective (anchored in John Dewey's conception of inquiry), where the working hypothesis is introduced as an important element of CAR [22].

Furthermore, Davison et al. introduce five principles for CAR which enable the researcher to evaluate the AR project in order to assess research quality [23]. These are: 1) the Principle of the Researcher–Client Agreement, 2) the Principle of the Cyclical Process Model, 3) the Principle of Theory, 4) the Principle of Change through Action, and 5) the Principle of Learning through Reflection. By applying these elements to the AR process, rigour in AR is strengthened, as is suitability as a framework in and for research situations in authentic settings; e.g. when problems occur during collaborative construction of artefacts in DSR conducted in an authentic development environment.

Being a methodology for improving practice, AR could be used in situations in which the researcher intervenes in an external organisation. However, as pointed out by Coghlan and Brannick, AR could also be performed from the inside of an organisation, and then aim at achieving both personal goals from the project as well as contributing to the organisation the researcher is a part of [24]. Amongst the challenges that occur are 1) the development of ways of knowing what is familiar for the organisation, and 2) the development of practical knowledge that enables the researcher to understand what is taking place, and how to appropriately intervene in her own organisation. Being up to the challenge of holding both an existing organisational role and an adopted AR role with the purpose to improve the ongoing practice is particularly important for project success.

## 4 Established Uses of AR in DSR

In recent years a number of researches have discussed the relationship between AR and DSR. In this paper we do not seek to confirm the validity of these argumentations, but rather to demonstrate how AR can be used as a framework to *obtain meta-design control in researcher-developer collaboration* during artefact construction when DSR is carried out in authentic settings.

Cole et al. suggest that AR could be used in DSR in two ways: 1) by adding an AR cycle in the last part of the evaluation to increase reflection and learning; 2) "*a DR project may be framed as an AR project if an organizational problem needs to be solved, and the action involves building a system" [4].* Iivari and Venable [5] find similar joint usage. Either AR can be used as a naturalistic evaluation method or when developing artefacts for and in socio-technical settings [13].

Lee finds striking similarities between AR and DSR and demonstrates how the research activities/research output framework proposed by March and Smith [26] may be framed using AR. Lee argues that a two-cycle AR version of March and Smith' framework could then serve for the development of the client's business practice and as a means for creating theories for the scientific community [25]. Interestingly, the build phase of an instantiation is however omitted in this framework [25].

In this paper, AR is used as means to improve DSR during the build process, i.e. construction/development activities. Our suggestion is that AR is a suitable framework for solving problems in relation to control that arises during collaborative artefact development. We hence add knowledge to the existing couplings between these two research approaches by recognizing that artefact development in an authentic design research setting may be of collaborative nature in which researchers, through the actions of others (i.e. developers), modify underlying technology. In such settings problems may arise between researchers and developers which need intervention in order for the DSR project to succeed (see figure 1), while simultaneously a way to increase the body of knowledge about DSR is achieved.

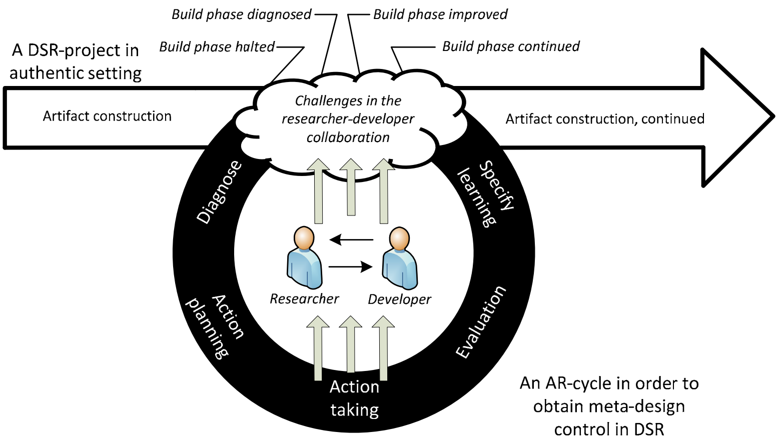



**Fig. 1.** AR as framework to obtain meta-design control in researcher-developer collaboration artefact construction is performed throughout DSR

We consequently propose another use of AR in DSR, this in addition to Cole et al. [4] suggestion of adding an AR cycle as a last part in the evaluation or framing of the DSR project as an AR project. AR is instead performed within a DSR-project as a means to obtain meta-design control when DSR is performed. The following case study demonstrate the challenges that may arise during artefact development, and how meta design control jointly could be obtained by using AR as a framework to improve the researcher-developer interplay.

## 5 Case Study: Improving Distributed Artefact Construction in DSR with AR

### 5.1 A Need of Control in DSR: The Halted Build Phase

The DSR in the subsequent case study was aimed to develop an architecture for a digital assistant through which users could induce their own custom-programmed services. It initially started in 2006 when an electronic assistant for students was elaborated in numerous workshops, verified through surveys and built as a prototype [27]. The evaluation of the implemented artefact indicated that it was considered useful and filled an important space not currently occupied. A first version of the assistant was built and evaluated in 2006 and 2007 [28].

In November 2008, a DSR project was organized for building an improved architecture with added functionality. It was also decided that the development methodology to be used for this DSR project was to adhere to agile principles – and the Scrum methodology was chosen [29]. This was due to the fact that the architecture should be explored together with prospective users which supposedly made a more plan-oriented method a bad fit and that the development organization was familiar with this method due to experiences from using this methodology in other projects. The project was staffed with two Scrum masters, both acting as DS researchers sharing far-reaching experiences in systems development, collaborative modelling, facilitation, and DSR. In addition, the core development team consisted of two senior developers/architects and one member responsible for quality assurance (QA). Later in the project one additional part-time developer joined the development team.

Sprint 1 unfolded roughly as planned. During this phase focus remained on testing the individual proposed technical frameworks that the architecture was to be built upon; e.g. the Google API and various open source software (such as MySql, JBoss and Spring).

In Sprint 2, the first working functionality was to be delivered. In the spirit of Scrum, it was important to deliver working features for evaluation early in the DSR project. The researchers found it especially important to test the implemented design with prospective users for naturalistic evaluation. At the start of Sprint 2, the activities were collectively defined and time-estimated during a four-hour session (as advised by the Scrum methodology [29]). Since this project was in the area of architectural innovation, the necessary time-estimated activities concerning system architecture identified at the time grew relatively large, ranging from approx. 20 to 160 hours per activity. These numbers were a result of the great uncertainty invoked by the process of building the innovative architecture. Although the task was considered complex, there was consensus in the team that the expected delivery was feasible.

Nonetheless, just one week before the testing was to begin the development team realized and communicated on the Scrum meeting that deliverables would not be completed in time. The construction of a component that translated the service need for information exchange had turned out to be more complex than initially estimated. The DSR researchers had stressed the importance of a truly generic framework, making it not only possible, but also inspiring for diverse stakeholders, to construct their own innovative services [30].

Although this vision served as an inspiration for the development team during Sprint 2, the consequences of realizing the ideal had become difficult to foresee during the development. For example, implementing a generic workflow engine revealed much more complexity than first anticipated; how to combine simplicity, flexibility, and third-party developer inspiration into an API required a lot of designing and re-designing. Nevertheless, in order to meet the delivery date, a first version of the artefact was implemented using less generic technology than first planned for. As a result, delivery happened on time, but the technology used in this implementation was reduced to an absolute minimum.

During the planning of Sprint 3 the objective was set to implement the ability to transfer information from the university scheduling system to an external calendar such as Google Calendar. More doubts were now raised during the planning as to whether the goal was actually achievable, but after discussing the matter and conducting collective time estimation the goal was maintained. About two weeks into Sprint 3 the development team announced that the deliverables would not be ready at the end of the Sprint. Each day that passed revealed that more things were necessary in order to succeed but were missing in the core architecture. This in turn made it difficult for developers to keep up with delivering the planned functionalities. At this point it was decided to black-box the scheduling system and instead build a mock-up service, using a correct programming interface but delivering nothing but predefined data (no real class schedules). This way a functional system could be delivered in order to QA the usability of the system, although the actual integration with the scheduling system was omitted. About a week later, this new goal turned out to be unreachable as well since the development work with the core architecture remained unfinished. The Scrum masters and developers tried to define yet another reduced set of features that could be delivered at the end of Sprint 3, but the ones that were considered achievable added too little value for the potential users compared to the QA resources needed in order to finalize them. Hence, all deliverables were postponed to Sprint 4.

### 5.2 Initiating an AR Cycle to Improve the Build Phase

At this stage it became apparent that the method-in-use needed to be reconfigured. In order to give the project new momentum and re-establish control, the DS researchers, acting as Scrum masters, and together with the rest of developers, agreed that the DS researchers should initiate an inquiry of the causes behind the problems. The results of this investigation would have provided the basis for a joint intervention to resolve them.

Hence, a client-system infrastructure was established between the developers and the researchers, allowing the former to switch roles from being DS researchers to being AR researchers. This effectively established an AR setting in which the AR researchers, with permission from the developers, implemented a CAR action cycle in the DSR project. The jointly-stated question was: *how should collaborative construction of architecture-intensive artefacts be pursued in order to continuously and incrementally deliver working functionality on time and with quality?* For the AR researchers the goal was both to contribute to the organization, by creating a sustainable improvement of development environment, and more generally, in line with Coghlan and Brannick [24], to improve the control and enhance the progress in the DSR project and develop sustainable knowledge for DSR as methodology.

### 5.3 Diagnosing the Halted Build Phase

Since time estimating was both central in this development environment and a familiar process to the developers, the AR researchers suggested that the estimated vs. actual time spent in the project predictions should be jointly analysed with the developers. The analysis clearly showed a large discrepancy between the two. This was thought to be caused by the estimation process used [29], collectively carried out during a four-hour meeting (for all activities). However, a lot of the difficulties in achieving the objectives emerged during implementation in the sprints. The absence of a thorough technical analysis was also thought to be one important cause to the halted build by all parties.

This was linked to a third cause, the fact that activities in the sprints were too large and not sufficiently defined. As a result, these became hard to follow-up in the daily scrums and potential warning signs were hence not identified. By dividing the activities into smaller entities (e.g. not exceeding 10 hours) it was believed it could be possible to both achieve better control on actual progress and potential obstacles, as well as increase the developers' work satisfaction through a sense of completion of smaller tasks carried out on a regular basis. This shift also implied that an on-going definition of tasks was encouraged, not only during the sprint planning, but also during the sprints themselves in order to stimulate technical analysis.

These changes guided the AR researchers to formulate a working hypothesis to guide the planning of the AR intervention: Continuous technical analysis, additional detailed activity scoping and more thorough follow-up's are achieved during collaborative construction of architecture-intensive artefacts via jointly reconfigured and appropriate ways of working in the project. This working hypothesis, grounded in the causes, of how to resolve the situation was presented to the developers and an agreement was reached to act upon it. At this stage the developers also underlined the importance that a reconfigured method should stimulate even more researcher-developer collaboration during the planning and the follow-ups. According to them, this was necessary in order to stimulate participation, shared responsibility and amplified technical analysis during the sprints. The intervention should therefore be done in a jointly fashion and result in a reconfigured method that in itself facilitated enhanced collaboration during the continuation of the build phase, redistributing control from the researchers to the researchers and the developers. Based on this comment, a revision was made in regard to the working hypothesis: Continuous technical analysis, additional detailed activity scoping and more thorough follow-up's are achieved during collaborative construction of architecture-intensive artefacts via jointly reconfigured and appropriate ways of working, which should facilitate the interplay between researchers and developers during build.

### 5.4 Action Planning: The Build Phase Improved

Equipped with an agreed-upon working hypothesis, the AR researchers were charged with the responsibility to plan the intervention. The researchers' first step was to apply activity theory [31] in order to grasp the notion of work from an existing conceptual framework. Activity theory states that subjects perform activities through tools in order to achieve objectives [32]. The causes elicited implied a need to reconfigure the method and this was perceived by the researchers as equivalent to tools in activity theory. Engeström states that in addition to tools, rules must be set up and used, as well as a division of labour which organises the collective accomplishment of the activity [33]. Therefore, the researchers planned the intervention so that it contained the development of a tool, rules and a revised division of labour (a reconfiguration of Scrum) to facilitate technical analysis, detailed activity scoping and more thorough follow-up's.

The tool should assist the splitting up of defined activities into smaller time-estimated actions. Consequently, a tool providing this functionality was set up and named the *sprintbook*. It consisted of one sheet in MS Excel for each Sprint. The specific sheet was divided into four sections with specific areas of concern for each section. By using the sprintbook, the researchers aimed to 1) separate and document larger activities into smaller ones, 2) keep track of the different prioritizations, and 3) distribute different activities to the participants in the DSR project.

In order to facilitate the interplay throughout the planning and follow-up meetings, additional rules and a revised division of labour was designed. The researchers turned to Conklin [34] for the theoretical principles in designing the rules. Conklin argues that collaboration in a social setting can be improved through collaborative displays constituted by a *notation*, an infrastructure and a facilitator.

The aim with this part of the intervention was to plan a more collaborative way for the whole development team to jointly transfer the agreements recorded in the sprint-book to individuals in the team during the Scrum meetings. The notation developed for this task included a set of simple rules, depicted in figure 2, and a collaborative display for documentation, in this case an ordinary whiteboard.

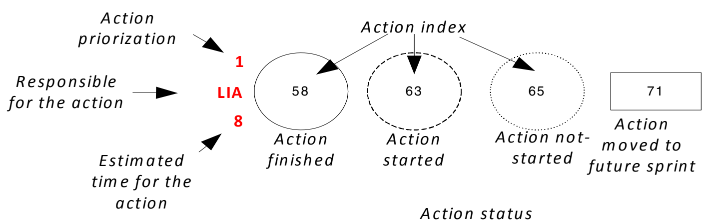



**Fig. 2.** Developed notation-in-action

The notation-in-action was consequently intended as a view over the same content that was available in the sprintbook. It was created to 1) make the agreements in the Sprint comprehensible for all participants at the same time, 2) to provide means to facilitate the joint tracking of achievement during the collaborative construction, and 3) to facilitate dynamically and cooperatively adjustments in the plan, based on any change of circumstances during the Sprint.

The third part in the reconfiguration of the method was the adjustment of the *division of labour* during the Scrum meetings. This part of the intervention was aimed at reconfiguring how meetings were conducted when the sprintbook was used together with the collaborative display. These aimed to assist the team to manage the meetings with a focus on potentially splitting up activities, estimate time, assign actions to individual developers, visualize the road to the sprint deadline and, also, to track progress in the project. Consequently, the combinations of these actions were designed to overcome the causes which had inflicted the project problems.

The new way to co-facilitate the meetings stated that one of the Scrum masters should facilitate the dialogue via the sprintbook. That amounted to asking the first Scrum-prescribed question to the developer in focus (What did you do yesterday?), based on the information in the sprintbook. As a response to the developer's answer, the second Scrum master had to relate the reply to the collaborative display on which the path to the sprint deadline was laid out. If changes in relation to the plan had occurred, this resulted in a proposed modification of the visualization of the Sprint. If the change was considered relatively trivial, an estimate was given upfront during this discussion. However, if the nature of the task was too complex to properly estimate during the meeting, a new task covering the needed technical analysis had to be brought up as an explicit task and its consequences on the sprint planning became immediately evident to the whole team. In this way, new unanticipated modules that needed to be constructed became explicit artefacts rather than just undefined subparts of a larger task (and for the designated developer to have in mind) as had been the case in previous sprints. When the group reached an agreement about the modification, then the first Scrum master reorganized the sprintbook and asked the next question (What will you do today?). This interaction circle was then repeated.

The revised division of labour consisted then in both new individual tasks for the two Scrum masters as well as a new structured interplay between the DS researchers and the developers, redistributing the control amongst them throughout the artefact construction.

### 5.5 Action Taking: The Build Phase Continued

The AR researchers reconfigured the method as described in section 5.4. They then notified the developers that a preliminary solution was ready to be used in the fourth sprint. The developers agreed that the reconfigured way of working was in line with their conception of the problems as well as the formulated suggestion (working hypotheses) and that it should be used during the sprint. The agreement to continue the build phase stated that the experiences from using the solution should be accumulated by the participants and discussed every week during the Scrum meetings. Changes in the method-in-use were consequently jointly identified, agreed-upon and implemented in the process, and put to effect during the following week, thus securing momentum to the DSR project. At the end of the sprint a major Scrum retrospective [29] was done to assess the progress in relation to the overall plan, as well as to evaluate the overall effects of the reconfigured method.

### 5.6 Evaluation

The retrospective clearly demonstrated that the intervention had effectively addressed the problems that halted the DSR efforts. Technical analysis was conducted with higher frequency by the developers in order to construct the functionality on a conceptual level before implementing it in the architecture. The development had now also included a more thorough activity scoping and although time estimations sometimes differed from actual time spent, warning signs were now early identified and each sprint delivered functionality for testing as planned.

Since the evaluation of the action taking was made a part of the system development work, it also became a natural part of the ongoing project. This unremitting evaluation played an important role in the intervention to 1) transfer the reconfigured method into daily work, 2) to evaluate its effects, and 3) to identify requirements for refinements. By executing these activities action taking was performed to secure the CAR principle of learning through reflection. In all, this implies that the work done in the action taking was guided by the CAR principle of change through action and has accomplished sustainable change.

The principle of theory in CAR influenced the intervention in two ways. After the joint diagnose of the causes which resulted in the breakdown, the AR researchers used Scrum methodology together with activity theory as a conceptual framework to grasp the constituents of an action. By using this theory generic categories became the foundation for planning the intervention as a whole. The reconfiguration of the method, as stated in the working hypothesis, was consequently directed with tools as a phenomena (the sprintbook and the collaborative display), rules as a phenomena (the logic in the sprintbook and notation used on the collaborative display) and division of labour during the meetings (the team model for co-facilitation). The logic in the sprintbook was based on experiences from the project and on the diagnosis of the problems at hand and it was then used as a theory together with collaborative theory to design the collaborative display [34]. The division of labour was at the same time based on classification of forms for co-facilitation [35]. Experiences from diagnosing the problem situation was induced and transformed into parts and logic for the reconfigured method, especially during the design of the sprintbook.

The principle of the researcher-client agreement came into effect when the initial client-system infrastructure was established. The agreement was however revised after this phase. The new agreement stated that the AR researchers themselves should plan the intervention and transform the working hypothesis into a reconfigured method which should resolve the temporary breakdown and stimulate improved collaboration. The developers did not want to participate in this preparation. One explanation is that the DS researchers, now acting as AR researchers, also acted as Scrum masters. Their perceived responsibility, from the developers point of view, was to provide methods for managing the project when they acted as Scrum masters. As systems developers they wanted suggestions from the researchers on concrete solutions, in line with the working hypothesis, to the problems that halted the build process. The AR framework gave the DS researchers a means to meet that expectation.

One important validity claim in AR is that the effects of the actions taken in the AR project should be sustainable or even irreversible [3]. Auditing the performance in the project provides convincing indications that the changes made via the AR intervention indeed became sustainable and that the improved results produced in sprint 4 and sprint 5 were carried on throughout the following sprints. Used in this way the AR framework worked as a structure to create sustainable changes in the development team.

### 5.7 Specify Learning

The learning outcomes from the AR cycle in this DSR project where fed into the revision of the reconfigured method, and became learning outcomes for this local practice. Significant learning outcomes were: 1) the joint planning of the Sprint was improved by the use of the collaborative display, the chosen co-facilitation form, and the designed workbook; 2) after the intervention, the technical analysis carried out through the reconfiguration of the method, was done more thoroughly as these analysis activities were now illustrated on the collaborative display and in cooperation assigned to individual developers who now to a higher degree a) felt responsibility to perform them, as well as after the intervention; b) could better understand the value the analysis brought in relation to later building activities in the sprint; 3) through the reconfigured method, the DS researchers could follow the progress in the DSR project much better, and as a result could identify signs of uncertainty and challenges much earlier on than before.

The contribution this project brings to the scientific body of knowledge is primarily that artefact development in DSR can be improved significantly by using AR as a means to enhance the researcher-developer interplay during build. In this authentic setting, CAR has proven to be a suitable framework to use in obtaining meta-design control during the build phase. An established framework, CAR, was used to diagnose the causes and take collaborative actions in order to improve researcher-developer collaboration in a authentic development environment. We argue that it is a valid structure to use for the establishment of a focused, but important, impact in an DSR project and at the same time results in findings which contribute to DSR as methodology.

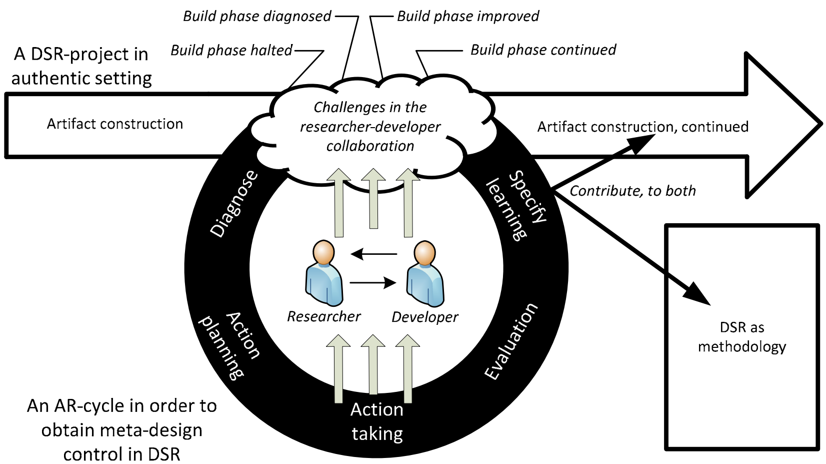



**Fig. 3.** AR as framework to obtain meta-design control in artefact construction and to contribute to DSR as methodology

## 6 Discussion: Using AR to Improve DSR When Designers Are Not in Control

This paper set out to explore and present experiences from managing challenges during collaborative construction of artefacts in DSR. AR has been used to both manage these challenges and develop the knowledge. As noted in the literature review, the discussion about AR/DSR has previously concerned intervention in a context similar to or actually where the artefact is to be used [4, 5, 9, 11, 13, 25] However, as identified in Markus et al. [7], researchers in DSR projects may not be developing and implementing technical artefacts by themselves, but rather participate in a researcher-developer collaboration to construct these artefacts. Moreover, the DSR community has not yet developed method components to support such DSR collaborations. As demonstrated by the empirical material, insufficient collaborative work procedures between researchers and developers indeed led to a halt in the progress, threatening the outcome of an entire DSR project. The data also shows that using AR to diagnose and actively change build activities led to sustainable improvements in a local DSR project.

As DSR is gaining increasingly wider acceptance and interest in the IS-field, both the background and research interests of the researchers in DSR become more and more heterogeneous. Although DSR has been identified to spring out of the engineering disciplines [36] there seems to be one expansion (among others) in recent years in IS DSR towards inquiring more socio-technical systems. This means that the researchers may have technology as one of many objects under study, compared to more traditional DSR projects having technical principles as their major focal point. In such multi-faceted research it seems likely that researchers in DSR projects, to different degrees, collaborate with specialists to modify the underlying technology. Since the researcher-developer interplay is mostly about communicating and transferring the researcher's design propositions to digital form, these two activities should be the focus of the rules governing this collaboration. We argue that the experiences presented in this paper add evidence to help understand this important interplay.

The research question in this paper was *how to obtain meta-design control in researcher-developer collaboration during artefact construction?*

Control was successfully obtained by redistributing it from the researchers to researchers and developers by means of an improved researcher-developer interplay facilitated by the results produced by the use of CAR. By using AR as framework, a resilient participatory design effort was thereby established to manage the problems that had occurred in DSR. In line with Fischer [17], our experiences is that meta-design control can be achieved when intended users (the developers and the researchers) of the results are brought into the process early and not misused during the design phase. Developers were a part of the CAR cycle, but were not forced to participate in all activities; i.e. in the action planning. The effort was characterised by a high degree of collaboration between developers and researchers: this was achieved through motivation, mutual support and sharing, and was also characterized by familiarity [24]. This also means that the development method was not completely changed or replaced. Instead the AR cycle resulted in *revised* method components based on the methodology already in place, which was well-known by the development team, adding a high degree of familiarity which allowed for the successful redistribution of control.

Hence, we conclude that AR, in addition to previously suggested uses [4, 5], may serve as a valuable and focused measure to be used throughout artefact development in authentic settings in order to develop wider knowledge about DSR as well as obtain control and enhance progress during DSR ... when designers are not in control.

## References


1. 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)
2. Au, Y.A.: Design science I: The role of design science in electronic commerce research. Communications of AIS 7 (2001)

3. Järvinen, P.: Action research is similar to design science. Quality & Quantity 41, 37–54 (2007)
4. Cole, R., Purao, S., Rossi, M., Sein, M.: Being proactive: Where action research meets design research. In: Proceedings for ICIS 2005, pp. 325–336 (2005)
5. Iivari, J., Venable, J.: Action research and design science research: seemingly similar but decisively dissimilar. In: Proceedings for ECIS 2009, Verona, Italy, June 8-10 (2009)
6. Gavish, B., Gerdes, J.: Anonymous mechanisms in group decision support systems communication. Decision Support System 23(4), 297–328 (1998)
7. Markus, M.L., Majchrzak, A., Gasser, L.: A design theory for systems that support emergent knowledge processes. MIS Quarterly 26(3), 179–212 (2002)
8. Susman, G., Evered, R.: An assessment of the scientific merits of action research. Administrative Science Quarterly 23, 582–603 (1978)
9. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MISQ 28, 75–106 (2004)
10. Vaishnavi, V.K., Kuechler Jr., W.: Design science research methods and patterns: innovating information and communication technology. Auerbach Pub. (2007)
11. Nunamaker, J., Chen, M., Purdin, T.D.M.: Systems development in information systems research. J. of Management Information Systems 7(3), 89–106 (1991)
12. Purao, S.: Design research in the technology of information systems: truth or dare. Pennsylvania State University (2002)
13. Venable, J.R.: A framework for design science research activities. In: Proceedings of the Information Resource Management Association Conference, Washington, DC, USA, May 21-24 (2006)
14. Gericke, A.: Problem solving patterns in design science research: learning from engineering. In: Proceedings for ECIS 2009, Verona, Italy, June 8-10 (2009)
15. Purao, S., Baldwin, C.Y., Hevner, A., Storey, V.C., Pries-Heje, J., Smith, B., Zhu, Y.: The sciences of design: observations on an emerging field. Harvard Business School Finance Working Paper No. 09-056 (2008)
16. Fischer, G., Giaccardi, E.: Meta-Design: A framework for the future of end user development. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) End user development: empowering people to flexibly employ advanced information and communication technology, pp. 427–457 (2006)
17. Fischer, G.: Meta-design: expanding boundaries and redistributing control in design. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007, Part 1. LNCS, vol. 4662, pp. 193–206. Springer, Heidelberg (2007)
18. Lewin, K.: Frontiers in Group Dynamics. Human Relations 1(1), 5–41 (1947)
19. Blum, F.: Action research: a scientific approach? Philosophy of Science 22(1), 1–7 (1955)
20. Baskerville, R.L., Wood-Harper, A.T.: Critical perspective on action research as a method for information systems research. J. of Information Technology 11, 235–246 (1996)
21. Baskerville, R.L., Wood-Harper, A.T.: Diversity in information systems research methods. European J. of Information Systems 7(2), 90–107 (1998)
22. Susman, G.: Action research: a sociotechnical perspective. In: Morgan, G. (ed.) Beyond method: strategies for social research, pp. 95–113. Sage Publications, Thousand Oaks (1983)
23. Davison, R.M., Martinsons, M.G., Kock, N.: Principles of canonical action research. Information Systems Journal 14, 65–86 (2004)
24. Coghlan, D., Brannick, T.: Doing action research in your own organization, 3rd edn. Sage Publications, Thousand Oaks (2010)

25. Lee, A.: Action is an artifact: what action research and design science offer to each other. In: Kock, N. (ed.) Information systems action research: an applied view of emerging concepts and methods, pp. 43–60 (2007)
26. March, S.T., Smith, G.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)
27. Albinsson, L., Lind, M., Forsgren, O.: Co-design: an approach to border crossing, Network Innovation. In: Cunningham, P., Cunningham, M. (eds.) Expanding the knowledge economy: issues, applications, case Studies, pp. 977–983. IOS Press, Amsterdam (2007)
28. Lind, M., Albinsson, L., Forsgren, O., Hedman, J.: Integrated development, use and learning in a co-design setting: experiences from the incremental deployment of e-Me. In: Cunningham, P., Cunningham, M. (eds.) Expanding the knowledge economy: issues, applications, case Studies, pp. 773–780. IOS Press, Amsterdam (2007)
29. Schwaber, K., Beedle, M.: Agile software development with Scrum. Prentice Hall PTR, Upper Saddle River (2001)
30. von Hippel, E., Katz, R.: Shifting Innovation to users via Toolkits. Management Science 48(7), 821–833 (2002)
31. Kaptelinin, V., Nardi, B.A.: Acting with technology: activity theory and interaction design. The MIT Press, Cambridge (2006)
32. Kuutti, K.: Activity theory and its applications to information systems research and development. In: Nissen, H.-E., Klein, H., Hirschheim, R. (eds.) Information Systems Research: Contemporary Approaches and Emergent Traditions, North Holland, Amsterdam, pp. 529–549 (1991)
33. Engeström, Y.: Activity theory and individual and social transformation. In: Opening address at the 2nd International Congress for Research on Activity Theory, Lahti Finland (1990)
34. Conklin, J.: Dialouge mapping: building shared understanding of wicked problems. Wiley, Chichester (2006)
35. Hogan, C.: Understanding facilitation: theory and principles, p. 36. Kogan Page (2002)
36. Kuechler, W., Vaishnavi, V.: The emergence of design science research in information systems in North America. J. of Design Research 7(1), 1–16 (2008)

# Relevance through Consortium Research? Findings from an Expert Interview Study

Boris Otto and Hubert Österle

University of St. Gallen, Institute of Information Management,
Müller-Friedberg-Str. 8, 9000 St. Gallen, Switzerland
{Boris.Otto,Hubert.Oesterle}@unisg.ch

**Abstract.** The Information Systems (IS) community is discussing the relevance of its research. Design-oriented IS research is considered a promising approach since it combines practical relevance and scientific rigor. Only limited guidance, however, is available for the researcher to gain access to and exchange knowledge from the practitioners' domain. This is surprising insofar as the IS "ecosystem" is under change and research and innovation largely takes place in the practitioners' community. Consortium research addresses the issue of getting access to and exchanging knowledge from the practitioners' community. It supports the development of artifacts and is characterized by close cooperation between the university and its partners in all stages of the design-oriented research process, practical validation of research results with partner companies, and a focus on the practical benefits of the research, with all research activities being funded by the consortium partners. The research question posed in this paper is what consortium research contributes to design-oriented IS research against the background of the aforementioned phenomena. The paper presents the findings from an expert interview study among professors of the German-speaking IS community in Europe.

**Keywords:** Consortium Research, Design Science Research in IS, Research Relevance, Expert Interviews.

## 1 Introduction

### 1.1 Motivation and Problem Statement

In the Information Systems (IS) research community, three phenomena can currently be observed. First, the community is debating on how to deliver results of practical relevance, which was illustrated by the theme "Doing IT Research That Matters" of 2009's International Conference on Information Systems (ICIS) and the focus topic "Relevant rigor - rigorous relevance" of 2007's European Conference on Information Systems. The transfer of principles of design sciences from other domains, such as engineering, to IS research [1] meanwhile is considered to be a promising way of addressing the problem appropriately. Design-oriented IS research aims at delivering results which are of scientific rigor and of practical relevance at the same time [2]. An integral part of design-oriented research is to identify and describe a relevant practical

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 16–30, 2010.

problem, with the design-oriented IS researcher gaining access to the knowledge of practitioners, i.e. the “research environment” [1].

Second, existing research so far has provided only little guidance and support for gaining this kind of access. Peffers et al., for example, mention that resources required for this activity would include knowledge of the state of the problem and the importance of its solution [3], but do not specify this any further. And Guide and van Wassenhove e.g. discuss partnerships of researchers and practitioners on a very generic level [4].

Third, the “ecosystem” in which IS research is taking place and the roles of the actors within this ecosystem are under change. Today, research and innovation in the IS domain are largely taking place in the practitioners community [5], i.e. in user companies, in consulting companies, in software companies, and, increasingly, in companies providing electronic services. In order to be able to accomplish innovation, all these companies are using resources that are much larger and more powerful than the resources available in academic research institutions. As a consequence, business decision-makers tend to ask industry experts for help instead of addressing academic researchers. Like the CEO of a large Swiss bank explained to the authors: “When we face a problem, we look for the best consultants worldwide. University research is government’s business.” This corresponds to past research findings saying that it remains difficult for researchers to get access to high potential research topics [6].

Motivated by the observation of the aforementioned phenomena, the authors were asking themselves how consortium research relates to the current debate. Consortium research is a form of cooperative research between researchers and practitioners without exclusive usage rights. The consortium research method supports the development of artifacts. It has been developed by the authors based on the experience of almost twenty years of collaborative research in IS. It is characterized by close cooperation between an academic research institute and its partners in all stages of the design-oriented research process, practical validation of research results with partner companies, and a focus on the practical benefits of the research, with all research activities being funded by the consortium partners.

### 1.2 Research Question and Contribution

In this context, the research question addressed by the paper is: What is the contribution of the consortium research method to design-oriented IS research against the background of the aforementioned phenomena? The research question can be further detailed:

- What role does academic IS research play in the ecosystem and what benefit does it provide for companies?
- How can IS research gain access to the practitioners’ knowledge base? Do universities encounter difficulties when trying to access this knowledge base?
- Can consortium research in general facilitate and contribute to the attainment of practical knowledge, and, if so, under what conditions?

The first question focuses on the overall context in which the method is supposed to be applied whereas the second question aims at studying the purpose for which the method was developed. Finally, the third question aims at evaluating the consortium research method itself. This differentiation follows the "situational" notion of a method [7], i.e. its adaptability to specific project conditions.

The paper follows a qualitative empirical approach. It acknowledges the fact that regional differences exist in the world-wide IS community. Whereas the Anglo-Saxon community is rather following a behavioristic research paradigm, European, and in particular Central and Northern European, researchers have a long tradition in design-oriented IS research [2, 8]. In German the discipline is referred to as "Wirtschaftsinformatik". In this regard, expert interviews were conducted with eleven IS professors holding chairs at universities in Austria, Germany, and Switzerland.

The paper contributes to the body of knowledge by providing expert assessments of the role of academic IS research in relation to the practitioners' community and of the role of consortium research in this context. Section 2 of the paper outlines the background of the research in the fields of design-oriented research, its organization, and the exchange of knowledge between researchers and practitioners, before the consortium research method is introduced. Section 3 then introduces the research approach. Results of the expert interview series are presented in Section 4. The paper concludes with a summary and an outlook to future research in Section 5.

## 2 Background

### 2.1 Design-Oriented IS Research

A first framework for design-oriented IS Research was introduced by March and Smith in the mid 1990s [9], followed by guidelines for design-oriented IS research issued by Hevner et al. [1]. Based on this theoretical foundation, standards and processes have been introduced that are supposed to guide the researcher through the research process. Among them are the Design Science Research Methodology (DSRM) [3] and the concepts presented by Rossi and Sein [10].

On top of that, significant attention has recently been given to the evaluation of artifacts [11, 12]. Only little research, however, has been done to help researchers in the early activities within the design-oriented research process, namely problem identification and motivation, and definition of objectives for a solution. Gill and Battacherjee propose recommendations for the improvement of the researcher-practitioner relationship, but focus on bilateral and not multilateral collaboration [13].

Corresponding to the fact that little research is available regarding access to and exchange of knowledge in design-oriented IS research, literally nothing can be found regarding the organization of design-oriented IS research. Back et al. have outlined the compliance of the Competence Center (CC) concept at the Institute of Information Management at the University of St. Gallen with the guidelines of design-oriented IS research [14]. Broadening the scope of analysis, there are forms of organization which foster user integration in the design and development process in the area of technological innovation. "Living labs", for example, have evolved in recent years to evaluate and validate new IS solutions in close collaboration of solution providers and users [15]. By their nature, their focus lies mainly on instantiations.

### 2.2 Research Collaboration and the Transfer of Knowledge

In design-oriented disciplines, such as engineering, research cooperation of different actors along the value chain has a long tradition. Different forms of cooperation from the perspective of a user company can be distinguished by reference to exploitation rights and the relationship between cooperation partners (suppliers/customers, neutral partners, competitors) [16].

Social sciences, and in particular management research, have long been aware of a "relevance gap" within their discipline [5]. Pettigrew identifies the need for a re-engagement between researchers and practitioners to overcome this gap [17], forming a starting point for a movement which is referred to as "engaged scholarship" [18]. It assumes that research is a collaborative achievement between researchers and practitioners which relies, among others, on the joint advancement of knowledge.

Knowledge, in general, can be either "explicit" or "tacit". Whereas the former refers to a systematization of cognitive content, the latter is not systematized and is possessed by individuals only [19]. Of high relevance for the cooperation between academic researchers and practitioners is the conversion of knowledge from explicit to tacit and vice versa. Four types of knowledge conversion and knowledge transfer, respectively, can be determined [20, 21]:

- "Socialization" describes the tacit-to-tacit knowledge transfer. An example of this would be the transfer of experiences about stakeholders and organizational change management within an organization through a participatory action research project.
- The second type of conversion is "Externalization", in the process of which tacit knowledge is converted to explicit knowledge. An example of this would be the evaluation of design artifacts by focus groups and interviews including subsequent explication according to grounded theory principles by using e.g. coding techniques.
- The explicit-to-explicit knowledge transfer is referred to as "Combination". An example of this would be a joint researchers-practitioners project team in which researchers bring in their expertise on reference modeling and practitioners deliver well-documented business processes.
- "Internalization", as the fourth type, refers to the conversion of explicit knowledge to tacit knowledge. An example of this can be found in participatory action research and training sessions.

At present, only little research is available investigating the transfer of knowledge between researchers and practitioners in the domain of design-oriented IS research or the application of engaged scholarship principles to the latter. One of only few examples is the work by Mathiassen and Nielsen who analyzed the adoption of engaged scholarship in the Scandinavian IS community [22].

### 2.3 Consortium Research

Consortium research [23] as a method aims at the development of artifacts within a collaborative environment. It focuses on research areas in which no exclusive exploitation rights are desired by the research partners. As a consequence, it mainly addresses research topics which are to be investigated along a value chain or in co-operation with neutral partners, such as industry associations, standards bodies, or software companies. The method comprises four phases, namely "Analysis", "Design", "Evaluation" and "Diffusion" (see Fig. 1) which is in accordance with the principles for design-oriented IS research proposed by the "Wirtschaftsinformatik" community [24]. The "Domain" is the area in which the method is to be applied and in which it is supposed to yield new insights [25]. It includes both "practical" and scientific knowledge with the former typically being tacit knowledge [20]. Often, it is not produced according to scientific standards and is usually not well-documented (which is why the document symbols have dotted lines) [13].

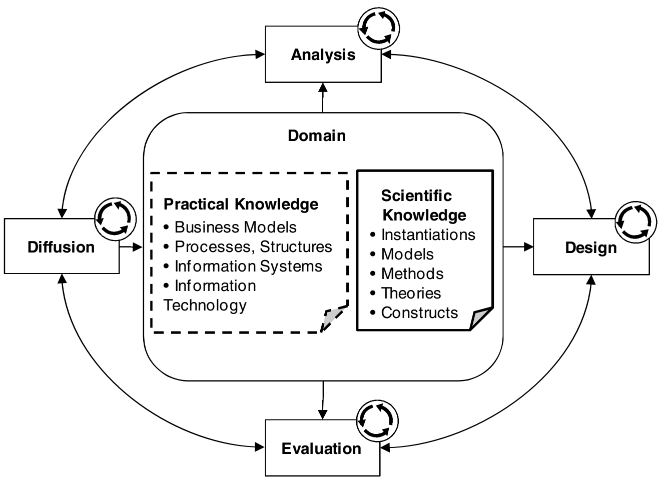



**Fig. 1.** Consortium Research

Consortium research refers to research projects in which a number of partner companies together with academic researchers work on a certain topic under the following conditions:

- Academic researchers and practitioners commonly define research objectives, assess progress of work, and evaluate project results.
- Research partner companies participate in research projects with their own experts and grant university researchers access to their knowledge resources.
- The results of the research are artifacts that offer substantial benefit for the companies participating.
- The companies participating test the artifacts developed in their business settings.
- The companies participating finance the research through money and human resources.
- The research results are made accessible to the public.

Consortium research uses different research approaches to transfer knowledge between academic researchers and practitioners from the partner companies (see Table 1). They are used according to the recommendations of existing inventories for IS research methods [26-28].

**Table 1.** Knowledge Transfer in Consortium Research

| "Socialization" (tacit → tacit) | "Externalization" (tacit → explicit) |
|---|---|
| Action research<br>Creativity techniques, such as morphological analysis | Case studies<br>Expert interviews<br>Focus groups<br>Grounded action research<br>Surveys |
| **"Combination" (explicit → explicit)** | **"Internalization" (explicit → tacit)** |
| Case studies<br>Content analysis<br>Market surveys | In-house seminars<br>Joint project teams |

Considering the portfolio of alternative forms of engaged scholarship proposed by Van de Ven [18], consortium research can be viewed as a combined instantiation of both the third and the fourth quadrant, namely "Design and evaluation research" and "Action/intervention research".

Due to space limitations this paper cannot further elaborate on method details. However, a full documentation is available as a working paper [23]. A case study illustrating and discussing the consortium research method has been accepted for presentation and publication at the 18th European Conference on Information Systems (ECIS 2010) [29].

## 3 Research Approach

The paper addresses the research question as to what is the contribution of consortium research to design-oriented IS research against the background of a number of phenomena observed. It follows a qualitative empirical approach and uses a series of expert interviews for data collection. An expert interview study is not based on a randomly selected sample, but rather on a group of intentionally selected individuals who have expert knowledge and, as a consequence, are of particular interest to the subject matter under study. Typically, expert interview studies involve only a small number of participants [30, 31].

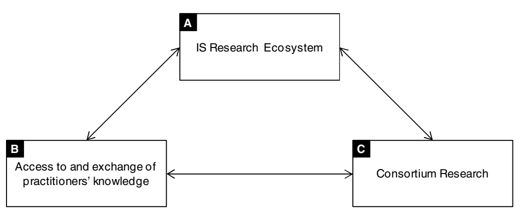



**Fig. 2.** Research Framework

Figure 2 shows the research framework underlying the expert interview survey. It illustrates the concepts and its interrelations as derived from the research question. Participants in the expert interview survey were eleven professors holding chairs in the field of IS at a university in the German-speaking countries in Europe. Table 2 shows the names of the experts, their affiliation, and the date and time of the interview. In addition to the experts participating, the authors invited another five experts of which one declined participation and four did not respond to the invitation.

**Table 2.** Expert Interviews

| Expert | Affiliation | Date | Time |
|---|---|---|---|
| Prof. Rainer Alt | University of Leipzig | 2009-07-29 | 15.00 - 16.00 |
| Prof. Walter Brenner | University of St. Gallen | 2009-07-22 | 10.00 - 11.00 |
| Prof. Hans Ulrich Buhl | University of Augsburg | 2009-06-19 | 12.15 - 13.15 |
| Prof. Elgar Fleisch | University of St. Gallen | 2009-09-07 | 11.00 - 12-00 |
| Prof. Ulrich Frank | University of Duisburg-Essen | 2009-07-21 | 15.15 - 16.00 |
| Prof. Lutz Heinrich | University of Linz | 2009-07-21 | 16.00 - 16.45 |
| Prof. Thomas Hess | LMU Munich | 2009-06-23 | 11.00 - 12.00 |
| Prof. Dimitris Karagiannis | University of Vienna | 2009-09-14 | 17.30 - 18.30 |
| Prof. Peter Mertens | University of Erlangen-Nürnberg | 2009-07-16 | 11.00 - 12.00 |
| Prof. Elmar Sinz | University of Bamberg | 2009-09-21 | 16.30 - 17.30 |
| Prof. Robert Winter | University of St. Gallen | 2009-06-22 | 11.00 - 12.00 |

Data collection was based on semi-structured interviews. A questionnaire was used consisting of ten open questions (see Appendix). Prior to the interviews, all experts were provided with a working paper describing the consortium research method and the questionnaire. The average duration of an interview was about one hour, all of them were tape-recorded. The recordings were then transcribed. Data analysis followed a stepwise approach [32]:

- First, transcribed data was paraphrased and condensed.
- Second, the data was compared to identify differences between and consensus among the experts with regard to certain questions.
- Third, the information was conceptualized following the principles of qualitative content analysis [33].
- Fourth, theoretical conclusions were drawn. Following Jarvinen's taxonomy of research methods, the paper falls in the category of "theory-creating approaches" [34]. More precisely, it aims at developing the foundation for an "explaining theory" [35].

The limitations of the study basically lie in its lack of representativeness, which holds true for expert interview studies in general, and in the specific selection of experts, which was mainly driven by the authors' subjective assessment regarding the competence and experience of the study participants in the subject matter investigated.

# 4 Result Presentation

## 4.1 IS Research Ecosystem

The role IS research is playing in the ecosystem was discussed intensively in all interviews. The question related to this issue explicitly referred to the interplay of researchers with industrial partners and software and consulting companies.

Two interviewees stated that the role of IS research had to be derived from its "mission". One expert said that the main objective of doing projects with practitioners was "to help practitioners and, what I judge equally important, to serve the discipline, to gain insight on its object of research". Moreover, it was mentioned that it was critical for IS research to "bring things forward". Despite the fact that this demand was undoubted, some interviewees explained that the means as to how IS research should do so were not clear for an integrative discipline such as IS. Since it combined computer science on the one hand and business economics on the other hand, it had to find the right balance between engineering and sociological approaches to research. The current debate about the epistemological foundation of the discipline was accompanied by a perception from practitioners which differed from that in our disciplines. One interviewee mentioned medicine and law as areas where it was common knowledge that "leading things take place at universities". He elaborated that this was totally different in IS.

Apart from that, it was stated as important to know the "rules of the game", and to know that they were different in the research community on the one side and the practitioners' community on the other side. One expert stated that synergies between them ranged in the area of "about 10 percent", and that at his chair he clearly distinguished between activities for the practitioners' community and activities for the research community.

Another concept was resources. It was commonly acknowledged that manpower was clearly bigger in the industry. Due to such limitations "IS research can only pick up individual questions". Also, in order to be taken seriously it was required to stick to a certain research topics for years. Otherwise, research would not be able to build up the knowledge to discuss on at "eye level".

One interviewee pointed out that the role of IS research changed with the "lifecycle" of research topics. He explained that research usually started with a technological invention and innovation. In this first lifecycle phase, IS research would help find innovative solutions. Also, in this phase academic IS research might be faster than industrial research because companies typically had to overcome barriers first. The second phase of the lifecycle was then characterized by applying innovation in certain domains, e.g. value chains. IS research would in these phases produce methods and reference models, for instance. In a third phase of the lifecycle, IS research focused on learning from applying the methods and models developed. The goal was to "finally extract fundamental concept", i.e. theorizing the findings.

Another question in this context related to the practical utility IS research is supposed to deliver. One expert said that originality of results was key for practical utility. Another interviewee explained that IS research had to find interdisciplinary answers to interdisciplinary questions. Considering that, IS research had to provide practitioners with methodologies and tools to solve their problems. This "means to an end" perspective on the outcome of IS research was mentioned by three participants in the study. One expert used the term "empowerment".

In this context, one interviewee agreed that such results only in few cases were exploitable for scientific purposes, and that it was up to "our 'parallel managers' of public and private funded projects to leverage the synergies".

On the other hand, this interviewee said that IS research did have advantages compared to industrial research because it "has a better research infrastructure: highly-skilled employees with latest methodological knowledge and thematic continuity".

The third question of the questionnaire explicitly referred to the result types IS research is expected to deliver. The answers given by the interviewees correspond with the design-oriented history of IS research in Central Europe. The list of responses includes (in alphabetical order):

- Business cases
- Case studies
- Concepts, constructs
- Expert assessments
- Evaluation
- Methods
- Prototypes
- (Reference) Models
- Reviews
- Simulations
- Surveys and studies

The list reflects the strong grounding of the experts in the design-oriented IS research approach. The majority of the interviewees considered artifacts as the main result type. However, critical opinions were articulated regarding the engineering roots of design-orientation. One expert stated that in engineering disciplines the proof of feasibility often was considered as artifact evaluation. He called it "somewhat frightening" that no attention was paid to economical implications. On the other hand, another interviewee warned of the technical sciences "decaying" to social sciences.

In contrast to the majority of the experts who tried to identify concrete examples of result types, one participant of the study tried to bring the question for result types into line with the overall purpose of science. In the trilogy of "describe", "explain", and "design", he argued, practitioners were interested in "everything relating to design". On the other hand, another expert stated that the development of common terminology had a practical value on its own, especially because the problems IS research is addressing are of interdisciplinary nature.

### 4.2 Access to and Exchange of Practitioner's Knowledge

The analysis of the data transcribed and the first coding revealed that items no. 4 and 7 of the questionnaire led to similar results. The questions as to how IS research can get access to practical knowledge and the question for solutions for potential problems in doing so were too closely related to each other, so that the concepts emerging from the data overlapped. Therefore, the two questions were merged into "How access practical knowledge”. Moreover, item no. 5 resulted in the same concepts as item 2. Obviously, “solutions” were subsumed under “utility”, being the broader concept.

The first concept regarding how practical knowledge could be accessed is “collaboration”. Three experts pointed out that a certain intensity in the collaboration between researchers and practitioners was needed in order to allow for access to the often tacit knowledge in the practitioners’ community. Moreover, it was stated that the supervision of master theses or infrequent interviews were not sufficient. Practitioners needed to make a clear commitment to the collaboration in order to make it productive. Examples of such a clear commitment could be the assignment of staff or the release of a budget to a joint project. Industrial partners’ funding of the project was considered a clear indication for the depth of the collaboration, because in each partner company someone “has to justify the Euros to be spent”. With regard to this, one interviewee pointed out that it was mandatory to offer “whatsoever incentives” to practitioners. Without, he argued, no access to experts in partner companies would be achieved.

Besides the involvement in joint research projects, networking ability was mentioned as a prerequisite for sound access to practical knowledge. Networks would enable researcher-practitioner relationships over a longer period of time compared to concrete research projects.

Two interviewees argued that collaboration with practitioners was not valuable per se. Instead, the researcher “has to catch the right persons”, those who were truly knowledgeable. This relates to the question regarding so-called “best practices”, which are often demanded by partner companies collaborating with academic researchers. The interviewees pointed out that it was often not easy to distinguish between “best practice” and “just good or moderate practice".

A significant part of the interview time was spent on the questions concerning problems in accessing and exchanging practitioners’ knowledge. One expert said that IS research first had to acquire a certain status (of expertise) before access is granted by partner companies.

Another concept in this context was scope. Often the scope of research collaboration is unclear, leading to expectations not met and decreasing confidence in researchers’ ability to solve problems relevant for practitioners.

Another concept mentioned was the necessity to have complementary goals. Quite often, goals differed, e.g. when researchers are interested in publishing the project results and practitioners want to keep the results confidential. One expert pointed out that the more innovative and original an outcome of a project was, the more unlikely was its publication. According to other interviewees, another conflict of interest lies in the different systems of evaluation criteria, with practitioners only being interested in economic and monetary evaluation, whereas these are evaluation criteria not satisfying rigorous scientific standards.

Apart from that, time was considered a problem when it comes to accessing and exchanging practical knowledge. This was true both for researchers and practitioners, with the researcher having limited time because of other goals he/she has or wants to achieve (academic career, teaching, academic self-administration etc.), and the practitioner demanding fast results, often at the cost of scientific rigor.

Finally, one expert said that different terminology and language in general hindered efficient collaboration between researchers and practitioners. He argued that as a researcher "if one does not speak the language of practitioners, he/she will not be able to understand the answers" to his/her questions.

### 4.3 Consortium Research

While the first two questions focus on the context in which the consortium research method is supposed to be used and the problem which it is supposed to solve, the third question deals with the consortium research method itself. Items no. 8 and 10 of the questionnaire were merged into one question. Almost all experts believe that consortium research makes a substantial contribution to accessing and exchanging practical knowledge. One of them stated that in IS research everything is allowed that pays off. "If it increases the body of knowledge or solves a problem, we will just use it." Two experts, however, were skeptical about the contribution of consortium research and said there was nothing special about it. Others stated that its "bundling effects" and its duration over a considerable period of time set it apart from comparable approaches, especially regarding research topics which were considered pre-competitive by partner companies.

Also, one expert mentioned that consortium research was an appropriate approach for "cross-topics", i.e. topics which require collaboration of companies along the value chain, across multiple corporate functions etc. In cases like these, consortium partners would benefit most from "many-to-many" collaboration.

Moreover, consortium research was considered by some interviewees to facilitate the transfer of knowledge not only from practice to research but also vice versa, allowing companies to "see if there is something new in research".

The last question dealt with the evaluation of the method design. The item in the questionnaire referred to enhancements, changes and deletions. A number of points were made by the experts here, one being that the consortium research method would not provide support in dealing with intellectual property rights. Another critical aspect referred to the method identifying the roles involved in doing consortium research, but falling short of explaining the role profiles sufficiently. A certain set of skills was needed for the post-doc researcher that goes beyond typical requirements in academic settings, among them project management, relationship management, and communication skills.

One expert said the method appeared to him like a combination of "a method for project management and a research approach", bearing the danger of not being understood by neither community. Consortium research probably might not meet the requirements of the research community nor the practitioners' community by "a hundred percent". This statement is in line with another comment describing consortium research as a meta-method which combines various different research approaches.

Another aspect which is missing relates to personal continuity. Especially publicly-owned companies tend to reorganize resulting in interrupted project involvement of partner companies. One expert wondered about how to deal with sequential vs. parallel iterations of design cycles. Since many companies were involved in the design of certain artifacts, he asked for the opportunities and limitations of parallel and sequential organization of design and evaluation activities, respectively.

One interviewee mentioned team size as a critical factor for an academic institution to do consortium research. In order to manage the consortium and produce rigorous and relevant results, a big enough team of researchers would be required.

Finally, one expert said the major constituent of consortium research's manifestation as a method was the stability it provides - with all its advantages and disadvantages. Among the advantages is planning security in terms of budget and results, whereas among the disadvantages is adherence to certain expected procedures or decision-making processes and inflexibility.

## 5 Discussion of Results

The data analysis results in nine theoretical categories. Three of them relate to the positioning of IS research in the ecosystem and its role in particular. Two categories have been identified for accessing and exchanging practical knowledge and for consortium research itself. Figure 3 shows the final concepts and categories. Certain concepts emerge from the data more frequently than others. An example is the lifecycle of research topics. It was discussed in the context of the IS research ecosystem as well as of potential scenarios for consortium research.

Another protruding concept relates to skills. Special skills are required to achieve research results of practical utility. And special expertise is needed for conducting consortium research since it combines project management and design-oriented IS research.

Moreover, a number of aspects have been identified for the advancement of the consortium research method. Among those are its demarcation from alternative design-oriented research approaches, the positioning in the lifecycle of research topics, the incorporation of necessary skills for the involved roles, and recommendations for appropriate consortium compositions.

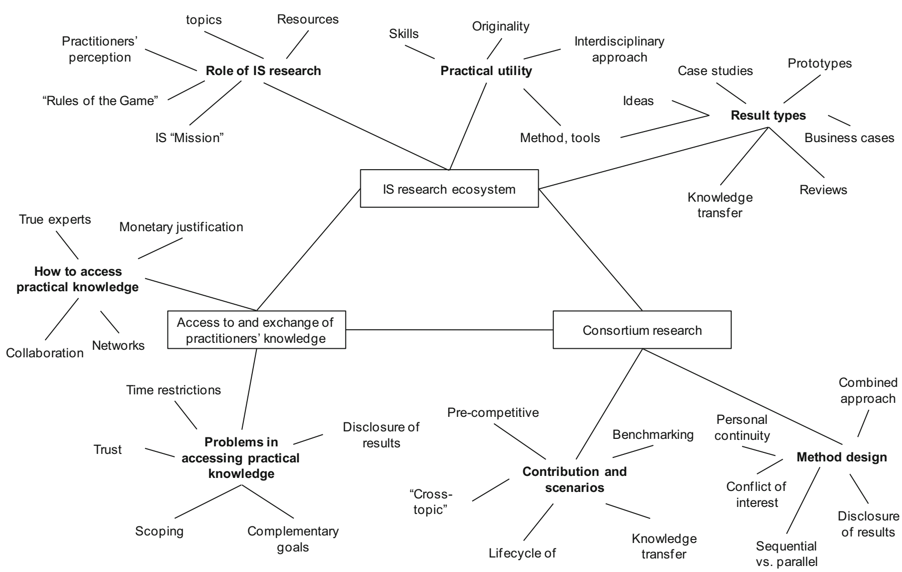



**Fig. 3.** Final concepts and categories

## 6 Summary and Outlook

The paper addresses the research question as to what consortium research can contribute to design-oriented IS research in the context of the current debate on the irrelevance of IS research and the changing research ecosystem. The paper presents and discusses the results of an expert interview study which was conducted among eleven professors holding chairs in IS research at universities in Austria, Germany, and Switzerland. In doing so, the paper focuses on the Central European form of IS research, known in German as “Wirtschaftsinformatik”. The paper investigates the context in which consortium research is applied and the problem it is supposed to solve, namely accessing and exchanging practitioners’ knowledge, before it evaluates the method itself.

The paper makes a contribution to the relevance of the discipline’s results. It shows that consortium research is not a panacea to the relevance debate, but that it combines certain constituents which facilitate access to and exchange of practitioners’ knowledge and, hence, support the early phases of design-oriented IS research. Apart from that, the expert interviews identified a number of aspects which need to be taken into account by any research method aiming at delivering research results of practical utility. Among those are the types of results required, required skills and competencies and potential problems which must be avoided.

The study, however, has some limitations. As the sample is relatively small, which lies in the nature of expert interview studies, generalizability of the results is not possible. Apart from that, while the selection of the experts was carried out randomly, it is based on the experience and assessment of the authors, i.e. it was subjectively biased. Also, the study only investigates the German-speaking IS research community. Its result cannot be transferred to other communities with different histories and self-conceptions.

Nonetheless, the study lays the foundation for further research. First, the findings from the interviews will be incorporated in the consortium research method. Second, Second, they will be mirrored against the perception of the practitioner’s community; in particular against the views of participants of former consortium research projects.. And third, the study might encourage future research aiming at the analysis of different design-oriented IS research approaches in terms of area of application, limitations, prerequisites etc.

## References


1. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information System Research. MIS Quarterly 28, 75–105 (2004)
2. Winter, R.: Design Science Research in Europe. European Journal of Information Systems 17, 470–475 (2008)
3. 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 (2008)
4. Guide, V.D.R.J., Van Wassenhove, L.N.: Dancing with the Devil: Partnering with Industry but Publishing in Academia. Decision Sciences 38, 531–546 (2007)
5. Starkey, K., Madan, P.: Bridiging the Relevance Gap: Aligning Stakeholders in the Future of Management Research. British Journal of Management 12, S3–S26 (2001)

6. Benbasat, I., Zmud, R.: Empirical Research in Information Systems: The Practice of Relevance. MIS Quarterly 23, 3–16 (1999)
7. Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. Information and Software Technology 38, 275–280 (1996)
8. Frank, U., Schauer, C., Wigand, R.T.: Different Paths of Development of Two Information Systems Communities: A Comparative Study Based on Peer Interviews. Communications of the AIS 22, 391–412 (2008)
9. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
10. Rossi, M., Sein, M.K.: Design Research Workshop: A Proactive Research Approach. Design Research Workshop within the IRIS26 (2003)
11. Frank, U.: Evaluation von Artefakten in der Wirtschaftsinformatikforschung. In: Häntschel, I., Heinrich, L.J. (eds.) Evaluation und Evaluationsforschung in der Wirtschaftsinformatik, pp. 35–48. Oldenbourg, München (2000)
12. Bucher, T., Riege, C., Saat, J.: Evaluation in der gestaltungsorientierten Wirtschaftsinformatik - Systematisierung nach Erkenntnisziel und Gestaltungsziel. In: Becker, J., Krcmar, H., Niehaves, B. (eds.) Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik. Arbeitsbericht Nr. 120 des Instituts für Wirtschaftsinformatik, pp. 69–86. Universität Münster, Münster (2008)
13. Gill, G., Bhattacherjee, A.: Whom Are We Informing? Issues and Recommendations for MIS Research From an Informing Sciences Perspective. MIS Quarterly 33, 217–235 (2009)
14. Back, A., von Krogh, G., Enkel, E.: The CC Model as Organizational Design Striving to Combine Relevance and Rigor. Systemic Practice and Action Research 20, 91–103 (2007)
15. Følstad, A.: Living Labs for Innovation and Development of Information and Communication Technology: A Literature Review. The Electronic Journal for Virtual Organizations and Networks 10, 99–131 (2008)
16. Brockhoff, K.: Forschung und Entwicklung: Planung und Kontrolle. Oldenbourg, München (1999)
17. Pettigrew, A.M.: Management Research After Modernism. British Journal of Management 12, S61–S70 (2001)
18. Van de Ven, A.H.: Engaged Scholarship. Oxford University Press, Oxford (2007)
19. David, P., Foray, D.: Accessing and expanding the science and technology knowledge base. In: Organisation for Economic Co-operation and Development (OECD), Paris (1994)
20. Rynes, S.L., Bartunek, J.M., Daft, R.L.: Across the Great Divide: Knowledge Creation and Transfer between Practitioners and Academics. Academy of Mangement Journal 44, 340–355 (2001)
21. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, Oxford (1995)
22. Mathiassen, L., Nielsen, P.A.: Engaged Scholarship in IS Research: The Scandinavian Case. Scandinavian Journal of Information Systems 20, 3–20 (2008)
23. Österle, H., Otto, B.: A Method For Consortial Research. Institute of Information Management, University of St. Gallen, St. Gallen (2009)
24. Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.J.: Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. In: Österle, H., Winter, R., Brenner, W. (eds.) Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz. infowerk ag, pp. 1–6 (2010)
25. Nunamaker, J.F., Chen, M., Purdin, T.D.M.: Systems Development in Information Systems Research. Journal of Management Information Systems 7, 89–106 (1991)

26. Cavana, R.Y.: Applied business research: qualitative and quantitative methods. John Wiley & Sons, Australia (2001)
27. Ethridge, D.: Research methodology in applied economics: organizing, planning, and conducting economic research. Iowa State University Press, Ames (1995)
28. Wilde, T., Hess, T.: Forschungsmethoden der Wirtschaftsinformatik. Eine empirische Untersuchung. Wirtschaftsinformatik 49, 280–287 (2007)
29. Otto, B., Österle, H.: Relevance Through Consortium Research? A Case Study. In: 18th European Conference on Information Systems (ECIS 2010), Pretoria (2010)
30. Knödler, T.: Public Relations und Wirtschaftsjournalismus: Erfolgs- und Risikofaktoren für einen win-win. VS, Verlag für Sozialwissenschaften, Wiesbaden (2005)
31. Meuser, M., Nagel, U.: Expertenwissen und Experteninterview. In: Hitzler, R., Honer, A., Maeder, C. (eds.) Expertenwissen. Die institutionelle Kompetenz zur Konstruktion von Wirklichkeit, pp. 180–192. Westdeutscher Verlag, Opladen (1994)
32. Meuser, M., Nagel, U.: ExpertInneninterviews - vielfach erprobt, wenig bedacht. Ein Beitrag zur qualitativen Methodendiskussion. In: Bogner, A., Littig, B., Menz, W. (eds.) Das Experteninterview. Theorie, Methode, Anwendung, pp. 71–93. Leske und Budrich, Opladen (2002)
33. Mayring, P.: Qualitative Inhaltsanalyse. Forum Qualitative Sozialforschung/Forum: Qualitative Social Research 1, Art. 20 (2000)
34. Jarvinen, P.H.: Research Questions Guiding Selection of an Appropriate Research Method. In: Proceedings of the 8th European Conference on Information Systems, pp. 124–131 (2000)
35. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30, 611–642 (2006)

## Appendix: Questionnaire

| ID | Question |
|---|---|
| A | Role of IS research in the ecosystem |
| 1 | What is the role of IS research in the interplay with user companies, software vendors, consulting companies etc.? |
| 2 | What benefit can IS research deliver to the practitioners' community? |
| 3 | What are appropriate result types? Can these result types be disseminated in the scientific community? |
| B | Access to and exchange of practitioners' knowledge |
| 4 | How does IS research get access to practitioners' knowledge? How can it be studied, analyzed and evaluated at reasonable cost? |
| 5 | Is IS research capable of delivering new solutions to the practitioners' community? |
| 6 | Is access to the practitioners' knowledge a problem for IS research? |
| 7 | If so, what are potential solutions? |
| C | Contribution of consortium research |
| 8 | Can the method for consortium research contribute to the solution of the problem? |
| 9 | How to assess the method for consortium research? What should be enhanced, changed, or deleted? |
| 10 | What are appropriate usage scenarios for the consortium research method? What are limitations? |

# Taking a Project Management Perspective on Design Science Research

Jan vom Brocke[1] and Sonia Lippe[2]

[1] University of Liechtenstein
Fuerst-Franz-Josef-Strasse 21, 9490 Vaduz, Principality of Liechtenstein
jan.vom.brocke@hochschule.li
[2] SAP (Switzerland) Inc., Blumenbergplatz 9. 9000 St. Gallen, Switzerland
sonia.lippe@sap.com

**Abstract.** In this paper we develop a project management (PM) perspective on design science research (DSR). We account for the increasing amount of DSR projects that are emerging in public-private research collaborations and that align both business needs and research rigor. In addition to the application of sound research methodologies, the successful management of the work relations constitutes an important success factor for DSR projects. Hence the need emerges for professional project management. However, certain features such as creativity, uncertainty in terms of the research method and outcome, and research rigor complicate the application of standard PM approaches and make certain adaptations necessary. The goal of this paper is to identify a set of characteristics specific to DSR projects and to analyse their implications for selecting and adapting established project management standards. For evaluation purposes, we are using the PMBOK® Guide by the Project Management Institute which is commonly accepted and widely used in practice and academia.

**Keywords:** Design Science Research Project, Project Management, Contingency Frameworks.

## 1 Introduction

Design science research (DSR) has proven successful in Information Systems and other disciplines in aligning both business needs and research objectives. The essential element of this approach is to pick relevant problems from business, design rigorous solutions in academia and bring them back to practice in order to evaluate the result [1]. A key characteristic in DSR is thus the relevance with respect to the targeted community of mostly practitioners [2, 3]. This specific focus on industry problems has lead to an increase in larger projects outside the purely academic world. As an example, most public funded collaborative research projects that are executed jointly between industry and academia follow the paradigm of design science to conduct research. In this kind of project, research is conducted by a heterogeneous consortium established loosely through the common research interest and the availability of a funding opportunity (e.g. projects funded by the European Commission). The objectives are driven by the involved end-users, while the academic partners are responsible for the rigorous research

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 31–44, 2010.

process and the novelty of results [4]. The project volume for public funded projects varies between 500.000 euros up to over 30 million euros per project with the overall funding volume steadily increasing[1].

Together with the increased volume, size and industry involvement, also emerges the need for more professional management of DSR projects. The implementation of a tailored project management method constitutes an important success factor, in addition to widely discussed methodological issues and guidelines. A major challenge in this respect is the adaptation of conventional knowledge and practices to the specific project needs [5-7]. Within the area of project management, certain standards have been established which provide proven traditional practices such as methods, techniques, processes and guidelines. The most prominent and widely used are the PMBOK® Guide by the Project Management Institute [8] and PRINCE2 [9]. However, these standards originate mainly from the execution of large defence and engineering projects. Based on the assumptions of a basic similarity between all projects and that both the concrete outcome and the steps of the work plan are well understood at the start of a project, they focus on detailed plans of actions to complete a project on time, on budget and within a given scope [10]. In comparison, DSR projects are characterised by the novelty of results (research contribution) and a search process where multiple solutions might be evaluated [2]. In consequence, their management turns out to be a unique challenge, as we are facing a significantly different situation compared to "conventional" projects and existing standards cannot be applied without certain adjustments. The goal of this paper is to determine how and to which extent the implementation of an existing standard is influenced for projects that follow the DSR methodology. We aim at answering the following research question:

*What are the implications of the design science research paradigm on the application of a conventional project management standard?*

Section two starts by giving a state-of-the-art overview on the management of research projects and summarises the current discussion on the relationship between specific project characteristics and the application of project management practices. We then determine the characteristics of research, and in particular DSR, projects from a project management perspective (section 3). These will be evaluated in terms of their significance in section four by analysing implications for using an established project management method (PMBOK® Guide by the Project Management Institute). We conclude with a summary and an indication of further research directions.

## 2 The Need for Tailored Project Management in DSR

Arising from the accretive "professionalisation" of DSR, there is a clear need for the application of professional management approaches. Effectively used, PM offers support in aligning research relevance for industry partners and rigor for academic partners equally and ensures the completion of the projects on time and within the

---

[1] The EU has subsequently increased the funding for each Framework Programme since their start in 1984 and the currently running 7th Framework Programme has a budget of 50.5 billion euros (http://cordis.europa.eu/fp7/budget_en.html)

given budget. Yet, research projects have rarely been considered in project management literature as opposed to the more traditional fields of construction, engineering and software development [11] and only few authors directly focus on their management. Erno-Kjolhede and Clarke elaborate on the large discrepancy between the nature of researchers and the strict formal processes and tasks required for professional project management and conclude that academic behaviour needs to be driven differently from the human resource management of other project types [4, 12]. In addition, Erno-Kjolhede assessed existing planning and scheduling techniques against certain requirements of research projects [4]. To be used successfully, these techniques need to be applied more flexibly than originally intended. Brown deals with the question whether research can be "project managed" and formulates some ground rules for the fruitful application of project management to research organisations [13]. In summary, all authors argue that research projects show particular features which complicate the application of existing PM practices and that their management is thus a crucial, yet often neglected task. A comprehensive examination of the project type and a detailed analysis of the implications on existing, widely-used PM standards is however missing, in particular for DSR projects.

In recent years project management has experienced a paradigm shift from the early assumption that "a project is a project" and can be handled via a uniform management approach, to a wide recognition of the variability of methods according to project type and contextual factors [5, 14, 15]. Payne & Turner show in a study that methods which are commonly used across various projects, although benefiting from standardisation, were generally rated less successful than the use of project specific adoptions [14]. However, they also stress the usefulness of company-wide standards for certain project types. This is supported by Besner & Hobbs who performed a "reality check" in currently-running projects and identified the use of common patterns and standards along with significant variations in project management practice [16]. Again, the perceived usefulness of common standards vs. tailored approaches depended on certain project attributes, such as the application area, size, and maturity of the executing organisation. Hence, a logical consequence for each project is to conduct an upfront analysis of the project type before deciding on the PM approach [15]. This will identify possible constituent characteristics that require and thus justify the additional effort of adapting and extending existing standards and methods.

Largely absent are subsequent studies on the relationship of project characteristics and suitable project management approaches and techniques [7]. No guidelines are given on how to proceed in finding a suitable method once the project type has been determined. Also, only a minor percentage of the contingency frameworks derive usable management recommendations for the identified project categories. Vom Brocke and Lippe analysed the recommendations given for the project type of collaborative research projects and concluded that recommendations are in general on a very high level and miss a link to concrete implementations of project management standards [17]. Within this paper we will close this gap for DSR by analysing the implications of each characteristic on a common standard. The most institutionalised are PRINCE2 by the Office of Government Commerce and the PMBOK® Guide of the Project Management Institute. In this case we have chosen the PMBOK® Guide out of the following reasons:

- It claims to provide fundamentals irrespective of the project type and to be a general guide to manage most projects most of the time [8].
- It is less business case driven than PRINCE2 and thus naturally more suitable to research projects.

We will assess the suitability and limitations of this widely accepted standard and thus confirm the assumption that DSR projects significantly differ from many other project types, in particular the more conventional ones for which most standards and best practices were created and thus lack sufficient PM support [4, 12, 18]. This will moreover serve as an evaluation of the identified characteristics and show to which extent they cover the uniqueness of DSR projects.

## 3 Project Type Characterisation for Design Science Research Projects

The goal of this section is to determine the essential characteristics of DSR projects and to identify those that significantly intensify the need for tailored project management. The identification is based on a literature review of design science and project management papers, existing classification and contingency frameworks and the guidelines for DSR of Hevner et al. It is not our intention to contribute to the ongoing discussion in DSR on the consistency of the methodology, the usability of reference process models, or the acceptance of certain outcomes or evaluation steps, but to use the various sources in order to sketch a complete picture of the project type.

### 3.1 Characteristics Derived from Project Management and Design Science Research Literature

The Project Management Institute (PMI) defines a project as "a temporary endeavour undertaken to create a unique product or service or result" [8]. It elaborates a little more with defining three major characteristics that are common to all projects:

- First, they are temporary which means that "every project has a definite beginning and a definite end" [8]. This certainly also applies to DSR projects. The usual timeframe for e.g. of public funded projects in Europe lies between 30 to 60 month and their end is fixed in the beginning through a contract with the European Commission. Also bilateral projects between academic and industry partners operate within a fixed timeframe.
- Second, they are unique in a sense that the outcome, organisational set up, location, etc. is changing for each project. This also holds for DSR projects, where different partners, scope, locations etc. determine each project set up.
- Third, the outcome of a project is a product or a service. Here DSR projects differ significantly. Lacking the commercial background, they neither directly deliver market-ready products nor services. The results however are often commercialised afterwards, which is not considered as being a research project as such and will not be discussed in this paper. The outcome of DSR projects are solutions for IS-related problems. Accepted solutions in the IS community are constructs, models, methods or instantiations [1, 3] and to a certain extent also theories [19].

The above definition of the PMI originates in the execution of large defence and engineering projects and it reflects their specific challenges and perspectives [10]. A similar focus can be found in the PRINCE2 definition of a project as a "temporary organization that is needed to produce a unique and predefined outcome or result at a pre-specified time using predetermined resources" [9]. The assumption is that both the concrete outcome in terms of a service or product and the method of creating them are well understood at the start of a project. This assumption needs to be taken a closer look at in the light of DSR projects as we are facing a different situation [20].

Focussing on business needs in relation to IS and aiming at "utility" as opposed to "truth" [3], DSR is fundamentally a problem-solving paradigm [2]. The goal is to solve an existing problem in the business world and what is known at the beginning of the project is the research question [21]. The corresponding solution however is unknown, except that it will be of a certain type (construct, model, method, instantiation or theory). In comparison, when e.g. dealing with civil engineering projects, most details of the envisaged building will be specified before the project start. Hence, additional characteristics of a DSR project are

- their problem-solving nature: in DSR two problems are mutually nested, namely practical problems which occur in the real world and knowledge problems which only change the knowledge base [22]. While practical problems are solved through changing the world based on stakeholder needs, knowledge problems are solved only through formulating propositions on the world [23], which naturally implies a different problem-solving process and evaluation criteria.
- an unknown outcome: compared with traditional projects, the concrete outcome of a research project is uncertain at the project start. Determining what kind of solution would best fit the inherent problem is a major task of the research effort and thus the outcome can only be determined during the course of the project and might be changed based on evaluation results.

Not only the research results are uncertain, but also the steps towards the results are mostly ill-defined at the beginning of the project. Again, DSR is on one hand characterised by the existence of well- defined and established set of plans and procedures in order to achieve scientifically acknowledged research rigor and on the other hand by substantial uncertainty in terms of the detailed steps to be performed within the project. There is an ongoing discussion within DSR literature on common reference process models for the research method [1-3, 24]. However, from a project management perspective it is nearly impossible to further breakdown these steps into smaller work-packages and to perform a detailed planning based on decomposition. To determine the exact working method is also part of the research itself and can thus only be done during the course of the project. However, it is not the case that DSR projects are totally without routine steps that can be planned and even standardised for various projects. Moreover, the combination of creative and thus unpredictable work with regular and thus pre-defined tasks is a key feature of any research project. The iterative, problem solving nature of DSR projects, which requires the combination of generate and test phases should also be added here [2]. The generate phases would fall into the category of creative tasks, whereas test activities are much more foreseeable and thus can be planned in advance.

### 3.2 Characteristics Derived from Project Management Contingency Frameworks

Within the project management literature, certain classification and contingency frameworks have been developed with the goal to categorise existing projects, determine similarities as well as differences and to suggest corresponding management styles. A general overview of existing frameworks is given in Crawford, Hobbs & Turner (2006) and Sauser, Reilly & Shenhar (2009) [25, 26]. Vom Brocke & Lippe applied selected frameworks to classify collaborative research projects and discuss resulting management guidelines [17]. They show that while no commonly accepted framework exists to be solely used and each framework addresses a certain perspective, collectively they offer a formal framework to categorise and describe research projects. The following table shows their findings together with characteristics that can be derived for DSR projects.

**Table 1.** Analysis of Project Management Contigency Frameworks based on [17]

| **Authors / Year** | **Description of classification approach** | **Classification of research projects and constituent characteristics for DSR projects** |
|---|---|---|
| Turner & Cochrane, 1993 [20] | 2x2 matrix that classifies projects according to the level of goal and methods definition. | DSR projects are classified as type 4 projects for which the goals and the methods of achieving them are ill defined. In terms of the characteristics it confirms the above described uncertain outcome and unknown research steps. |
| Shenhar & Dvir (2007) [27] | Multidimensional framework that classifies projects based on novelty, technology, complexity and pace (NTCP framework). | DSR projects are of regular pace and high-tech technology. Although often dealing with topics than can determine the long-term strategy of a company, they are not immediately critical to the organisational success and thus might experience less (frequent) management attention. The degree of complexity and novelty depends on the individual project scope. |
| Crawford & Pollak, 2004 [28] | Differentiate projects based on 7 dimensions related to hard and soft factors. | DSR projects as mostly “soft” projects. They are characterized through the definition of a research question rather than a concrete outcome, usually explore various possible solutions and will rather produce concepts than physical artefacts. Also the success measures are rather qualitative than quantitative. |

Jordan et al. offer a classification framework solely for research projects and categorise along three attributes, namely the complexity and size of the project, task and team. They identified four archetypes of research projects [29]: be new, be first, be better and be sustainable. For each project type they compare certain attributes to assess the differences within the project types. Concerning the characteristics of DSR projects, the following input can be derived from their study:

- The measures of success vary between various stakeholders of the projects. This is especially evident for DSR projects because of their strong involvement of industry partners. Academic partners usually rate a research project successful once the solutions have been conceptualised, evaluated and communicated within the academic community. This often leads to results being only “on paper” or partially implemented as a proof of concept, but additional implementation steps are not considered as being a research task. To be valuable and thus successful for industry partners it is however necessary that a DSR project produce working solutions of IT artefacts. Just "knowing" that a proposed artefact constitutes an improvement is not sufficient; the improvement needs to be implemented. This is particular a problem when the expected outcome of the DSR project is a construct or a model and thus naturally less tangible to the end user [3].

- A high amount of creativity within the work performed is characteristic to an environment that fosters excellence research.

### 3.3 Characteristics Derived from DSR Guidelines of Hevner et al.

Hevner et al. have developed common guidelines to conduct and report DSR and thus contribute to the twofold discussion on research relevance vs. research rigor [2]. The importance and acceptance of this work within DSR practice asks for a more detailed discussion of the guidelines. We will thus compare them to the so far identified characteristics for a final gap analysis (cp. table 2). This is expected to complete the overall investigation of DSR projects and their characteristics within this paper.

**Table 2.** Analysis of Guidelines for Design Science Research [2]

| Guideline | Coverage in 3.1 and 3.2 |
|---|---|
| Design as an artefact | Qualitative success measures to measure results |
| Problem relevance | Problem-solving nature |
| Design Evaluation | - |
| Research contribution | - |
| Research rigor | Existence of common reference process models in DSR literature for the research method |
| Design as a search process | Unknown outcome; Combination of creative, unpredictable work with regular, thus pre-defined tasks |
| Communication of research | - |

The above table shows shortcomings in our discussion with respect to three design guidelines, namely design evaluation, research contribution and communication of research:

*Design evaluation*: Evaluation methods constitute an important component of the research process and a high percentage of evaluation and testing activities need to be performed within the project scope. In fact, it is an essential part of any research activity to demonstrate the utility, quality, and efficacy of a research result [2].

*Research contribution*: The goal of any research activity is to extend the existing state of the art by creating new knowledge for a certain field of study [21]. In DSR this is done by in addressing relevant business requirements in novel or innovative ways. Only if results are able to prove novelty in terms of solving a previously unsolved problem, they are recognized as scientific [30]. Thus, novelty of results can be identified as a central characteristic of any research project and is seen as the key distinguishing feature between design research and design practice [31]. Additionally to the novelty, the generality of results is a second important aspect. The developed results need to be applicable not only within the given project setting, but researchers need to generalise from the given case to a broader set of problem situations.

*Communication of research*: The communication of the results, to technology-oriented as well as management oriented audiences is an important step within the project.

### 3.4 Summary of DSR Characteristics

The following table summarises the above identified characteristics which are defining for DSR project.

**Table 3.** Summary of Project Characteristics for DSR

| Number | Characteristic |
|---|---|
| C1 | Problem-solving nature |
| C2 | Unknown outcome |
| C3 | Novelty and generality of results |
| C4 | Combination of creative, unpredictable work with regular, thus pre-defined tasks |
| C5 | High percentage of evaluation and testing activities |
| C6 | The measures of success vary between various stakeholders |
| C7 | Qualitative success measures |
| C8 | Not immediately critical to the organisational success |

Erno-Kjolhede describes managing research projects as a balancing act of the distinct nature of research work:

> "To create an innovative research project the almost schizophrenic balance to be stuck is then on the one hand to create an atmosphere that facilitates the creativity and innovation associated with risk taking and on the other hand at the same time working hard to avoid failures stemming from such risk taking in the project." [4]

Further paradoxes have been identified within this paper:

- Generally accepted outcome types of DSR are extensively discussed and defined within the community. However it is nearly impossible to define the exact outcome of a certain project at its start as each project is characterised by novel results and an iterative problem-solving process which will design and evaluate various possible solutions.
- Reference process models and frameworks define research steps which need to be followed to comply with the rigorous guidelines of DSR. However, project execution is characterised by interplay of routine tasks which are common to most projects mixed with complex research steps for which further decomposition into dedicated work packages is nearly impossible and which require a large amount of creative thinking.

A suitable management method needs to be tailored at balancing these paradoxes in the best possible way.

# 4 Implications on Project Management

## 4.1 Introduction to the PMBOK® Guide

The Project Management Institute published the first version of "A Guide to the Project Management Body of Knowledge" (PMBOK® Guide) as a white paper in 1987. It was the first attempt to document and standardize generally accepted project management knowledge and practices. Since then various editions have been published, the content has been constantly extended and it is accepted as an internationally recognized standard (IEEE Std 1490-2003[2]).

The PMBOK® Guide is process-based and describes processes in terms of inputs (documents, plans, designs, etc.), tools and techniques (mechanisms applied to inputs) and outputs (documents, products, etc.). The guide recognizes 44 project management processes that fall into five basic process groups and nine knowledge areas that are typical in almost all projects.

- The five process groups are: Initiating, Planning, Executing, Controlling and Monitoring, and Closing.
- The nine knowledge areas are: Project Integration Management, Project Scope Management, Project Time Management, Project Cost Management, Project Quality Management, Project Human Resource Management, Project Communications Management, Project Risk Management, and Project Procurement Management.

This creates a matrix structure of process groups and knowledge areas in which each process is mapped such that every process can be related to one knowledge area and one process group.

## 4.2 Evaluation of DSR Characteristics

Within our research we have performed a comprehensive analysis of all processes in respect to the above identified characteristics. Five different projects were evaluated in terms of the difficulties with implementing the PMBOK® Guide. Data collection was based on project documentation and interviews with project personnel. For each knowledge area we collected the identified problems in respect to each characteristic.

An overview of the assessment is shown in figure 1 using Harvey balls. A full ball represents strong implications in respective to the standardised tools and outcomes of the PMBOK® Guide. The proposed processes are not directly usable and require adaptations and extensions. A half-full ball indicates minor implications in terms of how the standard can be used. Customisation is necessary, however only to a very small extend and most of the standard can be directly applied. No ball indicates that this characteristic has no impact. Figure 1 shows a bigger impact in knowledge areas that are dealing with the project scope, quality and risk. This is mostly due to the fact that the outcome is the most uncertain and changing variable of DSR projects as opposed to end date and costs which are (e.g. in public funded projects) totally fixed in the beginning of the project. The challenge is often not to align scope, budget and time as in classical project management, but to solve as much of the identified problem as possible within a given time- and cost-frame.

[2] http://standards.ieee.org/reading/ieee/std_public/description/se/1490-2003_desc.html

| No | Inte-gration | Scope | Time | Cost | Quality | HR | Commu-nications | Risk | Procure-ment |
|---|---|---|---|---|---|---|---|---|---|
| C1 | ● | ● | ◗ | | ● | ◗ | | ● | |
| C2 | ● | ● | ● | ◗ | ● | | ◗ | ● | ◗ |
| C3 | ● | ● | ◗ | ◗ | ● | ◗ | ◗ | ◗ | |
| C4 | ● | ● | ● | ◗ | ◗ | ◗ | | ● | |
| C5 | ● | ● | ● | | ◗ | | | ◗ | |
| C6 | ● | ● | | | ● | | ● | ● | ◗ |
| C7 | ● | ● | | | ● | | | | |
| C8 | ● | | | | | ◗ | | | |

**Fig. 1.** Evaluation of DSR Characteristics

Project integration management covers the high-level processes that are required for the overall coordination of the project. It is mainly concerned with effectively integrating all process groups and is mostly impacted as it ties together all project management work of the other knowledge areas.

In the following we will thus focus on the four knowledge areas that are mostly impacted besides integration management, namely project scope management, project quality management and project risk management and show the details of the analysis for those characteristics which indicate a full Harvey ball.

**Project scope management** ensures that the work is performed as required and is “primarily concerned with defining and controlling what is and what is not included in the project” [8].

**Table 4.** Implications on Project Scope Management

| No | Characteristic |
|---|---|
| C1 | *Problem-solving nature*<br>• Scope definition should follow a two step approach:<br>1. Definition of research questions and objectives as opposed to detailed product descriptions during planning.<br>2. Scope description of the envisaged solution during each design step.<br>• The definition of a problem scope and related questions is never as accurate as a detailed product description.<br>• The tools and techniques for scope verification and scope control need to rely on expert judgement and continuous communication of the stakeholders as opposed to quantifiable measurement techniques.<br>• Change control needs to distinguish between two kinds of scope changes and act accordingly:<br>1. Changes to the underlying problem and thus the research question (change to “scope 1). These are scope changes which correspond to the PMI definition and thus require a well defined integrated change control process.<br>2. Changes to the developed solution, which are an apparent step of any problem-solving processes where different solutions are developed and evaluated (change to “scope 2). |

**Table 4.** (*Continued*)

| | |
|---|---|
| C2 | *Unknown outcome*<br>• Scope definition largely dependent on expert judgement instead of "hard" techniques.<br>• Detailed product analysis or alternatives identification not possible at the start of the project. |
| C3 | *Novelty and generality of results*<br>• Completion of the scope needs to be measured against scientific criteria in addition to the end-user requirements. Additional steps might be required to prove novelty and generality.<br>• Changes in the knowledge base need to be closely monitored and considered as part of the scope control process. Although they would not directly impact the relevance, their impact on the research rigor would be considerable. |
| C4 | *Combination of creative, unpredictable work with regular, thus pre-defined tasks*<br>• Effort expended on scoping varies not on project level but on task level.<br>• Work breakdown structure needs to be "softened" for certain work packages and allow sufficient time to "think and explore", as rigid planning and control is counterproductive for creative phases. |
| C5 | *High percentage of evaluation and testing activities*<br>• Interplay of design and testing leads to continuous re-planning of the solution scope (scope 2). The project scope management plan needs to account for this. |
| C6 | *The measures of success vary between various stakeholders*<br>• To perform a stakeholder analysis that identifies the influences and specially the interests of the various stakeholders is a key task in project scope definition. The non-academic partners need to be made aware of the risks associated with research results and that solving research questions can also result in a negative answer. Especially the definition of assumptions and constraints should be emphasized and a description of the research nature needs to be included in the scope statement. |
| C7 | *Qualitative success measures*<br>• The main goal of project scope management is to define measurable objectives and corresponding deliverables during scope definition and confirm them during scope verification. This requires a definition of KPIs which take into account the qualitative nature of the success measures and reach an agreement of all stakeholders. |

**Table 5.** Implications on Project Time Management

| No | Characteristic |
|---|---|
| C1 | *Problem-solving nature*<br>• Impossible to determine detailed activity list at the project start, however to comply with the research guidelines the overall research process should be sketched and communicated. Time management can then work on milestones and dependencies for each phase and some detailed planning of routine task. However, activity estimation and sequencing techniques are only of limited use. |
| C2 | *Unknown outcome*<br>• PM techniques that rely on experience within similar previous projects and existing templates are hardly usable. |
| C4 | *Combination of creative, unpredictable work with regular, thus pre-defined tasks*<br>• Rolling wave planning needs to adapted and include "white spots" for creativity-intensive tasks that cannot be further broken down<br>• Usage of milestone list instead of activity list for creative phases.<br>• Activity estimating should contain fixed deadlines for creative phases.<br>• Schedule development tools and techniques un-usable as they are based on the assumption that the start and end date of schedule activities can be estimated which is not the case for creative tasks.<br>• Complex panning hinders creative tasks and the researchers working on them. |
| C5 | *High percentage of evaluation and testing activities*<br>• Number of work periods depends on amount of solutions and refinements that are necessary to solve the research problem and thus cannot pre-defined.<br>• Concurrent planning for each cycle necessary that is performed in conjunction with a refinement of the scope statement. |

**Project time management** manages the processes that ensure a timely completion of the project scope by managing the project activities.

**Project quality management** aims at delivering high quality project results that satisfy the originally intended need of the project. It includes the planning of quality management processes and tools as well as the actual execution of quality assurance and control.

**Table 6.** Implications on Project Quality Management

| No | Characteristic |
|---|---|
| C1 | *Problem-solving nature*<br>• Inherit search process needs to implement quality gates which are specified in the quality management plan.<br>• Reviews need to be scheduled in accordance with the research progress rather than at fixed points in time. |
| C2 | *Unknown outcome*<br>• Hindered application of established quality planning tools and techniques as most are based on a product definition. |
| C3 | *Novelty and generality of results*<br>• Similar to the scope management is an important factor in quality management the satisfaction of the research rigor. |
| C6 | *The measures of success vary between various stakeholders*<br>• Required is a strong focus on quality management to avoid potential conflicts. Quality metrics should be established taking into account each perspective. |
| C7 | *Qualitative success measures*<br>• Hindered application of well-established quality metrics which mostly measure product performance based on quantitative tests. |

**Table 7.** Implications on Project Risk Management

| No | Characteristic |
|---|---|
| C1 | *Problem-solving nature*<br>• Monetary risk impact hard to assess.<br>• Level of risk management and risk awareness should increase towards the end of the project and within the evaluation phases. Risk management planning should ensure that this is implemented in the risk management plan. |
| C2 | *Unknown outcome*<br>• Requires larger amount of time and resources to be allocated for risk management.<br>• Main cause of projects being highly risky which leads to a reluctance in dealing with risks as they are hard to identify and manage. Innovative risk management methods are required that do not focus on direct risk identification, but on assessing the risk of risk-seeking within the project. |
| C4 | *Combination of creative, unpredictable work with regular, thus pre-defined tasks*<br>• Risk attitude should vary depending on task. Required is a flexible risk model that allows for variations in the level of risk response depending on the task type and the required creativity.<br>• Frequency of risk monitoring and controlling cycles not fixed, but should vary depending on task type and project phase. |
| C6 | *The measures of success vary between various stakeholders*<br>• Increase the risk of conflicts between stakeholders concerning the acceptance of results.<br>• Risk cost mostly impossible to determine, however industry partner often require monetary risk assessment.<br>• Perception of risk probability and impact varies between stakeholders, not only based on optimistic and pessimistic attitudes, but based on the expectations and requirements which may diverge to a larger extend that in classical projects. Any risk analysis should also be accompanied by a stakeholder analysis. |

**Project risk management** includes all processes related to the identification and avoidance of project risks. Risk has its origin in uncertainty which is largely present in DSR projects due to the unknown outcome and the highly creative processes. Unknown risk cannot be managed; however increased risk awareness can significantly improve the project success which makes risk management a central task in project management for DSR project.

## 5 Conclusion

Within this paper eight defining characteristics that distinguish DSR projects from more traditional project types have been extracted from literature. With an increase in the amount and size of projects, the field of DSR seeks for dedicated management processes that are capable of addressing the inherent challenges. An analysis of the PMBOK® Guide was performed to evaluate the characteristics. It has shown that the problem-solving nature, the unknown outcome, the novelty of results and the resulting mixed of creative tasks with routine work mostly hinder the direct application of established standards. Using this result, the next step towards a project management method for research projects can be identified which is to develop novel extensions to existing project management approaches predominantly for scope, time, quality and risk management. Also, a deeper understanding of the role of creativity within the project needs to be developed as this needs to be fostered to achieve quality results while at the same time managed to reduce project risks.

## References


1. vom Brocke, J., Buddendick, C.: Reusable Conceptual Models. Requirements Based on the Design Science Research Paradigm. In: International Conference on Design Science Research in Information Systems and Technology (DESRIST), Claremont, CA, USA (2006)
2. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in Information Systems research. MIS Quarterly 28, 75–105 (2004)
3. March, S.T., Smith, G.G.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
4. Erno-Kjolhede, E.: Project management theory and the management of research projects. In: MPP Working Paper (ed.), vol. 3. Copenhagen Business School, Copenhagen (2000)
5. Dvir, D., Shenhar, A., Alkaher, S.: From a single discipline product to a multidisciplinary system: Adapting the right style to the right project. Systems Engineering 6, 123–134 (2003)
6. Shenhar, A.J., Dvir, D.: How projects differ and what to do about it. In: Morris, P.W.G., Pinto, J.K. (eds.) The Wiley guide to project, program and portfolio management, pp. 1265–1286. Wiley and Sons, Hoboken (2004)
7. Bubshait, K.A., Selen, W.J.: Project characteristics that influence the implementation of project management techniques: a survey. Project Management Journal 23, 43–47 (1992)
8. Project Management Institute: A guide to the project management body of knowledge (PMBOK Guide). Project Management Institute, Newton Square (2004)
9. Office of Government Commerce: Managing successful projects with PRINCE2 The Stationery Office, London (2005)
10. Andersen, E.S.: Towards a project management theory for renewal projects. Project Management Journal 37, 15–30 (2006)

11. Pinto, J.K., Cleland, D.I., Slevin, D.P.: The frontiers of project management research Project Management Institute (2002)
12. Clarke, T.E.: Unique features of an R&D work environment and research scientists and engineers. Knowledge, Technology & Policy 15, 58–69 (2002)
13. Brown, C.J.: Can research be project managed? South African Journal of Business Management 30, 72–77 (1999)
14. Payne, J.H., Turner, J.R.: Company-wide project management: The planning and control of programmes of projects of different types. International Journal of Project Management 16, 55–59 (1999)
15. Shenhar, A.J.: Contingent management in temporary, dynamic organizations: The comparative analysis of projects. Journal of High Technology Management Research 12, 239–271 (2001)
16. Besner, C., Hobbs, B.: Project management practice, generic or contextual: A reality check. Project Management Journal 39, 16–33 (2008)
17. vom Brocke, J., Lippe, S.: Towards Management Guidelines for Collaborative Research Projects on Information Systems – Learning from Project Management Contingency Theory. In: VI Conference of the Italian Chapter of AIS (itAIS 2009), Costa Smeralda, Italy (2009)
18. vom Brocke, J., Lippe, S.: Unfolding the nature of public funded research projects in Information Systems. In: ESWIS 2009, Pre-conference workshop of ECIS 2009, Verona, Italy (2009)
19. Kuechler, W.L., Vaishnavi, V.K.: Theory development in design science research: Anatomy of a research project. In: Third International Conference on Design Science Research in Information Systems and Technology, Atlanta, pp. 1–15 (2008)
20. Turner, J.R., Cochrane, J.R.: Goals-and-methods matrix: coping with projects with ill defined goals and /or methods of achieving them. International Journal of Project Management 11, 93–102 (1993)
21. Creswell, J.W.: Research design: qualitative, quantitative and mixed methods approaches. Sage, Thousand Oaks (2009)
22. Wieringa, R.: Design science as nested problem solving. In: International Conference on Design Science Research in Information Systems and Technology (DESRIST) Malvern, PA, USA (2009)
23. Wieringa, R.: Requirements engineering: Frameworks for understanding. Wiley, Chichester (1996)
24. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design research methodology for information systems research. Journal of Management Information Systems 24, 45–77 (2008)
25. Crawford, L., Hobbs, B., Turner, J.R.: Aligning capability with strategy: Categorizing projects to do the right projects and to do them right. Project Management Journal 27, 38–50 (2006)
26. Sauser, B.J., Reilly, R.R., Shenhar, A.J.: Why projects fail? How contingency theory can provide new insights - A comparative analysis of NASA‘s Mars Climate Orbiter loss. International Journal of Project Management (2009) (accepted January 15, 2009)
27. Shenhar, A.J., Dvir, D.: Reinventing project management: the diamond approach to successful growth and innovation. Haward Business, Boston (2007)
28. Crawford, L., Pollak, J.: Hard and soft projects: a framework for analysis. International Journal of Project Management 22, 645–653 (2004)
29. Jordan, G.B., Hage, J., Mote, J., Hepler, B.: Investigating differences among research projects and implications for managers. R&D Management 35, 501–511 (2005)
30. Frank, U.: Towards a pluralistic conception of research methods in Information Systems. ICB Research Report, vol. 7. Duisburg-Essen (2006)
31. Vaishnavi, V.K., Kuechler, W.L.: Design research in Information Systems (2004)

# A Multi-Grounded Design Research Process

Göran Goldkuhl[1,2] and Mikael Lind[3,4]

[1] Linköping University, Dep. of Management and Engineering, Sweden
Goran.Goldkuhl@liu.se
[2] Stockholm University, Dep. of Computer & Systems Sciences, Sweden
[3] Viktoria institute, Sweden
Mikael.Lind@viktoria.se
[4] University of Borås, School of Business & Informatics, Sweden

**Abstract.** There has been a growing interest in the philosophy and constituents of design research by a vast amount of IS-scholars. There are several unresolved concerns and issues in design research (DR). Some examples are the outcomes of design research, the role of theorizing in DR, how to conduct evaluation and validation, and the need for different grounding processes to generate valid knowledge from design research endeavors. This paper describes a multi-grounded approach for design research; consisting of three types of grounding processes (theoretical, empirical and internal grounding). The purpose is to investigate DR-based design knowledge and its roles during design research and design practice. A key feature in this approach is the division between the meta-design (within design research) producing abstract design knowledge and the empirical design practice producing situational knowledge and artefacts. The multi-grounding approach to design research will be illustrated by the support of two design cases.

**Keywords:** Design research, multi-grounded knowledge development.

## 1 Introduction

### 1.1 Background: Important Concerns in Design Research Evolution

There is an increasing interest for design research (DR) within the information systems (IS) community. One might add, a growing interest for the explicit notion of design research (or design science). Actually, there has been a great interest for design oriented research within IS for long time, although it did not had that specific label earlier; cf. e.g. the seminal paper by Nunamaker et al [27] who use the term "systems development research". There have during the IS research history been many research endeavours comprising the design and construction of information systems. Of course, the introduction of concepts like design theory [36] and design science [23] have put a lot of focus and emphasis on design research as an acceptable and viable research approach in IS.

There are several pending issues within DR [7, 19, 35, 37]: What outcomes are there from DR? What is the role of theory and theorizing in DR? What is the relation between design research and design practice? What is included in DR and what is

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 45–60, 2010.

not? What relations are there between DR and other research approaches like e.g. action research? How should evaluation and validation be conducted in DR?

There are different views on the outcome from DR. March & Smith [23] and Hevner et al [18] describe four typical outcomes: constructs, models, methods, and instantiations which are all seen as artefacts. We understand artefacts as "things", i.e. entities that have some separate existence. Hevner et al [18, pp. 82] explicitly exclude "people" and "processes" from artefacts (as outcomes from IS design research). Constructs are defined as "concepts" and "conceptualizations" [23, pp. 256] and "vocabulary and symbols" [18, pp. 77]. When exemplifying constructs, March & Smith [23, pp. 256] mention "relational data model" and "software incremental development". It is obvious from their examples and reasoning that constructs are *abstracted concepts* aimed for theorizing and trans-situational use. "Conceptualizations are extremely important in both natural and design science. They define the terms used when describing and thinking about tasks" [23, pp. 256]. Models are not conceived as abstract entities in the same way as constructs. "Models use constructs to represent a real world situation – the design problem and its solution space…" "Models aid problem and solution understanding and frequently represent the connection between problem and solution components enabling exploration of the effects of design decisions and changes in the real world." [18, pp. 78-79]. Models are thus defined as situational representations. When we move to methods (as a DR outcome), we have jump to a more abstract level again. A method is defined as "a set of steps (an algorithm or guideline) to perform a task" [23, pp. 257]. The authors exemplify with systems development methods, which makes it obvious that this is a general and abstracted artefact and not a situational. An instantiation is a prototype or a specific working system or some kind of tool. This is obviously a situational result. These different results may relate to design process or design product, which is in line with [36].

We find it conceptually problematic that these four different outcomes/artefacts vary between abstracted vs. situational result. March & Smith [23] and Hevner et al [18] are reluctant to bring in theories into design research. They keep them aside in a "natural science" domain and in the supporting "knowledge base". However parts of theories creep into design research outcomes through constructs and methods. These are seen as important elements of design theories and partially of kernel theories by other scholars [15, 35, 36].

What kind of knowledge results emerge through design research and how should they be seen in an abstract vs. situational dichotomy? This is one important research issue that we will address in this paper. There are however some more questions related this emphasised issue. One important issue is how to conceive design practice in relation to design research. In many descriptions, it is obvious that design practice is a part of design research and also considered as the key part. If so, what distinguishes normal design practice from design practice integrated into design research? This question will be necessary for us to discuss later in the paper. This involves also a clarification of relations between the design practice in DR and other activities in DR of more genuinely scientific character. The issue of interaction between design and theorizing has been addressed by several scholars [15, 19, 35, 36]. In the context of our reasoning we need to say something more about this. Design theory seems to play an important role in design research.

One core issue in making a design task a research endeavour is how the design is studied and evaluated. The design research literature consists of several suggestions how this can be done [18, 28]. From these examples a pluralistic view emerges. Evaluation and validation of results can be conducted in many different ways. Once again, the relations to theorizing appear to be essential. Hevner [17] describes the interaction between design research and the existing knowledge base as a "rigor cycle" consisting of "grounding" [17, pp. 88]. We agree that such type of grounding should be performed, but we claim that it is too restricted to limit grounding to the relations with existing knowledge base. There are more grounding activities to be conducted within DR. We will address this issue in more depth in the following. What kinds of different grounding activities should be performed within a DR endeavour?

Some authors describe evaluations to be performed through the conduct of case studies [18, 28]. This is however not a well-addressed issue within DR/IS. It is a bit astonishing actually that the case notion is not treated in-depth in the DR literature. We think that design research is always working with *design-cases*. The design of a specific IT artefact is a design-case and this makes this kind of research usually a one case research. But there may be research inquiries studying and evaluating several situations where a designed artefact is used. So, there is an important research design issue whether there should be a single design-case or a multiple design-case approach?

Broad surveys on the use of designed IT artefacts are usually not considered as design research. This means that design theorizing rather emanates from one or a limited number of design cases. In this respect it resembles grounded theory (GT) oriented research [34]. If so, what are the relations between design theorizing and GT theorizing? What are the resemblances and differences?

In classical GT there is an emphasis on building theory from data. GT comprises theory discovery through induction rather than a theory-testing approach. In a design situation (involving DR), there will usually be an active use of design knowledge (design theories or models) that should inform the design process. Contrary to GT, in design research there is an emphasis on hypothesis-testing [15, 36]. This implies that GT should not be used in a classical way in design research. If GT should be used at all, it needs to be adapted to the design frame.

Grounded theory has been used in change oriented research; cf. e.g. "Grounded Action Research" [6] and "Grounded Action" [31]. These approaches have been developed in combination with action research and not explicitly within a design research frame.

Multi-grounding in design theorizing [12, 21] is one approach with partial inspiration from grounded theory. In traditional grounded theory there is a focus on grounding the theory in data; which should be done by explicitly building the theory on data. Multi-grounding comprises empirical grounding but adds also theoretical and internal grounding. These three grounding processes should be used in a continual and iterative way through the emergence of a design theory. Grounding implies both informed generation and validation through proving.

### 1.2 Purpose and Research Approach

The purpose of this paper is to investigate DR-based design knowledge and its roles during design research and design practice. This implies also that we need to clarify how design practice is related to design research. In our inquiry into design knowledge we will especially look into the notion of multi-grounding as a possible approach to design research. We will discuss different types of design knowledge and how these are related to different grounding processes. We will not so much dig into the notion of design theory since there are several recent contributions here [15, 19, 35]. We find it more urgent to clarify design knowledge in terms of abstracted vs. situational knowledge.

Our research approach for clarifying and developing this multi-grounded design research (MGDR) will in itself be a case of design research. It will be a combination of conceptual and empirically oriented design endeavours. Two empirical cases will be used for illustration and partial validation. MGDR has emerged through a number of DR cases and we delimit our presentation here to two cases. MGDR should be seen as a pragmatic research approach following the spirit of Dewey's [10] pragmatic inquiry. In this pragmatic perspective, the world is always in a state of becoming and knowledge is thus provisional and always evolving and evolvable. Validation and grounding is always partial. Our knowledge evolves over time through continual cycles of generation, usage, evaluation and revision.

In our introduction above we have clarified some knowledge needs in design research that justify the purpose of this paper. In the next section we will elaborate on different types of design knowledge and outcomes from DR as well as the role of design practice within DR. Some aspects of the multi-grounded approach will be presented in section 3. In section 4 we will give an overview of two design cases. The MGDR approach will be articulated in section 5 and the paper is concluded in section 6.

## 2 Design Practices and Design Knowledge

One way to understand design research (as research) is to understand the outcomes from this kind of activity. However, following Hevner et al [18] it remains unclear what really are the scientific results from design research. They describe it in the following way: "The result of design-science research in IS is, by definition, a purposeful IT artifact created to address an important organizational problem." [18, pp. 82]. Later in the paper [18, pp. 87] they describe results as input to scientific knowledge base and this does not add much to the four defined outcomes (as mentioned above). They only add new/improved evaluation methods.

It is interesting to compare their description with a well-established definition of action research (another change oriented research approach). "Action research aims to contribute both to the practical concerns of people in an immediate problematic situation and to the goals of social science by joint collaboration within a mutually acceptable ethical framework" [29]. In this classical definition there is an explicit reference to two goals; practical and scientific goals. These two combined goals have been seen as a key characteristic of action research by several other scholars as well; e.g. [24]. We think that a proper definition of design research should include a reference to two such goals in a similar way. It could be formulated something like this: *Design research aims to contribute 1) through designs as solutions to practical problems and needs of people and 2) also to the knowledge goals of a scientific community.*

In our review of March & Smith [23] and Hevner et al [18] - in section 1 above – we introduced the notions of abstract vs. situational knowledge. We will use this knowledge pair in order to clarify both results and activities of design research. The purpose of abstract knowledge is to create an understanding of phenomena and be a basis for use in different (practical) situations. Abstract knowledge is generalized knowledge where specific situational properties are disregarded (abstracted away from). As said above, a systems development method is a good example of such abstract knowledge. Its aim is to be used in different development instances. Abstract knowledge is type-knowledge and situational knowledge is specific knowledge in instances. There is a knowledge flow between these two knowledge levels. Situational knowledge may be generalized and abstracted and thus become abstract knowledge. On the other side, abstract knowledge may be applied in situational instances and in such cases this might involve situational adaptation and modification of the abstract knowledge.

Design research produces different artefacts. Artefacts can be IT systems (software applications) but also other types of "meta artefacts" as constructs, methods and models [16]. Design research uses and produces design knowledge. It uses abstract design knowledge and it produces abstract as well as situational design knowledge. If it did not produce any abstract knowledge, then it would not be any scientific activity; it would only be plain design. Following this knowledge differentiation we divide design research into two activity layers: 1) design practice that produces situational design knowledge and concrete artefacts and 2) meta-design that produces abstract design knowledge. Meta-design can be seen as 1) a preparatory activity before situational design is started and 2) a continual activity partially integrated with the design practice 3) a concluding theoretical activity summarizing, evaluating and abstracting results directed for target groups outside the studied design and use practices. We chose the concept 'meta-design' although it also involves classical scientific activities like data analysis, evaluation and theorizing. We have depicted these activity layers in figure 1. As all models this is a simplification. It does not describe the important influences from communities/situations to design research/design practices.

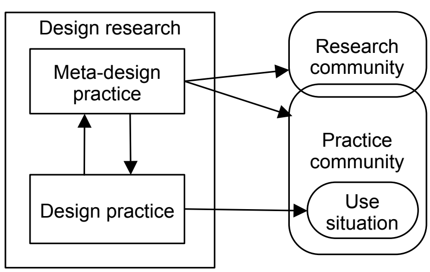



**Fig. 1.** Design research as meta-design and design practice serving communities and situations

In section 1 above, when inspecting the four outcomes described by March & Smith [23] and Hevner et al [18], we made a provisional classification of constructs and methods into abstract design knowledge and models and instantiations as situational knowledge/results. This was done from a close reading of [18, 23]. These types of results can however be expanded using the abstract - situational dichotomy; cf. table 1. Constructs do not only need to be abstract; there might also be situational constructs. These should be concepts that are specific in the design situation. Following Hevner et al [18] models are conceived as situational (cf. analysis and quotes in section 1 above). This is, however, a too restricted view on models. We can use one example to illustrate; the Action Workflow Loop [25]. This is a generic action pattern consisting of four action phases. Based on this generic model (functioning as a template) it is possible, in situational design to create situational models (loop models) consisting of these four action phases. We should therefore distinguish between generic models (as abstract design knowledge) and situational models (as situational design knowledge).

**Table 1.** Different outcomes differentiated into abstract vs. situational

| *Activity type*<br>*Outcome* | **From meta-design:**<br>**Abstract design knowledge** | **From design practice:**<br>**Situational design knowledge and results** |
|---|---|---|
| **Constructs** | Abstract concepts | Situational concepts (may be applied and adapted from abstract concepts) |
| **Models** | Generic models | Situational models |
| **Methods** | Guidelines for design practice | Parts of a situational system or process |
| **Instantiations** | (System abstraction with key properties) | IT systems<br>(prototype or working system) |

Following March & Smith [23] we classify methods as abstract design knowledge. Situational design can of course involve a design of a set of steps to perform a task within a system or related to a system. This means that situational design practice can involve design of situational methods.

Instantiations are not conceived as knowledge in the same way as constructs, models and methods. These three outcomes are inter-subjective knowledge usually expressed in texts or other symbols. We understand instantiation to be the key designed object and as such (in DR/IS) mainly an IT system (a software application). It can be a prototype/part of a system or a working system. IT systems are not knowledge per se, they are expressions and manifestations of knowledge. During the design of such systems, situational design knowledge may be expressed as vehicles for design. We conceive such expressed design knowledge (as descriptions/ specifications of systems) mainly to be in the form of situational models.

This conceptual expansion is summarized in table 1. Since March & Smith [23] and Hevner et al [18] do not use the knowledge pair abstract and situational design knowledge, we do not know whether our conceptual expansion is fully in line with their view on outcomes from design research. What is presented here should however be seen as a refinement of their framework.

This analysis is also partially in line with Sjöström & Ågerfalk [32] who have constructed a design research framework consisting of three polarities: "design v. research, product v. process, and abstract v. concrete". We use the terms abstract vs. situational instead. We do not use the terminology "design v. research" since design practice is an empirical part of research. Our distinction is meta-design vs. design practice. From our analysis it is also possible to clarify the differences between design practice as part of design research and design practice besides design research. To count as part of DR, design practice needs to be performed as an integral constituent of a research endeavour and it should be explicitly instrumental in relation to this scientific purpose of creating abstract and valid knowledge. Normal design practice is performed with no purpose of contributing to research and abstract research-based knowledge. Such normal design practice can of course apply abstract design knowledge (from scientific knowledge base) in design processes, but such knowledge application does not make it research in itself.

It is also important to state what the proper *research outcomes* from design research are. We make a distinction between *research end results* and *intermediary results* (within the research process). All examples of situational design outcomes are seen as intermediary results, also the instantiated IT artefact! These outcomes are, in a research perspective, to be seen as *empirical data*. These data are used for hypothesis testing, evaluation and theorizing into abstract knowledge. Our claim is that abstract design knowledge should be seen as the research end results from design research. A designed IT system is just an intermediary result in the same way as observations and other types of empirical data, as well as triggers for data (as hypotheses, interview questions, observation protocols) are intermediary results in the research process. This means that design practice (within design research) is to be seen as an exploratory empirical part of this research.

An IT system may be a deliverable from a design research endeavour to a user community. In that sense, it can be seen as a kind of end result – as something separate leaving the design research process. It is however not seen as a research knowledge end result, since it is not conceived as knowledge (it is a manifestation of knowledge) and it is a situational result (not an abstracted one). Practice/research communities can of course be interested in IT systems (from design research). They can study them in the same way as they can study other examples/illustrations (as data) from inquiries. Instantiations are exemplars, not abstract knowledge results.

We claim that the purpose of information systems as a science is to create *valid abstract knowledge*. This is mainly done through empirical and conceptual development. Empirical studies of diverse kinds play decisive roles for the creation of a scientific body of knowledge. Exploratory studies involving design practices contribute with powerful empirical data for developing and testing abstract design knowledge.

## 3 The Multi-grounding Perspective

Design theories and other design knowledge need to be justified. Goldkuhl [12] distinguishes between three ways to justify such design knowledge. There are three types of knowledge sources for justification:

- Empirical observations
- Other knowledge of theoretical character
- The design knowledge itself

The justification of design knowledge means to investigate and present warrants for such knowledge. The three types of knowledge sources and warrants give rise to three grounding processes: Empirical grounding, theoretical grounding and internal grounding (figure 2). Empirical grounding comprises grounding through application of design knowledge and observations of its utilisation and effects. The use of external theories can be a grounding of prescriptive design knowledge in explanatory kernel theories [12, 19]. Theoretical grounding includes also grounding in concepts and values. Internal grounding involves control of internal cohesion and consistency. Goldkuhl [12] argues that all three grounding processes should be applied. These three types of knowledge sources are used both for generation and justification of design knowledge. A design theory may be partially derived from and inspired by other theories and through empirical observations.

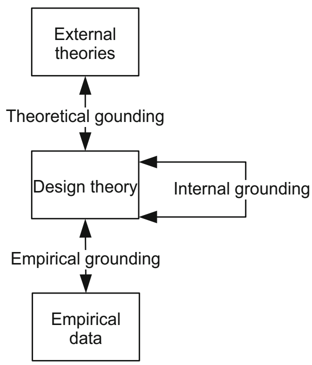



**Fig. 2.** Three types of grounding [12]

The development of design theory/design knowledge will be an emergent process with continually shifting focus between generation and validation and between the three types of knowledge sources (empirical data, other theories, the design theory itself). Examples of multi-grounding processes are given in [1, 9, 12, 21].

# 4 Two Design Research Cases

## 4.1 The Case of e-Me as a Design Research Endeavor

This case of e-Me is driven from a desire to make a contribution the situation of citizens' challenges of coping with a rapidly increasing number of sites and e-Services. Citizens are continually being exposed with numerous web-sites of different offerings and a challenge quest is to remember and cope with a multitude of user IDs, passwords and login procedures. Many citizens have several email accounts. Consequently a lot of time is spent on logging on to different e-mail systems, trying to find passwords and links to various sites. Sometimes e-services are not used, because it's easier to do it the "traditional" way than to figure out the electronic.

In the e-Me Project, a radical approach to this, is to issue the citizen with an electronic assistant, an e-Me, that providers of e-services are required to address with when interacting with the citizen [4]. It might be thought of as turning the internet around. Rather than having citizens find and keep track of sites, the sites will have to come to the citizen and interact with them in the way specified by them. This concept is to see as a counter reaction to the massive onslaught of sites, home pages and email spam. There is nothing inherent in the internet technologies that makes the internet necessarily organization centric. The e-Me concept clearly illustrates how the same technologies can be used to increase the initiative of the citizens.

This design research endeavour suited perfectly to both utilize the development and exploration of the role of an e-Me for coping with these problems (situational knowledge) and to develop valid abstract knowledge. The development of the abstract knowledge was to be seen as a contribution to an ongoing dialogue of how to utilize co-design in multi-stakeholder settings. The notion of co-design is inspired by Churchmann [8] and has then been brought into the IS-community by e.g. Forsgren [11] by the development of a first co-design framework. This framework is a multi-stakeholder model in which all stakeholders concerns, related to a certain co-design situation, are taken into consideration by either inviting, or considering perspectives of, diverse stakeholders.

In the project, a choice was made to use students as one category of citizens. Early in the project a design vision was formulated that expressed that "*the students should not need to go to the information - the information rather comes to the students based on the active profile set by the student.*" [20]. This design research had thus two missions:

- To contribute to the development of co-design as philosophy and method for stakeholder-driven business development (desired abstracted design knowledge)
- To contribute to the understanding of the next generation of internet-based services through the exploration of the role of an e-Me in a practical context (situational design knowledge)

The project (2005–2007) was organized in three phases; Concept development, prototype realization, and proof-of-concept. Each of the phases involved students as co-designers and resulted in both situational design knowledge as well as abstract design knowledge.

In the concept development a number of co-design workshops were conducted as ideal-oriented design studies engaging numerous (future) users and service providers resulting in eight different scenarios documented with text and cartoons as two types of models [3]. The situational design knowledge was a validated description of an ideal future with e-Me as an integrated part of the life situation. In the development of abstract design knowledge, founded in (soft) systems theory, experiences from conducting these co-design workshops were used as a basis resulting in co-design as an approach to border-crossing, network innovation [5].

In the prototype realization the results from the concept development was used as an input. In this phase a number of micro scenarios were derived from the co-design scenarios described. These micro scenarios (models) were used for the system developers to design and build a pilot version of the e-Me – an instantiation of e-Me in software (IT-artefact) as situational design result. A small group of users were involved in test and evaluation during this phase. This design practice resulted in abstracted design knowledge related to approaches to converting scenarios to software requirements, close user interaction integrated in the development process, and the role of systems development in a design research setting.

In the phase of proof-of-concept the first prototype of e-Me was deployed for a group of 120 people who became a part of the e-Me project group and co-designers. The students co-designed e-Me by trying out the prototype – both in order to identify shortcomings in the application and identify new situations, both within and beyond the school setting, when an e-Me would be of assistance. In design practice terms the e-Me became the IT-artefact that gave rise to use effects. These use effects also resolved in an increased understanding of the students' desires of IT-services related to their life situation. Founded in approaches to participatory design [26] as well as language/action approaches to communication modeling [25] an approach for integrated development, use, and learning in a co-design setting [22] as abstract design knowledge was put forward.

To summarize the essential characteristics from the e-Me project as a design research endeavour, this project has both resulted in an overall goal of designing and evaluating an e-Me artefact (on the situational level) and a multi-grounded emergent (co-)design theory (on the meta-design level).

### 4.2 The Case of BITA as a Design Research Endeavor

BITA is an e-government project [13, 33]. The area for development is personal assistance for disabled persons. In Sweden there are two legal acts (and several other statutes) governing this kind of public support. These acts regulate who can apply for personal assistance, how to apply, how to organise and deliver these services and the financing of these services. This egov project concerning allowances for personal assistance was started due to the very cumbersome administration and needs for better quality in time and cost accounting. The two legal acts have given rise to fairly complicated work processes and interaction patterns between different stakeholders. In the project group there were representatives both from several municipalities (responsible for the delivery of the personal assistance services) and from the Social Insurance Agency (responsible for authority decisions and financing). Two researchers participated actively in the project; making it an action research and design research endeavour.

The project started with a workpractice diagnosis and process analysis. New inter-organisational processes were designed and proposed. These proposals included also new IT solutions for time and cost accounting. In order to avoid much of the cumbersome paper work, the project group suggested that the signatures on invoices from municipalities to the Social Insurance Agency should be replaced by an IT system built on the idea of "social transparency" [33]. The system should comprise time information (concerning personal assistance services) of diverse types; scheduled time, reported time (through mobile media), determined time (by the managers) together with commentaries from the concerned stakeholders (the clients, their fiduciaries, the personal assistants, the managers). This was not just a loose design idea. It was operationalised and implemented first in prototypes and later in the system to run.

This made it possible for different stakeholders to judge and evaluate the idea and thereby for the researchers/designers to continually revise and refine the design.

This principle for social transparency was derived from a design theory that was actively used during the IS design: the IS actability theory [2, 14]. In this design theory there is one design criterion, "actor visibility", that was a key inspiration for the emergence of the design ideal of social transparency. This emergent concept has later been analysed and abstracted and fed back to the actability theory [33]. This means both a kind of further empirical grounding as well as a theoretical refinement. The emergence of the social transparency idea did not only come from theory. To a large degree, this was an innovative response to the experienced practical problems and the high ambitions to reduce costs and lead time for administrative work.

However, this idea of a new IT system to replace signatures from clients was not accepted by the Social Insurance Agency with reference to authority statutes. A conflict emerged between this agency and the project group. There were different legal interpretations made by different stakeholders. This led to an in-depth analysis of different legal pre-conditions for this egov initiative. A key notion emerged during this analysis: value balancing [13]. There may be different legal constraints in an egov development project. The way forward is not only to study regulations that may comprise rules which could hinder the proposed development. It is necessary to study regulations of different kinds (domain-specific regulations, general administrative regulations, egov policies) and on different levels (laws, ministry regulations, authority regulations, policy documents). The value balancing should identify values behind the regulations and give priorities to basic values [13]. Due to these conflicting legal interpretations in this egov initiative, the value balancing emerged as an important activity which later also was theorised [13]. This can be seen as an embryonic design theory for handling legal barriers in egov development. The idea of value balancing was later theoretically grounded in institutional theory [30].

## 5 Multi-Grounded Design Research

Based on the conceptual analyses (in sections 1-3 above) and the two empirical design cases (in section 4) we present a model for multi-grounded design research (MGDR).

Design research consists of an empirical part (a design practice) and a theoretical part (meta-design). There is a continual interaction between design practice and meta-design (figure 1). The two parts exchange knowledge. Meta-design produces abstract design knowledge and the design practice produces situational design knowledge and other situational results (instantiations/IT systems). These knowledge types and results are exchanged between the two parts of design research. The knowledge exchanges are also parts in grounding processes. Situational design knowledge is used for empirical grounding of abstract design knowledge and abstract design knowledge is used for theoretical grounding of situational results. This will be further explicated below.

Goldkuhl [12] presented a model for multi-grounding of design theories; cf. section 3 above. In this paper we mainly use the notion of abstract design knowledge instead of design theory. We conceive abstract design knowledge to be a broader notion, encompassing design theory and possibly also other knowledge types than those within a design theory. There are still controversies concerning what to include in and how to divide a design theory [15, 35, 36]. We would like to add that we are not in opposition against the notion of design theory. We do find it fruitful, but there would be another paper to resolve issues concerning constituents of design theories.

Instead of talking about design theory and its multi-grounding through theoretical, empirical and internal grounding (as in [12]), we would instead talk about multi-grounding of abstract design knowledge. We also expand the grounding reasoning to situational design knowledge and results. This is depicted in figure 3.

We distinguish between the theoretical level (of meta-design) and the practical-empirical level of situational design knowledge and design practice. Following Goldkuhl [12], abstract design knowledge should be grounded 1) empirically (in empirical knowledge), 2) theoretically in other theories (and similar abstract sources) and 3) internally making the abstract design knowledge a coherent whole. In abstract design knowledge we include constructs, methods and generic models (see section 2 above). We also include values [12] and (design) principles, which can be seen as abstractions of methods. A design theory may encompass all these parts (and possibly more) and therefore we add theory to the constituents of abstract design knowledge to have it complete (figure 3).

The situational level is centred around situational design knowledge. Design actions produce situational design knowledge. Such knowledge is expressed mainly in situational models (model-artefacts) and the produced IT artefacts (instantiations). The situational design knowledge should be theoretically grounded in the abstract design knowledge. This means that the abstract knowledge (e.g. constructs, values, methods, generic models) informs the design process. It is also so that the situational results are explicitly checked against the abstract knowledge to investigate compliance. It should however not be that theoretical compliance and grounding should a priori be given priority over proposed designs if these designs are found practical. Theories are (in the pragmatic view) always provisional and emergent. New knowledge can be developed through practical design. The produced situational design knowledge should be informed and governed by practical knowledge as for example problems, goals and needs. The proposed design should be a conscious and reflective response to these practical needs: a practical grounding of purpose, relevance and compliance. The proposed designs (models as well as instantiations) should also be evaluated against anticipated and observed use effects: a consequential grounding. These knowledge types (background and projection knowledge) constitute sources for a practical-empirical grounding of situational design knowledge/results.

To this we can add internal grounding. Systems are built from models. There are often several different types of models produced during an IS development and there are well-known model goals as traceability and cohesion. Models and systems should be a coherent whole without too much contradiction. Situational models should be checked that they are sufficiently congruent.

Abstract and situational design knowledge will continually emerge during the design research process. They will inform and influence each other in a dialectic dance. As can be seen from figure 3, there will be a mutual grounding between these kinds of knowledge.

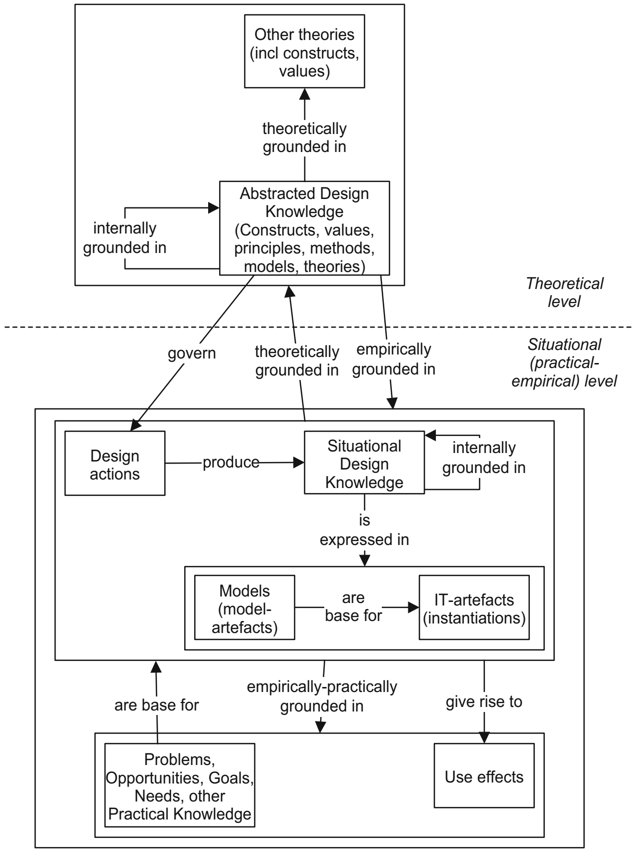



**Fig. 3.** Multi-grounding of design knowledge

Design practice (in DR) should be theory-informed. This does not necessarily mean that a design theory is first chosen and then used as a test basis in the design practice. The design research can be governed by an interest in some design challenges and available opportunities for practical design work. Such practical design concerns can govern the search for possible abstract design knowledge (can be generic models, prescriptive design theories or methods or even relevant explanatory kernel theories) to be used as a basis for practical design. If no proper theoretical basis is found in the search process, a development of some piece of abstract design knowledge can emerge as a direct response to the practical design needs. This abstract design knowledge can then continually be modified and checked against the emergent situational designs and their use effects.

This means also that research questions and design hypotheses need not be explicitly given at the start of the DR endeavour. They can continually emerge during the DR process when pre-knowledge meets practical design challenges. Some ideas will usually exist in the beginning, but they can be changed during the process if other and more relevant research and design issues are discovered.

## 6 Conclusions

In the IS-field there is and has been an intense debate related to the application of, and the role of, design research. In this paper some conceptual flaws related to knowledge creation in design research have been acknowledged. Special concern has been the different types of knowledge that could arrive from design research and its implications for knowledge generation and grounding processes.

We have made a distinction between two inter-related practices in design research; the theorising meta-design practice vs. the situational design practice. In design research these two practices produce knowledge and artefacts on two levels; abstracted vs. situational knowledge. In this paper we have distinguished between four types of outcomes (artefacts) which have different meanings related to the two inter-related practices. These four types of outcomes (following [18]), are constructs, models, methods, and instantiations. This means that we acknowledge IT artefacts vs. other types of artefacts as results of design research endeavours.

In knowledge development validity claims for produced knowledge should be raised. To some extent, processes of theoretical grounding by explicating relations to the knowledge base has been advocated by design researchers, but this is too limited. Three types of knowledge sources for justification are therefore acknowledged with appurtenant grounding processes in this paper; empirical grounding, internal grounding, and theoretical grounding. Based on these different knowledge sources, and the distinction between the meta-design practice and the design practice a framework for multi-grounding of design knowledge has been formulated.

Of special concern in design research is the emergence of knowledge through the interplay between the two main activities build and evaluate. However, the multi-grounded framework rests upon this, but emphasises also that the design researcher need to be flexible in which way and order abstract design knowledge influence situational design knowledge and vice versa. The implications of this are also that research questions and design hypothesis do not need to be stated at the start of the DR endeavour. They might emerge throughout the DR-process. The same goes for the identification of scientific dialogues in which evolving abstract knowledge become an important contribution to.

In clarifying the two layers of design knowledge we have also touched upon the relation to action research. Both design research cases in this paper included also action research. Future research will, based on this design knowledge dichotomy, try to develop a framework integrating design research and action research.

## References

1. Ågerfalk, P.: Grounding through Operationalization: Constructing Tangible Theory in IS Research. In: Proc of the 12th European Conference on Information Systems (ECIS 2004), Turku (2004)
2. Ågerfalk, P.J.: Investigating Actability Dimensions: A Language/Action Perspective on Criteria for Information Systems Evaluation. Interacting with Computers 16(5), 957–988 (2004)

3. Albinsson, L., Forsgren, O., Lind, M.: e-Me Stories & Scenarios - The Ideal Electronic Galaxy of the Student, University College of Borås, Sweden (2006a)
4. Albinsson, L., Lind, M., Forsgren, O., Ozan, H.: Turning the internet around - e-Me: The Students ideal e-Service. In: Cunningham, P., Cunningham, M. (eds.) Exploiting the Knowledge Economy: Issues, Applications, Case Studies. IOS Press, Amsterdam (2006b)
5. Albinsson, L., Lind, M., Forsgren, O.: Co-Design: An approach to border crossing, Network Innovation. In: Cunningham, P., Cunningham, M. (eds.) Expanding the Knowledge Economy: Issues, Applications, Case Studies. IOS Press, Amsterdam (2007)
6. Baskerville, R., Pries-Heje, J.: Grounded action research: a method for understanding IT in practice. Accting. Mgmt. & Info. Tech. 9, 1–23 (1999)
7. Baskerville, R.: What design science is not. European Journal of Information Systems 17, 441–443 (2008)
8. Churchman, C.W.: The design of inquiring systems. Basic Books, New York (1971)
9. Cronholm, S.: Illustrating Multi-Grounded Theory. In: 3rd European Conference on Research Methodology for Business and Management Studies, Reading (2004)
10. Dewey, J.: Logic, the Theory of Inquiry. H. Holt and Company, New York (1938)
11. Forsgren, O.: Samskapande Datortillämpningar: En systemteoretisk ansats för lösning av vissa förändringsproblem vid administrativ datoranvändning (Doctoral thesis), Umeå universitet (1988)
12. Goldkuhl, G.: Design Theories in Information Systems – A Need for Multi-Grounding. Journal of Information Technology and Application (JITTA) 6(2), 59–72 (2004)
13. Goldkuhl, G.: Innovation in a regulated environment? - Legal barriers for e-government development. International Journal of Public Information Systems 2009(2), 77–95 (2009)
14. Goldkuhl, G.: Pragmatic Qualities of Information Systems – Actability Criteria for Design and Evaluation. Invited paper to the 11th International Conference on Informatics and Semiotics in Organisations (ICISO), Beijing (2009)
15. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems (JAIS) 8(5), Article 2, 312–335 (2007)
16. Iivari, J.: A Paradigmatic Analysis of Information Systems As a Design Science. Scandinavian Journal of Information Systems 19(2), 39–64 (2007)
17. Hevner, A.R.: A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems 19(2), 87–92 (2007)
18. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MISQ 28, 75–106
19. 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)
20. Lind, M., Berglund, S.: e-Me – the students best friend. In: The 14th European University Information Systems Organisation (EUNIS) congress, EUNIS 2008 VISION IT - Visions for IT in Higher Education, University of Aarhus Universitet, Denmark (2008)
21. Lind, M., Goldkuhl, G.: How to develop a Multi-Grounded Theory: The evolution of a business process theory. Australasian Journal of Information Systems (AJIS) 13(2), 68–85 (2006)
22. Lind, M., Rittgen, P.: Challenges of Co-Design: The Case of e-Me. In: Handbook of Research on Socio-Technical Design. Idea Group, Hershey (2009)
23. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
24. McKay, J., Marshall, P.: The dual imperatives of action research. Information Technology & People 14(1) (2001)

25. Medina-Mora, R., Winograd, T., Flores, R., Flores, F.: The Action Workflow Approach to Workflow Management Technology. In: Turner, J., Kraut, R. (eds.) Proceedings of the Conference on Computer-Supported Cooperative Work, CSCW 1992. ACM Press, New York (1992)
26. Mumford, E.: Designing human systems, the ETHICS approach. Manchester Business School, Manchester, UK (1983)
27. Nunamaker, J.F., Minder, C., Purdin, T.D.M.: Systems Development in Information Systems Research. Journal of Management Information Systems I Winter 1990-91 7(3), 89–106 (1991)
28. Pries-Heje, J., Baskerville, R., Venable, J.: Strategies for design science research evaluation. In: Proceedings of the 16th European Conference on Information Systems, Galway, Ireland, pp. 255–266 (2008)
29. Rapoport, R.: Three dilemmas of action research. Human Relations 23, 499–513 (1970)
30. Scott, W.R.: Institutions and organizations. Sage, Thousand Oaks (1995)
31. Simmons, O.E., Gregory, T.A.: Grounded Action: Achieving Optimal and Sustainable Change. Forum Qualitative Social Research (FQS) 4(3) (2003)
32. Sjöström, P.J., Ågerfalk, J.: An Analytic Framework for Design-oriented Research Concepts. In: Proceedings of the Fifteenth Americas Conference on Information Systems, San Francisco, California, August 6-9 (2009)
33. Sjöström, J., Goldkuhl, G.: Socio-instrumental pragmatism in action. In: Whitworth, B., De Moor, A. (eds.) Handbook of Research on Socio-Technical Design and Social Networking Systems. IGI, Hershey (2009)
34. Strauss, A., Corbin, J.: Basics of qualitative research. Techniques and procedures for developing Grounded Theory, 2nd edn. Sage, Newbury Park (1998)
35. Venable, J.: The Role of Theory and Theorising in Design Science Research. In: First International Conference on Design Science Research in Information Systems and Technology, Claremont, California, pp. 1–18 (2006)
36. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3(1) (1992)
37. 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 (JITTA) 6(2), 43–58 (2004)

# Relevance and Problem Choice in Design Science

Roel Wieringa

University of Twente, Department of Computer Science, Information Systems Group
P.O. Box 217, 7500 AE Enschede, The Netherlands
r.j.wieringa@ewi.utwente.nl

**Abstract.** The supposed opposition of rigor versus relevance is based on the mistaken idea that rigor consists of linear technology transfer combined with positivistic science, and ignores the context-dependence of relevance as well as the incorporation of conditions of practice necessary for applicability of knowledge. Historical insights from the history of science and technology show that technology is not transferred linearly from research to practice, and that technical science has more in common with social science than a superficial comparison would reveal. In both fields, (1) practical problems are often solved *without* input from research, and (2) researchers often investigate *past* innovations rather than prepare future ones. And in both fields, (3) relevance is context-dependent, because it depends on changeable goals of stakeholders. Applicability is a more important requirement than relevance to a goal, where applicability is the match between theory and the condition of practice of a concrete case.

This paper summarizes insights from the history of science and technology to substantiate these points and provides an extended framework for design science to incorporate these insights. Since relevance depends on problem choice, the paper also summarizes what is known about classes of relevant practical problems and research questions in technical design science and discusses the relevance of this for IS design science. We finally discuss implications for research methods, research strategy, and knowledge transfer in IS design science.

## 1 Introduction

In 1983, Schön posed the dilemma of rigor versus relevance as one between technical rationality, where problem-solvers select from alternative solutions the one that optimally contributes to an agreed-upon end, and real-world action, where practitioners use their experience and intuition to muddle through unique, uncertain and unstable situations [1, pages 39–43]. According to Schön, the rational problem-solving view of technical rationality assumes a positivist philosophy, in which basic scientific results are applied to practical problems. This may be appropriate in the technical sciences, he said, but it is not applicable to complex social situations. Rather than emulate technical sciences in an attempt to be rigorous, let's emulate practitioners and be relevant, Schön said. Hence the dilemma of rigor versus relevance.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 61–76, 2010.

Since the 1990s, Schön's dilemma has been subject of discussion in information systems (IS). Various ways out of it have been proposed. Benbasat & Zmud [2] proposed to increase relevance by selecting interesting problems, accumulating context-rich knowledge about those problems and transferring the results to practitioners, all the while preserving the rigor of basic research methods. Davenport & Markus [3] propose a more applied research approach, emulating consultants to select relevant problems, borrowing research methods from evaluation and policy research to investigate them, and transferring the results to students who will later enter practice. Design scientists propose not just developing new knowledge but also new artifacts that solve practical problems [4,5,6,7]. This is an approach common in industrial research, namely developing an artifact to solve a problem, and then investigating the problem-solving properties of the artifact.

These three approaches assume that relevant artifacts and knowledge are developed in basic, applied or design research, respectively, and then transferred to practice. Historical research in science-technology interaction reveals a considerably more complex picture and there is no reason why this complexity should be absent from the interaction between IS research and practice. An analysis of science-technology interaction may reveal relevant implications for IS design science, that may increase our options for problem selection and research strategy.

A brief review of insights from the history of science-technology interactions reveals that in addition to the interaction between scientific research and artifact development typical of industrial research, technology often develops without input from science, and technical research often progresses by curiosity-driven research that solves no pressing practical problem (section 2). This motivates an extension of the framework for mutual nesting of practical problem solving and scientific research proposed earlier [8], which itself refines the framework of Hevner et al. [7]. The extension consists of adding a flow of goals and budgets from the economy to design science, and the production of practical knowledge by practical problem solving (section 3). Relevance of artifacts (the outcome of practical problem solving) and theories (the outcome of scientific research) is context dependent, for it depends on goals from the economy. Applicability of theory or artifacts, by contrast, depends on the incorporation of conditions of practice in the theory or in artifact behavior.

All three approaches listed above agree that relevance is determined, among others, by problem choice. Solutions to irrelevant problems will not be used and problem choice is therefore an important art for IS researchers [9]. Section 4 summarizes kinds of practical problems typically solved in design science, and gives examples from technical as well as IS design science. These problems have shown to be relevant for the economy and therefore there was budget available to solve them. The section also lists typical scientific research questions, with examples from technical and IS design science. This shows that incorporation of conditions of practice is an important prerequisite for applicability of design theories. In section 5 we discuss the implications for research methods, research strategy and transfer of results, and we return to the question to which extent we have now dealt with Schön's dilemma, and what part of it remains untouched by our analysis.

## 2 Science-Technology Interactions in History

The dominant view of the relation between science and technology is that it is a linear progression from basic research followed by applied research and development, ending with production and diffusion [10,11]. This is the linear model assumed by Schön to exist in the technical sciences. There are some spectacular examples of this in the history of science and technology. Shortly after Benjamin Franklin discovered that lightning is electricity, the first lightning rods appeared [12, page 154]; the theory of ultrasound developed in the 1870s was used in the 20th century in the development of sonar technology and medical ultrasound technology [13]; and basic research into polymer in the 1930s by Carothers at DuPont led to the invention of nylon, one of the biggest money-makers of the company [14].

However, extensive historical research has shown that these are exceptions and that it is hard to impossible to discover a linear handover of knowledge from basic science to technology [11,13,15,16,17]. The short summary is this: New technology does not spring from science but from the improvement of existing technology; and this improvement is motivated by the desire to meet perceived stakeholder needs.

But if technology is sometimes, but not always, applied science, then what other kinds of relationships are there? Following Gardner [18], we can distinguish cases where (1) there is no relationship from cases where (2) science follows technology, (3) technology follows science, and (4) science and technology develop in interaction. The historically earliest cases are those in which there is no relationship. Science as we know it did not exist in most of human history, and when it finally did it often played no role in the development of much of the technology currently still on the market. A few examples suffice to make the point: In the invention of windmills, the stirrup, barbed wire, the zipper, the revolving door and many other kinds of artifact, science played no role.

The second class of examples is where science follows technology. There are two subcases. First, technology may be transferred to science in the form of instruments . Early examples are such as the telescope, thermometer, barometer and air pump in the 17th century [19,20,21]. The second subcase is that science may study existing technology to discover how it works. A famous example is the investigation of steam machines by Sadi Carnot in the early 19th century to discover why they actually worked, leading to the new science of thermodynamics [22,23]. This is a very common way of working. Galileo studied how machines like levers and pulleys used by ship builders actually worked, starting the new science of machines and of strength of materials [24, pages 36–46]. And over hundred years after lead-acid batteries were introduced, researchers still try to understand how they work [18, page 15]. More examples are given by McKelvey [15] and Gardner [18].

In the third kind of case, technology follows science. There are two subcases here too. In one subcase there is linear progression from science to technology, as in the examples given at the start of this section. The agreement among historians of technology is now that these are spectacular exceptions. The second subcase is more common, in which technologists encounter some problem for the solution of which they turn to already published scientific results [25]. These scientific results have been developed earlier, not knowing for which technical problems they could be useful.

The fourth and last case is where science and technology develop in mutual interaction. This is typical of 20th century industrial research [14,26,27]. It is also the mode of working proposed in design science. I now give a framework that can accommodate all kinds of interactions reviewed here.

## 3 An Extended Framework for Design Science

### 3.1 Mutual Nesting of Practical Problem Solving and Research

In our design science framework we distinguish two kinds of problem solving activity, solving practical problems and answering research questions [28]. A *practical problem* is a problem to improve the world with respect to some stakeholder goals. To solve it some artifact, such as a software system, technique, method, process, treatment, etc. is needed. A *research question* is a knowledge question to be answered by scientific research. To answer it, some validated proposition about the world is needed. This distinction leads to a refinement of the design science framework of Hevner et al. [7] shown in figure 1. In this framework, practical problem solving delivers artifacts with the aim of solving practical problems in an organizational environment, and design science research investigates properties of these artifacts.

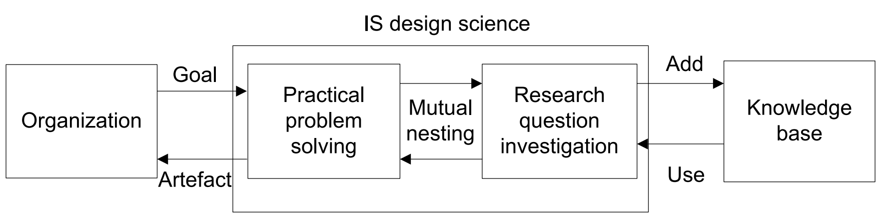



**Fig. 1.** Refinement of the framework of Hevner et al. [7], adopted from Wieringa [28]

### 3.2 Extended Framework of Interactions

To accommodate the historical insights from the previous section, we need to further elaborate this framework by indicating that research question investigation serves goals too, and that as any other activity it needs a budget. This gives us the elaborated framework of figure 2.

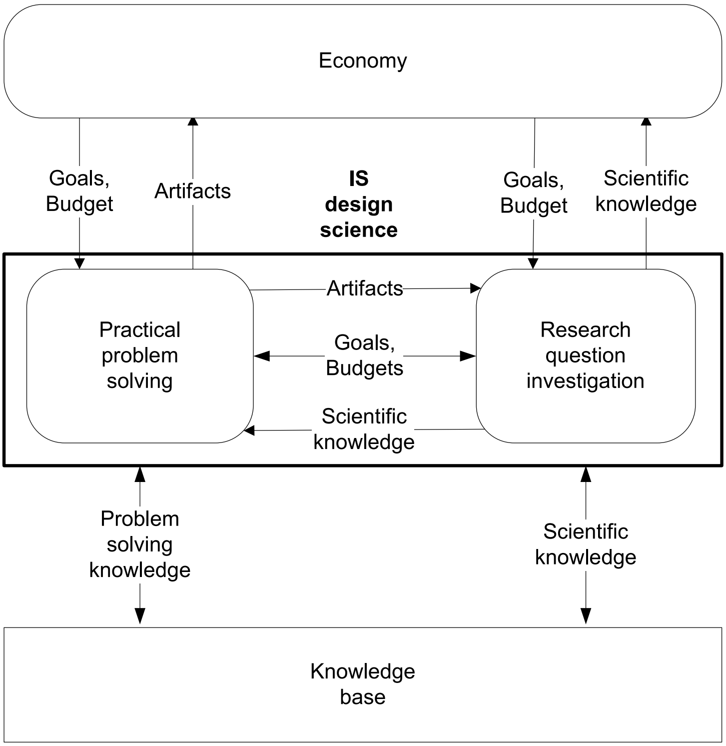



**Fig. 2.** Further elaboration of the framework for design science. Nodes represent activities or results of those activities (rounded corners) or enduring results of these activities (sharp corners), arrows represent flows of information, money or artifacts. Flow of control is not represented.

The nodes with rounded corners represent activities, not people or organizations. One person or organization can perform activities in any of these boxes at the same time. The sharp-cornered rectangle represents an enduring result of these activities, namely knowledge. The arrows represents flows of money, information or artifacts without indicating who triggers a flow: the sender or receiver. The arrows from the knowledge base box terminate on the design science box, meaning that these interfaces are accessible from both kinds of design science activities, practical problem solving and scientific research.

The environment of design science has been framed as the economy, which here is intended in a broad sense as the allocation of finite resources to goals, where not every goal can be allocated all the resources needed. In other words, in this paper we will view the economy as consisting of activities to achieve goals with a finite budget for means. My thesis is that these goals are the sources of relevance for design science.

I now discuss the interfaces of research question investigation and practical problem solving. Practical problem solving receives a budget to produce artifacts that are intended to achieve goals. This may add problem-solving knowledge to the knowledge base, an interface not shown earlier. Problem solving knowledge is what Vincenti [29, pages 217-222] calls practical considerations and design instrumentalities. *Practical considerations* consist of accumulated experience laid down in design procedures, rules of thumb, generalizations from observation not explained by scientific theory, etc. *Design instrumentalities* are how-to-do knowledge such as knowledge of procedures, ways of thinking, and judgmental skills that may partly be tacit.

Practical problem solving may also provide its artifacts to scientific research, for example as instrument to do research or as object of research itself. If transferred as instrument to do research, then apparently it received goals and budget from this research activity; if transferred as object to investigate, then it may transfer some of its goals and budget to the research activity and in return for that it receives scientific knowledge. Practical problem solving may also draw in the store of scientific knowledge published earlier, as indicated by the arrow from the knowledge base to the design science box. However, it cannot *add* scientific knowledge directly; this would involve a scientific research activity triggered by practical problem solving.

Research, like practical problem solving, receives goals and budget from the economy, either directly or indirectly from a practical problem solving activity. It returns the favor by producing knowledge, which is added to the knowledge based through scientific publication channels and communicated to the economy through the professional and popular press and, in the case of universities, in the minds of undergraduate and graduates who enter the economy. These channels should be used by IS design researchers too: The professional and popular press to reach managers and other practitioners (as urged by Benbasat & Zmud [2]) and students for a long-term upgrade of the workforce of practitioners (as urged by Davenport & Markus [3]).

The triangle relating the economy, practical problem solving and research was first proposed by Aitken [30], who used it to analyze the development of radio from laboratory instrumentation. It has independently been used by Lyytinen and King [31] to indicate that technology (corresponding to practical problem solving in figure 2) creates the economic surplus to make the budget available to investigate artifacts scientifically. The addition of this paper is to integrate the frameworks and elaborate the interfaces in the light of historical evidence.

## 4 Problem Selection in Design Science

Several historians of technology have inventoried classes of problems occurring many times in the development and investigation of artifacts [32,33,34,29]. In the next two sections I summarize this and discuss the relevance of these problem classes for IS design science. The practical problem-solving goals and design research questions listed below further refine the Practical problem solving and Research question investigation boxes in figure 1.

### 4.1 Sources of Relevance in Practical Problem Solving

Practical problems are characterized by improvement goals, but there are many special cases, as illustrated in figure 3.

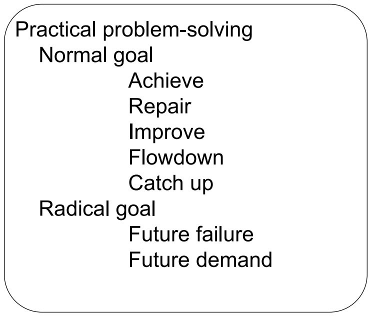



**Fig. 3.** Practical goals identified from the literature

- *Achieving some economic goal.* This is the normal model of practical problem solving, where the goal is set by some private (business or non-profit) or government stakeholder. Relevance is determined by the value of the economic goal.
- *Repairing failures.* When an artifact fails, a practical problem solver such as a technologists will try to diagnose the failure and repair it. Examples from IS are the attempt to configure an ERP system such that it stops failing to achieve the goal of cost reduction, or to improve effort estimation techniques so that effort estimation stops delivering underestimations.
- *Improving performance.* Even if an artifact achieves its goals satisfactorily, technologists and other practical problem solvers will aim to improve its performance. This will be an activity without economic budget if the problem-solver's honor is the only goal to be served, but if some economic stakeholder's goal is served by it too, budget may be available. For example, functionality and performance of a collaborative software tool may be improved by observing the behavior of its users and exploiting the possibilities of new technology; an implementation effort estimation technique may be improved by collecting data to fine tune the technique.
- *Flowing down system goals.* In the development of complex systems such as aircraft, overall system goals imply goals for subsystems such as the propulsion system or landing gear [32,29]. Deriving subsystem goals from the goals of the overall system is called flow-down in systems engineering. For example, implementation of an e-commerce sales channel may involve subsystems for order tracking, payment and security, and overall system goals will imply goals for those subsystems.
- *Catching up with large systems improvement.* Hughes [35] introduced the concept of large technological system as a system of diverse artifacts not

centrally managed but with a common goal, such as the system of private transport by car, which consists of car manufacturers, car financing, insurance, roads, petrol supply and legislation that jointly make it possible for individuals to drive cars. He also introduced the concept of *reverse salient* in large technological system as a part that holds back advancement of the system as a whole [33]. For example, a reverse salient for car transport by electrical cars is the scarcity of battery reloading stations. An example in IS may occur in value chain automation, where for example a shared goal of an extended enterprise may be thwarted by lack of an adequate information risk assessment techniques for extended enterprises.

– *Circumventing predicted performance limits.* In an analysis of jet engine development around the 1930s Constant [32] observed that a few visionary engineers predicted that contemporary aircraft propulsion technology would fail at higher speeds and altitudes, and also predicted that future economic goals would nevertheless require those speeds and altitudes. Constant called this kind of problem a *presumptive anomaly* but here I call it a predicted performance limit. This is interesting, for in contrast to all previous cases there is no experienced problem. Examples from information technology are the development of new computing paradigms (e.g. quantum computing) or storage technology to meet future performance limits. Examples from IS are harder to give, probably because in the technical cases just mentioned, future performance limits can be predicted with certainty from the laws of physics. As soon as a system contains social components, i.e. people, predictions would include a prediction of human performance and this is notoriously hard and controversial: As Popper famously made clear, the course of human history can perhaps be explained but not be predicted. The current goal of meeting the needs of the future enterprise by introducing service orientation and cloud computing depends on a prediction of the performance of future enterprises that is debatable: The future may develop this way only if people decide to follow this vision, and many unforeseen factors may then intervene for this scenario to fail.
– *Meeting predicted demand.* This differs from the circumvention of predicted performance limits in that in this case a practical problem solver predicts that there will be a demand for an artifact the he or she will develop. The artifact may generate new goals in the economy but it does not solve future performance limits of current artifacts because there is no current artifact. Predicting demands in this way is an entrepreneurial competence. Examples are the introduction of laptops and of mobile phone technology and of ambient technology on the market.

This list is not claimed to be exhaustive but we can nevertheless draw some interesting lessons from it. Circumventing predicted performance limits and meeting predicted demand are *radical goals* [32], and may lead to radical innovations. Typically, these involve entrepreneurs with a vision and the stamina to achieve it. Famous early examples are Edison and Marconi, and famous recent examples are Steve Jobs and Bill Gates. Radical problem solving is high-risk, high-reward.

A more normal problem solving goal is to achieve some economic goal with an incremental improvement of current technology, or to repair failures, to improve performance, to flow down system goals, or to catch up with large systems improvement. Any of these efforts may lead to radically new technology, but normally they lead to incremental improvements of existing technology.

Interesting in all cases is that goals come from outside practical problem solving. The practical problem solver must understand and deal with goals of other stakeholders, who are interested in a solution artifact only in as far it helps them achieve their goals. This points at the need, also in normal problem solving, to manage the relationships between stakeholders and practical problem solvers in figure 2, a role called *engineering manager* by Wise [17, page 245]. These must be people who understand market needs and can translate these in solution artifacts.

These observations puts Gray's call for IS researchers to lead the market with new technology into perspective [36, pages 337–338]. In our framework, managing the relations at the three activity interfaces are different activities and historical evidence indicates that they require different competencies. At the very least it would be far fetched to require of every IS researcher to produce radical innovations; incremental improvements are the normal mode in technical sciences, and should be so in IS design science as well. I return to this in the discussion at the end of the paper.

### 4.2 Examples of Design Research Questions

Turning to the research activity in design science, we can build on a list of kinds of research questions identified by Vincenti [29]. In figure 4 I classify them according to their place in the engineering cycle [8]. It turns out that these questions are easily recognizable in IS research too.

- *Why does it fail?* Failures are gold mines of information to improve technology [37]. IS research too investigates failures of projects to reach their targets, failures of implementations to achieve their goals, and failures of methods to deliver their promises.
- *Why does it work?* This is a common question in technical research, as many artifacts exist that do work but of which the underlying mechanisms are not well-understood. Examples mentioned earlier are the study of heat machines in the 19th century and of aircraft technology in the 20th. This and the previous question why an artifact fails always are combined with a third one, which is what actually happens when the artifact is used. This is an interesting analog to evaluation and policy research, which investigates the actual outcomes of an intervention in social systems and why these outcomes are produced [38]. In figure 4, these questions are lumped together in the category *problem investigation questions,* in which the goal is to find our how well stakeholder goals are achieved with artifacts they currently use.
- *What will be the problem-solving power of this proposed artifact?* Practical problem solving delivers a design of an artifact claimed to solve a practical

Design science research questions
- Problem investigation questions
  - Why does it fail?
  - Why does it work?
- Artifact validation questions
  - What will be the problem-solving power of this artifact?
- Conceptual questions
  - How to measure this?
  - How to compute this variable?

**Fig. 4.** Research questions identified from the literature

problem. The research question to be answered is then whether it will indeed solve the problem. One way is to try it and see what happens. A more rational way followed in design science is to predict what will happen and check whether this would help stakeholders. Earlier I decomposed this into two questions, namely a prediction (what effects will this artifact produce in this context?) and a valuation (what is the value of these effects for these stakeholders?) [28]. Another important question to answer is external validation, or in the words of Vincenti [29], assessing the certainty of these predictions for future artifacts once implemented. In figure 4, all these questions are called *validation questions,* in which the goal is to predict the properties of an artifact in a practical problem situation before it has been implemented in that situation.

– *How to measure this?* This question is as well-known in technical science as it is in social science. How to measure effort, speed, usability, maintainability, security, risk, or any of the other attributes relevant in IS research? The special constraint in design science is that measurement must be cost-effective not only for the researchers, but to be usable in practice it must also be cost-effective for practitioners. For example, it is of not much practical use to acquire knowledge about risk indicators that cannot be measured in practice.
– *How to compute this variable?* Answering research questions in the service of practical problem solving places the design science research under the constraints that solutions can actually be computed, not just mathematically proven to exist. This may involve trading mathematically rigorous methods that are not computable, or that are too expensive to use, for approximate methods that are computable and also cost-effective to use. The historian of technology Edwin Layton sees this as the characteristic feature that distinguishes an engineering science from physics [39, page 575]. An example in IS design science could be the development of practical techniques to estimate implementation cost of ERP systems, where these techniques may be less exact but more cost-effective than some other more accurate technique; combined with an empirical research to validate these techniques. Measurement and computation questions have been classified in figure 4 as

*conceptual questions,* in which the goal is to define constructs, indicators, and computations that are valid with respect to a class of phenomena.

This list of research questions is not exhaustive but it does indicate that sources of relevance of design science research include (1) the ability to contribute knowledge about practical problems, in particular about causes of failure and success, (2) the ability to predict the outcome of implementing an artifact in a context and (3) the satisfaction by this knowledge of the constraints of practical observability and computability.

This indicates a constraint on design knowledge that is not applicable to other kinds of scientific knowledge: applicability. Nineteenth century engineers contemplating to use the results of science found the results too abstract to be useful and have pointed out the fact that practical problem solvers cannot ignore conditions of practice [40, pages 692–693] [41, page 331]. In an ECOOP 2009 dinner speech, Bill Cook phrased this eloquently when he warned academics not to try transfer a solution to an abstraction of a real-world problem: Practitioners have to deal with the whole problem [42]. *Conditions of practice* are all natural and social factors present in a practical problem, including all natural causes and stakeholder-defined performance criteria that cannot be abstracted away from. An example of the difference between abstractions in basic science and the conditions of practice is given by Küppers in an analysis of the differences between thermodynamics and combustion technology [43]: In the natural science of thermodynamics, the goal of understanding a flame in a furnace is achieved when the shape of the flame, the flow pattern and the course of the reaction or the radiation pattern of the flames is understood. The combustion technologist needs to answer the same questions, but additionally needs to know whether the flame is stable (burns in the same place), when it does not oscillate, if the furnace will be damaged by turning the flame on or off, and whether a certain domain of regularity prescribed by safety requirements can be reached and maintained. The additional variables that interest the engineering researcher have their source in the fact that in real practical problems, variables cannot be abstracted away, and are relevant for stakeholder goals.

Conditions of practice are present in IS design science research too: An ERP implementation is subject to a large number of variables that cannot be wished away [44], process improvement is impacted by a large number of risk factors [45], etc. This has an important implication for research strategy, which is that design science researchers cannot stop when they have have understood a phenomenon in the laboratory, but must eventually *scale up* to investigate what happens under the conditions of practice of the intended practical problem situation. In the next section we discuss the implications for design science in more detail and discuss which part of Schön's dilemma has been touched by our analysis.

## 5 Discussion: Relevance and Applicability

Our analysis motivates a definition of *relevance* as suitability of an artifact or of knowledge to help achieving a goal, and *applicability* as sufficient incorporation of conditions of practice in a theory or in artifact behavior. Knowledge is applicable to a case if it can be related to the conditions of practice of this case. There is a one-way dependency, because non-applicability implies irrelevance. When a medical doctor investigates a patient, most of the results of medical science are applicable, because medicine is a practical science aiming a practical knowledge; but only some of it is relevant for the problem at hand. However, abstract knowledge that does not incorporate the conditions of practice of the problem at hand, is irrelevant for any practical goal. Now let us consider the implications of this distinction for design science.

Importantly, there is no particular implication for *research methods.* The fact that conditions of practice must eventually be included does not exclude a priori any scientific method from being used.

However, there is an implication for *research strategy:* Even if design science research starts investigating an artifact in the laboratory, it eventually needs to scale up to conditions of practice. For example the first prototype jet engines developed where small scale models that were investigated in the laboratory under controlled conditions, but these were subsequently scaled up to realistic sizes and eventually test models were used to propagate an airplane flown by a test pilot [32]. Scaling up is thus a requirement for design science. This means that some methods of real-world research such as case studies and action research become important towards the later stages of artifact development [46,47]. The only way to produce conditions of practice is to move to practice. I therefore agree with Benbasat & Zmud [2] that IS design research needs to produce context-rich knowledge. And I agree with Davenport & Markus [3] that evaluation and policy research [38] provide useful methods for doing so.

A third implication concerns *design theories.* If a significant number of conditions of practice must be incorporated, then design theories are likely not to be universal (nomothetic) but likely have a middle range generalization [48], an observation also made by Kuechler and Vaishnavi [45] but so far ignored by proposals for design theories [49]. The need for middle range diagnostic theories and treatment theories has also been observed in psychological practice [50].

A fourth implication is about *technology transfer.* If a theory does not relate to conditions of practice, then it will not be deemed relevant by practitioners. This can explain why, in a study of technology transfer at NASA, it turned out that managers are reluctant to use software technology that had been investigated empirically in the laboratory, and were more easily convinced by results from case study research [51]. My explanation is that managers can more easily relate case studies than laboratory research to their own conditions of practice. Technology transfer is risk taking, and managers need specific information to be able to estimate this risk.

Application of technology in a practical problem always implies applying knowledge about this technology, and calling this technology "transfer" is misleading, because a lot more is involved than simple transfer. As pointed out by Gardner [52, pages 9–12], application of design knowledge involves combining different conceptual frameworks, dealing with missing data and ill-defined variables, figuring out the interaction of conditions mentioned in different theories, and in general building a mini-theory of the case at hand. This is also pointed out by Van Strien [50]. Application of knowledge to practical problems is an underestimated problem.

Fifth, our framework (figure 2) indicates different *roles* to play in design science. Design science *researchers* operate on the interface of practical problem solving and artifact investigation. Research *managers,* mentioned earlier, operate on the interface of design science and the economy, matching economic realities with design science possibilities. *Entrepreneurs* also operate on this interface but take higher risks by speculating on future demand. It would be a tall order to ask of every design science project to be entrepreneurial in this sense, as some commentators seem to suggest in a panel discussion at ICIS 2002 [36, pages 337–338]. Studying existing artifacts to understand how and why they work and fail is a normal mode for design science that delivers applicable and potentially relevant results.

Finally, let us return to the dilemma of rigor versus relevance as posed by Schön. Our analysis has shown that design science research does not have to lead to a particular practical problem solving project, but should deliver applicable knowledge that may not be relevant to any current goal but is potentially relevant for some future and possibly unknown goal. Applicability must be achieved by incorporating conditions of practice and satisfying the constraints of practical measurability and computability mentioned earlier. Within these constraints, methods used in technical design science, such as pilot studies and test flights correspond closely to well-known methods in social science such as case studies and action research.

This deals with some of the issues of uncertainty and instability in practical situations mentioned by Schön. However, part of the source of instability of the subject of social science is the *historicity* of the subject: Human subjects may join the researcher in interpreting social phenomena, and eventually will learn about social theories, may internalize them and then change their behavior [53, pages 29 ff.]. This affects the applicability of IS design science theories, and hence their relevance in a given context. But this phenomenon, complex as it is, should not blind us for the fact that a large part of the problem of applicability of IS design theories is shared with technical design science, and can be answered in the same way.

Our research group has been using the framework presented in this paper for several years to structure PhD theses [54] and papers [55]. Our experience is that it helps to find the right questions to ask if we get stuck, and to improve our understanding of the problems we are aiming to solve. This experience should be followed up by a more objective evaluation and we plan to do so once enough experience has been collected. This will be an action research reflection in the sense that we have used our own artifact (the framework) to improve our own practice and will then reflect on the utility of this artifact to actually improve the practice.

**Acknowledgments.** Thanks are due to the anonymous reviewers, who made some useful improvement suggestions.

## References


1. Schön, D.: The Reflective Practitioner: How Professionals Think in Action. Arena (1983)
2. Benbasat, I., Zmud, R.: Empirical research in information systems: the practice of relevance. MIS Quarterly 23(1), 3–16 (1999)
3. Davenport, T., Markus, M.: Rigor vs. relevance revisited: response to Benbasat and Zmud. MIS Quarterly 23(1), 19–23 (1999)
4. Nunamaker, J., Chen, M., Purdin, T.: Systems development in information systems research. Journal of Management Information Systems 7(3), 89–106 (1990–1991)
5. Walls, J., Widmeyer, G., Sawy, O.E.: Building an information system design theory for vigilant EIS. Information Systems Research 3(1), 36–59 (1992)
6. March, A., Smith, G.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)
7. Hevner, A., March, S., Park, J., Ram, S.: Design science in information system research. MIS Quarterly 28(1), 75–105 (2004)
8. Wieringa, R.: Requirements Engineering: Frameworks for Understanding. Wiley, Chichester (1996), http://www.cs.utwente/nl/~roelw/REFU/all.pdf
9. Weber, R.: The problem of the problem. MIS Quarterly 27(1), iii–ix (2003)
10. Bunge, M.: Technology as applied science. In: Rapp, F. (ed.) Contributions to the Philosophy of Technology, pp. 19–39. Reidel, Dordrechtz (1974)
11. Godin, B.: The linear model of innovation: The historical reconstruction of an analytic framework. Science, Technology and Human Values 31(6), 639–667 (2006)
12. Cardwell, D.: Wheels, Clocks, and Rockets. A History of Technology. W.W. Norton & Company (1995)
13. Keller, A.: Has science created technology? Minerva 22(2), 160–182 (1984)
14. Carlson, W.: Innovation and the modern corporation. From heroic invention to industrial science. In: Krige, J., Pestre, D. (eds.) Companion to Science in the Twentieth Century, pp. 203–226. Routledge, New York (2003)
15. McKelvey, J.: Science and technology: The driven and the driver. Technology Review, 38–47 (1985)
16. Stokes, D.: Pasteur's quadrant: Basic science and technological innovation. Brookings Institution Press (1997)
17. Wise, G.: Science and technology. Osiris (2nd Series), vol. 1, pp. 229–246 (1985)
18. Gardner, P.: Representation of the relationship between science and technology in the curriculum. Studies in Science Education 24, 1–28 (1994)
19. DeSolla Price, D.: Of sealing wax and string. Natural History 84(1), 49–56 (1984)
20. Shapin, S., Schaffer, S.: Leviathan and the air pump: Hobbes, Boyle and the experimental life. Princeton University Press, Princeton (1985)
21. Middleton, W.K.: A history of the thermometer and its uses in meteorology. Johns Hopkins, Baltimore (1966)
22. Kuhn, T.: Mathematical versus experimental traditions in the development of physical science. In: Kuhn, T. (ed.) The Essential Tension, pp. 31–65. University of Chicago Press, Chicago (1977); Reprinted from The Journal of Interdisciplinary History, vol. 7, pp. 1–31 (1976)
23. Marsden, B.: Watt's Perfect Engine. Steam and the Age of Invention. Icon Books (2002)
24. Cardwell, D.: Turning points in Western technology. A study of technology, science and history. Science History Publications (1972)

25. Kline, S.: Innovation is not a linear process. Research Management 24(4), 36–45 (1985)
26. Reich, L.: The making of American industrial research: Science and business at GE and Bell, pp. 1876–1926. Cambridge University Press, Cambridge (1985)
27. de Vries, M.: 80 Years of Research at the Philips Natuurkundig Laboratorium. Pallas Publications (2005)
28. Wieringa, R.J.: Design science as nested problem solving. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, pp. 1–12. ACM, New York (2009)
29. Vincenti, W.: What Engineers Know and How They Know It. Analytical Studies from Aeronautical History. Johns Hopkins, Baltimore (1990)
30. Aitken, H.: Science, technology and economics: The invention of radio as a case study. In: Krohn, W., Layton, E., Weingart, P. (eds.) The Dynamics of Science and Technology. Sociology of the Sciences, II, pp. 89–111. Reidel, Dordrechtz (1978)
31. Lyytinen, K., King, J.: Nothing at the center? Academic legitimacy in the information systems field. Journal of the Association of Information Systems 5(6), 220–246 (2004)
32. Constant, E.: The Origins of the Turbojet Revolution. Johns Hopkins, Baltimore (1980)
33. Hughes, T.: Networks of Power. Electrification in Western Society, pp. 1880–1930. Johns Hopkins university Press, Baltimore (1983)
34. Laudan, R.: Introduction. In: Laudan, R. (ed.) The Nature of Technological Knowledge. Are Models of Scientific Change Relevant?, pp. 1–26. Reidel, Dordrechtz (1984)
35. Hughes, T.: The evolution of large technological systems. In: Bijker, W., Hughes, T., Pinch, T. (eds.) The Social Construction of Technological Systems, pp. 51–82. MIT Press, Cambridge (1987)
36. Kock, N., Gray, P., Hoving, R., Klein, H., Myers, M., Rockart, J.: Research relevance revisited: Subtle accomplishment, unfulfilled promise, or serial hyporcisy? Communications of the Association of Information Systems 8, 330–346 (2002)
37. Petroski, H.: To Engineer is Human: The Role of Failure in Successful Design. Vintage books (1992)
38. Pawson, R., Tilley, N.: Realistic Evaluation. Sage Publications, Thousand Oaks (1997)
39. Layton, E.: Mirror-image twins: The communities of science and technology in 19th century America. Technology and Culture 12(4), 562–580 (1971)
40. Layton, E.: American ideologies of science and engineering. Technology and Culture 17, 688–701 (1976)
41. Finch, J.: Engineering and science: A historical review and appraisal. Technology and Culture 2, 318–332 (1961)
42. Cook, W.: ECOOP 2009 banquet speech, `http://wcook.blogspot.com/2009/10/ecoop-2009-banquet-speech.html` (Accessed January 20, 2010)
43. Küppers, G.: On the relation between technology and science—goals of knowledge and dynamics of theories. The example of combustion technology, thermodynamics and fluid dynamics. In: Krohn, W., Layton, E., Weingart, P. (eds.) The Dynamics of Science and Technology. Sociology of the Sciences, II, pp. 113–133. Reidel, Dordrechtz (1978)
44. Daneva, M., Wieringa, R.J.: A requirements engineering framework for cross-organizational erp systems. Requirements engineering 11(3), 194–204 (2006)

45. 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)
46. Baskerville, R.: Distinguishing action research from participative case studies. Journal of Systems and Information Technology 1(1), 25–45 (1997)
47. Glass, R.: Pilot studies: What, why, and how. Journal of Systems and Software 36, 85–97 (1997)
48. Merton, R.: On sociological theories of the middle range. In: Social Theory and Social Structure, Enlarged edn., pp. 39–72. The Free Press (1968)
49. Gregor, S., Jones, D.: The anatomy of a design theory. Journal of the AIS 8(5), 312–335 (2007)
50. Van Strien, P.: Towards a methodology of psychological practice: The regulative cycle. Theory & Psychology 7(5), 683–700 (1997)
51. Zelkowitz, M., Wallace, D., Binkley, D.: Culture conflicts in software engineering technology transfer. In: 23rd NASA Goddard Space Flight Center Software Engineering Workshop (1998)
52. Gardner, P.: The relationship between technology and science: Some historical and philosophical reflections. part II. International journal of Technology and Design Education 5, 1–33 (1995)
53. Sayer, A.: Method in Social Science: A Realist Approach, 2nd edn. Routledge, New York (1992)
54. Mutschler, B.: Modeling and simulating causal dependencies on process-aware information systems from a cost perspective. PhD thesis, Univ. of Twente, Enschede (2008), http://eprints.eemcs.utwente.nl/11864/
55. Morali, A., Wieringa, R.J.: Risk-based confidentiality requirements specification for outsourced it systems. Technical Report TR-CTIT-10-09, Centre for Telematics and Information Technology, University of Twente, Enschede (2010)

# Artifact Types in Information Systems Design Science – A Literature Review

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.** Many information systems researchers designate their work as design science referring to the term "IT artifact" and the categorization systems that have emerged under this label. Alas, there is no consensus at this point as to what the research output in design science is and what types of artifacts exist. Using a widely accepted artifact typology would strengthen the scientific discussion and ease the categorization of contributions. Based on a literature review of all DESRIST publications and a special MISQ issue on design science, we derived such a typology. We identified eight relevant artifact types and related our typology to existing ones. With this contribution, we hope to enable a discussion about what legitimate design science outputs and their main types are.

**Keywords:** Design science, research output, IT artifact, typology, literature review.

## 1 Introduction

Design science has been established as a research paradigm in information systems for many years. According to the current consensus, design science is about designing IT artifacts [1, 2]. However, not even in the superordinate discipline of information systems research (ISR) has a common understanding of the IT artifact evolved. Current discussions in design science do not provide any grounding in existing publications, but are derived from analogies with natural or social science. Additionally, different views on the IT design artifact exist, making it difficult for the researcher to know which view to follow when classifying their own design research.

The purpose of this paper is to identify which types of IT design artifacts exist. Thereby, we hope to create a more uniform understanding, facilitating design output comparison. Additionally, we aim to reduce confusion about what a design output is and what it is not. To achieve this, we performed a literature review of design science publications. As literature, we selected the 2006-2009 DESRIST publications and the 2008 MIS Quarterly vol. 32 no. 4 special issue on design science, and qualitatively analyzed the statements about the design artifacts in those articles. The goal of the analysis was to identify artifact types that can be found in existing publications.

The paper is structured as follows: First, the current understanding of the IT artifact in design science is introduced. Then, our literature review and the design outputs found are presented. Following, results of the qualitative analysis are given and implications discussed. Finally, a conclusion is drawn.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 77–92, 2010.

## 2 Related Work and Background

The discussion about the object of research in information systems has been going on for some time [3-10], as summarized by Benbunan-Fich and Mohan [11] and Alter [12]. The sub-discipline of design science has developed more recently and accordingly offers less existing work on its research output. The link between information systems as behavioral science and design science is the overlap regarding their research objects. While related, the implications of the discussion about the "IT artifact" differ for the two approaches: Behavioral research usually produces theories about information systems, with information systems, or more general "IT artifacts" being the object of research. On the other hand, the output of design science are not theories as known from behavioral research but predominantly designs, so that the research output oftentimes is the same as the research object. This recursive relationship is discussed in other design disciplines as well, often with reference to Gadamer's "hermeneutical circle" [13].

For design science, Vaishnavi and Kuechler Jr. [14] identify the two perspectives on the research output from March and Smith [2] and from Purao [15]. March and Smith [2] state: "Design science products are of four types, constructs, models, methods, and implementations." For them, constructs or concepts are conceptualizations that "form the vocabulary of a domain". They see a model as "a set of propositions or statements expressing relationships among constructs". It represents "situations as problem and solution statements". Methods are based on constructs and models. They are "a set of steps (an algorithm or guideline) used to perform a task." Finally, an instantiation realizes "an artifact in its environment".

Purao [15], following Gregg et al. [16], has a perspective different from March and Smith. In his viewpoint, the most visible output of design research is "the situated implementation of an invention (artifact) as software or system". The implementation ensures that the design is feasible. However, for Purao, "two other kinds of outputs are more important for design research". The first are operational principles or reproducible knowledge. They are abstractions to explicate the intended behavior in accepted forms. The second kind of output is the "metaphorical understanding of how the artifact supports or controls the phenomenon of interest". Thereby, "the expected behavior of the phenomenon […] is articulated".

In addition to these two perspectives, Winter [17] and Bucher and Winter [18] extend the classification of March and Smith [2] by *theory*. Winter [17] argues: "Although theory building is not design science research, theories as 'intermediate' artifacts need to be included in the system of relevant artifacts for IS design science research."

Walls et al. [19] have a different view on theory. They propose "Information System Design Theory" (ISDT) to present research outputs of design science. An ISDT describes a *design product*, consisting of *meta-requirements* and a *meta-design*, and a *design process*. It combines different types of the categorization of March and Smith and includes additional information about the design.

Looking into further publications, Vahidov [20] proposes an output classification matrix that is aligned with the Zachman framework. As categories, he proposes *structure*, *behavior*, *motivation* and *instantiation*. As perspective, he proposes *analytical*, *synthetic*, *technological* and *implementation*. Any design is categorized along both dimensions to determine which cell of the matrix it fits in.

Carlsson [21] follows van Aken [22] in distinguishing three types of designs: "1) an object-design, which is the design of the IS intervention (initiative), 2) a realization-design, which is the plan for the implementation of the IS intervention (initiative), and 3) a process-design, which is the professional's own plan for the problem solving cycle and includes the methods and techniques to be used to design the solution (the IS intervention) to the problem."

In summary, some statements on the design science research output exist, but most are not grounded. We have the impression that many authors reference March and Smith or some other classification building on March and Smith in their design science papers. Few authors reflect on the quality of the classification used. A discussion has not yet taken place. Especially, a common view on the use of the categorization systems has not yet emerged.

## 3 Literature Review Approach

To help remedy the aforementioned situation, we performed a literature review to identify types of design science research output using a literature review.

We loosely follow a qualitative literature review methodology. Our methodology is based on the recommendations from [23], the methodologies described in [24] and the example paper [25]. Effectively, we applied a mix of coding and mutual agreement. Table 1 gives an overview of this approach. The remainder of this section describes each step in more detail.

**Table 1.** Sequence of steps to derive this article's typology

| Activity | Description | Results |
|---|---|---|
| Data set definition | Determine the set of articles to review | Set of 106 articles |
| Filtering of design articles | Identify the articles that contribute a design that is relevant for practitioners | Subset of 62 articles |
| Extraction of authors' categories | Identify if and how authors subsumed their output under any type or category of design | List of terms describing design categories |
| Determining individual typologies | Each researcher: abstract the identified terms into typology, grouped by similarity | Three individual typologies |
| Unified typologies | Unify the three individual groupings into one agreed-upon typology | Typology draft |
| Individual classification of articles | Each researcher: classify all design articles using the typology draft | Three classifications of design articles |
| Consolidation of classification | Discuss differences in classification and determine if differences affect typology | Unified classification of articles, final typology |

### 3.1 Data Set Definition and Filtering of Design Articles

The data set contains 106 papers; 102 papers from DESRIST 2006-2009 conferences and 4 papers from the MISQ vol. 32 no. 4 special issue. Arguably, the data set is far from comprehensive. This might lead to an incomplete typology. At the same time, the selected publications had an explicit focus on design science and arguably offered a better look at the complete bandwidth of design science in ISR than publications with an otherwise specialized focus. As we are not interested in quantitative statements and consider our study to be exploratory, we feel that this data set is of sufficient size and quality.

The research authors of this paper were involved in the literature review and interpretation; the research was conducted between December 2nd, 2009 and January 4th, 2010. By using publications that are specialized on design science as our data set, we expect to cover the majority of IT artifact types designed in actual research projects.

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 papers to exclude. We decided to include papers presenting prescription-driven design science according to van Aken [22] and papers prescribing design and action according to Gregor [26]. Based on these two papers, we developed two questions to classify 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.

Each of the researchers looked at each of the 106 papers as to whether it contained a designed artifact, using the two questions presented above. In 38 cases we independently classified a paper as not presenting a designed artifact. In 12 cases a consensus could be reached after a discussion. After the discussion, 62 of the 106 papers remained that we considered presenting a design within the realms of information system design science.

### 3.2 Extraction of Author's Categories

For the remaining papers, we extracted statements from the papers about what the proposed design is. Additionally, we extracted what type of design artifact the paper's authors classified their design as. This step did not involve any interpretation of the terms found; they were taken from the papers unchanged. A summary of types found in the 62 papers can be seen in table 2.

### 3.3 Individual Categories

We continued by interpreting the information found. Our aim was to identify a set of types that would be coherent and classify all artifacts found. Hence, the next step was for each researcher to look at the terms extracted from the 62 papers to come up with his own set of types. The guideline was to identify types of design research outputs, respectively IT artifacts that are structurally different, requiring different description meta-models. We believe that types, differentiated along structure, are important because they require different descriptions, evaluation techniques and research methodologies. To expose the process of our research, the types found by each of the researchers are shown in table 3.

**Table 2.** Extraction of authors' categories (multiple terms per paper possible)

| Category | Synonymous usage | Paper |
|---|---|---|
| Algorithm | | [27]; [28]; [29] |
| Approach | | [30]; [31] |
| Architectural style | | [32] |
| Architecture | Tool architecture | [33]; [34]; [35] |
| Concept | Modeling concept | [36]; [33]; [37] |
| Construct | | [36] |
| Design artifact | | [38] |
| Design guidelines | | [39] |
| Design implications | | [40] |
| Evaluation | | [41]; [42]; [43]; [40]; [44]; [45]; [46] |
| Framework | Process Framework | [41]; [47]; [48]; [39]; [49]; [50] |
| Grammar | Modeling grammar | [51]; [37] |
| Graphical representation | | [35] |
| Information System | | [52]; [53] |
| Instantiation | | [39]; [54]; [55] |
| IT-artifact | | [33]; [56]; [57] |
| Meta-Design | | [50] |
| Method | Method fragments, situational Method | [58]; [51]; [59]; [47]; [52]; [45]; [54]; [60]; [61]; [62]; [37] |
| Methodology | Modeling methodology | [30]; [47]; [63] |
| Metric | | [42]; |
| Model | Model prototype, design model | [64]; [65]; [66]; [31]; [49]; [46]; [55]; [53] |
| Principles | Design principles | [43]; [48] |
| Process | | [30] |
| Protocol | Protocol extension | [32]; [67] |
| Prototype | Software Prototype | [51]; [66]; [43]; [40]; [68]; [69] |
| Recommendation | | [44] |
| Rule | | [70] |
| Simulation platform | | [71] |
| System artifact | | [72] |
| System design | | [73]; [48]; [74]; [75]; [72]; [76]; [77] |
| Theory | Design Theory, Theory nexus | [78]; [79]; [65]; [41]; [44]; [39]; [80]; [81]; [54]; [57] |
| Tool | | [33] |
| Typology | | [36] |
| No term matching | *The paper delivers a designed artifact but it is not explicitly subsumed under any term* | [82]; [83]; [84] |

### 3.4 Unified Typology

Looking at the number of types found in the previous step, it is obvious that the granularity varies. To derive an agreed-upon typology, we discussed every proposed type. For some types, we noticed that we used different names for the same thing (e.g. *technology method* vs. *algorithm*). In these cases, we decided to agree on one of the names used. In some cases we used a different granularity, for example, one typology contained one type where another differentiated several types. In these cases, we had to discuss which types would require similar description structures. Finally, some types were vague and covered a large number of different concepts, all of which were structurally similar. In those cases we introduced an auxiliary criterion *use* (as in *how is the artifact used with regard to a system?*), which helped to uncover possible differences between subtypes and allowed to map them to existing types or to introduce new narrower types.

**Table 3.** Author's personal typologies

| **Author A** | **Author B** | **Author C** |
|---|---|---|
| Method | Construct | System Design |
| - Modeling Language | Model | Method |
| - Methodology | - Cost model | - Situational Method |
| - Evaluation | - Business model | - Guideline |
| Software | Requirements | - Design Method |
| - Prototype | Software architecture | - Business Method |
| - Algorithm | Instantiation | - Evaluation Method |
| - System | Method | - Learning Method |
| Model | - Technology Method | (Evaluation) Framework |
| - Algorithm | - Business Method | Instantiation |
| - Data Model | - Modeling approach | - Prototype |
| - Metric | IS curriculum | - Tool |
| - Generalized Description | Machine | (Modeling) Language |
| - Principle | | Design Theory |
| - Architectural Style | | - For Method |
| - Pattern | | - For System |
| - Guideline | | - Meta-Design |
| - Typology | | Algorithm/Protocol |
| - Taxonomy | | Model |
| - Ontology | | - Cost Model |
| | | - Concept |
| | | - Ontological Design |
| | | - Data Model |
| | | - Typology |
| | | - Architecture |
| | | - Metric |
| | | - Strategy |
| | | Curriculum |
| | | Codified Experience |
| | | - Design Principles |
| | | - Principles |
| | | - Architectural Style |
| | | - Pattern |

Furthermore, we excluded two types that arguably are designs: for *implementation* we agreed that it is used to evaluate some more general design proposition. We were able to reclassify all papers stating to present an implementation as *system design*. For *IS curriculum* we agreed that it is not considered an IS research output as it is not related to an information system and to ignore the type for the further proceeding.

After analysis of term the individual classifications, we agreed upon eight types, listed in table 4.

**Table 4.** Artifact typology by number of occurrences (multiple types/paper permitted)

| **Artifact Type** | **Use** | **Structure** | **Paper** |
|---|---|---|---|
| System design | Description | Structure or behavior-related description of a system, commonly using some formalism (e.g. UML) and possibly text | [73]; [48]; [74]; [75]; [72]; [82]; [76]; [77]; [56]; [79]; [33]; [78]; [31]; [34]; [66]; [40]; [52]; [64]; [43]; [68]; [53]; [29]; [71]; [81]; [69]; [51] |
| Method | Support | Definition of activities to create or interact with a system | [58]; [51]; [59]; [47]; [52]; [45]; [54]; [61]; [62]; [37]; [83]; [44]; [30]; [39]; [41]; [37]; [57]; [63]; [83] |
| Language/ Notation | Support | A (generally formalized) system to formulate statements that represents parts of reality | [65]; [37]; [63]; [55]; [35]; [51] |
| Algorithm | Description | Executable description of system behavior | [27]; [28]; [29]; [67]; [61]; [32] |
| Guideline | Support | Suggestion regarding behavior in a particular situation (if in situation X do Y) | [36]; [39]; [43]; [70]; [84] |
| Requirements | Description | Statement about System (A system of type X should have some property Y [because of Z]) | [80]; [49]; [81] |
| Pattern | Support | Definition of reusable elements of design with its benefits and application context | [63]; [32] |
| Metric | Support | A mathematical model that is able to measure aspects of systems or methods | [42] |
| *Re-classified as 'no design'* | | *After reading the whole paper the contribution was re-classified as not delivering/addressing a design artifact.* | [60]; [38]; [46]; [50]; |

In our typology, we define *system design* to be a description of an IT-related system. The description can be on any granularity level and can focus on any aspect like structure, process, and interactions. Examples are “software architecture”, “enterprise architecture”, “database schema” and “business process diagram”.

For us *requirements* are statements about a system in the form: “A system of type X should have some property Y.” Optionally a reason for the requirement can be given. Generally, requirements restrict the design space for a system design. Examples are: “A life-supporting system should never fail.” “A multi-tenancy system should never expose private data to other tenants.”

In our definition a *method* consists of activities, possibly in some order, that are performed by people in order to support the system development. Methods often define results/deliverables of activities and roles. Examples are “software engineering method”, “enterprise architecture method”, “requirements analysis method”, and “organizational change method”.

An *algorithm* is in some kind similar to a method in that it describes a sequence of activities. However, an algorithm is executed by a computer. In our definition, it is an executable description of a system behavior. Examples are “sorting algorithm”, “data mining algorithm”, and “protocol”.

A *pattern* provides generalized system design elements that can be used for many different kinds of system designs. In that sense, it is used not to describe how a specific system should look like. Rather, a pattern provides support to create such a system design. Patterns exist for programming, software architecture, enterprise architecture, organizational design etc. Usually, the pattern is described with its benefits and the context of application. Examples are “singleton”, “asynchronous message queue”, “service-oriented architecture”, and “matrix organization”.

A *guideline* provides a generalized suggestion about system development. In that sense it is similar to a pattern, but does not contain system design elements or statements about these elements. It does not have a fixed structure, but will usually make statements like: “In situation X one could/should do Y.” Examples are “If high code quality is required, pair programming should be used.” “If a project is late, don’t add more people.”

A *language / notation* provides concepts and their interrelation, used to support system development. A graphical notation might include modeling elements and rules how these elements can be related. Often, a language / notation is referenced as a result type in a method. Examples are “entity-relationship model”, “business process modeling notation”, and “object-oriented programming language”.

Finally, in our typology, we define *metric* as some kind of model that is used to evaluate aspects of a system design or a support for a system design. A metric will provide a conclusion about the evaluated construct. Usually, a metric is a mathematical model, but qualitative metrics are also possible. Examples for metrics are “business case”, “architecture evaluation model” and “cyclomatic complexity”.

### 3.5 Classification of Papers by the Typology

To verify how uniform our understanding of the eight types is and to ensure that we did not miss a type, each researcher reclassified each of the 62 papers using the unified typology. For this, only the extracted information from the first iteration and the papers' abstracts could be used. For 31 papers, all researchers used the same types. Two of these papers contained an IS curriculum, which we did not include in our final typology as discussed above. For 12 papers, a consensus could be reached after a short discussion. One further paper was classified as IS curriculum. We had to check the remaining 19 papers for two possibilities: whether the classification was ambiguous or impossible because the design was not clearly described in the excerpts reviewed by us. Or if they were an instance of a type that might still be missing from our typology. We re-read all papers in more detail and then discussed our individual findings. For five papers, we concluded that the paper actually did not contain any design. For the other 14 papers, a type could be identified after continued reading. We did not come across any papers that required adding a completely new type. Therefore, we believe that the typology is complete with respect to the data set used.

# 4 Discussion

The two major aspects of this article are to get a better understanding about what a design artifact in design science is and what types of artifacts can be separated. At this point we want to reflect on the results presented in the previous section, addressing the approach, the types identified and the relation to the existing artifact categorization systems.

## 4.1 Review of Our Approach

In contrast to other design artifact categorization systems (cf. section 2), we chose to derive the artifact types from existing literature. The reflection of our approach must therefore focus on the quality of the data and our choice regarding its interpretation.

We noted that within the DESRIST publication the heterogeneity of notions about "design" and "design science" converged, while the quality with which contributions and mapping to a particular artifact type increased. This is not surprising for a young discipline and a young conference, but highlights the fact that the notion of what design is and what design artifacts are is not yet stable. The possibility that our typology is affected by such a shift in notions is possible but not a problem, as our aim is to facilitate discussion and the narrowing of understanding about artifacts and not to settle it.

As described above, we analyzed the articles on two levels: the first level included the title, the abstract and the keywords; the second level was to look at the whole article and read as much as necessary. In principle it might be that the understanding of what an article is about shifts, depending on how much one reads. This could affect both the decision about whether a design is presented as well as the categorization of the artifact. This risk is mitigated by several factors: we do not depend on quantitative data, so the possibility that the type of an artifact changes does not change our results. Also, we are focused on what the authors state their artifact is, as we want to find out what researchers consider artifacts to be. If an author states clearly in the abstract that, for example, a method is constructed, we considered this to be the relevant piece of information. We wanted to analyze what authors claim their design is; we did not want to determine what the authors have actually designed. Only if the first level of analysis did not contain such a clear categorization by the author we did consider the full text, which further reduced the number of possible occurrences.

We attempted to separate different types by their internal structure; that is, by the constituting elements and their relations. This separation took place based on the terms in table 3, which describes the researchers' individual interpretation of possible type candidates. The structures were not derived directly from the structures inherent in the reviewed articles but were introduced as needed, when a decision had to be made whether to keep two terms as separate types or to merge them into one. One might argue that this approach facilitates ad-hoc decisions. Yet, looking at our results, we believe that the types we identified are justifiably different from each other and are defined in a way that is comprehensible by other researchers and practitioners in the field.

### 4.2 Review of Our Results

The eight final types are introduced in detail in section 3.4. Here, we want to discuss some of the backgrounds and intermediate candidates.

Several of the reviewed articles used the term *model* to describe their contributed artifact (cf. table 2), as did all researchers in this paper (cf. table 3) in their intermediate grouping of terms. Consequently, *model* was a candidate artifact type for a while, but was excluded in the end. We found it to be too generic and in any particular instance shared too many characteristics with other types. Design models are covered by system design; mathematical models (at least in our sample) could be all subsumed under metrics. Other models were more aptly described as languages or notations. The possibility remains that in a larger data set we would find another type of model. In such a case, it would most likely still be possible to introduce specialized types of artifacts instead of returning to a general type *model*.

In this and other discussions the dimension of *use* was helpful. We identified three different uses that the IT design research outputs had:

- Description: How should a system look like?
- Support: How do I create a system design and the IT system?
- Evaluation: What properties does a system have?

While we succeeded to refer to structure as our only criterion to differentiate the types, *use* was very helpful in getting a first and general understanding why two types should or should not be differentiated. With a better understanding how two types are different, it was then possible to also identify their structural differences.

Another term highly debated within the community is *design theory* (e.g. [1, 19, 85]). Several authors of reviewed articles explicitly referenced *design theory* or *theory* with focus on design (cf. table 2). We identified two views on what such a theory would be. One view considered a design theory to be a design augmented by theoretical background and proposition about how instances of the design would behave in real life. We extracted such a design and categorized it under the appropriate type. A second view considered design theories to be very high level and generalized statements about design, which does not necessarily contain any design fragments itself. This view does not contain any design and was excluded from the review.

The eight identified types are somewhat related, as to be expected. *Guideline* and *pattern* have similarities – a pattern is a specialized form of guideline, as well as *guideline* and *requirement*, as they both contain descriptions of goals. Also, the level of abstraction is different: *system design* subsumes low level system designs as well as architecture descriptions and business processes, whereas *algorithm* and *metric* are more precisely and narrowly defined. Such "unevenness" might be unsatisfying or aesthetically unpleasing for a typology. This is, however, no concern to us. For one, the limited data set in a developing field, we discussed earlier is partly responsible for this. Also, we would evaluate the quality of the typology pragmatically, that is, by looking at the ease with which articles can be classified into one of our types. For us, based on the available data, this typology worked sufficiently well.

### 4.3 Relation to Larger IT-Artifact Discussion

In section 2 we have discussed other definitions and typologies of artifacts that we could find within the information systems design science literature. When comparing those to our results, several differences become apparent. The literature identifies fewer types of artifacts and they are of a more abstract nature. Our typology shares with the published ones the separation between structural and behavioral artifacts (*model* / *method* in case of March and Smith [2] and *design product* / *design process* of Walls et al. [19]).

More insightful than comparing individual types might be to contrast the different groups of people addressed by the typologies. We argue that previous typologies of the IT artifact had the research community in mind, when deriving their typologies: March and Smith [2] with their the artifact type *construct* describe it as the vocabulary, with which to reason about design. Purao [15] explicitly considers the knowledge *about* an artifact as a relevant output. Similarly, Walls et al. [19] and Winter [17] focus on the design theory as a central output of design research. We agree with these authors in that all these outputs are relevant for design science and that building knowledge *about* designs is a crucial task for design science. Nevertheless, in our view the separation along the lines of who can make use of the artifacts (researcher or practitioner) is relevant and meaningful for the field of design science and that any discussion about types of artifacts should start with the "first order objects", namely the designs that are both object and result of research. The "second order objects", such as theories, representing knowledge about "first order objects" can follow. Any typology of artifacts though, should be grounded in the available literature, if only to have a chance to be established by its own outputs and not as possibly inaccurate analogies to other sciences.

## 5 Conclusion

The purpose of this paper was to identify types of design research output. By qualitatively analyzing all DESRIST publications and four MISQ papers, we have created a typology containing eight distinct types (see table 4). As we used 62 papers containing design research output to establish the typology, we are confident we covered most types currently relevant for the discipline. To our best knowledge, we are the first to ground such a typology in existing literature. Using the typology, researchers can clarify the object of their research, possibly also facilitating focus and usage of suitable research methodologies. Thereby, confusion of reviewers and readers is reduced and comparability of designs increased; the level of standardization as well as the quality of scientific communication within the design science community would increase.

Based on our grounded typology, a discussion can take place as to whether further types should be the object of design science research. It is possible that there are types that have not yet been presented. Reasons might be that such types are difficult to design, difficult to evaluate or even that a type has not yet been relevant. A periodical evaluation of "artifacts in use" can supply valuable insight for the ongoing discourse about the elements and boundaries of the discipline. A follow-up discussion could also sharpen the understanding of theory in design science research and its relevant artifacts.

## References

1. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS QUART 28, 75–105 (2004)
2. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
3. Orlikowski, W.J.: Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations. Organization Science 11, 404–428 (2000)
4. 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)
5. Benbasat, I., Zmud, R.W.: The Identity Crisis Within the IS Discipline: Defining and Communicating the Discipline's Core Proporties. MIS QUART 27, 183–194 (2003)
6. Agarwal, R., Lucas Jr., H.C.: The Information Systems Identity Crisis: Focusing on High-Visibility and High-Impact Research. MIS QUART 29, 381–398 (2005)
7. Alter, S.: Sidestepping the it artifact, scrapping the is silo, and laying claim to "systems in organizations". Communications of AIS 12, 494–526 (2002)
8. Alter, S.: 18 reasons why it-reliant work systems should replace "the it artifact" as the core subject matter of the is field. Communications of AIS 12, 365–394 (2003)
9. Alter, S.: The is core – xi sorting out issues about the core, scope, and identity of the is field. Communications of AIS 12, 607–628 (2003)
10. Davis, G.B.: Information Systems Conceptual Foundations: Looking Backward and Forward. In: IFIP TC9 WG9.3 International Conference on Home Oriented Informatics and Telematics, pp. 61–82. Kluwer, B.V., Deventer (2000)
11. Benbunan-Fich, R., Mohan, K.: Information technology and systems - iii research publications in systems development during 2000-2004. Communications of AIS 17, 2–33 (2006)
12. Alter, S.: Work systems and it artifacts – does the definition matter? Communications of AIS 17 (2006)
13. Snodgrass, A., Coyne, R.: Is Designing Hermeneutical? Architectural Theory Review. Journal of the Department of Architecture 1, 65–97 (1997)
14. Vaishnavi, V.K., Kuechler Jr., W.: Design Science Research Methods and Patterns: Innovating Information and Communication Technology. Auerbach (2007)

15. Purao, S.: Design Research in the Technology of Information Systems: Truth or Dare. Information Systems, Atlanta (2002)
16. Gregg, D.G., Kulkarni, U.R., Vinzé, A.S.: Understanding the Philosophical Underpinnings of Software Engineering Research in Information Systems. Information Systems Frontiers 3, 169–183 (2001)
17. Winter, R.: Design science research in Europe. European Journal of Information Systems 17, 470–475 (2008)
18. Bucher, T., Winter, R.: Dissemination and Importance of the “Method” Artifact in the Context of Design Research for Information Systems. In: DESRIST 2008, Atlanta, Georgia (2008)
19. 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)
20. Vahidov, R.: Design Researcher’s IS Artifact: a Representational Framework. In: DESRIST 2006, CGU, Claremont, CA (2006)
21. Carlsson, S.A.: Towards an Information Systems Design Research Framework: A Critical Realist Perspective. In: DESRIST 2006. CGU, Claremont (2006)
22. 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)
23. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS QUART 26, xiii–xxiii (2002)
24. Creswell, J.W.: Research Design - Qualitative, Quantitative, and Mixed Methods Approaches. SAGE Publications, Thousend Oaks (2009)
25. Stelzer, D.: Enterprise Architecture Principles: Literature Review and Research Directions. In: TEAR 2009, pp. 21–36 (2009)
26. Gregor, S.: The Nature of Theory in Information Systems. MIS QUART. 30, 611–642 (2006)
27. Bryant, W.: Mixer Selection in Peer-to-Peer Ad hoc Collaboration: A Design Science Approach. In: DESRIST 2007, CGU, Pasadena, CA (2007)
28. Li, S., Lee, R., Lang, S.-D.: Mining Distance-based Outliers from Categorical Data. In: DESRIST 2007, CGU, Pasadena, CA (2007)
29. Huang, W., Stohr, E.: Design and Implementation of a Business Process Rules Engine. In: DESRIST 2007, CGU, Pasadena, CA (2007)
30. Carlsson, S.A., Henningsson, S., Hrastinski, S., Keller, C.: An Approach for Designing Management Support Systems: The Design Science Research Process and its Outcomes. In: DESRIST 2009. ACM, Malvern (2009)
31. Li, L., Vashnavi, V.K., Vandenberg, A., Metikurke, S.: Generating Dynamic and Adaptive Knowledge Models for Web-Based Resources. In: DESRIST 2006, CGU, Claremont, CA (2006)
32. Erenkrantz, J., Gorlick, M., Taylor, R.N.: Rethinking Web Services from First Principles. In: DESRIST 2007, CGU, Pasadena, CA (2007)
33. Abhichandani, T., Kosaka, K., Chatterjee, S.: Designing a P2P Ad-hoc Secured Collaboration Tool. In: DESRIST 2006, CGU, Claremont, CA (2006)
34. Nguyen, V., Lee, R., Dutta, K.: An Aspect Architecture for Modeling Organizational Controls in Workflow Systems. In: DESRIST 2006, CGU, Claremont, CA (2006)
35. Woodard, J.: Modeling Architectural Strategy Using Design Structure Networks. In: DESRIST 2007, CGU, Pasadena, CA (2007)
36. Berente, N., Lyytinen, K.: The Iterating Artifact as a Fundamental Construct in Information System Design. In: DESRIST 2006, CGU, Claremont, CA (2006)

37. D'Aubeterre, F., Singh, R., Iyer, L.: Secure Activity Resource Coordination: A Method to Design Secure Business Processes. In: DESRIST 2008, Atlanta, Georgia (2008)
38. Umapathy, K., Purao, S., Barton, R.R.: Designing Enterprise Integration Solutions - Effectively. In: DESRIST 2008, Atlanta, Georgia (2008)
39. Gurzick, D., Lutters, W.G.: Towards a Design Theory for Online Communities. In: DESRIST 2009. ACM, Malvern (2009)
40. Roussinov, D., Chau, M.C.-L.: Web Fact Seeking For Business Intelligence: a Meta Engine Approach. In: DESRIST 2006, CGU, Claremont, CA (2006)
41. 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)
42. Tremblay, M.C., Berndt, D.J., Hevner, A.R.: Measuring Information Volatility in a Health Care Information Supply Chain. In: DESRIST 2009. ACM, Malvern (2009)
43. Vaishnavi, V.K., Vandenberg, A., Zhang, Y., Duraisamy, S.: Towards Design Principles for Effective Context- and Perspective-Based Web Mining. In: DESRIST 2009. ACM, Malvern (2009)
44. Pries-Heje, J., Baskerville, R.: Designing Organizational Change in IT: A Theory Nexus. In: DESRIST 2006, CGU, Claremont, CA (2006)
45. Gericke, A., Fill, H.-G., Karagiannis, D., Winter, R.: Situational Method Engineering for Governance, Risk and Compliance Information Systems. In: DESRIST 2009. ACM, Malvern (2009)
46. van Steenbergen, M., Bos, R., Brinkkemper, S.: The application of a prescriptive and a constructive perspective to design-science research. In: DESRIST 2008, Atlanta, Georgia (2008)
47. 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, CA (2006)
48. Chung, W.: Designing Web-based Business Intelligence Systems: A Framework and Case Studies. In: DESRIST 2006, CGU, Claremont, CA (2006)
49. Even, A., Shankaranarayanan, G., Berger, P.D.: Economics-Driven Design of a Data Warehouse. In: DESRIST 2007, CGU, Pasadena, CA (2007)
50. Eden, H., Fischer, G., Giaccardi, E., Ye, Y.: Report on NSF Science of Design Team Project: "A Meta-Design Framework for Participative Software Systems. In: DESRIST 2007, CGU, Pasadena, CA (2007)
51. Lee, J., Wyner, G.M., Pentland, B.T.: Process grammar as a tool for business process design. MIS QUART. 32, 757–778 (2008)
52. An, Y., Dalrymple, P.W., Rogers, M., Gerrity, P., Horkoff, J., Yu, E.: Collaborative Social Modeling for Designing a Patient Wellness Tracking System in a Nurse-Managed Health Care Center. In: DESRIST 2009. ACM, Malvern (2009)
53. Garrett, N., Thoms, B., Soffer, M., Ryan, T.: Extending the Elgg Social Networking System to Enhance the Campus Conversation. In: DESRIST 2007, CGU, Pasadena, CA (2007)
54. Pries-Heje, J., Baskerville, R.: The design theory nexus. MIS QUART. 32, 731–755 (2008)
55. 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, CA (2007)

56. 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, CA (2006)
57. Zheng, G., Vaishnavi, V., Vandenberg, A.: A Multidimensional and Visual Exploration Approach to Project Prioritization. In: DESRIST 2008, Atlanta, Georgia (2008)
58. Ramaprasad, A., Papagari, S.S.: Ontological Design. In: DESRIST 2009. ACM, Malvern (2009)
59. Liu, X.M., Wyner, G.M.: Coordination Analysis: A Method for Deriving Use Cases from Process Dependencies. In: DESRIST 2009. ACM, Malvern (2009)
60. Johnstone, M.N., Venable, J.R.: Enacting Design Science with Action Research: An Illustrative Case Study in Methodology Development. In: DESRIST 2008, Atlanta, Georgia (2008)
61. Chou, C.-h., Sinha, A.P., Zhao, H.: Designing Web Page Classifiers as Alternatives to Commercial Internet Filters. In: DESRIST 2008, Atlanta, Georgia (2008)
62. Ghaisas, S., Venkatesh, R., Kanhere, A., Kumar, A., Kumar, M., Rose, P.: Identifying unobvious requirements — an exemplar of the Design Science approach. In: DESRIST 2008, Atlanta, Georgia (2008)
63. Athenikos, S.J., Song, I.-Y.: A Framework of Ontology-based Modeling Patterns. In: DESRIST 2008, Atlanta, Georgia (2008)
64. Baloh, P., Desouza, K.C.: Towards knowledge needs-technology fit model for knowledge management systems. In: DESRIST 2009. ACM, Malvern (2009)
65. Mettler, T., Rohner, P.: Situational Maturity Models as Instrumental Artifacts for Organizational Design. In: DESRIST 2009. ACM, Malvern (2009)
66. Su, J., Parsons, J.: Analysis of Data Structures to Support the Instance-based Database Model. In: DESRIST 2006, CGU, Claremont, CA (2006)
67. Dutta, A.K., Molle, M., Phatak, T.: Protocol Alchemy: Designing a TCP-Based Approach to Support Large-Scale Deployment of VoIP on the Internet. In: DESRIST 2006, CGU, Claremont, CA (2006)
68. Murphy, J.D., Petter, S., Cheng, K., Briggs, R.O.: Hitting the Collaboration Target: Computer-Guided thinkLet Selection. In: DESRIST 2008, Atlanta, Georgia (2008)
69. Vandenberg, A., Vaishnavi, V.K., Li, L., Pandya, M.: LAIDBACK: A Workbench for User-Driven Dynamic Web Mining. In: DESRIST 2007, CGU, Pasadena, CA (2007)
70. Parsons, J., Wand, Y.: Using Cognitive Principles to Guide Classification in Information Systems Modeling. MIS QUART. 32, 839–868 (2008)
71. Oh, D., Shih-Fen, C., Dan, M., Bapna, R.: Designing an Experimental Gaming Platform for Trading Grid Resources. In: DESRIST 2007, CGU, Pasadena, CA (2007)
72. Vahidov, R., He, X.: Design for Situated Decision Support: Application to Personal Finance. In: DESRIST 2008, Atlanta, Georgia, pp. 1–6 (2008)
73. Reynoso, J.M.G., Olfman, L., Ryan, T., Horan, T.: An Information Systems Design Theory for an Expert System for Training. In: DESRIST 2006, CGU, Claremont, CA (2006)
74. Abbasi, A., Hsinchun, C.: CyberGate: A Design Framework and System for Text Analysis of Computer-Mediated Communication. MIS QUART. 32, 811–837 (2008)
75. Reiss, S.P.: Designing Internet-Based Software. In: DESRIST 2007, CGU, Pasadena, CA (2007)
76. Parsons, J., Ralph, P.: Generating Effective Recommendations by Exploiting Viewing Time and Item Attributes. In: DESRIST 2007, CGU, Pasadena, CA (2007)
77. Etezadi, J.: iMade (Internet Multi-Attribute Decision Expert): A system for analysis of multi-attribute decision problems. In: DESRIST 2008, Atlanta, Georgia (2008)

78. Burkhard, R.J., Horan, T.A.: A Design Theory Approach to Incorporating Emotional Intelligence into Group Support Systems. In: DESRIST 2006, CGU, Claremont, CA (2006)
79. Gregor, S., Jones, D.: The formulation of an Information Systems Design Theory for e-learning. In: DESRIST 2006, CGU, Claremont, CA (2006)
80. Park, Y., El Sawy, O.A.: Towards a Design Theory for Hedonic Systems: Delivering Superior User Experience in the Digital Home Entertainment Context. In: DESRIST 2008, Atlanta, Georgia (2008)
81. Singh, R., D'Aubeterre, F., Iyer, L., Salam, A.F., Ehrhardt, R.: Secure Semantic eBusiness Processes: A Design Theoretic Approach to Secure Semantic Systems. In: DESRIST 2007, CGU, Pasadena, CA (2007)
82. Green, K.E., Gugerty, L.J., Witte, J.C., Walker, I.D., Houayek, H., Rubinstein, J., Daniels, R., Turchi, J., Kwoka, M., Dunlop, I., Johnson, J.: The Animated Work Environment: A Design-Research Exemplar Supporting Social and Virtual Collaboration in an Increasingly Digital Society. In: DESRIST 2008, Atlanta, Georgia (2008)
83. Amoussou, G.-A., Cashman, E., Steinberg, S.: Building an Interdisciplinary Community to Learn and Teach Creativity and Design. In: DESRIST 2007, CGU, Pasadena, CA (2007)
84. Petter, S., Khazanchi, D., Murphy, J.D.: A design science based evaluation framework for patterns. In: DESRIST 2008, Atlanta, Georgia (2008)
85. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the AIS 8, 312–335 (2007)

# Quo Vadis, Design Science? – A Survey of Literature

Kalle Piirainen[1], Rafael A. Gonzalez[2], and Gwendolyn Kolfschoten[2]

[1] Lappeenranta University of Technology, Faculty of Technology Management, Department of Industrial Management
Skinnarilankatu 34, 53850 Lappeenranta, Finland
Kalle.Piirainen@lut.fi

[2] Delft University of Technology, Faculty of Technology, Policy and Management, Department of Systems Engineering
Jaffalaan 5, 2628 BX Delft, The Netherlands
{R.A.Gonzalez,G.L.Kolfschoten}@tudelft.nl

**Abstract.** Design science (DS) is increasingly a popular approach for research with a problem-solving perspective. The literature on DS has exploded in the past few years creating a lively discussion emphasizing the balance between rigor and relevance in research, particularly within the information systems field. However, there is still inconsistency with regards to the terminology, the underlying philosophy and the strategy to follow when doing DS. The field(s) into which DS can contribute or in which it can be placed is also an open issue. The advent of special issues, tracks and conferences on the subject is a sign of this and perhaps a suggestion that it constitutes a field on its own. This paper reports a survey on the most influential literature on DS and provides an analysis of it in order to facilitate the discussion, clarify the terminology and contribute to making DS more actionable for researchers.

**Keywords:** Design science, design research, information systems research, bibliometrics.

## 1 Introduction

Traditionally the main and even the only aim of science has been production of knowledge, describing how the world is and how it works. However, Herbert Simon [39] argued that professionals and researchers alike are continuously engaged in problem-solving, where many of the solutions involve building an artifact which is implemented in an existing system or organization. Significant progress has been made especially in technology through design and engineering artifacts to solve practical problems, rather than first trying to understand the underlying mechanisms through scientific inquiry before attempting to solve the problem. March and Smith [28] further discussed the difference between natural (descriptive) and design (prescriptive) inquiry, especially in the context of information systems (IS) research, where information technology (IT) artifacts can be either the object of study or the result of design as a mode of scientific inquiry. As a consequence, discussion on design as a scientific methodology or research approach has been growing in recent years and is

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 93–108, 2010.

intertwined with the discussion on the nature of the IS field, placing special emphasis on striking a balance between practical relevance and academic rigor [4,19,37]. Design science (DS), as a problem-solving paradigm for information systems research, seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, management, and use of information systems can be effectively and efficiently accomplished [19]. As such, a design science contribution requires identifying a relevant organizational IT problem, demonstrating that no solution exists, developing an IT artifact that addresses this problem, rigorously evaluating the artifact, articulating the contribution to the IT knowledge-base and to practice, and explaining the implications for IT management and practice [29]. From a practical perspective as well, DS has shown increasing relevance. As technology tends to get more complex and design projects accommodate multiple stakeholders, the design projects have further potential to generate new knowledge on how technologies and theories work within different environments, while managing these kinds of efforts can also benefit from structured design approaches, e.g. [32].

The momentum of design science, even called a revolution waiting to happen [12], has been gathering for some eighty to forty years, depending on who is counting [3,9,10,28,39]. Despite the long lead time from 1969 when the first edition of Simon's "Sciences" [39] appeared, it seems that DS is off to a good start especially in the IS field. Quite recently, three separate journals from two continents, that we know of, have celebrated DS with special issues, namely MIS Quarterly in December of 2008 (Vol. 32, No. 4), and the European and Scandinavian Journals of Information Systems, respectively: EJIS in October 2008 (Vol 17, No. 5) and SJIS in late 2007 (Vol 19 No. 2). In the management field, Organization Studies has also published a special issue on DS (Vol. 29, Issue 3). Besides these journals, the International Conference on Information Systems (ICIS) runs a separate track on DSR and there is now a separate conference called "Design Science Research in Information Systems and Technology" (DESRIST).

However, there are slightly different views on what design science is, depending on the background of the authors. There have been long-standing traditions of rational, scientific, design methodology under the labels of 'design science' and 'design research' in different fields [3,9,39,42], which creates confusion about the nature of DS. This ambiguity also creates different views regarding the particular research instruments that can be used as part of design science, the underlying epistemology that should guide it, and the criteria that should be used to evaluate a design science contribution.

Our objective with this paper is to facilitate the discussion on design science and to contribute to making it more actionable for researchers. We do this through: 1) identifying the most influential literature on the subject and 2) analyzing the literature and discussing the underlying terminology and epistemological considerations. To fulfill these goals we used bibliometric analysis to uncover citation patterns and to identify the most influential pieces of literature. Subsequently, we continue to give a concise overview to the contents of the literature and outline issues for further research in design science research and practice in information systems.

The remainder of the paper is organized as follows: The second section will describe our research approach and the third section will describe the results of the literature survey. In the fourth section, we will discuss the findings from the survey, analyze the literature and highlight observations from the literature. The fifth section will summarize the paper and present our conclusions.

## 2 Methodology and Data

Bibliometric methods have been used in a wide variety of disciplines. They are seen as helpful tools to uncover conceptual structures in a possibly fragmented field [36]. We use co-citation analysis and publication statistics to examine the citation patterns in design research to identify the key pieces of literature in the field and to uncover the structures in DS. Co-citation analysis and different methods are discussed by Gmür [16], and by his terms, our approach can be characterized as document-oriented co-citation analysis.

To begin with, we made a query to the ISI Web of Knowledge [25] database to retrieve a dataset to be analyzed. The main tool enabling the analysis was Henri Schildt's SITKIS toolset [35], which is able to import and manipulate Web of Knowledge datasets and analyze the citation patterns. The dataset was retrieved 28th of January 2009 with the keywords '"design science" OR "design research" OR "design theory"' in the title or topic. Before exporting the citation data, the dataset was narrowed down by concentrating on social, information and computer sciences, excluding sociological and political studies, as well as medical and life sciences. The filtered dataset, henceforth dataset A, which acted as a starting point for our analysis, was 189 documents with over 2000 references.

Dataset A was exported from ISI and imported to SITKIS, which then compiles a database from the citation data. We examined the list of cited documents for misquoted author names or article names. Examination of the first run of results revealed a problem with dataset A, as some of the most popular sources showed up as multiple instances in the co-citation network, due to several editions of the same work. Donald Schön's "Reflective Practitioner" [38] and Herbert Simon's "Sciences of the Artificial" [39] as the most notable examples. There was little correlation between the time of citation and the edition cited, so we merged the citations on the grounds that the core of the document content has remained the same over the editions.

We ran SITKIS reports for co-citation network with a threshold of 5 citations and publication citation statistics with the same threshold. The result was a network with 45 nodes used as the main data of our analysis, which we will henceforth call dataset B to distinguish it from the initial dataset (A) that was imported into SITKIS. In plain English, SITKIS composed a matrix showing which documents were cited together and how many times. The threshold of 5 means that only documents with a minimum of five citations from dataset A were included in the matrix. The threshold is used to raise the probability that the documents have a connection by substance, as the more times the same documents are cited together, the less coincidental the citations are. Further on, we used UciNet with NETDraw [5] to compose the network diagram presented in Appendix 1.

## 3 Survey Results

Looking at the yearly citations to the ten most cited documents (Figure 1), we instantly get the sense that apparently design literature is gaining momentum, as citations to the classics have increased significantly in recent years. Although some of the more cited documents have been published in the early nineties, the total number of citation to DS literature started to climb between 2003-2005, and has been growing since.

The two main documents that have managed to stay popular for the whole period are Schön's and Simon's works. Figure 1 shows that Schön [38] has been constantly popular in absolute terms since 2003-2004 but Simon [39] and some of the later published pieces have gained citations in absolute terms.

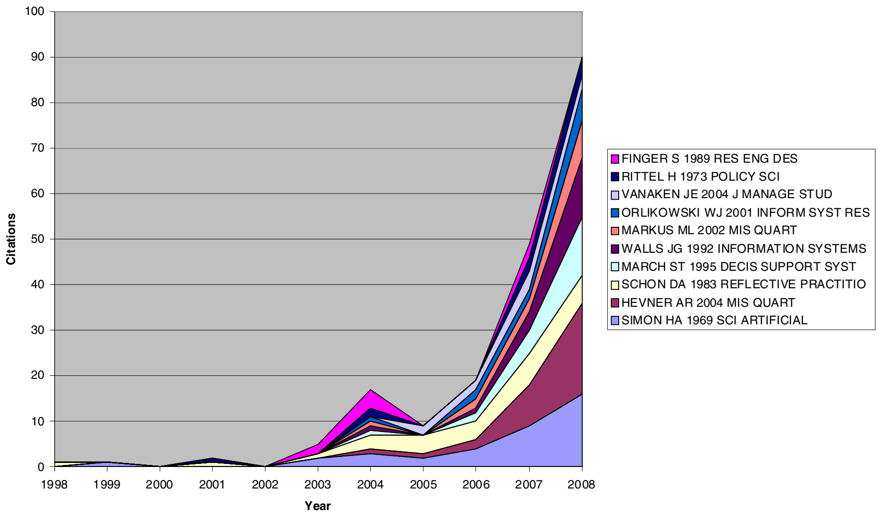



**Fig. 1.** Number of citation to the top ten documents

Looking at Figure 1, we can ask whether the growth in citations to the documents in dataset B is coincidental with the publication of Hevner et al. [19] or whether the citation pattern is an indicator of a larger movement in the field. The explanations behind the pattern may be varied, but a plausible one might be that together with their earlier paper in Computer [18], a well read journal among practitioners in Americas, the paper gives one of the first practically applicable and reasonably complete frameworks for DS. The explanation is reinforced by Indulska and Recker [24] who survey IS conference papers and found that DS paper that cite Hevner et al. had increased significantly between 2005-2007. However, we must also ask whether the amount of citations is inflated by better availability of papers through online databases or perhaps an unrelated general growth trend in the number of indexed research papers.

Beside Figure 1, Figure 2 shows the citations are distributed to different documents and disciplines, and reveals the balance within DS literature. The figure is overall largely dominated by IS researchers, and especially Hevner et al. [19] have gained citations recently. If we take out the effect of Hevner et al. [19] from Figure 2 the balance shifts somewhat, now March and Smith [28] and Walls et al. [43] assume the place behind Simon [39]. Schön's [38] setback also becomes perhaps even more apparent. Otherwise, it is interesting that during the five years time from 2003 to 2008, documents from the IS field have taken up almost 60% of citations to top 10 documents, whereas in the year 2003 60% of citations went to engineering through Finger and Dixon [13,14] and another 20% to Schön [38] alone.

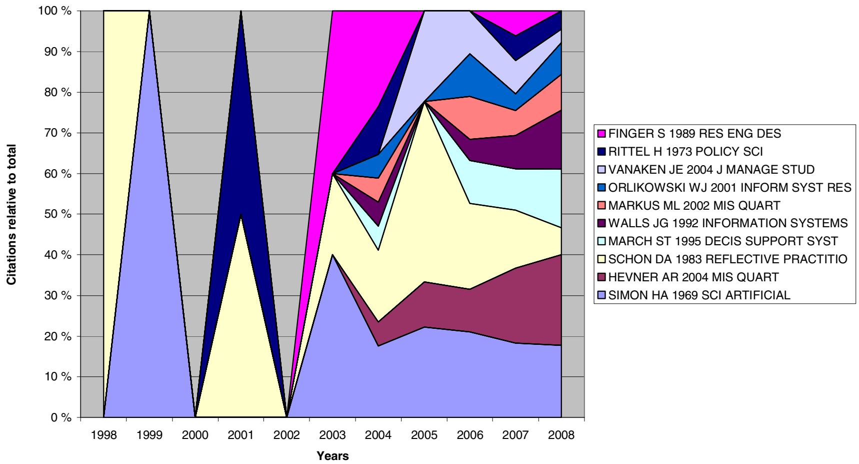



**Fig. 2.** Relative number of citations to the top ten documents

Regarding the publication statistics in general, it is notable that many of the documents in dataset B are books rather than journal articles. Dataset B contained 30 individual documents and 14 of these were published in a periodical, 13 of which in a peer reviewed academic journal. We can observe that many of the books seem to be from the engineering/design field, whereas articles are mostly published in the IS and management fields. An interpretation might be that methodology discussion is organized through journals in the IS field and through books in engineering, but we can not say so based on the data. However, this might indicate that DS is more established in engineering disciplines, but is still emerging in IS and the books are still in the making.

Besides the yearly statistics, the next point of interest is the co-citation network for dataset B (Appendix 1). The nodes in the figure represent the cited documents, and the size of the node indicates the number of citations to the document. The arcs or lines show the connections between the documents that have been cited together by the documents in dataset A. The weight of the line depends on the number of co-citations. Only connections stronger than five are shown in the plot i.e. each arc means that the nodes are cited together at least five times in different documents, due to the threshold previously determined.

The citation count suggests that the top documents capture a significant number of citations within dataset A so we may indeed call them the core documents as the same documents are also well represented in the co-citation matrix. The strongest linkages are, not surprisingly, found between the ten most cited documents. The attention turns to three main nodes also prominent in the citation statistics, Hevner et al. [19], Simon [39] and Schön [38]. There is a centre of gravity composed of a tightly connected set of documents around Hevner et al. [19], which consists of more recent documents published mainly in the IS field. Schön [38] is cited together with books from engineering and management disciplines, and Simon [39] seems to link these two traditions together. When turning to publication dates and connection strengths, the core of the literature seems to be a set of key articles published in MIS Quarterly and some other influential IS research journals.

The right side of the figure (Appendix 1) is filled mostly with articles from journals with stronger management or design orientation, and design books. At a first glance, it would seem that the common factor between IS and other disciplines in terms of design is Simon's "Sciences" [39], which is cited with IS and management articles equally. The initial expectation was that the literature would cluster roughly in three groups: theory and philosophy of design, design as a scientific methodology and domain specific and practitioner oriented (sub-) groups. Strictly speaking, the expectation seems to be supported only partly, as the methodological discussion in the IS field shows up as a tightly wound cluster, but otherwise the field is rather loosely coupled, in fact too loosely to draw meaningful cluster borders qualitatively.

To pursue the observed discipline clustering and to examine where DS literature could be positioned, we classified dataset B into disciplines in terms of book or journal title, counted the number of citations to each title, and aggregated the results into research fields, as shown in Figure 3. The classification was heuristic or judgmental and was based on the title and description of the document, we used the following codes: DES for design research in the sense Bayazit [3] uses the term, ENG for general engineering excluding design literature, IS for information systems, MGMT for management science and business administration and MISC for miscellaneous category includes research methodology books and articles, as well as references to fields which had a minor role in dataset B, such as AI and policy science.

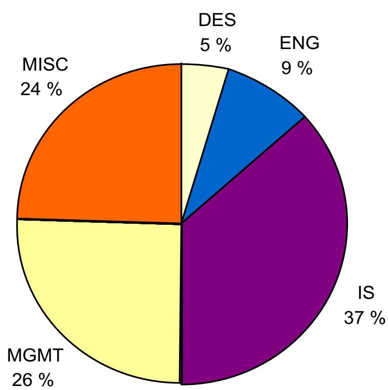



**Fig. 3.** Proportion of citations captured by each discipline in dataset B

In these results the IS field takes 37% of the total citations and management attracted a further 26%. By comparison, the engineering and design fields together attracted roughly half as many citations than the management field. The large weight of the MISC category is explained with the fact that we did not feel comfortable with placing Simon [39] into any particular field as the theme of the book is quite overarching and is cited in practically all disciplines in our classification. Another similar instance is Rittel and Webber [34] who together with Simon make up half of the citations to the MISC category.

When looking at the forums where the DS documents were published in, it would seem that the British Journal of Management and MIS Quarterly were the only ones to develop a longer discussion on the subject in terms of number of published documents; the other journals on the list have mostly published individual articles, although significant ones at that. Especially in the management side, it should be noted that some top (American) management journals, did not participate actively in the discussion on DS.

To explore the possible broadening of DS to different fields, we also classified the set of citing documents, as opposed to the classification of the dataset B presented above. Due to the threshold, the citing documents are a separate sub-set of documents within the 189 of dataset A. Now we counted the number of documents which cited dataset B and classified the titles into disciplines. In addition to the codes presented above, we used codes CS for computer science and SE for systems engineering. Here we see that the range of disciplines depicted in Figure 4 is broader than the one found in dataset B, including systems engineering and computer science. Also, the portion of documents from the engineering disciplines is considerably more pronounced than within dataset B.

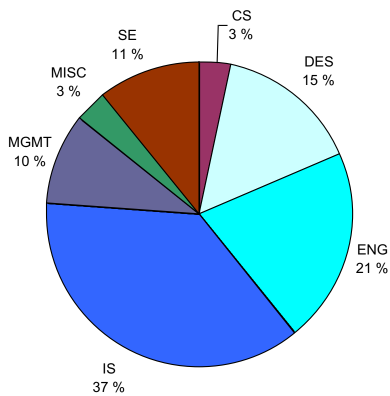



**Fig. 4.** Portion of citations to dataset B from citing disciplines

To sum up the description of the survey, the closer analysis of citing documents seems to suggest that there are four distinct disciplines interested in DS: engineering design, IS and management, and of course the design research field itself. It would seem that DS as a research approach has advanced mostly in the IS field, where it is well suited to the nature of the discipline. The design research field, as discussed below in more detail, is rooted in architectural and engineering design and is interested in rational, analytical methods of design and synthesis [3,9] whereas DS in IS field is more oriented toward research. It would also seem that DS is also acquiring more space in the management field, where the analogy with the IS discipline is often quite strong.

## 4 Overview of and Open Issues in the Design Science Literature

The following overview is primarily based on the key documents in dataset B (presented earlier). We will be concentrating on IS literature mostly, but we will discuss literature from other fields, as well as outside the dataset as appropriate. Starting from the definitions, after Simon [39] and in particular after March and Smith [28], the distinction between descriptive "natural science" research and design-oriented " design science" has been fairly unanimous. On the one hand, the interest of natural or social science-oriented research (in IS) is to describe and understand the phenomena under scrutiny. On the other hand, design science (research) deals with applying existing knowledge and technology to synthesize artificial things, artifacts, and prescriptions to solve problems.

While there is some consensus about the aim of DS, there is also some confusion over the use of the terms 'design science', 'design research' and 'design science research'. The matter is made more complex by sometimes interchangeable use of the terms DS and 'design research', which both often generically mean application of scientific knowledge to solve a problem through a research process. Cross [9] discusses the long history of design methodology movement under the labels 'scientific design' and 'design research', which, using DS wording, aim to formalize the synthesis of artifacts by means of explicit methodologies. However, there has been a similar movement under the label of 'design science' to reduce the synthesis of artifacts to a set of rules and procedures to fulfill the requirements by closed form rational reasoning [9], which can be associated with the design research and methodology movement. The discussion on terminology is partly associated with the question of where DS literature can be positioned. There is a difference between the definitions depending on the different disciplines and the author; where Winter [44] interprets Cross' [10] definition of 'science of design' or 'design science' as a more prescriptive or even normative discipline, Bayazit's [3] definition of 'design research' is more broad and descriptive, including study of design procedures and practices as well as designed artifacts including their performance, but he writes that the act of designing or synthesizing an artifact is not design research, i.e. in his view design research cannot be conducted from within the act of design.

Winter [44, p. 471] discusses the difference from an IS perspective by writing that "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 rigor.", which is to say that 'design science' gives a framework that describes how to properly execute 'design (science) research'. He [44] ties the term 'design science' to the tradition of research on formal design methodologies similar to efforts in engineering and architectural design, whereas he sees 'design (science) research' as an effort to solve problems through research within the framework set by 'design science'.

In this paper we take the position, following Winter [44] and Baskerville [2], that 'design science research' is systematic creation of knowledge about and through design, that is application of previous knowledge to a problem through a research process following the DSR(IS) framework, as well as the design-oriented tradition in management field which has largely similar intellectual roots as DSRIS (see e.g. [42]), while we give the badge 'design science' to prescriptive research on methodologies to design, to synthesize an artifact to fill a problem space through a rational problem solving process, and its evaluation within that framework in a more generic sense, and finally, following Baybazit [3], we can call descriptive research on design practices 'design research' as in 'research on design'.

A significant common strand in the literature on DS is a concern with the rigor and relevance of research and its ability to inform practice, e.g. [4,21,37,42]. Indeed, if one looks not at the cited documents (dataset B), but at the citing ones (dataset A), it seems that in some cases the references to design science are co-cited with other articles that treat the "rigor vs. relevance" debate. The underlying question is "what makes design scientific?", i.e. what is the difference between routine design or engineering application and a DS project. In different (sub-)fields of research the criteria for what constitutes a contribution can be quite different; while some fields and researchers require a proper contribution to be a refinement or falsification of an existing theory or indeed a new theory, others hold that demonstration of an artifact that enables something previously impossible or prohibitively resource intensive, is a contribution to the knowledge base in the field. For Baskerville [2], DSR, as opposed to design practice, is a systematic and organized effort to create knowledge. Hevner et al. [19] address the difference between routine design and DSR by defining design as the application of knowledge to solve a problem, while DSR adds to the existing body of (scientific) knowledge by examining uncharted non-trivial, possibly 'wicked' [34], problems and solving them in novel ways in a rigorous fashion through a systematic process. To be more specific, design science research is an activity that can be characterized as formulating valid prescriptions or 'design theories' [30,43] on how to develop methods [28] or classes of artifacts (constructs, models, methods, or instantiations) [28] to fill a certain problem space [30]. This discussion answers the previous question by outlining that the contribution of DS research is or ought to be twofold: it results in new knowledge through refinement and use of existing theories, as well as in new artifacts that enable possibilities previously unavailable to practitioners.

As for practical guidance for conducting DS research, Hevner et al. [19] describe a basic framework for DS research by explaining that IS research in general and DS research in particular should be linked to both the surrounding (business) environment and the knowledge base built by previous research. Hevner [20] later proposes that DS research is built from three related cycles of activities that aim to solve the research problem. The methodology and basic activities in DS projects have been discussed by Vaishnavi and Kuechler [40] as well as by Peffers et al. [31].

Moving from this overview towards open issues, while much of the discussion on relevance of research concerns the practical applicability of research in business, as design becomes a scientific activity the objective is not only utility but also knowledge production. Indeed, the quest for knowledge can create conflicts of interest if practitioners are not keen on investing time and effort on academic rigor, and if researchers are not so interested in the details of implementation besides focusing on delivering a 'proof of concept', rather than a production-ready solution. While this may be an overtly simplified distinction, many DS researchers will have to consider 'non invasive' research instruments and approaches, and in our experience, often data collection will have to be abandoned in favor of finishing the design within budget, suggesting the primacy of utility over rigor. Disregarding this as a simple project management issue is perhaps insufficient; in our opinion, DS is in need of a set of methods that enables research aimed at reusability and transferability of design knowledge, while economizing data gathering and analysis efforts aimed at e.g. confirmation of theory. Nonetheless, the potential conflict of interests need not develop into a conflict; for example, Holmströn et al. [22] propose that DSR projects can be organized to satisfy both stakeholder groups, filling practical needs by designing a solution and then engaging with theory development based on the created artificial phenomena.

From the perspective of knowledge creation, rigorous evaluation and reporting of the design are important steps in gaining more general knowledge in addition to solving the relevant original problem [19]. And as Winter [44] points out, to rival descriptive "social science" or "behavioural science" research in rigor, that is to say also in academic credibility, DS needs an established and transparent methodological framework and quality criteria. Starting from the most basic level, the interface between the designers' worldviews and the choice of methods in DS appears lacking. It is often easy to lose sight of why we should, in Dobson's [11] words, "bother with" philosophy. Dobson argues pointedly that in practice the opposite of philosophy is not the absence of philosophical thought, but "bad philosophy" instead. Dobson goes on to explain that unless the epistemological foundation is critically examined and purposefully built, it tends to be vague and shaky. Due to the nature and basic structure of scientific arguments, this shakiness is then transferred to the results and conclusions.

Philosophical discussion raises another important issue; the relationship between evaluation and validation of artifacts. Hevner et al. [19], for instance, raise the issue of artifact evaluation, whereas discussion on validation and the associated epistemological underpinnings are not explicitly addressed. To clarify, we understand that evaluation is concerned with utility of the artifact: it deals with requirements, performance and functionality, whereas validity is about truthfulness of claims and their reliability and robustness. The ensuing question "where is the truth in design" is intimately associated with this discussion. Iivari [23] writes that an artifact does not have truth value, i.e. an artifact as such is not true or untrue; it does however have interfaces and functionalities, which are evaluated against the specification to deem whether the goals are met. Truthfulness then applies to the conceptual background of the artifact: the kernel and design theories. According to the basic prescription [43], the DS artifacts are based on a kernel theory that should in effect predict the mechanism that transforms the input into the artifact as well as its effect in an organizational context. Validation should provide evidence of whether the artifact represents the theory sufficiently to give way to theoretical insights. As such, validation enables critical evaluation of the theory and additions to the existing theoretical knowledge. If we assume that DSR is supposed to be both useful and informative, in effect evaluation is concerned mostly with the first dimension of DS contribution, the utility, and validation then is concerned about the second dimension of contribution to the existing scientific knowledge, or more precisely the truthfulness, reproducibility, and predictability of that contribution. Following Hevner et al. [18, 19], DSR can contribute to the knowledge base through design of artifacts, but in terms of contribution to other foundations for DSR validation becomes a larger question. The issue of reproducibility has been explored more elaborately, for instance, in the use of Design Patterns (e.g. [1,15]) aimed to produce generalized reusable solutions to recurring problems, while issues of truthfulness pertain more to methodology and research philosophy.

Discussion on validation as defined above brings us back to the subject of research philosophy and specifically the domain of epistemological questions, namely "how do we get knowledge of the world?" and more specifically "how do we know this or that to be true?". The situation is made more complex by the fact that DS literature is not unanimous on the philosophical issues, as illustrated by the discussion on the Scandinavian Journal of Information Systems e.g. [23] and also in the discussion on the similarities of action research and design research [8,26]. Hevner and others [19], as an example, do not take part in the epistemological discussion explicitly, only by writing: "The goal of behavioural-science research is truth. The goal of design-science research is utility. As argued above, our position is that truth and utility are inseparable. Truth informs design and utility informs theory." Vaishnavi and Kuechler [40], on one hand, state that instrumentalism, or pragmatism, is built into DS. While on the other, if we take Iivari [23], he lays quite a different, more positivistic, foundation for his DS approach. The presence of Bunge [6] is also an interesting finding in dataset B, as in his later works on philosophy of technology [7] he exhibits claims similar to Iivari [23]. In effect, Bunge, together with Iivari, differ from both the instrumentalism of e.g. [40], as well as from the well accepted interpretivist movement in the IS field represented by e.g. Klein [27].

Beside research philosophy, the literature considered here seems to leave room beside the existing efforts [31,41] for more explicit discussion on the specific methodologies which can be employed inside this framework to achieve rigor. While the theoretical basis for an research approach are important for academic credentials and obviously for validity, DS literature could benefit from more explicit discussion about good practices and conduct; what methods and tools could or should be used in the process?

## 5 Discussion and Conclusion

We embarked on this journey with an objective to identify the key literature on DS and analyze the open issues for further research. Our objective was to clarify issues concerning the nature of design science, more specifically the foundations and the position of DS. Our approach to finding the answer was a literature survey and bibliometric analysis of the literature. The first contribution of this paper to the existing knowledge is identification and description of the core of DS literature. The second contribution is then overview of this significant and much cited literature and discussion of some of some open issues to fuel further research, which we summarize hereafter together with our other findings.

As we can gather from the literature, the general agreement in the IS field is that DS aims to contribute both to solving relevant problems and to adding to the existing body of knowledge, by examining uncharted problems in a real-world environment and solving them in novel ways in a rigorous fashion through the design of artifacts. However, we came to found that due to different backgrounds the literature as a whole was not unanimous in respect to terminology and definitions, which can, nevertheless, be reconciled relatively easily.

An interesting finding from the survey is that the citations to DS literature have been on steep rise and the patterns seem to indicate that much of this expansion of literature is concentrated on the IS and management fields. We found that the most cited core of the DS literature is spread into four main disciplines: IS, management, engineering and design research. An analysis of the citing documents reveals a similar pattern, although the disciplines that cite DS literature include also systems engineering and computer science.

While it seems that the literature on DS is quite strong and growing, the literature is not explicit on how the contribution to the knowledge base is supposed to be made and how it relates to the designed artifact. In this respect the framework and the foundations of DS as a program are still in the making. Particularly the linkage between the underlying ontological and epistemological assumptions and methodological issues, such as validation of knowledge contributions is an important gap especially considering the multi-paradigmatic nature of the IS field, e.g. [33]. A more practical, but equally important, issue is the ongoing work to compose explicit guidelines for research design and methods to build, evaluate, and validate artifacts where Vaishnavi and Kuechler [40, 41] as well as Peffers et al. [31] have already proceeded.

These topics are also intertwined with the discussion on the nature and underlying paradigms of IS as well as management fields. The question that comes to the mind easily when discussing research philosophy is "what does it matter?" or "what is the point?". The short answer is that rigorously documenting the philosophical assumptions and methodological choices should answer or at least pave the way for answering questions about validation and clarify how to do DS research at the project level given the chosen research philosophy. The longer answer is twofold: Firstly, a strong philosophical base gives strength to the arguments based on DS framework, and established methodological guidelines enable more transparent review of the research and its results, all to the end of stronger contribution to the knowledge base and to the practice. Secondly, while focusing only on the utility dimension through evaluation, DS effectively envelops much of present design and engineering oriented research where practical problems are solved by using existing knowledge. It can be hypothesized that this orientation adds to the rise in citation the DS literature has received as observed in the citation figures. However, leaving the other dimension, validity and truthfulness, to lesser attention might be detrimental to DS in the long time frame if it means that DS will not be able to stand beside behavioral science research in terms of scientific standards, or as Winter [44] reminds, there is still work to be done is DS wants to gain the same status as behavioral or natural science research has gained in the IS field.

Regarding the validity of our conclusions, we must concede that they stand or fall with the data. Assuming the data is not skewed, we should have a balanced view of DS for the parts that are visible in ISI. The validity of our discussion is a harder matter to evaluate. However, we find our discussion to be consistent with the literature in the field, and also consistent internally, we are confident that our findings can stand a critical examination. Our examination of the nature of DS covers mostly the IS field, but should be applicable with reservations to the management field as well.

The question "how can these results be applied?" is inevitably raised at this point of the paper. Our broad objective was to facilitate discussion on DS through description and discussion of the literature, and to that end the survey should serve to focus the attention in the field to the core DS literature. Our paper is not the first review paper on DS, but our contribution to the existing literature, e.g. [24,33], is the systematic approach to finding the documents for analysis through co-citation analysis. The survey serves as a vehicle for entering the discussion around DS and provides a base for further analysis. As a second or third more concrete contribution we have raised some topics we found interesting or surprising in the examined documents as discussed here.

Winter [44] also referred to a long standing design-oriented research tradition in Europe. A more complete framework for DS could also facilitate exploring these earlier contributions and extracting the lessons learned from this previous and supposedly valuable research. Similarly there is also a long tradition of design literature and theory as pointed out by Bayzit [3] and Cross [9], which could benefit DS were it included. We observed that, even though connected conceptually, the design science literature and emerging tradition in information and management science has developed different interests, and in the process have lost connection with design research, as suggested by the lack of traditional design journals in dataset B. Bridging these related traditions holds potential for greater transparency and understanding of previous contributions, while the traditions could also teach something to each other in terms of practices as well.

After these remarks we would like to conclude by creatively quoting Schneberger et al. [37]: DS can mould theories-in-use or theories with little t from grander Theories with a capital T, as theories for explaining and predicting [17] might be labeled. As such DS framework is in a position to bring valuable contributions to our collective knowledge as researchers, while at the same time is can also act an escalator between the abstract world of science and the practical world of business to the benefit of both worlds of course.

**Notes.** Both of the datasets are available from the authors at request.

**Acknowledgements.** This research has been conducted partly under a grant from the Academy of Finland.

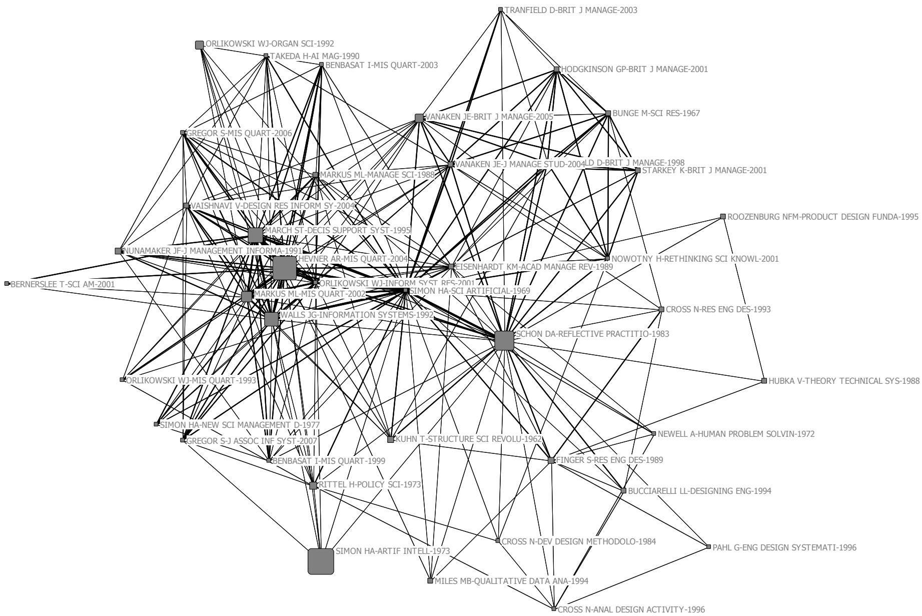
## References

1. Alexander, C.: The Timeless Way of Building. Oxford University Press, New York (1979)
2. Baskerville, R.: What design science is not. European J. of Inf. Syst. 17(5), 441–443 (2008)
3. Bayazit, N.: Investigating Design: A Review of Forty Years of Design Research. Design Issues 20(1), 16–29 (2004)
4. Benbasat, I., Zmud, R.W.: Empirical Research in Information Systems: The Practice of Relevance. MIS Quart. 23(1), 3–16 (1999)
5. Borgatti, S.P., Everett, M.G., Freeman, L.C.: UCINET for Windows: Software for Social Network Analysis. Analytic Technologies, Harvard (2002)
6. Bunge, M.: Scientific Research. Springer, Berlin (1967)
7. Bunge, M.: Philosophical Inputs and Outputs of Technology. In: Scharff, R., Dusek, V. (eds.) Philosophy of Technology: The Technological Condition - An Anthology. Wiley-Blackwell, Malden (2003)
8. Cole, R., Purao, S., Rossi, M., Sein, M.: Being Proactive: Where Action Research Meets Design Research. In: 26th International Conference on Information Systems, Las Vegas NV (2005)
9. Cross, N.: Science and design methodology: A review. Res. in Eng. Des. 5(2), 63–69 (1993)

10. Cross, N.: Designerly Ways of Knowing: Design Discipline Versus Design Science. Des. Iss. 17(3), 49–55 (2001)
11. Dobson, P.J.: Critical realism and information systems research: why bother with philosophy? Inf. Res. 7(2), Paper 124 (2002)
12. Dorst, K.: Design research: a revolution-waiting-to-happen. Des. Stud. 29(1), 4–11 (2008)
13. Finger, S., Dixon, J.R.: A review of research in mechanical engineering design. Part I: Descriptive, prescriptive, and computer-based models of design processes. Res. in Eng. Des. 1(1), 51–67 (1989)
14. Finger, S., Dixon, J.R.: A review of research in mechanical engineering design. Part II: Representations, analysis, and design for the life cycle. Res. in Eng. Des. 1(2), 121–137 (1989)
15. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Elements of Reusable Object-Oriented Software. Addison-Wesley Publishing Company, Reading (1995)
16. Gmür, M.: Co-citation analysis and the search for invisible colleges: A methodological evaluation. Scientometrics 57(1), 27–57 (2003)
17. Gregor, S.: The Nature of Theory in Information Systems. MIS Quart. 30(3), 611–642 (2006)
18. Hevner, A.R., March, S.T.: The Information Systems Research Cycle. Computer 36(11), 111–113 (2003)
19. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quart. 28(1), 75–105 (2004)
20. Hevner, A.R.: A Three Cycle View of Design Science Research. Scand. J. of Inf. Syst. 19(2), 39–64 (2007)
21. Hodgkinson, G.P., Herriot, P., Anderson, N.: Re-aligning the Stakeholders in Management Research: Lessons from Industrial, Work and Organizational Psychology. Brit. J. of Man. 12, 41–48 (2001)
22. Holmström, J., Ketokivi, M.: Bridging Practice and Theory: A Design Science Approach. Decision Sci. 40(1), 65–87 (2009)
23. Iivari, J.: A Paradigmatic Analysis of Information Systems as a Design Science. Scand. J. of Inf. Syst. 19(2), 39–64 (2007)
24. Indulska, M., Recker, J.C.: Design Science in IS Research: A Literature Analysis. In: The Proceedings of 4th Biennial ANU Workshop on Information Systems Foundations, Canberra, AU (2008)
25. ISI Web of Knowledge, `http://www.isiknowledge.com`
26. Järvinen, P.: Action Research is Similar to Design Science. Qual. and Quant. 41, 37–54 (2007)
27. Klein, H.K.: Seeking the new and the critical in critical realism: déjà vu? Inf. and Org. 14(2), 123–144 (2004)
28. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Syst. 15(4), 251–266 (1995)
29. March, S.T., Storey, V.C.: Design Science in the Information Systems Discipline: An Introduction to the Special Issue on Design Science Research. MIS Quart. 32(4), 725–730 (2008)
30. Markus, M.L., Majchrzak, A., Gasser, L.: A Design Theory for Systems That Support Emergent Knowledge Processes. MIS Quart. 26(3), 179–212 (2002)
31. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. J. of Man. Inf. Syst. 24(3), 45–77 (2007)

32. Piirainen, K., Kolfschoten, G., Lukosch, S.: Unraveling Challenges in Collaborative Design: A Literature Study. In: Carriço, L., Baloian, N., Fonseca, B. (eds.) CRIWG 2009. LNCS, vol. 5784, pp. 247–261. Springer, Heidelberg (2009)
33. Purao, S., Baldwin, C., Hevner, A., Storey, V.C., Pries-Heje, J., Smith, B., Zhu, Y.: The Science of Design: Observations on an Emerging Field. Com. of the Assoc. for Inf. Syst. 23, 523–546 (2008)
34. Rittel, H.W.J., Webber, M.M.: Dilemmas in a general theory of planning. Policy Sci. 4(2), 155–169 (1973)
35. Schildt, H.A.: SITKIS: Software for Bibliometric Data Management and Analysis v2.0. Helsinki University of Technology, Institute of Strategy and International Business, Helsinki FI (2002), http://www.hut.fi/~hschildt/sitkis
36. Schildt, H.A., Mattsson, J.T.: A dense network sub-grouping algorithm for co-citation analysis and its implementation in the software tool Sitkis. Scientometrics 67(1), 143–163 (2006)
37. Schneberger, S., Pollard, C., Watson, H.: Theories: For Academics and Practitioners. Inf. Syst. Man. 26(1), 52–60 (2009)
38. Schön, D.A.: The reflective practitioner - how professionals think in action. Temple Smith, London (1983)
39. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
40. Vaishnavi, V., Kuechler, W.: Design Research in Information Systems (2004), http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=279 (Retrieved September 2, 2009)
41. Vaishnavi, V.K., Kuechler Jr., W. (eds.): Design Science Research Methods and Patterns: Innovating Information and Communication Technology. Auerbach Publications, Boca Raton (2007)
42. van Aken, J.E.: Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management. Brit. J. of Man. 16(1), 19–36 (2005)
43. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Inf. Syst. Res. 3(1), 36–59 (1992)
44. Winter, R.: Design science research in Europe. Eur. J. of Inf. Syst. 17(5), 470–475 (2008)

# Appendix 1: Co-Citation Network for Dataset A



# Design Science Research Post Hevner et al.: Criteria, Standards, Guidelines, and Expectations

John R. Venable

School of Information Systems, Curtin University, G.P.O. Box U1987, Perth WA 6845, Australia
j.venable@curtin.edu.au

**Abstract.** There is ongoing debate about how the quality (rigour and relevance) of Design Science Research (DSR) should be judged. This research investigates the state of the debate by surveying the opinions of IS scholars who write, review, edit, and publish DSR papers. The survey respondents rated the relative importance of the seven guidelines (often used as evaluation criteria) laid out in Hevner et al. (2004) [6], more specific criteria about the evaluation activity in DSR, criteria concerning IS Design Theories, and miscellaneous other criteria, and made general open-ended comments. The findings indicate a lack of consensus, with much variability in ratings. The Hevner et al. [6] guidelines are largely endorsed, but caution is also raised to apply them less mechanistically than at present. Some criteria/guidelines are seen to be less important at earlier stages of research. Caution is also urged not to expect single papers to fit all criteria/guidelines.

**Keywords:** Design Science Research, Research Method, Research Standards, Evaluation, IS Design Theory.

## 1 Introduction

It has now been more than five years since the publication of Hevner, March, Park, and Ram in MIS Quarterly [6], which set the de facto standard for the conduct and evaluation of Design Science Research (DSR). Since then, there is healthy debate about how the quality (rigour and relevance) of DSR should be judged and what quality goals DSR should attempt to achieve. (e.g. [1, 12, 2, 3, 15, 16], panel discussions at DESRIST 2007, ECIS 2007 and ACIS 2007, and an entire conference (Information Systems Foundations: Answering the Unanswered Questions about Design Research, 2008). Still, in the author's experience from continuing conversations on these topics, open questions remain concerning the practicality and appropriate application of the Hevner et al. [6] guidelines, the role and form of IS Design Theories [18, 5], what (if any) more detailed standards or guidance concerning the evaluation activity in DSR (Hevner et al. 2004 Guideline #3) are appropriate, and whether various other goals and criteria should also be applied when evaluating DSR.

The objective of the research reported in the current paper is to develop a better understanding of the current state of the IS field's dialog and argument about appropriate criteria, standards, guidelines, and expectations for DSR. It seeks to answer the following research questions:

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 109–123, 2010.

- What is the level of consensus within the IS DSR field concerning criteria, standards, guidelines, and expectations for how IS DSR should be conducted and reported?
- What other views are there besides the received view of Hevner et al. [6] concerning how DSR should be conducted and reported?

This research investigates the opinions of members of the scholars who are engaged in that dialog through the processes of writing, reviewing, editing, and publishing DSR papers using a survey. The survey asks questions concerning the relative importance of the seven guidelines laid out in Hevner et al. [6], more specific criteria for the evaluation activity in DSR, criteria for the development and reporting of IS Design Theories [18, 5] in DSR, and miscellaneous other criteria concerning the process and output of DSR.

The paper is organized as follows. First, literature relating to criteria, standards, and expectations in IS DSR is briefly reviewed. In section 3, the survey method and sample are described. Following that, the results of the quantitative data analysis are described, as well as reporting on other issues in DSR raised by the survey respondents. Finally, the paper identifies key issues remaining to be resolved and makes recommendations for further research.

## 2 Literature Review

Despite the relative new recognition of its status in the IS field, there is an extensive literature both before and after Hevner et al. [6] about IS DSR, much of which is relevant to the concerns of criteria, standards, and expectations for IS DSR.

### 2.1 Pre Hevner et al. (2004)

While Hevner et al. [6] set the de facto benchmark for DSR, important earlier work helped set the stage. Aspects of criteria, guidelines, and areas of expectation are highlighted briefly here.

Nunamaker, et al. [11] wrote a seminal paper in design science, which focused on justifying system development (a subset of design science) as an IS research method. Their "Multimethodological Approach to IS Research" (figure 2, p. 94) included a theory building activity, which addressed "development of new ideas and concepts, and construction of conceptual frameworks, new methods, or models" (p. 94) as well as theories. Their five stage process (with backtracking) included "construct a conceptual framework" as a first step in which the researcher should "(a) declare the 'truth'", "(b) formulate a concept (i.e. a framework)", "(c) construct a method", and "(d) develop a theory" (p. 99) and an evaluation step in which one should "Develop new theories/models based on the observation and experimentation of the system's usage" (p. 98).

Walls et al. [18] identified theory as a desirable output of Design Science Research. They proposed that an Information Systems Design Theory (ISDT) should "be a prescriptive theory which integrates normative and descriptive theories into design paths intended to produce more effective information systems." (p. 36). They proposed that an ISDT would have seven components. Four components concern the design product, including (1) meta-requirements, (2) meta-design, (3) kernel theories, and (4) testable design product hypotheses. Three components concern the design process, including (5) design method, (6) kernel theories, and (7) testable design process hypotheses.

March and Smith [9] made two main contributions. First, they identified two main DSR processes: build and evaluate. Second, they identified four kinds of design artifacts (outputs of DSR): constructs, models, methods, and instantiations. In contrast to Nunamaker et al. [11] and Walls et al. [18], March and Smith [9] implicitly rejected the idea of design theories. “Rather than producing general theoretical knowledge, design scientists produce and apply knowledge of tasks or situations in order to create effective artifacts.” (p. 253)

Venable and Travis [17] built on the work of Nunamaker et al. [11], emphasising the key role of theory building, extending Nunamaker et al.’s notion of a Computer-Based Information System as the designed artifact to include IS Development Methods, Tools, and Techniques as relevant designed artifacts , and refining the Multimethodological IS Research Framework [11] to substitute “In Situ Investigation” for “observation” and include Action Research as a method for In Situ Investigation.

Markus et al. [10] used the ideas of the ISDT of Walls et al. [18] but developed more “layman’s” terminology for the concepts of meta-requirements and meta-design.

Rossi and Sein ([13] in acknowledged collaboration with Purao) added “better theories” (p. 5) to the four design artifacts in March and Smith [9] and “theorise” (p. 6) as a step in Design (Science) Research. During evaluation, Rossi and Sein (with Purao, [13]) propose both internal and external criteria. Among the internal criteria is “Match between the artifact and the ‘abstract idea’. How well does the artifact embody the abstract idea that is being researched?” (p. 8). This concerns the constructs or concepts and how faithfully they are implemented in the models, methods, and instantiations. Among the external criteria is “Advancement of design theory: Is the abstracted idea generalisable to other contexts or at least advance our understanding of other design contexts?” (p. 9). This proposes that goals for good design theory would include generalisability of the artefacts and the utility of the design artefacts in other problem contexts.

## 2.2 Hevner, March, Park, and Ram (Hevner et al. 2004)

As noted in the introduction, the paper by Hevner, March, Park, and Ram [6] represents the received view of the IS field’s conception of DSR. The publication has become the most-cited work in *MIS Quarterly*, the highest rated journal in the IS field, and is very influential. The authors draw on the earlier work of March and Smith [9] and others to develop an overall IS Research Framework as well as guidelines for the conduct and reporting of Design Science research.

Among other things, Hevner et al. [6] proposed seven guidelines for Design Science in IS Research, which can be summarised as follows:

1. Design as an Artifact – An identifiable and viable design artifact, as in March and Smith [9], must be produced.
2. Problem Relevance – The design must address a relevant and important problem.
3. Design Evaluation – The utility, quality, and efficacy of the design artifact must be rigourously evaluated.

4. Research Contributions – The contribution must be clear and verifiable. Contributions are seen to arise out of the novelty, generality, and significance of the designed artifact. Contributions include the design artifacts themselves, new foundations (constructs, models, methods, and instantiations), and new [evaluation] methodologies.
5. Research Rigour – Research methods must be rigorously applied.
6. Design as a Search Process – Research must be conducted with knowledge of other, competing approaches and should approach the process as a cyclical problem solving process, in which solutions are tested against each other and against their efficacy for solving the full problem.
7. Communication of the Research – Presentation of results needs to address both the rigour requirements of the academic audience and the relevance requirements of the professional (e.g. managerial) audience.

Interestingly, Hevner et al. [6] do not mention new or revised theory as a design artifact or research contribution (even though they reference Markus et al. [10]); it does not have a place in their guidelines.

Hevner et al. [6] emphasise Guideline 3, noting that "evaluation is a crucial component of the evaluation process" (p. 85). They further note that the evaluation method must be matched to the artifact and any evaluation metrics.

An important point about the Hevner et al. [6] guidelines is that the authors stated "Following Klein and Myers [8] we advise against mandatory or rote use of the guidelines." (p. 82).

### 2.3 Post Hevner et al. (2004)

Developed and written practically simultaneously with Hevner et at [6], the Design Research web pages [14], a portion of the AISWorld website, review much of the pre Hevner et al. [6] research and provide their own perspective on DSR. The website implicitly recognises a role for theory and theorising in DSR, identifying "better theory" as an important higher level of abstraction. However, it does not pick up the idea of an ISDT [18, 10].

Venable [15] made a case for the need to develop ISDTs, which he called "Utility Theories". He asserted that precisely formulated theories are a key vehicle for communication between scholars. He also asserted that a viable design theory does not really need kernel theories, testable hypotheses, of a design method [18]. Rather, he asserted that the essential part of an ISDT is simply a theory that some meta-design has utility for addressing some meta-requirement(s).

Expanding on Nunamaker et al. [11] and Venable and Travis [17], Venable [16] further developed the concepts of Evaluation in DSR, identifying two main classes of evaluation: Artificial and Naturalistic. He identifies naturalistic evaluation as being very important – "the proof of the pudding".

Gregor [4] developed a taxonomy of five theory types in IS. Type V, theory for design and action, is roughly synonymous with ISDT as in Walls et al. [18].

Gregor and Jones [5] expanded on the idea of theory for design and action, extending and refining the view of an ISDT in Walls et al. [18]. Their proposal for the anatomy of a design theory includes six essential (core) components, including purpose and scope, constructs, principles of form and function, artefact mutability, testable propositions, and justificatory knowledge. It also includes two optional components, including principles of implementation and an expository. This structure is largely compatible with that of Walls et al. [18].

Baskerville et al. [2, 3] also further develop the idea of evaluation, noting difficulties with snapshot approaches to evaluation and simplistic understandings of organizational situations and events during evaluation. As a solution, they propose a "soft" design science which applies more intensive interpretive methods to evaluation.

In addition to work exploring DSR activities, deliverables, and guidelines, some recent work has specifically looked at the application of the guidelines from Hevner et al. [6].

Arnott and Pervan [1] examined how well Decisions Support Systems (DSS) publications prior to 2004 that used DSR approach fulfilled the Hevner et al. [6] guidelines. They assessed that evaluation was the biggest weakness, asserting that "Some form of convincing evaluation should [be] mandatory for design-science research." (n.p.) and noted that qualitative methods should be considered and used more often in evaluation. They also found that, as a practical matter, it was very difficult to assess whether guideline 6 (Design as a Search Process) had been addressed, unless the paper made specific efforts to explicate how they had done so. Finally, Arnott and Pervan [1] also found that the "level and quantity of theorizing" in DSS DSR papers "needs significant improvement" (n.p.). They suggested that guideline 4 (Research Contributions) could be broadened to include explicit contributions to theory.

Indulska and Recker [7] examined the extent to which DSR papers in the major IS conference literature since 2004 addressed the Hevner et al. [6] guidelines. They found that 36.8% of the DSR papers analysed merely stated that they followed DSR guidelines, 22.8% focused on one guideline, 7.0% focused on some, but not all guidelines, and 19.3% elaborated on the research's implementation of all guidelines. Remarkably, only 14.0% of the papers did *not* mention or explicitly demonstrate the use of the guidelines, demonstrating a strong expectation that they be followed.

### 2.4 Literature Summary

The views espoused in Hevner et al. [6] have come to dominate the criteria, standards, guidelines, and expectations for how DSR should be conducted and written about. There are, however, other voices and views, some of which are reviewed above. In particular, the issues of the amount and means for evaluation and the need for theory as an output are seen by some to be key open issues.

## 3 Research Methodology

To address the issues raised above and answer the research questions given in the introduction, a survey of DSR participants was designed and distributed and the responses received and analysed.

The survey respondents were selected to represent three classes of participant roles, (1) the gatekeepers to high quality journals, (2) the gatekeepers to DSR conferences, and (3) DSR authors. For (1), the editors-in-chief, senior editors, and associate editors of the IS scholars' basket of eight journals (MIS Quarterly, IS Research, the European

Journal of IS, The Information Systems Journal, the Journal of the AIS, the Journal of MIS, the Journal of Strategic IS, and the Journal of IT) were chosen. For (2), the program chairs and program committee members of the DESRIST conferences (2006-2010) were chosen. For (3), the authors of papers published at DESRIST in 2006-2009 were chosen. While not perfectly representative of all IS DSR participants, the three groups seemed to be a suitable theoretical sampling method. In all, 338 journal editors, 10 DESRIST chairs, 79 DESRIST program committee members, and 242 DESRIST authors were in the sample. Due to overlaps in these categories, a total of 595 people were in the sample.

The survey itself comprised two sections. The first section collected some further demographic information, although much was already available as the survey was not done anonymously. Key questions included whether the respondent had DSR responsibility for a journal (not necessarily in the basket of eight) and whether any papers authored or co-authored by the respondent (not necessarily published at DESRIST) were either *about* DSR or *using* DSR.

The second section asked respondents to rate the importance of various guidelines or criteria on a 0-10 scale. The 0-10 scale was used because it would be clear that an interval scale was being used so that averages would be meaningful, a 0-10 scale is commonly used in day-to-day human activity, it allows fine levels of gradation if desired, and it is easy to remember. The specific instructions to the respondents were:

> "Please rate your perception of the relative importance of the following existing or potential areas of practice and standards for Design Science Research. Please rate them from 0 to 10, where 0 indicates the practice or standard area is of no importance and 10 indicates that it is mandatory that the practice or standard must be met to consider publication of the Design Science Research result."

The response items in section two were divided into four sub-sections. Section A was concerned with the Hevner et al. [6] guidelines, Section B with evaluation methods and aspects to be evaluated, Section C with IS Design Theories and Section D with miscellaneous items. The specific text of each item is shown with the research findings further below. All sections also included areas for open comments.

An email address list of all potential respondents was developed from the conference papers, DESRIST websites and calls for papers, journal websites, searches of the AIS faculty directory, and general searches on the web. The survey was emailed to all recipients in mid January 2010 and a follow up was made about a week later. Emails that bounced were further researched and the survey re-sent, in some cases successfully. A second follow-up was made to non-respondents at the beginning of March 2010, which yielded about 50% more respondents.

In addition to completed surveys, the author received replies indicating that the respondent had insufficient knowledge of DSR to meaningfully answer the survey or did not review or handle DSR papers. In response to that, survey non-respondents were asked on follow-up to please notify the author if they felt that they were unqualified to answer the survey and/or did not have DSR journal responsibilities – without answering the remainder of the survey. Other respondents gave other reasons for not answering the survey, as described in the next section.

# 4 Survey Results

## 4.1 Demographic Results

Of the original 595 intended survey recipients, 34 recipients could not be contacted or were ineligible (no longer journal editors or the author of this paper), reducing the sample population to 561.

Of the 561 eligible survey recipients, 234 or 41.71% responded. However, of those, 125 (53.42% of respondents) indicated that they did not feel qualified to answer the survey. Five (2.14%) indicated that they had philosophical differences with the survey or DSR that precluded them answering the survey. Eleven (4.70%) indicated that they didn't have time or didn't do surveys. In total, 141 (60.26%) of the respondents did not answer the survey, which leaves 420 (561-141) surveyed for response rate purposes.

Of the 93 respondents who answered the survey (a response rate of 22.14% of the 420), 8 (1.90% of 420) did not answer all the quantitative questions and thus should not be included in the calculations, as all item ratings are relative to all other ratings.

Concerning the overall valid response rate, in all, 85 valid survey responses were received, yielding a response rate of 20.24% of the 420 eligible survey recipients (who had *not* indicated that they felt insufficiently knowledgeable to answer the survey, had incompatible philosophical positions or didn't have enough time to answer the survey). The author deems this response rate to be sufficient.

Of the 85 valid survey responses, 40.00% had a senior scholars' basket of 8 journal editor role, 31.76% served as a DESRIST PC chair or member, and 61.18% were DESRIST authors. Based on the survey responses, 37.65% indicated that they had DSR responsibility at a journal (not necessarily for a basket of eight journal), 54.12% had authored or co-authored a paper *about* DSR, and 74.12% had authored or co-authored a paper *using* DSR (not necessarily at DESRIST). Many respondents also supplied answers to the open comment questions, totaling about 15 pages of text.

## 4.2 Detailed Results

Tables 1 through 4 below show the detailed results of the ratings of the quantitative survey items. Higher ratings indicate higher importance, with 10 being mandatory or essential and 0 being completely unimportant or irrelevant.

As can be seen in Table 1 (next page), fulfilling *all* of the Hevner et al. [6] criteria was rated on average as only of medium importance, while specific guidelines were rated higher. Guidelines 1, 2, 3, and 4 were rated as very important, with guidelines 5, 7, and especially 6 being rated less so. Of the four individual types of artifacts, instantiations were rated as most important and models rated as least important. The standard deviation and minimum and maximum ratings also provide important information. All areas relating to the guidelines were rated by at least one respondent at 10, being essential or mandatory for publication, while guideline 2 had the highest minimum (five). Items with minima of 0 and maxima of 10 indicate a very high level of disagreement among at least some respondents. Standard deviations are fairly high for all guideline ratings, also indicating high levels of disagreement.

The open ended comments also provided much information and enlightenment. Several respondents commented that the guidelines are "too mechanistic", "a bit too dogmatic", "a cookbook recipe" or "too simplistic to apply a guidelines checklist" and objected to their use as "mindless checklists". One even claimed that "Even Hevner et al. do not regard their guidelines as 'guidelines' anymore – rather as an evaluation instrument'. Despite the concerns raised, many respondents (often the same ones who raised concerns) noted that the guidelines were important and useful.

Many also noted that the importance of any one criterion or guideline is context dependent, e.g. depending on the kind of artifact developed, the stage of the research, or "the state of the art for the particular research area of the paper".

**Table 1.** Part A: Hevner et al. [6] Guidelines

| Survey Item | Mean | Std Dev | Max | Min |
|---|---|---|---|---|
| Addressing ***all*** of the guidelines given in Hevner et al. [6] (described below) | 5.36 | 3.00 | 10 | 0 |
| Presenting an identifiable and viable design artifact (concept, model, method, or instantiation) as in March and Smith [9] (Guideline 1: Hevner et al., 2004) | 8.39 | 1.78 | 10 | 3 |
| Presenting one or more clearly defined new concepts [9] | 6.52 | 2.47 | 10 | 0 |
| Presenting one or more clearly explained new models [9] | 6.16 | 2.41 | 10 | 0 |
| Presenting one or more clearly explained new methods for building the artifact [9] | 6.29 | 2.44 | 10 | 0 |
| Presenting one or more example instantiations of the artifact [9] | 7.30 | 2.23 | 10 | 0 |
| Addressing a relevant and important problem (Guideline 2: Hevner et al., 2004) | 9.05 | 1.21 | 10 | 5 |
| Evaluating the utility, quality, and efficacy of the designed artifact (Guideline 3: Hevner et al., 2004) | 8.31 | 1.61 | 10 | 2 |
| Clearly identifying the novelty, generality, and significance of the contribution (Guideline 4: Hevner et al., 2004) | 8.45 | 1.74 | 10 | 2 |
| Rigourous application of the research methods (Guideline 5: Hevner et al., 2004) | 7.33 | 1.90 | 10 | 0 |
| Developing the design using a cyclical, problem solving search process (Guideline 6: Hevner et al., 2004) | 6.09 | 2.46 | 10 | 0 |
| Presenting the research to address both rigour for the academic audience and relevance for the professional audience (Guideline 7: Hevner et al., 2004) | 7.20 | 2.07 | 10 | 1 |

Others commented on the unfulfillability of the criteria, especially in a single research paper or thesis. As one respondent stated,

> "The general problem I experienced when writing a DSR-paper is that it is practically impossible to address all the guidelines within one paper. If the artefact addresses a relevant problem for practice then normally the build/evaluate cycle is rather complex. But papers are limited to a certain amount of pages where one typically only can explain either the construction or the evaluation of the artefact in detail. However, in order to get accepted both parts have to be described. The result is that both construction and evaluation is described superficially. I think journals willing to publish DSR papers should know about this issue and differentiate between 'construction papers' and 'evaluation papers'."

A very few respondents objected to the "exclusive focus on technical artifacts" and "exclusion of context", reflecting a "narrow economic rationalist view of organizations" and "a narrow, functionalist nature of research".

With respect to the different types of artifacts, one respondent usefully noted that "I don't think that any artifact can be introduced (described) without constructs", thereby giving a clear rationale for why explication of constructs is (or should be) mandatory for reporting DSR.

**Table 2.** Part B: DSR Evaluation

| Survey Item | Mean | Std Dev | Max | Min |
|---|---|---|---|---|
| Conducting some sort of evaluation of the designed artifact(s), whether artificial (not real world) or naturalistic (in a real setting) [15] | 8.80 | 1.40 | 10 | 4 |
| Conducting an Artificial evaluation [15] of the designed artifact, using such methods as a criteria-based evaluation, mathematical proof, computer simulation, role-playing simulation, or lab experiment | 6.11 | 2.35 | 10 | 0 |
| Conducting a Naturalistic evaluation [15], i.e., in the real world, with real users using a real instantiation of the design artifact to do real tasks, using such methods as a case study, field experiment, survey of users or other stakeholders, phenomenological or ethnographic study, or action research | 7.18 | 2.16 | 10 | 0 |
| Evaluating the *utility* of the designed artifact for *solving the problem to be addressed* | 8.35 | 1.59 | 10 | 4 |
| Evaluating the *efficiency* of the design artifact | 6.35 | 1.88 | 10 | 0 |
| Evaluating the *efficacy* of the designed artifact in a realistic setting | 7.11 | 2.02 | 10 | 0 |
| Quantitatively measuring the utility, efficiency, or efficacy of the designed artifact | 5.74 | 2.42 | 10 | 0 |
| Evaluating the designed artifact in comparison to other extant solutions to the problem | 7.37 | 2.18 | 10 | 0 |
| Evaluating the designed artifact for side effects (undesirable or desirable) | 6.21 | 2.19 | 10 | 0 |

As shown in Table 2 above, at least some form of evaluation is rated as very important, consistent with the high rating of Hevner et al. [6] guideline 2 in Table 1. Of the two main kinds of evaluation, naturalistic evaluation is rated higher on average than artificial evaluation. Of the areas to be evaluated, utility for solving the problem to be addressed (i.e. meeting the meta-requirements in ISDT terms) is rated as more important than efficiency or efficacy. Of the other aspects of evaluation, rating in comparison to other extant solutions is rated as important, ahead of evaluation for side effects. Quantitative measurement during evaluation was rated the least important, but still of medium importance.

Many respondents commented on the essential, necessary nature of evaluation in DSR. Others were of the opinion that an implementation would need to exist before evaluation, so conceptual work might not be evaluated.

Some respondents noted that naturalistic evaluation is important, but also noted the difficulty in doing so ("resource intensive", "challenging") that the need for it depends on the nature of the designed artifact, and that it should not be mandatory.

One respondent noted that quantified measurements in evaluation increased the chance of publication, but reports of satisfaction from users of the instantiated artifact, especially from opinion leaders such as managers, were also persuasive to reviewers.

Overall, the respondents' comments reflected that the form of evaluation and what was to be evaluated could not be specified acontextually and should instead be selected carefully in line with resources, the stage of the research project or program, the kind of artifact, and the state of the art of the research area.

Table 3 (next page) shows the quantitative results of part C of the survey concerning the relative importance of IS Design Theories (ISDTs). Overall, the development and inclusion of all or part of ISDTs in DSR publications was rated on average as less important than fulfilling the individual guidelines in Hevner et al .[6], but not less important than meeting *all* of the Hevner et al. [6] guidelines and also less important than proper evaluation. Importantly, this area showed the highest variability in ratings, with maxima of 10 and minima of 0 for all items. This clearly reflects the divergence of opinion in the field concerning the mandatory view or irrelevant (nonsensical) nature of ISDTs.

The specification of meta-requirements [18] or purpose and goals [5] was rated the most important, followed closely by the specification of a meta-design [18] or constructs and principles of form and function [5]. The identification and relationship of kernel theories to the meta-design [18] and principles of implementation in specific contexts [5] were rated slightly less on average. Testable hypotheses were rated as the least important, but still of medium importance.

The open-ended comments raised several issues. Some respondents commented that the development and use of "design principles" was more relevant to them than "design theory". Related to this, others commented on the development of mid-range theories as being appropriate rather than full-blown design theories. Another respondent had a staged view of theory development, commenting that "a full and complete one may not be necessary, but a potential one that leads to a complete one should be important."

The difficulty in creating design theory also drew comments. One respondent wrote "Design Theory as of Walls et al. / Gregor&Jones is regarded as creating too much overhead - reduction to essential design theory seems to be useful". Another commented that there is a need for "a simple language for communicating theory".

Most importantly, other respondents indicated they had deep problems with design theory, in line with the division between March and Smith [9] and Walls et al. [18]/Gregor and Jones [5].

**Table 3.** Part C: IS Design Theories

| Survey Item | Mean | Std Dev | Max | Min |
|---|---|---|---|---|
| Specifying a full and complete design theory, e.g. as in Walls et al. [18] or Gregor and Jones [5] | 5.72 | 2.68 | 10 | 0 |
| Specifying the meta-requirements for the generalised problem to be solved [18] or purpose and scope [5] | 6.80 | 2.41 | 10 | 0 |
| Specifying a meta-design (generalised design for meeting the meta-requirements, [18]) or principles of form and function for the design artifact product [5] | 6.49 | 2.43 | 10 | 0 |
| Specifying a design Method for instantiating the meta-design [18] or principles of form and function for the design artifact process [5] | 5.99 | 2.28 | 10 | 0 |
| Specifying kernel theory(ies) [18] or justificatory knowledge [5] relevant to how the meta-design meets the meta-requirements (Walls et al., 1992) | 6.21 | 2.64 | 10 | 0 |
| Specifying kernel theory(ies) [18] or justificatory knowledge [5] relevant to the design method (Walls et al., 1992) | 5.95 | 2.41 | 10 | 0 |
| Specifying testable hypotheses [18] or testable propositions [5] about how well the meta-design meets the meta-requirements (Walls et al., 1992) | 5.79 | 2.72 | 10 | 0 |
| Specifying testable hypotheses [18] or testable propositions [5] about how well the design method results in an artefact consistent with the meta-design [18] | 5.73 | 2.64 | 10 | 0 |
| Specifying constructs as representations of the entities of interest in the theory [5] | 6.06 | 2.79 | 10 | 0 |
| Specifying principle(s) of implementation in specific contexts [5] | 6.09 | 2.46 | 10 | 0 |
| Specifying an expository instantiation [5] | 5.91 | 2.63 | 10 | 0 |

The findings of part D (Table 4 below) concern other miscellaneous guidelines. Relevance and significance of the problem and depth of analysis and clarity of understanding of the problem were rated as very important. Depth of analysis and clarity of problem understanding have not been addressed much in the literature, with the exception of Venable [15, 16]. Having a clear understanding of why an artifact works or doesn't work is also highly rated. Profoundness of insight and novelty are rated as important, but not as high. The size and complexity of the artifact and the effort that went into its development are rated as being of lesser importance, in fact the least important of all the items in the survey. Development effort and elegance were the only items that were not given a rating of 10 by any of the 62 respondents.

**Table 4.** Part D: Other Potential Criteria/Standards

| Survey Item | Mean | Std Dev | Max | Min |
|---|---|---|---|---|
| Relevance of the problem to industry/society clearly established | 8.05 | 1.77 | 10 | 1 |
| Significance of the problem to industry/society clearly established | 7.87 | 1.80 | 10 | 1 |
| Depth of analysis and clarity of understanding of the problem and its causes | 7.92 | 1.42 | 10 | 4 |
| Depth or profoundness of insight leading to the new design artifact | 7.35 | 1.61 | 10 | 3 |
| Novelty of the new design artifact | 7.29 | 2.02 | 10 | 0 |
| Size and complexity of the new design artifact | 4.51 | 2.36 | 10 | 0 |
| Amount of effort that went into the development of the new design artifact(s) | 4.25 | 2.29 | 9 | 0 |
| Elegance of the design of the new artifact(s) | 5.22 | 2.34 | 9 | 0 |
| Simplicity of the design of the new artifact(s) | 5.62 | 2.22 | 10 | 0 |
| Clear understanding of why the new artifact works (or doesn't work) | 7.68 | 2.04 | 10 | 0 |

Several respondents identified additional criteria. One area relates to the stakeholders' perspective(s), including "how easy it is for a user to understand, if it is "packaged" in a user-friendly way", "How usable the artifact is in addition to being useful. Will it burden the potential users instead of easing their effort?", "sensitivity to cultural and social contexts", "Stakeholder interests and analysis", "Impact in a real job/task of the design of the new artifact".

In another area, a few respondents highlighted the need to relate design theories more strongly to behavioural theories (i.e., related to kernel theories).

A third additional very criterion suggested is "Diffusion potential / economic potential (e.g. would you be willing to adopt the artefact? Would you pay for it?)". Market adoption is of course clear evidence of utility!

Finally, two respondents made suggestions that papers can contribute to improving DSR itself, e.g. through "novelty of the approach to Design Science Research" or "contribution to theoretical understanding and practical relevance of design science".

There were also very useful open-ended comments made. Some respondents justified low ratings for size and complexity and development effort. One stated, "A design artifact does not have to be complicated to be valuable. In addition, sometimes an individual may have a stroke of creativity and may arrive at a clever solution to a problem easily. The researcher should not be penalized for their 'amount of effort' nor should we encourage researchers to embellish their process to suggest that their artifact is indeed appropriate."

Other respondents reiterated their concern that guidelines need to be applied carefully according to the research context.

Finally, several respondents reiterated their concern with the whole idea of guidelines, lamenting how they are (inevitably) used, by reviewers and authors alike.

## 5 Discussion and Conclusions

In considering the first research question, this study clearly shows extensive disagreement on what guideline areas should be used as criteria and standards for evaluation of DSR. Nonetheless, there is near consensus on a few areas, such as the need to address and help solve an important problem, to have a clear design artifact, and to have some form of evaluation. Other areas, particularly the development and use of ISDTs are very controversial.

In considering the second research question, it is also clear that there are many views competing with the received view that the Hevner et al. [6] guidelines can or should be used as an evaluation checklist, even for top level journal publications.

First and foremost, many respondents cautioned against the use of the Hevner et al. guidelines (or any guidelines for that matter) as a mandatory checklist for evaluating DSR projects and publications. Further evidence is found in the ratings, with the rating of the survey item that *all* of Hevner et al. [6] guidelines *should* be met receiving only lukewarm support along with weak support for a few of the individual guidelines. The alternative perspective emphasises assessing whether the required rigour of the evaluation and the need to develop IS design theory are relevant and appropriate to the stage and scope of the research; more rigourous evaluation and ISDT development and validation are seen as appropriate for more mature DSR artifacts and not required for early stages of research.

Secondly, the ability of people to apply a fairly complex set of criteria is in more than a little doubt. Here, suggestions to simplify and clarify may be useful. On the other hand, perhaps one needs to recognise that reviewing and evaluating research is difficult and requires careful scrutiny and application of considered judgment.

Finally, some of the surveyed items outside of the Hevner et al. [6] guidelines received fairly average high ratings and additional areas for evaluating DSR were suggested in the open ended comments; these results conflict with a rebellion against mandatory criteria. A potential resolution that I suggest here is to use a cumulative model that adds up the value of the DSR work's contribution to some (but not necessarily all) of the various criteria, rather than the subtractive model inherent in a checklist approach (where all criteria not met fully count against the research). In such a model, the accumulated worth of the research might have a lower required level for acceptance in less rigourous publication venues. Only in top level journals (if indeed even there) would one consider that truly rigourous evaluation or theory development could be required and then only at later stages of the research and maturity of the artifact(s) as above. Moreover, if a high level of rigour is demanded, sufficient space needs to be allowed for explication of the motivation, development, description, evaluation, and theory outcomes of the research being reported.

A few limitations of the research are worth mentioning. First, as a first offering of the survey and initial analysis, the validity of the survey items may be questioned. Therefore, the specific numbers reflected in the averages should be considered as indicative at a high level rather than being precise measurements. Second, as pointed out by some respondents, the idea that some guidelines can be assessed independently from others is also questionable.

So, where to from here? Overall, it seems clear that the diversity of opinion on the topic of criteria, standards, guidelines, and expectations for DSR needs significantly more discussion and refinement before researchers and reviewers can be comfortable with what needs to be done to produce high quality or even acceptable DSR. While Hevner et al. [6] is useful, it is not the be all and end all to the development of DSR; further improvement to the criteria, standards, guidelines, and expectations for DSR seems necessary. The proposal herein for a cumulative or additive model for assessing DSR may be useful. What level of cumulative value is appropriate and the relative importance of the different criteria also need further discussion. Clearly, more work on ISDTs and their role and value in DSR publications is also needed. The additional, non-Hevner-et-al guidelines suggested by this research also need more consideration and discussion. This paper is only one input to that discussion, which, presumably, should never be completely finalized, only reach a point of acceptable equilibrium before the next issue is raised and further change considered.

## References


1. Arnott, D., Pervan, G.: An Assessment of DSS Design Science using the Hevner, March, Park, and Ram Guidelines. In: Information Systems Foundations: Answering the Unanswered Questions about Design Research. Australian National University (2008)
2. Baskerville, R., Pries-Heje, J., Venable, J.: Soft Design Science Research: Extending the Boundaries of Evaluation in Design Science Research. In: Chatterjee, S., Rossi, M. (eds.) Proceedings of the 2nd International Conference on Design Science Research in Information Systems and Technology (2007)
3. Baskerville, R., Pries-Heje, J., Venable, J.: Soft Design Science Methodology. In: Purao, S., Lyytinen, K., Song, I.-Y. (eds.) Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology. ACM Digital Library (2009)
4. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30, 611–642 (2006)
5. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems 8, 312–335 (2007)
6. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
7. Indulska, M., Recker, J.: Design Science in IS Research: A Literature Analysis. In: Information Systems Foundations: Answering the Unanswered questions about Design Research. Australian National University (2008)
8. Klein, H.K., Myers, M.D.: A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly 23, 67–93 (1999)
9. March, S., Smith, G.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)
10. Markus, M.L., Majchrzak, A., Gasser, L.: A Design Theory for Systems that Support Emergent Knowledge Processes. MIS Quarterly 26, 179–212 (2002)
11. Nunamaker Jr., J.F., Chen, M., Purdin, T.D.M.: Systems Development in Information Systems Research. Journal of Management Information Systems 7, 89–106 (1991)

12. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 3, 45–77 (2008)
13. Rossi, M., Sein, M.: Design Research Workshop: A Proactive Research Approach. Presentation delivered at IRIS 26 (2003), http://tiesrv.hkkk.fi/iris26/presentation/workshop_designRes.pdf
14. Vaishnavi, V., Kuechler, B.: Design Research in Information Systems, AISWorldNet, http://www.isworld.org/Researchdesign/drisISworld.htm
15. Venable, J.R.: A Framework for Design Science Research Activities. In: Proceedings of the 2006 Information Resource Management Association Conference (2006a)
16. Venable, J.R.: The Role of Theory and Theorising in Design Science Research. In: Hevner, A., Chatterjee, S. (eds.) Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (2006b)
17. Venable, J.R., Travis, J.: Using a Group Support System for the Distributed Application of Soft Systems Methodology. In: Hope, B., Yoong, P. (eds.) Proceedings of the 10th Australasian Conference on Information Systems, Wellington, New Zealand, pp. 1105–1117 (1999)
18. Walls, J., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3, 36–59 (1992)

# Meta-analysis of Design Science Research within the IS Community: Trends, Patterns, and Outcomes

Olusola Samuel-Ojo, Doris Shimabukuro, Samir Chatterjee, Musangi Muthui, Tom Babineau, Pimpaka Prasertsilp, Shaimaa Ewais, and Mark Young

School of Information Systems and Technology
Claremont Graduate University
Claremont, CA 91711 U.S.A.
osamuelojo@ato-mn.com, doris.shimabukuro@caltech.edu

**Abstract.** The knowledge of design problem and solution is obtained in the building and application of an artifact, which is the end-goals of the design science research. Our objective in this paper is to conduct meta-analysis of the research being published by DESRIST to date to better understand the paradigm of design science research, and to sense the direction that research undertakings are headed. We present a meta-analysis model and detailed analysis of applications, IT artifact types, multi-disciplinary teams, and impact results. Our findings show that the majority of the papers are negatively skewed, systematically clustering more IT artifacts for IS development problem domain. The most cited papers clustered around those that featured better design theories. We concur that researchers need to equally focus on areas with less research including organization and market domains, as well as causal theories of design through a rigorous formative or summative evaluation of artifacts.

**Keywords:** design science research, meta-analysis, problem domain, IT artifact, evaluation method, multi-disciplinary, impact.

## 1 Introduction

In the last few years design science research (DSR) has garnered attention within the IS community. After the landmark article published by Hevner, et al. appeared in MISQ in 2004 [1], this type of research paradigm became acceptable as an alternative to traditional behavioral research published by leading journals. In 2006 DESRIST was created as a stand-alone platform to publish and showcase design science research conducted within the community.

Since the dawn of the digital revolution, IT has changed the way we live, work, play and are entertained. Designers of IT-based digital technology products play a critical role in ensuring that their designed artifacts are not just beautiful, but provide value to their users. The experiences we have when we browse the web, or visit amazon.com, sell or buy on eBay, or play amusing games on our mobile phones have a tremendous impact on how we live our lives [2]. Designing interactions in the digital space presents interesting challenges for IS professionals especially as user interactions with the digital world continue to increase.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 124–138, 2010.

The fundamental principle of DSR is that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artifact. Design is often a complex process, and designing valuable artifacts is further challenged by the need for creative advances in domain areas where existing theories are often insufficient [2].

In the IS discipline, we are concerned with designing artifacts that have a direct impact on organizations and society in general. This is supported by Lee [3] distinguishes IS research from the other fields:

> *"Research in the information systems field examines more than just the technological system, or just the social system, or even the two side by side; in addition, it investigates the phenomenon that emerges when the two interact."*

The term *artifact* is used to describe something that is artificial, or constructed by humans, as opposed to something that occurs naturally [4]. Such artifacts must improve upon existing solutions to a problem or perhaps provide a first solution to an important problem. IT artifacts, which are the end-goal of any design science research project in our community, are broadly classified into [2]:

- Constructs (vocabulary and symbols),
- Models (abstractions and representations),
- Methods (algorithms and practices),
- Instantiations (implemented and prototype systems), and
- Better design theories.

In both Herbert Simon's seminal work, *The Sciences of the Artificial* [4] and Nigel Cross, *Developing a Discipline of Design Science Research* [8], we clearly see the importance they place on doing (i.e., construction). Simon believed that design is concerned with how things ought to be in order to attain goals [4]. Simon saw the design process as generally concerned with finding a satisfactory design, rather than an optimum design. He believed that "both the shape of the design and the shape and organization of the design process are essential components of a theory of design" [4]. Cross, on the other hand, places less importance on theory, emphasizing instead the knowledge that is acquired through the building process [8]:

> *"We must not forget that design knowledge resides in products themselves; in the forms and materials and finishes which embody design attributes. Much everyday design work entails the use of precedents or previous exemplars – not because of laziness by the designer but because the exemplars actually contain knowledge of what the product should be."*

The goal of this paper is to evaluate the trends, outcomes, and impact of design science research by conducting a meta-analysis of the research being published within the DESRIST community from 2006 to 2009 to better understand the paradigm of DSR. In order to achieve this goal, we generate a set of sub-questions to examine the problem domain being studied, interdisciplinary research, types of IT artifacts, evaluation methods, and impact.

### 1.1 Problem Domains Being Studied

- What proportion of papers address each problem domain and which domain reports the highest median artifact?

### 1.2 Interdisciplinary Research

- How are teams organized from various disciplines?

### 1.3 Types of IT Artifacts

- What proportion of papers address each types of IT artifacts?

### 1.4 IT Artifact Evaluation Methods

- How has the evaluation of the artifacts been performed?

### 1.5 Impact, Relevance and Outcomes

- How many DESRIST papers have been cited over the past 4 years?
- Does citation vary with the problem domain?
- Does citation vary with the types of artifact?
- Does citation vary with the inter-disciplinary factor?
- Which direction is design science research headed?
- How is design science research impacting other works?

Through a systematic meta-analysis of all papers published for each DESRIST conference (2006, 2007, 2008 and 2009), we hope to answer some or all of the above questions. In section 2, we present a meta-analysis model to investigate this paradigm. In section 3, we describe the research methodology followed to conduct the meta-analysis. In section 4 we present our basic findings, evaluation and analysis. This is followed by discussion in section 5. We finally conclude this paper in section 6 with some thoughts and insights on where DSR is headed and areas in need of further research.

## 2 Meta-analysis Model

We present in Fig. 1 a framework that guides our research and helps us to understand the paradigm of DSR in IS.

The model proposes that problems and domains studied in design science research inform the type of artifacts generated, the methods used to evaluate these artifacts, and whether inter-disciplinary research collaboration is indicated. These components, in turn, have an effect on relevance and impact of the research. The dimensions of each component of the model are described below.

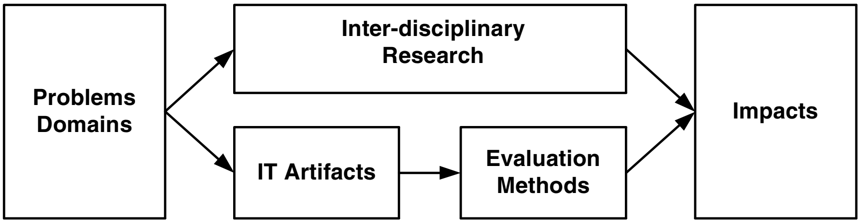



**Fig. 1.** A framework for analyzing DSR in IS

### 2.1 Problem Domains

For this research, we adopted the five core research areas identified by Sidorova et al. [10]: IT and organizations, IS development, IT and individuals, IT and markets, and IT and groups. These areas are considered as a system of problem domains comprising of declarative (IT and individuals, IT and groups, IT and organizations, and IT and markets) and generative (IS development) regularities which interact with each other, similar to interactions in system comprising of agency and structure regularities [18]. The regularities in the system of problem domains are unique and used to categorize the source of problems addressed in these DSR research papers.

### 2.2 IT Artifacts

The types of IT artifacts (i.e., research outputs in DSR) are conceptualized based on the work of March and Smith [11] as follows: construct, model, method, and instantiation. The IT artifact type of better design theories is based on the work of Walls et al. [5], Rossi and Sein [6] and Purao [7]. These IT artifacts are also considered as a system of artifacts consisting of declarative regularities (constructs, models, methods, and instantiations), and generative regularities (better design theories). In the same manner, the regularities in the system are unique and used to categorize the artifacts.

### 2.3 Evaluation Methods

Hevner et al. [1] categorized the methodologies typically used to evaluate artifacts into the following evaluation method groupings: observational, analytical, experimental, testing, and descriptive. The research team assigned a value of “no evaluation” when evaluation of the artifact was not presented.

### 2.4 Interdisciplinary Research

Interdisciplinary research is operationalized as research published by authors from different institutions or different departments within an institution.

### 2.5 Impacts

Impacts or relevance could be examined in terms of instrumental utility and scientific citations. In this study, we operationalized impact in terms of the number of citations a paper received both within and outside the DESRIST community. However, since papers typically require a period of time before they are incorporated into the academic literature as citations, and DESRIST papers are relatively recent publications, citations are expected to be low. We demonstrate the utility of the model by using it to evaluate all previously published DESRIST papers.

## 3 Research Methodology

### 3.1 Identification of Studies for Review

To study the DSR paradigm, we determined that DESRIST conference papers provided an especially appropriate, concentrated source of data for analysis. Created as an international conference, DESRIST specifically showcases design research conducted within the IS community. The first conference in 2006 followed the seminal 2004 paper on “Design Science in Information Systems Research,” in which Hevner et al. described the conduct of DSR in IS. Hevner et al. provide a concise conceptual framework and clear guidelines for understanding, executing, and evaluating this type of research.

To date, DESRIST provides four years of publications from 2006 to 2009, 92 papers in all, as a rich source for meta-analysis. This number includes three invited papers on the National Science Foundation (NSF) Science of Design Project, but excludes papers presented at poster sessions, panel discussions, and doctoral consortiums.

### 3.2 Meta-analysis

The 92 papers were reviewed and categorized along the dimensions of our meta-analysis model: problem domains, IT artifacts, evaluation methods, inter-disciplinary research based on contributing authors’ home department, and scientific impact based on citations. Each paper was reviewed and categorized by at least two authors of this paper, and all categorization differences were discussed and resolved.

## 4 Basic Findings, Evaluation, Analysis

We examine each research sub-questions in turn by performing frequency, variability summary using box plots, and time series analysis on the categories we identified. The time series analysis is based on relating the criterion variable, say 'citation impact' to forecast its future impact using *Auto Regression Integrated Moving Averages* (ARIMA) as implemented in EView software. We chose citation impact as the dependent variable, being explained by the following predictor groups.

### 4.1 Problem Domains Being Studied

**What proportion of papers address each problem domain and which domains need more attention?** Our data showed that 40.2% of the papers were categorized as falling within the core research area of IS development. These papers focused largely on the technical functionality. Another 29.7% of the papers addressed IT and organizations. The remaining papers were categorized as IT and groups (16.3%), IT and markets (8.7%), and IT and individuals (5.4%) respectively. The trend of each problem domain is shown in Fig. 2a.

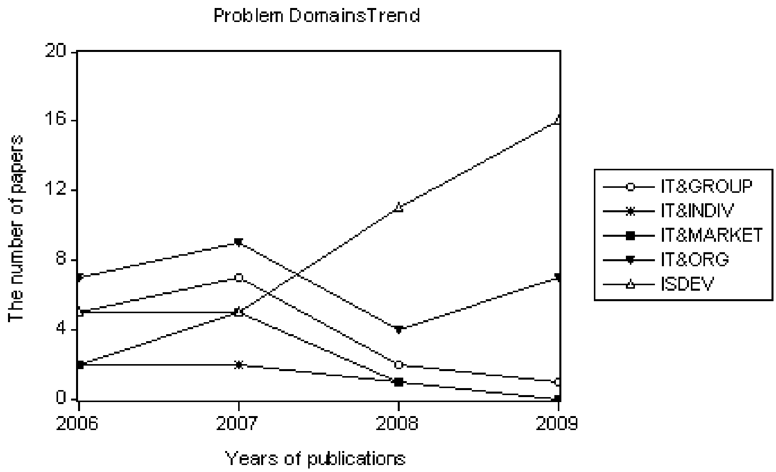



**Fig. 2a.** The trend of problem domains being studied

**Which problem domain reports the highest median artifact?** The composite box plot in Fig. 2b shows that most papers feature IS development related IT artifacts with a distribution that is negatively skewed (i.e. to the left). Interestingly, the median of IS development related IT artifacts is systematically above all artifacts in other problem domains.

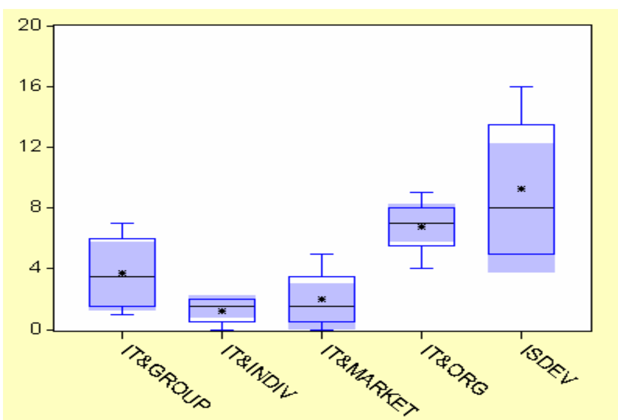



**Fig. 2b.** Problem domain box plot

The ARIMA trend of papers focusing on IT and organization shows that a trend exists with $r^2 = 0.09$ but is statistically insignificant. See fig. 3. More yearly data will be required to establish a better fit. From the T-Statistic test, the trend coefficient is -0.5 implying that the trend matters, but it is decreasing.

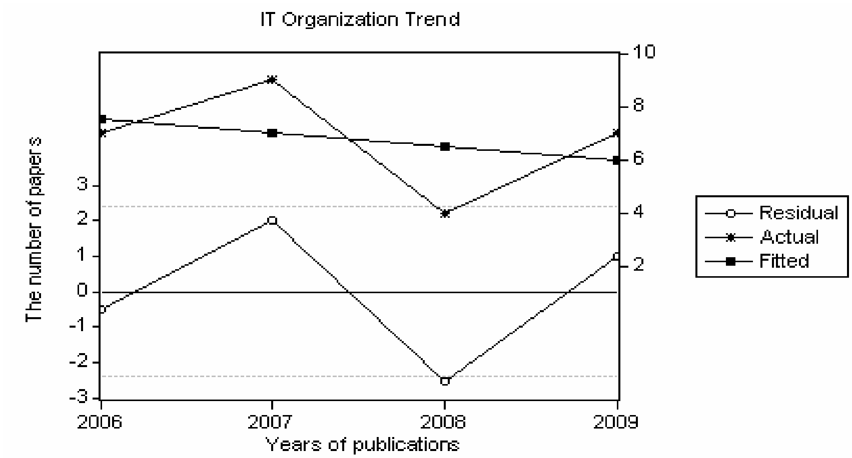



**Fig. 3.** Trend of research focusing on IT and organization problem domain

Chances are more papers would continue to address IS development which would be desirable for building the IS discipline foundation. Having and sustaining a sound IS foundation is salience. However, it is essential that the trend of research undertakings is near evenly distributed across the problem domains.

The IS community needs to focus equally on organizations and markets problems in line with the call made by Benbasat and Zmud [12]. Such efforts will raise DSR impact and benefit practitioners as well as researcher as noted by Sven Carlsson [13]. This area is especially impactful given the high risk of IT implementation failures.

### 4.2 Interdisciplinary Research

**How are teams organized from various disciplines?** Based on the data, approximately half of the papers were disciplinary while the other half were multi-disciplinary.

### 4.3 Types of IT Artifacts

**What proportion of papers address each types of IT artifacts?** The data showed that 40.2% of the papers presented artifacts that were categorized as models while 23.9% presented artifacts categorized as methods and 14.1% as instantiations. Only 12% presented constructs and 9.85% as better theories.

The box plot in fig. 4 shows the median of papers that featured model artifact is higher than those that featured other artifacts.

The ARIMA trend of fig. 5 shows, the regression fit, and a trend exists in the rate of model artifact generation with $r^2 = 0.03$ which is statistically insignificant. More yearly data will be required to establish a better fit. From the T-Statistic test, the coefficient of model artifact generation is 0.28 implying that the trend matters, but it is increasing.

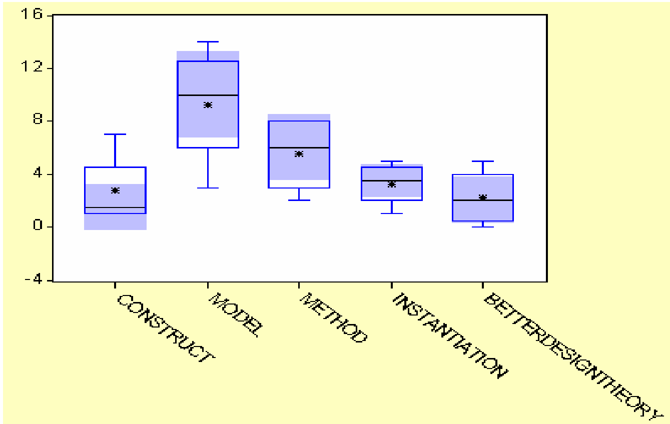



**Fig. 4.** Box plot showing the spread of the artifacts

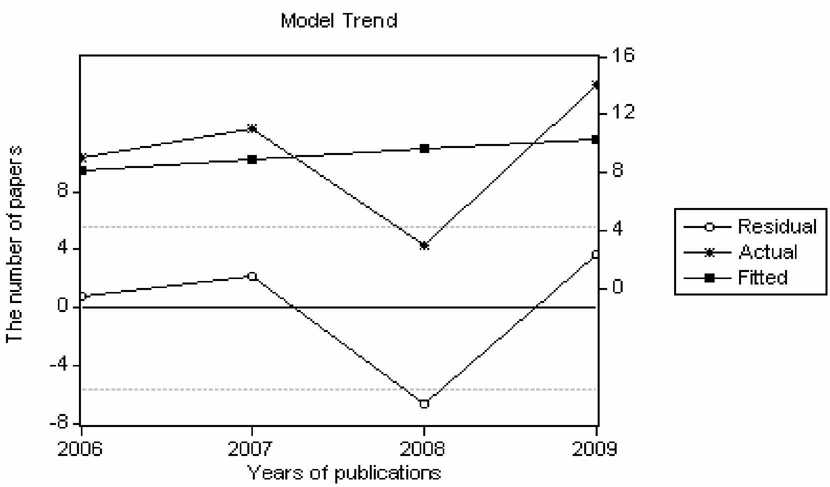



**Fig. 5.** Trend of model as an artifact

It should be noted, however, that few papers present better design theory artifacts. This is in line with Carlile and Christensen [14] that theory building is a cycle involving descriptive and normative stages. These stages demand more time and efforts. Nonetheless, more studies need to focus on building and evaluating artifacts, with particular focus on causal theories over- descriptive theories, are required as we build the IS discipline.

### 4.4 IT Artifact Evaluation Methods

**How has the evaluation of the artifacts been performed?** As fig. 6 shows, the descriptive evaluation method has the highest median of papers and is negatively skewed. The dataset for evaluation is generally skewed. Papers with no specific evaluation recorded a maximum number of up to 7 in 2008 with 10 total over the four-year period evaluated.

### 4.5 Impact

**How many DESRIST papers are being cited over the past four years?** The total number of citations over the four years of DESRIST papers was 196. The year and percent allocation is 65% (2006), 18% (2007), 10% (2008), and 6% (2009). This clearly shows that longevity of the paper being out there helps in building number of citations. But it is also important to note that DESRIST 2006 proceedings was and are still freely available on the Internet. Open access can help with citation as well as is evident from the data.

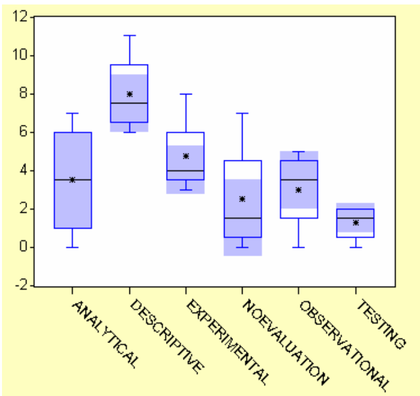



**Fig. 6.** Box plot of evaluation methods

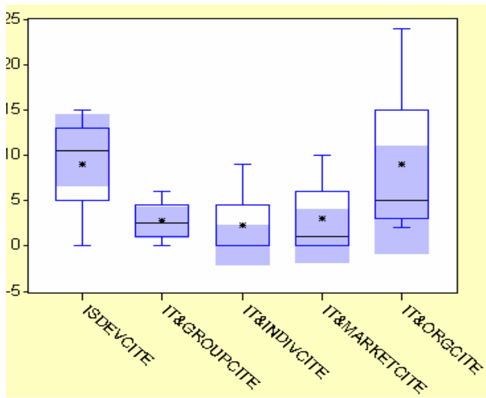



**Fig. 7.** Citation of problem domain

**Does citation vary with the problem domain?** The fig. 7 shows high median count of citation of papers focusing on the implication of IT use for organizations with a distribution that is negatively skewed.

**Does citation vary with the types of artifact?** The fig. 8 shows that half of the papers that feature better design theory had citation count that is more than the average but skewed with a wider variability.

Having a distribution that is negatively skewed with a longer box height for citation count for papers featuring better design theories implies that citations are highly variable. This may be due to the conventional nature of research in the higher education that allows researchers to autonomously determine what to research and at what level of complexity. To achieve the goal of science, we need a regular supply and test of theories of design for typology, prediction, explanation, a sense of understanding, and control of design events. We call for a low variable, regular collection of theories. These theories might take any of the three conceptions: set of theoretical statements of generalization or laws; a set of interrelated axioms; or, propositions featuring existence, relationship or causal processes [16].

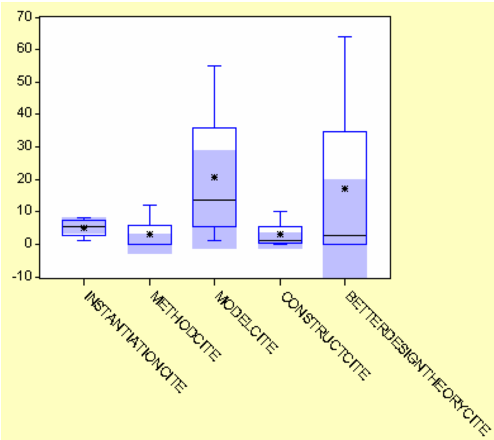



Fig. 8. Citation of types of artifacts

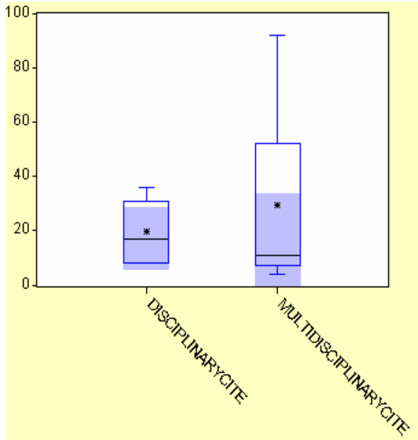



Fig. 9. Citation of multi-disciplinary

**Does citation vary with the inter-disciplinary factor?** The box plot in fig. 9 shows high citation count for papers that are multidisciplinary with a distribution that is negatively skewed.

**Which direction is design science research headed?** The composite box plot in fig. 10 shows high volume of model artifact generation, heavy use of descriptive evaluation method, high volume of work that are disciplinary based, and high volume of research in IS development.

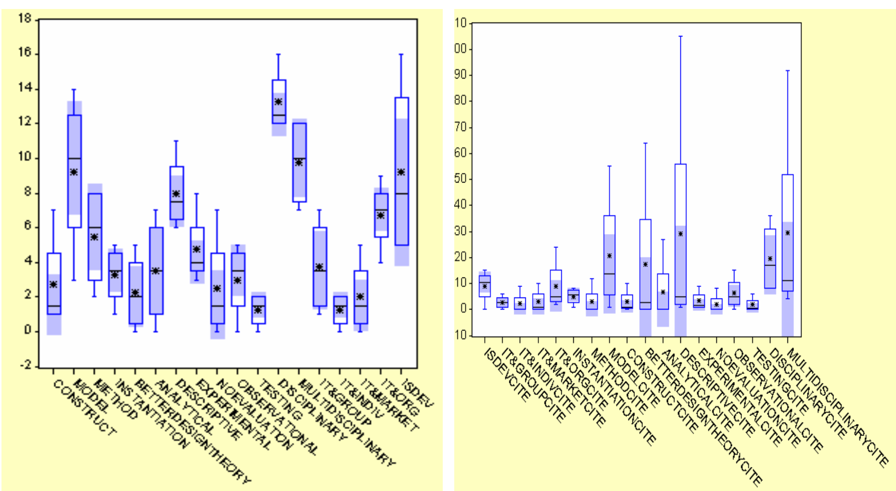



Fig. 10. Composite box plot

The composite box plots shows high citation of papers that feature research in IT and organization problem domain, so also are those that contain model artifacts, those that employ descriptive evaluation method, and those that are multidisciplinary.

Our data also shows the spread of DESRIST community of researchers and those who had published most. Almost 200 researchers contributed papers to DESRIST from 2006 to 2009. A core of 30 researchers from Europe, Australia, the United States and Canada contributed two or more papers each. See the appendix for details.

**How is the design science research impacting other works?** In addition to citations, we also noted the following events in the IS community that indicate increased visibility and acceptance of design science research:

- ICIS started a design science track
- Several WITS CFP has had DSR as major theme
- Special issue on DSR in MISQ in December 2008
- The editorial board in MISQ has added researchers with DSR background.

Also noteworthy is that INFORMS has started a special award for DSR given to systems and prototypes that demonstrate value, and even at the 2009 Workshop on Information Technologies and Systems (WITS 2) conference, prizes were awarded for design.

## 5 Evaluation of Meta-analysis Model

An evaluation itself should be formatively and summatively evaluated against other pertinent standards such that its conduct can be guided. In line with the Hevner et al. guidelines on evaluating DSR, along with the American Evaluation Association standards, we apply meta-evaluation techniques to our model in regards to its utility, feasibility, proprietary, and accuracy [15].

### 5.1 Utility

We used the meta-analysis model to develop the needed dimensions and classified the research based on research problem environment, design research artifacts, artifact evaluation, design process through interdisciplinary collaboration, and impact outcomes. This model might be used as search logic for IT artifacts and as a utility evaluation framework for intended use and intended users of DSR [17].

### 5.2 Feasibility

The parsimony of the model allows us to assess the conference papers and classify them without any disruption to the authors or the conference publishers. We operationalized the primary dimensions at a relatively low cost.

### 5.3 Proprietary

In terms of service, the model is applied to 92 conference papers and we are able to identify the domain and featured primary artifact.

### 5.4 Accuracy

We systematically review all papers, and another author of this paper reviews again. Any categorization differences between authors have been resolved; hence, we have obtained inter-classifier or inter-rater reliability.

## 6 Discussion

Accelerating DSR knowledge production outcome requires that research efforts embrace the cycle of theory building. This cycle includes descriptive and normative stages which must be rigorously evaluated. Evaluation allows the worth or merit of the artifact to be ascertained, and rules out alternate explanations. Evaluation helps balance performance vs. cost, usability vs. security, and timeliness vs. fashionability, which is critical to the design, procurement, and implementation of the artifact as well as accumulation of design science knowledge. The assessment techniques for performance evaluation should include analytical modeling, simulation, field measurement, and quasi- experiment in both formative and summative forms.

Building a robust IS foundation is a key to nurturing the identity. However, the IS community needs to equally focus on organizations and markets problems. While DSR has no widely recognized theories, it provides the research artifacts that help support other theories in related fields. Our research attempts to capture an understanding of the trends and evolution of DSR into a theory-based research field.

We also acknowledge limitations of this study. We obtained citation information based on a basic count from ISI Web of Science and Google Scholars. We did not investigate what the use of the citation was. Scrutinizing what was cited should be a focus of future research, as the citation of scholarly articles is often considered one of the main outcome measures. In addition, the newness of the paper will matter in terms of citation impact in that older papers have a better chance of wider exposure than newer ones and, thus, might be cited more. This newness factor was held as constant in this study.

## 7 Conclusion

We conducted a meta-analysis of the research being published by DESRIST to date using a model, and produced a classification schema for the problem domain, IT artifact and their evaluation types, inter-disciplinary collaboration, and impact results.

From the meta-analysis model, design science researchers should consider dimensions including research problem environment, design research artifacts, artifact evaluation, design process through interdisciplinary collaboration, and impact outcomes in formulating their IT artifact, which is in congruence with Hevner and Chatterjee's design science research cycles [2]. The model might also be used as search logic for IT artifacts and as a utility evaluation framework for IT artifact for intended use and intended user.

In terms of the IT artifact concentration, half of the papers that addressed IS development problem domain featured IT artifact count that is more than the average with a wider variability. The entire IS development problem domain based papers had a median paper count that is above others, thus systematically clustering more IT artifacts. This behavior is salience. However, it is desirable that the trend of research undertakings is near evenly distributed across IS problem domains.

Of a particular interest is the summary variability of the most cited papers, which negatively skewed for papers that featured better design theory. To achieve the goal of science, it is critical that we regularly supply and test theories of design for typology, prediction, explanation, a sense of understanding and control of design events, which can be achieved by focused, interdisciplinary collaboration, and autonomous research. These theories might take any of the three conceptions: set of theoretical statements of generalization or laws, set of interrelated axioms, and propositions featuring existence, relationship or causal processes. Researchers need to focus more on organization and market problem domains while aiming at crossing from descriptive to causal theories of design through a rigorous formative or summative evaluation of artifacts. The products and processes of such undertakings will strengthen the production of design knowledge. They will inform design and refine kernel theories, which in turn will benefit practice.

## References


1. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MISQ 28, 75–105 (2004)
2. Hevner, A.R., Chatterjee, S.: Design Research in Information Systems: Theory and Practice. Springer Publishing, Heidelberg (2010)
3. Lee, A.S.: Editorial. MISQ 25, iii–vii (2001)
4. Simon, H.A.: Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
5. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Assessing Information System Design Theory in Perspective: How Useful was our 1992 Initial Rendition? JITTA 6, 43–58 (2004)
6. Rossi, M., Sein, M.: Design Research Workshop: A Proactive Research Approach. Presentation delivered at IRIS 26, August 9 - 12 (2003), `http://www.cis.gsu.edu/~emonod/epistemology/Sein%20and%20Rossi%20-%20design%20research%20-%20IRIS.pdf` (last accessed January 18, 2010)
7. Purao, S.: Design Research in the Technology of Information Systems: Truth or Dare. GSU Department of CIS Working Paper. Atlanta. Georgia, USA (2002)
8. Cross, N.: Design/Science/Research: Developing a Discipline. In: 5th Asian Design Conference: International Symposium of Design Science, Seoul, Korea, Su Jeon Dang Printing Company (2001)
9. Gregor, S., Jones, D.: The Anatomy of a Design Theory. JAIS 8, 312–335 (2007)
10. Sidorova, A., Evangelopoulos, N., Valacich, J., Ramakrishnan, T.: Uncovering the Intellectual Core of the Information Systems Discipline. MISQ 32, 467–482 (2008)
11. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. DSS 15, 251–266 (1995)
12. Benbasat, I., Zmud, R.W.: The Identity Crisis Within the IS Discipline: Defining and Communicating the Community's Core Properties. MISQ 27, 183–194 (2003)
13. Carlsson, S.A.: Towards an Information Systems Design Research Framework: A Critical Realist Perspective. In: Proceedings of the First International Conference on Design Science in Information Systems and Technology, DESRIST 2006 (2006)
14. Carlile, P.R., Christensen, C.M.: The Cycles of Theory Building in Management Research (2005)

15. Reynolds, P.D.: A Primer in Theory Constructions. Pearson Education, Inc., London (2007)
16. Scriven, M.: The Logic of Evaluation. Edgepress, Inverness (1980)
17. Patton, M.Q.: Utilization Focused Evaluation. Sage Publications, Thousand Oaks (2008)
18. Pawson, R., Tilley, N.: Realistic Evaluation. Sage, London (1997)

## Appendix: The DESRIST Community of Researchers

| No | Name | **Country** | **Department** | **Institution** |
|---|---|---|---|---|
| 5 | Carlsson, SA | Sweden | Dept. of Informatics and Institute of Economic Research, School of Economics and Mgmt | Lund University |
| 4 | Venable, John R. | Australia | School of Information Systems | Curtin University of Technology |
| | Winter, Robert | Switzerland | Institute of Information Management | University of St. Gallen |
| | Vaishnavi, Vijay K. | USA | Computer Information Systems | Georgia State University |
| | Keller, Christina | Sweden | | Jonkoping International Business School |
| 3 | Henningsson, Stefan | Sweden | Dept. of Informatics and Institute of Economic Research, School of Economics and Mgmt | Lund University |
| | Baskerville, Richard | USA | CIS Dept | Georgia State University |
| | D'Aubeterre, Fergie | | Information Systems and Operations Mgmt, Bryan School of Business and Economics | University of North Carolina at Greensboro |
| | Chatterjee, Samir | | School of Information Systems & Technology | Claremont Graduate University |
| | Vandenberg, Art | | Information Systems and Technology | Georgia State University |
| | Singh, Rahul | | Information Systems and Operations Mgmt, Bryan School of Business and Economics | University of North Carolina at Greensboro |
| 2 | Gregor, Shirley | Australia | School of Accounting and Business Information Systems | The Australian National University |
| | Arazy, Ofer | Canada | School of Business | University of Alberta |
| | Parsons, Jeffrey | | Faculty of Business Administration | Memorial University of Newfoundland |
| | Ralph, Paul | | Sauder School of Business | The University of British Columbia |
| | Pries-Heje, Jan | Denmark | | IT University of Copenhagen |
| | Schelp, Joachim | Switzerland | Institute of Information Management | University of St. Gallen |
| | Amoussou, Guy-Alain | USA | | Humbolt State University |

| No | Name | **Country** | **Department** | **Institution** |
|---|---|---|---|---|
| | Hevner, Alan R. | | Information Systems, College of Business | University of South Florida |
| | Horan, Thomas A. | | School of Information Systems and Technology | Claremont Graduate University |
| | Ryan, Terry | | School of Information Systems and Technology | Claremont Graduate University |
| | Jones, Joshua | | Design Intelligence Laboratory, School of Interactive Computing | Georgia Institute of Technology |
| | Rugaber, Spencer | | Design Intelligence Laboratory, School of Interactive Computing | Georgia Institute of Technology |
| | Kuechler, Bill | | Accounting Information Systems | University of Nevada, Reno |
| | Li, Lei | | Computer Information Systems Department | Georgia State University |
| | Murphy, John D. | | Institute of Collaboration Science | University of Nebraska Omaha |
| | Petter, Stacie | | Institute for Collaboration Sciences | University of Nebraska Omaha |
| | Tremblay, Monica Chiarini | | Decision Sciences and Information Systems, College of Business Administration | Florida International University |

# Comparing Two Software Design Process Theories

Paul Ralph

Lancaster University, United Kingdom
paul@paulralph.name
http://paulralph.name

**Abstract.** This paper explores an ongoing conflict concerning the nature of software design. This conflict manifests itself as antagonism between managers and developers, debates about agile vs. plan-driven methodologies and aspiring developers' dissatisfaction with their courses. One side views design as a plan-driven information processing task involving rational decision-making (the Reason-Centric Perspective), while the other views design as an improvised, creative task involving naturalized decision-making (Action-Centric Perspective). Each perspective includes an epistemology, theory of human action and a software design process theory (an explanation of how software is created in practice). This paper reports the results of an exploratory questionnaire study that comparatively and empirically evaluated the two process theories. Results clearly favor the Action-Centric process theory: the Sensemaking-Coevolution-Implementation Framework.

**Keywords:** Design Science, Process Theory, Software Design, Questionnaire.

## 1 Introduction

Software design science is the philosophical, theoretical and empirical study of software creation and modification including its phenomenology, methodology and causality. It is distinct from the "design-science research paradigm" [1, 2], where design is a research method. A key element of design science involves theories of the shape and organization of the design process [3]. Yet, the shape and organization of the design process of software, in particular, is not well understood [3-6], as most academic work on software design is prescriptive rather than explanatory or descriptive [see 7, 8] – hence, my primary research question (as follows).

**Research Question:** *What is the process by which development teams create software in practice?*

To address this question, I evaluate competing process theories of software design, in terms of their descriptive and explanatory validity. A process theory is "an explanation of how and why an organizational entity changes and develops" [9, p. 512]. Process theories are distinct from process models – "A process model is an abstract description of an actual or proposed process" [10, p. 76]. A process theory seeks to explain how outcomes materialize *in general*, not simply one or several historical or possible activity sequences. Following [11], software design (verb) is

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 139–153, 2010.

the act of creating a specification of a software object, by an agent, intended to accomplish goals in a particular environment, using a set of primitive components, satisfying a set of requirements, subject to set of constraints.

This research contributes to the field of *software design science,* the philosophical, theoretical and empirical study of the creation of virtual artifacts, including performance (phenomenology), methods, tools and practices (methodology), and antecedents and outcomes (causality). This should not be confused with the *design science research paradigm* [1, 2, 12], where knowledge is created by building and evaluating technological artifacts.

This paper is organized as follows. Section two categorizes existing research on software design into two mutually-exclusive clusters of interrelated theoretical and philosophical concepts (Reason-Centric and Action-Centric Perspectives). Section three describes the design and results of a survey study that comparatively tested process theories from both perspectives. Section four summarizes the contributions of this phase of the study and outlines the next.

## 2 Two Perspectives on Software Design

This section summarizes the Reason- and Action-Centric Perspectives [8], and defines the process theories used to operationalize each perspective.

### 2.1 The Reason-Centric Perspective

"According to the model of Technical Rationality – the view of professional knowledge which has most powerfully shaped both our thinking about the professions and the institutional relations of research, education, and practice – professional activity consists in instrumental problem-solving made rigorous by the application of scientific theory and technique," [13, p. 21]. Technical Rationality requires given problems – goals are agreed in advance and constraints are knowable. Schön argues that Technical Rationality is foundational to both positivism [14] and the Technical Problem-Solving design paradigm [3]. The latter posits that professionals design by optimizing or "satisficing" a design candidate vis-à-vis known constraints and objectives. They engage in rational decision making – choosing the best option from a known set [15].

Technical Rationality and the Technical Problem-Solving paradigm are consistent with the *cognitivist view* of human action, wherein actions are executed and understood through a plan and defined as "a sequence of actions designed to accomplish some preconceived end" [16]. Plans are prerequisites to action. Unanticipated conditions trigger replanning; evaluation is performed by comparing resulting and planned actions and outcomes. In this view, design is a form of plan-driven problem-solving, where an agent seeks a goal state by executing a plan within a field of constraints [17]. Moreover, this view is guided by an Information Processing metaphor – "The designer is seen as a machine capable of rationally selecting and connecting together elemental information to satisfy a set of constraints" [18, p. 309].

The Function-Behavior-Structure (FBS) Framework [19, 20] is an engineering design process theory, broadly consistent with the above. Figure 1 shows one representation of the FBS Framework; Tables 1 and 2 define its artifacts and processes.

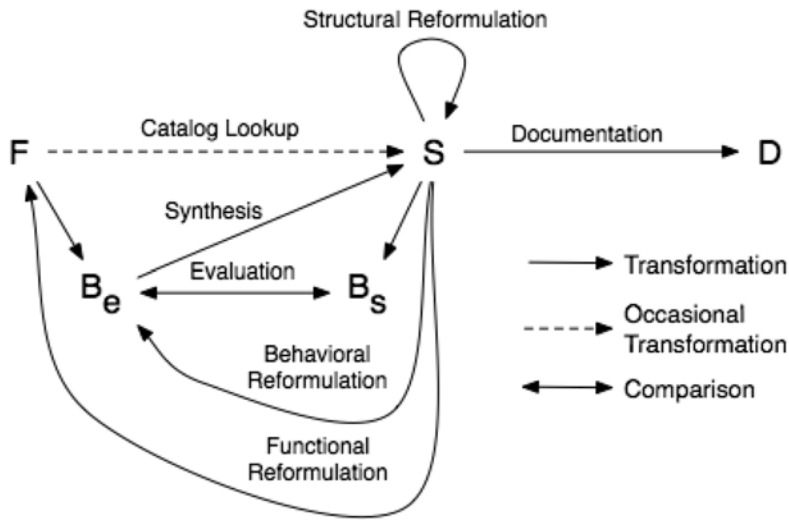



**Fig. 1.** The Function-Behavior-Structure-Framework

**Table 1.** Artifacts of the FBS Framework (adapted from [19])

| Symbol | Meaning |
|---|---|
| Be | expected (desired) behavior of the structure |
| Bs | "the predicted behavior of the structure" (p. 3) |
| D | a graphically, numerically and/or textually represented model that transfers "sufficient information about the designed artefact so that it can be manufactured, fabricated or constructed" (p. 2) |
| F | "the expectations of the purposes of the resulting artefact" (p. 2) |
| S | "the artefact's elements and their relationships" (p. 2) |

The core claim of the FBS Framework is that "the purpose of designing is to transform *function*, F (where F is a set), into a *design description*, D, in such a way that the artefact being described is capable of producing those functions," [19, p. 2, original italics]. Gero posited three "intermediate artifacts" – structure (S), structure's behavior, ($B_s$) and expected behaviors ($B_e$). Gero and Kannengiesser (2002) situated function, behavior and structure in three different "worlds" – desired, internal and external – where each concept exists in each world (e.g., desired functions, the designer's interpretation of the functions of the current design candidate, and external representations of said interpretations). Kruchten [21] claimed that software design may be "cast" in the FBS Framework and consequently mapped the Rational Unified Process [22] and Waterfall Model [23] onto it.

**Table 2.** Operations of the FBS Framework (adapted from [19])

| Operation | Inputs | Outputs | Meaning |
|---|---|---|---|
| Analysis | S | $B_s$ | the process of deriving the behavior of a structure |
| Catalog Lookup | F | S | selecting a known structure that performs the required function |
| Evaluation | $B_s$ & $B_e$ | Differences Between $B_s$ and $B_e$ | comparing predicted behavior to expected behavior and determining whether the structure is capable of producing the functions |
| Formulation | F | $B_e$ | deriving expected (desired) behaviors from the set of functions |
| Production of Design Documentation | S | D | transforming structure into design description suitable for manufacturing |
| Synthesis | $B_e$ | S & $B_s$ | "expected behavior is used in the selection and combination of structure based on a knowledge of the behaviors produced by that structure" (p. 3) |

### 2.2 The Action-Centric Perspective

Social constructivism posits that knowledge is derived from social interactions [24]. Building from social constructivism and empirical studies of professional practice, Schön [13] devised the Reflection-in-Action design paradigm, where design is a reflective conversation between the designer and the situation. The designer alternates between framing (conceptualizing the problem), making moves (where a move is a real or simulated action intended to improve the situation) and evaluating moves. Multiple agents may collectively reflect in action using boundary objects [25].

Schön [13] argued that "when someone reflects in action, … he does not keep means and ends separate … he does not separate thinking from doing" (p. 69). This idea is elemental to the *ethnomethodological view* of human action (ethno-view), in which "the organization of situated action is an emergent property of moment-by-moment interactions between actors, and between actors and the environments of their action" [16, p. 179], while "plans are representations, or abstractions over action" (p. 186). Both Reflection-in-Action and the ethno-view imply that innovation is based on the creativity and experience of the designer; consequently, the guiding metaphor is creativity [18] and decision making is naturalistic [26].

The Sensemaking-Coevolution-Implementation (SCI) Framework [27, 28] (Figure 2, Table 3) is a design process theory that is broadly consistent with the above concepts. Unlike the FBS Framework, the SCI Framework is specific to software design. Its core claim is that software design includes three primary activities (in no set order) – making sense of context, iteratively evolving mental pictures of context and software artifact, and writing code based on the mental picture of the software.

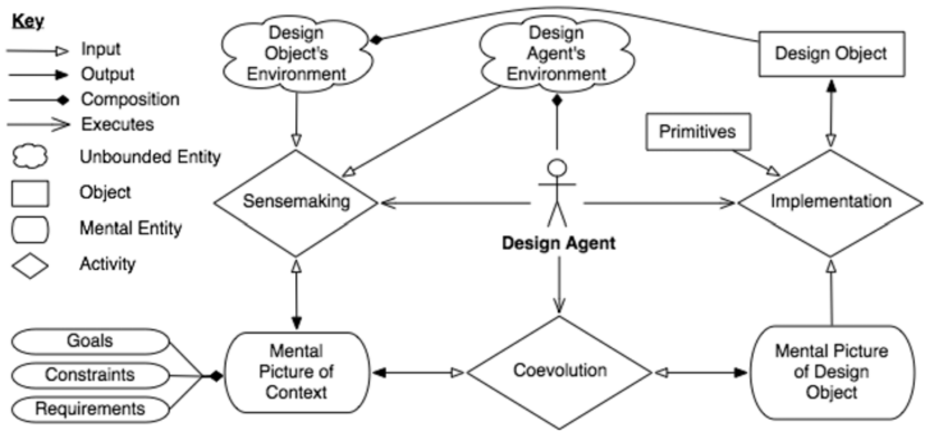



**Fig. 2.** The Sensemaking-Coevolution-Implementation Framework

### 2.3 Comparative Analysis of Perspectives and Process Theories

**Perspectives in Conflict.** Table 4 contrasts the Reason- and Action-Centric Perspectives, which largely constitute the assumptions of the FBS and SCI Frameworks. The conflict between these perspectives is evident in Beck's discussion of common tensions between managers who try to drive projects through cost estimates and developers who cannot reliably estimate complex projects [29]. Similarly, Graham explains misalignment between programming education and practice – "I was taught in college that one ought to figure out a program completely on paper before even going near a computer. I found that I did not program this way.... I tended to just spew out code that was hopelessly broken, and gradually beat it into shape" [30]. Moreover, "the concept of method ... occupies an extremely privileged status in formal information systems development thought" while "the possibility that amethodical development might be the normal way" of building systems has "almost entirely elud[ed] the systems development literature" [31, p.54, 58]. The Reason-Centric perspective has occupied an analogously privileged status in design research despite little empirical evidence concerning its assumptions and ramifications.

**Process Theory Similarities.** Despite differing assumptions, the two process theories are similar in several ways:

1. They are both teleological process theories; i.e., explanations of how and why an entity changes *wherein change is manifested by a goal-seeking agent that engages in activities in a self-determined sequence, and monitors progress* [9, 32, 33]. Therefore, they share fundamental aspects of teleological process theories, including goals and an agent.
2. They share fundamental design concepts; e.g., the FBS Framework's *expected behavior* and *structure* concepts are similar to the SCI Framework's *requirements* and *mental picture of the design object* concepts (see Tables 1 and 3).

**Table 3.** Concepts of the SCI Framework (adapted from [28])

| Concept | Meaning |
|---|---|
| Constraints | a restriction on a structural or behavioral property of the *design object* |
| Design Agent | an entity or group of entities that is capable of forming intentions and goals and taking actions to achieve those goals, and that specifies the structural properties of the *design object* |
| Design Object's Environment | the totality of the surroundings in which the *design object* exists or is intended to exist |
| Design Agent's Environment | the totality of the surroundings of the *design agent* |
| Design Object | a (possibly incomplete) manifestation of the *mental picture of design object*, composed of *primitives*, in the *design object's environment* |
| Goals | optative statements (which may exist at varying levels of abstraction) about the effects the *design object* should have on the *design object's environment* |
| Mental Picture of Context | the collection of all beliefs, held by the *design agent*, regarding the *design agent's environment* and the *design object's environments* |
| Mental Picture of Design Object | the collection of all beliefs held and decisions made by the *design agent* concerning the *design object* |
| Primitives | the set of entities from which the *design object* may be composed |
| Requirements | a structural or behavioral property that a design object must possess |
| Sensemaking | the process where the *design agent* perceives its *environment* and the *design object's environment* and organizes these perceptions to create or refine the *mental picture of context* |
| Coevolution | the process where the *design agent* simultaneously refines its *mental picture of design object* based on its *mental picture of context*, and vice versa |
| Implementation | the process where the *design agent* generates or updates a *design object* using its *mental picture of design object* |

**Table 4.** Comparison of Reason- and Action-Centric Perspectives

| Dimension | Reason-Centric Perspective | Action-Centric Perspective |
|---|---|---|
| Epistemology | Positivist | Constructivist |
| Theory of Action | Cognitivist | Ethnomethodological |
| Design Paradigm | Technical Problem-Solving | Reflection-in-Action |
| Decision Making | Rational | Naturalistic |
| Guiding Metaphor | Information Processing | Creativity |
| Process Theory | FBS Framework | SCI Framework |

3. *Both frameworks are consistent with models.* In the FBS Framework, the designer necessarily creates an external representation of the design artifact's structure and may also model functions and behaviors. In the SCI Framework, the design agent may model both the mental pictures of the context (conceptual models) or the design object (design models).

**Process Theory Differences.** Notwithstanding these similarities, the two theories differ in at least three ways.

1. Whether problem setting and problem solving are separate (FBS Framework) or cotemporal and inextricably linked (SCI Framework)
4. Whether the coding process is driven by prefigured decisions (FBS Framework) or evolves iteratively with the design process (SCI Framework)
5. Whether designers focus on models (FBS Framework) or code (SCI Framework)

The first difference results from the conflicting design paradigms underlying the two theories, the second from their dissimilar theories of action, the third from the differing guiding metaphor. Therefore, comparatively testing the two process theories on these dimensions may give insight into the descriptive validity of the underlying assumptions encompassed by the Action and Reason-Centric Perspectives.

## 3 Research Design and Results

Taking a comparative approach to testability [34, 35], my original research question may now be operationalized as *Which of the FBS and SCI Frameworks more accurately describes how software is created in practice?*

My literature review did not uncover any previous empirical evaluations of either theory in the software domain. Moreover, I uncovered little methodological advice on evaluating process theories. However, Wolfe [36] identified two common approaches to studying innovation processes – cross-sectional surveys and in-depth field studies. Considering the similarity between design and innovation, it would seem reasonable to adopt these methods here. Furthermore, combining the two approaches enables multi-method triangulation – the survey (phase 1) allows for larger sample size and reliability while the field study (phase 2) facilitates gathering deep insights into developer behaviors and cognitive processes. This paper focuses on the survey, which I designed based on well-known guidelines [37-39].

### 3.1 Hypothesis

I hypothesize that the SCI Framework is more accurate, as its underlying design paradigm (Reflection in Action) and theory of human action (Ethno-View) are better supported by empirical studies than their Reason-Centric alternatives [13, 16].

*Hypothesis H1:* The SCI Framework more accurately reflects how software is created in practice than the FBS Framework.

### 3.2 Instrument Development and Validation

The steps in the instrument development and validation were as follows.

1. The author identified differences between the two theories.
2. A colleague with expert knowledge of software design reviewed these differences, finding no bias in the interpretation of either theory.
3. The author generated approximately 80 items concerning these differences.
4. Items were reviewed by two MIS faculty, one with extensive experience in questionnaire-based research, the other with extensive knowledge of design.
5. A pilot was conducted with three professional developers and seven MIS PhD students to get research-oriented feedback. Items were revised.
6. A second pilot with 12 professional developers was conducted. Results indicated that the questionnaire was too long and difficult to understand. Most items were dropped; remaining items were simplified.
7. A third pilot with 10 professional developers was conducted. Minor revisions were made, resulting in the final version of the instrument.

Following this process, the questionnaire comprised 13 items (listed in the Appendix). Each item was constructed with six responses: one strongly supporting each framework; one supporting each framework; one neutral; one "Not Applicable / Don't know." The question order was randomized; the answer order varied by question. Please note, these items are *not* reflective indicators of latent constructs. Differences between process theories are not latent constructs and items do not reflect these differences as much as describe certain behaviors and attitudes related to the differences. For example, from Difference 1 (whether problem setting and solving are separate), the survey included the item "The process of designing the software has NOT helped my team better understand the context in which the software is intended to be used."

### 3.3 Sampling and Administration

The population of interest includes all members of all software development teams, worldwide. However, for practical reasons, I limit the sample to English speakers. Moreover, having no comprehensive population list, random sampling was impractical. Instead, participants were recruited through posts on popular software development blogs and through Twitter. The questionnaire was administered online.

Between December 2, 2009 and January 11, 2010, 1384 participants responded to the survey. The response rate cannot be calculated since, as in snowball sampling, the sample size is undefined. However, of the 4410 individual visitors to the survey page, 1384 completed it (31%), 1118 partially completed it and 1908 bounced (looked at the survey's front page and then left). Table 5 summarizes their key demographics.

Responses were received from 65 countries across six continents with concentrations in the United States (549), Canada (176), United Kingdom (118) and Australia (73). Participants indicated fulfilling varied roles (they could choose several), including developer (1325), analyst (569), quality assurance specialist (533), manager (266) and graphics designers (195). Respondents reported using a wide variety of agile (e.g., Scrum), plan-driven (e.g., the Rational Unified Process) and homegrown methodologies. When asked "Is your project more 'social' (like a website) or 'technical'

**Table 5.** Summary of Sample Demographics

| Dimension | Mode | Minimum | Maximum |
|---|---|---|---|
| Years of Experience | 1 to 5 years* (31.5%) | < 1 year (2.9%) | > 25 years (3.6%) |
| Education | Bachelor's Degree* (48%) | Some School (1.7%) | PhD (4.1%) |
| Company Size | 1 to 10 (29%) | 1 to 10 (29%) | >10 000 (10.5%) |
| **Dimension** | **Mean** | **Standard Deviation** | **Range** |
| Team Size | 11 members | 83 members | 3000 members |
| Project Length | 1.9 years | 2.4 years | 20 years |

(like a device driver)", participants answered: more social – 34%; more technical – 29%; in between – 36%.

### 3.4 Results

Before presenting the results, I enumerate the possible patterns and their interpretations, assuming responses are coded from 1 (strong support for the FBS Framework) to 5 (strong support for the SCI Framework).

1. A symmetric distribution (median of 3) would indicate that neither framework is substantially more accurate than the other.
2. A positively-skewed distribution (median of 1 or 2) favors the FBS Framework.
3. A negatively-skewed distribution (median of 4 or 5) favors the SCI Framework.
4. A bimodal distribution (e.g., modes of 2 and 4) would indicate that developers can be categorized into two groups, one supporting each framework.
5. A combination of symmetric, positively and negatively skewed items would suggest a problem with the survey instrument.

The results are given in Table 6 (please note: columns do not total 1384 as each question had a "N/A" option). It is clear from inspecting Figure 3 that the overall distribution favors the SCI Framework. The same pattern is observed at both the item level (each item had a median response of 4 or 5) and the individual level (96.6% of respondents had a median response of 4 or 5). In summary, the response distribution is negatively skewed, supporting the SCI Framework.

**Table 6.** Questionnaire Results

| Item | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Strong FBS | 7 | 13 | 14 | 20 | 62 | 22 | 22 | 13 | 17 | 58 | 23 | 13 | 9 |
| FBS Framework | 38 | 66 | 42 | 76 | 161 | 61 | 97 | 39 | 63 | 168 | 173 | 174 | 67 |
| Neutral | 72 | 162 | 109 | 120 | 195 | 78 | 113 | 55 | 122 | 148 | 320 | 299 | 303 |
| SCI Framework | 597 | 662 | 576 | 572 | 572 | 398 | 539 | 452 | 539 | 492 | 671 | 623 | 562 |
| Strong SCI | 656 | 446 | 628 | 575 | 349 | 819 | 592 | 796 | 620 | 505 | 173 | 155 | 425 |
| Median | 4 | 4 | 4 | 4 | 4 | 5 | 4 | 5 | 4 | 4 | 4 | 4 | 4 |
| Mode | 5 | 4 | 5 | 5 | 4 | 5 | 5 | 5 | 5 | 5 | 4 | 4 | 4 |

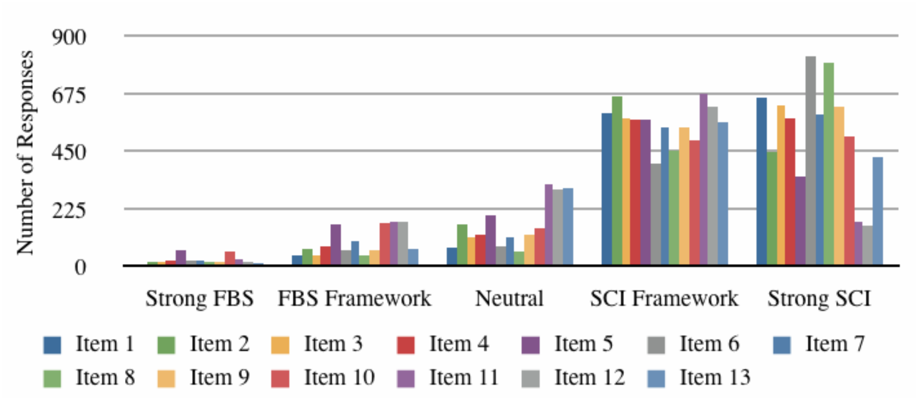



**Fig. 3.** FBS/SCI Agreement Across 13 Items

Many methodologists and statisticians disagree as to whether Likert scales of the kind used in this research produce interval or ordinal data, and consequently as to whether to apply parametric or nonparametric tests [40]. Here, I take the more cautious route, treating the data as ordinal.

Nonparametric tests (such as chi-square) require an expected distribution to compare with the observed distribution. Since there is no *a priori* "FBS-supporting distribution", I generated one (for each item) by reflecting the observed distribution (subtracting each response from 6). The resulting chi-square statistics (with significance via the sign test) indicate that the observed distribution of each item is significant at $p < 0.001$ (Table 7). This answers the question, 'is the observed distribution significantly different from an equally compelling distribution supporting the alternative hypothesis?' Substituting normal and uniform distributions produced similar results.

**Table 7.** Chi-square Test Results - Observed vs. Reflected Distribution

| Item | Z | Asymp. Sig. (2-tailed) | Item | Z | Asymp. Sig. (2-tailed) |
|---|---|---|---|---|---|
| 1 | -33.49 | p < 0.001 | 8 | -33.18 | p < 0.001 |
| 2 | -29.90 | p < 0.001 | 9 | -30.48 | p < 0.001 |
| 3 | -32.21 | p < 0.001 | 10 | -22.13 | p < 0.001 |
| 4 | -29.92 | p < 0.001 | 11 | -20.10 | p < 0.001 |
| 5 | -20.47 | p < 0.001 | 12 | -18.84 | p < 0.001 |
| 6 | -31.45 | p < 0.001 | 13 | -27.87 | p < 0.001 |
| 7 | -28.53 | p < 0.001 | | | |

In addition to the thirteen items, several demographic and project variables were included in the questionnaire, including gender, education, experience, nationality, occupation, team size, project duration, firm size, methodologies in use, and the nature of the software. Although space does not permit a thorough presentation of the analysis, *none of these variables had a measurable effect on individuals' overall agreement with the FBS or SCI Framework.*

# 4 Implications for Research and Practice

## 4.1 Contributions

To the best of my knowledge, this study is not just the first empirical evaluation of the FBS and SCI Frameworks but of any software design process theory. The evidence supporting the SCI Framework (and vicariously the Action-Centric Perspective) lends further support to a growing body of evidence questioning the centrality of rational thought in design and other professional activity [e.g., 13, 18, 31, 41-43]. Moreover, the SCI Framework is immediately useful for both research, practice and teaching.

1. For researchers, it may facilitate evaluating and improving design methods, tools and practices. For example, in evaluating a design methodology (e.g., Extreme Programming), we may ask, "does this methodology provide guidance concerning all three fundamental design activities – sensemaking, coevolution and implementation?" If not, can the methodology be improved by considering those omitted?" Moreover, it may inform development of an antecedent theory of design project success. In a strict interpretation of causality, causal theories imply precedence relationships. The SCI-Framework dispenses with Waterfall-like, artificial activity sequences. Therefore, it may help eliminate extraneous causal relationships during theory building (e.g., the hypothesis that analysis quality causes design quality is incorrect *a priori* since analysis and design are cotemporal in practice).
2. For educators, it may inform evaluation and improvement of software design curricula. For example, the presented evidence implies that the SCI Framework is a better description of software design than the Waterfall Model [23] (which is a subset of the FBS Framework [21]); therefore, it may be more useful to teach the concepts of the SCI Framework in design-oriented courses.
3. For managers, it suggests that developers may resist attempts to pressure them to separate analysis from design, write code linearly or iterate on models; that implementing a tool, practice or method that is incompatible with iterative coding and simultaneous analysis and design will likely be ineffective without corresponding changes in development practices; and that managers who believe that their employees build software according to the Reason-Centric Perspective (that is, rationally) or using a Reason-Centric method (e.g., Waterfall) are likely mistaken or possibly actively being deceived. Furthermore, if developers do not understand the problems that they are solving until the solution is well into development, any upfront budget and schedule estimates lack substantive understanding of the problem. It seems incredulous that anyone could accurately estimate the cost of solving a problem without knowing what identifying the problem.

## 4.2 Limitations

The results of this study should be considered in light of four limitations:

1. The sample is not random and may include some bias. However, given the variety in the reported demographics, suggesting that the sample comprises only one or several fringe developer communities seems incredulous.
2. The limitations inherent to survey research, including lack of depth and responder bias, obviously apply here. Phase two of the study (described below) is designed to mitigate these shortcomings.

3. As the test was comparative, it does not indicate that the SCI Framework is unequivocally “right” or “true”. It is simply more accurate than the alternative.

### 4.3 Future Work (Phase 2)

As mentioned above, a multimethodological research design combining a survey with one or more in-depth fields studies would provide more convincing evidence than either approach alone. Following this, the next phase of the current study involves comparatively evaluating the FBS and SCI Frameworks using a field study to corroborate (or contradict) and add nuance to the current evidence.

One form of field study with a rich methodological foundation in organizational research is the case study [c.f. 35, 44-47]. A case study is a “comprehensive research strategy” that “investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident … [and] relies on multiple sources of evidence, with data needing to converge in a triangulating fashion” [35, p. 13-14]. A case study approach is preferable when 1) the research focuses on how things are done in practice, 2) the research focuses on contemporary events, and 3) the research does not necessitate behavioral manipulations [35]. The present situation clearly meets these criteria.

I propose a three-case design comprising two literal replications and one theoretical replication (two studies of software development teams where the same result (SCI Framework superior) is predicted and one study of an engineering design team, where a different result (FBS Framework superior) is predicted). The proposed design is informed by the incisive summary of recommendations in [44]. Data collection may include interviews, recording meetings, direct observation and copying relevant artifacts (e.g., design diagrams). The resulting collection of statements, observations and artifacts can then be coded according to a closed coding scheme based on the two theories. Specifically, for each concept and relationship of each theory, related items of evidence would be classified as ether *supporting* or *opposing*. The extent of support for each theory would reflect the cumulative support for each concept and relationship. At least two coders will be used to facilitating measurement of reliability via intercoder agreement [35, 44, 46, 47].

### 4.4 Concluding Remarks

“The shape and organization of the design process is an essential component of a theory of design” [3, p. 130-1]. Since “the shape and organization of the design process” in the software domain is poorly understood [3-6], this study began with the question, *What is the process by which development teams create software in practice?* This question was operationalized as an empirical, survey study comparing two incompatible software design process theories; i.e., explanations of the shape and organization of the design process. The SCI Framework – in which design is modeled as an improvised, emergent activity wherein a self-directing agent alternates between three primary activities: 1) making sense of context; 2) iteratively evolving mental pictures of context and software artifact; 3) writing code based on the mental picture of the software – was supported. Since the differences between the FBS and SCI Frameworks tested reflect differences in the assumptions comprising the Reason and

Action-centric perspectives, this evidence also suggests that the Action-Centric Perspective is more consistent with the pragmatic reality of software design than the Reason-Centric Perspective. Since the Reason-Centric Perspective has held a privileged position in design literature for many years [31], this evidence calls into question much of the field's conceptual research, the potential usefulness of popular design methodologies, and the conventional wisdom surrounding how software designers are educated and how software projects are managed.

## References


1. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
2. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
3. Simon, H.A.: The Sciences of the Artificial. MIT Press, Cambridge (1996)
4. Freeman, P., Hart, D.: A Science of design for software-intensive systems. Communications of the ACM 47, 19–21 (2004)
5. Sullivan, K.: Preliminary Report: NSF Workshop on the Science of Design: Software and Software-Intensive Systems. University of Virginia Department of Computer Science, Airlie Center (2003)
6. Wynekoop, J., Russo, N.: Systems development methodologies: unanswered questions. Journal of Information Technology 10 (1995)
7. Wynekoop, J., Russo, N.: Studying system development methodologies: an examination of research methods. Information Systems Journal 7, 47–65 (1997)
8. Ralph, P.: The Sensemaking-Coevolution-Implementation Framework of Software Design. MIS Quarterly (under review), 76 pages (2010)
9. Van de Ven, A.H., Poole, M.S.: Explaining development and change in organizations. The Academy of Management Review 20, 510–540 (1995)
10. Curtis, B., Kellner, M.I., Over, J.: Process Modeling. Communications of the ACM 35, 75–90 (1992)
11. Ralph, P., Wand, Y.: A Proposal for a Formal Definition of the Design Concept. In: Lyytinen, K., Loucopoulos, P., Mylopoulos, J., Robinson, W. (eds.) Design Requirements Engineering: A Ten-Year Perspective. LNBIP, pp. 103–136. Springer, Heidelberg (2009)
12. 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)
13. Schön, D.A.: The reflective practitioner: how professionals think in action. Basic Books, USA (1983)
14. Hacking, I.: Scientific Revolutions. Oxford University Press, New York (1982)
15. Luce, D., Raiffa, H.: Games and Decisions: Introduction and Critical Survey. Wiley, New York (1957)
16. Suchman, L.: Plans and Situated Actions: The problem of human-machine communication. Cambridge University Press, Cambridge (1987)
17. Newell, A., Simon, H.: Human Problem Solving. Prentice-Hall, Inc., Englewood Cliffs (1972)
18. Love, T.: Philosophy of Design: A Meta-theoretical Structure for Design Theory. Design Studies 21, 293–313 (2000)
19. Gero, J.S.: Design prototypes: A Knowledge Representation Schema for Design. AI Magazine 11, 26–36 (1990)

20. Gero, J.S., Kannengiesser, U.: The Situated Function-Behaviour-Structure Framework. Design Studies 25, 373–391 (2004)
21. Kruchten, P.: Casting Software Design in the Function-Behavior-Structure Framework. IEEE Software 22, 52–58 (2005)
22. Kruchten, P.: The Rational Unified Process: An Introduction. Addison-Wesley Professional, Reading (2003)
23. Royce, W.W.: Managing the development of large software systems: concepts and techniques. In: Proceedings of Wescon (1970)
24. Berger, P., Luckmann, T.: The social construction of reality: a treatise in the sociology of knowledge. Penguin, London (1966)
25. Levina, N.: Collaborating on Multiparty Information Systems Development Projects: A Collective Reflection-in-Action View. Information Systems Research 16, 109–130 (2005)
26. Klein, G.: Sources of Power: How People Make Decisions. The MIT Press, Cambridge (1999)
27. Ralph, P., Wand, Y.: A Teleological Process Theory of Software Development. In: JAIS Theory Development Workshop, Paris, France. Sprouts: Working Papers on Information Systems, vol. 8(23) (2008)
28. Ralph, P.: Theories of Software Design. MIS Quarterly (under review), 76 pages (2010)
29. Beck, K.: Extreme Programming eXplained: Embrace Change. Addison Wesley, Boston (2005)
30. Graham, P.: Hackers and Painters (2003)
31. Truex, D., Baskerville, R., Travis, J.: Amethodical systems development: the deferred meaning of systems development methods. Accounting, Management and Information Technologies 10, 53–79 (2000)
32. Singer, E.A.: Experience and Reflection. University of Pennsylvania Press, Philadelphia (1959)
33. Churchman, C.W.: The design of inquiring systems: Basic concepts of systems and organization. Basic Books, New York (1971)
34. Sober, E.: Testability. Proceedings and Addresses of the American Philosophical Association 73, 47–76 (1999)
35. Yin, R.: Case study research: Design and methods. Sage Publications, California (2003)
36. Wolfe, R.A.: Organizational innovation: review, critique and suggested research directions. Journal of Management Studies 31, 405–431 (1994)
37. Straub, D.W.: Validating Instruments in MIS Research. MIS Quarterly 13, 147–169 (1989)
38. DeVellis, R.: Scale development: Theory and applications. Sage, Thousand Oaks (2003)
39. Fowler, F.J.: Improving survey questions: Design and evaluation. Sage, Thousand Oaks (1995)
40. Harwell, M.R., Gatti, G.G.: Rescaling Ordinal Data to Interval Data in Educational Research. Review of Educational Research 71, 105–131 (2001)
41. Parnas, D.L., Clements, P.C.: A rational design process: How and why to fake it. IEEE Transactions on Software Engineering 12, 251–257 (1986)
42. Nandhakumar, J., Avison, D.: The Fiction of Methodological Development: A Field Study of Information Systems Development. Information Technology & People 12, 176–191 (1999)
43. Zheng, Y., Venters, W., Cornford, T.: Agility, Improvisation and Enacted Emergence. In: International Conference on Information Systems, Montreal, Canada (2007)
44. Dube, L., Pare, G.: Rigor in Information Systems Positivist Case Research: Currect Practices, Trends and Recommendations. MIS Quarterly 27, 597–635 (2003)

45. Eisenhardt, K.M.: Building Theories from Case Study Research. The Academy of Management Review 14, 532–550 (1989)
46. Lee, A.S.: A scientific methodology for MIS case studies. MIS Quarterly 13, 33–50 (1989)
47. Benbasat, I., Goldstein, D., Mead, M.: The case research strategy in studies of information systems. MIS Quarterly 11, 369–386 (1987)

## Appendix: Questionnaire Items

Participants were asked to respond to the following items on a 5 point agreement scale.

- No one thing drives all design decisions – they are made based on a variety of information
- Changes to my team's understanding of what the software is supposed to do were triggered by changes in our understanding of the problem/situation
- My understanding of what the software is supposed to do has been influenced by several factors (e.g., management, marketing, clients, the dev team, standards, my own values, experience on previous products, etc.)
- My understanding of the software's purpose has been influenced by several factors (e.g., management, marketing, clients, the dev team, standards, my own values, experience on previous products, etc.)
- The process of designing the software has NOT helped my team better understand the context in which the software is intended to be used
- A complete, correct specification of low-level design decisions was available before coding began (*e.g., whether to use a hashtable or array to store usernames)
- The software was coded iteratively
- My team has revised the software code based on new information (e.g., bug reports, failed unit tests, feedback from Quality Assurance, etc.)
- My team now understands what the software is supposed to do better than we did when we started coding
- Low-level design decisions* were primarily made before the first line of code was written (*e.g., whether to use a hashtable or array to store usernames)

Participants were asked to respond to the following items on a 5 point ranging from "Exclusively with models," to "Exclusively with code."

- I do detailed design...
- My team does detailed design...

Participants were asked to respond to the following item on a 5 point ranging from "Exclusively prediction," to "Exclusively observation."

Which of these is more consistent with how your team does testing? (Required)

1. Prediction: testers inspect models of the software and predict how code based on those models will behave (e.g., predict from a UML class diagram how the code will handle an error).
2. Observation: testers run the code and see what it does (e.g., unit testing, manually test the interface).

# The Ecology of Learning-by-Building: Bridging Design Science and Natural History of Knowledge

Marco De Marco, Renato Fiocca, and Francesca Ricciardi

Catholic University, Milan, Italy
francesca.ricciardi@unicatt.it

**Abstract.** A growing stream of research in Information Systems – Organizational Studies is focusing on Design Sciences, not only because a deep understanding of design processes is perceived as fundamental in order to enhance artifacts quality, but also because design activity is more and more understood as a powerful opportunity to create new knowledge. But what are the relationships between design as a pragmatic, problem-solving activity, and design as a learning activity? Moreover, what are the relationships between design as a learning activity, and "proper" scientific research? Even more importantly, under what conditions does successful learning-by-building more probably take place? Natural sciences, and Konrad Lorenz in particular, have given important contributions to answer these questions. This paper seeks to demonstrate how eco-evolutionary thought, which has been quite overlooked in our disciplinary field so far, could be useful to build an effective, multi-disciplinary, epistemologically sound basis to Design Sciences.

## 1 Introduction

Design teachers often start their courses highlighting the fact that the word “design” is both a noun and a verb; then, when we study design, we can study both a product (the artifact, be it tangible or not) and a process (the whole of activities concerning the construction of the artifact).

It is commonly accepted in Design Science scholarly community that a key characteristic of any artifact is *utility* [1], [2], [3]; as a consequence, design processes tend to be defined successful when they result in problem-solving artifacts.

But if we look at design from an evolutionary point of view, we see that utility is not the only characteristic that makes a certain artifact and/or a certain artifact building process relevant.

In fact, if we include the time dimension in our reflections about design, we become aware that every artifact, independently from its utility, has another feature: *embedded knowledge*.

For example, a sand castle, built by a kid, is useless; but for the pure fact that it exists, it hides in itself much more knowledge than many physics textbooks (and by the way, this is the main reason why children are so fond of sand castles). Every time we build something, we put that artifact in relationship with the world; and the world *answers*. For example, the kid’s castle can stand, or can collapse. But in *both* cases, the artifact “takes the form” the world gives it, just like a footprint; and this *taking the form*, in its complexity, is the latin *in-formatio*, i.e. gathering in-formation: learning.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 154–166, 2010.

Every artifact, in other words, travels through time as a sort of knowledge capsule: knowledge about the world remains "trapped" in the very form and structure of the design outcome, even far beyond the designer's intentions and awareness, even far beyond the designer's assumptions and hypotheses.

For example, sails have been invented thousand of years before Bernoulli formalized the principle that explains and predicts how sailing boats are pulled and sucked in (and not pushed, like one intuitively may think) by the wind. Nevertheless, the lack of understanding of the functioning principle had not prevented an impressive evolution of dozens of different, refined and highly efficient types of sails (and sailing boats, and sailing procedures, and so on).

Also eagles, like middle-ages sailor men, would be unable to answer questions about Bernoulli's equation; but they *know* how the air can hold them up during gliding flight. Their wings have "taken the *form* of the air", i.e. they have trapped in their very form and structure an astonishing amount of *in-formations* about gravity, atmosphere density, Bernoulli motions, hunting strategies, and thousands of other factors. In fact, an extra-terrestrial could gather a great amount of information about our planet, just examining the eagle's wings.

So, a certain amount of knowledge always remains trapped, embedded, within any object that has been tried against reality. This knowledge can be used even if it is not made explicit, even if we are not aware if it: in fact, the world "answers" to our attempt to create a new type of sail, even if we know nothing about Bernoulli equation. Similarly, the world "answers" to our attempt to create a new software or a new information system. In this way, long chains of artifacts can be created through time, each of them treasuring also all the knowledge, though implicit, that was trapped in the older objects of the series.

Every such object is a knowledge capsule, be it living (the eagle's wings) or not (the sail, or the information system). The process that makes this possible is *evolution*.

Evolutionary concepts have played an important role in the history of economic thought. The classical economic concept of competition, above all, has been usefully confronted with the biology-rooted concept of dynamic selection processes [4].

This led to a stream of studies that understands design as a selection-driven process [5] [6]. In these writings, biological organisms, organizations and technological artifacts are all seen as complex systems competing in a fitness landscape.

Such approaches have received an important formal framework by Kauffman' NK Systems modeling, [7], [8], [9], [10], which gives tools to simulate the evolution of technological artifacts, and particularly to analyze the interdependencies between the constituting elements of the systems.

Nevertheless, all this stream of studies, though rich and fertile, has shown little interest in many *learning implications* of designing and building.

For example, in these writings the main learning process which is taken into consideration is trial-error, in that it is the most important process in genotype – phenotype mutations; but other basic learning processes, even if importantly rooted in biological evolution, are not taken into consideration. Moreover, the paramount effects of emotional triggers, which have a fundamental role in learning by-doing, are not easily understandable within the NK modeling.

As a consequence, we think that the important learning opportunities of the design/building process are not fully understood in our discipline yet.

In Design Science research, and even more in design practice, a central role is played by engineering approaches, rooted in positivistic tradition [11], [12]. They prescribe to:

- Define a desired final status (i.e. what problem you want to solve with the artifact, and how). Justify your goal demonstrating the potential utility of the artifact.
- Use pre-existing, formalized and explicit knowledge to design the solution.
- Prepare formal plans and blueprints, and possibly prototypes, as precise and exhaustive as possible.
- Test and try plans, blueprints and prototypes.
- Adjust plans, blueprints and prototypes, until virtual testing is successful.
- Build the artifact.
- Systematically check the artifact's utility in its real-world context.
- Carefully report about all the previous phases.

This approach is sound and consistent [2], and has many merits: after all, without it we would not have the Tour Eiffel, or plasma TVs. But this approach gives us guidelines for a specific, explicit, *short-term* type of design process only; it does not describe the whole of design processes as they are in reality. Should we remove from history all the artifacts whose building processes took place without complying with the engineering prescription described above, we would not have antibiotics, parachutes, wheels, Brunelleschi's Dome in Florence, the World Wide Web, the information systems for flight reservations, and hundred of thousands of most important things.

This means that traditional engineering approaches provide good specific guidelines for short-term engineering activities (e.g. [12]), but they do not provide an effective, complete picture of the general conditions under which successful design may occur.

In this paper, we will try to integrate traditional, engineering-based approaches into a wider approach, which has demonstrated suitable to thoroughly understand also the "dark side" of artifacts, i.e. their embedded knowledge.

Ecological and evolutionary thought, in particular, has yielded important studies about the natural history of knowledge, that go far beyond the traditional "Darwin + DNA model" which is at the basis of NK approach. These studies cast light on many aspects of design that have been quite neglected so far. Konrad Lorenz is one of the beginners, and one of the most important scholars, within this research stream.

This paper, then, will seek to describe the contributions coming from Lorenz and from eco-evolutionary thought to answer the following three research questions:

1. *What are the relationships between design as a pragmatic, problem-solving activity, and design as a learning activity?*
2. *What are the relationships between design as a learning activity, and "proper" scientific research?*
3. *Under what conditions does successful learning-by-building more probably take place?*

A paragraph will be dedicated to each of these questions.

Before addressing them, however, we would like to say that we are aware that biology-rooted approaches are considered inappropriate in our discipline by some scholars.

Information systems and organizational studies, and design sciences in particular, are multi-disciplinary in nature: they borrow plenty of concepts, paradigms and frameworks from sociology, economics, psychology, engineering, architecture, anthropology and philosophy; but there is a sort of silent resistance against sciences of life.

Objections against the adoption of eco-evolutionary approaches in our disciplines could be summarized as follows:

a) An artifact is an artificial object [5]; it cannot be understood with the same tools developed to understand living organisms.
b) Designing is a socio-cultural process [33]: it cannot be understood with the same tools developed to understand biological processes.

Well, take beehives. They're objects built outside their creators' bodies, to serve for useful purposes. In fact, they are considered very interesting by researchers of the artificial, such as engineers, or architects. Is this a sufficient reason to say that beehives should not be studied also from an eco-evolutionary point of view? Do we really think that sciences of life have nothing interesting to say about these artifacts? Note that we are not saying that engineers or architects should not study beehives: we are just saying that *also* eco-evolutionary thought can give important contributions to understand them. One could reply that beehives are not like human artifacts, because they are built instinctively, without a proper, aware design process. But this is an out-of-date objection, based on the idea that what comes from nature and what comes from cultural evolution can be somehow, as Descartes did in his famous error, separated. On the contrary, also the most sophisticated human design activities are importantly managed and triggered *also* by innate knowledge: if our DNA did not provide us with hundreds of innate tools, such as the concept of cause-and-effect or the mood of boredom, we would be unable to build even a sling.

There is an unanimous complaint in our scholarly community, that Design Sciences Research, though a promising and exciting field, lacks a common language [14], [15], a comprehensive view [16], [17], [18], [12], an effective epistemological basis [19], [11]. Maybe the eco-evolutionary thought, and Lorenz's work in particular, could give a viable contribution to enhance our discipline's relevance and rigor.

## 2 Design as Problem-Solving Activity, and Design as Learning Activity

In our field, many scholars agree that designing can be described as a problem-solving process [2], [20], [1], [21], [22], [23], [24], [25].

There is also a good agreement among scholars on the fact that the design process is an interesting object of study and research (see for example [26], [27], [28]). This view is rooted in the four decades' study tradition of the Design Research Society [22].

On the other side, the idea that design process *itself* is a learning process, and specifically a research activity, is not as widely acknowledged and accepted as the two statements mentioned above. For example, Järvinen [29], taking into consideration information systems as artifacts, says: «We prefer systems development as an object of study rather than a research method (…) we cannot imagine which kind of knowledge systems development as a research method could produce» (p. 61).

This difficulty in understanding designing as a learning/research activity probably stems from a quite rigid interpretation of the dominating engineering approach mentioned in the previous paragraph. From an "engineering textbook" point of view, in fact, firstly you should perform basic (pure) research to understand reality, and only after gaining sound knowledge about the involved phenomena you should apply this knowledge to solve problems in the world. The use of pre-existing, already tested knowledge provides engineers with a good degree of certainty that the final artifact will work. In effect, we must admit that "I just wanted to try" would not be probably considered a good justification for an engineer whose experimental building collapsed. In other words, according to an "engineering textbook" approach, if the scientific community has worked properly before, there should be little left to learn, when you arrive to implementation: you just have to identify the type of problem, and pick the correspondent type of solution. Just pay attention not to make mistakes in calculations, and everything will be all right.

Sadly, things are not that simple, as every practitioner engineer knows [30]. *Previous knowledge* is almost never enough: every new design process is very unlikely to be successful, unless knowledge levels are further and further enhanced each time. Messy situations are most common, and creative problem solving is called into action [30], [31], [32]. The growing Interpretivist school, for example, highlights the multiplicity of interpretations that always underpin every design process [11], [33], [34], [35]. Nevertheless, a sort of implicit trust in the optimistic "engineer textbook" approach continues to flow under our discipline's foundations.

From an eco-evolutionary point of view, on the contrary, design and artifact building *always* imply grasping new knowledge. Every problem solving activity, in fact, is a learning process in itself. Even if the subject finds a solution instinctively, or without being aware of it, the pure fact that this solution works (or not) in that specific context provides knowledge accumulation.

According to Konrad Lorenz, in fact, the basic strategy of life, even in its simplest forms, is grasping and accumulating knowledge. Every organism, far from just passively react to the random inputs coming from the environment, has at its disposal a heritage of knowledge that allows it to snatch at the opportunities, and, by exploiting them, to rise the probability that lucky chances will occur again. Life is then, from the simplest beings to the most complex ones, an endless cognitive process, continuously matching with a changing environment.

Lorenz's thought about these themes develops through more than 40 years, from 1944 to his death (in 1989). He wanted to found «a natural history of human knowledge» (1973), and started writing about it when a war prisoner in Russia, in 1944. Lorenz's main texts about these themes are: *On Aggression* (1966); *Behind the Mirror* (1973); *Civilized Man's Eight Deadly Sins* (1974) and *The waning of Humaneness* (1983). [36], [37], [38], [39].

Among these writings, the most important as to our goals is *Behind the Mirror* [37]. The following concepts are a synthesis from that dense, complex book, although important reflections on the issues being addressed here are available also in the other writings.

According to Lorenz, there are five basic learning processes, which humans share with many other animals:

a) Identification (e.g. a rabbit realizes “there is a predator here”)
b) Imitation (e.g. young artisans imitate old and expert master’s behavior)
c) Training (e.g. a dancer feels compelled to repeat his or her movement until perfection is attained)
d) Trial-error (e.g. an ant searches its way towards food in a labyrinth)
e) Exploration (e.g. a rat wanders in the surroundings, without any specific need, “just to know”).

These five processes are made possible, triggered and managed by pre-existing knowledge, in the form of patterns and procedures. All animals in fact, humans included, are born with a rich wealth of innate patterns and behavioral programs, which constitute the *a priori* basis of every further learning activity. For example, animals can rely on innate patterns that let them recognize a predator, even if they have never seen an individual of that predatory species before. This is type a) performance (identification), which is at the very base of all the others. In a similar way, animals can have programs that trigger and manage learning processes b), c), d) and e). A cat locked in a case will try to escape through trial-error learning, until it finds the solution (e.g. it finds out how to open a hidden door), but its trials will not be random: the cat will “reasonably” seek to scrape the walls for example, it will not try to escape by licking its own foot, nor by closing an eye. Lorenz calls these programs, which route learning activities towards more probable success, “innate instructors”.

But innate instructors do not just route learning activities: they also strongly intervene after learning, facilitating the creation of new, more tailored *a-priori* knowledge.

*Trial-error* processes, for example, tend to create *habits*. If the cat finds the hidden door and succeed in escaping, when put in a similar cage some days after, will go on repeating the previously successful solution, even if the opening mechanism of the new cage is actually different. A “new” cat, not affected by the habit, will more probably find the solution. Similarly, *identification* processes easily tend to *generalization*: “there is a predator here” is often transformed, by innate instructors, into “this is a dangerous place”.

The following framework seeks to synthesize how, according to Lorenz, the five different basic learning processes tend to create new *a priori* knowledge, which may also supersede innate patterns and innate behavioral programs:

a) Identification → Generalizations
b) Imitation → Traditions
c) Training → Fluent, almost automatic sequences
d) Trial-error → Habits in problem solving
e) Exploration → Maps.

On this basis, learning processes can go on, in growing levels of abstraction. A good amount of innate and acquired patterns may allow, at a certain stage of evolution, *pattern linking*. Pattern linking underpins language, and can further activate the vast cultural heritage of generalizations, traditions, fluent sequences, habits and maps, enabling new cycles of ever growing abstraction.

In some cases, in fact, the animal does not directly interact with the real situation: instead, it stops in front of the problem, and *thinks before acting*. Lorenz quotes (1973, VII, 3) an experiment in which an ape (an orang) sees a banana hanging from the ceiling. A box is in the room’s corner. The ape looks at the situation, without moving and without trying to get the banana; scrapes its own head, in an evident concentration effort; it gets angry, because it can’t find a solution; tries to forget the problem, turning on the other side; then it looks again at the elements of the dilemma, and suddenly its expression changes. The ape makes a joyful somersault, and then runs to the box, drags it under the banana, climbs it and gets the fruit.

In this case, the ape does not make trials in the real space: it *imagines itself acting* in the spatial situation before its eyes, makes imaginary trials (for example: it imagines its own body “trying” to get the banana), recognizes imaginary errors (the banana is visibly too high) and goes on trying *in the mirror of its own mind*, until it makes an imaginary trial that meets an imaginary success. It is important to note that the ape *shows joy for the imaginary success in itself*, before having actually reached the banana. This means that animals can have an *emotional reward for imagining solutions*, which is independent, to some degree, from the solution’s real achievement. It is on this basis, Lorenz says, that complex and typically human learning activities (e.g. reading, meditation, discussion) take place. Design processes are thus generated as learning and problem-solving activities in an imaginary, virtual space.

This is a very particular strategy embedded in some animals’ DNA: the manipulation of mental models, instead of real objects, preserves animals from real errors’ unpleasant consequences (risks, and waste of energy), but it tends to make them lose possibilities related to unforeseeable, improvised solutions, that one could only find by “wasting” random trials in the real world. In other words, there are complex, latent trade-offs between efficiency and effectiveness when problem-solving strategies are found and tested in an imaginary space.

Trade-offs between problem solving strategies can be seen also from another point of view in Lorenz’s work. Progressive accumulation of *a priori* knowledge tends to result in a loss of flexibility. The cat of the experiment mentioned above became less able to open different cages, *because* the solution found in the first cage had become a tailored, but rigid, *a priori* instructor. Economic thought tends to see successes, incentives and rewards as always positive forces [45], whilst, in an eco-evolutionary perspective, success can also make us stupid. That's why "antibodies" should be kept in circulation during design processes, to continuously challenge the knowledge base stored in the deeper layers (i.e. in generalizations, traditions, fluent sequences, habits and maps). According to Lorenz, natural evolution has provided human beings with such antibodies, embedded in our emotional attitudes.

We try to synthesize Lorenz’s complex writings on this issue by saying that human beings are "natural born shifters": i.e. we tend to alternate, within individual life and within the social fabric, constructive attitudes (optimism, discipline, trust in existing beliefs, belonging feelings) and destructive attitudes (watchful pessimism, love for "the different ones", doubt, boredom). If this shifting is interrupted, the learning ecosystem becomes pathologic. For example, an organizational culture that imposes too much conformism and optimism sooner or later falls in what Lorenz calls the "technomorph thought", which is considered by the scholar as the most dangerous pathology of our age [38], [39]. An organization (or a society) blocked in technomorph thought, Lorenz says, is sewing the very straightjacket that will kill it.

## 3 Design as Learning Activity and Scientific Research

As we have seen in the previous paragraph, every problem solving activity implies testing something (e.g. one's own body, or mental innate patterns, or trained behaviors, or artifacts) against the world. This process generates knowledge, both in the case of success and failure. For example, if a predator's physical structure is unfit to hunt available preys, its species will automatically learn from its failure: its genome will be eliminated by death.

But the genome is not the only storing possibility. A certain amount of knowledge is stored in the nervous system, for example in the form of innate moods or of almost unaware habits, traditions or maps. Many times, moreover, knowledge is stored outside our body: embedded, often tacitly, in our artifacts' forms and structures. In other words, a huge amount of the knowledge heritage thanks to which we solve problems and survive is stored in an implicit, unaware form. This lets us widen our learning capacities enormously; if we could rely only on aware, explicit knowledge, we would die immediately, because we have extremely limited aware computational and storing capacities, with respect to our needs.

As a consequence, design processes have much more chances to result in successful artifacts if designers do not limit themselves to exploiting explicit knowledge. Leveraging also the "black box" of embedded, unaware knowledge hugely multiplies our storing and computational (and then learning and problem-solving) capabilities.

That's why practitioners dealing with real, on-field artifact building are instinctively annoyed if they are requested to do their job in a totally rigorous, scientific way. This is felt as counter-productive, and not without reason.

In fact, scientific research is precisely about this: making explicit. At a certain step of scientific research, unambiguous, explicit statements are needed. And the making-explicit strategy has a heavy price: it makes the sea of knowledge much more transparent and clear, but also much more shallow.

Moreover, whereas design is aimed at *building*, scientific approach works (also) through *destroying* efforts. To the designer, the sailing boat is a *problem-solving artifact*, and then it is successful only if it works. To the scientist, the sailing boat is an *experiment*, and it is particularly meaningful if it does not work: a failure in predictions is at the basis of Popper's falsification, and a fundamental step forward in scientific research [40].

As Einstein used to say, the only definite answer that a scientist may extract by nature is "no, your theory is wrong". An experiment is considered interesting by the scientific community if it is able to test predictions, i.e. if it allows both success *and failure* of tentative predictions. In this sense, science is destructive in nature.

So, what is the relationship between scientific research and designing processes? When we see a child building a tower of wooden cubes, or a sand castle, he or she is training his or her problem-solving abilities: we are watching a young designer in action. But, a minute later, that same child may be taken by a furious desire of removing a cube from the tower basis, or of pouring water over the sand castle. When the destroyer goes into action, the scent of science fills the air.

Information systems and organizational studies scholars have perceived such intrinsic trade-off between scientific and building attitudes. The long lasting debate about "relevance versus rigor" [2] in systems development is a direct reflection of this dilemma.

Should we conclude that a design process cannot be also a scientific research activity? W do not think so. The common element between science and design is the artifact itself. Also scientists, in fact, like designers, build something: typically, they build measure systems and artificial contexts, aimed at testing their theories. Every scientific experiment is an artifact: an artificial object designed to capture, to fix, to embed a treasure of knowledge. As a consequence, every artifact can be seen as an experiment, though often unaware, or not rigorous.

Design science scholars commonly say that it is *intention* that distinguishes the two fields [41]: if the aim is to solve problems, we are in the field of design; if the aim is to understand reality, we are in the field of science. But also this distinction risks to be taken simplistically. The two intentions often coexist, if we take into consideration a sufficiently long-term learning process.

Let's think, for example, of the well-known Franklin's work on the lighting rod. Like children building their sand castle, Franklin kept shifting between a scientific intention (i.e. studying electricity, which had been a very poorly understood phenomenon till then) and problem-solving efforts (i.e. protecting buildings from lightings) for months. This shifting did not prevent him from achieving success in both challenges: on the contrary, it was just his shifting attitude that made it possible.

In other words, from an eco-evolutionary point of view, learning by (also destructive) pattern-testing, for example by scientific research, and learning by building, for example by the design lifecycle, are two different attitudes. But this diversity should not confine us in a sort of epistemological apartheid: on the contrary, problem understanding and problem solving attitudes have co-evolved through thousands of millenniums, as powerful complementary capabilities - even more, as indispensable reciprocal antidotes.

## 4 Eco-evolutionary Strategies for Learning-by Building

The brief notes above give us the possibility to start drawing a picture of the conditions under which, from an eco-evolutionary point of view, human beings tend to be most effective artifact builders and to best exploit learning-by-building.

The following four strategies result in a research agenda, which should be further developed in field studies aimed at assessing and enhancing the capacity of organizations, networks and teams to develop successful learning-by-building.

1. A first strategy to achieve successful design and successful learning by building is completely plunging artifacts into reality for a long time, allowing the *"taking-the-form"* feedback. Vast, diverse and long-term exposure to reality results in knowledge embedding into one of the most effective knowledge storing devices, i.e. the artifact itself. The longer the period of exposure, and the more numerous and diverse the groups of users, the more knowledge-intensive the artifact will be. This knowledge may remain implicit, may not be fully perceived, but it remains there, available for future needs. That's why historic awareness is important in all design-centered disciplines. Architecture, for example, has a strong tradition in anthological education: future designers visit and thoroughly study the most important buildings of the past. Design Sciences in Information System and Organizational studies, on the contrary,

still lack a similar familiarity with the time dimension and with their cultural heritage [42], [43], [44].

*Research agenda: assess anthological and historic awareness of Information Systems/Organization Studies researchers and practitioners, and test how this awareness can be related to success/failure in artifact design processes.*

2. The whole range of basic learning activities (i.e. *identification, imitation, training, trial-error, exploration*) should have adequate space for developing during the design process.

   *Research agenda: assess how, and to what extent, the five basic learning activities take place in specific design processes, and test how viable identification, imitation, training, trial-error and exploration can be related to success/failure in artifact design processes.*

3. The design process should be founded on a rich basis of *a-priori* knowledge (*generalizations, traditions, fluent sequences, habits and maps*); moreover, further accumulation and sharing of generalizations, traditions, fluent sequences, habits and maps should be facilitated during design process.

   *Research agenda: assess the basis of a-priori knowledge in design contexts, and assess how it can be related to success/failure in artifact design processes. Assess how accumulating and sharing new a-priori knowledge (generalizations, traditions, fluent sequences, habits and maps) affects success/failure of designing organizations, networks or teams in the long run.*

4. The organization/team/network should be able to alternate constructive attitudes (optimism, discipline, trust in existing beliefs, belonging feelings) and destructive attitudes (watchful pessimism, love for "the different ones", doubt, boredom), to provide an appropriate level of soundness, on the one side, and flexibility, on the other side.

   *Research agenda: in design contexts, assess the level of rigidity provided by the accepted a-priori knowledge, along with the system of incentive-rewards-punishments. Assess how, and to what extent, the organization/team/network gives space to what has described above as "destructive attitudes". Assess the "shifting strategy" of the organization/team/network in the long run (this research agenda probably requires an interpretivist/ethnographic approach to focus on the non-quantitative, emotional aspects of the process). Assess how shifting strategies affect success/failure of designing organizations, networks or teams in the long run.*

## 5 Conclusions

If we try to stop thinking about artifacts as if they were completely artificial objects, and start considering them as *prostheses* of our evolving bodies and minds, a whole world of rich new views opens in front of our eyes.

The first thing that becomes evident, as soon as we accept this multi-disciplinary approach, is that utility is not the only crucial characteristic of artifacts; they are also, just like the eagle's wings, extraordinary capsules of embedded knowledge, traveling and evolving throughout time.

Moreover, such an approach casts a fascinating light on the double-way relationship between the so-called pure scientific research, and design processes.

Following Lorenz's biological studies, we can find new tools to study several aspects of learning-by-building: these tools are particularly interesting in our opinion, because they take into consideration that knowledge (and knowledge-by-building is not an exception) does not evolve linearly, but zigzagging, with recurrent, continuous shifts between different learning attitudes; and these shifts are essentially *emotional* in nature.

Like a sailing boat coursing against the wind, on a bowline, a healthy learning system, be it an individual, an organization or a society, adopts a *tacking strategy*, alternating constructive optimism and destructive doubts, conformism and rebellion, rituality and humor. This continuous shifting is a most refined strategy, evolved in thousands of millenniums, to manage the important trade-offs that intrinsically affect problem-solving processes. For example:

- Manipulating reality vs. manipulating models;
- Long-term, low-cost, semi-automatic knowledge (procedures, traditions, habits…), vs. short-term, flexible, high-cost knowledge (improvisation, creativity…);
- Building useful artifacts vs. destroying wrong (though maybe useful) beliefs.

All these issues lie at the very core of design disciplines. We hope that this brief contribution may trigger a thorough multi-disciplinary debate.

## References

1. March, S.T., Smith, G.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)
2. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
3. March, S., Storey, V.C. (eds.): MIS Quarterly, Special Issue on Design Science (September 2008)
4. Alchian, A.A.: Uncertainty, evolution, and economic theory. Journal of Political Economy 58, 211–221 (1950)
5. Simon, H.: The Sciences of the Artificial. MIT Press, Cambridge (1969)
6. Nelson, R.R., Winter, S.G.: An Evolutionary Theory of Economic Change. Belknap Press of Harvard University Press, Cambridge (1982)
7. Kauffman, S.A.: Adaptation on rugged fitness landscapes. In: Stein, D.L. (ed.) Lectures in the Sciences of Complexity, pp. 527–618. Addison-Wesley, Redwood City (1989)
8. Kauffman, S.A.: The Origins of Order. Self-Organization and Selection in Evolution. Oxford University Press, Oxford (1993)
9. Kauffman, S.A., Lobo, J., Macready, W.G.: Optimal search on a technology landscape. Journal of Economic Behavior and Organization 43, 141–166 (2000)
10. Kauffman, S.A., Macready, W.G.: Technological evolution and adaptive organizations. Complexity 1, 26–43 (1995)
11. Niehaves, B.: On Episemological Diversity in Design Science: New Vistas for a Design-Oriented IS Research? In: Proceedings of the International Conference on Information Systems (ICIS), paper 133 (2007)
12. McKay, J., Marshall, P.: A Review of Design Science in Information Systems. In: Australasian Conference on Information Systems (ACIS 2005), Sydney (2005)

13. Nunamaker, J., Chen, M., Purdin, T.: System Development in Information Systems Research. Journal of Management Information Systems 7(3), 89–106 (1991)
14. Purao, S.: Design Research in the Technology of Information Systems: Truth or Dare. GSU Department of CIS Working Paper. Atlanta (2002)
15. Indulska, M., Recker, J.C.: Design Science in IS Research: A Literature Analysis. In: Gregor, S., Ho, S. (eds.) Proceedings 4th Biennial ANU Workshop on Information Systems Foundations, Canberra, Australia (2008)
16. Purao, S.: The Sciences of Design: Observations from an Emerging Field. Communications of the Association for Information Systems, vol. 23, art. 29 (2008)
17. Love, T.: Constructing a Coherent Cross-Disciplinary Body of Theory about Designing and Designs: Some Philosophical Issues. International Journal of Design Studies 23(3), 345–361 (2002)
18. Iivari, J.: A Paradigmatic Analysis of Information Systems as a Design Science. Scandinavian Journal of Information Systems 19(2), 39–64 (2007)
19. Gregg, D., Kulkarni, U., Vinze, A.: Understanding the Philosophical Underpinnings of Software Engineering Research in Information Systems. Information Systems Frontiers 3(2), 169–183 (2001)
20. Lee, A.S.: Systems Thinking, Design Science, and Paradigms. Heeding Three Lessons from the Past to Resolve Three Dilemmas in the Present to Direct a Trajectory for Future Research in the Information Systems Field (Keynote Speech). In: 11th International Conference on Information Management, Kaohsiung, Taiwan (2000)
21. 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)
22. Au, Y.A.: Design Science I: The Role of Design Science in Electronic Commerce Research. Communications of the AIS (7) (2001)
23. Boland, R.J., Day, W.F.: The Experience of System Design: A Hermeneutic of Organizational Action. Scandinavian Journal of Management 5(2), 87–104 (1989)
24. Rossi, M., Sein, M.: Design Research Workshop: A Proactive Research Approach. In: IRIS 2003, Helsinki (2003)
25. Markus, M.L., Majchrzak, A., Gasser, L.: A Design Theory for Systems that Support Emergent Knowledge Processes. MIS Quarterly 26(3), 179–212 (2002)
26. Archer, L.B.: Whatever Became of Design Methodology? Design Studies 1(1), 17–20 (1979)
27. Cross, N.: Designerly Ways of Knowing. Birkhäuser, Basel (2007)
28. Cross, N., Christiaans, H., Dorst., K. (eds.): Analysing Design Activity. Wiley Publications, Hoboken (1997)
29. Järvinen, P.: On design Methods. Opinpaja Oy. Tampere, Finland (1999)
30. Schön, D.A.: The Reflective Practitioner. Basic Books, New York (1983)
31. Cross, N.: Design/Science/Research: Developing a Discipline. In: International Symposium on the 5th Asian Design Conference, Seoul, Korea (2001)
32. Baskerville, R., Wood-Harper, A.T.: Diversity in information systems action research methods. European Journal of Information Systems 7(2), 90–107 (1998)
33. Klein, H.K., Myers, M.D.: A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly 23(1), 67–94 (1999)
34. Venable, J.R.: A Framework for Design Science Research Activities. In: Proceedings of the Information Resource Management Association Conference, Washington, DC, USA, May 21-24 (2006)

35. Venable, J.R., Travis, J.: Using a Group Support System for the Distributed Application of Soft Systems Methodology. In: Hope, B., Yoong, P. (eds.) Proceedings of the 10th Australasian Conference in Information Systems, Wellington, New Zealand, December 1-3, pp. 1105–1117 (1999)
36. Lorenz, K.: On Aggression, Harcourt. Brace and World, New York (1966)
37. Lorenz, K.: Behind the Mirror. A search for a natural history of human knowledge. Harcourt Brace, New York (1973)
38. Lorenz, K.: Civilized Man's Eight Deadly Sins. Harcourt Brace, New York (1974)
39. Lorenz, K.: The Waning of Humaneness. Little, Brown & Co., Boston (1983)
40. Dey, D., Sarkar., S.: Modifications of Uncertain Data: A Bayesian Framework for Belief Revision. Information Systems Research 11(1), 1–16 (2000)
41. Nelson, H., Stolterman, E.: The Design Way—Intentional Change in an Unpredictable World, vol. 544 Volume 23 Article 29. Educational Technology Publications, New Jersey (2003)
42. Batra, D., Davis, J.: Conceptual Data Modeling in Database Design: Similarities and Differences between Expert and Novice Designers. International Journal of Man-Machine Studies 37, 83–101 (1992)
43. Oxman, R.E.: Precedents in Design: A Computational Model for the Organization of Precedent Knowledge. Design Studies 15(2), 141–157 (1994)
44. Purao, S., Han, T., Storey, V.C.: Improving Reuse-based Design: Augmenting Analysis Patterns Reuse with Learning. Information Systems Research 14(3), 269–290 (2003)
45. Baldwin, C.Y., Clark, K.B.: Design Rules. The Power of Modularity, vol. 1. MIT Press, Cambridge (2000)

# On Computer Simulation as a Component in Information Systems Research

Sebastian Hudert, Christoph Niemann, and Torsten Eymann

Chair of Information Systems Management
University of Bayreuth
95440 Bayreuth, Germany

**Abstract.** Computer simulation is widely regarded as a useful activity during various phases of research. However, depending on its context, the meaning, definition, and focus of the term can vary: While in traffic planning, for example, simulation is used to determine useful configurations of a road network and thus focuses on the environment, there is an entirely different perspective of simulations when used within multi-agent systems. In such settings, the environment of the agents remains static, while the interesting research questions concern the behavior of the agents themselves. The research focuses on the microscopic level and resulting emergent behavior. This article addresses the different meanings of simulation and puts them in the context of a research process that treats descriptive and prescriptive research as two sides of the same coin. Building on this abstract research process, we develop a framework to classify different types of simulation, based on the actual research activity they are intended to be used for. This framework can thus serve subsequently as a guideline on the usage of computer simulation as a research tool.

## 1 Introduction and Motivation

Simulation has always been a part of different kinds of research processes in varying research disciplines. However, there exists no common definition of this term in the different areas of information systems research (ISR). On the one hand, it is used to describe a method for evaluation of scientific work; on the other hand, it is used as part of the theory building process. This broad range of meanings makes the term simulation prone to misunderstandings – especially in the communication between researchers using different methods.

To shed a light on the different uses of simulation, this paper frames simulation in a research process for ISR. The process has been distilled from a literature review and tries to combine descriptive and prescriptive research into one consolidated model. It identifies four main activities in the research cycle and explains their relationships. Simulation can be used in almost any of these steps but with different purposes and implications.

What is of particular importance here is the object of simulation which can be either the environment (in which real entities are being watched) or it can

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 167–179, 2010.

be the entity itself (e. g. a limited prototype of a software system) that is put in a real environment. The third possibility is the simulation of an abstract entity, such as a formal model, in a simulated environment. This is the setting that predominates most scientific simulation efforts within ISR, as the simulation takes place in a computer most of the time. This case is thus the main focus of this article. Furthermore, we explicitly exclude from our investigation simulations with real entities in real environments as it is used in biology for instance. In ISR, such a setting would be the development of a prototype and thus a different research method.

This leads to the main research question of this paper: How can simulation be applied to different phases within the research cycle? For that purpose, we employ a categorization of simulation situations proposed by Hartmann [8] and map the different simulation types to the various phases of the research process developed by March and Smith [13]. Since not all possible combinations are useful, with this we hope to provide a structured guideline on when to use what type of simulation, given a specific research question at hand.

The remainder of the paper is structured as follows: After the motivation, Section 2 introduces the consolidated research process. Section 3 starts with an overview on different uses of the term simulation in research. It puts the different meanings into the context of the research process and the article concludes with our actual guideline framework for the usage of simulation as a research tool in Subsection 3.2. We finish our paper with a short summary and description of future work.

## 2 Research Process

While ISR has been dominated historically by descriptive research, in recent years there is a growing trend to integrate Design Science (DS) into ISR to generate a more holistic view of research as such. This stream of research has been started by Herbert Simon in his influential book “The Sciences of the Artificial” [17]. He calls for a design oriented approach to research that does not aim to *explain* the environment, but rather to *improve* it. DS oriented research produces artifacts that serve a distinct purpose. Such artifacts should be evaluated on their utility instead of on their explanatory power.

One reason to integrate DS into ISR is the claim for relevant (and not only rigorous) research. ISR has always aimed for rigorous research processes but might have neglected the relevance of the research issues [7, p. 26]. Relevance, on the other hand, has been the traditional strength of DS, as indicated continuously by large amounts of industry funds or by the stable demand of graduates of the German “Wirtschaftsinformatik” that focuses on the DS approach [7, p. 5].

The duality of research approaches seems to foster the “rigor vs. relevance” debate that continues to surface in ISR. However, different authors claim that research can be rigorous *and* relevant [7,14,15]. Indeed, ISR not only can but should satisfy both requirements at the same time [1, p. 223]. Based on this assumption, the two research approaches should be integrated in such a way that both profit from one another. March and Smith developed a view on ISR that sees both approaches as structurally equivalent on an abstract level: Descriptive research (and natural sciences in general) consists of two activities. A theory has to be developed *(theorize)* and *justified* [13, p. 255]. They draw on earlier work by Kaplan that uses the same two activities but calls them *discovery* and *justification* [12, p. 14].

DS is based on two activities as well. A researcher has to *build* an artifact that improves the environment. Subsequently, to prove that the research has been effective, the artifact needs to be *evaluated* [13,9]. On this level of abstraction, the different aims of the approaches (truth or utility) are irrelevant: Both approaches try to either discover or to develop something new. Subsequently, this new entity has to prove its value in explanatory power or utility respectively.

Figure 1 shows the four activities in research. It combines them with Hevner's view that both approaches complement each other [9, p. 98]. The top half of the research cycle displays descriptive research, the bottom half shows DS with the corresponding activities.

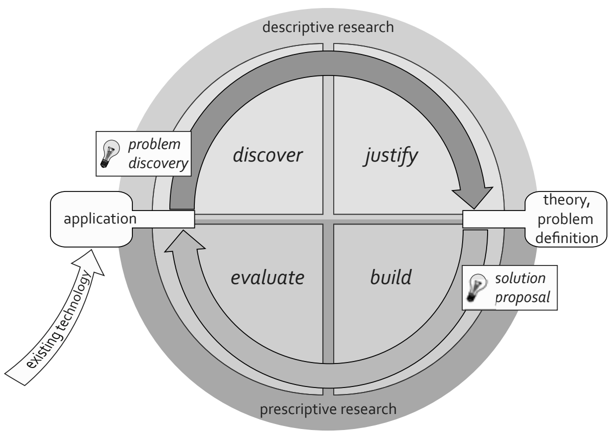



**Fig. 1.** Consolidated Research Process

The cycle has two potential starting points, depending on the research question to be addressed. If a researcher seeks to explain an observed phenomenon, the entry point is on the left hand side. He starts with an existing technology and discovers a behavior that has no explanation yet. In the discovery phase, he develops a hypothetical theory that could explain the phenomenon. To obtain a useful theory, it needs to be justified. All explanations and predictions of the theory must be consistent with the empirical findings. If the theory remains unrefuted by the available empirical findings it can be used as a "tentative theory". In line with Popper's philosophy of science, it remains "tentative" [16, p. 280] because it can still be refuted by empirical findings. This test of the theory occurs in the justification phase.

The second potential entry point is on the right hand side. If the research question aims to solve a problem [9, p. 78], the research is design oriented. Building on the theoretical foundations that have been generated by descriptive research, a researcher implements an artifact. After the implementation, the artifact must be evaluated. Only if it provides greater utility than other, existing artifacts, it is useful.

One challenge with the evaluation in DS is the choice of metrics that provide a useful measurement of the utility of the artifact. In cases, where a hitherto not addressed research problem is considered, the "research contribution lies in the novelty of the artifact and in the persuasiveness of the claims that it is effective" [13, p. 260].

A more general position is taken by van Aken [1] who states that not the artifacts themselves but rather *rules* that can be deduced from the artifacts are useful results [1, p. 227]. It is not the very instance of a particular solution to a problem but generalized rules that can be applied elsewhere as well that advance the state of knowledge.

To complete the cycle, an evaluated artifact can be the source of new descriptive research. The consolidated research process thus integrates descriptive as well as prescriptive research and allows for entry points using both methods. The choice of methods is based on the research problem to be addressed: If its aim is to explain something, it starts on the left hand side of the cycle and uses the two activities of descriptive research. Research that tries to improve the environment with a novel artifact uses the DS part of the cycle. It starts on the right hand side and uses the two activities of prescriptive research. The research process thus provides a dynamic view on research. The next section introduces simulations seen from a static perspective and classifies them based on a literature review.

## 3 Simulation as Component of the Research Process

To put the different uses of simulation in the proper positions in the cycle, we draw on classification schemes that identify different classes of simulation. The integration of the dynamic and the static perspective is done on the level of these classes.

### 3.1 State of the Art

Before defining our actual methodical framework for simulation-based research, we will now give a broad overview on how simulation is perceived as a tool in the research world. To this end we will present which categorizations for simulation processes in research and industry are available at the moment, acting as a fundamental basis for our own work.

In order to understand simulation as a research tool it is necessary first to derive a comprehensive definition of this concept. A quite prominent definition originates in the social sciences: Bratley et al. define a simulation to be a process of "driving a model of a system with suitable inputs and observing its corresponding outputs" [3] They very much follow the pragmatic view of Dooley who argues that simulations should answer a scientific question in the form of "What if?" instead of traditional research tools concentrating on "What happened and how and why?" [5, p. 829]. Humphreys on the other hand regards (computer) simulations simply as a computer-based "solution method for mathematical models where analytic methods are ... unavailable" [10, p. 502]. While this stresses the necessity for mathematical models it also restricts simulation to those cases where analytical methods do not work any more.

In the light of those considerations the most comprehensive definition of simulation was proposed by Hartmann: "simulation imitates one process by another process. In this definition, the term 'process' refers solely to some object or system whose state changes in time. If the simulation is run on a computer, it is called a computer simulation" [8, p. 82]. We will use this definition for the remainder of this paper as it provides as a very generic definition capable of coping with the whole research process underlying our work.

The basic definitions above already show a very tight link between simulation as a tool for scientific investigations and theoretical or mathematical models. According to Bunge [4], such models consist of a general theory as a conceptual foundation and a special description of an object or system (model object). The general theory provides fundamental rules describing the context of the scientific investigation (examples would be very comprehensive theories such as the theory of relativity). On the other hand, the model object defines an abstract description of a given object or process in the light of the underlying theory. Hartmann uses the Billiard Ball Model of gas as an example for a model object building on the Newtonian Mechanics as underlying theory. Scientists employ such models (mostly generated by abstraction [6]) for investigation or communication of single aspects of a formerly very complex real-world system. Following the rationale above, simulation therefore constitutes the process of testing or investigating such a theoretical model by observing its development or performance over time, given a set of input parameters. It thus "imitate[s] the time-evolution of a real system" [8, p. 82].

Several researchers involved in different scientific communities working with simulation techniques (mostly social and natural sciences as well as research theory) identified ways for further categorizing these approaches.

The first dimension used for distinguishing simulation processes is the purpose of the actual simulation runs, i. e. the goals of the respective scientific investigations. The two main taxonomies were proposed by Hartmann [8] and Axelrod [2]. Due to its more comprehensive nature, we will build on Hartmann's taxonomy for the remainder of this paper. In the following, we will present his categorization in more detail, which in turn builds on Axelrod's original work.

Hartman identified simulation to be one of the following [8]:

1. a *technique* – for investigating the detailed dynamics of a system
2. a *heuristic tool* – for redefining or developing hypotheses, models or even theories
3. a *substitute for an experiment* – for the execution of numerical experiments

4. a *tool for experimentalists* – for supporting or calibrating actual laboratory experiments
5. a *pedagogical tool* – for explaining a given process

*Simulation as a technique* helps to understand a given system's evolution over time [8]. Especially very complex systems often render it impossible to develop analytical solutions to research questions on a system's behavior. In contrast to often applied approximation methods in analytical tools, simulation provides researchers with a possibility to investigate the complete mathematical model without wiping out extreme values etc. This allows even for the testing of the underlying simulation model or theory [8]. This category roughly corresponds to Axelrod's sixth purpose of simulation: *proof* [2].

*Simulation as a heuristic tool* can play an important role in "developing hypotheses, models or even theories" [8, p. 86]. Based on the data generated by simulation runs, researchers can identify new and simple regularities eventually leading to the formulation of new hypotheses and theories. Axelrod describes this flavor of simulation as *discovery*-oriented [2].

*Simulation as a substitute for an experiment* can be invaluably important in situations where researchers want to explore settings "that cannot (yet?) be investigated ... by experimental means" [8, p. 87], due to pragmatic (e. g. investigations on fluid behavior in the core of the sun), theoretic (e. g. what-if questions on different values for natural constants) or ethical reasons [8]. To this end, simulations can be a quite accurate tool for the prediction of future system behavior, given the underlying assumptions hold. An example for this use of simulation is the weather forecast that relies on simulation. This is analogous to Axelrod's *prediction* class of simulations [2].

*Simulation as a tool for experimentalists* allows researchers, especially in the natural sciences, in inspiring them for new experiments, pre-selecting possible experiment setups (especially important in cases of very high experiment costs) or analyzing experiments (identifying statistical noise to be subtracted from the results) [8].

*Simulation as a pedagogical tool* finally stresses its potential for "instructing students ... by playing with a simulation model and visualizing [its] results on a screen." [8, p. 87]. Axelrod elaborates a little more on this aspect as he further distinguishes pedagogical simulations for *training*, *entertainment* and *education* uses [2].

A second dimension for categorizing simulation processes is the way the actual simulation runs are conducted. In this dimension, mainly two distinct classes of simulations are present: Simulations are either building on *continuous* or *discrete* models. For the former "the underlying space-time structure as well as the set of possible states of the system is assumed to be continuous" [8, p. 83]. Sometimes such models are also called *system dynamics* [5]. On the other hand, discrete simulations build on "a discrete space-time structure right from the beginning" [18]. "Moreover, the set of possible states of the system is assumed to be discrete" [8, p. 83].

Some researchers, mainly originating in the social sciences, introduced a third category: *agent-based simulations*. In such simulations individual software agents [11] act and react within a virtual world in order to "maximize their fitness (utility) functions by interacting with other agents and resources" [5, p. 829]. Agent-based models take decentralized view on a given system by trying to capture the individual utility functions and behavioral schemata for each entity (agent) in the virtual world and simulate the global system states emerging from those individual actors.

On a more theoretical note, simulation can be viewed in the light of scientific theory. To this end, Axelrod regards simulation processes as a third way of doing research apart from the traditional methods of induction and deduction. While induction is known as "the discovery of patterns in empirical data" [2, p. 24], deduction tries to define a set of abstract axioms for a given field along with a set of "proving consequences that can be derived from those assumptions" [2, p. 25]. Simulation is thus viewed as similar to deduction as it also starts with the development of explicit assumptions (i. e. models). However, it does not directly prove some theorem, but generates structured data as a result of the simulation that can in turn be analyzed as in inductive disciplines [2]. This view again shows how extremely important the simulated models are, as all results inducted from the simulation data depend on the validity of the simulated model.

These concepts lead to an interesting aspect of simulation research in the literature. Simulation as a research tool is mainly investigated in the context of social or natural sciences following the descriptive research paradigm. Simulations are used in these communities for investigating a given analytical model of a real world system. A completely other type of simulation is applied in engineering or computer sciences following a design oriented research approach. In such endeavors, the main concern is to create a simulated contextual environment in order to evaluate given characteristics of a developed artifact. A very prominent example for such a simulation process is the well-known wind tunnel in which new cars or airplanes are tested for their aerodynamic characteristics. This aspect will be central to the discussions within the next section.

All the categorization approaches above mark a valuable input for our theoretical framework presented in the next subsection. However, they focus on a static perspective on computer simulations whereas we intend to provide a process-based view on simulation in research. Thus our framework augments current work with an additional dynamic perspective.

### 3.2 A Guideline Framework for the Usage of Simulation as a Research Tool

As shown in the previous sections, numerous researchers have discussed both research theory and simulation as a research tool or method. Surprisingly very little discussions can be found on how these two areas link, i. e. in what phases of the research cycle simulation can be used and for what purpose. This section aims at closing that methodological gap by relating the different categories of computer simulations to the four fundamental phases in research: *theorize*, *justify*, *design* and *evaluate*. To this end, each of the different simulation types is investigated in the following and subsequently linked to the respective research phases, to which it can contribute.

When looking at simulation literature it becomes quite obvious that most of the discussions do not comprise the prescriptive branch of the research cycle. Being the oldest, and thus traditional research approach, all discussions tend to circle around descriptive sciences of one form or the other. Nevertheless, following the rationale from Section 2, we think such approaches take on a perspective too narrow as they basically omit just about half of the overall research cycle. Although it is understandable that the descriptive sciences, representing the traditional research approach, receive the most attention, we deem it absolutely crucial also for prescriptive researchers to critically reflect the research tools at their disposal. Simulation is but one example for such a tool.

In the following, we first try to explicitly relate the identified simulation approaches with the two descriptive research phases. In a second step, we attempt to port these views to the prescriptive side. We try to show perspectives on using simulation as part of prescriptive research and give researchers following that paradigm access to the powerful tool computer simulation.

Table 1 gives an overview on which types of simulation can sensibly be used in which of the four research phases:

The first of Hartmann's categories is *simulation as a technique*. The purpose of such simulations is to give a researcher a more profound knowledge of the internal dynamics of a system, ultimately providing him with a tool for the confirmation or disconfirmation of a theory under investigation. This type of simulations can be related to a distinct research phase very easily, as its purpose is basically congruent with the definition of the *justify* phase. In this step a researcher tries to confirm or even prove a given theory accordingly. On the other hand, simulations supporting the proof of a given theory or hypothesis are not actually suitable for discovering new hypotheses. Consequently, this table cell remains empty.

However, Hartmann also refers to a type of simulation aiming at exactly the task undertaken in the *theorize* phase: *simulation as a heuristic tool*. In such simulations, new patterns or regularities are sought in the data produced by the simulation of a given theory and respective model. Based on these patterns researchers are subsequently trying to generate new hypotheses or theories on the simulated systems. This matches the task undertaken in the *theorize* phase whose sole purpose is to define new hypotheses or theories to be confirmed or falsified in subsequent steps. Following the rationale above, *simulations as heuristic tools* are not suitable for the *justify* phase.

*Simulations as a substitute for experiments* are closely related to Hartmann's first category described above. They also ultimately aim at confirming or disconfirming a theory using simulations, in this case as a substitute for a laboratory

**Table 1.** Simulation purposes for different purposes and different phases in the research process

| | **Theorize** | | **Justify** |
|---|---|---|---|
| **Technique** | — | — | Gaining understanding and finally confirmation of theories and models |
| **Heuristic** | Discovery of new models, hypotheses and theories | — | — |
| **Substitute** | — | — | Confirmation or disconfirmation of theories |
| **Tool** | — | Identification of experiments for evaluation | — |
| **Pedagogical Use** | — | — | — |

| | **Build** | | **Evaluate** |
|---|---|---|---|
| **Technique** | Gaining understanding of used artifacts and grounding the design on proven theories | — | — |
| **Heuristic** | — | — | — |
| **Substitute** | — | — | Evaluation of the artifact |
| **Tool** | — | Identification of experiments for evaluation | — |
| **Pedagogical Use** | — | — | — |

experiment. The main difference is probably that the mere goal of gaining understanding of the dynamics of the model is not as prominent as with *simulations as a technique*. Following the rationale from above *simulations as a substitute for experiments* are also not really suitable for discovering theories as desired in the first phase of the research cycle.

The last class, *simulations as a tool for experimentalists*, is hard to relate to just one phase of the research cycle. On the one hand, such simulations directly relate to the developed theories (*theorize* phase) when inspiring new experiments focusing on the theory-relevant aspects of a system, on the other hand they help to set up further experiments, which in turn aim at proving or at least confirming a theory (*justify* phase). To this end, we locate this type of simulation in between

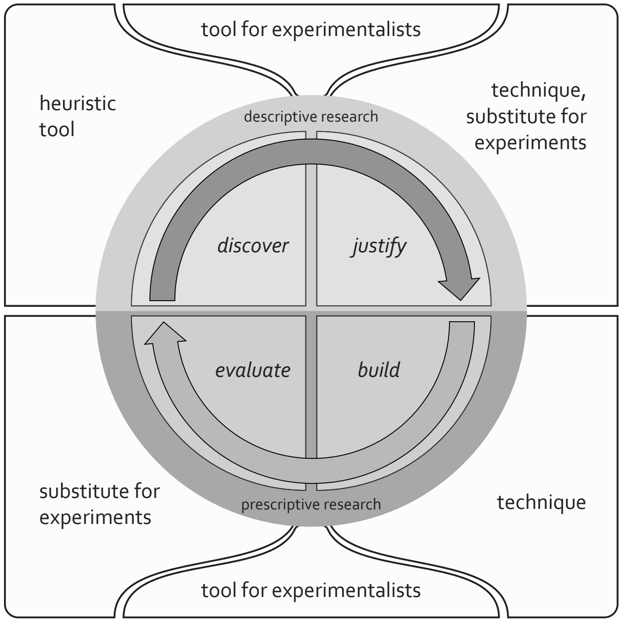



**Fig. 2.** Purpose of simulation applied to the research process

both descriptive research phases as they are basically applied when researchers take a theory developed in the first phase and try to design experiments to be conducted in the second. Figure 2 applies the different purposes to the research cycle and shows the usage during the four phases.

In his categorization of simulation, Hartmann does not include the DS oriented research in his approach. Consequently, he does not map any of the the five different flavors to DS activities. However, a mapping seems possible if one takes Hartmann's descriptions of the various purposes into account.

Even Hartmann did not mention (DS) artifacts while defining the description of *simulation as a technique*, we view simulation as a valid tool within the *build* phase of DS. Fundamental theories and models need to be integral parts of any artifact (and thus of the proposed solution to the research problem) to make a valid research contribution. In particular if the artifact makes use of different theories, a researcher needs to gain an understanding of their relation before the mix of theories can be applied to actually implement an artifact. Another, closely related application of *simulation as a technique* is the understanding of the interplay of different components that add up to the final artifact. The interaction of the sub-systems [8, p. 7] must be understood if a useful artifact is to be constructed. In a simulation the theories can be tested in different settings and improve the understanding of the interaction. Since a "simulation is no better than the assumptions build into it" [17, p. 14], the grounding of the used theories and models is indeed crucial to generate a valid research contribution.

As such, Hartmann's first purpose represents a valid use of simulation to provide input for the *build* phase.

*Simulations as a heuristic tool* is a purpose, which is not applicable to DS research. Hartmann states the aim of the purpose as the development of new "hypotheses, models or even new theories" [8, p. 85]. Simulations are a tool to find new regularities in potentially interesting settings that a new model (or theory) could explain. In DS, however, research starts with a narrowly defined question that shall be solved. Therefore, the second purpose of simulation is used in descriptive research only.

One of the major uses of simulation in DS is the *substitute for an experiment*. DS cannot stop after the implementation of a new artifact, but must evaluate it regarding its utility afterwards. Without rigorous evaluation, the artifact may be useful but does not contribute to the advancement of science [1, p. 229]. One way of evaluating an artifact is the construction of a prototype that is placed in the real environment. If it works as expected and solves the problem (either for the first time or better than any existing solution) it is a useful artifact. The definition of the term "better" and "useful" in the evaluation of new artifacts is domain specific and must be proposed by the researcher or (if there are artifacts already) is provided exogenously. In both cases, the artifact is measured on metrics that are context bound: The artifact does not aim for general truth, but for utility in a given situation.

The development of a prototype, however, can be restricted or even be impossible for the same three reasons that form the basis of the use of simulation as a *substitute for experiments* in descriptive research: It may be *theoretically*, *ethically* or *pragmatically impossible* to conduct a real experiment. If either one of these reasons holds, simulation can be a valid means to evaluate an artifact in DS.

The evaluation in DS remains a subject of debate. If an artifact solves an hitherto unsolved problem [9, p. 78], the metrics for evaluation must be proposed as well. Once they are fixed, the researcher must devise experiments that can be used to measure the degree of performance regarding the new metrics. Simulation can be a means for the inspiration of new experiments, analogous to descriptive research efforts, or of the range of system setups [8]. Another reason for the use of simulations as a tool prior of the evaluation is the identification of "trivial or well-understood" [8, p. 87] effects that prescind the attention from the interesting results. Simulating the actual experiment can determine and quantify such effects. Their measurements are subtracted from the results of the experiment to account for uninteresting "noise" that stems from those effects.

The last assertion that can be made on simulations as a research tool is that Hartmann's fifth flavor, simulation as a pedagogical tool, is not really relevant for an investigation on research methods. While such simulations are extremely valuable for the training of students and researchers new to the field of investigation, they play no significant role once actual research questions are addressed.

## 4 Conclusion and Outlook

This article has presented a consolidated research process that includes both descriptive and prescriptive approaches in ISR, pointing out that the researchers in ISR are not bound to one of the two, but can still choose their method depending on the research problem at hand.

However, while ISR can make use of descriptive as well as prescriptive research (and indeed it should include both approaches), a particular research problem should still be addressed with the suitable approach to yield reasonable results. This decision can be made based on the purpose of the actual research problem: If it aims to explain a phenomenon, the research is descriptive and should employ the corresponding activities. If the research aims for improvement of the environment and is evaluated based on utility (with a suitable definition of utility), it is prescriptive and should incorporate the two activities *build* and *evaluate*.

The research cycle thus provides a process based view on ISR activity acting as a fundamental basis for our guideline framework which tries to link different classes of computer simulations to the right activities within the cycle. Consequently, Hartmann's classification of simulations, obviously having descriptive research in mind, is extended analogously to prescriptive research phases. In doing so, we identified that different classes of simulation correspond to distinct activities within the research cycle. Only if the right types of simulation are used during the right activities, they can be a useful tool in the overall research process. If unsuitable classes of simulation are used (for instance: *simulations as heuristic tool* during the *evaluation* phase), these simulations are unnecessary and do not contribute to the research result. Hence, our framework can aid researchers in employing simulations in ISR as it provides a guideline on when to use what kind of simulation throughout the whole research cycle.

In the future we will investigate whether our guidelines can be empirically validated based on the quite significant simulation works done in ISR. In a second step we hope that our framework can help to identify reasons why some simulation works are quite well received in the community (potentially because they intuitively adhered to our framework) and why some are not. This could ultimately give researchers the perspective on their work needed to create the impact it deserves.

## References

1. van Aken, J.E.: Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules. Journal of Management Studies 41(2), 219–246 (2004)
2. Axelrod, R.: Advancing the art of simulation in the social sciences. In: Conte, R., Hegselmann, R., Terna, P. (eds.) Simulating Social Phenomena, pp. 21–40 (1997)
3. Bratley, P., Fox, B.L., Schrage, L.E.: A guide to simulation, 2nd edn. Springer, New York (1986)
4. Bunge, M.A.: Method, model, and matter. In: Bunge, M. (ed.). Reidel, Dordrecht (1973)

5. Dooley, K.: Simulation research method. In: Baum, J. (ed.) Companion to Organizations, pp. 829–848. Blackwell, London (2002), http://www.public.asu.edu/~kdooley/papers/simchapter.PDF
6. Epstein, J.M.: Why model? Journal of Artificial Societies and Social Simulation 11(4), 12 (2008), http://jasss.soc.surrey.ac.uk/11/4/12.html
7. Frank, U.: Towards a pluralistic conception of research methods in information systems research. ICB Research Report 7, Universitt Duisburg-Essen, Duisburg, Essen (2006)
8. Hartmann, S.: The world as a process: Simulations in the natural and social sciences. In: Hegselmann, R., Mueller, U., Troitzsch, K.G. (eds.) Modelling and simulaton in the social sciences from the philosophy of science point of view, pp. 77–100. Kluwer Academic Publishers, Dordrecht (1996), http://philsci-archive.pitt.edu/archive/00002412/01/Simulations.pdf
9. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
10. Humphreys, P.: Computer simulations. PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association 1990, 497–506 (1990), http://www.jstor.org/stable/193093
11. Jennings, N.R.: On agent-based software engineering. Artificial Intelligence 117, 277–296 (2000)
12. Kaplan, A.: The Conduct of Inquiry – Methodology for Behavioral Science. Chandler Publishing Company, San Francisco (1964)
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., 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: First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 83–106. Claremont Graduate University, Claremont (2006)
15. Pettigrew, A.M.: The double hurdles for management research. In: Clark, T. (ed.) Advancement in Organizational Behaviour – Essays in Honour of Derek S. Pugh, Ashgate, Aldershot, Brookfield USA, pp. 277–296 (1997)
16. Popper, K.R.: The Logic of Scientific Discovery. Taylor & Francis, Abington (2002)
17. Simon, H.A.: The Sciences of the Artificial. MIT Press, Cambridge (1996)
18. Wolfram, S.: Cellular Automata and Complexity. Westview, Boulder (1994)

# Design Logic and the Ambiguity Operator

Richard Baskerville[1] and Jan Pries-Heje[2]

[1] Georgia State University
35 Broad Street NW, Atlanta, Ga 30302, USA
Tel.: +1 404.413.7362; Fax: +1 404.413.7394
baskerville@acm.org
[2] Roskilde University
Universitetsvej 1, 4000 Roskilde, Denmark
Tel.: +45 4674 3051
janph@ruc.dk

**Abstract.** Technological rules are one form of expressing management design activities like organizational design, decision design, and information systems design. However, the notion of a "rule" can imply an unintended over-specification of premises and outcomes. We propose a design logic using the concept of an *ambiguity operator* in the predicate logic format. To test the validity of the ambiguity operator, we used it to express the theory under test in a field experiment. The field experiment demonstrated that the ambiguity operator is both useful and valid in logically capturing the field reality when applying designs expressed in the form of technological rules.

**Keywords:** Design science, design theory, design logic, technological rules, ambiguity, field experiment.

## 1 Introduction

The science of design differs from design implementations in that it involves development of general solutions to general problems. This scientific perspective searches for a class of design solutions that is developed to treat a class of design problems [1, 2]. This design science perspective differs from ordinary design in which designers search for a specific design solution to a specific design problem.

Much of the current work in information systems design science research is grounded more-or-less directly on Simon's concept of the science of design [3]. Simon's notion for the operationalization of scientific design involved the use of imperative logic, or at least, a process substitute for imperative logic involving a search through a declarative logic solution space. Despite Simon's foundation in design logic, much of the work that followed him, including much of the information systems design science research, used different foundations for operationalizing design science, and there has been very little treatment of design logic.

This tendency in information systems to bypass Simon's logical operationalization of design science should not be surprising. The information systems discipline is quite naturally focused on systems design. In this arena, generalized design is historically

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 180–193, 2010.

operationalized as methodology rather than design logic. In these methodologies, general designs are often represented as notation, diagram protocols or relational algebra. Design science research is sometimes invoked for its focus on the IT artefact [4] and is regarded as offering a conducive philosophy for a greater practice-orientation in research [5].

In concert with interests in design methodology, information systems research has a high degree of respect for research methodologies. The tendency of the information systems field to focus on “design theory” rather than “design logic” is also unsurprising. Ours is a relatively young field, and the centrality of theory helps legitimate design-oriented research as a scientific pursuit. The prominence of hypotheses and support for deductive models of design evaluation is consistent with this legitimation role for design theory [2, 6].

In contrast to information systems, other areas of management research have employed design science as a vehicle focused on decision design. From this perspective, design science opens avenues to discover general decision programming and heuristics [7] or to crack open intractable design problems like multi-criteria decision making [8]. The rule orientation of this work is more closely linked to design logic, but like information systems, has not specifically adopted expressions in design logic. Instead, the decision designs are focussed on more traditional expressions of decision heuristics, such as rules.

These areas of management research, information systems and decision science, are developing advanced understanding of how our fields relate to the science of design. However, design logic is underdeveloped across the disciplines. This may be because logical expressions of general designs appear too rigid and imperative for application to diverse problems. Unbending design logic is easily discarded as too strict for the "real world". Given that design logic was Simon’s original, central approach to expressing design, this paper explores how expressions of design logic can be made expressly flexible and consequently used to describe generalized design solutions for generalized design problems.

In particular, we introduce the necessity for ambiguity in generalized design solutions as a means for achieving generalizable designs and generalized problem settings. This ambiguity is necessary because each problem setting in organizations is unique to a certain extent. This means that each design solution must also be unique to a certain extent. The ambiguity is necessary in positing a general version of a unique setting or design solution without expressing exactly how this setting or solution might be similar to some other, future, setting or solution.

## 2 Design Logic in Analytical and Generative Settings

The notion of a design logic is not well explored in the literature. Most work in this area regards electronic circuit logic design, or uses the term informally as a reference to the logic behind designs for communication such as rhetorical design logic [9] or community design [10]. For our purposes we will use the term design logic to refer to a set of formal principles of reasoning employed by designers for creating a design. Because such a set of principles maybe used across a general doss of designs, it falls into the realm of meta design and design science.

At least two general intellectual approaches to design creation have been notable. Analytical design is characterized by its basis on rules. With analytical design, the outcome is determinate, defined by a form of propositional understanding [11]. This is perhaps the ideal for relating science and design. The alternative general intellectual approach is generative design. With generative design, the out come is indeterminate, defined more by the subjective feelings of the designer. Both generative and analytical productions are manifestations of working reason leading to figural schemas, but in quite different ways. But generative productions are those in which the faculties of reason align differently than with the analytical productions that are prized in science [cf. 12]. Generative designs have been associated with Kant's conceptualization of aesthetics: intellectual production, in which faculties of reason and are aligned in fundamentally different relationships that emerge from a momentum of ideas into a figural schema more complete than nature [11]. Over simplifying a bit, analytical designs are calculated, generative designs are invented creatively.

In his conceptualization of a science of design, Simon [3] operated almost exclusively in the realm of analytical design. For Simon, the science of design required an imperative logic, one involving not statements of the way things 'are', but statements of the way things 'should become'. However, Simon dismissed imperative logics as flawed. This may be because systems of imperative logic have a different purpose. Formal imperative logic is more associated with ethics than design [13]. As a substitute he introduced declarative logic within a search process. Essentially, the designer sets up a declarative framework for expressing the design solution, and then searches through various values for the framework elements until a satisfactory design is discovered. Consequently, the search for alternatives is a prominent aspect of design research. This search represents the systematic process for discovering design solutions, or the components of design solutions. This search aspect emphasizes the means-ends operations in design research, and the rationality of allocating resources both to the design process and the representation and acquisition of the constituent elements of the artifact being designed. "Problem-solving systems and design procedures in the real world do not merely assemble problem solutions from components but must search for appropriate assemblies" [3, p. 124].

## 3 Technological Rules; Analytical versus Heuristic Rules

Joan van Aken developed the notion of technological rules as a direction to take design science for applying to management. He was concerned over the usefulness of the corpus of research into management phenomena. In his view, "academic management research has a serious utilization problem" [7, p. 219]. This viewpoint criticizes management research that is too often descriptive and historical. There is little direct usefulness of such reflective studies for managers facing newer and more current problems. Management research would advance if it becomes less descriptive and more prescriptive and less historical and more design-oriented. Van Aken's [14] work argues that a new form of theory, a design theory consisting of "field-tested and grounded technological rules" [7] offers a design science research approach to management.

Van Aken's notion of design science for management recognizes two possible outputs: artefacts or interventions. Three designs can inhabit in a professional episode. The object-design defines the artefact or intervention. The realization-design is the plan for implementing the artefact or intervention. The process-design is plan for the design process itself. In this sense designing is similar to developing prescriptive knowledge.

Van Aken suggests expressing a design in the form of technological rules: "A technological rule follows the logic of 'if you want to achieve **Y** in situation **Z**, then perform action **X**'. The central element of the rule is action **X**, a general solution concept for a type of field problem" [15, p. 23]. A formal expression of this rule would be,

**(Z,Y) → X**

In this case, **X** is the imperative "Do **X**". Imperative logic seeks to control human actors. An example of this logic would be, "If you want to achieve user acceptance of a new technology in a situation of user alienation, then adopt a participative design approach". This technological rule might be stated,

((**USER ALIENATION**), (**USER ACCEPTANCE OF A NEW TECHNOLOGY**))

→ **ADOPT A PARTICIPATIVE DESIGN APPROACH**

Van Aken [15] warns that technological rules need grounding, Grounding prevents technological rules from degenerating to a form of instrumentalism which operates with 'just' rules of thumb. "In engineering and in medicine, grounding of technological rules can be done with the laws of nature and other insights from the natural and the life sciences (as well as from insights developed by these design sciences themselves). In management, grounding can be done with insights from the social sciences" (p. 25). No matter how helpful technological rules may be to managers, they are not design science unless they are grounded in a way acceptable to social science.

In the example above, the underlying theory is fundamental socio-technical theory, which establishes that participative approaches build acceptance through involvement and commitment among users because participation in the design decisions empowers users and allows them shared control over their futures.

Like Simon, van Aken is using technological rules in a declarative logical mode that is invoking analytical productions at design time. However, van Aken regards these as algorithmic and deterministic prescriptions and continues the development of technological rules beyond such strict analytical productions. He permits the designer important latitude to interpret both the situation and the action.

"However, many prescriptions in a design science are of a heuristic nature. They can rather be described as 'if you want to achieve Y in situation Z, then something like action X will help'. 'Something like action X', means that the prescription is to be used as a *design exemplar*. A design exemplar is a general prescription which has to be translated to the specific problem at hand; in solving that problem, one has to design a specific variant of that design exemplar." [7, p. 227].

This distinction between algorithmic and heuristic prescriptions, and the notion of the design exemplar opens the designer process for both analytic and generative productions. Generative productions are required to make the evaluations of "something like" because creative invention is required to adapt the prescriptive action X to the exact context at hand.

Operating with completely unambiguous rules seems problematic. Some form of generative function is necessary to permit management designers to adapt the rules to situations. In discussing technological rules Pawson and Tilley [16] raised the issue of generative causality. Generative causality recognizes that the outcome is not caused naturally by the interventions of managers. Rather, the outcome is an intended outcome being sought in such interventions. This is a generative form of the causal relationship.

This perspective focuses on the possible ambiguity in the intervention (X), the action being taken with the aim of developing the desired results. Which of the generative mechanism(s) (the various X alternatives) that are used in an intervention actually produces the outcome in a given context? This question leads to the formulation of the CIMO-logic that can be formulated in the following way, "In this class of problematic Contexts, use this Intervention type to invoke these generative Mechanism(s), to deliver these Outcome(s)." [17, p. 395].

Besides detailing the formulation of technological rules by virtue of the CIMO-logic, Denyer et al. [17] suggest the term 'design proposition' instead of 'technological rule' arguing that "the latter term suggests—contrary to our intentions—a rather mechanistic, precise instruction".

The empirical cases we are reporting below used the technological rules rather than the CIMO-logic. While perhaps less logically comprehensive the technological rules were simpler and more accessible for our cases.

## 4 Ambiguity Operators – Notions of Design Logic

In order to represent this notion of "something like" action X, an ambiguity operator (~) is introduced into the logical representation of the technological rule.

$$(Z,Y) \rightarrow \sim X$$

Which now represents "if you want to achieve Y in situation Z, then something like action X will help." This rule represents a design production that contains both analytical and generative elements. The analytical element arises from the core rule,

$$(Z,Y) \rightarrow X$$

The generative element arises in the open ambiguity around the action to be taken. The rule expects the designer to invent an adaptation of action X that depends on the situation. This variated action, ~X, makes the design at least partly generative, requiring a different form of reasoning for deciding how a special form of action X should emerge.

Because human organizations are so multivariate, it may he the case that action ~X is more the norm than action X. For many technological rules the ideal setting for action X may arise rarely, making action~X a necessity for most cases. An example of this logic would be, "If you want to achieve user acceptance of a new technology in a situation of user alienation, then adopt something like a participative design approach". This technological rule might be stated,

((**USER ALIENATION**), (**USER ACCEPTANCE OF A NEW TECHNOLOGY**))

→ **ADOPT SOMETHING LIKE A PARTICIPATIVE DESIGN APPROACH**

Following from the notion of multivariate human sittings it is also likely that situation Z is also an idealization that will rarely be found in an exact form. Here again the ambiguity operator can be used to represent an adaptive technological rule. "If you want to achieve Y in a situation something like Z, then something like action X will help."

**(~Z,Y) → ~X**

In the absence of specific deductive logic to help disambiguate **~Z ,** the decision as to whether the situation at hand is in fact something like Z will itself be a generative production requiring the designer to imagine the relevant ways in which the two situations are similar.

Indeed, the technological rule can be made fully ambiguated. If you want to achieve something like Y in a situation something like Z, then something like action X will help."

**(~Z,~Y) → ~X**

Without specific deductive logic to help disambiguate **~Y ,** the decision as to whether the goals at hand are in fact something like Y will demand a generative production requiring the designer to imagine the relevant ways in which the two sets of goals are similar.

Essentially, the ambiguity operators offer the necessary latitude to design logic to be flexible and to permit generative productions that adapt the design logic to the setting.

In addition to permitting generative productions in a design science, ambiguity operators introduce generality into design logic. For example, the core technological rule, has often arisen as an empirical point solution:

**(Z,Y) → X**

A rule such as , "If you want to achieve user acceptance of a new technology in a situation of user alienation, then adopt a participative design approach" will have arisen as a field experience in which participative design was tried as a way to overcome user alienation. At that time, the ideas embodied in the rule were a solution to a quite pointed, specific, practical problem. This *point solution* has since been advanced as the general rule above. We substitute the design logic for the point logic used in the setting. However, there are differences. In the point logic, the participative design was quite specific, for example, the point logic might have involved assigning users to design teams, user specification review sessions, or prototyping with user experimentation. In the design rule, these point solutions are expressed generally as "participative design".

The ambiguity operator opens up further generality, suggesting that "something like" participative design should operate successfully. This enables the designer to consider alternatives to participative design that may work better in the setting-at-hand, and effectively making participative design an element of some unstated general class of solutions that will need to be conceptualized in the future.

From this perspective the ambiguity operator defers the invention of a new general form to future design scientists.

## 5 Applying the Ambiguity Operator: A Working Theory

We applied the ambiguity operator initially to express our working theory about our expectations for common patterns of ambiguity in organizational settings. This working theory involved two working propositions:

Working proposition 1: If the situation (Z) is disambiguous, designers are driven to more analytical mental productions, and this will disambiguate the action (X) and the goal (Y).

Working proposition 2: If the situation (~Z) is ambiguous, designers are driven to more generative mental productions and these will tend to ambiguate either or both the action (~X) and the goal (~Y).

If these working propositions hold, common patterns of ambiguity should cluster around four of the eight possible rule patterns:

**(Z,Y) → X**

**(~Z,~Y) → ~X**

**(~Z,~Y) → X**

**(~Z,Y) → ~X**

If the working propositions hold, the other patterns should be rare:

**(Z,~Y) → ~X**

**(Z,~Y) → X**

**(Z,Y) → ~X**

**(~Z,Y) → X**

## 6 Research Method

We examined the validity of the ambiguity operator using a qualitative field experiment to test the working theory. Our central purpose in this experiment was to test the operability of the ambiguity logic to express and test theoretical propositions. *Whether the outcome of the experiment confirms or disconfirms the propositions is less important than the validity and clarity of the logic used to express the propositions and the results (for our purposes).* In the case at hand, as the reader will see, the experiment disconfirms its propositions. The logical clarity with which this result finds expression, and the ability to reformulate the propositions for further result, may indeed provide better evidence for the strength of the ambiguity operation than an alternative result that simply confirmed the theory under test.

In this experiment, practicing information systems project managers explored the ambiguity in their technological rule settings. The purpose of this experiment was to test whether the use of an ambiguity operator as a logical modifier for technological rules would lead to insights into the use of technological rules in real project settings.

### 6.1 Field Experiment

The technological rules for this setting were derived from the design science nexus [8]. This framework for organizational change was presented to a number of experienced project managers undertaking an executive master in project management and process improvement. The technological rules that follows the 'basic' **(Z,Y) → X** form can be found in [18]. An example is shown in Figure 1.

**If you want** to initiate organizational change **in a situation** where you:
   Believe that formal structures needs change
   Where change is needed fast
**Then choose** a *Commanding* approach where change is driven and dictated by (top) management; one where management takes on the roles as owner, sponsor and change agents.
**If you want** to initiate organizational change **in a situation** where:
   You believe that target group is very diverse and has large individual differences
   The target group are experts
**Then choose** an *Optionality* approach where change is driven by the motivation and need of the individual; it is to a large degree optionality.

**Fig. 1.** Example technological rules from the organizational change nexus

The organizational change design nexus includes ten different approaches for change each having a separate set of technological rules. The project managers were presented the 10 approaches in a 4-day seminar (40 teaching hours) in November 2009. After the 4-day seminar they were asked to hand in their organizational change plan after having applied the technological rules in their own organization to their own change project. In Table 1 we have shown the participating change projects.

The participants' projects plans were then coded for ambiguity in one or more of the three elements in the technological rules. In the coding we looked for any and every possible sign of ambiguity in Y (aim), Z (situation) and X (action).

### 6.2 Results from Field Experiment

Table 2 depicts the results of this experiment. We have included quotes from the participants' project plans that clarify how these situations are, or are not, ambiguous.

**Table 1.** Overview over the 15 project managers, companies and change projects engaged

| Name | Company | Change project |
|---|---|---|
| 1. Alfa | Engineering company that runs projects for customers | Implementing a new project calculation tool for all projects in company |
| 2. Bravo | School which have a drug problem | Change attitude towards drugs; it is not socially acceptable |
| 3. Charlie | Large governmental organization | Implementing a new budget and accounting model |
| 4. Delta | Fusion between two Engineering companies | Integrate the two former companies better |
| 5. Echo | Hospital in the Greater Copenhagen Area | Improve |
| 6. Foxtrot | Software House selling solutions to customers for electronic case and document handling | Implementing solutions for electronic case and document handling at customer sites |
| 7. Golf | Large software house in financial sector | Fusion two companies successfully |
| 8. Hotel | Local Healthcare Center in Copenhagen | Relatively newly established public organization. Needs to start working according to defined goals for local healthcare |
| 9. India | Silviculture Association | Implementing time and material registration |
| 10. Juliet | Intelligence Service | Teach personal to be able to cope with modern terrorism |
| 11. Kilo | Large Danish Municipality | Changing work processes and work routines in part of the organization |
| 12. Lima | Smaller hospital close to Copenhagen | Changing the ways patients are treated at the hospital |
| 13. Mike | Danish Parliament | Changing workflows |
| 14. Novemb | Psychiatric and handicap administration in a Region of Denmark | Optimizing workflows |
| 15. Oscar | Major Bank | Change projects in general |

**Table 2.** Coding of ambiguity for the 15 cases in the field experiment

| | You want to achieve Y | in situation Z, | then perform action X |
|---|---|---|---|
| 1. Alfa | Y very clear;<br>Implement a tool | Z clear.<br>Consistent number of customers | **X ambiguous;**<br>"Specialist driven change … I cannot see that"<br>"Combined with the optionality change strategy it is more meaningful"<br>"Model needs to be seen either in relation to internal or external customers" |

**Table 2.** (*Continued*)

| | | | |
|---|---|---|---|
| 2. Bravo | **Y ambiguous;**<br>"… present *a number of ways* that the individual can identify with" | Z clear;<br>Students with drug problems | X relatively clear;<br>Combine optionality with a specialist driven strategy "… to ensure that the individually preferred methods are analyzed, diffused and in fact used." |
| 3. Charlie | Y somewhat clear;<br>Implement new budget model before the end of 2010 | **Z ambiguous;**<br>"Large geographical distribution"<br>"… it is not sufficiently taken into account how large a part of the organization that is involved" | X clear,<br>Commanding will fit well with hierarchical structure. Combine with BPR-strategy to ensure that the old budget model is extinguished |
| 4. Delta | Y clear;<br>Better integration to achieve the benefits of being together | Z clear;<br>Two companies brought together as one (fusion) a year ago | X seems to work;<br>"Socializing and specialist driven "seems to work fine here a year after the fusion …"<br>"Commanding scores low. That was the strategy used the first period after the fusion. It is clear that it did not work" |
| 5. Echo | **Y ambiguous;**<br>"It *may* the tasked to management to help staff create *a room* for ideas and creativity" | **Z ambiguous;**<br>"The hospital has large cultural *differences* from one department to another" | **X ambiguous;**<br>"The tool [change nexus] is *somewhat useful*. I will not suggest change strategies based on this tool alone" |
| 6. Foxtrot | Y very clear;<br>Implement the tool sold | **Z ambiguous;**<br>"Major difference between a strategic change for a whole organization, and a change in one or few departments" | **X ambiguous;**<br>"I believe the type of change has as much to say as the situation"<br>"You would probably score differently if you had has mapped the situation right before using the Nexus" |

**Table 2.** (*Continued*)

| | | | |
|---|---|---|---|
| 7. Golf | **Y ambiguous;** "The new organization faces a relative long process where they choose infrastructure and work processes for future systems development" | Z clear; Company has just agreed to fusion with another | **X ambiguous;** "It seems to be a very flimsy foundation to base a change strategy on 2 to 4 parameters" |
| 8. Hotel | Y clear, Start working as planned | Z clear; Newly established. Goals for local health-care defined from central healthcare | **X somewhat ambiguous;** "Model is useful but before the final decision on strategy one needs to think in more parameters and not be seduced by the one-string structure [of the Nexus]" |
| 9. India | **Y ambiguous;** "It is our wish to undertake adjustments that can increase profit …" | **Z ambiguous;** "It is a fact the organization has responded very slowly to using time and material registration" | X clear; [BPR strategy scoring highest]. "I am convinced that this model is useful" |
| 10. Juliet | Y clear; To be able to cope with modern terrorists | Z clear; The "modus operandi" of modern terrorists is known | X became clear; "Originally I was skeptical... but the model ashamed me …" "The model recommended the strategy that I would intuitively have chosen" |
| 11. Kilo | **Y ambiguous;** "… it is the insights of the specialists that will be used to optimize workflows and higher quality" | **Z ambiguous;** "There are large differences between tasks in the organization so it is natural that it will take time to implement changes" | **X but with some ambiguity;** "Tool is easy to use and a good help … will work fine in combination with other models so you don't base your strategy choice on just one model" |
| 12. Lima | Y somewhat clear; "We do what we have always done" "The environment is characterized by carelessness" | **Z ambiguous;** "I believe the model doesn't paint a true and fair view of the organization" | X somewhat clear; "The nexus creates some overview and some recommendations that fits well" |

**Table 2.** (*Continued*)

| | | | |
|---|---|---|---|
| 13. Mike | Y clear; New workflows | **Z ambiguous;** "the divergence between the units means that the change doesn't have to look the same in different places" | X clear; "I had my boss fill out the Nexus questionnaire, and he came up with the same" |
| 14. Novemb | Y clear, A template for administrative consideration of cases | Z clear; Long and tedious case workflows today | **X ambiguous;** Argues that one cannot uncritically use the results; and argues for another than the model |
| 15. Oscar | **Y ambiguous;** "… without having specific changes in mind." | **Z ambiguous;** Many knowledge workers and "highly diverse departments" | X clear; [Optionality strategy scoring highest]. "… fits well with the organization." |

From these data in table 2, we can see that there were four rule patterns that occurred multiple times

**(Z,Y) → ~X** (appears three times)

**(~Z,Y) → X** (appears three times)

**(~Z,~Y) → X** (appears twice)

**(Z,Y) → X** (appears twice)

The remaining rule patterns that occurred once each:

**(~Z,~Y) → ~X**

**(~Z,Y) → ~X**

**(Z,~Y) → ~X**

**(Z,~Y) → X**

## 7 Discussion

The field experiment offers results at two levels. The first level regards the outcome in terms of the propositions that were disconfirmed by the data. While this is not our primary objective, we would be amiss not to discuss these results in terms of their implications for future research. This discussion will lead to the results from second level. The second level regards the validity and usefulness of the ambiguity operator when used to modify the predicate logic of technological design rules.

### 7.1 Discussion of the Field Experiment

This data does NOT support the working propositions and consequently do NOT support the working theory. Indeed, the most common patterns appeared from those expected to be rare. Instead, it appears that the data align better with an extension of the generative causality and CIMO logic discussed earlier. Recall that the CIMO logic emphasized the generative causality arising from the ambiguity in actions that may be taken in an intervention. This situation is a common pattern in our data, similar to one in which the action to be taken is ambiguous, (Z,Y) → ~X.

However, the data suggest that this logic may need to be expanded. The other most-common pattern involved ambiguity in the situation (Z). We can speculate that ambiguity in Z drives analytical treatment of Y and X, (~Z,Y) → X.

### 7.2 Discussion of the Ambiguity Operator

While this simple qualitative field experiment is insufficient to prove or disprove these theories, it does strongly validate the usefulness of the ambiguity operator in design logic. As we mentioned earlier, whether the outcome of the experiment confirms or disconfirms the propositions is less important than the validity and clarity of the logic used to express the propositions and the results.

The ambiguity operator permits a clearer understanding and representation of the expected and unexpected variation between the elements of a design logic expression and an instance of the application of this design logic. In this experiment, the ambiguity operator enabled us to clearly express a working theory and its system of propositions. Based on these expressions we were able to formulate and execute a simple, qualitative field experiment. Surprisingly, the results of the experiment upsets this theory, an outcome that perhaps provides more convincing evidence of the value of the ambiguity operator than a confirmation of the theory. We find this evidence especially convincing because of the immediate ability to use the logic to reformulate new propositions for further research.

## 8 Conclusion

Design is a core part of many management activities be it organizational design, decision design, information systems design, etc. Technological *rules* can be used to express a design. But the word rule is often mis-interpreted as something that should be strictly followed.

In this paper we have shown that an ambiguity operator can be added to the basic formulation of technological rules thereby formulating a design logic in predicate logic format. Indeed, technological rules can be made fully ambiguated in the following way: If you want to achieve something like Y in a situation something like Z, then something like action X will help."

**(~Z,~Y) → ~X**

We examined the validity of the ambiguity operator using a field experiment with 15 people. We expected that if the situation (Z) is disambiguous, designers are driven to more analytical mental productions, and if the situation (~Z) is ambiguous, designers are driven to more generative mental productions and these will tend to ambiguate either or both the action (~X) and the goal (~Y). None of these working propositions were supported.

Instead we found that the data align better with an extension of generative causality and the so-called CIMO-logic expanded.

Thus our conclusion is that the concept of the an ambiguity operator seems to be very useful in representing what really happens when a design expressed in the form of design logic meets reality in the field. Because it makes design logic more diversely applicable, the ambiguity operator makes design logic a more useful tool for the purpose of design science research.

## References


1. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)
2. 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)
3. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
4. Orlikowski, W.J., Iacono, C.S.: Research commentary: Desperately seeking "IT" in IT research - A call to theorizing the IT artifact. Information Systems Research 12(2), 121–134 (2001)
5. Iivari, J.: A paradigmatic analysis of Information Systems as a design science. Scandinavian Journal of Information Systems 19(2), 39–63 (2007)
6. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems 8(5), 312–335 (2007)
7. van Aken, J.E.: Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. The Journal of Management Studies 41(2), 219–246 (2004)
8. Pries-Heje, J., Baskerville, R.: The design theory nexus. MIS Quarterly 32(4), 731–755 (2008)
9. Bryan, S.P.: Cognitive complexity, transformational leadership, and organizational outcomes, Louisiana State University and Agricultural & Mechanical College: United States – Louisiana (2002)
10. Fritzen, S.A.: Can the Design of Community-Driven Development Reduce the Risk of Elite Capture? Evidence from Indonesia. World Development 35(8), 1359–1375 (2007)
11. Groat, L., Wang, D.: Architectural Research Methods. Wiley, New York (2002)
12. Kant, I.: The Critique of Pure Reason (1781). In: Rand, B. (ed.) Modern Classical Philosophers, pp. 370–456. Houghton Mifflin, Cambridge (1908)
13. Gensler, H.J.: Formal Ethics. Routledge, New York (1996)
14. van Aken, J.E.: Valid knowledge for the professional design of large and complex design processes. Design Studies 26(4), 379–404 (2005)
15. van Aken, J.E.: Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management. Journal of Management 16(1), 19–36 (2005)
16. Pawson, R., Tilley, N.: Realistic Evaluation. Sage, London (1997)
17. Denyer, D., Tranfield, D., van Aken, J.: Developing Design Propositions through Research Synthesis. Organization Studies 29(3), 393–413 (2008)
18. Pries-Heje, J., Baskerville, R.: Management Design Theories. In: Pries-Heje, J., et al. (eds.) Human Benefits Through the Diffusion of Information Systems Design Science Research. Springer, Boston (2010)

# Information Model-Based Configuration of Situational Methods – A Foundation for Design Research Applying Situational Method Configuration

René Fitterer

University of St. Gallen, Institute of Information Management,
Müller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland
Rene.Fitterer@unisg.ch

**Abstract.** Design research as an information systems research paradigm aims for the construction and evaluation of innovative artifacts for relevant problems. Design science reflects the design research process aiming to create standards for its rigor. Design science produces guidelines and metrics for the construction and evaluation of artifacts. In the field of method engineering design science established situational method engineering as a means to maintain rigor of artifacts, while at the same time integrating situation-specific adaptation mechanisms to increase applicability and thus improve relevance. The design science knowledge base on situational method engineering however lacks a systematic integration between the semantics of a method and the semantics of the situations it is adapted for. The work presented in this paper applies ontological meta modeling to address a better representation of the interdependencies between method elements and configuration rules in situational method configuration and demonstrates its exemplary application on an existing artifact.

**Keywords:** Design science, situational method engineering, adaptation mechanisms, method configuration.

## 1 Introduction

Design research (DR) as a research paradigm of Information Systems Research (ISR) has significantly matured over the last years. Defined to "produce and apply knowledge of tasks in order to create effective artefacts" that "serve human purposes", design research (DR) can be considered a well accepted problem-oriented ISR approach. DR aims for the construction and evaluation of innovative artifacts which can be in the form of "constructs, models, methods, and implementations" [1, p. 253]. Not solely initiated, but significantly catalyzed by the contributions of March, Smith and Hevner et al. [1, 2] the sciences of DR – design science (DS) – reflect the DR process in an effort to create standards for its rigor [3]. As a result, numerous guidelines, process models and metrics for the construction and evaluation of DR artifacts are published, e.g. [1, 2, 4, 5], to ensure rigor of and differentiate DR from construct-based solution engineering.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 194–209, 2010.

The necessary level of generality of DR artifacts however also imposes a challenge to the second, central target variable of DR, i.e. relevancy. Acknowledging the limited utility of generic artifacts, which is pointedly illustrated by the equation "Generality * Utility = C (a constant)" [6, p. 58], situational artifacts, which recognize situation-specific factors that influence their application [7], are considered a means to maintain rigor of artifacts, while at the same time integrating situation-specific adaptation mechanisms in order to increase applicability and thus improve relevance [3, 8].

Originating back to organization theory research in the early 1970s [9], the concept of creating situational artifacts is established today within the area of reference modeling [10-12] and situational method engineering [13-17]. The respective DS research defines guidelines and meta methods for designing reference models and method artifacts for re-use. DS in this context addresses procedural and structural aspects of artifact design or the foundations of adaptation mechanisms for situation-specific artifact adoption based on accepted modularization concepts. However, little work has been directed to address a systematic, content-related foundation of DR on situational artifacts [18, 19]. The work presented in this paper aims to enable the definition of adaptation mechanisms for situational methods based on semantics of the problem domain. Therefore this paper proposes an extension of existing meta methods for situational method engineering, which applies the concept of ontological meta modeling [20, 21] to address a better representation of the interdependencies between method elements and configuration rules in situational method configuration. It thereby provides a comprehensive analysis and discussion of the existing DS knowledge base on situational artifacts in order to draw conclusions for the proposed contribution to the foundation of situational method configuration.

The remainder of this article is structured as follows: The following section discusses the fundamentals of situational artifacts with a particular focus on method engineering and method engineering-relevant implications of DS guidelines for other artifact types. This forms the foundation for the review of the existing DS knowledge base on the construction of situational methods in section 3 and a detailed analysis of existing approaches for method configuration in section 4. Subsequently, the approach for configuration of situational methods based on ontological meta models is presented in section 5. The presentation of the approach for adaptation of situational methods using ontological meta models is based on experiences from the construction of an artifact for requirements-based assessment of IS architectures in hospitals. An exemplary application of the presented approach to this artifact is presented in section 6. The article concludes with a critical discussion of the presented work and an outlook on future research in the field of the sciences of situational method engineering.

## 2 Situational Artifacts

Addressing the previously stated relevancy dilemma of generic artifacts, the ISR fields of reference modeling and situational method engineering have developed scientific foundations for the construction of models and methods that include mechanism for situation-specific adaptation [3]. Reference models constitute generic conceptual models that formalize state-of-the-art or best-practice knowledge of a certain domain [22]. These models thereby describe robust yet flexible representations of a problem (or solution) space by means of generic information that suits more than one situation. Analogues, a generic method is defined as "an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products" [13, p. 275-276]. Both artifact types hence share the characteristics of generality and reference (i.e. constitute good/best practice) [23], whereby models take a product view and methods take a process view on the respective problem or solution space [24].

The scientific foundation of the adaptation of reference models led to the definition of a discrete set of adaptation mechanisms, which are classified into two groups – generating and non-generating adaptations [25, 26]. Non-generating adaptations require the model user to create situation-specific content by aggregating, instantiating, specializing or creating analogies of the elements of the generic reference model [11]. By contrast, generating adaptation mechanism, generally also referred to as configurative adaptation mechanisms, define clear rules of how the defined reference model can be adapted to the situation-specific requirements. The configuration is thereby achieved by model type selection, element type selection, element selection, synonym management or presentation variation [27, 28]. The adaptation mechanisms provide different levels of support and respective restrictions on the use of reference models (cp. fig. 1). While configurative approaches generally provide a more comprehensive support to a reference model user, the need to ex-ante identify and define a diverse set of configuration rules results in a higher complexity and a respectively higher effort of method construction [11].

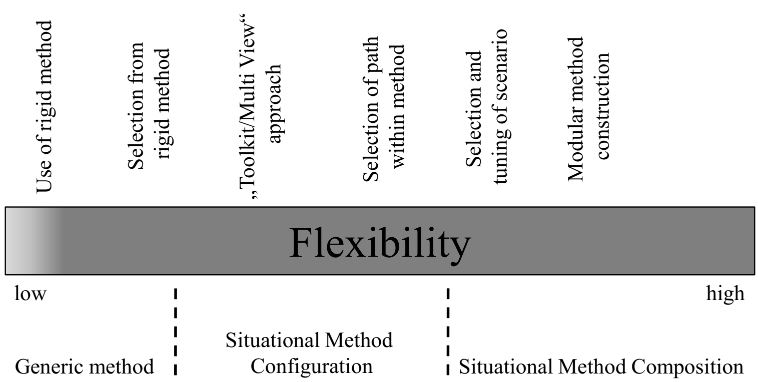



**Fig. 1.** Flexibility of situational methods based on [17]

The described reference model adaptation mechanisms are also considered in situational method engineering [8, 23]. A differentiation into situational method configuration and situational method composition can be applied analogue to the classification of reference model adaptation mechanisms [29, 30]. While situational method configuration applies the generating adaptation of a so called base method for the specific problem situation, situational method composition applies a selection and orchestration of individual method fragments [29, 31]. The later approach is thereby not aimed at the configuration of a single base method. Instead a situation-specific new method is constructed based on the application of non-generating adaptation mechanism. According to literature analyses of the state-of-the-art of procedures and meta methods for situational method engineering, method composition dominates the current DS and DR knowledge base on situational method engineering, whereby aggregation and specialization are mostly applied [8, 31]. The knowledge base is therefore primarily focused on the DS foundation of rather flexible, less restrictive method engineering (cp. fig. 1).

## 3 Related Work – Design Processes for Method Adaptation

Situational method engineering as a field of DSR can be considered a rather young research field. With its roots in organization theory, early DS publications on method engineering and the consideration of situational method composition or configuration date back to the mid 1990s [e.g. 13, 17, 32] and are still a field of ongoing DS research [e.g. 16, 29, 33]. The following paragraphs discuss accepted DS publications which define meta methods addressing the design process and adaptation of situational methods. The selection from the knowledge base is thereby based on the previously cited literature analyses contained in [8, 29] as well as on the author's own forward and backward analysis of cited references of these articles.

With respect to the previously mentioned degree of flexibility of situational methods, one can clearly state a dominance of both design processes and design results (i.e. situational methods) which follow the more flexible, less guiding method composition approach [8]. Based on the concept of self-contained modules of situational methods, so called method fragments (also referred to as method chunks or method components)[1] Brinkkemper et al. define a design process, that is based on method adaptation using aggregation mechanisms, including rules that guide the assembly of the fragments [13]. Similar approaches are proposed by [16, 34, 35], whereby [35] extends the scope of method composition to the disassembly of generic methods in order to identify relevant method fragments. In addition to method assembly, [36] defines patterns as a means to extend existing methods through specialization. Two of the few DS contributions on method configuration are proposed in [33, 37], which describe approaches that use a specific base method as a basis for the creation of project specific configurations.

The presented approaches share a common view on the fundamental components of situational methods. A respective generic meta model of the elements used to describe situational methods based on [29, 38] and extended with the adaptation mechanisms (highlighted in grey) described in [31] is shown in fig. 2. These components can be separated into process-related and product-related components [13]. Process-related components are comprised of the set of responsibilities, activities, techniques and guidelines and their relationships that define the necessary activities to achieve the goals of the method. These are represented in the generic meta model by the

[1] Despite their origin in method composition, for the remainder of the method fragments are used to refer to coherent sets of design activities which are grouped to realize a certain design result and therefore also used in the context of method configuration.

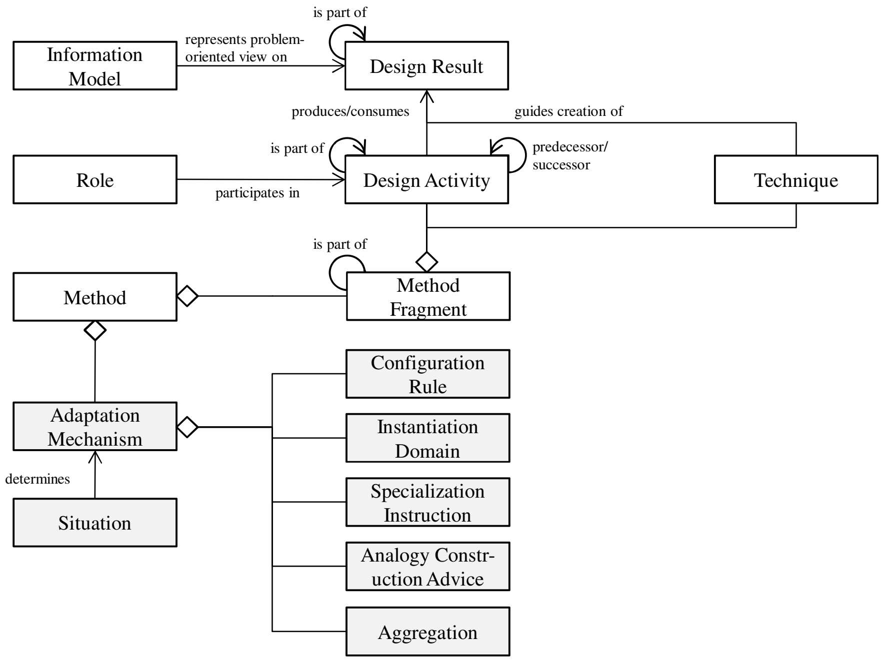



**Fig. 2.** Generic meta model of situational methods based on [29, 31, 38]

entities "role", "design activity", "technique" and their relationships. Correspondingly the product-related components of the method define the set of concepts, their properties and relationships necessary to express the outcome of the process [35]. These are represented by the entities "design result" and "information model". Defining the problem-oriented view on design results, the information model thereby represents an ontological meta model [20, 21] of the product-related components of a method.

With respect to the adaptation mechanisms described in [31], an overview of the respective procedures of adaptation mechanisms for artifacts is shown in table 1. The following section discusses the capabilities of DS contributions on situational method engineering to guide in the creation and application of adaptation procedures Addressing the previously motivated need to provide a better guidance to developers about the steps of the method to be modified or omitted (cp. section 2) as described in the previous section, the focus thereby lays on the DS contributions on method configuration presented in [31, 33, 37]. In contrast to reference modeling, where adaptation mechanisms and the respective attributes are assigned typically to only one type of artifact component, i.e. model elements, adaptation mechanism in situational method engineering can be based on a number of different component types (information model, design activity, design result, etc.) [31]. The subsequent analysis therefore focuses particularly on the types of method components that are either used to define the condition part (the act of configuration) or the conclusion part (the object of configuration).

**Table 1.** Adaptation mechanism and procedures based on [31]

| Adaption Mechanism | Adaptation Procedure |
|---|---|
| Configuration rule | A condition part defines a number of states. Based on the condition value a conclusion part defines the type of adaptation (e.g. elimination of elements). |
| Instantiation domain | The method contains a number of generic placeholders, so called instantiation domains, which are filled with valid occurrences as part of the situational adaptation. |
| Specialization instruction | A specialization instruction defines the possibilities to change, extend and/or partially modify the general parts of the method. |
| Aggregation | Interface descriptions of method fragments (i.e. inputs/ outputs) define compatibility and guide their combination. |
| Analogy construction advice | Annotation of an analogy construction advice guides this generally rather unrestrictive approach of adaptation. |

## 4 Analysis of Configuration Mechanisms for Situational Method Configuration

The "Method for Method Configuration" (MMC) presented in [33] describes a meta method for situational method configuration with a particular focus on the configuration of the process-related components of a method. Consequently the objects of configuration are "prescribed actions" [33, p. 623], i.e. method fragments and activities according to the meta model shown in fig. 2. The condition part of MMC adaptations is based on the definition of different sets of configuration packages, which represent a configuration of the base method suitable for specific characteristics of an application situation. The configuration mechanism of MMC applies certain "classifications" to method fragments in order to realize situation-specific methods. These classifications define to what extent the activities of the method fragment need to be performed or not (e.g. omit, perform reduced, etc.) [33]. The act of configuration, i.e. the rationale for configuration is thereby based on "characteristics" which represent instances of problems addressed by certain method fragments. No further recommendations are however made on how to identify such characteristics. Particularly, no semantic relation is established between the information model of the method and the respective semantics of the situation(s) which would express rationales for the relationships between individual configuration parts and conclusion parts of the configuration rules.

A second approach for situational method configuration is described by the "Process Configuration Approach" (PCA) presented in [37]. While PCA also applies primarily a configuration of the process-related components of a method, in contrast to MMC it proposes the "links" between method elements as the objects of configuration. In addition to process flow rules, which constrain links between method fragments or activities, PCA also proposes so called structure rules which address the configuration of links between other method elements, e.g. between activities and techniques [37]. With respect to the act of configuration, PCA uses the concept of "facts" to define the condition parts of the configuration mechanisms [37]. These facts are assertions which define characteristics of the project for which a project-specific method is created. PCA thereby distinguishes between base facts and derived facts, which are derived based on inferences or calculations. Analogue to MMC, the meta model of the configuration rules of PCA [37], also does not imply any semantic mappings between the situational configuration and the method itself. In addition to rules that constrain the configuration of the process flow and structure of the method, PCA also defines a category of constraints rules comprised of completeness and consistency rules [37]. These rules are applied to the information model or meta model of a method and enable the expression of interdependencies of elements as well as the definition of mandatory model elements in a similar fashion to the assembly rules defined in [39].

A third publication on situational method configuration is presented in [31]. However the presented guidelines and findings are predominantly based on the authors' experiences in reference modeling and are consequently rather general. With respect to the object of configuration, the authors state that common configuration points for modeling methods are at the levels of design objects, design activities or roles [31]. The configuration parts of configuration rules are described to "include but are not limited to objectives of the method's use, application systems or respectively organizations to be designed, roles, qualifications, and other attributes of the method users as well as financial, time, personnel, and other restrictions" [31, p. 7].

Based on the presented DS contributions on situational method engineering, one can conclude that, in contrast to reference modeling and situational method composition, this DS field of situational artifacts has a rather limited knowledge base. Despite the adoption of adaptation mechanisms from reference modeling and the definition of various method configuration procedures which address both process-related and product-related components, a major challenge of method engineers to reflect the interdependencies between method elements and configuration rules remains [31]. Also relating to the argumentative dimension, i.e. the methods' inherent rationale as well as the rationale of the defined configuration rules, a stronger semantic linkage is necessary in order to provide better guidance and rationale to method users as to how and why method configuration rules are defined [25, 33, 40].The following section thus presents an approach which aims to extend the existing knowledge base on situational method configuration by proposing the integration of the information model of the method with its configuration mechanisms.

## 5 Information Model-Based Configuration of Situational Methods

Aiming to address a better representation of the interdependencies between method elements and configuration rules in situational method configuration, this section describes an approach that is based on the ontological meta model, i.e. the information model, of generic methods. Information models have not been considered as the basis for definition of configuration rules in the existing approaches described in the previous section. Following the findings in [41, p. 1730] however "integration and rigor [of artifacts] can best be appropriated by the use of a metamodel that defines

[…] the work products […] together with a metamodel describing process and producer elements". While the later meta model relates to a definition of the fundamental component types of methods as shown in fig. 2, the first matches the definition of an ontological meta model such as the information model of a method.

As pointed out in [37] the construction of configurative methods is typically based on existing, generic "base" methods. Drawing on the experiences from reference modeling, the presented approach therefore proposes an extension of the information model of the generic method with the semantics of the situation that it can be adapted for [27]. If possible, such situation semantics should be expressed by the entities of the information model. If the situation requires additional semantics to be defined, two means of extension of the information model are proposed, i) definition of situation-specific entities as a situational extension of the information model and ii) instantiations of entities of the information model. Based on the generic meta model of situational methods presented in fig. 2, the meta model of situational methods for information model-based method configuration is shown in fig. 3. The newly added entities are highlighted by a dashed outline.

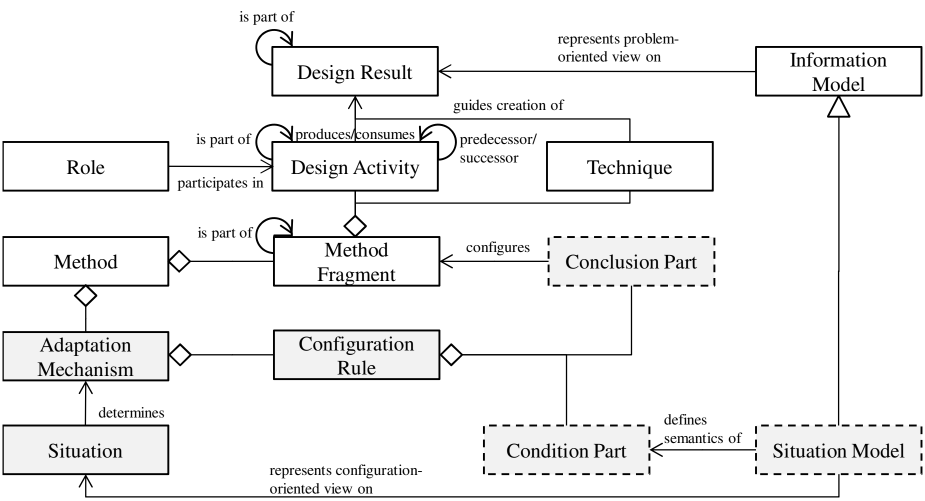



**Fig. 3.** Meta model of situational methods for information model-based method configuration

The situation model shown in fig. 3 contains the previously mentioned extensions and instantiations of the information model necessary to define the semantics of the situation. The entities of the situation model thus define the semantics of the conditions specified for the configuration of methods and hence address the requirement to better represent the interdependencies between method elements and configuration rules. With respect to the conclusion parts of configuration rules, this approach follows the accepted adaptation procedures of reference modeling, i.e. model type selection, element type selection, element selection, synonym management or presentation variation [27, 28], which are applicable to both process-related and product-related components of the generic method. The conclusion part thus configures whole method fragments, respective activities, assigned roles, applied techniques, characteristics of the design result or representations in the information model[2]. In order to exemplify the application of the described approach for information model-based configuration of situational methods, the subsequent section describes its application to a generic method for requirements-based assessment of IS architectures in hospitals.

# 6 Demonstration on a Method for Requirements-Based Assessment of IS Architectures in Hospitals

The generic method for requirements-based assessment of IS architectures in hospitals is constructed in order to address the challenge of CIOs to strategically manage healthcare IS (HIS) in response to the increasing demand for IS to contribute to business and operational goals of healthcare providing organizations [42, 43]. The information model of the initially developed generic method is presented in Fig. 4. In order to realize a holistic assessment based on enterprise architecture considerations [44], the information model contains elements on a strategy level, organization level and system level.

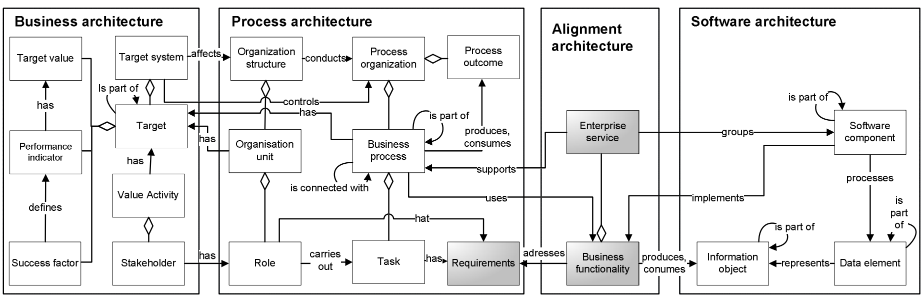



**Fig. 4.** Information model of the generic method for Requirements-Based Assessment of IS Architectures in Hospitals based on [45]

The method consists of four fragments which are shown in table 2. The requirements-based assessment of the IS architecture is based on both strategic and operational priorities. Accordingly the first method fragment M1 addresses the identification of strategy targets and the linkage to operational goals in order to identify value activities of the organization [46]. The method fragment M2 builds on the results of M1. It applies a prioritization of the IT requirements based on the stakeholders affected by or responsible for the value activities. As a result, the target space for the IT architecture is defined based on a prioritized set of IT requirements.

In order to model the existing capability of the IT architecture of an organization to address the requirements, M3 applies the concept of enterprise services, i.e. logic groups of IT functionalities from a business perspective [47], to realize a mapping between the IT components of the organization and the business requirements for the IT. Based on a

[2] For reasons of readability the relationship in fig. 3 is assigned to the method fragment as the integrating component.

reference catalogue of enterprise services for hospitals and respective maturity levels of functionalities contained in the enterprise services, M3 produces an IT capability profile for the organization based on the provided functionality and the availability of this functionality to the organization (local, department, organization-wide). A subsequent grouping of the available enterprise services into different types of HIS, i.e. stakeholder HIS, value activity HIS or process-oriented HIS [cp. 48], in M4, which is based on the analysis preferences of the organization, produces HIS-specific capability profiles for the requirements-based assessment of the IS architecture. The following paragraphs present two configuration rules and entities of the situation model which are constructed for the situational adaptation of the presented method in order to exemplify the application of the approach. Due to the page limitations, these extensions are only described textual rather than model-based.

As shown in table 2, the second activity of M1 proposes three alternative means to identify value activities (strategy execution plans, operational targets or strategic projects) without defining any rules with respect to the reason for choosing one of the alternatives. In order to define configuration rules for the three alternatives, "IT target system" is added as a specialization of the entity "target system" and three respective instances are defined based on the experiences from [49]. These instances are "IT is part of long-term strategy planning and centrally managed", "IT is managed decentralized in the departments" and "IT is not part of strategic planning". Also, the three alternatives for strategy analysis "strategy execution plans", "operational targets" or "strategic projects" are defined as specializations of the entity "goal". Accordingly the following three rules are specified to define the respective element type selection, following the proposed structure for definition of configuration in natural language defined in [37] (the elements of the information/design activity/situation model are highlighted by italic formatting):

- IF *IT target system = IT is part of long-term strategy planning and centrally managed* THEN restrict *activity 2* of *M1 to goals* of type *strategy execution plans*
- IF *IT target system = IT is managed decentralized in the departments* THEN restrict *activity 2* of *M1 to goals* of type *operational targets*
- IF *IT target system = IT is not part of strategic planning* THEN restrict *activity 2* of *M1 to goals* of type *strategic projects*

A second set of configuration rules is defined for the identification of stakeholders for the prioritization of IT requirements in M2. The existing generic activity requires the identification of stakeholders affected by or responsible for value activities. This activity can be interpreted quite flexibly. To improve the intersubjectivity of this activity, the value activities of hospitals defined in [42, 46] (e.g. firm infrastructure, human resources management, diagnosing, intervening, etc.) are added to the situation model as instances of the entity "value activity". Also four generally accepted types of hospital stakeholders are added as specializations of the entity "stakeholder", i.e. "healthcare provider" (e.g. medical and nursing professionals), "supporter" (e.g. administrative staff), "healthcare acceptor" (e.g. healthy people, patients and their relatives) and "controllers" (e.g. management) [50]. As a result, configuration rules for element type selection which consistently limit the scope of the included stakeholders based on their relationships with the concise set of value activities are defined. Further similar rules can be defined e.g. for the grouping mode of HIS in M4 as well as the mapping between enterprise services and IT requirements based e. g. on the value dimensions of HIS presented in [51]. An example of such a rule is:

- IF prioritized value activity = diagnosing THEN restrict activity 2 of M2 to stakeholders of type healthcare provider

**Table 2.** Extract of the components of the method for requirements-based assessment of IS architectures in hospitals

| **Method fragments** | **Design activities, design results and addressed entities of the information model** |
|---|---|
| **M1** Linking strategy to operational goals | **Design activities:**<br>(1) Identify strategy targets of the hospital<br>(2) Analyze strategy execution plans, operational targets or strategic projects<br>(3) Define strategic value activities |
| | **Design result:** Goal map, Value activities |
| | **Addressed information model entities:** Target value, performance indicator, success factor, target system, target, value activity, process outcome, business process, requirements |
| **M2** Identification of stakeholder-specific strategy perspectives | **Design activities:**<br>(1) Map value activities to processes, tasks or roles<br>(2) Identify stakeholders affected by or responsible for value activities<br>(3) Prioritize value activities based on stakeholders priorities<br>(4) Identify IT requirements |
| | **Design result:** Prioritized IT requirements |
| | **Addressed information model entities:** Stakeholder, organization structure, process organization, organization unit, business process, role, task |
| M3 IT Capability Modeling | **Design activities:**<br>(1) Identify enterprise services contributing to prioritized value activities<br>(2) Modeling enterprise service's functionality embedded in the IT architecture components |
| | **Design result:** IT capability profile |
| | **Addressed information model entities:** Enterprise service, business functionality, software component, information object, data element |
| M4 IT Capability assessment | **Design activities:**<br>(1) Calculate EA capability scores of the enterprise services based on maturity and breadth of their adoption within the organization<br>(2) Group enterprise services to stakeholder HIS, value activity HIS or process-oriented HIS according to the strategic targets of the organization<br>(3) Assess strategic value based on the functional capability and the breadth of adoption |
| | **Design result:** Requirements-based assessment of IS architecture |
| | **Addressed information model entities:** Requirements, enterprise services, business functionality, target |

## 7 Conclusion and Outlook

The work presented in this paper aims to provide a contribution to the DS knowledge base on situational method configuration. Considered as a means to maintain rigor and generality of artifacts while at the same time integrating situation-specific factors into artifacts, existing DS contributions have developed numerous guidelines for the construction of situational artifacts. While the full spectrum of flexibility as regards to the definition of adaptation mechanisms is broadly covered in reference modeling DS literature, DS contributions on situational method engineering have mostly focused on the more flexible, less guiding approach of method composition (cp. section 3). By defining discrete sets of configuration mechanisms, the concept of method configuration as proposed by [10, 33, 37] supports a more rigorous definition and limitation of a method's adaptation mechanisms and hence ensures completeness, consistency and correctness of the outcomes of a method. The existing DS contributions on method configuration define meta methods for construction and configuration of situational methods (cp. section 4) The related work however ignores semantic relations between situations and the components of the methods, accordingly both the rationale for configuration parts (why configure) as well as for the conclusions parts (how configure) of configuration rules cannot be rigorously expressed. Existing DS guidelines of method configuration are accordingly limited to ensuring the syntactic and structural quality of configurable methods. Aiming to improve this DS knowledge particularly with respect to the semantic quality of the configuration mechanisms of a situational method, we propose a situational model as an extension of the information model of a method. Such an extension of the overall ontological meta model of a method enables the intersubjective definition and application of configuration mechanism and hence contributes to the goal stated at the beginning of this paper, i.e. to advance the systematic, content-related foundation of DR on situational methods. While existing configurable methods may have implicitly provided such an integration of the situation semantics with the method semantics, the DS knowledge base to the best knowledge of the author lacks such guidelines.

The application of the described approach for information model-based configuration supports its utility for creating configurable methods. However, for the purpose of evaluation, this demonstration is limited in a number of ways. First, the proposed approach has only been applied to one method and its application is only exemplary presented in this contribution. Also, the selected base method is characterized by a strong information focus, i.e. the method fragments and activities are focused on modeling and analysis. Consequently, further evaluation, i.e. the application of the approach to define configuration mechanisms for other generic methods, may show that methods with a construction rather than an analysis focus (e.g. methods for IS design) require situation semantics which cannot as easily be integrated with model semantics as shown in the previous section. A second limitation is related to a general challenge of configurative adaptation mechanism. Configurative adaption necessitates the ex-ante definition of all situation-specific configuration options and is thereby limited to the method designers capability to foresee and implement all relevant adaptations at design time [12]. This problem may be relaxed by adopting extension-based approaches which support identification of extension situations and support the definition of necessary extensions as described in [36]. Besides a sound evaluation of the proposed approach for information model-based configuration of situation methods the integration of extension-based approaches with method configuration is therefore identified as core area of future research, as the semantic situation model may support in better guiding and hence improving the quality of model extensions.

## Literature


1. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)
2. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
3. Winter, R.: Design Science Research in Europe. European Journal of Information Systems 17, 470–475 (2008)
4. 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: Chatterjee, S., Hevner, A. (eds.) Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 83–106. Claremont Graduate University, Claremont (2006)
5. Cao, J., Crews, J.M., Lin, M., Deokar, A., Burgoon, J.K., Nunamaker, J.F.J.: Interactions Between System Evaluation and Theory Testing: A Demonstration of the Power of a Multifaceted Approach to Information Systems Research. Journal of Management Information Systems 22, 207–235 (2006)
6. Vaishnavi, V.K., Kuechler, W.J.: Design Science Research Methods and Patterns – Innovating Information and Communication Technology. Auerbach Publications, New York (2007)
7. Olle, W.T., Hagelstein, J., Macdonald, I.G., Rolland, C., Sol, H.G., van Assche, F.J.M., Verrijn-Stuart, A.A.: Information Systems Methodologies – A Framework for Understanding. Addison-Wesley, Wokingham (1988)
8. Becker, J., Janiesch, C., Pfeiffer, D.: Reuse Mechanisms in Situational Method Engineering. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences (ME 2007), vol. 244, pp. 79–93. Springer, Boston (2007)
9. Simon, H.A.: Administrative Behavior. Free Press, New York (1997)
10. Becker, J., Delfmann, P.: Reference Modeling. Efficient Information Systems Design Through Reuse of Information Models. Physica, Heidelberg (2007)
11. 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 Publishing, Hershey (2007)
12. vom Brocke, J., Buddendick, C.: Reusable Conceptual Models. Requirements Based on the Design Science Research Paradigm. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 576–604. Claremont Graduate University, Claremont (2006)
13. Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. Information and Software Technology 38, 275–280 (1996)

14. Karlsson, F., Wistrand, K.: Combining Method Engineering with Activity Theory - Theoretical Grounding of the Method Component Concept. European Journal of Information Systems 15, 82–90 (2006)
15. Wistrand, K., Karlsson, F.: Method Components - Rationale Revealed. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 189–201. Springer, Heidelberg (2004)
16. Ralyté, J., Rolland, C.: An Assembly Process Model for Method Engineering. In: Dittrich, K.R., Geppert, A., Norrie, M.C. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 267–283. Springer, Heidelberg (2001)
17. Harmsen, A.F., Brinkkemper, S., Oei, H.: Situational Method Engineering for Information System Project Approaches. In: Verrijn-Stuart, A.A., Olle, T.W. (eds.) Proceedings of the IFIP 8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle, pp. 169–194. North-Holland, Amsterdam (1994)
18. Kurpjuweit, S., Winter, R.: Viewpoint-based Meta Model Engineering. In: Reichert, M., Strecker, S., Turowski, K. (eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications: Proceedings of the 2nd Int'l Workshop EMISA 2007, Köllen, Bonn, pp. 143–161 (2007)
19. Ågerfalk, P.J., Wistrand, K.: Systems Development Method Rationale - A Conceptual Framework for Analysis. In: Proceedings of the 5th International Conference on Enterprise Information Systems (ICEIS 2003), Angers, vol. 3, pp. 185–190 (2003)
20. Atkinson, C., Kühne, T.: Model-Driven Development: A Metamodeling Foundation. IEEE Software 20, 36–41 (2003)
21. Höfferer, P.: Achieving Business Process Model Interoperability Using Metamodels and Ontologies. In: Proceedings of the 15th European Conference on Information Systems (ECIS 2007), St. Gallen, pp. 1620–1631 (2007)
22. Fettke, P., Loos, P.: Classification of Reference Models: A Methodology and its Applications. Information Systems and e-Business Management 1, 35–53 (2003)
23. Schelp, J., Winter, R.: Method Engineering - Lessons Learned from Reference Modeling. In: Chatterjee, S., Hevner, A. (eds.) Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 555–575. Claremont Graduate University, Claremont (2006)
24. Winter, R., Gericke, A., Bucher, T.: Method versus Model – Two Sides of the Same Coin? In: Albani, A., Barijs, J., Dietz, J.L.G. (eds.) Advances in Enterprise Engineering III – 5th International Workshop, CIAO! 2009, and 5th International Workshop, EOMAS 2009, held at CAiSE 2009, Amsterdam, The Netherlands, June 2009, pp. 1–15. Springer, Heidelberg (2009)
25. 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 – Efficient Information Systems Design Through Reuse of Information Models, pp. 27–58. Physica, Heidelberg (2007)
26. Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H.: Configurable Process Models – A Foundational Approach. In: Lehner, F., Nösekabel, H., Kleinschmidt, P. (eds.) Tagungsband der Multikonferenz Wirtschaftsinformatik 2006 (MKWI 2006). GITO, Passau (2006)
27. Knackstedt, R., Janiesch, C., Rieke, T.: Configuring Reference Models - An Integrated Approach for Transaction Processing and Decision Support. In: Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS 2006), Paphos, Cyprus, pp. 135–143 (2006)

28. Becker, J., Algermissen, L., Delfmann, P., Niehaves, B.: Configurable Reference Process Models for Public Administrations. In: Anttiroiko, A.-V., Malkia, M. (eds.) Encyclopedia of Digital Government, pp. 220–223. Idea Group Publishing, Hershey (2006)
29. Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R.: Situational Method Engineering – On the Differentiation of "Context" and "Project Type". In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering - Fundamentals and Experiences, vol. 244, pp. 33–48. Springer, Boston (2007)
30. Bucher, T., Winter, R.: Dissemination and Importance of the "Method" Artifact in the Context of Design Research for Information Systems. In: Vaishnavi, V.K., Baskerville, R. (eds.) Proceedings of the Third International Conference on Design Science Research in Information Systems and Technology (DESRIST 2008), pp. 39–59. Georgia State University, Atlanta (2008)
31. Becker, J., Knackstedt, R., Pfeiffer, D., Janiesch, C.: Configurative Method Engineering - On the Applicability of Reference Modeling Mechanisms in Method Engineering. In: Proceedings of the 13th Americas Conference on Information Systems (AMCIS 2007), Keystone, CO, pp. 1–12 (2007)
32. Gutzwiller, T.A.: Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen. Physica, Heidelberg (1994)
33. Karlsson, F., Ågerfalk, P.J.: Method Configuration: Adapting to Situational Characteristics while Creating Reusable Assets. Information and Software Technology 46, 619–633 (2004)
34. Punter, T., Lemmen, K.: The MEMA-Model: Towards a New Approach for Method Engineering. Information and Software Technology 38, 295–305 (1996)
35. Kumar, K., Welke, R.J.: Methodology Engineering - A Proposal for Situation-Specific Methodology Construction. In: Cotterman, W.W., Senn, J.A. (eds.) Challenges and Strategies for Research in Systems Development, pp. 257–269. John Wiley & Sons, Chichester (1992)
36. Ralyté, J., Deneckère, R., Rolland, C.: Towards a Generic Model for Situational Method Engineering. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 95–110. Springer, Heidelberg (2003)
37. Bajec, M., Vavpotič, D., Krisper, M.: Practice-Driven Approach for Creating Project-Specific Software Development Methods. Information and Software Technology 49, 345–365 (2007)
38. Braun, C., Wortmann, F., Hafner, M., Winter, R.: Method Construction – A Core Approach to Organizational Engineering. In: Proceedings of the 20th Annual ACM Symposium on Applied Computing (SAC 2005), Santa Fe, NM, pp. 1295–1299 (2005)
39. Brinkkemper, S., Saeki, M., Harmsen, A.F.: Meta-Modelling Based Assembly Techniques for Situational Method Engineering. Information Systems 24, 209–228 (1999)
40. Rossi, M., Ramesh, B., Lyytinen, K., Tolvanen, J.-P.: Managing Evolutionary Method Engineering by Method Rationale. Journal of the Association for Information Systems 5, 356–391 (2004)
41. Hug, C., Front, A., Rieu, D., Henderson-Sellers, B.: A Method to Build Information Systems Engineering Process Metamodels. Journal of Systems and Software 82, 1730–1742 (2009)
42. Porter, M.E., Olmsted Teisberg, E.: Redefining Health Care – Creating Value-Based Competition on Results. Harvard Business School Press, Boston (2006)
43. Lippeveld, T., Sauerborn, R., Bodart, C. (eds.): Design and Implementation of Health Information Systems. World Health Organization, Geneva (2000)

44. Winter, R., Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. Journal of Enterprise Architecture 3, 7–18 (2007)
45. Fitterer, R., Rohner, P.: Entwicklung eines Metamodells zur Modellierung der fachlichen Fähigkeiten der Softwarearchitektur in Krankenhäusern auf Grundlage von Stakeholder Concerns. In: Schumann, M., Kolbe, L.M., Breitner, M.H., Frerichs, A. (eds.) Tagungsband der Multikonferenz Wirtschaftsinformatik (MKWI) 2010, pp. 91–102. Universitätsverlag Göttingen, Göttingen (2010)
46. Porter, M.E.: Competitive Advantage: Creating and Sustaining Superior Performance. Free Press, New York (1985)
47. Aier, S., Winter, R.: Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. Business & Information Systems Engineering 51, 150–163 (2009)
48. Winter, R.: An Architecture Model for Supporting Application Integration Decisions. In: Proceedings of the 11th European Conference on Information Systems (ECIS 2003), Naples, pp. 1–13 (2003)
49. Lorenzi, N.M., Novak, L.L., Weiss, J.B., Gadd, C.S., Unertl, K.M.: Crossing the Implementation Chasm: A Proposal for Bold Action. Journal of the American Medical Informatics Association 15, 290–296 (2008)
50. Mantzana, V., Themistocleous, M., Irani, Z., Morabito, V.: Identifying Healthcare Actors Involved in the Adoption of Information Systems. European Journal of Information Systems 16, 91–102 (2007)
51. Fitterer, R., Mettler, T., Rohner, P., Winter, R.: A Taxonomy for Multi-perspective ex ante Evaluation of the Value of Complementary Health Information Systems – Applying the Unified Theory of Acceptance and Use of Technology. In: Sprague, R.H. (ed.) To appear in Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS-43). IEEE Computer Society Press, Los Alamitos (2010)

# A Methodology for Content-Centered Design of Ambient Environments

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

[1] Furtwangen University, Robert-Gerwig-Platz 1, 78120 Furtwangen, Germany
{sabine.janzen,wolfgang.maass}@hs-furtwangen.de
[2] University of St. Gallen, Dufourstrasse 40a, 9000 St. Gallen, Switzerland
{tobias.kowatsch,wolfgang.maass}@unisg.ch

**Abstract.** The design of ambient environments does not depend on technical issues exclusively but also on social aspects. There are several design specifications for ambient environments as well as development principles for the design of such systems, a design method should address. In this paper, we survey design methodologies considering the fulfilling of the design principles and their applicability for ambient environments. Because unprecedented, we introduce a methodology for *Content-Centered Design of Ambient Environments (CoDesA)* and apply this method in parts to an ambient bath environment.

**Keywords:** Design Method, Ambient Environment, Pre-Artifact, Narrative.

## 1 Introduction

Designing ambient environments affects several scopes of computer science and research on information systems. Chalmers et al. [1] define three distinct perspectives that have to be converged and blended within ambient environments: experience, engineering and theoretical perspective. An ambient environment should be designed based on a design method embedded in a design theory. To ensure the linkage of the emerging artefacts with real world situations and their users, appropriate experience-based approaches and evaluation cycles have to be applied. We itemized three scopes of design specifications for ambient environments: Physical Environment, Technology and Human. There are various design methods for information systems that examine the aspects Technology and Human but none of them integrates the perspective of the Physical Environment. We assume that the consideration of the physical environment is essential for designing ambient environments successfully. New ambient technologies have to be integrated into established business or home organization structures [2]. As derived from ethnographic studies, these technologies should be linked with other existing media such as paper and telephone at spots where media in general build clusters, for instance the phone shelf. In this sense, the integration of new technologies into home environments should represent the extension of current business or home "systems" by adding hardware devices and software

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 210–225, 2010.

agents for supporting intelligence [3,4]. To realize a seamless integration of intelligent business and home technologies, they need to be embedded into daily products and should be designed for business or domestic use [5]. In this context, the reliability of intelligent technologies is similar to well-known devices, e.g. a microwave oven, and they have to be easy to understand such as common devices in current homes. Furthermore, ambient technologies should enable the access to computing resources within and outside of time and places that are traditionally reserved for access of them [6]. Because no adequate design methodology for ambient environments exists, we introduce a methodology for *Content-Centered Design of Ambient Environments (CoDesA)* which considers the physical environment, technological aspects and the prospective users. Our design method is developed and applied in various AmI development projects for different domains, such as well-being, healthcare and shopping.

Next, we discuss existing design methodologies concerning their applicability for the design of ambient environments against several development principles. Then, we describe all required steps of CoDesA in Section 3 and illustrate them to that extent the current state of the ambient bathroom project allows. Thereafter, we exemplify the fulfillment of the aforementioned development principles by CoDesA in Section 4 and conclude this work with a summary and future work (Section 5).

## 2 Existing Design Methodologies

According to Walls et al. [7], a design process deals with three components: a design method, kernel theories and design process hypotheses. A design method "describes procedure(s) for artifact construction" (ibid., p. 43) whereas kernel theories from the natural or social sciences inform the design method. Last, design processes have to result in testable design process hypotheses, for example, theorems or proofs. A good design of an information system is not only concerned with technically issues but also with managerial ones that affect organizations and their individuals as stated by Markus & Keil [8]: "What developers think makes a good system - it works, it's technically elegant, and it's easy to use - is not necessarily what makes people want to use it - a good fit with their natural incentives and motivation" (ibid., p. 18). In this sense, there exist several development principles towards the design of information systems a design method should address:

- **Principle 1:** The evolving information system has to be "linked" with aspects of its usage within the real world [8,7], e.g., based on the specification of requirements, use cases and scenarios.
- **Principle 2:** A design method has to integrate diverse design steps and stakeholders (e.g. users, professionals) into the development [8,9,7], for instance, by means of creativity workshops with domain experts or users.
- **Principle 3:** A method should grant the option of discussions about diverse design proposals [8,7], e.g., supported by feedback loops.

– **Principle 4:** Concepts and prototypes have to be evaluated, in particular, after the implementation [8,7].
– **Principle 5:** System designs should be emphasized [8,9,7] and represented in a formalized way.
– **Principle 6:** A design method has to forward the development of functional (rapid) prototypes and their iteration [9,7], which enables the integration of feedback within the development.
– **Principle 7:** A method should represent guidance through a dialectic development process [9,7] in all design steps.

The principles can be assigned to several scopes, namely user requirements that are derived from kernel theories (principle 1), principles governing the development process (principles 2,3,6 and 7) and principles governing the design of a system (principle 4 and 5) [9,7].

According to Pries-Heje & Baskerville [10], we analyzed and compared diverse design methods concerning the seven principles and their applicability for ambient environments. We therefore derived a design method pattern that consists of five generalized phases from leading design science approaches [11,12,13,14,15]: (1) identification of problem and needs, (2) design of solution based on scenarios, use cases or requirements, (3) development of solution, (4) evaluation of solution, and (5) specification of a design theory. Based on this generalized structure of a design method, we were able to compare these design methods with regard to their composition and coverage.

In the following we discuss design methods that are relevant for our work. Taylor & Swan [2] specify use cases of ubiquitous systems based on interviews with stakeholders whereas Ross & Keyson [16] try to "sculpt" atmospheres based on the methodological development of design principles for tangible interaction. A Focus Group method is applied by Le Rouge & Niederman [17] for designing public health knowledge management architecture designs. Crabtree & Rodden [4] use ethnographic inquiries for studying routine work at home. Then, they derive requirements the development of ambient systems. For the design and development of domestic ubiquitous computing applications, Schmidt et al. [18] apply a multi-techniques investigation which combines methods of contextual inquiry, cultural probes, technology probes, scenarios-based participatory design and interviews in a qualitative research approach. Perrone et al. [19] define a stakeholder-centered approach for a conceptual modeling of communication-intensive applications that distinguish between problem and solution domains. The Interactive Scenario method is defined by Stroemberg et al. [20] as a promising tool for an early concept definition phase that increases participation of potential users in early stages of a concept design. "The interactive scenario method including improvisation and user acting seems to be very suitable for early-phase concept definition of complex systems that require *"off the desktop"* kind of activity (i.e. ubiquitous computing especially)" (ibid., p. 7). A different aspect is focused by a design method, called Interactive Thread [21]. This design method helps gathering detailed and contextualized data from a large user population while sharing interaction design methods with professional designers from different disciplines. Maiden et al. [22] present RESCUE, a scenario-driven requirements engineering process that integrates creativity techniques with different types of use cases and system context modeling. Buur et al. [23] wants to solve problems with the potential users rather than for the users. In this way the participants become part of the research process and contribute to the results through feedback and discussions. They combine two design methods that set the focus on skilled actions in the design of tangible user interaction - the Hands-Only Scenario and Video Action Wall. Chung et al. [24] developed a method for designing an initial and emerging pattern language for ubiquitous computing, consisting of pre-patterns describing application genres, physical-virtual spaces, interaction and techniques for managing privacy, and technologies for fluid interactions. Finally, Essence is a creativity method for software development that is based "on principles similar to role-playing games and improvisational theater" ([25], p. 549). The design method "melds creative sessions into agile development to employ development speed and flexibility throughout the project" (ibid.). Essence focuses on People, Product, Process and Project in each process step. Tab. 1 provides a survey of the analyzed design methods concerning the seven development principles for the design of information systems. Almost all considered design methods contain a step for specification of requirements, use

**Table 1.** Analyzed design methods concerning the seven principles for developing information systems

| | Principle 1 | Principle 2 | Principle 3 | Principle 4 | Principle 5 | Principle 6 | Principle 7 |
|---|---|---|---|---|---|---|---|
| *Taylor & Swan [2]* | • | • | n/a | - | - | - | - |
| *Ross & Keyson [16]* | • | ○ | n/a | • | - | - | • |
| *Le Rouge & Niederman [17]* | • | • | • | - | • | - | ○ |
| *Crabtree & Rodden [4]* | • | • | n/a | - | - | - | - |
| *Schmidt et al. [18]* | • | • | ○ | - | • | • | ○ |
| *Perrone et al. [19]* | • | • | ○ | - | • | • | ○ |
| *Strömberg et al. [20]* | • | • | n/a | - | - | - | - |
| *Mackay [21]* | • | ○ | n/a | - | - | - | - |
| *Maiden et al. [22]* | • | • | • | - | - | - | - |
| *Buur et al. [23]* | • | • | • | - | - | - | - |
| *Chung et al. [24]* | - | - | • | • | - | - | ○ |
| *Aaen [25]* | n/a | n/a | n/a | n/a | n/a | n/a | ○ |

(• =Complete; ○ =Partly; - =No match; n/a =Not applicable) cases etc. Similarly, the majority integrates diverse stakeholders such as users or experts, for instance, via creativity workshops. By contrast, only half of the methods provide guidance with the integration of results of discussions and evaluations of prototypes into the further development in the sense of feedback loops. In this context, only the methods of Ross & Keyson [16] and Chung et al. [24] cover a dedicated evaluation phase. Furthermore, only a few of the design methods focus on a formalization of the results within the implementation phase. Hence, the development of mock-ups or rapid prototypes to get early feedback in rapid evaluation cycles is a marginal phenomenon. In summary, none of the considered design methods fulfills all development principles for the design of information systems and thus, we describe our methodology in the next section.

## 3 A Methodology for the Design of Ambient Environments

Based on the literature review in Section 2, we identified seven development principles for the design of information systems. We analyzed existing design methods with regard to these principles, but none of these methods fulfills all development principles for the design of information systems. We therefore developed a *Methodology for Content-Centered Design of Ambient Environments (CoDesA)* (cf. Fig. 1). Our methodology consists of four phases: *Identification of Problem & Needs, Design of Solution, Development of Solution* and *Evaluation of Solution*. The latter phase includes also the *Specification of a Design Theory*. This structure is related to the method pattern that we have derived from prior design science research in information systems [11,12,13,14,15]. The four phases of CoDesA cover nine tasks. The methodology is elaborated in the next sections and specific inputs, outputs and involved stakeholders are provided for each task. Here, we present results for Tasks 1 to 4. Other tasks are target of our current research.

### 3.1 Identification of Problem and Needs

According to the aforementioned method pattern in Section 2, this phase targets the awareness of a problem and identification of the motivation to design a solution [13,14,15].

**Task 1 - Identification of Problem and Needs.** The objectives of this task are to identify the (business or private) problems and needs and to derive requirements for the expected solution. In this task, workshops with domain experts are conducted to identify a problem that has to be solved by the application of CoDesA and the resulting solution.

Consistent with the principles of Ambient Intelligence, we intend to integrate contents and information technology into a physical environment. The use case of an ambient bathroom represents a "far out" vision for direct user interactions with contents, and combines advanced content and knowledge management with

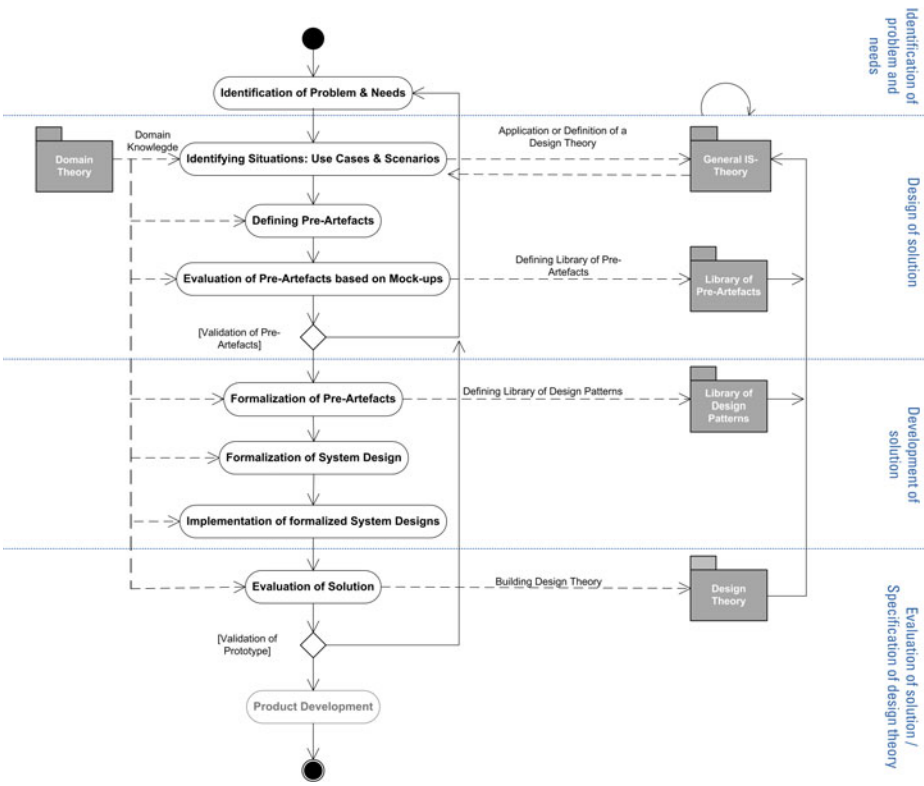



**Fig. 1.** Methodology for Content-Centered Design of Ambient Environments (CoDesA)

an Ambient Intelligence scenario and the Internet of Things - in a place which everybody is familiar with - the bathroom. In that use case, we want to show how users can interact with contents in physical environments in a way that leaves the dimension of "small windows to the infosphere".
**Involved stakeholders:** Domain experts and computer scientists
**Input:** -
**Output:** Description of (business or private) problems and/or needs

### 3.2 Design of Solution

In the second phase of CoDesA, solutions to meet the problems or to satisfy needs are specified through scenarios, use cases or requirements. Creativity workshops with different stakeholders are required [11,12,13,14,15].

**Task 2 - Identifying Situations: Use Cases & Scenarios.** This task focusses on the specification of situations in the domain of interest according to the problems and needs. More precisely, situations that address the defined problems or satisfy the needs are specified together with domain experts in creativity workshops. Situations are textual descriptions of different entities (objects, roles, information, background environments and services), which perform particular activities and interact with each other. These entities represent characteristic features of a corresponding class of situations within a domain, e.g., shopping situations. Thus, situations resemble frames [26], schemas [27], and use cases. The application of a domain theory integrates domain-specific common sense into the process of specification. A general IS theory constitutes the frame for the analysis of situations. In the case, that there exists no appropriate theory, a general IS theory can be defined incrementally during the application of CoDesA. In our case, we held a creativity workshop with a leading manufacturer of high-quality bathroom equipment. With the help of several creativity techniques, situations, or more precisely *narratives* have been specified according to the question: "Where and in which way do we face information and media in our daily life?" The results were transformed into a textual description, for instance:

> *"Anna gets site-specific weather information when she is brushing her teeth in the bathroom. Based on weather information and her calendar, free-time event suggestions are given, e.g. "Today, 8 p.m. - Sneak Preview at CinemaOne. Do you want to order tickets?"*

**Involved stakeholders:** Domain experts and computer scientists
**Input:** Description of (business or private) problems and/or needs
**Output:** Specification of situations in form of narratives, graphics or other non-formal descriptions

**Task 3 - Defining Pre-Artifacts.** The objective of this task is the transformation of the identified narratives into semi-formal structures, i.e. *Pre-Artifacts.* Pre-Artifacts are semi-formal perspectives of the narratives, i.e. the situations, and highlight the essential elements of each narrative [28]. There are two alternatives for deducing a Pre-Artifact from a narrative: (a) extraction of the essential concepts by means of a general IS theory and a domain theory, or (b) in the case, that there exists no appropriate IS theory, the narrative has to be analyzed according to several steps that will be elaborated in the following. The domain theory constitutes a frame for the process. Following case (b), there are five steps to deduce a Pre-Artifact from a narrative.

1. **Extraction of terms according to the Abstract Information Systems Model (AISM).** *AISM* [29] describes situations in ambient environments in form of generic concepts. Each narrative has to be analyzed concerning several aspects. *Agents* take specific *Roles* within a situation. *Services* provide *Information* that can be used by *Agents*. *Agents* and *Services* have specific intentions to participate in a situation. *Information* has several realizations, e.g., image, text, video. Communication takes place between *Service* and *Service, Service* and *Agent* as well as *Agent* and *Agent* in a specific context.
2. **Assignment of terms to categories.** The extracted terms have to be assigned to the essential categories of AISM: Agents & Roles, Information

and Services. The derivation of *Competency Questions* [30] for each term enables a later validation of the Pre-Artifact.

3. **Description of concepts and their relations.** Within this step, the categorized terms of the narrative and their relations have to be represented in AISM. This step covers the proper definition of each Pre-Artifact. It describes conceptual entities and their relations that are key elements within the narrative.
4. **Description of Pre-Artifacts.** As part of the library of Pre-Artifacts, the resulting Pre-Artifact has to be described according to a specific structure. The description consists of several components, for instance information and their realizations; goals, i.e. intentions of services or agents; roles and conditions; services etc. (cf. Tab. 3)
5. **Validation of Pre-Artifacts.** The quality and coverage of the Pre-Artifact can be evaluated by means of the defined Competency Questions. In the case that some questions cannot be answered, steps 1-5 have to be repeated.

In our bathroom project, we followed option (b) because of the absence of a general IS theory. Before we started to define Pre-Artifacts, we asked 46 potential early adopters of ambient bath environments, i.e. technical-savvy graduates, to rate the ten narratives which resulted from Task 2. This pretest was done in order to identify only those situations that are relevant from an end-user perspective. Therefore, on a seven-point Likert-scale ranging from strongly disagree (1) to strongly agree (7) and consistent with the compatibility measure of Moore and Benbasat [31] subjects had to indicate the degree to which each situation fits into their personal life. Results indicate that Narratives 1, 6 and 10 (see Appendix A) performed best with average values of 5.0, 5.1 and 5.3. Further, all of these ratings lie all significantly above the neutral test value of 4 by conducting a one-sample t-test. Based on the resulting three narratives, we defined the corresponding Pre-Artifacts. In the current work, we will describe this process for Narrative 1 (see Task 2). After the analysis of the narrative and extraction of the terms, we assigned them to the essential categories that are provided in Table 2. Furthermore, we derived the following Competency Questions:

- *Who is Anna?*
- *How is the weather in CityX today?*
- *Whats going on today?*
- *Where is Anna's calendar?*
- *In which format are the events for today available?*
- *Who is the ContentProviderX?*
- *What is the day of the week?*
- *Where is Anna?*

In a next step, we modeled the concepts and their relations according to AISM graphically and described the Pre-Artifact based on the structure that is shown in Tab. 3. Finally, we validated the Pre-Artifact by answering the Competency Questions successfully.

**Table 2.** Categorized Terms of Narrative 1

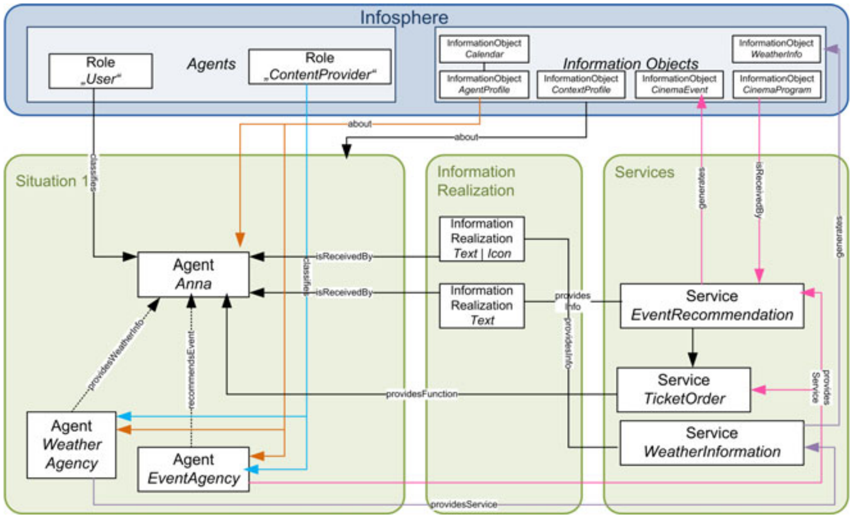
| Agents & Roles | Information | Services |
|---|---|---|
| Anna (Role: User) | Weather information | WeatherInformation |
| | Cinema event information | EventRecommendation |
| | Cinema program | TicketOrder |
| | Anna's calendar | |
| | Day of week | |
| | Time | |
| | Location | |

**Involved stakeholders:** Ontologists, IS engineer
**Input:** Specification of situations in form of narratives, graphics or other non-formal descriptions
**Output:** Representation of narratives in form of semi-formal Pre-Artifacts

**Task 4 - Evaluation of Pre-Artifacts Based on Mock-ups.** In this task, narratives in form of semi-formal Pre-Artifacts are evaluated with the objective to generate preliminary implications regarding user acceptance and marketing strategies. Because we are in the domain of ambient environments, we evaluate the narratives with the help of mock-ups at this early stage of the development process. Based on the Technology Acceptance Model and the more marketing-related work of Kamis et al. [32], perceived characteristics of the information and communication services of these narratives are measured, namely expected usefulness, intention to use and intention to subscribe. We focus on the services as such, because there is no concrete information available by this mock-up bases approach. Based on the results of the evaluation, domain experts are able to discuss the narratives more in detail and thus, are able to skip narratives that are not relevant any more. Further, if the analysis of the evaluation results is negative at all, Tasks 1 to 4 have to be repeated.

In the our project, we exemplify this task with the three information and communication services of Narrative 1 as shown in Tab. 3. We adopted the TAM measures from Wixom and Todd [33] and adapted the intention to subscribe measure from the purchase construct described by Kamis et. al [32]. Thirty-three technical-savvy subjects, i.e. potential early adopters, participated in this evaluation. Each narrative was presented with the help of dolls, a mock-up, i.e. a midget bathroom, and a slide show that exemplified the services (cf. Fig. 2). After each narrative was presented, the participants had to rate the services according to the measures on 7-point Likert-scales ranging from strongly disagree (1) to strongly agree (7). Results show that expected usefulness (3 items, Alpha=.93), intention to use and intention to subscribe for a monthly fee have average scores of 4.1, 5.5 and 2.7 for the Weather Information Service, 3.7, 4.8 and 2.9 for the Event Recommendation Service and 3.5, 4.0 as well as 2.4 for the Ticket Order Service. Thus, domain experts would rather focus on the weather information service due to the relative high user acceptance ratings but all of the services should be made available for free as the intention to subscribe for a monthly fee was perceived

**Table 3.** Pre-Artifact based on Narrative 1




| | |
|---|---|
| **Narrative** | *It's Thursday morning. Anna gets site-specific weather information when she is brushing her teeth in the bathroom. Based on weather information and her calendar, free-time event suggestions are given, e.g. "Today, 8 p.m. - Sneak Preview at CinemaOne. Do you want to order tickets?"* |
| **Information** | Calendar of Anna; AgentProfile of Anna; AgentProfile of EventAgency; AgentProfile of WeatherAgency; ContextProfile of Situation; CinemaProgram; CinemaEvent (InformationRealization.Text); WeatherInfo (InformationRealization.Text / InformationRealization.Icon) |
| **Goals** | Extract:<br>(a) Informing the user about the weather in New York<br>(b) Informing the user about potential events today<br>(c) Enabling the user to know how to order tickets for events |
| **Roles & Conditions** | *User* (Condition for role: Agent is authorized for using bathroom); *ContentProvider* (Condition for role: Agent is authorized for providing content in bathroom) |
| **Services** | EventRecommendation; TicketOrder; WeatherInformation |
| **Types of Communication** | – *(A) Agent.EventAgency ⟶ Agent.Anna (unidirectional) (symbolic)*<br>– *(B) Agent.WeatherAgency ⟶ Agent.Anna (unidirectional) (symbolic)*<br>– (C) Service.EventRecommendation ⟶ Service.TicketOrder (unidirectional)<br>– (D) Service.EventRecommendation ⟶ Agent.Anna (unidirectional)<br>– (E) Service.TicketOrder ⟶ Agent.Anna (unidirectional)<br>– (F) Service.WeatherAgency ⟶ Agent.Anna (unidirectional)<br>– Synchronal: D AND E<br>– Asynchronous: C AND(D AND E) AND F |
| **Competency Questions** | Extract:<br>– Who is Anna?<br>– How is the weather in New York today?<br>– Whats going on today?<br>– Where is Anna's calendar? |

**Fig. 2.** Mock-up-based Evaluation of Pre-Artifacts

significantly negative at the .001 level when conducting a t-test with 4 being the neutral test-value.
**Involved stakeholders:** Potential early adopters of ambient bath environments, domain experts
**Input:** Representation of narratives in form of semi-formal Pre-Artifacts
**Output:** Preliminary implications for Pre-Artifacts regarding user acceptance and marketing strategies

### 3.3 Development of Solution

The solution consists of a library with evaluated Pre-Artifacts, i.e. semi-formal structures of the identified narratives. In this phase, the defined Pre-Artifacts have to be transformed into an ontological structure. These formalized structures will be stored in a library of design patterns. Based on these design patterns, requirements and a formalized system design can be derived. Furthermore, the formalized system design is implemented as service architecture [15,13]. This phase of CoDesA will be applied in the second and third year of the bathroom project; therefore, we exemplify the following tasks without actual results. In this contribution, we will rather give a rough idea of the development of the solution.

**Task 5 - Formalization of Pre-Artifacts.** The objective of this task is to formalize the semi-formal Pre-Artifacts in order to derive processable design patterns and specifications. Christopher Alexander introduced the term "design pattern" for shared guidelines that help to solve architectural design problems [34]. Later, the potential for reusing ontological structures through a pattern-based approach was investigated [35,36]. There are several opportunities to formalize the identified Pre-Artifacts, for instance, their transformation into Prototypical Ontology Design Patterns (PODPs) [28] that are derived and formally modeled by reusing Ontology Design Patterns (ODPs) grounded in DOLCE [36]. Furthermore, a formalization based on UML is conceivable. Both possibilities enable the specification of requirements and components for the final system design.
**Involved stakeholders:** Computer scientists and ontologists
**Input:** Representation of narratives in form of semi-formal Pre-Artifacts (library of Pre-Artifacts)
**Output:** Formalized representation of Pre-Artifacts (library of design patterns)

**Task 6 - Formalization of System Design.** This task focusses on the specification of the formalized system design. It covers supporting architectures, data and service infrastructures. The system design is formalized based on the library of design patterns. Furthermore, the design of the later implementation is completed with adequate architectures (e.g. Service Oriented Architectures (SOA), RESTFul) and data infrastructures, for instance, persistence layers of ontology repositories.
**Involved stakeholders:** Computer scientists
**Input:** Library of design patterns
**Output:** Specification of formalized system design

**Task 7 - Implementation of Formalized System Designs.** Here, the formalized system design is transformed into machine-processible code. Thereafter, this code has to be linked with the hardware components integrated in the physical environment. As a result, the prototype of the ambient environment is finalized.
**Involved stakeholders:** Computer scientists
**Input:** Specification of formalized system design
**Output:** Prototype of an ambient environment

### 3.4 Evaluation of Solution and Specification of Design Theory

In the last phase, the prototypical ambient environment has to be evaluated empirically. As a result, design theories can be developed or further specified [11,12,13,14,15]. At the same time, the evaluated prototype constitutes the basis for a deployment in real life - the product development in the free economy. This phase of CoDesA will also be applied in the third year of the bathroom project.

**Task 8 - Evaluation of Solution.** With the help of an empirical evaluation, the implemented prototype is tested against the requirements specified by the Pre-Artifacts. In case the prototype is not successful, Task 5 to 8 have to be repeated. Learning from the results, design theories can be built up or further specified.
**Involved stakeholders:** Computer scientists, end users such as information and communication service providers and their customers
**Input:** Prototype of an ambient environment
**Output:** Valid ambient environment prototype tested against the specified requirements

**Task 9: Product Development.** The final task of the method covers the serial production of the evaluated ambient environment prototype in an economic sense by a domain-specific company.
**Involved stakeholders:** Domain experts
**Input:** Valid ambient environment prototype tested against the specified requirements
**Output:** Serial production of the ambient environment

## 4 Discussion

We proposed seven development principles for the design of information systems in Section 2 and we analyzed several design methods concerning these principles. None of the design methods has fulfilled all development principles. Therefore, we discuss the fulfillment of these principles by CoDesA in the following:

1. The resulting information system is linked to the real world through creativity workshops that are part of CoDesA. The narrative approach ensures further a representation of real world situations. The evaluations are conducted based on physical mock-ups and prototypes. This integrates the concept of physical environments into user studies and grounds the design process in practice outside the scientific world.
2. CoDesA integrates several stakeholders into diverse design steps during the development process, e.g., domain experts, users and ontologists. In this way, diverse prospects, intentions and expertises are integrated into the design process, which defends from "lack of objectivity".
3. Our method provides the option of discussions and iterations after the evaluation Tasks 4 & 8. Depending on the results of the evaluation, a feedback loop can be initiated for reengineering the artefacts of the preceding tasks.
4. CoDesA schedules the evaluation of the identified Pre-Artifacts based on a mock-up (cf. Task 4) as well as the validation of the resulting prototype (cf. Task 8). Furthermore, an additional pretest with end-users can be conducted to rank the defined narratives if required. In summary, CoDesA covers a continuous *build & evaluate cycle* during the whole design process [11].
5. The identified Pre-Artifacts as well as the derived system design are represented in a formalized way and stored in a repository, i.e. the library. This ensures standardization and adequate reusability within further design processes.
6. CoDesA forwards the development of rapid prototypes and their iteration, for instance the development of mock-ups for evaluating the Pre-Artifacts (cf. Task 4). Within the design process, rapid results are generated that can be analyzed for detecting design failures very early.
7. Our method gives guidance during the whole development process in all design steps - from the identification of (business or private) problems and needs to the serial production of the ambient environment. CoDesA accompanies designers of ambient environments beyond the specification of requirements and helps to "pursue the path".

As a limitation of the current work, we could only apply CoDesA to Tasks 1 to 4. We are therefore not able to exemplify results of the other task. This will be part of future work as discussed in the following section.

## 5 Conclusion and Future Work

A good design of an ambient environment does not depend on technical issues exclusively but also on aspects concerning the human and the physical environment. Ambient Intelligence implies modularized computing environments and specific interfaces and therefore requires several specifications and development principles for the design of such systems. In this article, we proposed seven development principles for the design of information systems, which should be addressed by a design method [8,9,7]. We further analyzed capabilities of different design methodologies concerning these principles. Because none of the considered design methods fulfilled all development principles, we introduced a new methodology for *Content-Centered Design of Ambient Environments (CoDesA)* which fulfills all the principles. The linkage of the resulting information system with the real world is ensured through the integration of different stakeholders into the design process and through a narrative-based approach that represents real world user interactions. CoDesA forwards the development of rapid prototypes as well as their evaluation. Furthermore, our method gives guidance during the whole development process in all design steps. Currently, we are able to present experiences in applying Task 1-4. The application of further tasks will take place until 2012 as part of the bathroom project.

In our future work, we will therefore proceed with several tasks: (1) ontological formalization of Pre-Artifacts, (2) development and evaluation of a prototype of an intelligent bathroom based on the results the first four tasks, and (3) automatic mapping of Competency Questions, narratives, and terms onto Pre-Artifacts.

## Acknowledgement

This paper resulted from project Interactive Knowledge Stack (IKS) (FP7 231527) co-funded by the European Commission.

## References

1. Chalmers, D., Chalmers, M., Crowcroft, J., Kwiatkowska, M., Milner, R., ONeill, E., Rodden, T., Sassone, V., Sloman, M.: Ubiquitous computing: Experience, design and science (2006)
2. Taylor, A.S., Swan, L.: Artful systems in the home. In: CHI 2005: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 641–650. ACM Press, New York (2005)
3. Bonino, D., Castellina, E., Corno, F.: Dog: An ontology-powered osgi domotic gateway. In: ICTAI (1), pp. 157–160 (2008)
4. Crabtree, A., Rodden, T.: Domestic routines and design for the home. Comput. Supported Coop. Work 13(2), 191–220 (2004)
5. Edwards, W.K., Grinter, R.E.: At home with ubiquitous computing: Seven challenges. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) UbiComp 2001. LNCS, vol. 2201, pp. 256–272. Springer, Heidelberg (2001)
6. Cousins, K.C., Varshney, U.: Designing ubiquitous computing environments to support work life balance. Commun. ACM 52(5), 117–123 (2009)
7. Walls, J.G., Widmeyer, G.R., Sawy, O.E.: Building an information system design theory for vigilant eis. Information Systems Research 3(1), 36–59 (1992)
8. Markus, M.L., Keil, M.: If we build it, they will come: Designing information systems that people want to use. Sloan Management Review 35, 11–25 (1994)

9. Markus, L.M., Majchrzak, A., Gasser, L.: A design theory for systems that support emergent knowledge processes. MIS Quarterly 26(3), 179–212 (2002)
10. Pries-Heje, J., Baskerville, R.: The design theory nexus. MIS Quarterly 32(4), 731–755 (2008)
11. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
12. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decis. Support Syst. 15(4), 251–266 (1995)
13. Pfeffers, K., Tuunanen, T., Gengler, C.E., Rossi, M., Hui, W., Virtanen, V., et al.: The design science research process: A model for producing and presenting information systems research. In: Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, CA, USA, pp. 83–106 (2006)
14. Rossi, M., Sein, M.K.: Design research workshop: A proactive research approach (2003)
15. Kuechler, W.L.J., Vaishnavi, V.K.: An expert system for dynamic re-coordination of distributed workflows. Expert Syst. Appl. 34(1), 551–563 (2008)
16. Ross, P., Keyson, D.V.: The case of sculpting atmospheres: towards design principles for expressive tangible interaction in control of ambient systems. Personal Ubiquitous Comput. 11(2), 69–79 (2007)
17. Le Rouge, C.M., Niederman, F.: Information systems and health care xi: Public health knowledge management architecture design: A case study. Communications of the Association for Information Systems 18 (2006)
18. Schmidt, A., Terrenghi, L., Holleis, P.: Methods and guidelines for the design and development of domestic ubiquitous computing applications. Pervasive Mob. Comput. 3(6), 721–738 (2007)
19. Perrone, V., Bolchini, D., Paolini, P.: A stakeholders centered approach for conceptual modeling of communication-intensive applications. In: SIGDOC 2005: Proceedings of the 23rd annual international conference on Design of communication, pp. 25–33. ACM, New York (2005)
20. Strömberg, H., Pirttilä, V., Ikonen, V.: Interactive scenarios—building ubiquitous computing concepts in the spirit of participatory design. Personal Ubiquitous Comput. 8(3-4), 200–207 (2004)
21. Mackay, W.E.: The interactive thread: exploring methods for multi-disciplinary design. In: DIS 2004: Proceedings of the 5th conference on Designing interactive systems, pp. 103–112. ACM, New York (2004)
22. Maiden, N., Manning, S., Robertson, S., Greenwood, J.: Integrating creativity workshops into structured requirements processes. In: DIS 2004: Proceedings of the 5th conference on Designing interactive systems, pp. 113–122. ACM, New York (2004)
23. Buur, J., Jensen, M.V., Djajadiningrat, T.: Hands-only scenarios and video action walls: novel methods for tangible user interaction design. In: DIS 2004: Proceedings of the 5th conference on Designing interactive systems, pp. 185–192. ACM, New York (2004)
24. Chung, E.S., Hong, J.I., Lin, J., Prabaker, M.K., Landay, J.A., Liu, A.L.: Development and evaluation of emerging design patterns for ubiquitous computing. In: DIS 2004: Proceedings of the 5th conference on Designing interactive systems, pp. 233–242. ACM, New York (2004)
25. Aaen, I.: Essence: Facilitating agile innovation. In: XP, pp. 1–10 (2008)
26. Minsky, M.: A framework for representing knowledge. Technical report, Cambridge, MA, USA (1974)

27. Schank, R.C., Abelson, R.P.: Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures. L. Erlbaum, Hillsdale (1977)
28. Maass, W., Janzen, S.: A pattern-based ontology building method for ambient environments. In: Workshop on Ontology Patterns - WOP 2009 at the 8th International Semantic Web Conference (ISWC 2009), Washington, DC, USA (2009)
29. Maass, W., Varshney, W.: A framework for smart healthcare situations and smart drugs. In: SIG-Health Pre-AMCIS Workshop at the 15th Americas Conference on Information Systems (AMCIS 2009), San Francisco, USA (2009)
30. Grueninger, M., Fox, M.S.: The role of competency questions in enterprise engineering. In: Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice (1994)
31. Moore, G.C., Benbasat, I.: Development of an instrument to measure the perceptions of adopting an information technology innovation. Information Systems Research 2(3), 192–222 (1991)
32. Kamis, A., Koufaris, M., Stern, T.: Using an attribute-based decision support system for user-customized products online: An experimental investigation. MIS Quarterly 32(1), 159–177 (2008)
33. Wixom, B.H., Todd, P.A.: A theoretical integration of user satisfaction and technology acceptance. Info. Sys. Research 16(1), 85–102 (2005)
34. Alexander, C.: The timeless way of building. Oxford University Press, New York (1979)
35. Clark, P., Thompson, J., Porter, B.: Knowledge patterns. In: Proc. of KR 2000, pp. 591–600. Morgan Kaufmann, San Francisco (2000)
36. 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)

## Appendix A: Extract of Identified Narratives (Task 2)

All narratives can be requested from the authors.

- **Narrative 6.** Fortunately, it is weekend. Robert has taken a shower listening to his favorite music. Leaving the bathroom, Anna flits into the room. Robs music has become silent and Anna is welcomed by music from her own music collection. The music starts at the point in the playlist where Anna stopped listening the evening before. After a while, she says Stop music and the song falls silent. Anna wants to see her personal news collage while taking a shower.
- **Narrative 10.** Robert is brushing his teeth in the morning; he listens to the news on the radio. Then, he takes a shower. Now, the news messages are displayed in form of pictures and text at the glass door of the shower.

# Extending the Design and Engineering Methodology for Organizations with the Generation Operationalization and Discontinuation Organization

David Aveiro[1,2], A. Rito Silva[2,3], and José Tribolet[2,3]

[1] Exact Sciences and Engineering Centre, University of Madeira,
Caminho da Penteada 9000-390 Funchal, Portugal
[2] Center for Organizational Design and Engineering, INESC-INOV
Rua Alves Redol 9, 1000-029 Lisboa, Portugal
[3] Department of Information Systems and Computer Science, Instituto Superior Técnico,
Technical University of Lisbon
{david.aveiro,jose.tribolet,rito.silva}@ist.utl.pt

**Abstract.** We propose an extension for the Design and Engineering Methodology for Organizations – DEMO – to support organization and model change dynamics: the ontological model of the generic *G.O.D. organization*, considered to exist in every organization and being responsible for the Generation, Operationalization and Discontinuation of organization artifacts – e.g., actor role *pizza deliverer* – as a consequence of the process of handling unexpected exceptions causing dysfunctions in the organization's activity. The G.O.D. organization keeps a thorough trace of all acts regarding the diagnosis of problems (dysfunctions) and the design and operationalization of their respective solutions. Such an historical trace provides useful information to each organizational engineering process (OEP) handling unexpected exceptions. Another benefit is to provide a base for a constantly updated model of organizational reality, useful to guide the general activity of organization agents and to provide up to date information of current organizational reality to each OEP.

**Keywords:** organizational design, organizational engineering, model, dysfunction, unexpected exception.

## 1 Introduction

Our initial research efforts had the general purpose of understanding and clarifying what the *function perspective of an organization* should be. Normally, the function concept is associated with behavior, *activity* or operation of an organization or of a certain organizational unit like a marketing or IT department, normally responsible for the respective function [1]. In [2] we find that the function perspective means looking at a system from the point of view of the using system, in terms of provided functionality, i.e., kinds of behavior that can be caused. We regarded this to be an incomplete use of the term function. As a result of a review that we undertook on how this concept is used in such diverse areas as enterprise engineering, information systems, biology, sociology and philosophy, we found that, besides the aspect of *behavior*, also central to

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 226–241, 2010.

the function concept is the *normative* aspect (e.g., [3]), that is, the existence of certain normally expected values – *norms* – for certain vital properties of a system. In an organization, *deviations* from such norms imply a state of *dysfunction* that can possibly compromise its *viability*.

We present examples from the scenario of a library, introduced in [2] and extended in our research, as to better accommodate concepts we're proposing. The main activities of the library are *book loaning* and *offer book history courses*. We can define three norms: (1) *min average number of registrants in book history courses 1 week before start is 14*, (2) *min total income per month is 900€* and (3) *max loan declines per week is 30*. A possible dysfunction in the second norm is: *average number of registrants in book history courses is 7 on March 23th 2009*. This can be a very serious situation because, as a consequence, the library may lose income needed to acquire enough resources and eventually go bankrupt, closing down the business.

Dysfunctions will have a *cause* which may be *expected* or *unexpected*. If the cause is expected, certain *resilience strategies* may already exist that can be activated to eliminate or circumvent dysfunctions [4], [3]. Continuing in the library scenario, *it is known that sometimes clients of the library are not enough to fill book history courses and some extraordinary advertising needs to be done, which solves the issue of not enough students to cover expenses*. If the cause is unknown we will be in the presence of an *unexpected exception*. This unexpected exception will have to to be *handled* so that its concrete nature is detected and actions are undertaken that either eliminate or circumvent it, solving the dysfunction. The first time not enough students were registering it had, as a consequence, a dysfunction in the norm of *min total income per month*. As a result of this dysfunction, a handling process was initiated to detect the root cause (unexpected exception) of lack of income, namely: *lack of advertisement of courses*. The resilience strategy *distribute course fliers* was designed and chosen as solution to avoid the referred (previously unexpected) exception.

The handling of unexpected exceptions constitutes another central aspect of the function perspective, namely *change* through the *(re)Generation*, *Operationalization* and *Discontinuation* of organizational artifacts which will eliminate or circumvent the determined cause of dysfunction. We consider an *organization artifact* (OA) as a construct of an organization like a business rule (e.g. "if invoice arrives, check list of expected items") or an actor role (e.g. library member). Change of OAs to handle dysfunctions is considered a special kind of dynamics that – inspired in philosophy literature on this subject – we call *microgenesis* [5]. We find that change is also driven by the detection of *opportunities of improvement* which will increase the viability of an organization and place it ahead of *competition* [6]. This is proactive change, as opposed to reactive change in the cases of resilience and microgenesis.

The focus of our research is on modeling two aspects of reactive change: (1) the resilience dynamics of strategies to solve known exceptions causing dysfunctions and (2) the microgenesis dynamics of handling unexpected exceptions also causing dysfunctions. This paper, in turn, focuses on the second aspect: microgenesis dynamics.

In section 2 we develop our research problem and related work. Section 3 presents our proposed extension to DEMO, in order to express microgenesis dynamics of organizations. Section 4 concludes this paper with a critical review on work done and future lines of research.

## 2 Problem, Motivation and Related Work

Above findings helped us to identify two relevant and closely interrelated more focused problems. On one hand, *a large amount of time is lost, in organizations, in the handling of unknown exceptions causing dysfunctions*. On another hand, *current Organizational Engineering (OE) approaches seem to lack in concepts and method for a continuous update of organizational models, so that they are always up to date and available as a more useful input for the process of continuous change of organizational reality and decision on possible evolution choices*. We focus on these problems in the context of small timely changes, as opposed to large impact changes in the context of IT/IS projects, mergers, acquisitions and splittings of organizations. *Why are these problems relevant?* As we saw in the example of the library dysfunctions like lack of enough income can compromise a whole organization. Also, exception handling can sometimes take almost half of the total working time, and the handling of, and recovering from, exceptions is expensive [7].

*What causes can be identified relating to these problems?* We identify what seems to be a lack of capture and management of relevant information of past unknown exceptions and their handling. Many events (which were previously unknown exceptions) can have already been known or expected in the past, but can be (frequently) forgotten and become again unexpected (unknown) due to: (1) absence of explicit representation of (i) specific exceptions and actions that were executed (in an Ad hoc and unstructured way) for their handling and (ii) engineered OAs to solve them [8] or (2) removal of human agents from a certain organizational actor role which had established and tacitly memorized specific (informal) rules to handle specific exceptions occurring in such actor role [7].

It seems that the root problem for the above mentioned interrelated problems is an *absence of concepts and method for explicit capture, and management of information of exceptions and their handling, which includes the design and selection of OAs that solve caused dysfunctions*. Not immediately capturing this handling and the consequent resulting changes in reality and the model of reality itself, will result that, *as time passes, the organization will be less aware of itself than it should be*, when facing the need of future change due to other unexpected exceptions.

In terms of related research, the lack of awareness of organizational reality has been addressed in [9], with the coining of the term “Organizational Self-Awareness” (OSA). This construct has been further refined in [10] and [11]. OSA stresses the importance and need of continuously available, coherent, updated and updateable models of organizational reality. A recently proposed research discipline named Organizational Design and Engineering (ODE) [12], also defends this and further raises the importance of capturing and making organizational history and lessons learned available to organizational actors. OSA, and ODE claim that current OE approaches have the shortcoming of lacking in concepts and methods for a continuous update of models of organizational reality, aligned with the continuous change happening in the real terrain. However, both OSA and ODE have, for the most part, only addressed the issues of identification and formulation of this problem and, in terms of solution, mostly the aspect of representation, leaving the change aspect as future work.

This shortcoming of lack of continuous update of models aligned with the continuous change of reality has been addressed, by and large, in research and practice in the context of Workflow Management Systems (WfMS) – see, for example, [8] and [13]. However, current solutions assume that an organization will be using a WfMS, which will not be the case of many organizations. And, even in the case of organizations using WfMS, relevant activities may happen outside of IT context and we may also want to address exceptions related to them.

From our review and proposal of a broader notion of the function perspective and related insights brought from Complex Adaptive Systems (CAS) literature, we find that we may have two main types of change dynamics: *resilience* and *microgenesis.* From CAS [4] (p. 33) and philosophy [3], we find that systems maintain an internal model of the world (of themselves and the environment) so that they can activate specific resilience strategies to react, appropriately and in time, to certain known exceptions or fluctuations in critical norms that guarantee the system's viability. We also find that a system adapts with incremental changes [5], having as a main purpose to survive and evolve among competition, by having credit mechanisms which favor changes (adaptations) that increase the system's viability and constitute criteria of measuring success [4] (p. 34), [14] (p. 5). One of the premises from CAS theory is that, to solve new exceptions, "rule pieces" that constitute current resilience strategies that solve similar exceptions may be re-utilized to build new resilience strategies or new organization artifacts to solve the new exceptions. From unexpected exception handling in WfMS [8] and ODE [12], we find that information on the history of organization change is an essential asset in moments where change is again needed, i.e., in microgenesis dynamics.

Modeling resilience and microgenesis dynamics and keeping a systematic history of their execution is deemed as a solution to our main research problem, so that exception handling and organization change is more efficient and effective. Microgenesis is the main focus of our main research project, but to precisely specify its dynamics we needed to precisely specify resilience dynamics. This has been the focus of another report to be published elsewhere. In this paper we will present a brief summary of essential notions proposed for specifying resilience dynamics needed to, in turn, specify the notions of our main focus in this paper: microgenesis dynamics.

To ground our solution, we decided to narrow our research focus, choosing a particular OE approach, namely, the Design & Engineering Methodology for Organizations (DEMO) [2]. From several approaches to support OE being proposed, DEMO seems to be one of the most coherent, comprehensive, consistent and concise [2]. It has shown to be useful in a number of applications, from small to large scale organizations – see, for example, [15] and [16] (p. 39). Nevertheless, DEMO suffers from the shortcoming referred above. Namely, DEMO models have been mostly used to devise blueprints to serve as instruments for discussion of broader scale organizational change or development/change of IT systems [16] (p. 58) and does not, yet, provide modeling constructs and a method for a continuous update of its models as reality changes, driven by exceptions (microgenesis) nor for the continuous control (resilience) that we need to exert on organizations to guarantee viability.

Contributions of our research – presented in the next sections - extend DEMO, with the devising of concepts and a method that systematically address the elicited problem. While proceeding, the reader which is unfamiliar with DEMO is advised to also consult [2] or [15] or other publications in: www.demo.nl.

## 3 Applying DEMO to Specify the G.O.D. Organization

One of the main points of the proposed solution in our research is that every organization implicitly has a G.O.D. Organization, responsible for changing the organization system's structure, composition and production. We borrow the ontological system definition from [17] (citing [18]) which concerns the construction and operation of a system. The corresponding type of model is the *white-box model*, which is a direct conceptualization of the ontological system definition presented next. Something is a system if and only if it has the next properties: (1) *composition*: a set of elements of some category (physical, biological, social, chemical etc.); (2) *environment*: a set of elements of the same category, where the composition and the environment are disjoint; (3) *structure*: a set of influencing bonds among the elements in the composition and between these and the elements in the environment; (4) *production*: the elements in the composition produce services that are delivered to the elements in the environment.

DEMO doesn't currently contemplate the aspect of change of the organization system properties above. The "business" of the G.O.D. organization is precisely this kind of change which we have previously labeled as microgenesis. As we also saw previously, microgenesis will be triggered by dysfunctions and implies the execution of organizational engineering processes. After detection of the causing exception, organization artifacts will be generated and operationalized that handle or circumvent such exception, eliminating the caused dysfunction. We will now present our proposal for the ontological model of the G.O.D. organization (GO). We will present it, in a intertwined manner, and in three parts, focusing mostly on the State Model (SM).

In the first part we will focus on the aspect of handling an unexpected exception which will consist in monitoring, diagnose and recovery actions. We propose a table named Monitoring, Diagnosis, Exception and Recovery Table (MDERT) to consolidate information highly useful for this part of microgenesis dynamics.

On the second part we focus on the issue of the life cycle of organization artifacts and on the third we briefly touch on the issues of allocating and training organization agents (human and/or IT based) so that the generated organization artifacts can be operationalized.

### 3.1 Monitoring, Diagnosis, Exception and Recovery

Following the defined functions of the unexpected exception handling cycle found in [8] and presented in Figure 1, in an OEP, monitoring, diagnosis and recovery acts will be executed in an intertwined and ad-hoc manner, leading to the creation of the respective facts. In [8] this cycle begins with the so called exception detection function.

We argue that the majority of the handling cycle is indeed dedicated to understand and detect (diagnose) the unexpected exception. So it seems exception detection is not the proper name for this first function and that we should use, instead, dysfunction observation, which is the name we give to the first stage of what we call the dysfunction handling process (DHP) – the focus of resilience dynamics in another report of our research – which, in turn, can then initiate an OEP. We find that the area of unexpected exception handling in WfMS – to our knowledge, the most advanced in the characterization of exceptions in the functioning of organizations – seems to suffer from the shortcoming of not clearly separating the notions of dysfunction and exception.

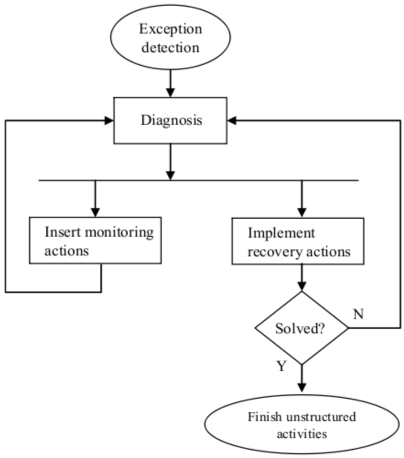



**Fig. 1.** Exception handling cycle

This shortcoming has also existed in the area of IT Service Management (ITSM), where, with the maturing of standards like ITIL, a clear separation has been done between the concepts of incident and problem, each having a separate process for their management [19]. The processes of incident management and problem management in ITSM have a similar nature, respectively, with our proposals for the DHP and OEP. So one of the contributions of our research is bringing a separation of concerns between dysfunction (incident) handling and unexpected exception (problem) handling to the field of organizational engineering, using DEMO as a base.

Figure 4 presents part 1 of the GO's State Space Diagram (SSD), the formulation of the respective part of the GO's SM. To keep a record of facts resulting from the actions of an OEP, we find object classes, MONITORING, DIAGNOSIS, EXCEPTION KIND and RECOVERY and associated binary fact types and unary result kinds:

- *[monitoring] done in [organizational engineering process]*
- *[monitoring] has been done*
- *[diagnosis] made as a result of [monitoring]*
- *[diagnosis] has been made*
- *[exception kind] detected in [diagnosis]*
- *[exception kind] has been detected*
- *[recovery] done in [organizational engineering process]*
- *[recovery] has been done*

Table 2, that we propose to name as the *Monitoring, Diagnosis, Exception and Recovery Table* (MDERT), presents fact instances of the above mentioned object classes and fact types, for the case of the library.

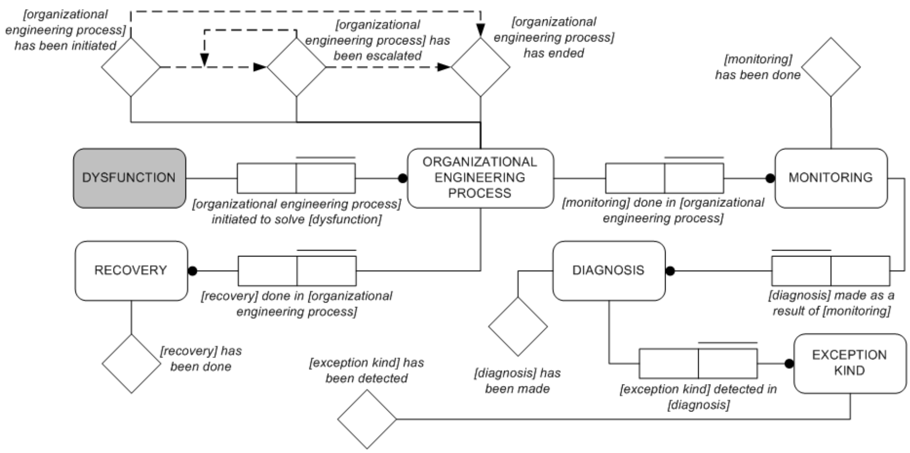



**Fig. 2.** G.O.D. Organization's SSD – part 1

External object class DYSFUNCTION and fact type: *[organizational engineering process] initiated to solve [dysfunction]* serve as a bridge between the Control Organization – that realizes resilience dynamics – and the G.O.D. Organization. An OEP will necessarily be initiated because of some dysfunction in some viability norm. It may happen that such viability norm was tacit (not formally explicit) but, the moment that it becomes relevant to solve its related dysfunction, it has to be made explicit. This means that the respective organization artifacts (new measure, viability norm, etc.) will need to be generated in the context of such OEP. All OEPs will start with some monitoring action that can have, as a result, some diagnosis action. A diagnosis can, in turn, lead to the detection of a root cause of the dysfunction – an exception kind. In parallel, recovery actions can be immediately executed to eliminate the dysfunction, while monitoring and diagnosis progress.

**Table 1.** Dysfunctions affecting viability norm VN01 of the Library

| **viability norm** | | **measurement** | | | **dysfunction** | **solved** | **solution** |
|---|---|---|---|---|---|---|---|
| min total income per month | 900 € | total income per month | 805 € | Jun 15 2008 | DF01 | Mar 10 2009 | OEP01 |

Returning to the library scenario, we present, from its Dysfunctions Table the only dysfunction affecting viability norm VN01 – DF01. In the first three lines of Table 2 we find all monitoring and diagnosis facts that led to the detection of the exception kind – *abnormal high rate of loan requests due to exams season* – as causing DF01. We find also a recovery fact, reflecting a recovery action to immediately solve the dysfunction - *increase value of property max_copies_in_loan during exam season*.

**Table 2.** Monitoring, Diagnosis, Exception and Recovery Table of the Library

| handling process | dysfunc-tion | monitoring | diagnosis | exception kind | recovery |
|---|---|---|---|---|---|
| OEP01 | DF01 | verify loan register | loans have been decreasing in the last year | - | - |
| | | ask secre-tary's opinion on decrease in loan requests | the secretary informed that there are less visitors than usual, but in some months many students have their requests declined due to the max_copies_in_loan limit | - | - |
| | | ask students why they need so many books | every three months its exam season in nearby colleges and they need more books than normal | abnormal high rate of loan requests due to exams season | increase value of property max_copies_in_loan during exam season |
| OEP02 | DF02 | do street survey to find cause | most people (70%) didn't saw our add at the newspaper nor the radio, some interested people (10%) suggested to distribute fliers and posters in bars and restaurants nearby | insufficient advertising | delay course start dates for 2 weeks; design, print and distrib-ute fliers announcing courses |
| OEP03 | DF03 | do street interview to find cause | great majority of people (90%) show total lack of interest in the course | general lack of interest in courses | delay course start dates until having enough registrants |
| OEP04 | DF06 | observe previous resilience and OEP logs | it appears that the general lack of interest in courses is getting worse as delaying courses start is not enough and more drastic actions are needed | general lack of interest in courses | cancel courses and transfer students to prevent dys-function in monthly income |

We next present in Table 3, again from the DFT of the library, dysfunctions affecting viability norm VN02. We find, also in Table 2, actions of OEP02, initiated to solve DF02. While facing dysfunction DF02, the library manager decided to create a questionnaire and request some students to do a street survey to find out the cause for so few applications for the book history course. In parallel he decided, as a recovery action, to delay the start date of the courses for two weeks. As a result of this monitoring action it was found (diagnosed) that *most people (70%) didn't saw our add at the newspaper nor the radio, some interested people (10%) suggested to distribute fliers and posters in bars and restaurants nearby*. The manager then concluded that the exception kind causing DF02 was *insufficient advertising*. Then he immediately proceeded, as other recovery actions, to request to the secretary to design and print course fliers and for her to request some students to distribute them in nearby restaurants and bars. As a result of this OEP some organization artifacts were generated, formalizing the referred recovery actions and constituting a resilience strategy – called *R02 - distribute course fliers*. The topic of generation of organization artifacts is addressed in detail ahead.

**Table 3.** Dysfunctions affecting viability norm VN02 of the Library

| viability norm | | measurement | | | dysfunction | solved | solution |
|---|---|---|---|---|---|---|---|
| min average # of registrants in book history courses 1 week before start | 14 | average # of registrants in book history courses 1 week before start | 9 | Sep 12 2008 | DF02 | Sep 26 2009 | OEP02 |
| | | | 11 | Jan 4 2009 | DF03 | Jan 25 2009 | OEP03 |
| | | | 10 | Feb 8 2009 | DF04 | Feb 15 2009 | RS02 |
| | | | 8 | Mar 23 2009 | DF06 | Apr 7 2009 | OEP04 |
| | | | 7 | Apr 18 2009 | DF07 | May 14 2009 | RS04 |

Regarding OEP03, it was initiated to solve DF03 where resilience strategy RS02 did not stop violation of viability norm VN01. So the manager decided to create another questionnaire to determine the cause while, again, delaying start date of the courses as a recovery action. It was found that, contrary to some months ago, the *great majority of people (90%) show total lack of interest in the course*. This determined the causing exception kind as being *general lack of interest in courses*. It was decided, as a recovery action, to keep delaying courses start dates until having enough registrants. As ahead will be seen, this action gave origin to resilience strategy *RS03 - delay courses start*.

OEP03 was initiated to solve DF03 where the manager, by observing previous resilience and OEP logs, found that neither resilience strategy RS02 nor RS03 could solve DF03 and concluded that lack of interest was simply growing and the best option would be cancel part of the courses and transfer students from the closed courses to the ones still planned to happen. In this manner the minimum average number of registrants would be achieved and a dysfunction in monthly income would be prevented as room renting expenses would not be covered with so few students.

As it should be clear by now, keeping a record of past monitoring, diagnosis and recovery facts aids the creative and ad-hoc process of handling current unexpected exceptions affecting the same viability norm. In our example, we saw how the idea of a street survey was reused from OEP02 to OEP03. Thus, similar (or identical) monitoring and recovery actions may be effectively (re)used in the process of handling a certain exception and we also find that recovery actions may become approved as formal organization artifacts. Namely, recovery actions can become control transaction kinds – to handle the occasional, now known, exception – or normal transaction kinds – to handle this exception which in fact will, in the future, become a frequent and expected event which should become part of normal operation of the organization.

### 3.2 Organization Artifacts Life Cycle

Figure 3 presents the Meta Construction Model, part of DEMO's Meta Model available in [20].

In Figure 4 we find part 2 of the GO's SM where we see, as a central piece, the ORGANIZATION ARTIFACT object class. This class aggregates all “real” organization artifacts that constitute an organization system's composition, structure and production. Following the ontological parallelogram from [21], one can say that objective DEMO representations – like diagrams – designate subjective (in the mind) concepts

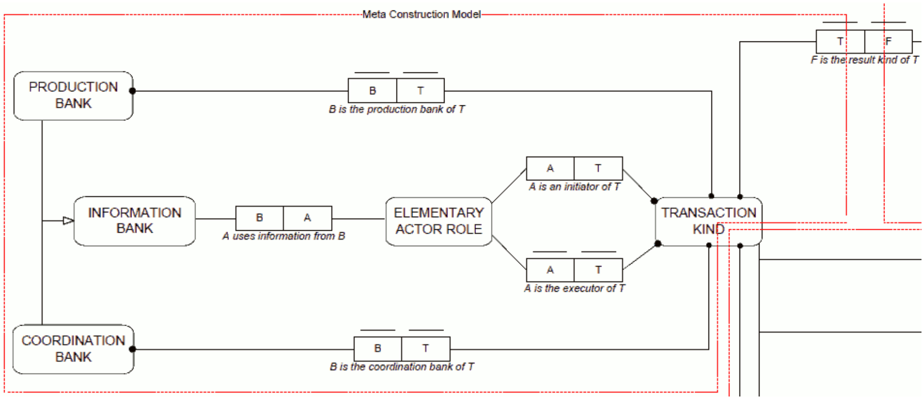



**Fig. 3.** DEMO Meta Model – Meta Construction Model

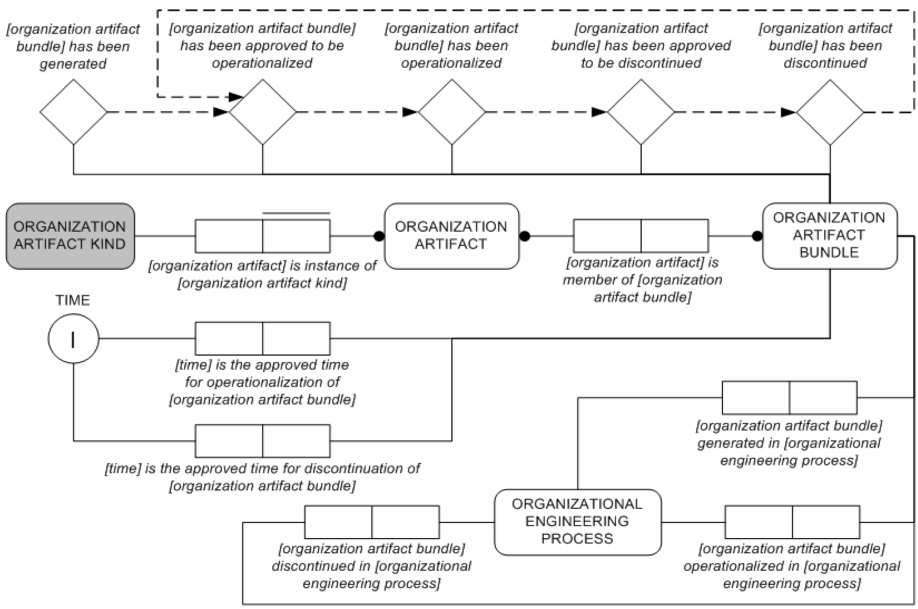



**Fig. 4.** G.O.D. Organization 's SSD – part 2

that constitute the ontological model (e.g. concept of transaction kind distribute fliers). These subjective concepts will, in turn, refer to “real” objective organization artifacts. They are real because they belong to the collectively shared inter-subjective reality of the organization system but are abstract, in the sense that they have no physical existence. Thus, instances members of object class ORGANIZATION ARTIFACT are referred by their respective concepts of the ontological model or, in other words, the ontological model of an organization is the conceptualization of its set of organization artifacts. Such concepts must be instances of a certain type which is a generic subjective concept that refers to a class of organization artifacts. Object class ORGANIZATION ARTIFACT KIND, thus, represents all (meta) fact types of the meta model out of which instances can occur in an organization. Because an organization artifact can be both viewed from an objective perspective – as belonging to the shared inter-subjective reality of the organizational self – and from a subjective perspective – as being a concept in the minds of an organization's human agents it seems to be appropriate for the G.O.D Organization to have, in its SM, the fact type: *[organization artifact] is instance of [organization artifact kind]*. That is, the concept of a particular organization artifact will be an instance of a particular type corresponding to a particular organization artifact kind of the organization space. As the GO is where the organizational self is produced, it is essential to keep information of which type a particular organization artifact is, so that, for example, automatic creation of an ontological representation is possible.

Typically, organization artifacts will always be generated and operationalized in bundles. For example, if we add a transaction kind, we necessarily need to add facts specifying which elementary actor role can initiate it, which can execute it, what's the associated result kind, etc. Thus, the need of object class ORGANIZATION ARTIFACT BUNDLE and fact type: *[organization artifact] is member of [organization artifact bundle]*. So, as a result of an OEP, a certain bundle of organization artifacts will be generated which then needs to be operationalized. This is captured by fact types: *[organization artifact bundle] generated in [organizational engineering process]* and *[organization artifact bundle] operationalized in [organizational engineering process]*. It may happen that a certain set of organization artifacts may become obsolete as a result of the soon to be operationalized ones. So it should be formally explicit which organization artifacts are discontinued in a certain OEP. This is possible by creation of instances of fact type: *[organization artifact bundle] discontinued in [organizational engineering process]*. Going back to our example of the library, a series of organization artifacts were generated as part of bundle OAB01. The GO will have to explicitly create instances of fact type *[organization artifact] is instance of [organization artifact kind]* relating each organization artifact with their respective kind and also instances of fact type *[organization artifact] is member of [organization artifact bundle]*, relating the same organization artifacts with bundle OAB01. Several kinds of tables can be specified to express relevant information of organization artifacts. Table 4 presents an example – a proposal of an Organization Artifacts Table (OAT) of the library.

Before being operationalized all organization artifacts of a bundle need to be approved for operationalization. The same applies for discontinuation. The time instant when the approval occurs may be different of the time instant of consummation of operationalization (and/or discontinuation). For example, the approval, by the library administration board, of OAB01, may occur in one day and the operationalization only after some days, due to time needed for implementation. So the time instant for operationalization and discontinuation of organization artifact bundles should be formalized by instances of the following fact types: *[time] is the approved time for operationalization of [organization artifact bundle]*; *[time] is the approved time for discontinuation of [organization artifact bundle]*. Note that in the context of an OEP we may have generated organization artifacts that were never operationalized but can provide good ideas to be reused in the generation of other organization artifacts in future OEPs, or even operationalized in their original shape.

**Table 4.** Organization Artifacts Table of the library

| organization artifact | kind | id | name/predicative sentence | bundle | last state |
|---|---|---|---|---|---|
| OA41 | exception kind | E01 | abnormal high rate of loan requests due to exams season | OAB01 | operationalized |
| OA42 | measure | M01 | loan declines per week | OAB01 | operationalized |
| OA43 | viability norm | VN01 | max loan declines per week | OAB01 | operationalized |
| OA44 | measure restriction | MRF01 | VN01 restricts M01 | OAB01 | operationalized |
| OA45 | exception causing dysfunction | EDF01 | E01 causes dysfunction in VN03 | OAB01 | operationalized |
| OA46 | transaction kind | T21 | general management | OAB01 | operationalized |
| OA47 | actor role | A21 | director | OAB01 | operationalized |
| OA48 | transaction initiation | TIF20 | A21 is an initiator of T21 | OAB01 | operationalized |
| OA49 | transaction execution | TEF20 | A21 is the executor of T21 | OAB01 | operationalized |
| OA50 | viability norm control | VNCF01 | T21 controls VN01 | OAB01 | operationalized |

We find, also in Figure 4, all result kinds characterizing the life cycle of organization artifacts. The ones regarding approval serve the purpose to have formal ownership of the relevant parts of an organization system. Discontinuation is an often neglected step regarding organization artifacts, which can lead to effects such as unneeded bureaucracy. Taking again the example of the library, considering that all books get equipped with hidden localizing chips one can eliminate the need of restricting maximum books a member can have on loan since at all times one can always pinpoint with GPS where a certain book is. So the above generated bundle can be discontinued.

### 3.3 Organization Artifacts Operationalization

Another issue currently not explicitly addressed in DEMO is that of allocation of real agents, for certain time periods, to fulfill certain actor roles. It seems the G.O.D. Organization is the right place to keep information bridging the world of the organizational self with the world of agents and their training and allocation to the respective actor roles as, without this, the operationalization of new organization artifacts cannot be realized. In other words, so that a certain dysfunction caused by an unexpected exception is solved, it is not enough to generate organization artifacts in an OEP. One has to operationalize them and this operationalization should occur in the context of the same OEP. Thus, in Figure 5 we present the SSD consisting in the formulation of the third and final part of our proposal for the GO's SM.

We find object class AGENT TRAINING where instances of this class will be part of a certain OEP and be related with a certain organization artifact bundle to be operationalized, and a certain agent which will fulfill the needed actor role. These relationships are specified by instances of fact types: *[organization artifact bundle] of [agent training]; [agent training] of [organizational engineering process]; [agent] of [agent training]*. A certain agent training will be part of exactly one OEP and consist in the training of exactly one agent with exactly one organization artifact bundle, thus, the

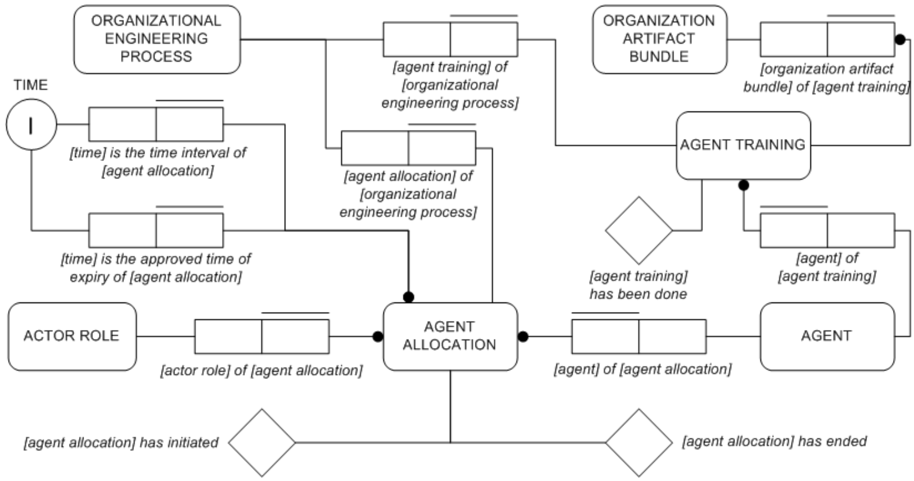



**Fig. 5.** G.O.D. Organization's SSD – part 3

specified unicity laws. Concerning the 10 organization artifacts previously defined as members of bundle OAB01, after training a certain agent (or agents) with the devised specifications the operationalization is realized with the allocation of such agent(s) to fulfill the relevant actor role(s). Thus, we also have object class AGENT ALLOCATION where instances of this class will be part of exactly one OEP and related with exactly one agent, one actor role, one time instant of expiry of such allocation and one time interval where such allocation is valid (it can be a composite time interval, like a normal work schedule with breaks for lunch and weekend break). The presented unicity laws are derived from the previous explanation and the expressed relationships are specified by instances of the following fact types: *[agent allocation] of [organizational engineering process]*; *[agent] of [agent allocation]*; *[actor role] of [agent allocation]*; *[time] is the time interval of [agent allocation]*; *[time] is the approved time of expiry of [agent allocation].*

The specifications of a certain organization artifact bundle need to be devised, down to the implementation model, which is the most detailed specifications needed to realize operation [2]. To be operationalized, these specifications need to be implemented in real physical organization agents: human or IT agents (or a mixture of both). These agents are then allocated to actor roles so that these can be fulfilled. Implementing specifications in human agents means some kind of training like personal interactive training or giving a text manual or even the relevant organization artifacts themselves in some representation like diagrams, tables or action rules. Implementing specifications in IT agents means coding – e.g., creating program from scratch – parametrization – e.g., configuring COTS (Commercial Off-The-Shelf) or ERP (Enterprise Resource Planning) software – or a mixture of both. Although it is quite acceptable to say that one implements specifications in IT resources, it is rather inappropriate to say one implements specifications in human resources. We thus propose the adoption of the more neutral terms "training" – as a synonym of implementation – and "agent" – as a synonym of a human or IT resource constituting actor technology.

## 4 Conclusions

In the reality of an organization facing unexpected exceptions, microgenesis dynamics occur in the form of what we call organizational engineering processes where there is an intertwining play between three major categories of acts: unexpected exception handling acts, organizational artifact state change acts and organizational artifact operationalization acts. The trouble is that, like we saw in our problem definition, in most cases these acts and their respective facts (reflecting analysis and change of organizational reality) are not captured and are forgotten. For a certain OEP, instances of all fact types defined in the GO's SM can collectively be considered as the full description of the ad-hoc original set of actions and facts that constitute the execution of the OEP itself and its effect on the organizational self. One major contribution of our proposal of extension to DEMO is that our approach provides contextual information to organization artifacts whereas in DEMO's traditional "static" view one does not have such information. The state base of the GO will be able to provide, for a certain organization artifact, information like: the dysfunction that was happening and respective exception that led to its generation and operationalization; who did what and decided what in the respective OEP. As we have seen, such information may be quite valuable in future change contexts affecting such artifact.

We have presented our proposal for a G.O.D. Organization existing in every organization and responsible for the generation, operationalization and discontinuation of organization artifacts that constitute the organization self, i.e., its composition, structure and production. We gave more focus to the aspect of handling unexpected exceptions, showing how keeping structured historical information of dynamics of this handling can be quite useful to handle exceptions occurring in the present. We then showed how the G.O.D. Organization makes a bridge between the worlds of model and meta model, where the latter contains the set of generic organization artifact (meta level) types out of which a set of (model level) instances can be generated that constitute an organization. With our proposal of the G.O.D. Organization, DEMO no longer is limited to a "static" picture of an organization and we can now have a full trace of the state of the organization system. The current picture of the organization, or, in other words, its ontological model, simply consists in the conceptualization of the set of organization artifacts that are current, i.e., whose last event was "has been operationalized". We then briefly touched on the aspect of operationalization of organization artifacts, an issue that seems to also fall in the responsibility of the G.O.D. Organization. Operationalization implies training IT or human agents with the devised specifications and then allocate such agents to fulfill the necessary actor roles. The state base of the G.O.D. organization seems to be the right place to keep state information of who is responsible to fulfill what and who was responsible for generation and training of organization artifacts. For space reasons many aspects of the specification of the G.O.D. Organization were left out of this paper. For example, all specified result kinds will have their associated transaction kinds and actor roles as well as associated action rules. But the most important part was addressed here: the State Model.

As future research, we foresee that what we call an OEP is a particular case of a more generic Change Process, which can include other particular cases like Improvement Handling Process or Innovation Handling Process. The G.O.D. Organization that we propose is surely incomplete and it can be extended to include the aspect of proactive change, whereas we have focused, on this paper, on reactive change. Another issue to be addressed in future research is practical tools and guidelines to implement the G.O.D. organization as, in our research, we have mostly tread the conceptual and proof of concept aspects.

## Acknowledgment

Research work that led to results presented in this paper was possible thanks to the financial support of a PhD scholarship (Ref.: SFRH / BD / 13384 / 2003) subsidized by "Fundação para a Ciência e a Tecnologia - Ministério da Ciência, Tecnologia e Ensino Superior" of the Portuguese government and by the European Social Fund.

## References

1. Applegate, L.M., McFarlan, F.W., McKenney, J.L.: Corporate information systems management: text and cases. McGraw-Hill, Irwin (1999)
2. Dietz, J.L.G.: Enterprise ontology: theory and methodology. Springer-Verlag New York, Inc., Secaucus (2006)
3. Christensen, W.D., Bickhard, M.H.: The process dynamics of normative function. The Monist 85, 3–29 (2002)
4. Holland, J.H.: Hidden order: how adaptation builds complexity. Basic Books, New York (1996)
5. Bickhard, M.H.: Error dynamics: the dynamic emergence of error avoidance and error vicariants. Journal of Experimental & Theoretical Artificial Intelligence 13, 199–209 (2001)
6. Brown, S.L., Eisenhardt, K.M.: Competing on the edge: strategy as structured chaos. Harvard Business School Press, Boston (1998)
7. Saastamoinen, H., White, G.M.: On handling exceptions. In: Proceedings of conference on Organizational computing systems, pp. 302–310. ACM, New York (1995)
8. Mourão, H.: Supporting effective unexpected exceptions handling in workflow management systems within organizational contexts. Science Faculty of Lisbon University (2007)
9. Tribolet, J.: Organizações, pessoas, processos e conhecimento: da reificação do ser humano como componente do conhecimento à "consciência de si" organizacional [organizations, people, processes and knowledge: from the reification of the human being as components of knowledge to the knowledge of organizational self]. Sistemas de informação organizacionais. Sílabo Editora. Lisbon, Portugal (2005)
10. Magalhães, R., Zacarias, M., Tribolet, J.: Making sense of enterprise architectures as tools of organizational self-awareness (osa). In: Proceedings of the Second Workshop on Trends in Enterprise Architecture Research (TEAR 2007), June 2007, vol. 6, pp. 61–70 (2007)
11. Zacarias, M., Magalhães, R., Caetano, A., Pinto, H.S., Tribolet, J.: Towards organizational self-awareness: an initial architecture and ontology. In: Handbook of ontologies for business interaction. Information Science Reference, pp. 101–121 (2007)
12. Magalhães, R., Silva, A.R.: Organizational design and engineering (ode) - ode white paper - version 1 (2009)
13. Casati, F., Pozzi, G.: Modeling exceptional behaviors in commercial workflow management systems. In: Proceedings of the Fourth CoopIS - International Conference on Cooperative Information Systems, pp. 127–138 (1999)

14. Axelrod, R., Cohen, M.D.: Harnessing complexity: organizational implications of a scientific frontier. Basic Books, New York (2001)
15. Dietz, J.L.G., Albani, A.: Basic notions regarding business processes and supporting information systems. Requirements Engineering 10, 175–183 (2005)
16. Op' t Land, M.: Applying architecture and ontology to the splitting and allying of enterprises. TU Delft (2008)
17. Dietz, J.L.G.: On the nature of business rules. Advances in Enterprise Engineering I, 1–15 (2008)
18. Bunge, M.A.: Treatise on basic philosophy, a world of systems, vol. 4. Reidel Publishing Company (1979)
19. Bon, J.V.: It service management, an introduction. itSMF-International (2002)
20. Dietz, J.L.G.: Demo meta model specification (forthcoming) (2009), `http://www.demo.nl`
21. Dietz, J.L.G.: A world ontology specification language. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2005. LNCS, vol. 3762, pp. 688–699. Springer, Heidelberg (2005)

# Towards Deterministically Constructing Organizations Based on the Normalized Systems Approach

Dieter Van Nuffel, Philip Huysmans, David Bellens, and Kris Ven

Department of Management Information Systems,
University of Antwerp, Antwerp, Belgium
{dieter.vannuffel,philip.huysmans,david.bellens,kris.ven}@ua.ac.be

**Abstract.** Contemporary organizations need to be more agile to keep up with the swiftly changing business environment. This means that their organizational structure, business processes and information systems should evolve at the same pace. This proves to be quite a challenge due to the invasive nature of these changes and a lack of alignment between these artefacts. It has therefore been argued that more determinism is needed when engineering these artefacts. 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). This study is part of ongoing design science research to incorporate determinism in the construction of an organization's artefacts. Our results show that such approach is feasible and could increase traceability from the organizational level to the information systems.

## 1 Introduction

Contemporary organizations need to be more agile to keep up with the swiftly changing business environment. Agility is described as the measure of the organization's ability to change and adapt to its new environment [24]. This means that the organizational structure, business processes and information systems should be able to evolve at the same pace [29]. Research shows that the alignment between these artefacts should be strong in order to successfully implement an information system [23]. The notion of enterprise agility is also investigated by the Services Computing field, sometimes also referred to as Services Science [3]. This domain tries to better align the technological foundation—mostly represented as Service-Oriented Architectures (SOA)—and the business foundation—mostly focused on service innovation and business services—of an organization [37]. However, this proves to be quite a challenge. Since changes to the technological and/or business foundation often affect the core of the organization, these changes often have an invasive nature. Moreover, the artefacts in both domains display a lack of alignment.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 242–257, 2010.

Since organizations need to be able to quickly adapt to changing requirements, this means that these requirements must be translated into changes to the enterprise architecture, business processes, and ultimately the underlying information system. This suggests that the link between the artefacts in these domains should be made stronger, so that if a change in one of these artefacts occurs, the required modifications to the other artefacts can be derived. It has indeed been argued that more determinism—i.e., applying principles to obtain a predictable and desired result—is required when engineering these artefacts in order to introduce traceability from the business requirements to the underlying information systems [7].

It is therefore our belief that this stack of requirements and enterprise models should be approached in a uniform way to achieve this traceability and alignment. Recently, the normalized systems (NS) approach has been proposed to design information systems exhibiting proven evolvability [19]. Because NS is built upon the systems theoretic concept of stability, and principles to isolate change drivers on the software architecture were derived from this concept, this approach seems to be highly suited to provide the required uniform theoretical foundation. In this paper, we extend the NS approach's basic principles to the related fields of Enterprise Architecture (EA) and Business Process Management (BPM). These domains were chosen because a fundamental principle when designing an enterprise is to view the enterprise in its overall context. Moreover it is clear that there is a form of deterministic influence between the fields mentioned and the way an information system can be constructed [30]. Extending the NS approach to the fields of EA and BPM seems feasible and could increase traceability from the organizational level to the information systems.

The remainder of the paper is organized as follows. We will start by briefly describing the normalized systems approach to introduce the main concepts of interest to this paper. The third section discusses the applied Design Science Research methodology. We subsequently describe how the normalized systems approach is extended into the two mentioned fields. Finally, conclusions and future research are presented.

## 2 Normalized Systems

The basic assumption of the normalized systems approach is that information systems should be able to evolve over time, and should be designed to accommodate change. As this evolution due to changing business requirements is mostly situated during the mature life cycle stage of an information system, it is coined as software maintenance. Software maintenance is considered to be the most expensive phase of the information system's life cycle, and often leads to an increase of architectural complexity and a decrease of software quality [8]. This phenomenon is also known as Lehman's law of increasing complexity, expressing the degradation of information system's structure over time [17]. Because changes applied to information systems are suffering from Lehman's law, the impact of a single change will increase over time as well [16]. Therefore to genuinely design information systems accommodating change, they should exhibit *stability* towards these requirements changes. In systems theory, stability refers to the fact that bounded input to a function results in bounded output values, even as $t \rightarrow \infty$. When applied to information systems, this implies that no change propagation effects should be present within the system; meaning that a specific change to an information system should require the same effort, irrespective of the information system's size or point in time when being applied. *Combinatorial effects* occur when changes require increasing effort as the system grows; and should thus be avoided. Normalized systems are therefore defined as information systems exhibiting stability with respect to a defined set of changes [19], and are as such defying Lehman's law of increasing complexity [16,17] and avoiding the occurrence of combinatorial effects. In this sense, evolvability is operationalized as a number of anticipated changes that occur to software systems during their life cycle [20].

The normalized systems approach deduces a set of four *design principles* that act as design rules to identify and circumvent most combinatorial effects [20,19]. It needs to be emphasized that each of these principles is not completely new, and even relates to the heuristic knowledge of developers. However, formulating this knowledge as principles that identify these combinatorial effects aids to build systems containing minimal combinatorial effects. The first principle, *separation of concerns*, implies that every change driver or concern should be separated from other concerns. This theorem allows for the isolation of the impact of each change driver. Parnas described this principle already in 1972 [25] as what was later called *design for change*. Applying the principle prescribes that each module can contain only one submodular task (which is defined as a change driver), but also that workflow should be separated from functional submodular tasks. For instance, consider a function $F$ consisting of task $A$ with a single version and a second task $B$ with $N$ versions; thus leading to $N$ versions of function $F$. The introduction of a mandatory version upgrade of the task $A$ will not only require the creation of the additional task version of $A$, but also the insertion of this new version in the $N$ existing versions of function $F$. The number $N$ is clearly dependent on the size of the system, and thus implies a combinatorial effect.

The second principle, *data version transparency*, implies that data should be communicated in version transparent ways between components. This requires that this data can be changed (e.g., additional data can be sent between components), without having an impact on the components and their interfaces. For instance, consider a data structure $D$ passed through $N$ versions of a function $F$. If an update of the data structure is not version transparent, it will also demand the adaptation of the code that accesses this data structure. Therefore, it will require new versions of the $N$ existing processing functions $F$. The number $N$ is clearly dependent on the size of the system, and thus implies a combinatorial effect. This principle can, for example, be accomplished by appropriate and systematic use of web services instead of using binary transfer of parameters. This also implies that most external APIs cannot be used directly, since they use an enumeration of primitive data types in their interface.

The third principle, *action version transparency*, implies that a component can be upgraded without impacting the calling components. Consider, for instance, a processing function $P$ that is called by $N$ other processing functions $F$. If a version upgrade of the processing function $P$ is not version transparent, this will cause besides upgrading $P$, it will also demand the adaptation of the code that calls $P$ in the various functions $F$. Therefore, it will require new versions of the $N$ existing processing functions $F$. The number $N$ is clearly dependent on the size of the system, and thus implies a combinatorial effect. This principle can be accomplished by appropriate and systematic use of, for example, polymorphism or a facade pattern.

The fourth principle, *separation of states*, implies that actions or steps in a workflow should be separated from each other in time by keeping state after every action or step. For instance, consider a processing function $P$ that is called by $N$ other processing functions $F$. Suppose the calling of the function $P$ does not exhibit state keeping. The introduction of a new version of $P$, possibly with a new error state, would force the $N$ functions $F$ to handle this error, and would therefore lead to $N$ distinct code changes. The number $N$ is clearly dependent on the size of the system, and thus implies a combinatorial effect. This suggests an asynchronous and stateful way of calling other components. Synchronous calls resulting in pipelines of objects calling other objects which are typical for object-oriented development result in combinatorial effects.

The design principles show that software constructs, such as functions and classes, by themselves offer no mechanisms to accommodate anticipated changes in a stable manner. The normalized systems approach therefore proposes to encapsulate software constructs in a set of five higher-level software elements. These elements are modular structures that adhere to these design principles, in order to provide the required stability with respect to the anticipated changes [19]. From the second and third principle it can straightforwardly be deduced that the basic software constructs, i.e. data and actions, have to be encapsulated in their designated construct. As such, a *data element* represents an encapsulated data construct with its get- and set-methods to provide access to their information in a data version transparent way. So-called cross-cutting concerns, for instance access control and persistency, should be added to the element in separate constructs. The second element, *action element*, contains a core action representing one and only one functional task. Arguments and parameters need to be encapsulated as separate data elements, and cross-cutting concerns like logging and remote access should be again added as separate constructs. Based upon the first and fourth principle, workflow has to be separated from other action elements. These action elements must be isolated by intermediate states, and information systems have to react to states. To enable these prerequisites, three additional elements are identified. A third element is thus a *workflow element* containing the sequence in which a number of action elements should be executed in order to fulfill a flow. A consequence of the stateful workflow elements is that state is required for every instance of use of an action element, and that the state therefore needs to be linked or be part of the instance of the data element serving as argument. A *trigger element* is a fourth one controlling the states (both regular and error states) and checking whether an action element has to be triggered. Finally, the *connector element* ensures that external systems can interact with data elements without allowing an action element to be called in a stateless way.

## 3 Research Methodology

In this section, we align our research efforts with existing Design Science literature. Regarding the research project's nature to generate a deterministic approach within the mentioned domains, only a Design Science Research methodology [26] is suited to provide the required research setting as it is primarily aimed at *solving* problems by developing and testing *artefacts*, rather than *explaining* them by developing and testing *theoretical hypotheses*. The design science research tradition focuses on tackling ill-structured problems, in this research the lack of determinism within the engineering of organizational artefacts, in a systematic way [11]. The researcher develops "a means to an end", an artefact to solve the problem, in which either the means or the end, or both, must be novel [11]. This research project's deliverable is a set of methods, mainly based on the normalized systems approach, providing guidelines to purposefully design enterprise architectures and business processes. Therefore, the research entry point is problem-centered [26] as a lack of determinism when constructing organizational artefacts inhibits the required enterprise agility. In accordance with Simon [32], who makes a distinction between the Behavioral Science and the Design Science, building a (part of a) method is actually studying the artificial as a method is a man made object designed to meet certain desired goals. In addition, a method is defined as "*These [methods] can range from formal, mathematical algorithms that explicitly define the search process to informal, textual descriptions of "best practice" approaches, or some combination.*" [10, p.79]. Also Winter [35] mentions the paucity of design science research aimed at constructing methods. In this sense, this study is concerned with the only modestly researched area of Method Engineering. Moreover the research topic can unambiguously be positioned within the design science classification scheme suggested by March et al. [21]. As table 1 illustrates, this research will build and evaluate methods, being the artefacts constructed by the research. We will first elaborate on how the methods will be constructed, before discussing

**Table 1.** Classification of research topic within scheme of March et al. [21]

| | | Research Activities | | | |
|---|---|---|---|---|---|
| | | Build | Evaluate | Theorize | Justify |
| Research Outputs | Construct | | | | |
| | Model | | | | |
| | Method | X | X | | |
| | Instantiation | | | | |

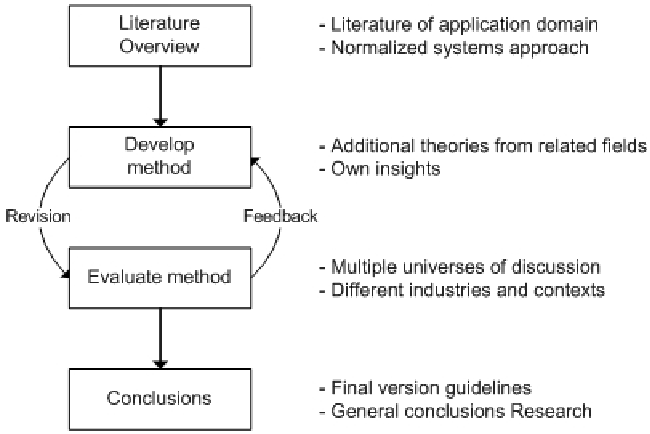



**Fig. 1.** Research design

how the evaluation of the methods will be performed further on in the section. Regarding the artefact construction, it should be mentioned that the two identified domains will be approached by different researchers. Although the domains differ from each other, the applied research method exhibits an identical research trajectory, as illustrated by figure 1. This procedure mimics the "Generate/Test cycle" suggested by Simon [32]; and a similar process is proposed by Peffers et al. [26]. The initial iteration consists of screening the literature of the respective domains for already available useful input. The retrieved sources are complemented with the insights and principles from the normalized systems approach in order to identify a preliminary set of deterministic rules. The guidelines constituting the method of the first iteration are thus only theoretically-grounded, as they are constructed using already available literature on the topic. Initial results of the first iterations will be presented within section 4.

Evaluating the proposed guidelines will occur by applying the guidelines on different problem domains. These problems domains will be purposefully sampled, controlling for different industries and organizational dimensions. Regarding the BPM field, processes are taken from the banking, government and discrete manufacturing industry. In addition, processes differ along their administrative dimension, ranging from operational order to management reporting processes. Regarding the EA domain different industries are sampled on a case-based view as to identify different kinds of architectural aspects, e.g. supply chain aspects, accounting aspects, etc.

An important property of Design Science is its iterative character to which this research adheres by repeating the development and evaluation several times [26,32]. As a result, the guidelines are built progressively: when being confronted with a set of inadequate or incomplete set of guidelines, further examination (or development) is necessary. For example, knowledge from related fields such as Service Computing can provide useful input to revise and/or extend the method.

**Table 2.** Matching research evaluation with framework of Cleven et al. [4]

| Variable | Value in our research |
|---|---|
| Approach | Qualitative |
| Artefact Focus | Organizational |
| Artefact Type | Method |
| Epistemology | Positivism |
| Function | Controlling function |
| Method | Case Studies & Action Research |
| Object | Artefact |
| Ontology | Realism |
| Perspective | Engineering |
| Position | Internally |
| Reference Point | Artefact against research gap and against real world |
| Time | Ex post |

The methods constructed will be evaluated using two approaches. First, through the multiple iterations the method will be tested and altered to better suit the research objective of enhancing determinism. This approach can be labeled as case study research. The cases studied during initial iterations will mainly consist of rather pedagogical, theoretical cases. Further iterations include more complex cases, based upon real-life organizations in order to enhance the generalization of our results. As mentioned earlier, these cases will be purposefully sampled to assure validity. Secondly, to firmly evaluate the proposed methods, they will finally be applied to real-life cases to assess their practical applicability. This kind of evaluation is based on the action research methodology [1] because the researcher actively cooperates within the case. The application of the proposed guidelines is the action executed. Table 2 summarizes our evaluation approach based upon the evaluation framework presented by Cleven et al. [4]. Our evaluation can be interpreted to be positivist as the results of the evaluation are independent from the evaluator's subject. In our opinion, it is possible to assess the deterministic nature of the resulting artefacts using the same objective interpretation. By applying a dual evaluation approach, a dual reference point is realized as well. During the initial iterations, the methods will be evaluated whether they realize deterministic artefacts. In this sense, the evaluation's function is mainly controlling whether the defined criteria to enhance determinism and evolvability of the researched artefacts are met. In addition, by extending the evaluation into real-life settings, an evaluation against the real world is performed as well.

Regarding the overarching research project, an iteration integrating the methods of the different research streams should be executed. In this sense, an overall method providing guidelines to introduce determinism in the organizational artefacts is constructed. Such integration seems to be feasible as the main theoretical foundation provided by the normalized systems approach grounds both research streams.

The integrated research project clearly illustrates the use of a proven approach of the software engineering field in related fields. This is in line with the Design Science methodology. Various authors indicate that the use of theories of related fields should indeed be an essential part of a design science approach. According to Klahr and Simon [12], the notion of "parallel domains of human expertise" should be the core of design science research. Simon [32] argues that design science research should be at the center of "distinguishable yet connected research domains". Peffers et al. [26] call IS an applied research discipline, meaning that theory from disciplines such as economics, computer science and social sciences are frequently used to solve problems between information technology and organizations. The normalized systems approach is specifically useful for this purpose, since it expresses proven design experience through principles which can be proven to be necessary. Both aspects are needed to be usable in a design science context. On the one hand, a well-founded theory which does not offer practical implications for the design of artefacts is of limited practical use. On the other hand, design guidelines which are not verifiable do not contribute to the science of design. Moreover, the correlation of normalized systems design principles with more general theory such as systems theory and modularity, indicates its aptness for extension to other research fields.

## 4 Application Domains

This section will elaborate on how the normalized systems principles can be extended in the fields of Enterprise Architectures and Business Processes.

### 4.1 Enterprise Architecture

When market threats, opportunities or changes arise, the organization as a whole has to adapt. In order to be able to comprehend and manage the complexity of modern organizations, enterprise architecture frameworks have been introduced. These frameworks usually distinguish between the business system and the information system. The business system consists of elements such as goals, people, processes, data and events. These elements are usually placed on a horizontal axis.

By specifying conceptual models for the elements, requirements for the supporting information system are formed. The integration between the conceptual models should facilitate the translation of a single change in the outside world to all the different aspects of the organization. As such, the models are translated from abstract business concepts to concrete information system artefacts. The vertical axis usually specifies certain layers or steps in which this translation occurs. Despite the common goal of enterprise architectures, many different frameworks are available. Various authors (e.g. [13,18,14]) have compared these frameworks and identified differences and similarities. The GERAM framework (Generalized Enterprise Reference Architecture and Methodology) was created to provide a reference framework onto which the individual frameworks could be mapped. Given the broad scope covered by these frameworks and the multitude of frameworks, it is logical that not every framework contains all elements present in other frameworks. Should an enterprise architect require to use all available elements, several (complementary) frameworks can concurrently be used, or a particular framework can be extended (as reported by e.g. [27]).

However, by combining or extending existing frameworks, the issue of *integration* becomes even more complex. While most frameworks reduce the inherent complexity of an organization by offering separate views, it is not always clear how these views relate to or affect each other. The proposed integration or mapping methods are mostly based on refinement or reification, and focus on the vertical dimension. While some frameworks offer dedicated constructs for combining models (e.g. the process view in ARIS), it is not clear how this integration affects the ability of the models to change independently. If a change in a certain model affects other models it is combined with, a *combinatorial effect* occurs. While originally used to describe evolvability in software, combinatorial effects also seem to affect evolvability on the Enterprise Architecture level. Analogously with combinatorial effects on the software level, this implies that organizations would become less evolvable as they grow. While the issue of integration has been acknowledged by other authors (e.g. [15]), it has, to our knowledge, not yet been studied based on system theoretic concepts such as stability. By applying the design principles from normalized systems to Enterprise Architecture, we attempt to introduce these concepts in this field. We work towards a deterministic method to build evolvable enterprise architectures. The focus of the method will be on the deterministic combination of models within the architecture, in order to avoid combinatorial effects. Put differently, the method will prescribe how the different aspects of the organization have to be integrated in order to be evolvable. Based on the literature on enterprise architectures, we will therefore work towards a method to integrate the set of models which make up the enterprise architecture. As discussed in Section 3, the research consists of several iterations. We will outline here the results of our first research iteration. This iteration consisted of the selection of a core diagram, and ensuring that the core diagram adheres to the normalized systems design principles.

The core diagram [28] is a model which provides an overview of the organizational scope which will be designed. On this abstract level, the model should not distinguish between different aspects, but represent the core of the enterprise. Therefore, we will base ourselves on the Enterprise Ontology models [7]. Enterprise Ontology was selected since it models the abstract working of the organization without specifying how the organization is implemented. The ontological models of the Ford BPR-case for example, are identical before and after the redesign [6]: the purchasing department still fulfills the same ontological process, it is just implemented differently. Enterprise Ontology regards organizations as social entities and bases its constructs around the creation of so-called ontological *facts*. The ontological facts correlate with the goods or services that are delivered by the organization to the market. For example, an ontological fact for a company which produces computers would be: *"The computer with id#385* *has been produced"*. The coordination between the customers and the organization needed to produce the fact is represented in a *transaction pattern*. In our example, the customer would first *request* the fact *"The computer with id#385 has been produced"*. Next, the computer company would *promise* to produce the fact, it would actually *execute* it, and then *state* that the fact was completed. Finally, the customer would *accept* the creation of the fact. By modeling the organization as a collection of transactions, compact models can be created which show the construction of the enterprise.

Since these models are implementation-independent, we can base our method on these models to implement the needed organizational aspects in the transactions. In the first design iteration, the Enterprise Ontology models were selected as the core diagram, and mapped the transaction construct to normalized systems elements. The resulting artefact is a so-called *Normalized Systems Business Transaction (NSBT)*. It was shown that the NSBT adheres to the normalized systems design principles. In subsequent iterations, we will integrate other aspects present in Enterprise Architecture frameworks. This will be done analogously to the integration of cross-cutting concerns on the software level into normalized systems elements. Once these cross-cutting concerns were managed by the elements, the promise of isomorphism could be delivered. For example, a common aspect in enterprise architectures is the process aspect (e.g. the how-column in the Zachman framework [36]). This aspect does not occur in the transaction: it is not specified how the organization produces the computer. Of course, the process aspect has to adhere to the normalized systems design principles as well. We elaborate on the *Normalized Systems Business Processes (NSBP)* in the next section. Since we expressed the NSBT in normalized systems elements, the process aspect can be added by using a bridge action element [19, p.148].

### 4.2 Business Process Management

Business Process Management (BPM) is defined as the domain encapsulating "the concepts, methods, and techniques to support the design, administration, configuration, enactment and analysis of business processes" [34, p.5]. Our research effort is mainly targeted at the analysis and design activities. Recently, business process models are considered first-class citizens as process-centric representations of an enterprise [34]. Whereas earlier, mostly data-driven approaches have been pursued as a starting point for information systems analysis, design and implementation, there is currently a tendency to apply process-driven requirements engineering. Although a relatively large number of notations and languages exist to model business processes, these representations suffer from a number of shortcomings [5]. Moreover only very few guidelines and best practices are available to design or engineer business processes. Either the available theoretical frameworks are too abstract and require a certain level of modeling competence, or the guidelines are more practically-oriented and mostly lack empirical and/or theoretical support. A first set for instance, defines guidelines to enhance the understandability of processes [22]. As they provide useful insights about the format of the process on the modeling desk, they do not state any principles regarding the content of the process. Second, recent work by Silver [31] establishes an empirically-based set of principles to model business processes within the industry standard Business Process Modeling and Notation (BPMN). Three abstraction levels are defined and on each of these levels, a number of principles are given to model business processes according to a specific style.

If business processes are however describing requirements when developing software, more determinism is needed [9]. The quality of the models should be secured as they should both correctly represent the requirements and describe these requirements in an unambiguous way to the software developers. In order to obtain the required determinism, our research applies the normalized systems principles to the Business Process Management domain. In order to obtain such *Normalized Systems Business Processes (NSBP)*, the concept of a production line, that assembles instances of a specific product that is being created, is applied to a business process flow, that performs operations on instances of a specific target data argument. Though production lines seem highly integrated at first sight, they actually exhibit loose coupling. Although every single processing step requires the completion of the previous steps on that instance of the product that is being created, it does not require any knowledge of the previous processing steps, nor of the subsequent steps. Moreover, they do not have to be aware of the timing of the other steps. Any step can be performed on thousands of product instances that have been prepared hours, or even days, earlier. Referring to the research methodology set out earlier, two results from the first iterations are presented. First, a timer element was added to the normalized systems elements to implement the omnipresent task of timing functionality. Second, an initial set of guidelines to introduce normalized systems principles to business process models was developed [33].

A first result attained by the research, is the purposeful design and implementation of timer functionality. Within business processes, timers are required to represent timing dependencies such starting a process at a specified point in time (e.g. every morning at 7AM a management reporting process has to be executed) or only allowing a stakeholder to complete a task within a particular time frame. It should be mentioned that when designing business processes adhering to the normalized systems principles, a flow element can only operate on a single data element driving the flow through its state attribute. Interaction with other data elements is of course needed, but is implemented using so-called bridge actions [19, p.148]. Based on the principle of *separation of concerns*, managing a time constraint should be separated in its designated element. Also BPMN defines a separate artefact, a timer event, to denote this functionality. Timers independently execute from both other action elements defined on the same data element and from the flow element driving the process. To illustrate our reasoning, consider a electronic holiday request process where a manager has to approve holiday requests of her personnel and where a reminder is sent to the manager when she has not made a decision within one week. The flow element driving this business process defines the sequencing of the constituent tasks, implemented by action elements. One of these action elements consists of a manual task performed by the manager to decide on the submitted holiday request. This task is probably implemented by clicking on the "approve" or "reject" button provided in an e-mail or on a GUI. Clearly, the timer element realizing the one week timing constraint is another concern than the above mentioned action elements. On the other hand, the timing constraint is also another concern than the sequencing of the constituent action elements. When this timing constraint would be added to the flow element's functionality, this flow element would encapsulate different change drivers evolving rather independently from each other: the order in which activities are performed within an expense report process and the time period defined to send a reminder. Therefore, a designated reusable timer data element is constructed of which an instance thus represents a timing constraint operating on a single life cycle data element. Such a timer specifies a maximum allowed period between two states or anchor points in a flow. The timer may identify a specific action element to be executed in case the timer expires, and/or a new state that needs to be set in any instance of the data element for which the timer expires.

A second research deliverable consists of an initial set of guidelines on how to design business processes based on the normalized systems principles [33]. This set provides a proof-of-concept that the NS principles are applicable to contemporary business processes, and illustrates the possibility of introducing determinism within business process descriptions. A first deterministic guideline is the fact that a business process corresponds with a flow element driven by a state data field of a single data element. If a described business process however requires processing on multiple data elements, the different flow elements should be integrated using bridge action elements. This kind of action elements links different life cycle data elements in a loosely coupled way, in order to obey the principle of *separation of concerns*. Additional guidelines are also provided on how to deal with these multiple data elements when the flows on these data elements have to interact with each other. Depending on their relationship, an action element on the triggering life cycle data element has to be implemented that will verify the state of the initiated instances of the related data element [33]. Moreover applying the *separation of states* principle combined with a very concise labeling of each state results in the status of every data element to be uniquely described by the value of the state data field.

Of course, prescriptive and deterministic rules to identify the life cycle data elements within the business processes are necessary as well. The identification of a life cycle data element can however be considered relatively straightforward as they represent the business entities going through their life cycle during the business processes execution. The three conditions of Bhattacharya et al. [2, p.290] to distinguish such business entities – a record storing information pertinent to a given business context, possessing a distinct life cycle from creation to completion, and having a unique identifier within the organization – are added to identify life cycle data elements. Furthermore, the principles prescribe certain functionality to be separated in its designated flow, and thus also in its designated data element. Besides the earlier described timer element encapsulating timing functionality, also the concern of notifying several stakeholders should be isolated. Such notifications consist of two concerns: the extraction of the information that makes up the message's content on the one hand, and the actual sending of the message on the other. This means that, in accordance with *separation of concerns* and *separation of states*, they have to be separated into two different action elements. As the second task is a quite generic one, it should operate on a corresponding generic life cycle data element `Notifier`, in a corresponding separate flow.

Finally, an additional iteration delivered insights on how to deal in a prescriptive way with cancelations. To illustrate our viewpoint, consider the case in which a customer decides to cancel an order, e.g. a custom-made cupboard. When the order is canceled, the process state – and thus also the underlying life cycle data element's state – cannot simply be set to *canceled* and by consequence disregard everything that has already been done. This would lead to an infinite amount of reserved parts in stock, e.g. wooden shelves, as these parts will be kept reserved for an already canceled order. Therefore, based on the normalized systems principles, an entire branch should be added to the process flow of the order, where the assembly request is withdrawn and the various reserved parts are released. The following way of working can be followed to implement this branch. The cancel event has to be captured by a dedicated data attribute of the order data element. The value of this dedicated data attribute will trigger a state transition of the regular state field – triggering on its turn the respective cancelation flow – and will also store the initial state persistently in another state field, a *parking state field*. The initial state should be kept because this state is needed to trigger the correct cancelation flow. Because of the earlier mentioned combination of concise state labeling and *separation of states*, each state uniquely describes the state of a data element within its life cycle. Thus, this state can also uniquely determine which tasks should be executed to compensate the already performed process activities. The value attributed to the regular state attribute must moreover be the same for every data element, as this will uniquely define the situation of an element being canceled and can thus be recognized within all data elements. As such determinism is introduced both through concise use of designation, and handling a cancelation request in a uniform way based on the normalized systems principles.

## 5 Discussion and Conclusions

Our research is concerned with extending the normalized systems approach to the related fields of business process management and enterprise architecture. In this paper, we outlined the design science methodology which is followed, and presented the results of our first iteration. We have shown that combinatorial effects do not only occur at the level of information systems, but also on the level of enterprise architectures and business processes. Since evolvability of information systems is shown to be inhibited by combinatorial effects, similar consequences can be expected for these other two domains. The normalized systems approach further shows how such combinatorial effects can be restrained in information systems by using a set of principles. We outlined how we intend to study and remedy this problem in the mentioned fields by developing a similar set of principles for the respective domains. Our results show that this approach is feasible and that these principles provide a strong foundation for constraining combinatorial effects in enterprise architectures and business processes as well. Moreover, using a uniform and theoretically-grounded approach as a starting point could increase traceability from the organizational level to the information systems.

This study has a number of contributions. First, we show that the systems theoretic concept of stability can be used as a single starting point in three separate domains to constrain combinatorial effects. Although systems theory is well-known in academic literature, few prior studies have attempted to apply the concept of stability to information systems, enterprise architecture of business process management. In this paper, we have demonstrated the potential of using stability in these three domains. Second, we extended the normalized systems approach into two other domains. We thereby illustrate that this approach is also applicable to domains other than information systems. Moreover, by using a single starting point in three domains that are essential to the functioning of organizations, we have illustrated the potential of increasing traceability between the business and technological levels in organizations. Finally, our study contributes to the design science methodology in two different ways. First, we have applied a proven theory from a related field (i.e., information systems) to two additional fields, which is frequently considered an essential part of a design science approach. Second, we contribute to the research area on method engineering, which has received little attention so far in literature.

However, this study also has some limitations that provide opportunities for future research. First, the systems theoretic concept of stability was used as a starting point. While normalized systems have shown that information systems based on stability exhibit other important characteristics (e.g., performance) as well, it is not necessarily the best, and certainly not the only possible foundation. Therefore, future research could use other starting points to provide integration with, or additions to, our research efforts. Second, we selected certain existing approaches to base our methods on: the enterprise architecture method starts from Enterprise Ontology models, while the business process management method uses BPMN models for illustrative purposes. Similarly, future research could use different models in each domain to provide additional insights. Third, given the current state of the research, the development and validation of the designed artefacts is not yet complete. This will be addressed in subsequent iterations of our research. Nevertheless, this paper demonstrates the feasibility of our approach and already resulted in valuable insights and therefore has a number of important contributions.

## References

1. Baskerville, R.L., Wood-Harper, A.T.: A critical perspective on action research as a method for information systems research. Journal of Information Technology 11(3), 235–246 (1996)

2. Bhattacharya, K., Gerede, C., Hull, R., Liu, R., Su, J.: Towards formal analysis of artifact-centric business process models. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 288–304. Springer, Heidelberg (2007)
3. Chesbrough, H., Spohrer, J.: A Research Manifesto for Services Science. Communications of the ACM 49(7), 35–40 (2006)
4. Cleven, A., Gubler, P., Hüner, K.M.: 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 (DESRIST 2009). ACM Press, New York (2009)
5. De Backer, M., Snoeck, M., Monsieur, G., Lemahieu, W., Dedene, G.: A scenario-based verification technique to assess the compatibility of collaborative business processes. Data and Knowledge Engineering 68(6), 531–551 (2009)
6. Dietz, J.L.: The deep structure of business processes. Communications of the ACM 49(5), 58–64 (2006)
7. Dietz, J.L.: Enterprise Ontology: Theory and Methodology. Springer, Berlin (2006)
8. Eick, S.G., Graves, T.L., Karr, A.F., Marron, J., Mockus, A.: Does Code Decay? Assessing the Evidence from Change Management Data. IEEE Transactions on Software Engineering 27(1), 1–12 (2001)
9. Gruhn, V., Laue, R.: What business process modelers can learn from programmers. Science of Computer Programming 65(1), 4–13 (2007)
10. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
11. Holmström, J., Ketokivi, M., Hameri, A.P.: Bridging Practice and Theory: A Design Science Approach. Decision Sciences 40(1), 65–87 (2009)
12. Klahr, D., Simon, H.A.: Studies of scientific discovery: Complementary approaches and convergent findings. Psychological Bulletin 125(5), 524–543 (1999)
13. Kozina, M.: Evaluation of ARIS and Zachman Frameworks as Enterprise Architectures. Journal of Information and Organization Sciences 30(1) (2006)
14. Op 't Land, M., Proper, E., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture: Creating Value by Informed Governance, 1st edn. The Enterprise Engineering. Springer, Heidelberg (2008)
15. Lankhorst, M.M.: Enterprise architecture modelling–the issue of integration. Advanced Engineering Informatics 18(4), 205–216 (2005)
16. Lehman, M.: Programs, life cycles, and laws of software evolution. Proceedings of the IEEE 68, 1060–1076 (1980)
17. Lehman, M.M., Ramil, J.F.: Rules and Tools for Software Evolution Planning and Management. Annals of Software Engineering 11(1), 15–44 (2001)
18. Leist, S., Zellner, G.: Evaluation of current architecture frameworks. In: SAC 2006: Proceedings of the 2006 ACM symposium on Applied computing, pp. 1546–1553. ACM, New York (2006)
19. Mannaert, H., Verelst, J.: Normalized Systems: Re-creating Information Technology Based on Laws for Software Evolvability. Koppa, Hasselt (2009)
20. Mannaert, H., Verelst, J., Ven, K.: Exploring the Concept of Systems Theoretic Stability as a Starting Point for a Unified Theory on Software Engineering. In: Mannaert, H., Ohta, T., Dini, C., Pellerin, R. (eds.) Proceedings of Third International Conference on Software Engineering Advances (ICSEA 2008), pp. 360–366. IEEE Computer Society, Los Alamitos (2008)
21. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)
22. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven process modeling guidelines (7pmg). Information and Software Technology 52(2), 127–136 (2010)

23. Middleton, P., Harper, K.: Organizational alignment: a precondition for information systems success? Journal of Change Management 4(4), 327–338 (2004)
24. Neumann, S.: Strategic Information Systems: Competition Through Information Technologies. Macmillan College Publishing Company, New York (1994)
25. Parnas, D.L.: On the criteria to be used in decomposing systems into modules. Communications of the ACM 15(12), 1053–1058 (1972)
26. 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 (2007–2008)
27. Pereira, C.M., Sousa, P.: A Method to define an Enterprise Architecture using the Zachman Framework. In: SAC 2004: Proceedings of the 2004 ACM symposium on Applied computing, pp. 1366–1371. ACM, New York (2004)
28. Ross, J.W., Weill, P., Robertson, D.C.: Enterprise Architecture as Strategy – Creating a Foundation for Business Execution. Harvard Business School Press, Boston (2006)
29. Sambamurthy, V., Bharadwaj, A.S., Grover, V.: Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms. MIS Quarterly 27(2), 237–263 (2003)
30. Schekkerman, J.: How to survive in the jungle of Enterprise Architecture Frameworks. Trafford, Victoria (2004)
31. Silver, B.: BPMN: Method and Style. Cody-Cassidy Press, Aptos (2009)
32. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
33. Van Nuffel, D., Mannaert, H., De Backer, C., Verelst, J.: Deriving Normalized Systems elements from business process models. In: Boness, K. (ed.) Proceedings of the Fourth International Conference on Software Engineering Advances, pp. 27–32. IEEE Computer Society, Los Alamitos (2009)
34. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer, Heidelberg (2007)
35. Winter, R.: Guest editorial - Design Science Research in Europe. European Journal of Information Systems 17(5), 470–475 (2008)
36. Zachman, J.A.: A framework for information systems architecture. IBM Sys. J. 26(3), 276–292 (1987)
37. Zhao, J.L., Tanniru, M., Zhang, L.J.: Services computing as the foundation of enterprise agility: Overview of recent advances and introduction to the special issue. Information Systems Frontiers 9(1), 1–8 (2007)

# Harness Mobility: Managing the Off-Task Property

Bo Andersson[1] and Christina Keller[2]

[1] Lund University, School of Economics and Management
Ole Römers väg 6, 23263 Lund, Sweden
bo.andersson@ics.lu.se
[2] Uppsala University and Jönköping International Business School
P.O. Box 1026, 55111 Jönköping, Sweden
christina.keller@ihh.hj.se

**Abstract.** Technological advancements in mobile computing and wireless networks open up to new applications and new user-groups in the mobile workforce. However, a considerable part of the mobile workforce, such as e.g. drivers or healthcare staff, is chiefly performing other tasks than interacting with their computers. As a result, they are not able to pay attention to computer interaction, making them mainly off computer tasks. The aim of the paper is to develop a design theory to manage off-task situations in mobile computing. Interviews were performed with developers of an information system comprising mobile devices for drivers. From the interviews, a design of an artifact and a design theory based on a strategy to automate routine administration task using place awareness is presented. The eight components of IS design theory by Gregor and Jones is applied as a theoretical framework.

**Keywords:** Mobile Information Systems, Off-Task, On-Task, Design Theory, Mobile Users, Location Awareness.

## 1 Introduction

That mobile information systems are spreading and that more users are connected to their organizations via handheld computers and wireless technologies is an undisputable fact in the year of 2010. The benefits of these applications and systems are widely discussed [1, 2] as well as the specific physical features of handheld computing devices, such as small screen, cumbersome input, wireless connection et cetera [3, 4]. One of these physical features that has reached attention is *location awareness* [5-7]. This feature has been used in several innovative applications ranging from advertising to tourist information on historical buildings [8, 9]. Location awareness describes the ability and features of the device to know its actual location and can be achieved by triangulation, by GPS (Global Positioning System) or other techniques. Even more interesting is that location awareness can coincide with a more subjective property of mobile computing, the so-called *off-task* property. It concerns the condition that the mobile user chiefly is doing other tasks than interacting with the mobile device. This makes the user off-the-computer. The opposite is of course when the user is solving some task on the computer making the user on-the-computer (on-task). For instance, a

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 258–269, 2010.

taxi driver is mainly driving, chatting with passengers and therefore mainly being off the computer, being off-task [5].

The term “mobile” is an ambiguous concept. In this study, the mobile workforce in focus are drivers and in this sense truly mobile. This narrows down the scope of the study and excludes, e.g. local mobility, travelers and some aspects of mobility as by any media [10, 11].

### 1.1 Objectives

A considerable amount of applications for mobile users are due to be developed. In the current situation, research informing developers on lessons learned from successful design ought to have high relevance, as this knowledge can, transformed to design propositions, enhance the systems to be built. 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 devices due to, e.g. contextual concerns as dynamic use situations and the small form factor [12]. This renders a certain interest in design theory expanding the design space to embrace also mobile devices such as handheld computers.

There has been a considerable amount of work done on new and innovative applications within the domain of mobile computing and mobile information systems, but the underlying rhetoric has often been one of technology-push [13, 14] rather than market-pull. Market-pull typically represents a situation where a problem exists alongside with a lack of solutions, the technology-push situation is one where solutions are looking for problems to solve [15]. In this paper the rhetoric is market-pull: An aspect on mobile computing (off-task) is problematic, let’s test if place awareness can help to solve this problem.

In order to harness the mobility that the handheld computer can offer, there are reasons to extend the design space and acknowledge that the user in most cases is not particularly interested in using the handheld computer during driving or accomplishing other task belonging to their core work process. Instead, the design of applications for the mobile workforce should relieve the user from tasks using in-built features of the handheld device [5]. The objective of this study is to present a design of an artifact and a design theory using place awareness to relieve the user from administration of tasks not belonging to the core process, thus supporting the workflow. The findings is based on a case study of the development of a information system used by the County Council of Kalmar in Sweden.

Some clarifications on what sort of terms used are helpful to the reader of the paper. The mobile device is a handheld computer that can be a Personal Digital Assistant or a mobile phone with some computing ability (often labeled smart phone). Henceforth the more generic term *handheld computer* will be used in the text illustrating a small form factor device with computing and wireless networking abilities. The label *mobile computing* portrays the *use* of handheld computers and a *mobile information system* depict a computerized information system with parts that are mobile, i.e. handheld computers.

### 1.2 Background

The mobile workforce is often occupied with other task than working with their computer, a property often labeled as off-task. This property is studied in several empirical studies on the mobile workforce and the characteristics of their computer support. Ecologists working in Kenya with wild life protection and how they were supported by a handheld computer and a special application has been studied [16, 17] as well as mobile service technicians at a telecom company and consultants working with certifications of marine vessels [18]. Some of the findings of these studies were related to the off-task property. It was found that the mobile user was occupied with other task than computer work, and that the regular desktop computer users mainly had their computer in sight. These observations put demands on the developer to manage differences between mobile users and regular desktop computer users.

One suggested strategy to manage the off-task property is to offer the mobile user so-called *active behavior*. This means that the device itself should take action when certain circumstances occur, thus relieving the user from computer interaction. As a result, the mobile user is being able to stay focused on the main task, which is something else than managing a handheld computer [5].

Both stationary and mobile devices can be aware, or at least store information of its geographical location. This attribute is generally labeled *place awareness*. In this paper the storage of items location in a database, such as Wal-Marts use of RFID [6] for logistics are outside the scope. Place awareness can be categorized into area and location awareness. Area is relevant to both stationary and mobile units and could be e.g. time zones or VAT regions. Given an area, certain conditions may adhere. Location awareness is a more distinct phenomenon as it illustrates the device's ability to know its current location at a given moment. This location awareness ability is a property belonging to mobile devices. There is a fundamental difference between area and location awareness, as the area is known in advance, while the location is knowable by the application even before the user is aware of it. It can be used by applications in order to specify the place, which brings reduced need for input from the user. By location awareness, the application can save your location and use information about previously known places, such as waypoints. The application can also send its location to other devices, which e.g. can be used to keep track of where other colleagues are located [5]. There has been a considerable amount of work on location-based services mainly of conceptual type or for marketing [19-21] but less work has been done on supporting the mobile workforce with applications using location awareness. With the introduction of GPS, the location of a device can be calculated with an accuracy of a few meters.

## 2 Research Approach

Design science research includes the building, or design, of an artifact as well as the evaluation of its use and performance [22]. Research frameworks of design research typically include activities of theory building, solution technology invention, and evaluation, which can be naturalistic or artificial. The research framework and the connections between the research activities are presented in figure 1. The arrows show that the researcher, over time can alternate between different activities as the research aim dictates [23].

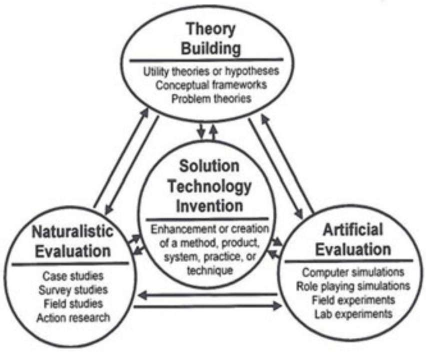



**Fig. 1.** Framework and context for design research [23]

Our study can be mapped according to Venable's [23] framework; *Problem theories* initiating the research is the off-task property, a property identified by ethnographically influenced studies. The implemented solution at the County Council of Kalmar is part of the *Artificial Evaluation.* The *Naturalistic Evaluation* in 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 artifact and a design proposition. Building on this knowledge, a design theory is developed, thus informing *Theory Building.*

In IS design science research, scholars have suggested different ways to identify what problems to address, for example case studies or focus groups [24]. The approach used for identifying problems, a case study, was a mix of several methods and techniques. One of the authors attended a presentation of the system, a presentation focusing on the development of the system and notes were taken during presentation. Three semi-structured interviews with the developers of the WinHAST system (a more detailed description of WinHAST in following section) with duration of 60-120 minutes each were made and field notes were taken. Studies of online material describing the system were conducted with focus on functionality of WinHAST. Complementing information was gathered by telephone interview with the manager of the system to verify the developer's statements on perceived benefits. The empirical data collection was broad and covering several aspects of the development of WinHAST as active behavior, remote administration of devices, support problems with geographical distributed staff, technological risk management and synchronization issues. In the analysis, decisions and assumption made on off-task and active behavior were focused. The parts of the implemented solution that managed these two aspects were studied closer.

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 will help. The contextual dependency and the condition that design propositions must be interpreted in a specific setting also indicate that design proposition is a more suitable label than technological rule [25].

## 3 Findings from the Case Study

In Sweden the County Councils offer *service travels* to the citizens. A service travel is a transportation of people to and from health care units in the county council, not to be mistaken with ambulance transports. Service travels manage only non-urgent travels like scheduled medical consultations or an appointment with an optician for those who cannot manage the transport on their own. Local transportation firms as taxi firms or bus firms make the actual transportations. To manage the vehicles a logistic system, WinHAST, was developed. WinHAST is an information system that monitors all vehicles on duty. A billing function is also implemented calculating mileage, time, and fares. In the vehicles a handheld computer with GPS reports to WinHAST the vehicles location. Approximately 170 vehicles are equipped with handheld computers connected to the WinHAST system. A dispatch central in the municipality of Högsby administrate the booking and invoicing. One of their tasks is to optimize the transportation in order for the vehicles to transport as many clients as possible at the same time. This is made by route optimizing and just-in-time adjustments on planned routes. The staff at the dispatch office monitor all vehicles in service and are able to trace their actual location at any given moment. The interface of the monitoring is presented in figure 2.

The drivers receive their driving assignments through the handheld computers (i.e. mobile phones). The interface of the drives transport assignments on the handheld computers is illustrated in figure 3.

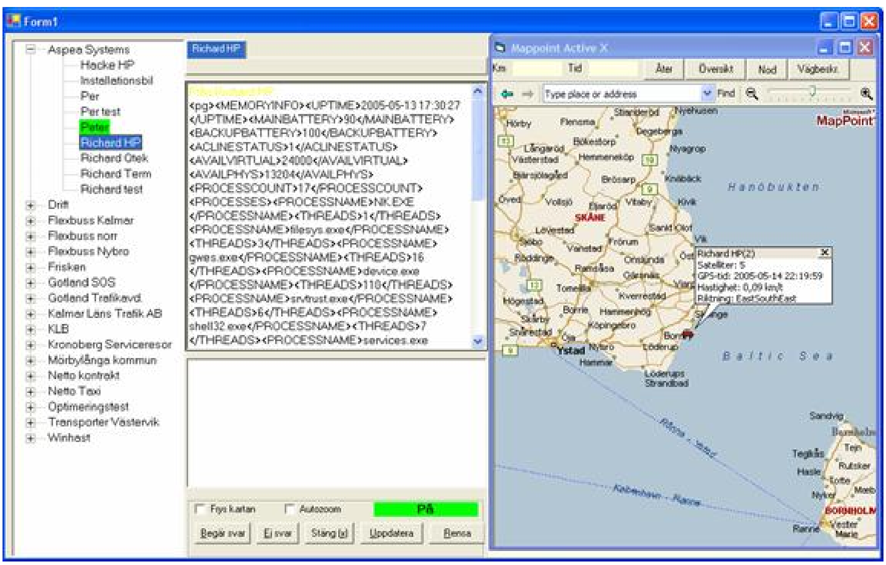



**Fig. 2.** The interface offered to the dispatchers, at the left the dispatcher can filter which vehicle or which transportation firm's fleet to view in the map at the right. In this particular view only one vehicle is selected.


**Fig. 3.** The driver's user interface displaying the destinations and passengers

The drivers are mainly off-task during their workday accomplishing other task than those requiring computer interaction. To reduce the distraction that the mobile device and the administration of driving instructions cause, the developer set off to create a system with active behavior, that is, implementing functionality that automated some tasks and making the application responsible for decisions that otherwise the user must make. Aspea System, a small software development firm, built the system during 2002-2009. The developers worked close to the customer, with an iterative approach. This enhanced the perceived usability of the system due to a high degree of user involvement. Several aspects on how to manage properties of mobility was tested and elaborated as different interfaces on the handheld computers. The developers realized that driving and managing the computer was two different tasks that could conflict, or at least distract the driver. Direct manipulation [26] was conceived as undesirable due to the possible distraction of the driver, and other strategies were investigated. The dispatcher part of the system was using two map engines, one with road information (distances and speed limits) and one to enhance the usability for the dispatcher with a more ordinary map view. Every address was mapped with coordinates possible to use in the GPS applications.

By using the coordinates the developer constructed a *TuneIn* function relying on the geographical data about the start destination and end destination of a certain driving assignment. When the dispatcher assigns transportation by sending start destination, end destination and time to a driver the system stores information of the coordinates of the places. During the transportation the mobile device sends its position every 60 seconds, thus updating the position on the map in front of the dispatcher (figure 1). When the driver approaches the address and the speed decreases the systems starts updating the position more often and when the average speed is low enough and the car close enough to the destination in the transportation order is activated and the billing starts, this is shown in the screen shot from the administrators console of WinHAST in figure 4. The system continues to automatically calculate time and traveled distance until the car (i.e. the handheld computer) slows down close enough to the end destination of the transportation. By the TuneIn function the driver is relieved from administration of travels and the invoicing is automated.

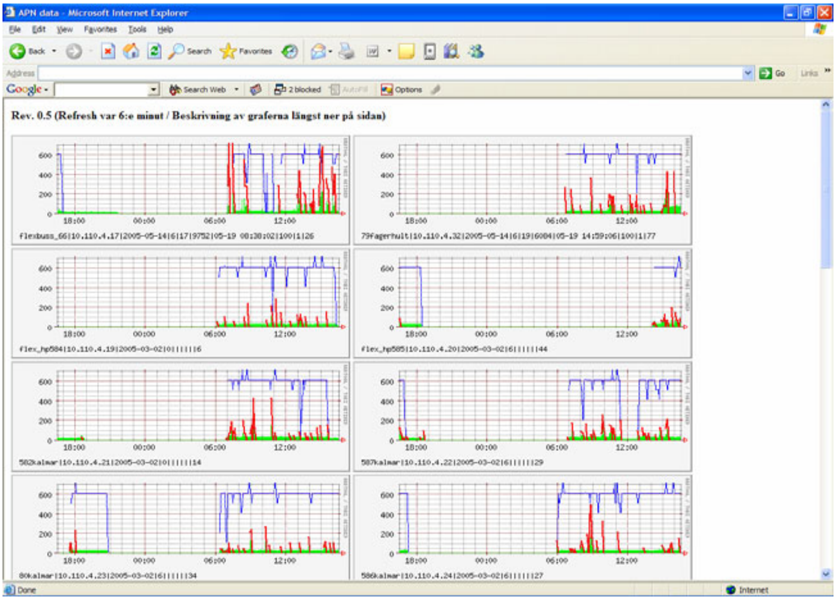



**Fig. 4.** Example on stored information on vehicles movement and satellite tracking

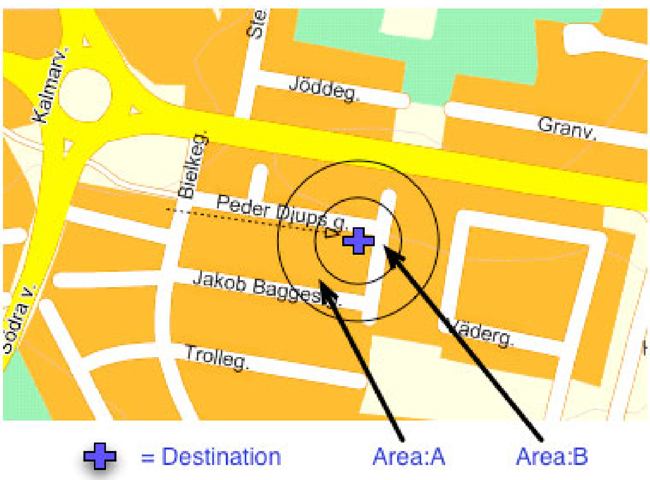



**Fig. 5.** In the working example vehicle is travelling eastward on Peder Djups g (along the dotted arrow), and via the handheld computer and GPS the vehicles actual position is mapped to the destinations position

The procedure of a picking up a passenger is exemplified with the following case (real/actual coordinates are used): Transportation is ordered from *Peder Djups g*. The coordinates for the destination is *N 56° 43′ 59″ E 15° 56′ 17″*.

If traveling on street Peder Djups g. eastward (the dotted arrow in figure 5) the car enters Area A at N 56° 44′ 1″ E 15° 56′ 10″. When entering this area the systems starts to collect coordinates on 10 seconds intervals instead of the regular 60 seconds interval during regular transportation away from destinations. When entering Area B at N 56° 44′ 0″ E 15° 56′ 15″ the systems starts to calculate the average speed and if the car stops during a predetermined time the system automatically registers a successful pick up (see figure 5). The procedure is the same when reaching a destination with the obvious difference that it is about a drop of this time.

## 4 Towards a Design Theory for Managing the Off-Task Property of Mobile Systems

When engaged in building artifacts, we are engaged in design [27]. 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 user. A design theory is suggested and is evaluated according to Gregor and Jones framework for information systems (IS) design research [28].

As the theoretical framework for portraying the properties of the suggested design theory, Gregor and Jones' 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 artifact that could be constructed: (1) purpose and scope, (2) the constructs, (3) the principles of form and function, (4) the artifact mutability, (5) testable propositions, and (6) justificatory knowledge. The first five components have direct parallels to components proposed as mandatory for natural sciences theories [23, 25]. The sixth component has been added to provide an explanation of *why* the design works or not. Purpose and scope says "what the system is for". To understand an artifact it is necessary to understand the context and the circumstances it operates within. To make a valid description of purpose and scope, the context and reason for the design theory existence must be clarified. Constructs concern representations of the entities central in the design theory. They can be assembled of words, diagrams or mathematical symbols. Principles of form and function describe how the artifact is constructed, a blueprint of the artifact. Artifact mutability illustrates the evolutionary properties of IS artifacts, that it is difficult to define a design due to this ever-changing material. An ambition should 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 strive to achieve these type of propositions. Justificatory knowledge concerns the explanatory knowledge that links goals and materials.

The two additional components are (7) principles of implementation and (8) expository instantiation. Principles of implementation concern the means and processes by which the design is brought into being, including agents and actions. Expository instantiation is a physical implementation of the artifact that can assist in representing the design theory in the form of a construct, model, method or instantiation. The framework is presented in table 1.

Following van Aken's [25] advice in formulation 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 relieve the mobile workforce from routine administration and allowing the workforce to stay on main task (X) in an environment predisposed repeatable information handling task related to location (Y) then implement location aware solutions as TuneIn to automate information handling task automatically (Z).* The proposed design theory for managing the off-task property of mobile information systems using Gregor and Jones' framework [28] is summarized in table 2.

**Table 1.** Eight components of an Information Systems Design Theory [28]

| 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 artifact 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 artifact, either product or method/intervention. |
| Artifact mutability | The changes in state of the artifact anticipated in the theory, that is, what degree of artifact 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). |
| Additional components | |
| Principles of implementation (the *causa efficiens*) | A description of processes for implementing the theory (either product or method) in specific contexts. |
| Expository instantiation | A physical implementation of the artifact that can assist in representing the theory both as an expository device and for purposes of testing. |

**Table 2.** Eight components of an Information Systems Design Theory for managing the off-task property of mobile information systems

| Component | Description |
|---|---|
| Core components | |
| Purpose and scope (the *causa finalis*) | The aim is to develop a system with functionality to manage the off-task property of the mobile workforce. |
| Constructs (the *causa materialis*) | Mobile workforce, off-task users, location awareness, active behavior of an mobile information system |
| Principle of form and function (the *causa formalis*) | Administration of remote mobile users tasks through an information system, wireless communication and handheld computers by positioning technologies. |
| Artifact mutability | Suggestions for improvement during the development phase were given from the users due to high degree of end-user involvement during the seven years of iterative development. |
| Testable propositions | If you want the mobile workforce to stay on main task in an environment with information handling tasks related to location then implement location aware solutions to automate information handling task. |
| Justificatory knowledge | The underlying perspectives stem from design science and empirical studies of the properties of mobile workforce, such as active behavior, off-task and location awareness. |
| Additional components | |
| Principles of implementation (the *causa efficiens*) | The system has been incrementally implemented during the development phase. |
| Expository instantiation | The system has demonstration functionalities suitable for training and evaluation. |

## 5 Conclusions and Discussion

Considerable parts of the mobile workforce are mainly off-task during their working day. This off-task behavior is regarded as a specific property of mobile computing by several researchers and portrays the users relation to the computer. Working with the computer is considered as on-task and working with other tasks is regarded as off-task. One strategy to manage off-task is to implement functions based on active behavior. Active behavior is a descriptor on functions performing without a user calling for the function. If the ambition is to harness mobility and benefit from this technology we need to look at new ways to use the available technology. In this paper we have suggested a design theory based on location awareness as a means for active behavior to manage the off-task property. The design theory originates from the lesson learned by developers during the development of WinHAST, a system for administration of transportation. The central aspect is automation of administration of pick ups and drop offs.

The bottom line is that we need to trust the technology in order to automate and release the user from routine tasks although this can be doubtful. The authors' first impression on the TuneIn was that it seemed to be an unreliable solution. What if the driver stopped for a pedestrian close to the destination? Wouldn't the system create a large amount of faulty invoices? According to the managers the amount of faults origination from the system misinterpreting the pickups was marginal and easy to correct due to the large information on geographical movement stored in the system. An advantage was that problems stemming from drivers missing to create receipts were reduced made this part of the system a success even among the dispatchers. Also the County Council of Värmland is at the time of writing implementing WinHAST, making a sound argument of the perceived success of the system and the TuneIn feature. Some limitations in the study are identified, such as the selections of respondents. The informants are only the developers and managers at the dispatch office; the driver's opinion on the functionality of TuneIn would be valuable complement to the study.

Future work on the proposition is of course needed. Future work will also consider if it is possible to apply this design theory to other areas of application for the mobile workforce. In this paper a developers perspective is applied, the users belief on privacy and trust related to the positioning ability should be elaborated in future work.

**Acknowledgments.** We would like to thanks the developers at Aspea System for their support and their willingness to inform us on the functionalities of the dispatch system they have developed.

## References

1. Barnes, S.: Enterprise mobility: concept and examples. International Journal of Mobile Communications 1(4), 341–359 (2003)
2. McIntosh, J., Baron, J.: Mobile commerce's impact on today's workforce: issues, impacts and implications. International Journal of Mobile Communications 3(2), 99–113 (2005)
3. Satyanarayanan, M.: Fundamental challenges in mobile computing. In: Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing, Philadelphia, Pennsylvania (1996)
4. Tarasewich, P., Nickerson, R.C., Warkentin, M.: Issues in Mobile E-Commerce. Communications of the Association for Information Systems 8(1) (January 28, 2002)
5. B'Far, R.: Mobile computing principles: designing and developing mobile applications with UML and XML. Cambridge University Press, New York (2005)
6. Junglas, I.: On the usefulness and ease of use of location-based services: insights into the information system innovator's dilemma. International Journal of Mobile Communications 5(4), 389–408 (2007)
7. Leonhardt, U., Magee, J., Dias, P.: Location service in mobile computing environments. Computers & Graphics 20(5), 627–632 (1996)
8. Cheverst, K., Davies, N., Mitchell, K., Friday, A.: The role of connectivity in supporting context-sensitive applications. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 193–207. Springer, Heidelberg (1999)
9. Long, S., Kooper, R., Abowd, G., Atkeson, C. (eds.): Rapid prototyping of mobile context-aware applications: The cyberguide case study, New York (1996)
10. Kristoffersen, S., Ljungberg, F.: Representing Modalities in Mobile Computing - A Model of IT use in Mobile Settings. In: Proceedings of Interactive Applications of Mobile Computing, Rostock (1998)

11. Makimoto, T., Manners, D.: Digital nomad. Wiley, Chichester (1997)
12. Fällman, D.: In: Romance with the Materials of Mobile Interaction: A Phenomenological Approach to the Design of Mobile Information Technology (2003)
13. Nichols, J., Myers, B.A.: Controlling Home and Office Appliances with Smart Phones. IEEE pervasive computing 5(3), 60–67 (2006)
14. Toye, E., Sharp, R., Madhavapeddy, A., Scott, D.: Using smart phones to access site-specific services. IEEE pervasive computing, 60–66 (2005)
15. Eppinger, S., Ulrich, K.: Product design and development. McGraw-Hill, New York (2003)
16. Pascoe, J., Morse, D., Ryan, N.: Developing personal technology for the field. Personal and ubiquitous computing 2(1), 28–36 (1998)
17. Pascoe, J., Ryan, N., Morse, D.: Using while moving: HCI issues in fieldwork environments. ACM Trans Comput-Hum Interact. 7(3), 417–437 (2000)
18. Kristoffersen, S., Ljungberg, F.: "Making place" to make IT work: empirical explorations of HCI for mobile CSCW. In: Proceedings of the international ACM SIGGROUP conference on Supporting group work, Phoenix, Arizona (1999)
19. Mennecke, B., Strader, T.: Where in the world on the Web does location matter? In: A framework for location based services in m-commerce. AMCIS 2001 Proceedings, p. 90 (2001)
20. Rodden, T., Friday, A., Muller, H., Dix, A. (eds.): A lightweight approach to managing privacy in location-based services (2002)
21. Tilson, D., Lyytinen, K., Baxter, R. (eds.): A framework for selecting a location based service (LBS) strategy and service portfolio (2004)
22. 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)
23. Venable, J.: A framework for design science research activities. In: Proceedings of the Information Resource Management Conference, Washington D.C. (2006)
24. Hevner, A., Chatterjee, S.: Design Science Research in Information Systems: Theory & Practice. Springer, Heidelberg (2009)
25. Van Aken, J.: 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)
26. Hutchins, E.L., Hollan, J.D., Norman, D.: Direct manipulation interfaces. Human-Computer Interaction 1(4), 311–338 (1985)
27. Simon, H.: The Science of the Artificial. MIT Press, Cambridge (1969)
28. Gregor, S., Jones, D.: The anatomy of a design theory. Journal of the Association for Information Systems 8(5), 312–335 (2007)

# Dynamically Generating Context-Relevant Sub-Webs

Art Vandenberg[1], Vijay K. Vaishnavi[2], Saravanaraj Duraisamy[2], and Tianjie Deng[2]

[1] Georgia State University, Information Systems & Technology,
P.O. Box 3994, Atlanta Georgia, 30302-3994, U.S.A.
[2] Georgia State University, Computer Information Systems Department,
P.O. Box 4015, Atlanta Georgia, 30302-4015, U.S.A.
{avandenberg,vvaishna}@gsu.edu,
{sduraisamy1,tdeng1}@student.gsu.edu

**Abstract.** There is unprecedented growth of Web information but challenges on mining this vast information resource remain. This paper addresses designing an effective prototype tool that dynamically generates sub-webs of information from a web-based resource (World Wide Web or a subset). Sub-webs present context-relevant results to individuals or groups. Given that the prototype tool is technically implemented from multiple components, each of which has efficacy, there still remains the challenge of devising an appropriate evaluation of the complete model. This is difficult when the search scope is the entire World Wide Web and a vast number of result pages are technically good on Recall but low on Precision. This paper describes an iterative approach to finding an effective technical prototype using an evaluation method that can a) reasonably model the search environment of the World Wide Web and b) provide convincing metrics for evaluating efficacy of solutions.

**Keywords:** Context, Sub-Web, Web Mining, Evaluation, Metrics.

## 1 Introduction

There is unprecedented growth of information available on the Web in all fields of human endeavor but challenges on mining this vast digital information remain. This paper addresses the goal of a research prototype tool that can generate sub-webs of information dynamically from a web-based resource (the entire World Wide Web or a subset thereof such as the NSF or NIH portal) – where sub-webs will present context-relevant results to users (individuals or groups).

Researchers in all fields of human endeavor, including science and engineering, recognize the potential and the challenges of exponential growth of information in the World Wide Web [1], [2], [3] and [4]. Taming the Web has spurred considerable research and commercial activity, such as [5], [6], [7] and [8]. The available approaches can be broadly grouped into search engines [5] and [9], directories [10], [11] and [12] or web user adaptation and personalization systems [4], [13], [14], [15], [16], [17] and [18].

Simple keyword searches may return hundreds (even thousands) of individual web pages but often with two deficiencies: (1) keywords may not explicitly reflect the

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 270–285, 2010.

relevant *context* of a user's requirements and (2) the results may not provide context-relevant, thematic interest, the (*cognitive) perspective* – such as "protein secretion" or "financial econometrics" – for a user's point of view. Yet, could we improve the process by providing web resources representing user interest, then applying social network based ontologies (such as Open Directory Project (ODP) [12] or WordNet [19]) and search engines to dynamically generate context-relevant sub-webs?

A practical, scalable, dynamic web mining solution is envisioned to help the user process existing web-based resources to generate specific, sub-webs of information content. This is especially important for researcher communities where data deployed on the World Wide Web are characterized by autonomous, dynamically evolving, and conceptually diverse information sources as described by [20], [21] and [22]. Our goal is a prototype artifact but we focus here on evaluation strategy and testbed.

### 1.1 Real World Scenario, Problem Relevance

The following real-world scenario forms the motivation as well as one of the application domains for the research. Campuses have extensive web-based information sources comprising faculty web pages, research missions, key focus areas, or university research administration guidance about funding opportunities and successful awards. Further, national and other funding agencies make readily available digital information on core programs, strategic areas of investigation and abstracts of awarded research. Such digital resources are part of "a huge ecosystem of services and tools [that] will emerge around data mesh instances. Such tools and services will allow us to move beyond current practice of information management by incorporating more automation" [23]. As motivation, consider that Vice Presidents for Research seek to improve research funding success by encouraging interdisciplinary research collaborations that combine individual researcher skills around a university's strategic areas of strength. Identifying research teams helps universities appropriately target research proposals to funding agencies, focusing university resources on research programs relevant for researchers and the university. Indeed, a prototype tool is envisioned that may be used by: 1) Researchers to identify potential teams of research collaborators; 2) Research teams to identify potential funding opportunities relevant to their research interests.

### 1.2 Definitions of Terms

*User Context* – Information (sets of related URLs, documents, etc.)
*Sub-Web* – A representation of web-based content with pages presented as a pruned web structure retaining only context-relevant result pages.
*Context-Relevant* – Results closely related to user interest, not required to be exact (as with keywords) but having thematically matching interest.

## 2 Related Work, Context of Prototype and Its Implementation

We are cognizant of a rich body of related work and note some areas of interest.

Information Retrieval – Research leveraging context. A special browser is described that uses ODP to categorize results and disambiguate queries through interaction with users, though user context is not addressed [24] and [25]. Applying data mining methods to Web search logs (context of the search process) can present search results in a better way [26]. A novel technique to supplement term frequency measure by dynamically generating a context-based measure of document term significance during retrieval is described by [27]. Such context matching significantly improves retrieval results when coupled with user supplied context documents. A user personalization model that depends on context, such as user location, may improve performance results by relaxing context constraint and replacing a hierarchical attribute by one at a higher level [28].

Relation-based searches based on the Semantic Web RDF relation tuples can improve results [29], [30]. Well-established concepts like "collaborative filtering" can be relevant [15] and [31]. The use of topical n-grams [32], rather than relying only on a bag-of-words approach, can enhance the discovery of topics and topical phrases.

Ontology – As early as 1995, the importance of ontology and ontology structure for efficient information integration was discussed [33] and [34] and the significance of ontology in Enterprise modeling examined [35] and [36]. Given the Web being a large, complex, evolving information set about virtually every aspect of human endeavor, it may not be practically possible to construct and follow a well-defined ontology. Experiments have been conducted to enable automated ontology learning from domain text using Natural Language Processing and Machine Learning techniques [37] and [38]. Significant efforts have been made to organize the information available in the Web using directory structure (e.g. ODP), a form of knowledge network [39]. ODP, as of now, has categorized 4,525,920 pages into 590,000 categories with 84,588 editors and is considered to be the largest and most comprehensive human-directory on the web. Social bookmarking and Folksonomies have also gathered momentum in classifying and tagging the publicly available Web information [40] but they need to handle semantic heterogeneity. The WordNet natural language ontology has been used to extract concept forests from a document [41]. Using manually built topic models derived from a handcrafted directory resource (socially constructed ODP) can improve retrieval performance [42].

These social network efforts, however, while also challenged by the exponential growth and scale of the Web, are perhaps suited to the evolving nature and inherent ontological drift [43] of Web data.

Classification – Text categorization is an important research problem in Information Retrieval. Use of multiple models may improve classification accuracy [44]. Support Vector Machines have been used for classification with very large-scale taxonomy [45]. Feature selection in text categorization (methods such as Information Gain, Support Vector Machine feature selection, Genetic Algorithm with SVM) can increase efficiency by reducing dimensionality [46]. Organization of search results in meaningful groups using clustering and faceted categorization is discussed by [47].

Clustering – A knowledge-based approach to organizing retrieved documents (clustering) has been examined by [48], [49] and [50]. A probabilistic method has been applied to online document clustering for novelty detection [51]. A time-based self-organizing model for document clustering by considering non-stationary features of real world document collections has been shown to be useful [52]. Hierarchical Clustering to organize the on-the-fly search results drawn from 16 commodity search engines is discussed in [14].

Document clustering based on phrases provides improved results [53]. Optionally search results can be presented to users as SOM-based clusters, and could be adapted to use context-relevant phrases [54]. Using SOM for the clustering of textual documents based on conceptual representation of texts and n-grams can provide "clustering based on concepts" rather than based on n-grams [55], though the disambiguation technique is perhaps weak in that using the first concept for a term from WordNet may be the best overall concept.

A novel passage-based approach to re-ranking search results with respect to the initial document list is described by [56]. SOM clustering is well-suited to presenting high-dimensional data [57] by compressing information while preserving important topological and geometric relationships of the primary data elements.

## 3 Problem Abstraction and Research Approach

Given a web-based knowledge resource K such as the NSF or NIH portal, or the entire World Wide Web, and given a set of documents representing a certain context, C, our general research question is to find documents that are *relevant* to C. To scope the research problem we assume that K is a set of documents and C is a small set of documents drawn from K. We would like the documents found to be organized as a sub-web that is context-relevant to C. As an example assume that K is the NSF portal and that C is a subset of abstracts of projects funded by NSF.

Assume that K is organized as a tree (or an acyclic directed graph), T, according to topic areas. Let N be the lowest common ancestor of the context documents, i.e. it is a node in T such that C is a subset of the documents in the sub-tree rooted at N but is not contained in the set of documents in the sub-tree rooted at a child of N. For the example of the NSF portal, K is organized into a tree (actually an acyclic directed graph) of topic areas. NSF portal is organized into topic areas, which are further organized into sub-topic areas, etc. (Fig. 1) but the classification is only partially according to content of the topic areas; in many cases the classification is ad hoc and the documents are misclassified. The NSF portal is classified as Biology, Computer, Information Science & Eng. (CISE), Cross-Cutting NSF-wide, Cyber-infrastructure, Education, etc. CISE is further classified into Computing and Communication Foundation (CCF), Computer and Network Systems (CNS), and Information & Intelligent Systems (IIS). IIS is further classified into Core Programs and CISE Cross-Cutting Programs. Core Programs is further classified as Human-Centered Computing (HCC), Information Integration and Informatics (III), and Robust Intelligence (RI). In this example, N could be a lower level node such as III or could be a higher level node such as IIS.

Let us now turn to the operationalization of the concept, *relevance*. We say that *a document D (not in C) is relevant to C if and only if D is in the set of documents contained in the sub-tree rooted at the parent of N.* We are purposefully not restricting the definition of relevance such that a document is relevant to C if and only if D is in the set of documents contained in the sub-tree rooted at N. This will be too restrictive since it will result in excluding documents that pertain to the topic representing C. On the other hand, it may be useful to relax the definition of relevance even further to mean that D is relevant if and only if D is in the set of documents contained in the subtree rooted at the grandparent of N.

For the NSF example (Fig. 1), let us assume that N is III, i.e. all the documents in C are abstracts classified as III (Information Integration and Informatics). Here D is relevant if it is contained in III, HCC, or RI. On the other hand, if N is the node IIS, that means that all the abstracts in C are not contained in a specific sub-area such as Core or Cross-Cutting. In this case an abstract D is relevant if it is within CISE.

If K is always properly classified into a tree (or directed acyclic graph) of documents and the classification is dynamically maintained, then one can easily find the documents relevant to C, and thus the problem is not a research problem. All one would have to do is to find N in the classification tree and then return all the documents in the sub-tree(s) rooted at the parent of N. However, the problem is a research problem since one cannot assume the existence of such a dynamically maintained classification. There may exist a classification such as that for the NSF portal but it may not be a classification purely based on the contents of the documents. In most cases such a classification may not even exist, for example when K is the entire World Wide Web or some arbitrary subset thereof.

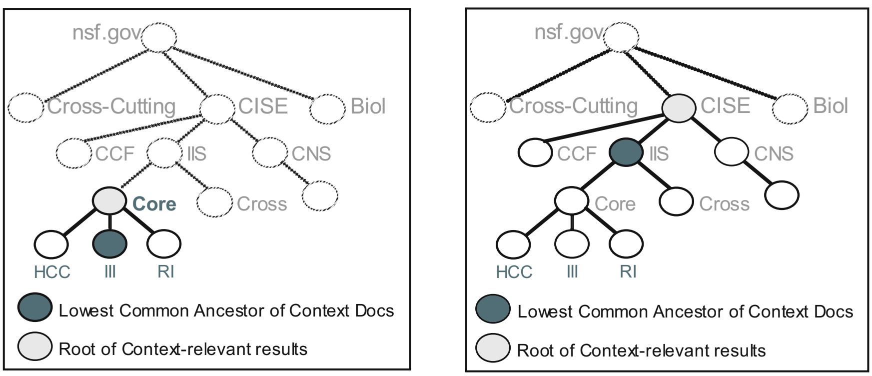



**Fig. 1.** Examples of Context-Relevant Search Results

In our research we are using Reuters Corpus, Volume 1 (RCV1) containing 810,000 Reuters, English language News stories as a test-bed. The corpus is classified using a topic non-cyclic directed graph; a document may belong to multiple categories. The corpus is well classified and has been used extensively in information retrieval research. Based on this fact we are assuming that the research results will have external validity. In other words, if a technique discovered using the test-bed finds documents relevant to C with a reasonably good value for the usual metrics (recall, precision, F-measure, etc.) then it will prove to be a reasonably good technique for finding documents relevant to a certain context set (C) for any K (the entire World Wide Web or a portal such as NSF or NIH). The external validity will eventually need to be confirmed using field studies and we note that in a real world (more general) scenario such as the World Wide Web, metrics other than Recall and Precision (e.g. panel of experts, focus groups) may better provide alternative evaluation [58], [59] and [60].

We are particularly interested in this more general approach, where a user search is not simply *an answer* to a keyword query ("avatar definition" or "DESRIST 2010"), but a more complex operation such as *given representative research abstracts of several faculty, discover Web resources reflecting a likely research theme.*

We consider this problem as perhaps of philosophical interest related to human tool use. Adams and Aizawa [61] discuss the bounds of cognition with respect to brain-tool and brain-world interactions. We seek to design a prototype tool that might go beyond mechanical information processing (returning pages based on pre-indexing of terms) so as to discover relevant information on the web in a dynamic way, leveraging available information, but not limited by strict interpretation of keywords.

### 3.1 Research Approach

The basic idea of our research approach is to analyze a user's context documents with information retrieval techniques, process the context data with ontologies to find and refine phrases for search engine input, and then generate search engine results as a structured sub-web that present context-relevant information to the user.

Our research approach addresses areas that have potential to generate new methods and design principles/theories by:

a. Implementing an innovative, integrated prototype to execute a search paradigm using context, ontologies (such as WordNet) and social networks (such as ODP) to return context-relevant Web information to a user;
b. Generating Sub-Webs (filtered, structured results of searches) by implementing algorithms and techniques for semi-structured or unknown structures of data; and
c. Conducting formative and summative evaluation of the prototype tool using models and metrics appropriate to evaluating dynamically varying datasets.

The proposed approach builds upon related research of the authors' Information Integration Lab in exploring and developing innovative dynamic Web mining techniques for document classification and clustering.

Justification of approach – *Philosophically one may ask, why take this approach to mining digital information? Why explore a tool based on user context, processing with ontologies such as WordNet or Open Directory Project, and using search engines to generate results presented as a sub-web?*

Generally, the answer is two-fold: *The approach a) might work and b) has research merit.*

There is a chance such an approach will work and based on preliminary results of an ongoing design science research prototype, the approach merits further investigation. Two specific factors point to the potential success of the approach:

Dynamic Approach – No dependency on pre-processed, prepared or annotated Web data. The Web is constantly evolving and changing so, if can we develop solutions that work within such an "as is," dynamic environment, all the better.

Scalable Approach – Leveraging the infrastructures that are present, such as existing ontologies being created by communities and social networks around the world, and many search engine infrastructures of web crawlers and indexes.

## 4 Implementation of Research Approach, Preliminary Evaluation

Three subsystems are built on the top of an existing workbench (Fig. 2).

Context/Perspective Extractor – The component addresses the extracting of context/perspective details from information provided by the user (relevant URLs or documents). The challenge is to identify the correct theme (context) and present it to the user in the shortest time possible. To help identify the theme, ODP, a human-edited hierarchy of categories (ontology) with descriptions, WordNet and/or other ontologies are used.

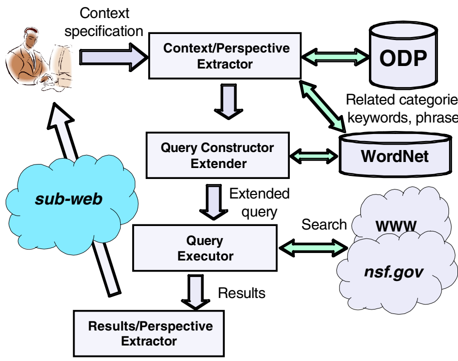



**Fig. 2.** Prototype Component/Process Flow

A number of algorithms can be used to improve the theme(s) identification. One approach is a word-topic occurrence matrix and/or phrase-topic matrix. The user-provided URLs/documents are matched against the word/phrase-ODP topic matrices after Singular Value Decomposition (SVD) by identifying cosine similarities (Fig. 3).

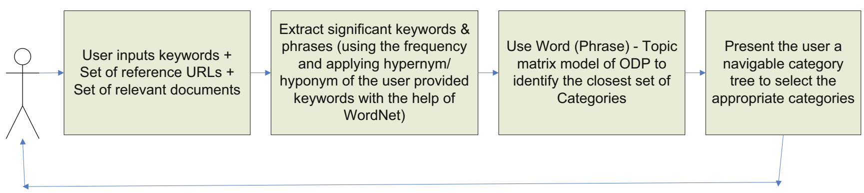



**Fig. 3.** Identifying User Context/Perspective by Word/Phrase-ODP Topic Matrices

Another approach can be *tf x idf (term frequency-inverse document frequency)* – indexing the documents under each topic in ODP, identifying the right set of keywords and phrases for the user-inputted documents (high frequency and common words/phrases across the documents), and then identifying the closest match. WordNet ontology has been used to extract concepts used, rather than keywords, to cluster documents [41]. Retrieval performance can be improved with manually built topic models to refine the original query [42].

Query Constructor/Extender – Semantic query construction is an important part of the architecture and can use ODP or WordNet [19] to disambiguate query terms. Significant work has done about enhancing queries by the industrial ontology group in Finland – http://www.cs.jyu.fi/ai/OntoGroup/InBCT_May_2004.html. WordNet's hypernyms can be used to provide a more generic representation of the query. Enhanced queries can also be constructed with the help of the most common keywords and phrases in the context information provided by the user and by using the keywords/phrases representative of the perspectives selected/provided by the user. Query refinement with lexicons and ontologies has been explored using a methodology called CONQUER (CONtext-aware QUERy processing) [62].

Results/Perspective Extractor – Results can be characterized with a set of keywords/phrases and their respective calculated distance to the retrieved results and presented to the user. This approach also paves the way to explore certain hidden information, based on the user's discretion. We define a Sub-Web as a representation of web-based content such that pages are presented as a web structure retaining only a user's context-relevant result pages and parent pages and links between these pages – with all other pages and links (i.e. the extended World Wide Web) removed.

Additionally Self-Organizing Maps (SOM) [63] may be used to cluster the result sets and provide enhanced contextual perspective. The Kohonen SOM network is very effective for visualization of high-dimensional data [57]. It compresses information while preserving the most important topological and geometric relationships of the primary data elements on the display. The main advantage is to gain insight into the (hidden) structure of data by observing the map, due to the topology preserving nature of SOM. Others describe the use of phrases for document clustering with SOM [53] or use SOM for concept based clustering of textual documents [55]. Organization of search results in meaningful groups using clustering and faceted categorization is discussed by [47].

### 4.1 Datasets

We identified the *NSF Research Awards Abstracts 1990-2003* and *Reuters Corpus Volume 1* (RCV1) datasets to model the problem and for experimental formative and summative evaluation of the prototype. NSF dataset has 129,000 abstracts of awards for basic research, with bag-of-word data for abstracts, and index words for indexing the bag-of-word data. Reuters includes over 800,000 news stories from multiple categories, overlapping and nonexhaustive, with relationships among categories. We integrated these datasets into the research prototype so they are accessible by other components, appearing as URLs – effectively modeling the World Wide Web. The datasets are relatively constrained, compared to the vastness of the World Wide Web, which helps focus the research problem. The NSF dataset is loosely classified (using NSF program titles), while the Reuters dataset is more rigorously classified.

### 4.2 System Workbench Model of Prototype

Fig. 4 shows context processing: 1) Set analysis parameters, e.g. top 10% words (by frequency occurrence) and top 50% of bigrams. 2) Specify Web pages of interest (e.g. NSF ECOSYSTEMS STUDIES abstracts). 3) Select specific Web pages for detailed frequency analysis of words (unigrams), pairs (bigrams) and triples (trigrams). Fig. 5

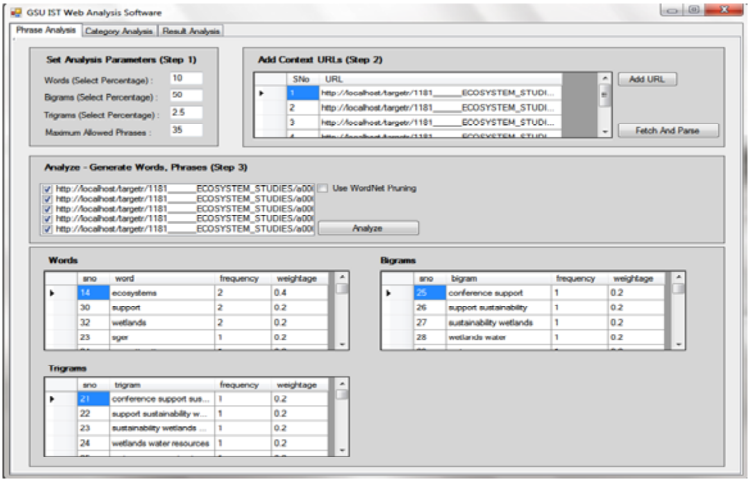



**Fig. 4.** Prototype interface for analysis of context documents supplied by user

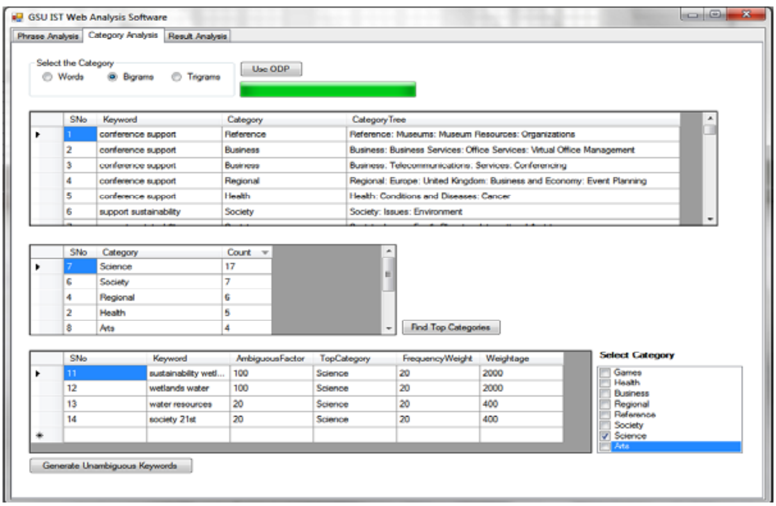



**Fig. 5.** Prototype analysis of bigrams from Figure 4 with suggested search terms

shows resulting analysis of bigrams for context documents supplied by user. Using Open Directory Project categories to disambiguate bigrams, the prototype suggests ODP category of SCIENCE and search terms “sustainability wetlands resources.”

### 4.3 Prototype: Experimental Results and Evaluation

Evaluation of the research approach and technologies used in implementation was conducted by using the Research Prototype as a design artifact that supports experimental testing and validation of hypotheses using formative and summative evaluation.

This workbench-based approach can be a good solution to *plug & play* different algorithms and to analyze effectiveness of each.

We tested different contexts with different combinations of query term enhancement to increase retrieval efficiency. The prototype was validated with formative evaluation – iteratively implementing, testing and revising each component.

Evaluation metrics used for information retrieval include Recall, Precision, Precision at N, and F-Measure (an overall metric) [64], [65] and [66]. It may be that search results from the World Wide Web are not easily measured using standard metrics, especially given that so much of the data remains unauthoritatively classified (or not at all), such that "correct" and "incorrect" notions used with Recall and Precision are difficult [58] and [59].

We propose that the research be considered successful to the degree that it can demonstrate an effective approach for: 1) extracting and enhancing search terms of information retrieval for the user, and 2) supporting dynamic generation of context-relevant results that may be present via a user navigable sub-web.

A formative evaluation methodology guided potential solutions. Preliminary results used the NSF dataset, drawing context from 558 documents in ECOSYSTEMS STUDIES category (one of 640). The 129,000 NSF abstracts represent a Knowledge source and ECOSYSTEMS STUDIES a relevant context.

The basic idea is to explore parameters: e.g number of context documents (NSF abstracts) to use, word sets (unigrams, bigrams, trigrams) to use, and how to enhance terms with ontologies (ODP and/or WordNet) to generate search terms. The prototype workbench facilitates experimentally exploring solutions: a user selecting parameters (percent words or ngrams) and context documents to use (from user editable list), and specifying whether to use Open Directory Project to disambiguate terms or WordNet for term enhancement. In particular, this prototype makes an iterative, formative evaluation of potential solutions very accessible to the researcher. Table 1 shows results of four trials using random sets of NSF abstracts as context. Keywords were extracted and/or supplemented with ODP and/or WordNet hypernyms.

**Table 1.** Summary of Trials to Increase Average Effectiveness

| **Trial** | **# Abstracts in Context** | **# Iterations in trial** | **Average Recall (relevant results in all results)** | **Precision_15* (relevant results in first 15 results)** |
|---|---|---|---|---|
| 1 | 5 | 3 (unigrams) | 25.70% | 42.2% |
| 2 | 5 | 3 (bigrams) | 81.87% | 26.70% |
| 3 | 5 | 5 (bigrams + Word- | 70.34% | 28.00% |
| 4 | 25 | 3 (bigrams + Word- | 75.16% | 46.70% |

**Precision_15=percent relevant abstracts R in first N abstracts (Precision at N).*

These preliminary trials showed some efficacy for using bigrams rather than unigrams (Average Recall 81.87% Trial 2 vs. 25.70% Trial 1) and the prototype was able to *increase "effectiveness" to 46.7%* (Trial 4) albeit at the expense of Recall (Table 1). While encouraging, evaluation improvement was sought: the data (NSF Abstracts) is unauthoritatively classified (only by names of NSF Programs) such that “correct” and “incorrect” notions of Recall and Precision are somewhat imprecise [58] and [59]; and we needed a baseline to compare prototype components (ODP disambiguation, supplementing terms with ODP categories, and/or WordNet).

Having preliminary results demonstrating potential of the prototype tool, we next worked with the Reuters data that are authoritatively classified (more amenable to measurement of Recall and Precision) and designed a baseline evaluation model.

**Table 2.** Representative Trial Using Reuters to Evaluate Prototype Components vs. Baseline

| **Query Format** | **Precision** | **Recall** | **F-Meas** | **Query Terms** | **P_100** |
|---|---|---|---|---|---|
| *unigram baseline* | *0.0846* | *0.1088* | *0.0952* | *energy, canada, pena* | *0.1400* |
| disambiguated by ODP | 0.0671 | 0.2459 | 0.1054 | trade, car, energy | 0.3200 |
| disambiguated by ODP + ODP categories | 0.0660 | 0.2945 | 0.1078 | trade, Trade Shows, car, Car Sharing, energy, Gas and Electricity | 0.4200 |
| disambiguated by ODP + WordNet hypernyms for ODP categories | 0.0682 | 0.2833 | 0.1099 | trade, Commerce, car, Motor Vehicle, energy, Physical Phenomenon | 0.5100 |
| *bigram baseline* | *0.0575* | *0.8665* | *0.1078* | *plan reduce, said prepared, wants plan* | *0.0700* |
| disambiguated by ODP | 0.0943 | 0.2901 | 0.1424 | oil imports, plan reduce | 0.6400 |
| disambiguated by ODP + ODP categories | 0.0931 | 0.3206 | 0.1443 | oil imports, Wholesale and Distribution, plan reduce, Consulting | 0.3900 |
| disambiguated by ODP + WordNet hypernyms for ODP categories | 0.0896 | 0.3090 | 0.1389 | oil imports, Lipid Commodity, plan reduce, Idea | 0.4600 |

The baseline evaluation model was to first measure the performance of the prototype using only terms generated from the *tf x idf (term frequency-inverse document frequency)* matrix to search against the Reuters data. Baseline trials were done using just unigrams (single words) and bigrams. Five news stories were selected at random from a randomly chosen Reuters category containing 452 stories. These 5 documents were processed to create a *tf x idf* matrix from which the 3 top terms (unigrams or bigrams) were selected. A search was performed and Recall, Precision and F-Measure noted. Then for each baseline (unigram, bigram), we tested prototype components to enhance the initial baseline terms: enhancing the terms by:

a. Submitting a term to ODP API, finding five ODP categories; selecting the term if it was unambiguous – i.e. ODP categories were "consistent" rather than varying over multiple categories. Three consistent (disambiguated) search terms were chosen.
b. Supplementing terms found in a) by adding the ODP categories to the search.
c. Supplementing terms found in a) by using WordNet to suggest hypernyms for the ODP categories.

We repeated the experiment multiple (20) times, selecting a random set of five Reuters news stories each time. (A representative trial is shown Table 2). Results indicate that prototype components (disambiguated terms; disambiguated terms enhanced by ODP categories; disambiguated terms with WordNet hypernyms for ODP categories) consistently improved the baseline results:

- In 19 of 20 trials at least one prototype component (using unigram or bigram search terms) provided improvement over baseline.
- In 10 of 20 trails all three components found unigram enhanced queries that improved the baseline search.
- In 11 of 20 trials all three components found bigram enhanced queries that improved the baseline search.
- In 5 of 20 trial all three components, whether using unigram or bigram enhanced queries, showed improvement.

We note that the baseline trials were based on search terms found using a *tf x idf (term frequency-inverse document frequency)* approach. That is, the search terms from context documents were enhanced by the analysis of the tf x idf approach.

## 5 Conclusions and Discussion

We identified a real world problem, modeled an abstraction of the problem, and identified realistic external datasets (Reuters and NSF) that are amenable to experimental validation and summative evaluation using standard metrics.

We have two major steps in dynamically generating context-relevant sub-webs: (1) determining the context-relevant class; (2) determining pages not in the relevant class. The idea is that if all pages were classified into two classes (relevant vs. all the rest) we would like to find pages relevant to our context. What makes the problem difficult is that we cannot actually classify all potential pages and still be dynamic.

(1) Determining the context-relevant class. We leverage an ontology such as ODP to analyze context pages to find "optimal" ODP categories. This set of categories should in effect represent the class we are seeking. The method we have been using so far is to conduct a frequency analysis of words in the context pages and using that information to determine characteristics of the context-relevant class. Even within this method there can be other sub-methods and there certainly are other strategies. For example how about focusing on meta-data about the context pages that may be available? Such meta-data could be titles or tags of the context pages, etc.

(2) Determining the rest of pages – those not in the relevant class. Since all pages are not actually classified with respect to the initial context, we leverage a Search Engine to attain the dynamic quality that we are seeking. Determining the right set of keywords (with help of social ontologies such as ODP) and executing the search engine based on

those keywords we can hope to get all or almost all the pages that will include pages we seek – but we cannot hope that returned pages comprise only those we seek. Thus even though the right set of keywords can be found that will result in high "recall," we can still be missing on high level of precision. To achieve high precision as well, we need to add a missing step that analyses pages returned from the search engine and deletes pages that cannot be relevant. How do we conduct this pruning step? This step needs to be conducted in a way that is efficient since the set of returned documents (from the search engine) can be very large and each returned document can itself be very large. Here meta-data needs to be used to the fullest extent. The actual analysis technique used could include Self-Organizing Maps or post-processing by ODP or WordNet, recursively invoking the prototype component in similar way that initial context pages are processed by ODP and/or WordNet. SOM, ODP and WordNet components are included in the prototype.

We follow a design science research methodology [67] and [68] for our work. Building research artifacts, evaluating feasibility and effectiveness, and abstracting knowledge gained in terms of design principles and theories [69] are among important research activities in design science research [70]. This methodology has guided us in iteratively testing, evaluating and revising the prototype. Development of a model that abstracts the World Wide Web and implementation of datasets to represent this model are the most recent additions to the workbench artifact.

The proposed research aim is to have broad impact by focusing real world scenario as noted in the introduction. Recognizing the value of collaborative, interdisciplinary research teams, our VP for Research has issued a challenge to our research faculty to identify research collaboration opportunities that align top faculty with complementary strengths around our university strategic research initiatives (such as "molecular basis of disease" or "economic risk analysis") and so become more competitively successful in research funding. Pragmatically, we look toward summative evaluation in field trials at our university where the workbench artifact assists researchers in 1) identifying teams of research collaborators; and 2) identifying funding opportunities relevant to their combined research interests.

**Acknowledgments.** We appreciate comments of the three anonymous reviewers.

## References

1. Etzioni, O.: The World Wide Web: Quagmire or Gold Mine. Communications of the ACM, 65–68 (1996)
2. Kobayashi, M., Takeda, K.: Information Retrieval on the Web. ACM Computing Surveys 32(2), 144–173 (2000)
3. Dhyani, D., Ng, W.K., Bhowmick, S.R.: A Survey of Web Metrics. ACM Computing Surveys 34(4), 469–503 (2002)
4. Mikroyannidis, A., Theodoulidis, B.: Heraclitus: A Framework for Semantic Web Adaptation. IEEE Internet Computing, 45–52 (2007)
5. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertext Web Search Engine. In: Proc. 7th WWW Conference (1998)
6. Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.M.: Self-Organization and Identification of Web Communities. IEEE Computer 35(3), 66–71 (2002)
7. Chen, H.: Web Retrieval and Mining. Decision Support Systems (35), 1–5 (2003)

8. Thelwall, M.: Scientific Web Intelligence: Finding Relationships in University Webs. Communications of the ACM, 93–96 (2005)
9. Clusty, `http://clusty.com/` (last accessed December 16, 2009)
10. Gardner, D., Shepherd, G.M.: A gateway to the Future of Neuroinformatics. Neuroinformatics 2(3), 271–274 (2004)
11. Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.W.: Learning to Cluster Web Search Results. In: Proc. 27th Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 210–217. ACM Press, Sheffield (2004)
12. Open-Directory-Project, `http://dmoz.org/` (last accessed December 16, 2009)
13. Gauch, S., Chafee, J., Pretschner, A.: Ontology-based personalized search and browsing. Web Intelligence and Agent Systems 1(3-4), 219–234 (2004)
14. Ferragina, P., Gulli, A.: A Personalized Search Engine Based on Web-snippet Hierarchical Clustering. In: Proc. 14th International World Wide Web Conference, Chiba, Japan, pp. 801–810 (2005)
15. Das, A., Datar, M., Garg, A., Rajaram, S.: Google News Personalization: Scalable online collaborative filtering. In: Proc. WWW 2007 / Track: Industrial Practice and Experience (2007)
16. Google News, `http://news.google.com/` (last accessed December 16, 2009)
17. NETFLIX, `http://www.netflix.com/` (last accessed December 16, 2009)
18. Amazon.com, `http://www.amazon.com/` (last accessed December 16, 2009)
19. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. The MIT Press, Cambridge (1998)
20. Beltrame, F., Koslow, S.: Neuroinformatics as a megascience issue. IEEE Trans Info. Tech. Bio. Med. 3(3), 239–240 (1999)
21. Koslow, S.H., Hirsch, M.D.: Celebrating a Decade of Neuroscience Databases: Looking to the Future of High-Throughput Data Analysis, Data Integration, and Discovery Neuroscience. Neuroinformatics 2(3), 267–270 (2004)
22. NIH-Roadmap, NIH Roadmap for Medical Research, `http://nihroadmap.nih.gov/` (last accessed December 16, 2009)
23. Parastatidis, S., Viegas, E., Hey, T.: VIEWPOINT A 'Smart' Cyberinfrastructure for Research. A view of semantic computing and its role in research. Communications of the ACM 52(12), 33–37 (2009)
24. Zhu, D., Dreher, H.: Determining and Satisfying Search Users Real Needs via Socially Constructed Search Concept Classification. In: Proc. Inaugural IEEE International Conference on Digital Ecosystems and Technologies, IEEE DEST 2007 (2007)
25. Zhu, D., Dreher, H.: Improving web search by categorization, clustering and personalization. In: Tang, C., Ling, C.X., Zhou, X., Cercone, N.J., Li, X. (eds.) Advanced data mining and applications, pp. 659–666. Springer, Berlin (2008)
26. Wang, X., Zhai, C.: Learn from Web Search Logs to Organize Search Results. In: Proc. 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), pp. 87–94 (2007)
27. Zakos, J., Verma, B.: A Novel Context Matching Based Technique for Web Document Retrieval. In: Proc. Eighth Intl. Conf. on Document Analysis and Recognition (2005)
28. Stefanidis, K., Pitoura, E., Vassiliadis, P.: On Relaxing Contextual Preference Queries. In: Proc. International Conference on Mobile Data Management (2007)
29. Li, Y., Wang, Y., Huang, X.: A Relation-Based Search Engine in Semantic Web. IEEE Transactions on Knowledge and Data Engineering 19(2), 273–282 (2007)
30. Chen, T.: Integrating Unstructured Data Using Property Precedence, Master's Thesis, Memorial University of Newfoundland (August 2008)

31. Harpale, A.S., Yang, Y.: Personalized Active Learning for Collaborative Filtering. In: Proc. SIGIR 2008, Singapore, July 20–24 (2008)
32. Wang, X., McCallum, A., Wei, X.: Topical N-grams: Phrase and Topic Discovery, with an Application to Information Retrieval. In: Proc. Seventh IEEE International Conference on Data Mining (2007)
33. Farquhar, A., Fikes, R., Pratt, W., Rice, J.: Collaborative Ontology Construction for Information Integration. Technical Report, Stanford University (1995)
34. Farquhar, A., Fikes, R., Rice, J.: The ontolingua server: A tool for collaborative ontology construction. Intl. Journal of Human-Computers Studies 46(6), 707–727 (1997)
35. Uschold, M., Grüninger, M.: ONTOLOGIES: Principles, Methods and Applications. Knowledge Engineering review 11(2), 93–155 (1996)
36. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. Knowledge Engineering Review, 31–89 (1998)
37. Navigli, R., Velardi, P., Gangemi, A.: Ontology Learning and Its Application to Automated Terminology Translation. IEEE Intelligent Systems 18(1), 22–31 (2003)
38. Navigli, R., Velardi, P.: Learning domain ontologies from document warehouses and dedicated web sites. Computational Linguistics 30(2), 151–179 (2004)
39. Contractor, N.S., Monge, P.R.: Managing knowledge networks. Management Communication Quarterly 16(2), 249–258 (2002)
40. Mathes, A.: Folksonomies - Cooperative Classification and Communication through Shared Metadata. University of Illinois Urbana-Champaign (December 2004)
41. Wang, J.Z., Taylor, W.: Concept Forest: A New Ontology-assisted Text Document Similarity Measurement Method. In: Proc. IEEE/WIC/ACM International Conference on Web Intelligence (2007)
42. Wei, X., Croft, W.B.: Investigating Retrieval Performance with Manually-Built Topic Models. In: Conference RIAO 2007, Pittsburgh PA, U.S.A., May 30-June 1 (2007)
43. Vaishnavi, V., Kuechler, W.: Universal: Enterprise Integration: The Challenges of and Approaches to Web-Enabled Virtual Organizations. Information Technology and Management 6(1), 5–16 (2005)
44. Morariu, D.S., Vintan, L.N., Tresp, V.: Meta-Classification using SVM Classifiers for Text Documents. Intl. Jrnl. of Applied Mathematics and Computer Sciences 1(1) (2005)
45. Liu, T.-Y., Yang, Y., Wan, H., Zeng, H.J., Chen, Z., Ma, W.-Y.: Support Vector Machines Classification with Very Large Scale Taxonomy. SIGKDD Explorations, Special Issue on Text Mining and Natural Language Processing 7(1), 36–43 (2005)
46. Morariu, D.S., Vintan, L.N., Tresp, V.: Evolutionary Feature Selection for Text Documents using the SVM. In: Proc. World Academy of Science, Engineering and Technology, October 2006, vol. 15 (2006)
47. Hearst, M.A.: Clustering versus faceted categories for information exploration. Communications of the ACM 49(4), 59–61 (2006)
48. Pratt, W., Hearst, M.A., Fagen, L.M.: A Knowledge-based approach to Organizing retrieved documents. J. Amer. Assn. for Artificial Intelligence, 480–484 (1999)
49. Blake, C., Pratt, W.: Collaborative information synthesis I: A model of information behaviors of scientists in medicine and public health. Journal of the American Society for Information Science and Technology 57(13), 1740–1749 (2006)
50. Blake, C., Pratt, W.: Collaborative information synthesis II: recommendations for information systems to support synthesis activities. Journal of the American Society for Information Science and Technology 57(14), 1888–1895 (2006)
51. Zhang, J., Ghahramani, Z., Yang, Y.: A Probabilistic Model for Online Document Clustering with Application to Novelty Detection. In: Proc. NIPS 2004, Vancouver, CA (2004)

52. Hung, C., Wermter, S.: A Time-Based Self-Organizing Model for Document Clustering. In: Proc. International Joint Conference on Neural Networks, Budapest, Hungary, July 2004, pp. 17–23 (2004)
53. Bakus, J., Hussin, M.F., Kamel, M.: A SOM based document clustering using phrases. In: Proc. 9th International Conference on Neural information Processing (ICONIP‘OZ), vol. 5 (2002)
54. Vandenberg, A., Vaishnavi, V.K., Li, L., Pandya, M.: LAIDBACK: A Workbench for User-Driven Dynamic Web Mining. In: Proc. 2nd International Conference on Design Science Research in Information Systems and Technology (DESRIST), Pasadena, CA (2007)
55. Amine, A., Elberrichi, Z., Bellatreche, L., Simonet, M., Malki, M.: Concept-based clustering of textual documents using SOM. In: Proc. IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2008 (2008)
56. Bendersky, M., Kurland, O.: Re-ranking search results using document-passage graphs. In: Proc. SIGIR, pp. 853–854 (2008)
57. Zhao, H., Ram, S.: Clustering Schema Elements for Semantic Integration of Heterogeneous Data Sources. Journal of Database Management 15(4), 88–106 (2004)
58. Hripcsak, G., Rothschild, A.S.: Agreement, the F-Measure, and Reliability in Information Retrieval. Journal of the American Medical Information Association 12(3), 296–298 (2005)
59. Sakai, T., Kando, N.: On information retrieval metrics designed for evaluation with incomplete relevance assessments. Information Retrieval 11, 447–470 (2008)
60. Tremblay, M.C., Hevner, A.R., Berndt, D.J.: The Use of Focus Groups in Design Science Research. In: Vaishnavi, V., Baskerville, R. (eds.) Proc. Third International Conference on Design Science Research in Information Systems and Technology, Atlanta, Georgia (2008)
61. Adams, F., Aizawa, K.: Why the Mind is Still in the Head: Challenges to Active Externalism. In: Robbins, P., Aydede, M. (eds.) Cambridge Handbook of Situated Cognition, pp. 78–95. Cambridge University Press, Cambridge (2008)
62. Storey, V.C., Burton-Jones, A., Sugumaran, V., Purao, S.: CONQUER: A Methodology for Context-Aware Query Processing on the World Wide Web. Information Systems Research 19(1), 3–25 (2008)
63. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Berlin (2001)
64. Van Rijsbergen, C.: Information Retrieval. Butterworth, London (1979)
65. Larsen, B., Aone, A.: Fast and Effective Text Mining Using Linear-time Document Clustering. In: Proc. Fifth ACM SIGKDD Int’l. Conference on Knowledge Discovery and Data Mining, pp. 16–22 (1999)
66. Stein, B., Eissen, S.M.Z.: Document Categorization with Major CLUST. In: Proc. 12th Annual WITS (WITS 2002), Barcelona, Spain (2002)
67. Montgomery, D.C., Runger, G.C.: Applied Statistics and Probability for Engineers. John Wiley & Sons, Inc., Chichester (1999)
68. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
69. 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)
70. Vaishnavi, V.K., Kuechler, W.L.: Design Science Research Methods and Patterns: Improving and Innovating Information and Communication Technology. Auerbach Publications, Taylor & Francis Group, New York (2008)

# Designing Business-Intelligence Tools with Value-Driven Recommendations

Adir Even[1], Yoav Kolodner[1], and Roy Varshavsky[2]

[1] Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev, P.O. B. 653, Beer-Sheva, Israel
{adireven,kolodner}@bgu.ac.il
[2] Israel Innovation Labs, Microsoft Israel R&D Center, 13 Shenkar st. Herzelia, 46733, Israel
royv@microsoft.com

**Abstract.** Business-intelligence (BI) tools are broadly adopted today, supporting activities such as data analysis, decision making, and performance measurement. This study investigates a new approach for designing BI tools – the integration of feedback and recommendation mechanisms (FRM), defined as embedded visual cues that provide the end-user with usage and navigation guidelines. The study focuses on FRM that are based on assessment of previous usage, and introduce the concept of value-driven usage metadata - a novel methodology for linking the use of data resources to the value gained. A laboratory experiment, which tested the design of FR-enhanced BI with 200 participants, confirmed that FRM integration will improve the usability of BI tools and increase the benefits that can be gained from using data resources. Further, the experiment highlighted the potential benefits of collecting value-driven usage metadata and using it for generating usage recommendations.

**Keywords:** Business Intelligence, Decision Support Systems, Recommender Systems, Data Warehouse, Metadata.

## 1 Introduction

Data repositories, along with the information systems (IS) utilizing them, had long been recognized as a critical resource. Recent years have witnessed a major growth in the use of data resources for business analysis, performance measurement, and managerial decision support. Firms may gain strong competitive advantage by investing in the development of data analysis capabilities and data-driven analytics [6]. This trend is well-supported by the rapid progress in the capacity and the performance of information and communication technologies (ICT) for managing and utilizing large data resources. Most notable is the broad adoption of business intelligence (BI) platforms and tools, which permit rapid development and distribution of data analysis and decision support utilities.

High complexity is a major limitation in current BI environments. The common end-user, in search of an answer to a business question, often finds large data repositories too difficult to navigate for reaching the right data, and BI tools too difficult to

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 286–301, 2010.

use for answering the question. Furthermore, it is even not uncommon for end-user to know neither the right business question to ask, nor the full range of capabilities offered by data repositories and BI tools. This study investigates the integration of feedback and recommendation mechanisms (FRM) into BI tools. We define FRM as visual cues that guide the end-user to consider using certain data subsets and/or analysis forms. We suggest that the FRM integration can improve the usability of BI tools and increase the benefits that end-users and organizations can gain from data resources, this by facilitating effective and efficient navigation, and by helping to reveal undiscovered potential of unused data and analysis forms, and thus add business value. The experiment described later investigates value-driven FRM – a novel form of a recommender system, based on quantitative assessments of business-value gains and their attribution to the data resources being used.

Our study makes a few contributions. First, it presents the concept of integrating FRM into BI tools and highlights a few possible approaches for generating them. Second, it proposes a novel methodology for tracking the use of data resources, termed as value-driven usage metadata, which integrates in assessments of both the frequency of use and the associated value gains toward generating FRM. Finally, it explores the potential contribution of collecting value-driven metadata and generating value-based FRM through a comprehensive laboratory experiment, in which participants were asked to evaluate different variants of FRM, integrated into a BI tool. An in-depth analysis of the experiment's results supports the assumption that value-based FRM may significantly improve the effectiveness of using BI tools and provides some additional important insights. In the reminder of this paper, we first provide the background to our work, and introduce the two novel concepts that underlie it – the integration of FRM into BI tools, and the collection of value-driven usage metadata. We then describe in detail the laboratory experiment, present the results and discuss the findings. To conclude, we highlight the potential contributions of the new concepts that we present and discuss directions for future research.

## 2 Background

Our study aims at improving usability and effectiveness of BI systems, which use a data warehouse (DW) as an infrastructure. The software market offers a plethora of commercial platforms for supporting BI activities, which typically offer a variety of presentation capabilities (e.g., tables, charts, statistics, and advanced analytics), rapid-development utilities, and administrative tools. BI platforms permit different forms of data usage such as reports, spreadsheets, OLAP (On-Line Analytical Processing), digital dashboards, and data mining. This variety of presentation and analysis forms confers the flexibility to use the same data resource for supporting different analytic tasks and to adapt the presentation style to end-users' capabilities and skills.

The increasing popularity of DW/BI environments can be attributed to benefits such as gaining broad business coverage, leveraging data-collection investments, and shortening implementation cycles ([8], [13], [19]). Implementing DW/BI environments is challenging both technically, due to the many components and the high complexity of configuration decisions involved [8], and organizationally, due to the substantial managerial support and financial resources needed [18]. Moreover, DW/BI design and configuration decisions are often associated with substantial cost-benefit tradeoffs [8]. Despite the increasing popularity of DW and BI, so far these concepts have attracted only limited academic research aimed at the challenge of increasing the effectiveness of DW/BI utilization.

### 2.1 Feedback and Recommendation Mechanisms (FRM)

As a contribution to improving the effectiveness of DW/BI, we propose to integrate FRM capabilities in a manner that would help the end-user navigate through complex data resources and highlight usage directions with a high benefit potential, while still maintaining simple and easy-to-learn functionality. FRM, in the form visual cues, would provide the end-user with feedback on the analysis done so far, and some guidelines for further actions to consider – e.g., approach certain data subsets and/or apply certain analysis forms. FRM can be seen as a form of a recommender system – an automated mechanism which provides end-users with some rating of items not seen so far [1]. Recommender systems aim at improving usability and decision-making outcomes, enhancing the end-user's experience, and reducing information overload ([16], [17]). Such systems are a common practice today in commercial and social websites [2], and in information retrieval systems such as digital libraries [15]. In [1] two categories of methods for generating recommendations are identified – content-based methods, driven be the choices made by the user in the past, and collaborative-filtering methods, driven by the choices made by other users with similar preferences. They also suggest that some methods introduce a hybrid between these two approaches – the recommendation form that we test in our experiment can be seen as such a hybrid.

The integration of recommender systems into BI tools is not a common practice today, and has not been significantly explored so far, although some commercial software vendors have introduced recommendation utilities to an extent (e.g., Bissantz, in their data mining tool, http://www.bissantz.com/deltamaster). We would argue that the motivations that drive the integration of recommender systems into websites and digital libraries - improving decision-making outcomes, enhancing end-user's experience, and reducing information overload – apply in BI environments as well. We suggest that similar enhancement to BI systems design may have important contribution to better usage of BI tools, and improve the decisions made.

Fig. 1 offers a simplified illustration of integrating FRM capabilities into a BI tool. The illustrated tool lets the end-user navigate through sales data, and slice it along certain customer characteristics (e.g., Income, Occupation, and Gender), towards identifying profitable customer segments. The FRM-enhanced version of the BI tool provides color-coded numeric rating of the different attributes, and of the data values in each attributes. These rates provide certain recommendations to the user on how to slice the data further at any given point of time (the rating method used in our experiment will be described in detail later). In this illustrative example – a high characteristic rate would recommend the user to slice the data along that characteristic (here, "Education"), indicating that the underlying data values differentiate well between customers with high profit potential, versus less-profitable customers. Similarly, a high data-value rate (here, the age group of "30-60") would identify a segment which is more likely to be profitable and worth further investigation. Notably, navigation decisions in this tool are left to end-users; however, they are now provided with visual cues on how to navigate more effectively.

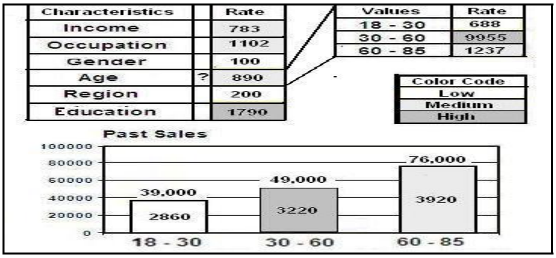



**Fig. 1.** A Business Intelligence Tool Enhanced with FRM

Obviously, there are other possible forms for visualizing FRM besides color-coding (e.g., textual/graphical pop-up messages, "mouse-over" tool-tips, and side bars). Such FRM forms could indicate, in addition to the actual recommendations, the level of confidence and relevance of each recommendation based on the parameters that construct it. We discuss other possible FRM forms in our concluding section.

## 2.2 Value-Driven Usage Metadata

Usage has long been identified as an important factor in explaining IS success. The actual usage of IS strongly explains its success and payoff– even more than the level of ICT-investments made [7]. A few conceptualizations of IS usage in past research are reviewed in [5] – e.g., based on the extent, the nature, and the frequency of use. In this study, we contrast frequency-based with value-based, measurement of use.

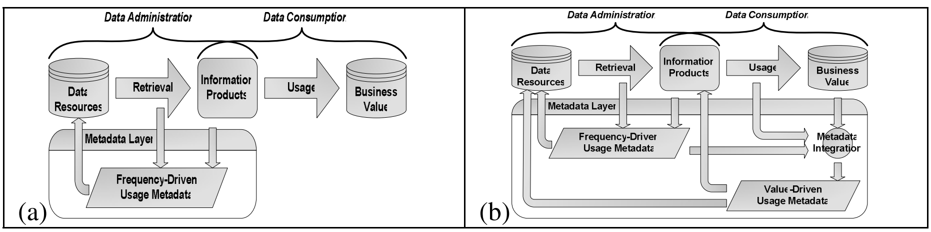



**Fig. 2.** (a) Frequency-Driven versus (b) Value-Driven Usage Metadata

We particularly observe usage in DW/BI environments. These are often described as a complex manufacturing process, consisting of interconnected acquisition, processing, storage, retrieval and usage stages [4]. This manufacturing process can be conceptualized as having two high-level stages – data administration versus data consumption (Fig. 2) - each associated with different stakeholders, goals, motivations and tasks. Data administration addresses technical aspects – providing the ICT capacity needed to store and process data, and the tools for implementing information-product outcomes (e.g., datasets, reports, and analyses). Data consumption, on the other hand, would seek to transform data resources and information products into business value, through their effective usage, less concerning about the technical aspects associated with managing them.

Tracking the usage of data subsets (e.g., tables, attributes, and records) and applications in DW/BI environments has been identified as an important form of metadata [14]. Usage-tracking utilities are offered by some specialized commercial solutions and, to an extent, by database management and BI platforms. We term the common approach implemented by today's solutions as frequency-driven usage metadata (Fig. 2a). This approach is based on tracking data-retrieval requests and identifying the data subsets being most-frequently used (e.g., by parsing the underlying SQL statements). Frequency-driven metadata collection may provide important inputs to the data administrator, toward improving system design and prioritizing data and system administration efforts. It is common in databases that some records and attributes are accessed more frequently than others. The assumption that drives this approach is that frequent usage reflects higher importance. Accordingly, the results of frequency-driven usage tracking may lead to important data management decisions such as giving frequently-used data subsets higher priority in terms of data-quality improvement – i.e., watch these data subsets closely, detect and correct defects, and make sure to keep them up-to-date.

While seeing the merits of collecting frequency-driven usage metadata, we question - would it truly benefit data consumers? Frequent usage may reflects higher significance of certain data subsets; hence, to an extent, higher value-contribution potential. However, one could argue that frequent usage reflects certain stagnation and a tendency to "dig into the same well" – i.e., re-using certain data subsets repetitively, while possibly ignoring unused but important subsets. A potential risk with basing decisions solely on frequency-driven metadata is a possible loss of opportunity to benefit from data subsets that consumers have neglected to use so far. There is possibly no "clear cut" answer to this question, as it largely depends on business context, the timing, and the nature of usage tasks. However, we suggest that important insights can be gained from tracking not only the frequency of data usage, but also the decisions made based on data retrieval and the associated value gains. We term this novel approach value-driven usage metadata (Fig. 2b).

The notion underlying this approach is that the purpose of using data and information resources is generating value. The benefits gained from the use of information resources have been conceptualized as utility [3], which can be measured in terms of revealed (objective, measurable) value, perceived value, and/or willingness to pay. Capturing value measurements (e.g., production increase, sales activity, revenues and costs) and storing them in dedicated data repositories is a common practice in organizations today. However, such measurements are rarely linked to the data resources and the decision-support tools that were used in the process of value generation. Utility assessments have been used to optimize the configuration of data processes [4], DW datasets [9], and data-quality improvement policies [10]. We suggest that, beyond the benefits offered to data administration, collecting quantitative value assessment as a form of metadata can improve data consumption as well. The novel usage-tracking approach promoted in this study (Fig. 2b), extends the current common approach (Fig. 2a), by combining measurements of both the frequency of use and of the associated business-value assessments. In certain cases, value assessments can be based on data that exists within the same data resource (e.g., sale transactions, linked to marketing campaigns that were based on analysis of previous sales). In other cases – such assessments may reside in other information resources such as CRM and accounting systems. The integration is done by observing data at the decision-task level – observing the value gained by decision task outcomes and linking it to the queries that have supported each task.

Once the link between decision tasks and queries is established, different methods can be used for attributing value to specific data subsets. We describe here a relatively simple method, which assumes that value is attributed to the last in a sequence of queries that support a decision task. We assume that to support a certain decision, users query repetitively a certain tabular dataset with N records indexed by [n] and M attributes indexed by [m]. We consider Q queries indexed by [q], each associated with a business value $V^q$. The binary indicator Rqn indicates whether record [n] was retrieved by query [q] ($R^q_n=1$), or not ($R^q_n=0$). Similarly, $R^q_m$ indicates whether attribute [m] participated in query [q] or not. The value of a certain query ($V^q$) is attributed between the participating data items, using a certain value-attribution function $V^q_{n,m}=u(V^q, R^q_n, R^q_m)$, such that $V^q=\Sigma_n\Sigma_m V^q_{n,m}$. For simplification, we use here an equal attribution of value among all participating data items. Accordingly, the overall value attributed to a certain data items $V_{n,m}$ is given by:

$$V_{n,m} = \sum_{q=1..Q} V^q_{n,m} = \sum_{q=1..Q} u\left(V^q, R^q_n, R^q_m\right) = \sum_{q=1..Q} R^q_n R^q_m V^q \Big/ \left(\sum_{n=1..N} \sum_{m=1..M} R^q_n R^q_m\right) \quad (1)$$

Where,

| | |
|---|---|
| Q - | The number of queries performed (indexed by [q]) |
| M, N - | The number of attributes (indexed [m]) and records (indexed [n]), respectively |
| $V^q$, $V^q_{n,m}$, u - | Query [q] value, its attribution to data item [n,m], and the attribution function used, respectively |
| $R^q_n$, $R^q_m$ - | Binary indicators of the participation (=1) of record [n] and attribute [m] in query [q], respectively |

The experiment described next tested FRM driven by usage-tracking (both frequency-driven and value-driven). As a preliminary step, we have successfully implemented a working prototype of a metadata-layer module, which would permit a front-end BI tool to access the usage-tracking scores on demand through function calls. These scores can be integrated into front-end tools, enhance the visual presentation, and communicate important information to both data consumers and administrators. A key challenge with the value-driven metadata collection approach is the fact that most data environments today are not designed to establish an explicit link between decision outcomes and the underlying data and queries. An implicit link can be possibly created through inference mechanisms - e.g., by comparing user identifiers and time-stamps (e.g., in [10]). We acknowledge this challenge as a major limitation that would require more research.

## 3 Laboratory Experiment

Our study follows the design-research paradigm, which targets the creation of new artifacts toward improving IS implementation and use [12]. The success of design-science research is judged by the quality, the contribution, and the impact of the developed artifacts [11]. To back our arguments about the contribution potential of the new artifacts presented (FRM integration into BI tools, and value-driven usage tracking), we have performed a laboratory experiment for studying and assessing them further in a controlled environment. The experiment was guided by the following questions:

**Does FRM integration improve decision outcomes?** We assume that FRM integration will significantly improve decision-making outcomes; however, repetitive use is other important factor that may explain such improvement. As end-users repetitively use a decision-support tool, and get more familiar with the underlying data and the decision task – it is reasonable to assume that some improvement to the decision outcome will be observed, regardless FRM integration. However, our assumption is that decision-outcome improvement with FRM integration will be above and beyond the improvement gained with repetitive use alone.

**Does FRM integration affect usage style?** We assume that FRM integration will significantly change the way people use a BI tool, in terms of making data navigation more focused. Again, we assume here that repetitive usage will play an important role in changing usage style, but that the changes detected with FRM integration will be significantly greater, above and beyond the changes caused by repetitive usage alone.

**With respect to the questions above, would different forms of FRM lead to significantly different results?** When discussing frequency-driven versus value-driven collection of usage metadata earlier, we have suggested that the latter form is likely to be superior to the former. In our experiment, we explore whether FRM generated by value-driven usage tracking will indeed outperform frequency-driven. Further, we explore whether certain forms of value assessment and attribution will significantly outperform others.

### 3.1 Experiment Settings

Our experiment simulated a marketing decision task – given a list of customers, choose those customer segments that will be targeted in a promotion campaign that offers a certain product. To aid this task, users were provided with a BI tool, similar to the one shown in Fig. 1, and a simulated dataset of past sales, which could be analyzed by the tool (further described in Appendix A). The experiment was conducted with 200 participants, all undergraduate engineering students around the same age, with a majority of them majoring in IS engineering in their 3rd year of study or later. Some participants have indicated previous exposure to marketing tasks and/or BI applications; hence, we have controlled for these effects as well. All sessions were conducted in labs with similar room conditions, each with identical personal computers, and were all scheduled to similar hours. All participants received some course credit. Additional cash prizes were offered to 5 participants picked by a raffle, in which the chance of winning was correlated to the decision outcome.

Each participant was asked to attend two one-hour sessions. In the beginning of each session, the participants were given a scripted description of the task and instructions on using the tool. After this introduction, the participants were asked to perform the same decision task 6 times repetitively, given a maximum of 5 minutes per repetition, after which the performance measures per repetition (units sold, costs, and net-benefit) were recorded. We have recorded a few measures per repetition that reflect users' interaction - the time spent per task, the number of customer segments chosen and the number of mouse-clicks made. Upon completing 6 repetitions, the participants were asked to estimate their own performance, using a short survey.

In the first session, all participants were provided with identical BI tools that have no FRM. This session served a few purposes – first, to create a baseline for assessing the impact of FRM usage. Second, to familiarize the participants with the task and with the BI tool, and third, to collect usage metadata – as soon as each task repetition was completed, the decision value (the net-benefit), and the segments selection were passed to a usage-tracking metadata module. Using these inputs, the module calculated how frequently each data item in the customer dataset was used, and the value attribution among data item, using the attribution method described earlier.

For the second session, participants were divided randomly into 5 groups, each provided with a different FRM variant:

1. **No FRM** – participants in this control group received the exact same version of the BI tool in both sessions.
2. **Frequency-Driven FRM** – the FRM in this version were based on the frequency of usage. The rate per characteristic value reflected the number of times this value was used for defining a customer segment.
3. **Value-Driven FRM** – here, the FRM reflects the relative value contribution, based on the net-benefit achieved in each decision task. The rate reflects the cumulative value attributed to each characteristic category.
4. **Expert FRM** – this version is similar to the value-driven FRM, but here the value attribution is based only on the best 20% scores achieved.
5. **Subjective FRM** – here, we used the users' perceived performance in the first session as a value proxy. Unlike the 3 other FRM form that rely on objective value, this kind of FRM rely on users' subjective assessment.

We will note an important difference in the scoring of the latter group - while the objective value scores were collected one per repetition, the subjective score was collected only once per session; hence, we have attributed the same score to all the 6 tasks within a session – what biases the results of this group to an extent.

**Table 1.** Experiment Groups

| Group | Participants (M/F) | Age μ/σ | IS major | English Fluency | Native | Mktg. Exp. | BI Exp. |
|---|---|---|---|---|---|---|---|
| 1. None | 39 (20/19) | 26.3/1.8 | 95% | 100% | 95% | 21% | 15% |
| 2. Frequency | 42 (21/21) | 26.2/1.2 | 93% | 98% | 88% | 17% | 17% |
| 3. Value | 40 (15/25) | 25.8/1.6 | 95% | 100% | 88% | 10% | 20% |
| 4. Expert | 40 (18/22) | 25.9/1.8 | 90% | 97% | 93% | 10% | 10% |
| 5. Subjective | 39 (18/21) | 26.1/1.7 | 90% | 94% | 92% | 10% | 10% |
| **Overall** | 200 (92/108) | 26.0/1.7 | 93% | 98% | 91% | 14% | 86% |

(*) μ: Avg. , σ: STDEV, Exp. – Experience, Mktg. - Marketing

The groups and the associated demographics are summarized in Table 1. All participants completed the assignments successfully and no data collection issues could be detected. To ensure readiness, the experimental task, tools, and procedures were all tested in a pilot study done with 6 research students.

From the collected data, we have measured a few objective performance-based variables (The numbers in Table 2 reflect an average among 6 decision tasks per session). The variables where Decision Value, Time, Segments, and Clicks – the average net-benefit, time (in seconds) spent on a decision, number of customer-segments chosen, and number of mouse-clicks made, respectively. In addition we have measured the perceived performance at the end of each session.

**Table 2.** Summary Statistics

| Variable | S1: μ | S1: σ | S2: μ | S2: σ |
|---|---|---|---|---|
| **Value** | 30.90 | 498.00 | 574.52 | 593.11 |
| **Time** | 163.74 | 90.02 | 156.24 | 94.85 |
| **Segments** | 2.47 | 1.43 | 2.30 | 1.32 |
| **Clicks** | 82.37 | 39.76 | 46.75 | 24.89 |
| **Performance** | 4.88 | 1.12 | 5.19 | 1.14 |

(*) S1/2: Session 1 or 2, μ: Avg. , σ: STDEV

Testing these variables against the control variables (Gender, Age, Major, English Fluency, Country of Birth, Exposure to Marketing Task, Exposure to BI Tools), has shown no significant effect.

### 3.2 Results

To assess this question, we have used the "Decision Value" variable. Our base assumption that repetitive use would improve decision outcomes was indeed supported: considering the control group alone, which received the same BI tool in both sessions (group 1, 29 participants), the average decision value increased from 93.7 in the first session to 381.2 in the second. An ANOVA test shows that the increase was indeed significant (F-value: 5.90, P-Value: 0.017).

**Table 3.** Decision Value Analysis

| Group | S1: μ | S1: σ | S2: μ | S2: σ | Δ: μ | Δ: σ | Δ: F | Δ: P.V. |
|---|---|---|---|---|---|---|---|---|
| **1. None** | 93.72 | 504.19 | 381.20 | 540.02 | 287.48 | 511.02 | 5.90 | 0.02 |
| **2. Freq.** | 36.98 | 508.13 | 359.63 | 560.24 | 322.65 | 600.88 | 6.76 | 0.01 |
| **3. Value** | 28.19 | 408.95 | 968.13 | 372.66 | 939.94 | 544.22 | 107.97 | ~0 |
| **4. Expert** | -46.35 | 526.68 | 987.69 | 370.71 | 1034.04 | 561.46 | 99.13 | ~0 |
| **5. Subj.** | 43.55 | 546.87 | 171.79 | 574.20 | 128.24 | 577.89 | 0.68 | 0.41 |
| **Overall** | 30.90 | 497.99 | 574.52 | 593.1 | 543.62 | 666.89 | 98.54 | ~0.00 |

(*) S1/2: Session 1 or 2, μ: Avg., σ: STDEV, Δ: Difference, F: F-Value, P.V.: P-Value

When all 5 groups considered (Table 3) – the value increase between sessions is even more significant (F-Value: 98.54, P-Value: ~0). As expected, with respect to first-session performance there is no significant difference between groups (F-Value: 0.399, P-Value: 0.809). However, the difference in second-session performance is significant (F-Value: 23.47, P-Value: ~0). With group 3 (Value) and 4 (Expert) – the increase was sharp and highly significant, where the latter increase is slightly greater than the former. With group 2 (Frequency), the increase was significant, but only marginally higher than the increase gained by the control group. With group 5 (Subjective), the increase was insignificantly lower than the control-group's increase.

To assess the effect on usage style, we have used the "Time", "Segments", and "Clicks" variables. We assumed that, as users get more familiar with the decision tool through repetitive usage, they will tend to reach a decision faster, refine the decision by extending the number of segments chosen, and reduce navigation intensity – i.e., use the mouse less often to change data segmentation. Further, we assumed that FRM-inclusion will even increase these effects. The findings for "Time" and "Segments" did not support our assumption. The correlation between these variables was positive, high and significant (0.89 in the first session, 0.83 in the second). However, their scores did not show significant difference neither between the two sessions, nor among FRM groups within a session.

Conversely, the change in "Clicks" was apparent (Table 4). Considering the control group alone, the average clicks per task decreased significantly from 86.33 in the first session to 64.33 in the second (F-Value: 5.80, P-Value: 0.018). When considering all 5 groups, the decrease in clicks between sessions is even more significant (F-Value: 115.34, P-Value: ~0). As expected, there was no significant difference between groups in first-session clicks (F-Value: 1.92, P-Value: 0.109); however, the difference between groups in second-session clicks is significant (F-Value: 7.68, P-Value: ~0). Significant decrease in the number of clicks could be detected for each group that was provided with FRM, where all decreases are greater than the decrease in the control group. Interestingly, the number of clicks decreased significantly with no apparent correlation to the quality of the decision-outcome improvement offered by the FRM – for example, the decrease in group 2 was a lot greater than the decrease in group 4, although the decision-value improvement in group 4 was much higher (Table 3). This may imply that when FRM are provided, end-users tend to accept the recommendations and reduce navigation intensity, regardless their quality.

**Table 4.** Clicks Analysis

| Group | S1: μ | S1: σ | S2: μ | S2: σ | Δ: μ | Δ: σ | Δ: F | Δ: P.V. |
|---|---|---|---|---|---|---|---|---|
| 1. None | 86.33 | 47.83 | 64.23 | 31.53 | -22.10 | 5.94 | 5.80 | 0.02 |
| 2. Freq. | 90.58 | 39.56 | 39.31 | 16.20 | -51.27 | 5.35 | 60.41 | ~0 |
| 3. Value | 88.10 | 39.82 | 45.18 | 20.01 | -42.92 | 5.71 | 37.09 | ~0 |
| 4. Expert | 70.87 | 39.10 | 46.77 | 27.28 | -24.10 | 5.20 | 11.75 | ~0 |
| 5. Subj. | 75.44 | 33.09 | 38.84 | 18.65 | -36.60 | 4.74 | 36.23 | ~0 |
| **Overall** | 82.36 | 39.75 | 46.74 | 24.88 | -35.62 | 2.52 | 115.34 | ~0 |

(*) S1/2: Session 1 or 2, μ: Avg. , σ: STDEV, Δ: Difference, F: F-Value, P.V.: P-Value

To assess the extent to which interaction style affects decision value, we ran a regression of decision-value scores in session 2 versus time, segments, and clicks (Table 5). When lumping together the 5 groups, all 3 regressions are significant with positive slopes – meaning that the decision-value increases with the more time spent, with a larger number of segments chosen, and with higher navigation intensity. However, the adjusted R-Square for these regressions is small, meaning that the decision performance is explained only to a small extent by these factors.

**Table 5.** Value Regression against Time, Segments and Clicks (in Session 2)

| **Group** | **Time** | | | | **Segments** | | | | **Clicks** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **α** | **T** | **P.V.** | **A-R2** | **α** | **T** | **P.V.** | **A-R2** | **α** | **T** | **P.V.** | **A-R2** |
| 1. None | 2.01 | 2.73 | 0.01 | 0.15 | 127 | 3.02 | 0.01 | 0.17 | 3.38 | 1.22 | 0.23 | 0.01 |
| 2. Freq. | 0.37 | 0.37 | 0.71 | -0.02 | 84 | 1.33 | 0.19 | 0.02 | 9.72 | 1.84 | 0.07 | 0.06 |
| 3. Value | 1.59 | 2.03 | 0.05 | 0.07 | 217 | 4.07 | ~0 | 0.29 | 7.13 | 2.55 | 0.01 | 0.12 |
| 4. Expert | 1.16 | 1.82 | 0.07 | 0.06 | 140 | 3.48 | ~0 | 0.22 | 2.61 | 1.11 | 0.23 | 0.01 |
| 5. Subj. | 1.41 | 1.73 | 0.09 | 0.05 | 220 | 2.08 | 0.04 | 0.08 | 0.93 | 0.18 | 0.85 | -0.26 |
| **Overall** | 1.49 | 3.44 | ~0 | 0.05 | 149 | 4.95 | ~0 | 0.11 | 3.79 | 2.26 | 0.02 | 0.02 |

α: Slope, T: T-Test, P.V.: P-Value, A-R2: Adjusted R-Square

When treating each group individually, time seems to significant affect the decision value only for the control group. The other effects on decision outcome are less significant. The positive effect of clicks on the decision was significant only for group 3 (Value FRM). For most groups (with the exception of group 2), the number of segments significantly affected the decision value. This result not surprising, as maximizing the net-benefit in the given decision task would require selecting a large number of small segments. Interestingly, this optimal decision policy was not explained explicitly to the participants. However, participants who understood and adopted this policy, regardless the form of FRM provided – indeed performed better.

### 3.3 Discussion

Overall, our experiment was successful. The number of participants was relatively large, and the data collection procedures all worked well, what permitted getting a clean and complete dataset and results with high significance. The results provide some interesting and thought-provoking results.

A key assumption of our study was confirmed by the finding – certain forms of FRM-integration into a BI tool indeed improved decision outcomes significantly, above and beyond the improvement gained with repetitive use alone. Moreover, FRM-integration has not also affected the decision outcome, but also navigation intensity – based on the clicks-count, it appears to be that participants who received recommendations tended to use them, and navigate less. On one hand – this may point out a certain advantage of FRM-integration, in terms of promoting a more focused and efficient navigation. On the other hand, the results may reflect some risk – it appears that users tended to follow the recommendations even when they do not lead to any major improvement. This underscores the need for further investigation, that would detect which forms of FRM work better, and which should be avoided.

Another key assumption - that value-driven FRM will outperform frequency-driven FRM – was supported to a great extent. With members of groups 3 and 4, who received recommendations based on value-mapping, the increase in performance was high and significant for both variants. Interestingly, the improvement gained with “expert-based” recommendations was only marginally greater than the improvement gained by considering all scores. The small difference could be incidental, however it raises a question – would it be sufficient to create value-driven usage metadata for generating FRM generate based on a sample of tasks, rather than on the whole population? If yes, would be the optimal sampling policy? Answering this would require some more analytical and empirical investigation. The low magnitude of improvement gained by using frequency-driven FRM was somewhat surprising. Although we assumed value-driven FRM to outperform frequency-driven FRM, we still expected to see some improvement with the latter, beyond the improvement gained by repetitive use, and some more investigation will be required to detect whether these results are incidental. Notably, implementing frequency-based usage tracking is less demanding technically than implementing value-driven tracking.

A notable result is the low and insignificant improvement gained with FRM that were based on subjective assessment. The reason for this lower performance is not clear, and the use of subjective assessments for generating FRM indeed requires some more investigation. The different nature of this FRM can be possibly attributed in part to the averaging effect, caused by the different value-attribution method used, as underscored earlier. We would hasten to say that this result is particularly interesting, due to the fact that many common collaborative filtering recommender systems are based on the user's subjective assessments. This result may indicate that, in the BI context, be cautious about using subjective assessments to generate recommendations.

## 4 Conclusions

Our research investigated the design of FRM-enhanced BI tools. The experiment confirmed our assumption that integrating certain forms of FRM can improve decision-making outcomes. Another key contribution is the novel approach for data-usage tracking. This approach suggests that integrating quantitative assessments of usage-frequency together with the associated value gained would offer substantial benefits to both data administration and consumption. Joint frequency and value assessments can help identifying unused data subsets with high value-contribution potential and, consequently, motivate new usage forms. Complementing frequency assessments with value assessments may provide feedback based on usage performance, and reducing the potential risks. First, value allocation gives higher weight to past usages with high contributions. Second, it reflects variability in the importance of data subsets within a usage context. Lastly, it can help detecting data subsets with high contribution potential that have not been frequently used.

Obviously, more research will be needed to address some key limitations with our study. First is the issue of quantifying data value. As often stated in past research, quantifying the value of information resources is a challenging task. Organizations maintain performance measurements (e.g., productivity, income, and profitability) that can be possibly linked to decision tasks. However, decision performance may depend on resources other that data – e.g., human knowledge and financial assets. Further, the value depends on the data-usage context, and value assessment for a certain type of use does not necessarily apply to others. Further, value varies with time, as data that can be used effectively at a certain point of time, might become obsolete later. We hasten to say that the value-allocation methodology, which we apply in this study, appears to be a better fit to operational environments in which decision tasks have a high degree of repetition, and in which causal relations between data usage and business performance are easier to establish. Promotion-campaign management, such as in our illustrative example would be a good representative for this type of decision-making. Financial-investment decisions would be another example for data-driven decisions, in which outcomes are measurable (e.g., the change in the value of the financial asset) and linkable to the data resources being used. Conversely, quantifying the value of decision outcomes might turn out to be challenging in strategic decision scenarios, which are not repetitive in nature and often relay on information resources other than organizational data repositories.

Another issue that needs addressing is a method for linking value to specific queries. Performance assessments are rarely linked explicitly to the data resources and tools used. Our preliminary prototype includes inference mechanisms for creating implicit links – e.g., based on the user name, and/or time proximity. Obviously, implicit links cannot be absolutely precise and might bias the value allocation significantly. Establishing explicit links will require stronger metadata integration between systems and, likely, redesign of data environments (e.g., joint codes that link each decision task and queries). One could question whether or not making such a high investment in redesigning data environments and BI tools would justify the benefits gained. There is also a need to explore further how to attribute value to specific data objects, as the attribution method may have a critical impact on the results. In our experiment, we have attributed value only to the last query in the sequence that led to the decision, and distributes the value equally among all the data items involved. A different allocation method may consider, for example, allocating the usage value among all queries and/or consider unequal allocation.

Further extensions may also test the integration of FRM based on other recommender mechanisms. One possible approach would be to consider task and user characteristics - the same DW/BI environment can be used to support a plethora of business processes and tasks, each with different data usage needs. FRM capabilities can acknowledge such needs, by creating either task profiles that capture specific task characteristics or by asking expert users to identify certain data elements or analysis results that are more useful and relevant for a given task. FRM capabilities can also be driven by analysis of the data using algorithmic data-mining techniques. Data mining can suggest alternatives to decisions and actions that are about to be taken and allow users to re-consider them. A possible drawback of using statistical analysis and data mining to derive FRM is the risk that recommendations will be based solely on the data resource, without considering the context in which it is used.

Finally, future research should also address organizational and economic perspective of the proposed concepts. As organizations today do not possess the appropriate infrastructure for value-driven collection of usage metadata, adopting this approach may mean a substantial investment. One could ask – would the benefits of adopting value-driven usage tracking and using it for FRM integration justify the cost? Our experiment indicates that usage-driven FRM can indeed improve decision outcome; however, the results were obtained under a simulated and well-monitored environments; hence, cannot be generalized to real-world scenarios. Evaluating and testing value-driven metadata and FRM integration in real-world environments would therefore be an important follow-up step in furthering this line of research, toward gain more knowledge and insights about the challenges that firms may face when implementing these new concepts that we propose.

## References


1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Tr. on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Adomavicius, G., Kwon, Y.: New Recommendation Techniques for Multicriteria Rating Systems. IEEE Intelligent Systems 22(3), 48–55 (2007)
3. Ahituv, N.: A Systematic Approach towards Assessing the Value of Information System. MIS Quarterly 4(4), 61–75 (1980)
4. Ballou, D.P., Wang, R., Pazer, H., Tayi, G.K.: Modeling Information Manufacturing Systems to Determine Information Product Quality. Management Science 44(4), 462–484 (1998)
5. Burton-Jones, A., Straub Jr., D.W.: Reconceptualizing System Usage: An Approach and Empirical Test. Information Systems Research 17(3), 228–246 (2006)
6. Davenport, T.H.: Competing on Analytics. Harvard Business Review 84(11), 99–107 (2006)
7. Devaraj, S., Kohli, R.: Performance Impacts of Information Technology: Is Actual Usage the Missing Link? Management Science 49(3), 273–289 (2003)
8. Even, A., Shankaranarayanan, G., Berger, P.D.: The Data-Warehouse as a Dynamic Capability: Utility-Cost Foundations and Their Implications for Economics-Driven Design. In: Proceedings of the International Conference on Information Systems (ICIS), Milwaukee, WI, USA (December 2006)
9. Even, A., Shankaranarayanan, G., Berger, P.D.: Economics-Driven Design for Data Management: An Application to the Design of Tabular Datasets. IEEE Tr. on Knowledge and Data Engineering 19(6), 818–831 (2007)
10. Even, A., Shankaranarayanan, G.: Comparative Analysis of Data Quality and Utility Inequality Assessments. In: Proceedings of the European Conference on Information Systems (ECIS), Galway, Ireland (June 2008)
11. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 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. March, S.T., Hevner, A.R.: Integrated Decision Support Systems: A Data Warehousing Perspective. Decision Support Systems 43(3), 1031–1043 (2005)
14. Shankaranarayanan, G., Even, A.: The Metadata Enigma. Comm. of the ACM 49(2), 88–94 (2006)
15. Song, X., Tseng, B.L., Lin, C.Y., Sun, M.T.: Personalized Recommendation Driven by Information Flow. In: Proceedings of the 29th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, Seattle, WA, USA (2006)
16. Sun, Y., Li, H., Councill, I.G., Huang, J., Lee, W.C., Giles, L.: Personalized Ranking for Digital Libraries Based on Log Analysis. In: Proceeding of the 10th ACM workshop on Web Information and Data Management (WIDM), Napa Valley, CA, USA (2008)
17. Wei, Y.Z., Moreau, L., Jennings, N.R.: Learning Users' Interests by Quality Classification in Market-Based Recommender Systems. IEEE Trans. on Knowledge and Data Engineering 17(12), 1678–1688 (2005)
18. Wixom, B.H., Watson, H.J.: An Empirical Investigation of the Factors Affecting Data Warehousing Success. MIS Quarterly 25(1), 17–41 (2001)
19. Wixom, B.H., Watson, H.J., Hoffer, J.A.: Continental Airlines Continues to Soar with Business Intelligence. Information Systems Management 25(2), 102–112 (2008)

## Appendix A: Experiment Task and Tools

The experiment task was defined as – given a list of customers, choose those who will be targeted in a promotion campaign that offers a certain product. Due to certain costs (e.g., printing charges, mailing fees, and call time), promoting the product to the entire list would be sub-optimal, as only some customers are likely to purchase. An optimal decision would therefore be to target only those customers with a high likelihood to purchase enough units to justify the promotion cost. Such a decision can be formulated as maximizing an objective function:

$$V = \left(\sum_{m=1..M} I_m \left(PQ_m - C^V\right)\right) - C^F \quad (2)$$

Where,

| | |
|---|---|
| V - | Net-benefit, the decision value |
| $M, \{I_m\}$ - | The total number of customers (indexed [m]), and the set of binary decision variables, each indicating whether to include customer [m] in the campaign (=1) or not (=0), respectively |
| $P, Q_m$ - | Unit price, and the expected quantity of units that customer [m] will purchase ($Qm \geq 0$), respectively |
| $C^F, C^V$- | Fixed campaign cost, and promotion cost per customer, respectively |

The optimal decision would be to include only customers for which expected revenue is greater than promotion cost (i.e., $I_m=1$, when $P*Q_m>C^V$). However, while costs and unit price are typically known, the expected quantity of units is subject to uncertainty. Marketing professionals often estimate purchase intent by analyzing past sales, and identifying segments of customer who are likely to accept a promotion. Segments can be defined along certain customer characteristics and the associated values (e.g., "target male customers with no children who are between 30-45 years old"). Accordingly, the task was defined as – given (a) a list of customers, each with a known set of characteristics values (e.g., age, gender, occupation, etc.), (b) data on past purchases, and (c) unit price and cost parameters, choose the customer segments that would maximize the campaign's net-benefit.

To aid this task, participants were provided with a BI tool (similar to the one shown in Fig. 1), offered in two base versions – with and without FRM. The version with no FRM permitted the following BI functionality: (a) "Slicing": upon selecting a customer characteristic, the bar graph at the bottom summarizes the number of customers under each category, and the total sales associated with those customers. The tool presented 12 categorical customer characteristics, each with 3 possible values (e.g., High/Medium/Low Income). To prevent biased pre-assumptions on the predictive power of each characteristic, the characteristic names were coded with capital letters (A, B, C, …, L) and the associated value with enumerated lower-case letters - e.g., instead of High/Medium/Low Income, users would see a characteristic named A with associated values {a1, a2, a3}, (b)"Filtering": when slicing along a certain characteristic, a user would have the choice to limit the presentation to show only certain values (e.g., under characteristic F, show only values f1 and f3), and (c) "Drilling": after slicing along a certain characteristic, a user would have the choice to slice the data further along others. The numbers would then be summarized and presenters along the value-combinations of all the characteristic included – for example, if H(h1, h2) and G(g1, g3) where chosen, the tool will present summaries for the combinations {h1, g1}, {h1, g3}, {h2, g1}, and {h2, g3}.

The FRM-enhanced version included all the above BI functionality, implemented in a similar manner. However (as illustrated in Fig. 1), a certain rate was added per characteristic and per value, indicating a certain recommendation, which reflects potential contribution to a better decision. A high characteristic-value rate (r) would indicate a favorable customer segment - e.g., if, under D, r(d1) >> r(d2), the recommendation would be to prefer customers who belong to category d1. A high characteristic rate (R) would indicate a high rate variance among characteristic values; hence, a higher likelihood to single out better customer segments. If, for example, R(J) >> R(K), would mean a high variability between the rates of {j1, j2, j3} versus low variability between the rates of {k1, k2, k3}; hence, a recommendation to prefer slicing the data along J. Importantly, the FRM-enhanced tool recalculates the rates dynamically, depending on the characteristic-value combination observed, meaning that at each point of time the user would get a recommendation how to proceed which depends on the "Slicing", "Filtering" and "Drilling" choices made.

The simulated database included a list of 1000 customers, each with 12 associated characteristic values (e.g., {a1, b3, … , k2, l1}, which were randomly drawn using given value distributions (e.g., a1:0.3, a2:0.5, a3:0.2). Using a set of rules, which associated value combinations with levels of purchase intent, each customer [m] was assigned with a set of likelihood numbers $\{P_{m,z}\}$ of purchasing z (between 0 and 5) units, such that $\Sigma_z P_{m,z},=1$, and $Q_m=\Sigma_z ZP_{m,z}$, is the expected quantity of units purchased. We then generated a list of sale transactions per customer, by simulating a sequence of campaigns and randomly drawing the quantity of items purchased per campaign (including only transactions with quantity greater than 0). This random draw used the set of likelihood numbers per customer, with some level of randomness added. Users were also provided with a screen for selecting one or more customer segments to be targeted. Upon completing the selection – the campaign performance would be evaluated using Eq. 2.

# Process Performance Management – Identifying Stereotype Problem Situations as a Basis for Effective and Efficient Design Research

Anne Cleven, Felix Wortmann, and Robert Winter

Institute of Information Management, University of St. Gallen
Mueller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland
{anne.cleven,felix.wortmann,robert.winter}@unisg.ch

**Abstract.** Just recently many organisations get involved with process performance management (PPM). It appears, however, that PPM initiatives confront organisations with multi-faceted and complex challenges that call for a detailed problem analysis before any solution is developed. In this paper we introduce two patterns for identifying stereotype problem situations in design research (DR) and apply one to the field of PPM. The application gives detailed insights into typical PPM problem situations and illustrates the usefulness of our approach.

**Keywords:** Design Research, Design Science, Process Performance Management, Problem Definition.

## 1 Introduction

In today's hypercompetitive and globalized world measuring and in particular managing organisational performance is obligatory [8]. Hence, a myriad of frameworks has been developed during the last few decades each focussing on a somewhat different set of performance-related aspects [20]. Only few of these frameworks, however, put an emphasis on processes as the primary performance management object [20]. Likewise have performance issues been widely neglected in available business process management (BPM) approaches and systems so far. BPM systems are in fact "still very much workflow management systems (WfMS) and have not yet matured in the support of the BPM diagnosis" [32]. Although some software suites provide features like business activity monitoring (BAM) or audit trails the generation of meaningful reports or the prediction of process trends are still only rarely feasible [32]. Elbashir et al. [18] thus propose that effectively managing performance on a process level requires bringing together BPM on the one hand and business intelligence (BI) capabilities and techniques on the other. As the case may be, it is undisputed that a prerequisite for the development of a suitable PPM solution is a detailed analysis of the respective problem situation. Within the information systems (IS) research community the DR approach has been recognised as a valuable approach for rigorously developing useful problem solutions. With respect to how problem situations should be identified, however, there still is a lack of guidance as will be shown below.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 302–316, 2010.

The aim of this paper thus is twofold: Firstly, we seek to suggest two different approaches that offer advice on how to identify stereotype problem situations in DR in general. Secondly, applying one of the proposed approaches we strive for understanding the nature of stereotype PPM problem situations in particular. We address these objectives by first providing a conceptual background including a brief overview over the current DR literature and the concepts of BPM, BI and PPM (section 2). Subsequently, we propose two approaches for identifying stereotype problem situations and apply the first approach in the field of PPM (section 3). The concluding section 4 summarizes and discusses our approach and gives an outlook on future work.

## 2 Background

### 2.1 Design Research

The DR paradigm has been discussed thoroughly in recent years and is just now gaining ground as a valued contributor to both "building knowledge and improving practice" [39, p. 8] in the IS discipline. As opposed to natural and social research DR does not crave ultimate truths, grand theories or general laws, but seeks to identify and understand problematic real-world situations and transform these into more desired states by creating solutions that hurdle the respective dilemma [38, 52][1]. A number of authors have proposed reference processes sketching the activities that are essential for DR [25, 26, 38, 39, 44, 50]. Juxtaposing these approaches it becomes evident that the following six activities are commonly perceived as constituent for the DR process: (I) 'identify problem/need', (II) 'define objectives/requirements', (III) 'develop solution', (IV) 'evaluate', (V) 'reflect and theorize', and (VI) 'communicate/publish findings'. Owing to the fact that the DR process may either generate new research interests or uncover that the developed solution turns out unsatisfactory (or both) it is often described as cyclic or iterative [25, 26, 44]. With respect to DR output forms – commonly referred to as artefact types – March and Smith were the first to conceive a systemization. In their well-known 1995 paper they proposed the following four artefact types: constructs, models, methods, and instantiations [38]. Among others Walls et al. [60, 61], Venable [59] and Gregor [22] argue that theories, too, can and should be regarded as an essential fifth artefact type. Since DR represents a subfield of the IS discipline which aims at purposefully designing artefacts that address human and organisational problems [1], purely organisational as opposed to solely information technology (IT)-related artefacts may as well be seen as appropriate outcomes of DR [63].

As mentioned above DR artefacts are developed with the overall objective of transforming unsatisfactory real-world situations into more desired ones. The appropriateness or usefulness of a planned solution for a certain problem is thereby greatly contingent upon the unique factors that characterise the problem situation. Some researchers like Fiedler [19] in his work on the so-called 'Contingency Model' or Mitroff in his work on wicked problems [41] have emphasized the importance of thoroughly

[1] While DR as one part of IS design science research is aimed at creating solutions to specific classes of problems and thus involves the *actual* 'building', design science (DS) reflects on the artefact development process and – as such – involves *research into* the 'building' [49, 63].

analyzing and defining investigated problems in order to reveal their distinct characteristics. Up to now, however, extant DR literature has not incorporated these insights. Quite the contrary, while the analysis and specification of the problem situation within a DR project is of utmost importance for the success of the whole project it has so far not received much academic attention (as opposed to e.g. evaluation, which has been considered in great detail, cf. [3, 9, 47, 48]).

March and Smith, two of the very early authors of DR in IS, do not address the necessity of analyzing the problem situation at all [38]. Hevner et al. in their much-cited MISQ paper admittedly realize the importance of examining the 'environment', stating that "business needs are assessed and evaluated within the context of organisational strategies, structure, culture, and existing business processes" [26, p. 79]. Nonetheless, they do not provide any normative or prescriptive guidelines on *how* to actually assess the business needs in their specific context. Peffers et al. emphasize the importance of the "knowledge of the state of the problem" [44, p. 55] for effectively providing a solution. Further advice, however, only gets as specific as the recommendation that "it may be useful to atomize the problem conceptually so that the solution can capture its complexity." [44, p. 52]

### 2.2 Business Process Management

Although adjusting a business's activities along its value chain – as opposed to its functions or departments – had already been suggested by Nordsieck as early as in 1934 [43] the actual change from a vertical, functional to a horizontal, process oriented organisational structure only proceeded in the late 1980s. Especially Porter [45], Hammer and Champy [24], and Davenport [13] have contributed to the paradigm shift and promote a holistic and customer-focused process-orientation [15]. Nowadays, BPM is a well-established concept and a number of authors have developed a great variety of different approaches.

Across these approaches BPM is commonly understood as a set of methods, techniques, and tools to continuously (re-) design, enact, analyze, control and improve the an organisation's fundamental operational activities [34, 58, 64]. While putting different emphases in their works the majority of researchers agree in that BPM consists of four main processes that are affiliated with and build on each other, namely: (I) 'process definition, design and modelling', (II) 'process implementation and enactment', (III) 'process monitoring and controlling', and (IV) 'process optimisation and refinement'. Kueng and Krahn survey that in quite a number of today's organisations performance measurement and assessment on a process level is realized only in a mediocre way [33].

### 2.3 Business Intelligence

One of the very first persons to use the term 'business intelligence' (BI) was Hans Peter Luhn in his 1958 article "A Business Intelligence System" [36]. While in this early stage the perception of BI had a slightly different tinge and was mainly restricted to system- and technology-related issues, later definitions reflect the business-proximity and the methodological character of the concept. Today, BI (often likewise called business analytics) may be defined as "concepts and methods to improve business decision making by using fact-based support systems” [46, par. IV.2] and constitutes a crucial component within a company’s management reporting infrastructure. BI systems provide decision makers with timely, relevant, and easy to use information and can be defined as “specialized tools for data analysis, query, and reporting, (such as OLAP and dashboards)” [18, p. 138]. They are complemented by special IT infrastructure such as data warehouses, data marts, and “Extract, Transform and Load” tools which are required for their deployment and effective use [18].

BI serves a multitude of different purposes. Such being the case, Sircar calls it the “latest battlefield” [53, p. 293] for organisations striving for either survival or – even more ambitious - for a competitive advantage in nowadays global and fiercely rival business environment. The concept holds a bundle of analytical capabilities that allows organisations to effectively use their data and keep track of how exactly their business performs. These capabilities reach from retrospective, descriptive analyses, answering questions like ‘What happened?’ and ‘Where exactly was the problem?’ to future-oriented, predictive analyses, answering questions like ‘What happens if these trends continue?’ and ‘What will happen next?’ [14]. Descriptive or core analytics thus facilitate quick reactions or corrections, whereas predictive analytics allow for the anticipation of future values like, for example, customer needs. Current research issues in the field of BI are said to cover the quantification, measurement and evaluation of BI systems benefits and improved tactics for efficiently managing the BI processes within organisations [27, 35].

## 2.4 Process Performance Management

Organisational performance has an imperative influence on both the actions of companies and their competitiveness. Thus, the development of means for accurately measuring this performance has highly occupied researchers and practitioners in recent years [31, 37, 42]. The following list represents a digest of dimensions that are included in existing performance measurement frameworks: cost, quality, productivity, flexibility, efficiency, growth, processes, customer satisfaction, delivery, and environment [8, 20]. As of lately, a shift is perceptible that is taking the focus from the pure measurement of performance to a broader view of an actual management of performance [2, 20]. Performance Management can then be defined as “the use of performance measurement information to effect positive change in organisational culture, systems and processes” [20, p. 674].

Performance measurement and management systems address a variety of different aspects of performance, like in production planning [30] or cost accounting [17]. Only few, however, enable an organisation to effectively manage its performance on the level of business processes. Elbashir et al. state that the use of BI systems is obligatory for a company seeking to enhance its business process performance via measures like cost reduction and productivity improvement [18]. Some currently available WfMS provide rudimentary process monitoring and control functions but “have not yet matured in the support of the BPM diagnosis.” [32, p. 748]. Ko et al. further point out that some software suites include BAM dashboards and facilitate the creation of useful audit trails but the generation of “meaningful reports displaying process trends still requires external specialized reporting tools” [32, p. 749].

# 3 Identifying Fundamental PPM Problem Situations

## 3.1 Two Approaches for the Identification of Problem Situations

As has been pointed out in section 2.1 extant DR literature has only on the surface touched the question of how to exactly analyse the problem situation under consideration and how to identify and specify its unique characteristics. However, like with any other research endeavour specifying the exact problem situation is inevitably necessary in order to prevent both wasted time and money [11]. In this section we set out to propose two related approaches for the task of rigorously analysing a DR problem. The approaches may be interpreted as patterns as they are known and frequently used in the field of object-oriented programming [21]. A pattern describes a pair of a regularly recurring problem and a respective general solution. The general solution serves as a template that can be reused for fastening the problem solving process for a certain problem-class. The subsequently proposed patterns provide two general variants for systematically examining the specific characteristics of DR problems.

A major subfield of the IS discipline deals with software design and software engineering [4]. It is this subfield that we consulted when developing our approaches. A software development project always starts off with a detailed analysis of the exact functional and non-functional conditions and needs the planned software or system has to meet – the requirements engineering (RE) [5]. In their seminal 1998 book Sommerville and Kotonya, two authors who can be called pre-eminent in the field of RE, identified four essential activities for the RE process, namely: (I) 'requirements elicitation', (II) 'requirements analysis and negotiation', (III) 'requirements documentation', and (IV) 'requirements validation' [54]. The two herein proposed approaches relate to the first two activities of this process. The first approach, named *Lean Situation Identification Process* is depicted on the left-hand side of Fig. 1, whereas the second, named *Rich Situation Identification Process* is shown on the right-hand side. As there is never a "one-size-fits-all" solution to a problem [6], both approaches aim at identifying a number of different starting positions organisations may commence from. This allows for the development of solutions as close to the actual requirements as possible. Subsequently, the approaches are described in detail.

When starting a DR project the very first step concerns the identification of the problem to be solved. The problem may either be identified deductively by means of a literature analysis or be apprehended directly from real-world incidents [56]. In order to achieve a deeper understanding of the problem situation as relates to its very specific characteristics and to facilitate both a higher relevance and utility of the final solution(s) we propose to conduct an empirical investigation based on a survey. In order to identify fundamental, i.e. stereotype problem situations thus a questionnaire needs to be developed based on knowledge to be acquired in the area under investigation (cf. Fig. 1, phase (1) in both processes). Thereon, the questionnaire must be distributed to a population big and representative enough to warrant a reliable impression of the different situation stereotypes (cf. Fig. 1 phase (2) in both processes). While the two patterns proposed herein work alike during the phase of requirements elicitation, they differ with regard to the phase of requirements analysis and negotiation.

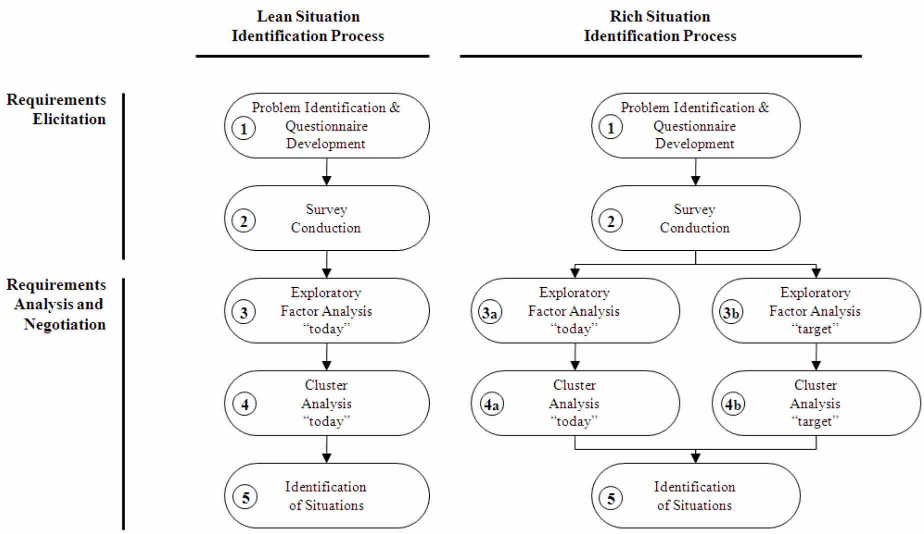



**Fig. 1.** DR Problem Situation Identification Patterns

The *Lean Situation Identification Process* is applied if the respondents selected for the survey are under time pressure and have a limited amount of time, whereas the *Rich Situation Identification Process* may be employed if the respondents have a considerable amount of time. In the first pattern the survey participants are only asked to answer the questionnaire as to how they conceive the current situation within their organisations ("today"). In contrast, the second pattern includes the investigation of the target situation, too. The actual analysis now begins with an exploratory factor analysis (EFA) which serves the purpose of developing a deeper understanding of the characteristics of the current problem situation (cf. Fig. 1 phase 3 in the first approach and phase 3a in the second) and those of the targeted situation (3b in the second approach). In general, EFA serves the identification of a number of important and mutually independent factors from a multiplicity of contingent variables [10]. Subsequently, the question of whether there are common situations which feature the same characteristics can be tackled using cluster analysis. A cluster analysis serves the purpose of partitioning a set of observations into subsets that are homogeneous within and heterogeneous amongst each other [23]. Again, the first pattern exclusively addresses the current situation, whereas the second also reveals situations the organisations polled are targeting at. Both approaches conclude with an interpretation of the identified clusters (cf. Fig. 1, phase (5) in both processes).

Either of the two approaches is associated with some specific assumptions as well as advantages and disadvantages. The *Lean Situation Identification Process* shows the clear advantage of having half as many questionnaire items as the second approach. Consequently, the analyses are only half as elaborate. However, the first approach comes along with the assumption that irrespective of the individual starting situation all companies are striving for one ideal target situation in which each of the identified factors or characteristics describing the starting position will be maximised. The characteristics of the first approach are briefly summarised in the following table.

**Table 1.** Characteristics of the Lean Situation Identification Process

| **Usage Scenarios** |
|---|
| – Respondents have a limited amount of time for questionnaire completion<br>– Exploratory analysis with a considerable amount of items |
| **Advantages** |
| – Lightweight questionnaire covering "today" questions only<br>– Lightweight analysis (only one factor and one cluster analysis)<br>– Low number of situations (cluster "today") results in less complex artefact construction and validation process |
| **Disadvantages** |
| – Situations are derived on the assumption that companies strive for a general improvement in regards to all factors, i.e. one ideal target situation exists |

The second approach, too, features some specifics. Also addressing the targeted situation, it produces a much bigger deal of work. In return, it allows for a much more precise analysis of the requirements for each situation by facilitating a detailed migration analysis from initial to target situations. The following table briefly summarises the features of the second approach.

**Table 2.** Characteristics of the Rich Situation Identification Process

| **Usage Scenarios** |
|---|
| – Respondents have a considerable amount of time for questionnaire completion<br>– Confirmatory analysis with a limited amount of items |
| **Aadvantages** |
| – Situations precisely depict the requirements |
| **Disadvantages** |
| – Time consuming questionnaire covering "today" and "target" questions<br>– Complex analysis ( two factor and two cluster analyses)<br>– High number of situations (cluster "today" x cluster "target") results in complex artefact construction and validation process |

Subsequently, we will apply the first pattern – the Lean Problem Situation Identification Process – for the examination of typical problem situations in the field of PPM.

## 3.2 Lean PPM Problem Situation Identification

### 3.2.1 Problem Identification and Questionnaire Development

As has been pointed out in the previous section gaining knowledge through reviewing existing literature in the field under investigation is essential for the development of the questionnaire (cf. section 3.1, [12]). As PPM represents a multidisciplinary field of study that builds on a variety of concepts, methods and techniques from other research areas, the most important ones being BPM, BI and performance management [40] the following sections represent the major building blocks of the questionnaire:

– *BPM*: BPM is a fundamental pillar for PPM. Thus, the questionnaire contains items addressing the degree to which an organisation is process oriented. Consequently, factors like clear process responsibilities (process officers) [51] and well-defined process documentation are covered. Moreover, the degree of consistent and transparent processes across systems and organisation are taken into account [7].

- *BI*: BI serves as the information backbone of PPM (cf. section 2.4). As well-defined BI responsibilities and processes are a prerequisite for effective and efficient information delivery, the questionnaire comprises both aspects. Furthermore, data integration and data quality are key for holistic and reliable PPM. Therefore, data integration and quality are also covered by the questionnaire.
- *PPM*: Moreover, the questionnaire contains items addressing what organisations actually measure (i.e., process cycle time, process costs, process quality). Furthermore, the data set covers information on where measures are deployed (i.e., production processes, sales processes). Finally, the questionnaire asks for the process performance management approach of organisations. Specifically, if process performance management is integrated into an overarching management approach (i.e., Balanced Scorecard, Corporate Performance Management) and if the plan-do-check-act cycle is applied for PPM.

#### 3.2.2 Survey Conduction

The questionnaire was distributed at a BI and DWH practitioner event held in October 2009. The participants were specialists and executives, working in the field of BI and DWH on both the IT and the business side thus having the required knowledge and information to answer the questions [12]. The questionnaire was designed to assess the current state of PPM in the participating organisations. Respective statements were formulated, and the respondents were requested to indicate current values for the degree of realization of each variable using a five-tiered Likert scale. Before being used, the questionnaire was pretested, both on an individual item level in early phases and as a whole before finally being distributed [12].

A total of 49 questionnaires were returned. If a data set was incomplete, i.e. if one or more than one of the 21 items was missing, the questionnaire was disregarded. On the basis of this criterion, 45 questionnaires were selected for further analysis. Although the data set is rather small, the sample can be considered adequate for the purpose of an EFA [16]. The interviewed organisations are primarily large and medium-sized companies from the German-speaking countries. 60% have more than 1000 employees and another 22% have more than 100 employees. The sectors mainly represented were professional services (40%), banking, finance and insurance (29%), high tech (11%) manufacturing and consumer goods (7%), media and telecommunication (5%), and others (8%).

#### 3.2.3 Exploratory Factor Analysis

The EFA was performed on the data set covering 21 items. The measure of sampling adequacy (MSA, "Kaiser-Meyer-Olkin criterion") for the data set is 0.777. MSA represents an indicator for the extent to which the input variables belong together and provides information on whether a factor analysis can reasonably be performed or not. Kaiser and Rice appraise a value of 0.7 or more as "reasonable", i.e. the data set is considered to be appropriate for applying EFA [29, 55]. Five factors that jointly explain about 75.6% of the total variance were extracted by means of principal component analysis. Both the Kaiser criterion and the scree plot point to this solution. The resulting component matrix was rotated using the Varimax method with Kaiser normalization in order to improve the interpretability of the items' assignment to the factors [28]. The rotated component matrix is depicted in Table 3.

**Table 3.** Factor loadings

| Item description | F1 | F2 | F3 | F4 | F5 |
|---|---|---|---|---|---|
| **Factor (1) Broad, PDCA-based use of PPM measures**<br>*Eigenvalue = 9.510; Variance explained = 45.287%* | | | | | |
| Adherence to schedules is measured for processes. | **.834** | .304 | .059 | .165 | .102 |
| Capacity utilization is measured for processes. | **.787** | .345 | .332 | .202 | .083 |
| Quality is measured for processes. | **.758** | .209 | .228 | .202 | .190 |
| Process resource utilisation is measured. | **.735** | .395 | .069 | .060 | .059 |
| Process costs are measured. | **.728** | .362 | .305 | .067 | -.030 |
| Process cycle times are measured. | **.723** | .221 | .324 | .331 | .190 |
| The plan-do-check-act (PDCA) cycle is applied for PPM. | **.543** | .168 | .214 | .415 | .308 |
| **Factor (2) BPM maturity**<br>*Eigenvalue = 2.530; Variance explained = 12.049%* | | | | | |
| Process flows are consistent and transparent beyond functional borders (organisational unit, division and department). | .239 | **.796** | .284 | .045 | .063 |
| Processes have defined process officers. | .203 | **.765** | .057 | .105 | .044 |
| Processes are consistently documented and/or modelled. | .321 | **.758** | -.062 | .180 | .216 |
| Process flows are consistent and transparent beyond system borders. | .404 | **.718** | .358 | -.047 | .167 |
| Process orientation is a central paradigm. | .289 | **.660** | -.110 | .220 | .046 |
| **Factor (3) PPM process diffusion**<br>*Eigenvalue = 1.527; Variance explained = 7.269%* | | | | | |
| PPM also covers non-financial measures. | .034 | -.047 | **.843** | .280 | -.121 |
| PPM is deployed for purchasing processes. | .282 | .156 | **.797** | .158 | .003 |
| PPM is deployed for production processes. | .260 | -.064 | **.714** | .272 | .249 |
| PPM is deployed for sales processes. | .268 | .394 | **.702** | .008 | ..137 |
| **Factor (4) BI-enabled, integrated PPM**<br>*Eigenvalue = 1.209; Variance explained = 5.759%* | | | | | |
| Defined BI governance responsibilities and processes are in place. | .033 | .270 | .134 | **.775** | .094 |
| PPM is part of the enterprise-wide Balanced Score Card (BSC). | .392 | -.031 | .387 | **.750** | -.020 |
| PPM is part of the Corporate Performance Management (CPM). | .297 | .153 | .469 | **.613** | .102 |
| **Factor (5) High quality information base**<br>*Eigenvalue = 1.109; Variance explained = 5.283%* | | | | | |
| A central integrated data base is in place (e.g., an Enterprise Data Warehouse). | .009 | .125 | .018 | .107 | **.907** |
| Data quality is consistently high. | .279 | .128 | .075 | .023 | **.814** |

The five factors constitute themselves as follows:

- There are seven items loading on the first factor, in the following referred to as *Broad, PDCA-based use of PPM measures*. All of the items are metric-related and address two questions: a) what is measured in the context of PPM (which key performance indicators (KPIs) are used), and b) how is the measurement accomplished. Process KPIs explicitly addressed by this factor are adherence to schedule, capacity utilization, process quality, process resource utilization, process costs and process cycle times. The plan-do-check-act (PDCA) cycle is applied for KPI management.

- All five BPM-related items were found to have significant impact on the second factor, accounting for the degree of *BPM maturity*. Organisations having a high BPM maturity, advocate process orientation as a central paradigm, foster process documentation/modelling and clear process responsibilities thereby assuring consistent process flows across organisations and systems.
- Another four items load high on the third factor, representing *PPM process diffusion*. This factor expresses the degree of KPI usage in core business processes (covering procurement, production and sales activities). Furthermore, this factor strengthens the importance of non-financial measures.
- Three variables were found to have significant impact on the fourth factor, subsequently referred to as *BI-enabled, integrated PPM*. Organisations showing a high performance regarding this factor rely on well-defined, well-coordinated processes and management approaches in performance management as well as in BI. In these companies PPM is part of an integrated and comprehensive management approach i.e. BSC or CPM. To enable concepts like BSC or CPM defined BI governance responsibilities and processes are in place.
- Finally, there are two variables loading on the fifth factor, in the following referred to as *High quality information base*. According to our analysis, companies having a high quality information base build upon a central integrated data store (e.g., an Enterprise Data Warehouse) thereby assuring consistently high data quality across the organisation.

#### 3.2.4 Cluster Analysis

In order to identify organisations with similar problem situations, cluster analysis is used. The cluster analysis is based on factor scores being calculated using the regression method [57]. The Ward fusion algorithm and the squared Euclidean distance are applied for clustering, as this combination finds very good partitions resulting in an appropriate number of clusters and similar number of observations in each cluster [23, 62]. On the basis of the dendrogram, i.e. the graphical representation of the fusion process and the cluster sizes the final number of clusters was defined [23]. Table 4 contains the arithmetic means of the factor scores for each of the four clusters.

**Table 4.** Arithmetic means of factors per cluster

| Cluster | n | F1 | F2 | F3 | F4 | F5 |
|---|---|---|---|---|---|---|
| Cluster 1 | 11 | -0.264 | -0.214 | 0.206 | -0.559 | -1.159 |
| Cluster 2 | 16 | -0.234 | 0.301 | -0.069 | -0.605 | 0.702 |
| Cluster 3 | 11 | 0.743 | -0.626 | -0.599 | 0.580 | 0.056 |
| Cluster 4 | 7 | -0.220 | 0.632 | 0.775 | 1.351 | 0.127 |

#### 3.2.5 Identification of Situations

For an easier interpretation and better comparison the cluster means in Fig. 2 are standardized: It depicts how many standard deviations a cluster value (cluster means from Table ) is above or below the overall factor mean (across all cluster values).

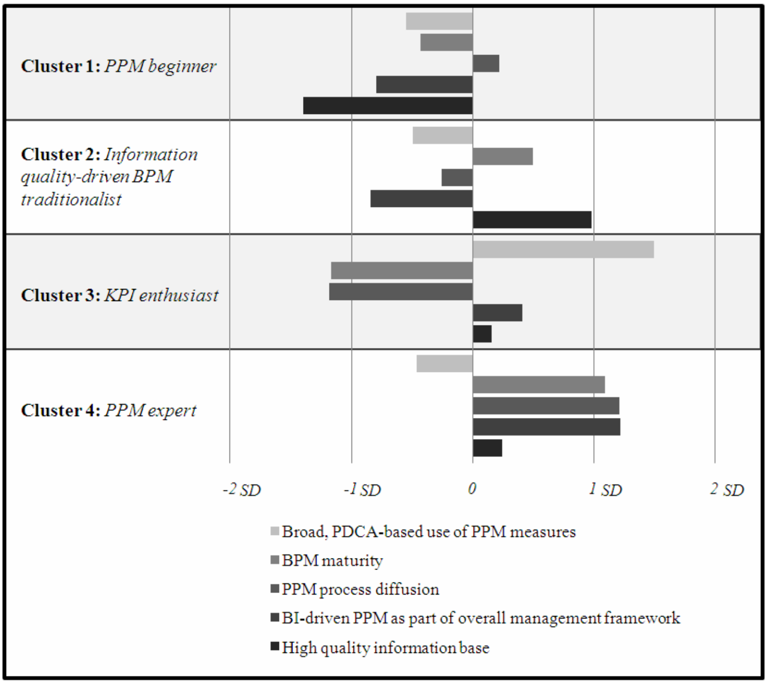



**Fig. 2.** Standardized arithmetic means of factors per cluster

The four clusters can be interpreted as follows:

- **Cluster 1** – *PPM beginner*: The first cluster is characterized by poor performance regarding four of the five factors. Only PPM process diffusion, i.e. the degree of KPI usage in core business processes is at an above average level. The organisations of this cluster show a significant lack of performance in regards to a high quality information base. Neither from a BI perspective nor from a BPM perspective are fundamental concepts in place that could be leveraged as a basis for PPM.
- **Cluster 2** – *Information quality-driven BPM traditionalist*: The second cluster shows positive performance with respect to BPM and information quality. BPM as well as information quality form a solid basis for PPM. Nevertheless, all three PPM related factors show low performance. Companies of this cluster specifically lack well-defined, well-coordinated processes and management approaches in performance management as well as in BI.
- **Cluster 3** – *KPI enthusiasts*: The third cluster is characterized through high BI and low BPM performance. The extend to which KPIs are employed, appears very ambiguous: On the one hand the number of different measures in place is very high. On the other only a very small number of business processes are being measured.
- **Cluster 4** – *PPM expert*: Companies of the fourth cluster show a high performance across four of the five factors. The only low performing factor is factor one: In contrast to the KPI enthusiasts, PPM experts apply PPM to a significant number of their business processes building upon a consistent set of limited metrics.

The above described clusters represent typical problem situations, or ‘points of departure’ organisations may commence from when getting involved with PPM. These stereotype situations may now serve as a basis for defining the objectives of any type of artefact required to support the respective PPM initiative.

## 4 Conclusion and Future Research

The aim of this paper was twofold: For one thing, we sought to suggest two patterns offering advice on how to identify stereotype problem situations in DR in general. For another thing, by applying the first of the proposed patterns we aimed at understanding the nature of stereotype PPM problem situations in particular. With the former we hope to contribute to the extant DR literature by providing a rigorous and yet easy-to-use way for systematically analysing the specific characteristics of a problem under investigation. Applying the approach in the context of PPM a) in fact revealed four distinct problem situations that mark the ‘point of departure’ for possible PPM projects and b) showed that the approach provides valuable assistance for analysing a DR problem. We do, however, not want to restrain that there is need for further research: In their current version the patterns are not yet deeply elaborated and approved, and surely need further evaluation and refinement. Moreover, the patterns assist a design researcher in gaining a first good understanding of a problem situation. Most certainly, however, additional deeper analyses will be required for the actual building of artefacts. Not least, the applicability of the second pattern has to be assessed and both patterns should be used in further contexts to carve out potential deficiencies.

## References


1. Alter, S.: Work Systems and IT Artifacts - Does the Definition Matter? Commun. Assoc. Inf. Syst. 17(14), 2–41 (2006)
2. Amaratunga, D., Baldry, D.: Moving from Performance Measurement to Performance Management. Facil. 20(5/6), 217–223 (2002)
3. Baskerville, R., 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, Pasadena, CA (2007)
4. Benbasat, I., Zmud, R.W.: The Identity Crisis within the IS Discipline: Defining and Communicationg the Discipline’s Core Properties. Manag. Inf. Syst. Quart. 27(2), 183–194 (2003)
5. Berenbach, B., Paulish, D., Kazmeier, J., Rudorfer, A.: Software & Systems Requirements Engineering: In Practice. McGraw-Hill, New York (2009)
6. Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R.: Situational Method Engineering - On the Differentiation of "Context" and "Project Type". In: IFIP WG8.1 Working Conference on Situational Method Engineering - Fundamentals and Experiences, pp. 33–48. Springer, Geneva (2007)
7. Bucher, T., Winter, R.: Taxonomy of Business Process Management Approaches: An Empirical Foundation for the Engineering of Situational Methods to Support BPM. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management. Springer, Heidelberg (2010) (to be published)

8. Carton, R.B., Hofer, C.W.: Measuring Organizational Performance. Metrics for Entrepreneurship ans Strategic Management Research. Edward Elgar Publishing Limited, Cheltenham (2006)
9. Cleven, A., Gubler, P., Hüner, K.: Design Alternatives for the Evaluation of Design Science Research Artifacts. In: 4th International Conference on Design Science Research in Information Systems and Technology, Philadelphia, PA (2009)
10. Conway, J.M., Huffcutt, A.I.: A Review and Evaluation of Exploratory Factor Analysis Practices in Organizational Research. Organ. Res. Meth. 6(2), 147–168 (2003)
11. Creswell, J.W.: Research Design. Qualitative, Quantitative, and Mixed Methods Approaches. Sage Publications, Inc., Thousand Oaks (2003)
12. Czaja, R., Blair, J.: Designing Surveys - A Guide to Decisions and Procedures. Pine Forge Press, Thousand Oaks (1996)
13. Davenport, T.H.: Process Innovation. Reengineering Work through Information Technology. Information Technology. Harvard Business Press, Boston (1993)
14. Davenport, T.H., Harris, J.G.: Competing on Analytics. Harvard Business School Publishing Corporation, Boston (2007)
15. de Toro, I., McCabe, T.: How to Stay Flexible and Elude Fads. Qual. Prog. 30(3), 55–60 (1997)
16. de Winter, J.C.F., Dodou, D., Wieringa, P.A.: Exploratory Factor Analysis With Small Sample Sizes. Multivar. Behav. Res. 44, 147–181 (2009)
17. Durden, C.H., Hassel, L.G.: Cost accounting and performance measurement in a just-in-time production environment. Asia Pac. J. Manag. 16(1), 111–115 (1999)
18. Elbashir, M.Z., Collier, P.A., Davern, M.J.: Measuring the Effects of Business Intelligence Systems: The Relationship between Business Process and Organizational Performance. Int. J. Account. Inf. Syst. 9, 135–153 (2008)
19. Fiedler, F.E.: A Contingency Model of Leadership Effectiveness. Adv. Exp. Soc. Psychol. 1, 149–190 (1964)
20. Folan, P., Browne, J.: A review of performance measurement: Towards performance management. Comput. Ind. 56, 663–680 (2005)
21. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, Reading (1994)
22. Gregor, S.: Design Theory in Information Systems. Australasian J. Inf. Syst. 10, 14–22 (2002)
23. Hair, J.F.J., Black, W.C., Babin, B.J., Anderson, R.E., Tatham, R.L.: Multivariate Data Analysis. Pearson Prentice Hall, Upper Saddle River (2006)
24. Hammer, M., Champy, J.: Reengineering the Corporation - A Manifesto for Business Revolution. HarperCollins Publishers, New York (1993)
25. Hevner, A.R.: A Three Cycle View of Design Science Research. Scand. J. Inf. Syst. 19(2), 87–92 (2007)
26. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. Manag. Inf. Syst. Quart. 28(1), 75–105 (2004)
27. Jourdan, Z., Rainer, K., Marshall, T.E.: Business Intelligence: An Analysis of the Literature. Inf. Syst. Manag. 25, 121–131 (2008)
28. Kaiser, H.F.: The Varimax Criterion for Analytic Rotation in Factor Analysis. Psychometrika 23(3), 187–200 (1958)
29. Kaiser, H.F., Rice, J.: Little Jiffy, Mark IV. Educ. Psychol. Meas. 34(1), 111–117 (1974)
30. Karacapilidis, N.I., Pappis, C.P.: Production planning and control in textile industry: A case study. Comput. Ind. 30(2), 127–144 (1996)

31. Keegan, D., Eiler, R., Jones, C.: Are your performance measures obsolete? Manag. Account. 70(12), 45–50 (1989)
32. Ko, R.K.L., Lee, S.S.G., Lee, E.W.: Business process management (BPM) standards: a survey. Bus. Process Manag. J. 5, 744–791 (2009)
33. Kueng, P., Krahn, A.: Building a Process Performance Measurement System - Some Early Experiences, in: Journal of Scientific and Industrial Research. Journal of Scientific and Industrial Research 58(3/4), 149–159 (1999)
34. Lee, R.G., Dale, B.G.: Business Process Management: A Review and Evaluation. Bus. Process Manag. J. 4(3), 214–225 (1998)
35. Lönnqvist, A., Pirttimäki, V.: The Measurement of Business Intelligence. Inf. Syst. Manag, 32–40 (Winter 2006)
36. Luhn, H.P.: A Business Intelligence System. IBM J. Res. Dev. 2(4), 314–319 (1958)
37. Lynch, R., Cross, K.: Measure Up – The Essential Guide to Measuring Business Performance. Mandarin, London (1991)
38. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decis. Support Syst. 15(4), 251–266 (1995)
39. Marshall, P., McKay, J.: A Review of Design Science in Information Systems. In: 16th Australasian Conference on Information Systems, Sydney, Australia, pp. 1–11 (2005)
40. Melchert, F., Winter, R., Klesse, M.: Aligning Process Automation and Business Intelligence to Support Corporate Performance Management. In: Tenth Americas Conference on Information Systems (AMCIS 2004), New York, NY, USA, pp. 4053–4063 (2004)
41. Mitroff, I.I.: William James and a Theory of Thinking. J. Inf. Technol. Theory Appl. 6(2), 83–89 (2004)
42. Neely, A., Adams, C., Kennerley, M.: The Performance Prism: The Scorecard for Measuring and Managing Business Success. Prentice Hall, London (2002)
43. Nordsieck, F.: Grundlagen der Organisationslehre. Poeschel, Stuttgart (1934)
44. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. J. Manag. Inf. Syst. 24(3), 45–77 (2007)
45. Porter, M.E.: Competitive Strategy: Techniques for Analyzing Industries and Competitors. Free Press, New York (1980)
46. Power, D.J.: A Brief History of Decision Support Systems, http://dssresources.com/history/dsshistory.html
47. Pries-Heje, J., Baskerville, R., Venable, J.: Strategies for Design Science Research Evaluation. In: 16th European Conference on Information Systems, Galway, Ireland, pp. 255–266 (2008)
48. Pries-Heje, J., Venable, J., Baskerville, R.: A Risk Management Framework for Design Science Research. In: 31st Information Systems Research Seminar in Scandinavia, Åre, Sweden, pp. 1–18 (2008)
49. Purao, S., Baldwin, C., Hevner, A., Storey, V.C., Pries-Heje, J., Smith, B., Zhu, Y.: The Sciences of Design: Observations on an Emerging Field. Commun. Assoc. Inf. Syst. 23, 523–546 (2008)
50. Rossi, M., Sein, M.: Design Research Workshop: A Proactive Research Approach. Presentation delivered at IRIS 26, http://tiesrv.hkkk.fi/iris26/presentation/workshop_designRes.pdf
51. Samson, D., Challis, D.: Patterns of Business Excellence. Meas. Bus. Excell. 6(2), 15–21 (2002)
52. Simon, H.A.: The Sciences of the Artificial. The MIT Press, Cambridge (1969)

53. Sircar, S.: Business Intelligence in the Business Curriculum. Commun. Assoc. Inf. Syst. 24(17), 289–302 (2009)
54. Sommerville, I., Kotonya, G.: Requirements Engineering. Processes and Techniques. Wiley, New York (1998)
55. Stewart, D.W.: The Application and Misapplication of Factor Analysis in Marketing Research. J. Mark. Res. 18(1), 51–62 (1981)
56. Swamidass, P.M.: Empirical Science: New Frontier in Operations Management Research. Acad. Manag. Rev. 16(4), 793–814 (1991)
57. Thompson, B.: Exploratory and Confirmatory Factor Analysis: Understanding Concepts and Applications. American Psychological Association, Washington (2004)
58. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, W.: 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. 1611–3349. Springer, Heidelberg (2003)
59. Venable, J.R.: The Role of Theory and Theorising in Design Science Research. In: 1st International Conference on Design Science Research in Information Systems and Technology, Claremont, CA, pp. 1–18 (2006)
60. Walls, J., Widmeyer, G., Sawy, O.E.: Building an Information System Design Theory for Vigilant EIS. Inf. Syst. Res. 3(1), 36–59 (1992)
61. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Assessing information system design theory in perspective: how useful was our 1992 initial rendition. J. Inf. Technol. Theory Appl. 6(2), 43–58 (2004)
62. Ward Jr., J.H.: Hierarchical Grouping to Optimize an Objective Function. J. Am. Stat. Assoc. 58(301), 236–244 (1963)
63. Winter, R.: Design science research in Europe. Eur. J. Inf. Syst. 17, 470–475 (2008)
64. Zairi, M.: Business Process Management: A Boundaryless Approach to Modern Competitiveness. Bus. Process Manag. J. (1997)

# The Design of Focus Area Maturity Models

Marlies van Steenbergen[1], Rik Bos[2], Sjaak Brinkkemper[2], Inge van de Weerd[2], and Willem Bekkers[2]

[1] Sogeti Netherlands, Wildenborch 3, 1112 XB Diemen, The Netherlands
Marlies.van.Steenbergen@sogeti.nl
[2] Department of Information and Computing Sciences, Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands
{R.Bos,S.Brinkkemper,I.vandeWeerd,Bekkers}@cs.uu.nl

**Abstract.** Maturity models are a well-known instrument to support the improvement of functional domains in IS, like software development or testing. While maturity models may share a common structure, they have to be developed anew for each functional domain. Focus area maturity models are distinguished from fixed-level maturity models, like CMM, in that they are especially suited to the incremental improvement of functional domains. In this paper we present a generic method for developing focus area maturity models based on both extensive industrial experience and scientific investigation. In doing so, we show two examples of focus area maturity models, one for enterprise architecture and one for software product management. We used a design science research process to develop the method presented.

**Keywords:** Design Research Methodology, Design Science, Enterprise Architecture, Software Product Management, Maturity Model, Maturity Matrix, Method Engineering.

## 1 Introduction

Capability development in functional domains in IS, like enterprise architecture or software product management, is a complex issue. Decisions have to be made with regard to how to develop new processes, deliverables and competences. As it is not possible to implement a fully mature function from scratch, functional domains are developed incrementally, improving them step by step. Maturity models are a means to support such incremental development, as they distinguish different maturity levels that an organization successively progresses through. As such they can be used as a guideline for balanced incremental improvement of a functional domain.

Numerous maturity models for various functional domains have been developed over the past years [1, 2]. Most of these maturity models are so-called fixed-level models, like CMM [3]. Fixed-level maturity models distinguish a fixed number, usually around five, of generic maturity levels. Each maturity level is associated with a number of processes that have to be implemented. A limitation of fixed-level models is that they are not geared to expressing interdependencies between the processes making up the maturity levels [1, 4]. Because of this, they provide little guidance in

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 317–332, 2010.

determining the order in which to implement these processes, other than that lower level processes are to be implemented before higher level processes. Fixed-level models are perceived too large and heavy to use by some organizations [16].

Another type of maturity models are the so-called focus area maturity models [5]. Focus area maturity models are based on the concept of a number of focus areas that have to be developed to achieve maturity in a functional domain. Examples of focus areas are the development and maintenance of certain processes or deliverables, alignment with other disciplines, and training of certain competences. The identification of the exact focus areas depends on the functional domain. A focus area maturity model defines for each of its focus areas a series of development steps in the form of progressively mature capabilities. These capabilities are specific to the focus areas identified. This is a departure from the fixed number of generic maturity levels that the fixed-level maturity models are based on. The variation in levels that can be noticed between different fixed-level maturity models [4] suggests that the assumption of the existence of generic maturity levels is an oversimplification. We share the view that different dimensions have different maturity levels [2], taking it even one step further and claiming that each focus area has its own number and type of maturity levels. By juxtaposing all capabilities of all focus areas relative to each other, a balanced, incremental development path, taking all focus areas into account, is defined. This juxtaposition of capabilities is done by positioning the capabilities in a matrix as shown in figure 1, which gives an example of a focus area maturity model in the functional domain of enterprise architecture.

| ***Maturity Scale***<br>***Focus Area*** | ***0*** | ***1*** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** | ***7*** | ***8*** | ***9*** | ***10*** | ***11*** | ***12*** | ***13*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Development of architecture | | A | | | B | | | C | | | | | | |
| Use of architecture | | | A | | | B | | | | C | | | | |
| Alignment with business | | A | | | | B | | | | C | | | | |
| Alignment with the development process | | | A | | | | B | | C | | | | | |
| Alignment with operations | | | | | A | | | B | | | C | | | |
| Relationship to the as-is state | | | | | A | | | | B | | | | | |
| Roles and responsibilities | | | | A | | B | | | | | C | | | |
| Coordination of developments | | | | | | | A | | | B | | | | |
| Monitoring | | | | A | | B | | C | | D | | | | |
| Quality management | | | | | | | | A | | B | | | C | |
| Maintenance of the architectural process | | | | | | | A | | B | | C | | | |
| Maintenance of architectural deliverables | | | | | A | | | B | | | | | C | |
| Commitment and motivation | | A | | | | | B | | C | | | | | |
| Architectural roles and training | | | | A | | B | | | C | | | D | | |
| Use of an architectural method | | | | A | | | | | | B | | | | C |
| Consultation | | | A | | B | | | | C | | | | | |
| Architectural tools | | | | | | | A | | | | B | | | C |
| Budgeting and planning | | | | A | | | | | | | B | | C | |

**Fig. 1.** A focus area maturity model for the functional domain of enterprise architecture

The focus areas are given in the left column, the capabilities per focus area are depicted by the letters A to D, which stand for progressively mature capabilities. The actual maturity of a specific organization can be depicted by coloring the cells up until the next capability that has not been implemented yet. The rightmost column that is completely colored indicates the maturity scale of the organization assessed. Thus, the organization in figure 1 is at maturity scale 1 as the capability A of `Use of` architecture in column 2 has not been achieved. We will explain the model in more detail in section 3.

The focus area maturity model makes it possible to distinguish more than five overall stages of maturity. This results in smaller steps between the stages, providing more detailed guidance to setting priorities in capability development. The positioning of the capabilities in a matrix makes this kind of model well suited to express the sometimes complex combinations of different factors that determine the effectiveness of a function. Departing from the five fixed maturity levels makes the focus area oriented model more flexible in defining both focus areas and interdependencies between focus areas. Especially in relatively new functional domains, such a fine-grained approach is asked for to guide incremental improvement.

The focus area maturity model originated in the domain of software testing [6]. Subsequently, focus area maturity models were developed for the domains of enterprise architecture and software product management. From these applications we derived a generic development method for focus area maturity models for other functional domains in IS. We did so by applying the design science research methodology for information systems research introduced by Peffers et al. [7].

The research contribution of this paper is the presentation of a development method for focus area maturity models. In terms of the research contributions guideline of Hevner et al. this is a contribution to the design foundations [8]. The practical relevance lies in the fact that we provide practitioners and researchers with a method to develop new focus area maturity models that may support practitioners in developing IS functional domains more effectively.

In section 2 we present our research approach. Section 3 discusses the application of the focus area maturity model in two fields: enterprise architecture and software product management. In section 4 the maturity matrix is mathematically formalized. A development method for focus area models is presented in section 5. In section 6, we discuss conclusions and suggestions for further research.

## 2 Research Approach

The objective of our research is to define a development method for focus area maturity models that aids researchers and practitioners in developing a maturity model for incremental improvement of a specific IS functional domain. Peffers et al. distinguish four different entry points to the design science research process distinguishing problem-centered, objective-centered, design and development-centered and client/context initiated approaches [7]. As our purpose is to contribute to the improvement of the IS function we applied an objective-centered approach: "an objective-centered solution (...) could be triggered by an industry or research need that can be addressed by developing an artifact.". We use the design science process to present our research approach:

1. **Problem identification and motivation.** The problem motivating our research is how to develop capabilities in a given functional domain in an incremental, balanced manner. In their quests for continuous improvement, practitioners and researchers are looking for well-founded development paths [2, 9]. As argued in the introduction, focus area maturity models provide such development paths.

2. **Define the objectives for a solution.** The objective of our solution, a development method for focus area maturity models, is to provide a method to develop a step by step improvement approach for a specific functional domain. This method must be well-founded and enable practitioners and scientists to design an optimal and feasible improvement path to a fully mature function.
3. **Design and development.** The focus area maturity model development method is derived from both literature review and practical experience. From the literature review we defined a number of generic phases in developing maturity models. To detail these phases for the development of focus area maturity models we also draw on the lessons learned from the development of focus area maturity models in the fields of enterprise architecture and software product management.
4. **Demonstration.** The use of the development method is initially demonstrated by retrospectively applying it to two cases. Further demonstration must take place by applying it to a new field. This is to be done yet.
5. **Evaluation.** The development method is evaluated by applying the requirements for the development of maturity models defined by Becker et al. [9], which were derived from the seven guidelines presented by Hevner et al. [8].
6. **Communication.** Besides communication of the development method in the scientific community by publication in conferences and journals, the method will be published in practitioners' forums.

## 3 Focus Area Maturity Models

The core of the focus area maturity model consists of the focus areas. Each focus area can be divided into a number of capabilities. By positioning these capabilities against each other in a matrix, as shown in figure 1, the model presents the order in which the different aspects of a functional domain should be addressed and implemented. A *functional domain* is the whole of activities, responsibilities and actors involved in the fulfillment of a well-defined function within an organization. We define a *focus area* as an aspect that has to be implemented to a certain extent for a functional domain to be effective. The collection of focus areas provides a complete and mutually disjoint coverage of the functional domain. With each focus area a number of capabilities are associated, depicted in the matrix by capital letters. A *capability* is here defined as an ability to achieve a predefined goal that is associated with a certain maturity level. For example in the enterprise architecture matrix in figure 1, the focus area Use of architecture has three capabilities A: architecture used informatively, B: architecture used to steer content and C: architecture integrated into the organization, representing a progression in maturity. The position of the letters in the matrix indicates the order in which the capabilities of the different focus areas must be addressed and implemented to build an architecture practice in a balanced manner. With the matrix we can define both intra-process dependencies between capabilities, where one capability must be implemented after another capability in the same focus area and inter-process dependencies, where a capability must be implemented after a capability in another focus area.

The fourteen columns in the enterprise architecture matrix of figure 1 define progressive overall maturity scales, scale 0 being the lowest and scale 13 being the highest scale achievable. An organization is said to be at the maturity scale represented by the rightmost column for which the organization has achieved all focus area capabilities positioned in that column and in all columns to its left. The organization depicted in figure 1 has already implemented some capabilities, indicated by the colored cells. It shows an unbalance, however, in that some focus areas, like Alignment with the development process, are quite advanced, while others, like Use of architecture, are not yet developed at all. Thus despite the development of some of the focus areas, on the whole the organization in figure 1 is still only at scale 1. To achieve a balanced enterprise architecture function, its first step should be to develop the focus area Use of architecture to its first capability (the A in column 2), followed by the first capability of Monitoring (the A in column 3). By implementing these capabilities the organization will progress from maturity scale 1 to scale 3.

While the focus area oriented model originates from the field of testing [6], we applied it in the last seven years to the IS fields of enterprise architecture and software product management. We discuss these applications in the next sections.

### 3.1 The DyA Architecture Maturity Matrix

The DyA Architecture Maturity Matrix (DyAMM) is the application of the focus area maturity model in the field of enterprise architecture. The DyAMM is developed as part of the DyA program in which an approach to enterprise architecture is developed, called Dynamic Architecture (DyA), that focuses on a goal-oriented, evolutionary development of the architectural function [10, 11]. The DyAMM was developed in 2002 and has been applied to over 50 organizations since. In 2004 it was slightly adjusted based on the first few applications. The resulting version was qualitatively validated in a case study [5]. A number of organizations use the DyAMM to give direction to an improvement program of years, performing a yearly assessment to monitor progress. In 2009 a quantitative analysis of the DyAMM was performed with a dataset of 56 cases [12].

The core of the DyAMM is the matrix depicted in figure 1, with each capability associated with one to four yes/no assessment questions to assess its implementation and one or more improvement actions that may support achieving it. Maturity assessment is performed by answering the yes/no questions. Only if all questions associated with a capability can be answered confirmatively, the associated capability can be said to be achieved. Table 1 shows as an example the questions associated with capability A of the focus area Use of architecture.

**Table 1.** Questions to measure maturity level A of focus area Use of architecture

| Nr. | Question |
|---|---|
| 9 | Is there an architecture that management recognizes as such? |
| 10 | Does the architecture give a clear indication of what the organization wants? |
| 11 | Is the architecture accessible to all employees? |

In all there are 137 assessment questions associated with the DyAMM. The primary use of the DyAMM is as an assessment instrument used by independent assessors. The assessors fill the matrix by answering all 137 questions, basing their answers on interviews with relevant stakeholders and studying documentation. In addition, the

DyAMM is used as a self assessment to be completed by individuals for their own organization. Architects can answer the 137 questions for themselves, which leads to a matrix profile, like the example in figure 1.

## 3.2 The SPM Maturity Matrix

Many software companies have made a shift from developing custom-made software to developing product software [13]. This means that many internal processes in these companies need to be adapted. Instead of developing a customized product for one customer, a standard product is developed for a whole range of customers. To cope with this, product software companies need to introduce the right software product management (SPM) processes.

In [14], the reference framework for SPM was proposed. This framework presents 14 SPM processes divided over four business functions: Portfolio management, Product roadmapping, Release planning and Requirements management. In addition, the SPM maturity matrix was developed in order to support local analysis and incremental improvement of SPM processes [15, 16]. The SPM matrix has been validated in approximately 15 case studies in Dutch companies of varying sizes. In addition, a survey was conducted to validate the positioning of the capabilities.

Similarly to the DyAMM, the SPM maturity matrix consists of focus areas and capabilities. The focus areas correspond directly with the SPM processes in the earlier published reference framework. In addition, the focus areas are divided into four groups, corresponding to the four identified business functions that are mentioned in the preceding paragraph. In Figure 2, the SPM Maturity Matrix is presented.

| ***Maturity Scale***<br>***Focus Area*** | ***0*** | ***1*** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** | ***7*** | ***8*** | ***9*** | ***10*** | ***11*** | ***12*** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Requirements management* | | | | | | | | | | | | | |
| Requirements gathering | | A | | B | C | | D | E | F | | | | |
| Requirements identification | | | A | | | B | | C | | D | | | E |
| Requirements organizing | | | A | | B | | C | | | | | | |
| *Release planning* | | | | | | | | | | | | | |
| Requirements priorization | | | | A | | B | C | D | | E | | | |
| Requirements selection | | | A | | | | | | B | C | D | | |
| Release definition | | | | A | B | C | | D | | E | | | |
| Release validation | | | | A | | | B | | | C | | D | |
| Launch preparation | | A | | | | B | | C | | | D | | |
| Scope change management | | | | A | | B | | | | C | | D | |
| *Product roadmapping* | | | | | | | | | | | | | |
| Theme identification | | | | | A | | B | | | | | | |
| Core asset identification | | | | | | A | | | B | | | | C |
| Roadmap construction | | | A | | | B | C | | D | E | F | | |
| *Portfolio management* | | | | | | | | | | | | | |
| Market trend identification | | | A | | B | | | | C | | | | |
| Partnering & contracting | | | A | | B | | C | | D | | | | |
| Product lifecycle management | | | A | | | B | | | C | | | | |
| Product line identification | | | | | | | A | | | B | | | |

**Fig. 2.** The maturity matrix for Software Product Management

The letters A to F represent the capabilities. Each focus area has its own unique capabilities and the amount of capabilities within a focus area varies from two (A-B) to six (A-F). Each capability has five attributes:

1. **Name.** A name describing the capability in a few words.
2. **Goal.** The goal describes what purpose the capability serves and it indicates the advantage of executing the capability.
3. **Action.** The action describes what must be done in order to meet the capability.
4. **Prerequisite(s).** Some capabilities require that one or more other capabilities be achieved first. This relation is described by listing all the capabilities that have to be implemented first.
5. **Reference(s).** This optional attribute describes related literature which can aid in understanding and implementing the capability, thus having a supporting role.

In Table 2, we elaborate on the capability attributes by providing an example. The capability that is used is Capability C of the focus area Requirements organizing.

**Table 2.** Capability C of the focus area Requirements Organizing (RO:C)

| Attribute | Content |
|---|---|
| Name | Requirement dependency linking |
| Goal | The existence of requirements interdependencies means that requirements interact with and affect each other. Requirement dependency linking prevents problems that result from these interdependencies, and therewith enables better planning of the development process. |
| Action | Dependencies between market and product requirements are determined and registered. A dependency exists when a requirement demands a specific action of another requirement. E.g. a requirement demands that another requirement be implemented too, or that another requirement is not implemented in case of conflicting requirements. The linkage can be supported by using advanced techniques, such as linguistic engineering. |
| Prerequisite(s) | Requirements Gathering: A (RG:A) |
| Reference(s) | Dahlstedt & Persson (2003) |

For a capability to be achieved it must be institutionalized and documented.

## 4 Mathematical Formalization

In order to provide rigorous fundamentals for focus area maturity models, we need to abstract the commonalities from the cases into a mathematical model. To introduce this model, we first have to define the fundamental concepts defining the maturity matrices. For convenience we will refer in the following way to the different types of matrices: the EA-matrix will refer to the DyA architecture maturity matrix, and the SPM-matrix will refer to the software product management maturity matrix.

Both types of matrices use the concept of focus area (the rows of the matrices) for which we introduce the set $F$ of focus areas. The number of focus areas within each matrix differs slightly: 18 for the EA-matrix and 16 for the SPM-matrix.

Another fundamental concept comes from the assessments organizations have to pass in order to reach a certain level for a specific focus area. We therefore introduce a totally ordered set $(L, \leq_L)$ of levels and since an assessment is specific for a pair consisting of a focus area and a level, we are interested in the Cartesian product $F \times L$. We abstract away from the 'assessment' and concentrate on the set $F \times L$. Since not every element of $F$ needs to have the same number of levels, this Cartesian product is in general a little bit too large. For the general definition of maturity matrix we allow subsets $C$ of $F \times L$. In the two example matrices, $C$ denotes the set of capabilities and the pairs $(f, l) \in C$ correspond to the cells in the matrix that are filled with a capital letter. The columns in the example matrices are the final concept we need and are formally described by a specific mapping $S$ from $C$ to the natural numbers. This puts us now in a position to give the following

**Definition**
A maturity matrix consists of

1. A triple $(F, (L, \leq_L), (C, \leq_C))$ where $F$ is a set, $(L, \leq_L)$ is a completely ordered set and $(C, \leq_C)$ is a partially ordered set with $C \subseteq F \times L$. Moreover, the ordering on $C$ respects the ordering on $L$ in the sense that if $c_1 = (f, l_1), c_2 = (f, l_2) \in C$ and $l_1 \leq_L l_2$ then $c_1 \leq_C c_2$.
2. An order preserving mapping $S: C \rightarrow \mathbb{N}$ with $\text{Im}(S) = \{1, ..., m\}$ for some $m \in \mathbb{N}$.

As an example take the SPM-matrix where $F$ is the set of 16 focus areas, $L$ = {A,B,…} is the set of 6 levels (so $F \times L$ consists of 96 elements), and $L$ is totally ordered in the obvious way (A < B <…). Furthermore, $C$ is the set of 63 capabilities, consisting of specific pairs $(f, l)$ where $f \in F$, $l \in L$ and $C$ is partially ordered by the intra- and inter-process capability dependencies, e.g. relations of the form (*f*,A) < (*f*,B) (intra-process) and relations of the form $(f_1, l_1) < (f_2, l_2)$ where $f_1 \neq f_2$ (inter-process). Finally, the mapping $S$ assigns every capability to one of the numbers 1 through 12 while preserving the order (so if $c_1 \leq_C c_2$, then $S(c_1) \leq S(c_2)$).

The maturity scale of an organization can now be defined. Since an organization that just started the development of a functional domain could very well have none of the capabilities defined for this domain, it makes sense to allow a zero scale. Even if they have acquired some capabilities of scale 1, but not all of them, we still define their scale as zero. Only if they have acquired all capabilities of scale 1 (i.e. all capabilities of the set $S^{-1}(1)$), then their scale will be 1 or higher.

In general, if the set of capabilities acquired by the organization is denoted by $C_A$ (a subset of $C$), then the *scale* of that organization is the maximum value $s$ for which $S^{-1}(\{1, ..., s\}) \subseteq C_A$. Note that if we substitute $s = 0$ the set $S^{-1}(\{1, ..., s\})$ $= S^{-1}(\varnothing) = \varnothing$ is a subset of $C_A$, so this definition also holds if $C_A$ is empty or if $C_A$ doesn't contain all capabilities with scale 1 (in both cases the maturity scale of the organization will be 0).

## 5 Developing a Focus Area Maturity Model

In this section we define a method for the development of focus area maturity models. We do so by drawing on literature review and by generalizing the lessons learned from the applications to the fields of enterprise architecture and software product management. First, we provide an overview of existing approaches on developing maturity models. From this we derive four generic phases in developing maturity models. As the existing approaches all deal with fixed-level maturity models, we need to elaborate the generic phases for developing focus area maturity models.

### 5.1 Existing Maturity Model Development Approaches

Several researchers have described methods or procedures on how to create maturity models. For example, De Bruin et al. have investigated several maturity models in different domains [1]. Based on their literature research, they identified six general phases that are part of the process of developing a maturity model: 1) Scope: deciding on the focus and the stakeholders; 2) Design: deciding on the architecture of the model, the type of audience and respondents of the model, and the method and driver of application; 3) Populate: identifying "what needs to be measured" and "how this can be measured". Domain components are identified and defined, and the method of measurement is determined; 4) Test: the construct of the maturity model and its instruments are tested for relevance and rigor; 5) Deploy: the model is being made available for usage and the model's generalizability is verified; 6) Maintain: some sort of repository is kept in order to support model evolution and development.

A more recent method is developed by Mettler and Rohner [2]. They propose "a first design proposition for situational maturity models". The design exemplar they describe in order to explain this design proposition consists of the following steps: 1) problem identification and motivation; 2) objectives of the solution; 3) design and development. This last phase is the most elaborated. First, a basic design for the maturity model is developed. Then, the maturity levels are identified and specified and the configuration parameters determined. Finally a proof of concept is delivered.

Another method is published by Becker et al. [9]. By proposing a number of requirements concerning the development of maturity models and comparing these to existing maturity models, they deduct a "generic and consolidated procedure model for the design of maturity models". The procedure model consists of the steps: 1) problem definition; 2) comparison of existing maturity models; 3) determination of development strategy; 4) iterative maturity model development; 5) conception of transfer and evaluation; 6) implementation of transfer data; and 7) evaluation.

Finally, Maier et al. [4] propose a "practitioner guidance" that supports developing and applying "maturity grids to assess organizational capabilities" [4]. They propose the following phases: 1) Planning: the aim, purpose, requirements, scope and target audience of the maturity model are identified; 2) Development: the different parts of the maturity model are defined, which are the process areas, the maturity levels, the cell descriptions, and the administration mechanism. In addition the role of the facilitator is elaborated here; 3) Evaluation: the model is validated, verified and, if necessary, iteratively refined; 4) Maintenance: changes on process areas and cell description must be properly evaluated and documented.

We can identify common elements in these development methods (table 3): a scoping phase in which purpose and scope of the maturity model are defined, the design of the model, followed by the development of the assessment instrument, and an implementation and exploitation phase in which the model is put to use and consequently exploited. Evaluation is not included as a common process phase as it is considered an integral part of each of the other phases.

**Table 3.** Maturity model development methods compared

| Common process phase | De Bruin et al. | Mettler and Rohner | Becker et al. | Maier et al. |
|---|---|---|---|---|
| **Scope** | Scope | Problem identification and motivation | Problem definition | Planning |
| | | Objectives of the solution | Comparison of existing maturity models | |
| **Design model** | Design | Design and development | Determination of development strategy | Development |
| | Populate - components | | Iterative maturity model development | |
| **Develop instrument** | Populate - measurements | | Conception of transfer and evaluation | |
| | Test | | | Evaluation |
| **Implement & Exploit** | Deploy | | Implementation of transfer data | Maintenance |
| | Maintain | | Evaluation | |

We recognize these phases also in developing a focus area maturity model. In the next section we detail these phases specifically for focus area maturity models.

## 5.2 Development Method for Focus Area Maturity Models

The method for developing focus area maturity models is depicted graphically in figure 3. We use the notation presented by [17], which is based on standard UML conventions, with some minor adjustments.

*Scope*

**Step 1: Identify and scope the functional domain.** A focus area maturity model can be developed for any functional domain. However, in order to develop a useful model, the domain must be scoped properly. This means deciding on what to include and exclude. In this phase it is also important to identify existing maturity models for the same or similar domains that may be used as a starting point for further development [9].

In DyAMM the functional domain is scoped to include all activities, responsibilities and actors involved in the development and application of enterprise architecture within the organization, where enterprise architecture is defined as a consistent set of rules and models that guide the design and implementation of processes, organizational structures, information flows, and technical infrastructure within an organization [18].

*Design model*

**Step 2: Determine focus areas.** Within the chosen domain, the focus areas must be identified, i.e. set $F$ in the formalization in section 4. In a relatively new field, literature review will provide a theoretical starting point which has to be followed by exploratory research methods like expert groups or case studies [1, 2, 4, 9]. A useful source for identifying focus areas are critical success factors found in previous

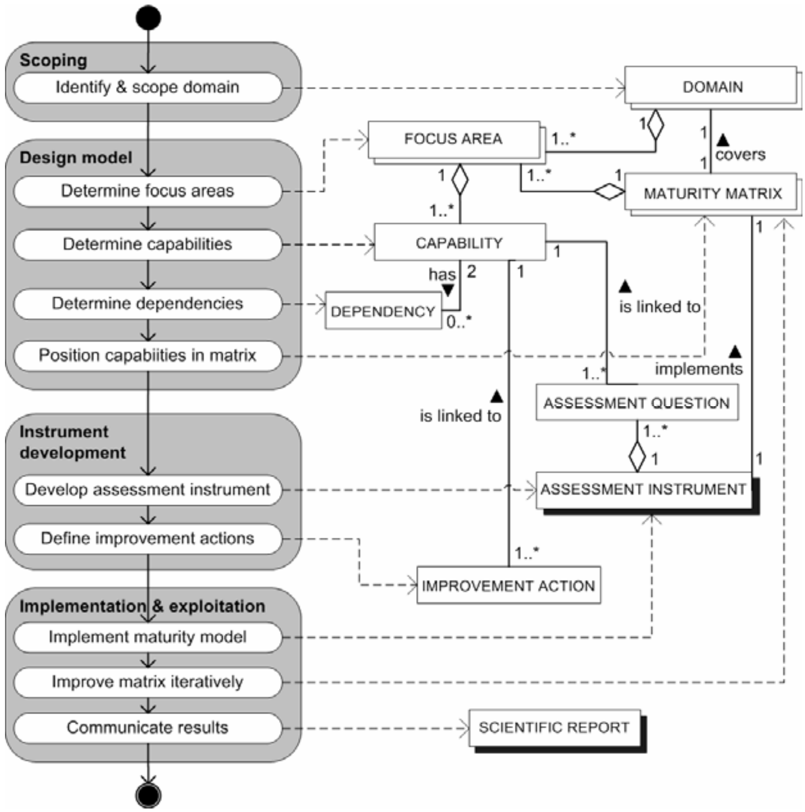



**Fig. 3.** The development method for focus area maturity models

research [1]. According to [4] a number of around 20 focus areas is on average a good number. It is important for means of validation to make explicit the underpinning conceptual framework used in defining the focus areas. Grouping the focus areas into a small number of categories may add to the accessibility of the model and is also a means of achieving completeness.

In the SPM matrix, the focus areas are deducted from the previously developed Reference Framework for Software Product Management [14]. This framework consists of the main internal and external stakeholders in the product management domain and of the main activities that are carried out by a product manager. These activities are directly transformed into focus areas for the maturity matrix. The focus areas are grouped into Requirements Management, Release Planning, Product Roadmapping and Portfolio Management. The number of focus areas defined is 16.

**Step 3: Determine capabilities.** Each focus area consists of a number of different capabilities representing progressive maturity levels. The definition of these capabilities depends on the underlying rationale of how the focus area can be incrementally developed in an evolutionary way [4]. Per focus area the evolutionary path of capabilities is defined. The definition of these capabilities is again based on literature review complemented with expert discussions. There are two ways of defining maturity levels: top down and bottom up [1]. In a relatively new field, the top down approach is more suitable. This implies first identifying the capabilities and then detailing them into descriptions of how these capabilities present themselves in practice. In this step the set *C*, a subset of the Cartesian product of *F* and *L*, is defined.

For the SPM matrix, the capabilities were identified in a brainstorming session with four SPM experts. After the brainstorming session, the results were compared with the existing SPM literature and, if necessary, refined or redefined. Finally, the capabilities were iteratively refined in expert interviews until agreement was reached.

**Step 4: Determine dependencies.** In this step dependencies between capabilities are identified, providing the partial ordering of *C*. This regards not only the dependencies between the capabilities of the same focus area, representing progressive maturity levels, but also the dependencies that may exist between capabilities of different focus areas.

The dependencies of the capabilities in the SPM matrix were identified by stating the prerequisite(s) per capability. Some capabilities required that one or more other capabilities, either of the same focus area or of another focus area, would be implemented first. An example is the Requirements prioritization capability. In order to be able to prioritize requirements, they need to be gathered first. The prerequisite for this capability is thus the capability Gather requirements. Consequently, there exists a dependency from Gather requirements to Prioritize requirements.

**Step 5: Position capabilities in matrix.** In this step the capabilities are positioned in the matrix. Capabilities that are dependent on other capabilities are always positioned further to the right. Capabilities that are not dependent on each other may be put in the same scale. For reasons of practicality, however, if many capabilities are contained in one scale, they may be assigned to a number of scales to get a more balanced matrix. This assignment is based on past experiences on preferences of implementation order. By this positioning the number of scales of the matrix is revealed, or, in mathematical terms, the mapping *S* from *C* to $\mathbb{N}$ is defined.

To position the SPM capabilities in the matrix, first an initial positioning was done based on the dependencies in the previous steps and on the experience of the researchers. Subsequently, the maturity matrix was validated with expert validation and a survey among 45 product managers and product management experts [16]. In this survey, participants were asked to position the different capabilities in the order they would implement them in their own organization. The result was a validated maturity matrix.

*Develop instrument*

**Step 6: Develop assessment instrument.** To be able to use a focus area maturity model as an instrument to assess the current maturity of a functional domain, measures must be defined for each of the capabilities. This can be done by formulating control questions for each capability. These questions can be combined in a questionnaire that can be used in assessments. Formulation of the questions is usually based on the descriptions of the capabilities and on experience and practices.

In the DyAMM each capability relates to 2 to 4 yes/no assessment questions. All questions associated with a capability must be answered with yes in order to claim achievement of that capability. The questions were based on the descriptions of the capabilities and on practices. The list of questions has been reviewed by experts.

**Step 7: Define improvement actions.** For each of the capabilities improvement actions can be defined to support practitioners in moving to that capability. These too, will usually be based on experience and practices. Improvement actions will in general be rather situation specific [4]. Therefore it is advisable to present them as suggestions, rather than as prescriptions. The improvement actions can also be used to provide situation specific application of the maturity model.

In the DyAMM improvement actions were identified for each capability by suggesting practices that may be implemented to realize the specific capability. These improvement actions are presented as examples meant to inspire rather than as prerequisites. An example is the improvement action to implement some form of account management within the architectural team to initiate a dialogue with business management, in order to achieve capability B, architectural processes geared to business goals, of focus area Alignment with business.

*Implement & exploit*

**Step 8: Implement maturity model.** Implementation can be done in various ways. A questionnaire can be distributed by electronic means which allows for collecting many assessments in a relatively short timeframe [1]. The assessment questions can also be answered by discussion in workshops or by holding interviews. This is especially appropriate when raising awareness is the aim [4]. The very first applications of the model can be used to evaluate the model.

The DyAMM was validated firstly by applying it in a few cases. This led to an adjustment of the model in a very early stage. After this adjustment the model was validated in a number of new cases [5]. This did not lead to further adjustments.

**Step 9: Improve matrix iteratively.** Once enough assessments have been collected, quantitative evaluation becomes possible. To evaluate how the model assists in incremental improvement interventions must be tracked longitudinally [1]. A repository must be kept to collect assessment results.

The DyAMM has been quantitatively validated after a repository of 56 assessments was collected. This led to a few adjustments in the assessment questions [12]. The effectiveness of the DyAMM is further illustrated by companies that have been using the model over the years to evolve their architecture practice and consequently established greater effectiveness of the practice.

**Step 10: Communicate results.** To further the field, the results of the design should be communicated to practitioners as well as to the scientific community.

The DyAMM has been communicated to the practitioners community by way of books, articles in professional journals and presentations on seminars. It has been communicated to the scientific community by presenting it on scientific conferences.

### 5.3 Evaluation

In developing the focus area maturity model development method we made use of previous research on developing fixed-level maturity models. From this research we derived four generic phases which we next elaborated for focus area maturity models on the basis of experience in developing focus area maturity models for the functional domains of enterprise architecture and software product management. We found that we could apply the generic phases retrospectively to these two cases and that we could describe both applications in terms of our development method. The mathematical formalization of the matrix helped us to clearly define its concepts. It also helped us in checking the completeness of the design phase, in the sense that all elements of the formalization are addressed. Further validation of the development method can be done by applying it to a new functional domain. This is yet to be done.

To provide a further initial evaluation of the development method presented here, we apply the requirements on the development of maturity models formulated by Becker et al. as presented in table 4 [9].

**Table 4.** Evaluation of the development method against the requirements by Becker et al.

| Requirement | Evaluation |
|---|---|
| R1 – comparison with existing maturity models | This is included as part of step 1. |
| R2 – Iterative Procedure | The determination of the focus areas and of the capabilities, as well as the positioning of the capabilities in the matrix is done iteratively, starting from literature followed by rounds of expert interviews and, possibly, surveys, until agreement is reached. |
| R3 – Evaluation | Evaluation is described as an integral part of each of the steps. The primary type of evaluation is by expert review and case study. |
| R4 – Multi-methodological Procedure | Literature review is combined with exploratory research methods. |
| R5 – Identification of Problem Relevance | The development of a focus area maturity model is especially relevant to functional domains that are still in the development stage in the majority of organizations. |
| R6 – Problem Definition | The problem definition is part of the identification and scoping of the functional domain in step 1. |
| R7 – Targeted Presentation of Results | Step 10 explicitly addresses the communication of results, both to practitioners and to the scientific community. |
| R8 – Scientific Documentation | Step 10 explicitly addresses the communication of results, both to practitioners and to the scientific community. |

We found that applying the design science research process of Peffers et al. helped us to fulfill most of the requirements as they can be recognized in the process step descriptions of the design science research process [7].

## 6 Conclusions and Further Research

In this paper we present a method for developing focus area maturity models. Focus area maturity models are especially suited to relatively new IS fields that require incremental, evolutionary capability development. The few focus area maturity models that have been in use up till now, show definite value in supporting organizations to incrementally improve their practices. Though many maturity models have been developed in the past few years, which is an indication of the need for maturity models, most of these are fixed-level and therefore less suited to incremental capability development. With our development method for focus area maturity models we hope to contribute to the design foundations and to further the research and practice of gradual improvement of functional domains.

The development method presented is based on both literature review in maturity model development and practical experience in applying the focus area maturity model concept to two distinct functional domains. The concept of the maturity matrix is refined by building a mathematical formalization of the matrix. This provided us with solid foundation for the focus area maturity model development method, in particular in ensuring the completeness of the design phase. In addition, our discussions about the mapping *S* which the formalization engendered, made explicit the distinction between hard and soft requirements on the order of implementation of the capabilities: on the one hand, the partial ordering on *C* reflects the hard requirements and poses restrictions on the order in which the capabilities *must* be implemented; on the other hand, there are also other reasons (primarily based on considerations of practicality) why one capability *should* be implemented before another one (the soft requirements). Both hard and soft requirements can be captured by putting capabilities in scales. A venue for further research is to investigate whether the soft requirements may be the basis for developing situation-specific variants.

The research approach taken is that of objective-centered design research where the development of an artifact is initiated by an industry or research need [7]. The resulting development method is evaluated against the requirements formulated by Becker et al [9]. Further evaluation by applying the method to other IS domains is still to be done.

A venue for further research is to elaborate on how situationality can be brought into the focus area maturity model development, enabling model developers to tune a focus area maturity model to a specific organization.

**Acknowledgments.** The authors wish to thank the experts that participated in the expert groups as well as the many organizations that have participated in projects for capability development using the DyAMM or SPM matrix.

## References

1. de Bruin, T., Freeze, R., Kulkarni, U., Rosemann, M.: Understanding the Main Phases of Developing a Maturity Assessment Model. In: Proceedings of the 16th Australasian Conference on Information Systems, Sydney (2005)
2. Mettler, T., Rohner, P.: Situational Maturity Models as Instrumental Artifacts for Organizational Design. In: Proceedings of the 4th international Conference on Design Science Research in information Systems and Technology, Philadelphia (2009)
3. CMMI:CMMI$^{SM}$ for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1) Staged Representation; CMU/SEI-2002-TR-012; ESC-TR-2002-012 (2002)
4. Maier, A.M., Moultrie, J., Clarkson, P.J.: Developing Maturity Grids for Assessing Organisational Capabilities: Practitioner Guidance. In: 4th International Conference on Management Consulting, Academy of Management (MCD 2009), Vienna (2009)
5. van Steenbergen, M., van den Berg, M., Brinkkemper, S.: A Balanced Approach to Developing the Enterprise Architecture Practice. In: Filipe, J., Cordeiro, J., Cardoso, J. (eds.) Enterprise Information Systems. LNBIP, vol. 12, pp. 240–253 (2007)
6. Koomen, T., Pol, M.: Test Process Improvement, a Practical Step-by-step Guide to Structured Testing. Addison-Wesley, Boston (1999)

7. 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), 45–78 (2008)
8. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Research in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
9. Becker, J., Knackstedt, R., Pöppelbuß, P.: Developing Maturity Models for IT Management - A Procedure Model and its Application. Business & Information Systems Engineering 1(3), 213–222 (2009)
10. van den Berg, M., van Steenbergen, M.: Building an Enterprise Architecture Practice. Springer, Dordrecht (2006)
11. Wagter, R., van den Berg, M., Luijpers, L., van Steenbergen, M.: Dynamic Enterprise Architecture: How to Make it Work. Wiley, Hoboken (2005)
12. van Steenbergen, M., Schipper, J., Bos, R., Brinkkemper, S.: The Dynamic Architecture Maturity Matrix: Instrument Analysis and Refinement. In: To appear in the Proceedings of the 4$^{th}$ Workshop on Trends in Enterprise Architecture Research, Stockholm (2009)
13. Xu, L., Brinkkemper, S.: Concepts of Product Software. European Journal of Information Systems 16(5), 531–541 (2007)
14. van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: Towards a Reference Framework for Software Product Management. In: Proceedings of the 14th International Requirements Engineering Conference, Minneapolis/St. Paul, Minnesota, USA, pp. 319–322 (2006)
15. Bekkers, W., Spruit, M., van de Weerd, I., Brinkkemper, S.: A Situational Assessment Method for Software Product Management. To appear in ECIS 2010 proceedings (2010)
16. van de Weerd, I., Bekkers, W., Brinkkemper, S.: Developing a Maturity Matrix for Software Product Management. In: Jansen, S., Tyrvainen, P. (eds.) Proceedings of the 1st International Conference on Software Business, Jyväskylä. LNBIP, vol. 51 (to appear 2010)
17. van de Weerd, I., Brinkkemper, S.: Meta-modeling for Situational Analysis and Design Methods. In: Syed, M.R., Syed, S.N. (eds.) Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, pp. 38–58. Idea Group Publishing, Hershey (2008)
18. van Steenbergen, M., Brinkkemper, S.: Modeling the Contribution of Enterprise Architecture Practice to the Achievement of Business Goals. In: Papadopoulos, G.A., Wojtkowski, W., Wojtkowski, W.G., Wrycza, S., Zupancic, J. (eds.) Information Systems Development: Towards a Service Provision Society. Springer, New York (2008)

# Applying Design Research Artifacts for Building Design Research Artifacts: A Process Model for Enterprise Architecture Planning

Stephan Aier and Bettina Gleichauf

Institute of Information Management
University of St. Gallen
Müller-Friedberg-Str. 8
CH-9000 St. Gallen
{stephan.aier,bettina.gleichauf}@unisg.ch

**Abstract.** Enterprise architecture (EA) describes the fundamental structure of an organization from business to IT. EA as a practice as well as a research topic has been around for several years. However, existing methods largely neglect the existence of time which is essential in order to systematically approach EA planning. The article at hand builds a process model for EA planning as a design research artifact. We therefore use another more general design research artifact – a method for process engineering – in order to systematically build our proposed planning process. From a design science research (DSR) perspective we demonstrate how elements of the DSR knowledge base can be applied to create new DSR artifacts and how DSR might build a toolbox as it is available in other mature engineering disciplines.

**Keywords:** enterprise architecture, planning, process design.

## 1 Introduction

Design science research (DSR) deals with the design of useful artifacts[1] [1, 5, 6] for relevant problems [3] as well as with research on DSR itself [3, 7, 8]. The primary goal of the paper at hand is to create a useful artifact and to demonstrate the execution of a DSR process while doing so. A secondary goal of the paper is to demonstrate the employment of a more general DSR artifact in a design research (DR) process which finally leads to a discussion of the necessity of a structured DSR body of knowledge as it is available in mature engineering disciplines.

Our artifact construction aims at the field of enterprise architecture (EA). The ANSI/IEEE Standard 1471-2000 defines architecture as "the fundamental organization of a system, embodied in its components, their relationships to each other and the

[1] In literature there is some agreement on the artifact types of constructs, models, and methods [1-3]. In some cases also instances are considered DSR artifacts while e.g. Gregor and Jones state "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" [4].

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 333–348, 2010.

environment, and the principles governing its design and evolution" [9]. Most authors agree that EA targets a *holistic scope* and therefore provides a broad and aggregate view of an entire corporation or government agency [10] covering strategic aspects, organizational structure, business processes, software and data, as well as IT infrastructure [11-14]. EA management can provide systematic support to organizational change that affects business structures as well as IT structures by providing constructional principles for designing the enterprise [15]. A suitable degree of *formalization* is needed in order to ensure traceable and repeatable results. Furthermore (semi) formalized models and sound methods are needed to enable division of labor among heterogeneous stakeholder groups [16, 17].

However, so far EA modeling has been restricted to modeling of (dateless) states of an enterprise by the majority of contributions [18, 19]. EA did not address the modeling of transformation and thus of the engineering process itself. Therefore this paper aims at answering the following research question: *Which activities should be performed for EA planning and how can these activities be structured in an EA planning process?* To answer this question our core contribution is an EA planning process which builds the foundation of an EA planning method. In order to systematically build this process we will apply another DR artifact – PROMET BPR [20]. PROMET BPR is a method for the (re-)design of business processes in general.

Scientific literature provides a number DSR procedure models [1, 3, 21, 22]. Specifically we follow the "identify a need–build–evaluate–learn and theorize" approach by Rossi and Sein [21]. We identify the need and define the objectives of our artifact by analyzing current practitioners' solutions and by analyzing related work (section 2). In section 3 we briefly introduce PROMET BPR and in section 4 we apply PROMET BPR for the construction our artifact. In section 5 we summarize an industry case which serves as an evaluation of our artifact. Finally in section 6 we reflect our solution and discuss the findings from a DSR perspective.

## 2 Identify the Need for EA Planning

In order to identify the need for a sound EA planning process we will discuss two industry cases for a practitioners' perspective first (section 2.1). Then we will discuss existing contributions from literature for an academics' perspective (section 2.2).

### 2.1 Practitioners' Needs

The following case studies exemplify how EA planning is currently done in industry practice. They also show which requirements arise from the current approaches and thus motivate the need for a method to systematically plan EA transformation.

#### 2.1.1 Company A

Company A provides IT outsourcing services and banking solutions. The primary product is a banking platform which is offered to private and universal banks. The company focuses on three main fields, namely application development, application management and operations, and therefore offers an integrated portfolio to its customers. The application development division is responsible for the development of the banking platform which is described by an application architecture model, an IT architecture model, and an operating model.

Major challenges within the architectural development plan are the coordination of activities of development teams and assurance that all dependencies are addressed and that milestones of integration and development activities are met simultaneously. If, for example, a component of an application needs an interface to a component of another application at a certain time for a certain milestone (e.g. test or release), it has to be assured that both components are available at that very point in time. This simple example grows very complex as the banking platform comprises over 200 applications, each consisting of a multitude of components that each has its own lifecycle as well as precursor and successor relationships.

The following questions are crucial to the architectural development process:

- How can the necessary changes for achieving the desired to-be state be identified?
- How can the necessary changes be coordinated?
- How can transformation be decomposed into project activities?
- How can the necessary development activities be bundled in order to be integrated into one release?
- Who should be responsible for which activity that has to be performed for answering these questions?

### 2.1.2 Company B

Company B is a globally operating bank based in Switzerland. During recent decades, mergers led to an increasing complexity of its application landscape. Architecture management is carried out by more than 90 architects and comprises architecture governance which is enforced in individual information system development projects. However, while IT architecture is strong in the bank's home country, the bank has to face challenges due to heterogeneous local solutions in almost every other country.

In order to enable better management of the heterogeneous application landscape, an EA project is currently being conducted. The project focuses on an integrated view on the different solutions the IT departments offer to the company's operating departments worldwide. Solutions, in this case, denote bundles of product components configured for a certain application scenario. The intended integrated view should also enable solution roadmap planning, i.e. the continuous development of the contained components. Therefore, the following questions need to be answered:

- Which projects affect which lifecycle planning of a certain solution?
- Does postponing of a project affect the lifecycle planning of a certain solution?
- Which requirements will be addressed in which project for which release?

### 2.1.3 Implications for Method Construction

Although the two case studies reveal different wordings, the central questions aim at similar challenges which need to be encountered. This leads to general requirements concerning EA planning. Hence, a comprehensive method that supports EA planning must respect the following requirements:

- EA models represent the changes of enterprise artifacts required by stakeholders. Therefore, at least (temporal) successor relationships and lifecycles need to be captured in one or several EA to-be models for one or several points in time.
- Since there will be alternative solutions in most cases, alternatives need to be evaluated.
- Actual project activities can be derived from EA to-be models.
- EA models reflect the interdependencies between the elements, i.e. the enterprise artifacts, as well as between their lifecycles.

In this paper, however, we focus on the process of EA planning (cf. 3.1)[2]. Such a process needs to respect the aforementioned model-related requirements, although it does not address them directly.

### 2.2 Scientific Need

Up to now only a few approaches for EA planning exist. While there are various contributions dealing with related questions, none of the existing approaches addresses EA planning from business to IT covering artifact relationships in semi-formal models and/or addressing model dynamics.

Historically, the topic evolved from strategic IS (information systems) planning which was firstly addressed by King in 1978 [24]. King proposes a process to design a management information system in accordance to the strategy of a corporation or government agency. As markets, organizational structures and system landscapes added more complexity to the matter of strategic planning and the alignment of business and IT, this approach as well as similar contributions were evolutionary refined. Strategic enterprise-wide information management [25] and more institutionalized IS planning processes became an issue in the 1990ies [26]. A prominent example for IS planning methods is IBM's Business System Planning (BSP) [27]. BSP aims to (re-) group IT functionalities according to data use and thereby identify application candidates with high integration intensity, but limited interfacing to other applications.

IS planning and EA planning differ in their approach, goal, and scope. While IS planning is rather technology driven and refers to the planning of systems (What IS do we need?), EA adopts a broader perspective and straightly integrates a business view on IS (What do we do now and what do we want to do? What information is needed to conduct our business in the future?) [28]. The availability of new architectural paradigms, such as service orientation, requires EA planning to focus on supplying information to stakeholders in order to support change projects. Therefore also in EA related approaches for planning were developed e.g. by Spewak (the wedding cake model) [28, 29], Pulkkinen and Hirvonen [30, 31], Op't Land et al. [32] and Niemann [33]. However, the majority of research results only focus on a unidirectional planning process that aims at improving the current structure, i.e. establishing a to-be architecture. The process of transforming the current architecture into the target architecture is only considered negligibly.

Recently the works of Buckl et al. [18, 34] and Aier et al. [19] address a comprehensive modeling approach for planning purposes in an EA context. While Buckl et al. propose a set of meta model requirements for modeling temporal aspects, their

[2] For a discussion of method vs. (process) model see [23].

proposal focuses on application landscapes [18]. However, they take into account important temporal dimensions, e.g. the time a model is created and the time a model should be valid for the past, the present and the future, as well as different variants of future models [18]. Recommendations or a framework addressing the planning process itself cannot be found in these approaches though. Aier et al. have analytically derived an EA planning process from possible planning information models [19], have analyzed existing planning processes in practice [35] and have specified necessary techniques in approaching a planning method [36]. The paper at hand will be based on this work and design a process which can be integrated into the aforementioned method (cf. section 3.3 on preliminary work).

## 3 Method Foundation for Building an EA Planning Process

### 3.1 Method Engineering

Design research aims at the design of problem-oriented artifacts like constructs, models, and methods. A method is considered as a systematic and goal-oriented procedure to solve certain problems [37, 38]. Thus, methods provide the basis for effective and efficient procedures while (re-)engineering enterprise components like business processes, information systems or IT infrastructure in a systemic way.

A method consists of design activities whose sequence is defined by a procedure model. The produced design results of the design activities are represented by an information model. Additionally, a method consists of roles which describe the participants involved in the design activities. At the same time the inclusion of roles introduces various perspectives different stakeholders may have on the design activities. Instructions on how design results are produced and documented are provided by adequate techniques [37].

We use a process design method in order to build a process model for EA planning. Following the distinction by Winter et al. [23], the process model is the design result of the process design method, and therefore represents a result-oriented view on a problem solution. At the same time the process model serves as a description of design activities to be performed within an EA planning method, as it is described by Aier and Gleichauf [36]. This method provides an activity-oriented view on the process model and uses it as its procedure model, adding roles and other activity recommendations on how to perform EA planning in a comprehensive manner. In other words, we use a DR artifact (process design method) in order to construct a process model which serves as a part (procedure model) of another DR artifact (method).

### 3.2 PROMET BPR

In order to design the EA planning process model we use the process design method PROMET BPR (PROject METhod Business Process Redesign) [20]. PROMET BPR has been designed as a DR artifact and is therefore thoroughly documented regarding its design activities, design results, techniques, roles and information model.

PROMET BPR targets a process design project as opposed to continuous process management. Process design involves the preliminary study, macro design, micro design, and implementation. PROMET BPR defines detailed activities and results for each phase which are documented in [20]. However, due to space limitations in this paper, we will focus on a limited number of activities and results that appear useful in the context of a generic EA planning process. Table 1 lists the design activities and results applied here. Those design results that are italicized are only outlined while actual result diagrams or tables are produced in chapter 4. Some activities and results can only be designed in a specific situation. Our approach, however, aims at the design of a generic EA planning process.

The preliminary study lays the foundation for the design project. An analysis of potential benefits helps to guide the whole design process. A first description of the process architecture, i.e. a process map, depicts the interdependencies of processes and provides insight into possible output and input relationships. When adapting process in a specific situation, a more detailed description of benefits is advisable. Furthermore, the identification of a process manager and/or process committee, as recommended by PROMET BPR, as well as the description of the involved business objects should be performed.

In order to align process design with customers' needs, an analysis of the customer relations and their involvement in the process should be performed at the beginning of the macro design phase. Second, a rough process vision will guide further design of the process. Therefore, general principles and basic conditions will be defined in the process framework. With regard to the necessary process management critical success factors of the process should be identified. After these preparative steps, the context of the process will be described, as a refinement of the process map from the preliminary study. This will help to identify required outputs and leads to the identification of the activities constituting the process. A list of involved IT applications completes the macro design phase.

The aims of the micro design phase are twofold: First, it aims at further refinement of the process description, i.e. the activity flows. Second, process management is prepared by refining critical success factors, performance indicators, process managers etc. As these tasks require in-depth information about specific situations of EA planning, we will not consider them in our application of PROMET BPR.

**Table 1.** PROMET BPR activities and results applied

| Activity | Result |
|---|---|
| **Preliminary Study** | |
| Design process architecture | *Process Map, Process List* |
| Analyze potential benefits | Benefits List |
| **Macro Design** | |
| Identify customer needs and processes | *Customer Relations Diagram* |
| Develop process vision | *Process Framework* |
| Plan process for critical success factors | *Critical Success Factors* |
| Check and define outputs | Contextual Diagram, Outputs List |
| Redesign process flow at macro level | Activity Chain Diagram (Macro), Activities List (Macro), *Index of Applications* |

Finally, the micro-designed processes are transposed from model into practical use in an implementation phase. As PROMET BPR focuses on the (re-)design, the implementation phase is neither subject to the core of the method nor to our application of it.

### 3.3 Preliminary EA Planning Process Models

PROMET BPR aims at redesigning business processes. Thereby it is assumed that the new design is influenced by existing business processes, though the method suggests workshop and brainstorming techniques in order to enable innovative ideas. In analogy, there already exist some ideas about an EA planning process which serve as a starting point as well as a source of inspiration for redesign.

In 2009 we conducted two workshops with experts from various companies active in the field of EA. Those experts include enterprise architects, chief architects, IT architects as well as representatives from business and strategy development fields who shared their experiences as well as their ideas concerning a comprehensive EA planning process. As a result, two proposals for a process model were developed which addressed the requirements expressed by the workshop participants.

Aier and Saat conducted six expert interviews and used them to construct an extended planning process based on approaches found in literature [35]. Their findings are also taken as a basis for the process redesign tasks in the paper at hand.

Finally, the process model is intended to be integrated in a comprehensive EA planning method which is described by Aier and Gleichauf [36]. In their article, they develop a list of design results, information model requirements as well as a preliminary list of design activities. The paper at hand aims at complementing this approach in terms of the design activities and process.

## 4 Construction of the Process Model for EA Planning

### 4.1 Preliminary Study Phase

Redesigning a process requires an estimation what potential benefits will be generated. Therefore, the first step in the preliminary study is a benefits analysis. It helps to clarify the goals of the redesign project and provides economic arguments for the project at the same time. In order to identify the processes that are relevant to the process that is to be redesigned, a process map is established in the preliminary study. The map depicts the main processes that exchange inputs/outputs with the process of interest. As the map is further refined in the macro design phase as a contextual diagram, we abstain from presenting the process map at this point.

*Benefits List.* In general, EA aims at enabling and ensuring transparency, consistency, flexibility and finally agility throughout the organization [39]. The EA planning process will ensure transparency and consistency by enabling coordinated further development of business and IT structures by using EA models. Moreover it will help to prioritize necessary development projects, i.e. to optimize budget and resource allocations and thereby realize global optima instead of local (project) optima. Being based on EA models and intertwined with EA management processes, the EA planning process will enforce the application of EA standards and principles in development projects. This will in turn improve the cost/benefit ratio of EA as the costs of information gathering and modeling will generate benefit for planning tasks: Applying the EA planning process will provide a consolidated information base for release management, project and program management.

## 4.2 Macro Design Phase

### 4.2.1 Customer Relations Diagram

PROMET BPR suggests that customer processes and customers' needs are identified in order to gain insight into the customer relations and enhance the value of outputs. While customers can easily be identified as extra-organizational customers in a provider-customer relationship, from a process' point of view also intra-organizational customers are relevant. These customers might more precisely be called stakeholder of the process. In the context of EA, stakeholder-oriented approaches have been presented e.g. by Kurpjuweit and Winter [40]. Stakeholders of the EA planning process might include the management body, enterprise architects or IT architects.

### 4.2.2 Process Framework

The cornerstones of the process on which the detailed design will be based are captured in the process framework. This includes assumptions about the process to be designed and about its surrounding processes. In the case of the EA planning process it is assumed that requirements as input for EA planning are provided by external processes, e.g. by requirements management. Further assumptions for the process framework are highly dependent from the actual situation of application.

### 4.2.3 Critical Success Factors

With regard to the necessary process management critical success factors of the process are identified. Ylimäki has presented a list of critical success factors for EA [41]. Among them, e.g. the overall maturity of the EA discipline and its penetration throughout the organization is critical. Especially more advanced EA processes like EA planning require a rather high EA maturity level.

### 4.2.4 Contextual Diagram

Before planning actual activities constituting the process a definition of the required outputs is necessary. Therefore, a contextual diagram describes input and output flows with other processes. This sets a basic framework for the process to be designed (figure 1).

The EA planning process is one of several EA management processes, complementing EA as-is documentation or maintenance processes as well as EA analysis processes. EA planning is driven by development requirements from the different sub-architectures. Therefore it receives input from processes in the fields of product portfolio management, business process management, application portfolio management, IT infrastructure management and the like. As the planning process is based on EA models, information about as-is models and condensed information from EA analysis is also needed. Besides that, strategic principles from business and IT strategy processes guide the EA planning process. Main outputs of the planning process are to-be models of future architecture states as well as project descriptions about development projects that will implement the planned enhancements. Those project proposals are accompanied by EA principles that are developed within the EA management processes and distributed into implementation processes via EA planning. As EA principles are main components of EA (cf. definition of EA in section 1) their

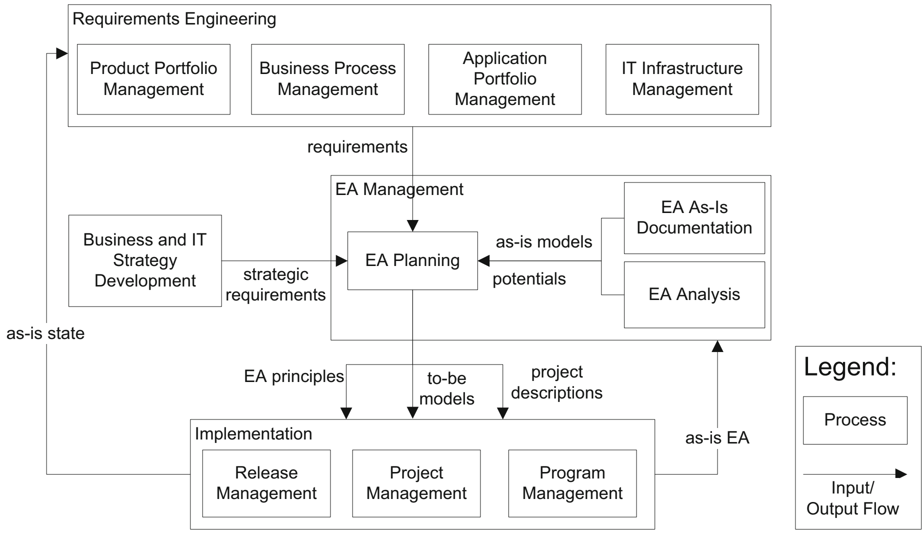



**Fig. 1.** Contextual diagram of the EA planning process

deduction and dissemination requires for adequate processes. However, in this paper this is not discussed in detail. The actual implementation is then done by release management, project management and program management processes. From there, information about the current state of EA is feed back to EA management and to sub-architectures' management processes.

### 4.2.5 Outputs List

As already indicated in the contextual diagram, outputs are exchanged between the EA planning process and other processes. In addition, intermediate outputs between the activities within the EA planning process itself are relevant in order to identify process activities. The main output of the EA planning process are project proposals that roughly describe which development tasks are necessary (cf. [36]). These proposals might also include an assessment of which tasks are more relevant than others and which should be performed in conjunction or in parallel. As intermediate outputs the EA planning process will generate a new EA vision (or update an existing one) and one or several EA roadmaps. Most likely the planning process will generate alternative to-be models of the EA, so an assessment framework as well as the assessed to-be models themselves will be outputs of the process. As described in the EA planning method [36], the process will in detail produce a list of model elements to be changed, a list of interdependencies between model elements within and between models as well as a list of predecessor and successor relationships of the model elements.

As already mentioned in the contextual diagram, EA standards and principles are developed and their compliance is controlled by EA management processes. EA planning in particular gathers and adapts EA principles in order to transfer them to implementation processes.

### 4.2.6 Activities

The main step of the macro design phase is flow planning which takes results of the contextual diagram as well as the list of outputs and derives the actual tasks involved in the process, their sequence and who performs them. The activity chain diagram (figure 2) depicts the activity flow. A detailed description including involved roles is presented in the list of activities below (table 2).

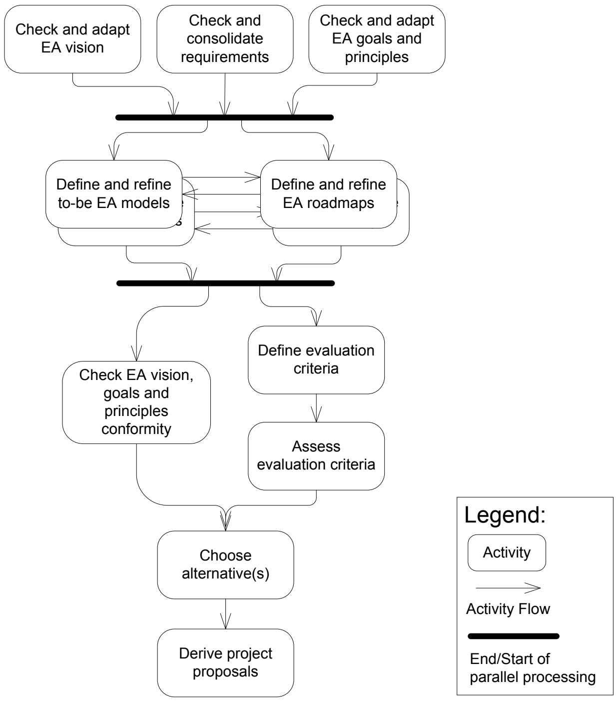



**Fig. 2.** Activity Chain Diagram

**Table 2.** Activities List

| Activity | Description | Roles |
|---|---|---|
| Check and adapt EA vision | The EA vision is defined in the EA management processes and/or in the business and IT strategy development. EA planning uses it as an input and guideline for its planning activities. | Enterprise architect |
| Check and consolidate requirements | Requirements are gathered in the various partial architectures, such as IT infrastructure or product development processes. EA planning uses them as an input. | Business and IT architects |
| Check and adapt EA goals and principles | EA goals and principles are defined in the EA management processes. EA planning uses them as an input and guideline for its planning activities. | Enterprise architect, business |

**Table 2.** (*Continued*)

| Activity | Description | Roles |
|---|---|---|
| Define and refine to-be EA models | Snapshots of future states of the enterprise architecture are established (partial architecture snapshots might be consolidated). Alternatives are depicted and aligned with roadmap plans.* | Enterprise architect, business and IT architects |
| Define and refine EA roadmaps | Alternative EA roadmaps are depicted and aligned with to-be models.* | Enterprise architect, business and IT architects |
| Define evaluation criteria | In order to assess alternatives and guide a decision process, evaluation criteria are defined. | Enterprise architect |
| Check EA vision, goals and principles conformity | Chosen alternatives need to adhere to EA vision, goals and principles. Therefore, alternatives are checked against their conformity. | Enterprise architect |
| Assess evaluation criteria. | Assessment of alternatives may e.g. include the consideration of costs, benefits and risks. | Enterprise architect, management (sponsor) |
| Choose alternative(s) | Choose the favored alternative on the basis of evaluation criteria. | Enterprise architect, management (sponsor) |
| Derive project proposals | One or more project proposals are derived. It might be useful to have alternative project proposals for different future scenarios. | Enterprise architect, management (sponsor) |

* Depending on the actual situation, planning activities start from roadmaps or to-be models respectively.

## 5 Evaluation

As already mentioned above, preliminary considerations about an EA planning process were incorporated in the application of process (re-)design using PROMET BPR. Among those considerations practical experiences at company A, which has already been presented in chapter 2, were accounted for. At the same time, the actual application of the process model at the company serves as an evaluation.

The planning activities at the company aim at the further development of their core product, a banking platform, which consists of various applications, interfaces and middleware components. As the company also provides the hosting of the platform, also hardware issues are important.

Requirements that guide the further development of the banking platform are gathered in two ways. On the one hand, high level guidelines for the general development vision of the platform are established by a superordinate planning team. These guidelines correspond to general EA goals and principles that are continuously adapted and updated. On the other hand, requirements for the development of the individual components and sub-architectures are collected separately.

Partial architecture roadmaps are defined following the general EA guidelines that are grouped by functional domains. The roadmaps consist of models representing snapshots of the desired architecture for up to three points in time in the future, taking into account existing vendor specific constraints if applicable. Individual component development plans are then integrated into these roadmaps. The coordination between high level guidelines, partial roadmaps and detailed to-be models is ensured by an organizational structure that defines roles of lead architects being responsible for the partial architectures (business, applications, IT, operations, and security) as well as roles of domain architects who are responsible for functional domains. While domain architects control the development and realization of high level roadmaps, lead architects drive and ensure the further development of components taking detailed restrictions into account. In joint workshops all parties consolidate their plans. This also improves the conformity with EA goals and principles. Possible alternatives are discussed and evaluated in these workshops.

In order to determine a sequence of tangible development activities general milestones and components to be developed are defined. Afterwards, the development phases of the elements' lifecycles are planned in detail, i.e. in terms of specification and testing phases. Upon those specifications a rough project program schedule can be defined. Finally, on the basis of the project outlines actual project planning is enabled. For that purpose, the project proposals can be enriched with information about costs, quality metrics, staff, risks, and resources.

The application at company A shows that the presented EA planning process is indeed useful in practice. Of course, a lot of more details need to be added to the process model in order to be actually implemented. For example, the process customers' needs play an important role regarding the coordination of possible alternative roadmaps or to-be models. At company A process customers are e.g. the teams of the lead architects which are affected by EA principles. By incorporating them into the planning activities, their needs are addressed. At the same time, these organizational structures constitute an important critical success factor for the EA planning process.

## 6 Discussion and Outlook

The aim of our research has been to build a process for EA planning as a DR artifact. In order to systematically build this process we have used the PROMET BPR method for process (re-) design. Derived from desired design results we have described the necessary design activities and linked these activities in the EA planning process. The resulting process uses alternative to-be models that represent different points in time and enables the evaluation of alternatives as well as the derivation of project activities. Thus it respects the requirements derived in section 2.1.3.

In contrast to a typical application of PROMET BPR our planning process stays on a more generic level and is not build for a specific situation. However, the application of the process in an industry case demonstrates how this generic process can be further detailed. This leads to the question of situational methods [42, 43]: The distinction of a limited number of well defined situations described by certain project types and contingency factors [44] could guide a configuration of a more detailed planning process. While some contributions [45, 46] have made first steps in describing and identifying situations for EA management we did not yet comprehend the relationships of EA management goals, EA realization approaches, enterprise engineering goals, and planning approaches. Thus our approach still lacks the concept of situation.

Although the application of our process illustrates the usefulness of our artifact in this specific situation, we could not show the superiority of our approach. This is due the lack of concepts for scientific progress of DR artifacts. While Aier and Fischer

[47] propose criteria for assessing the progressivity of a design research artifact their criteria are still heavily dependent from the specific purpose and scope of the artifact and thus from the situation of application.

Finally we demonstrate in our paper how DR artifacts (e.g. methods) can be employed in order to build new – more specific but still applicable in a number of situations – DR artifacts. This leads to the question, whether design research (as an engineering approach [3]) needs a more structured catalogue of design knowledge. While Gericke [48] and Vaishnavi and Kuechler [49] discuss this question on a meta level under the term DSR patterns, Shaw [50] presents an analysis of mature engineering disciplines. Among other points she finds that mature engineering disciplines (e.g. mechanical engineering) all provide a commonly accepted, well structured knowledge base (e.g. [51]) that provides guidance especially for more routine like engineering tasks. Process design or the assignment of organizational responsibilities to certain tasks in a process represents such routine like engineering tasks. One of the major obstacles for providing such a knowledge base, however, is the form of documentation for these mostly complex DR artifacts. First approaches to this issue may be found in the work of Frank [52] who proposes a structure for description of artifacts.

## References

1. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)
2. vom Brocke, J., Buddendick, C.: Reusable Conceptual Models – Requirements Based on the Design Science Research Paradigm. In: Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST), pp. 576–604 (2006)
3. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
4. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems 8, 312–335 (2007)
5. Venable, J.: The Role of Theory and Theorising in Design Science Research. In: Proceedings of the 1st International Conference on Design Science in Information Systems and Technology (DESRIST 2006), pp. 1–18. Claremont Graduate University, Claremont (2006)
6. 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)
7. Cross, N.: Designerly Ways of Knowing: Design Discipline Versus Design Science. Design Issues 17, 49–55 (2001)
8. Winter, R.: Design Science Research in Europe. European Journal of Information Systems 17, 470–475 (2008)
9. IEEE: IEEE Recommended Practice for Architectural Description of Software Intensive Systems (IEEE Std 1471-2000). IEEE Computer Society, New York (2000)
10. Tyler, D.F., Cathcart, T.P.: A structured Method for Developing Agile Enterprise Architectures. In: International Conference on Agile Manufacturing (ICAM 2006), Norfolk, Virginia, USA, pp. 1–8 (2006)

11. Winter, R., Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. Journal of Enterprise Architecture 3, 7–18 (2007)
12. Jonkers, H., Lankhorst, M.M., Doest, H.W.L., Arbab, F., Bosma, H., Wieringa, R.J.: Enterprise architecture: Management tool and blueprint for the organisation. Information Systems Frontiers 8, 63–66 (2006)
13. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin (2005)
14. Schelp, J., Winter, R.: Language Communities in Enterprise Architecture Research. In: Diversity in Design Science – Proceedings of the 4th Conference on Design Science Research in Information Systems and Technologies (DESRIST 2009), Philadelphia, PA, USA, May 7–9, pp. 1–10. ACM, Philadelphia (2009)
15. Lindström, L., Johnson, P., Johansson, E., Ekstedt, M., Simonsson, M.: A survey on CIO concerns – Do enterprise architecture frameworks support them? Information Systems Frontiers 8, 81–90 (2006)
16. Frank, U.: Perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages. In: 35th Hawaii International Conference on System Sciences (2002)
17. Jonkers, H., Lankhorst, M., van Buuren, R., Hoppenbrouwers, S., Bonsangue, M., van der Torre, L.: Concepts for Modelling Enterprise Architectures. International Journal of Cooperative Information Systems 13, 257–287 (2004)
18. Buckl, S., Ernst, A.M., Matthes, F., Schweda, C.M.: An Information Model for Managed Application Landscape Evolution. Journal of Enterprise Architecture 5, 12–26 (2009)
19. Aier, S., Gleichauf, B., Saat, J., Winter, R.: Complexity Levels of Representing Dynamics in EA Planning. In: Proceedings of the 5th International Workshop on Cooperation & Interoperability – Architecture & Ontology (CIAO! 2009), pp. 55–69. Springer, Amsterdam (2009)
20. IMG: PROMET BPR: Method for Business Process Redesign, Release 2.0. Information Management Group/Institute of Information Management, University of St. Gallen, St. Gallen (1997)
21. Rossi, M., Sein, M.K. (eds.): Design Research Workshop: A Proactive Research Approach. IRIS Association, Haikko (2003)
22. 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 (DESRIST), Claremont, CA, pp. 83–106 (2006)
23. Winter, R., Gericke, A., Bucher, T.: Method versus Model – Two Sides of the Same Coin? In: Advances in Enterprise Engineering III: 5th International Workshop, CIAO! 2009, and 5th International Workshop, EOMAS 2009, held at CAiSE 2009, pp. 1–15. Springer, Amsterdam (2009)
24. King, W.: Strategic Planning for Management Information Systems. MIS Quarterly 2, 27–37 (1978)
25. Targowski, A.: The architecture and planning of enterprise-wide information management systems. Idea Group Publishing, Harrisburg (1990)
26. Eliot, L.B.: Information systems strategic planning. Computer Technology Research Corporation, Charleston (1991)
27. IBM: Business Systems Planning: Information Systems Planning Guide, vol. 1. IBM, Atlanta (1984)
28. Spewak, S.H., Hill, S.C.: Enterprise Architecture Planning – Developing a Blueprint for Data, Applications and Technology. John Wiley & Sons, New York (1993)

29. Spewak, S.H., Tiemann, M.: Updating the Enterprise Architecture Planning Model. Journal of Enterprise Architecture 2, 11–19 (2006)
30. Pulkkinen, M., Hirvonen, A.: EA Planning, Development and Management Process for Agile Enterprise Development. In: Proceedings of the 38th Annual Hawaii International Conference on Systems Sciences (HICSS-38), p. 223.223 (221–210). IEEE Computer Society, Los Alamitos (2005)
31. Pulkkinen, M.: Systemic Management of Architectural Decisions in Enterprise Architecture Planning. Four Dimensions and Three Abstraction Levels. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS-39), pp. 179a (171–179). IEEE Computer Society Press, Honolulu (2006)
32. Op 't Land, M., Proper, E., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture – Creating Value by Informed Governance. Springer, Berlin (2009)
33. Niemann, K.D.: From Enterprise Architecture to IT Governance. Elements of Effective IT Management. Vieweg, Wiesbaden (2006)
34. Buckl, S., Ernst, A., Matthes, F., Schweda, C.M.: An Information Model for Landscape Management – Discussing Temporality Aspects. In: Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research (TEAR 2008), Sydney (2008)
35. Aier, S., Saat, J.: Understanding Processes for Model–based Enterprise Transformation Planning. International Journal of Internet and Enterprise Management 7 (2010)
36. Aier, S., Gleichauf, B.: Application of Enterprise Models for Engineering Enterprise Transformation. Enterprise Modelling And Information Systems Architectures forthcoming (2010)
37. Braun, C., Wortmann, F., Hafner, M., Winter, R.: Method Construction – A Core Approach to Organizational Engineering. In: Applied Computing 2005, Proceedings of the 2005 ACM Symposion on Applied Computing, pp. 1295–1299. ACM Press, New York (2005)
38. Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. Information and Software Technology 38, 275–280 (1996)
39. Dietz, J.L.G.: Enterprise Ontology – Theory and Methodology. Springer, Heidelberg (2006)
40. Kurpjuweit, S., Winter, R.: Concern-oriented Business Architecture Engineering. In: Proceedings of the 24th Annual ACM Symposium on Applied Computing (SAC), pp. 265–272. Honolulu, Hawaii (2009)
41. Ylimäki, T.: Potential Critical Success Factors for Enterprise Architecture. Journal of Enterprise Architecture 2, 29–40 (2006)
42. Harmsen, A.F., Brinkkemper, S., Oei, H.: Situational Method Engineering for Information System Project Approaches. In: Methods and Associated Tools for the Information Systems Life Cycle, pp. 169–194. North-Holland, Amsterdam (1994)
43. Karlsson, F., Ågerfalk, P.J., Hjalmarsson, A.: Method Configuration with Development Tracks and Generic Project Types. In: 6th CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in System Analysis and Design (EMMSAD 2001), Interlaken, Switzerland (2001)
44. 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. Springer, Boston (2007)
45. Aier, S., Riege, C., Winter, R.: Classification of Enterprise Architecture Scenarios – An Exploratory Analysis. Enterprise Modelling and Information Systems Architectures 3, 14–23 (2008)

46. Leppänen, M., Valtonen, K., Pulkkinen, M.: Towards a Contingency Framework for Engineering an Enterprise Architecture Planning Method. In: Proceedings of the 30th Information Systems Research Seminar in Scandinavia (IRIS 2007), Tampere, Finland (2007)
47. Aier, S., Fischer, C.: Scientific Progress of Design Research Artefacts. In: Proceedings of the 17th European Conference On Information Systems (2009)
48. Gericke, A.: Analysis of Design Science Research Patterns from an Engineering Perspective. In: Proceedings of the IADIS International Conference Information Systems 2009 (IS 2009), pp. 64–72 (2009)
49. Vaishnavi, V.K., Kuechler, W.: Design Science Research Methods and Patterns: Innovating Information and Communication Technology. Auerbach Publications, New York (2007)
50. Shaw, M.: Prospects for an Engineering Discipline of Software. Software, IEEE 7, 15–24 (1990)
51. Dubbel, H., Kuttner, K.H., Beitz, W.: Dubbel. Handbook of Mechanical Engineering. Springer, Berlin (1994)
52. Frank, U.: Towards a Pluralistic Conception of Research Methods in Information Systems Research. Institut für Informatik und Wirtschaftsinformatik (ICB), Universität Duisburg-Essen, Essen (2006)

# Realization Approaches of Customer Relationship Management – A Design Science Research Contribution to Support the Construction of Situational CRM Artifacts

Anke Gericke[1,3] and Tobias Bucher[2,3]

[1] Boschetsrieder-Str. 61a, 81379 Munich, Germany
anke.gericke@alumni.unisg.ch
[2] Goethestr. 14, 68161 Mannheim, Germany
tobias.bucher@alumni.unisg.ch
[3] Institute of Information Management, University of St. Gallen, Mueller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland

**Abstract.** In this paper, customer relationship management (CRM) is focused as one particular field of application for design science research (DSR). In managerial practice, it can be observed that CRM is approached differently by diverse organizations; moreover, the need for adaptable ("situational") models and methods in support of CRM (so-called "CRM artifacts") is postulated both in scientific theory and in practice. The paper at hand aims at contributing to this field by reporting on the results of an exploratory analysis. Based on our empirical evidence, we propose to differentiate between four distinct realization approaches of CRM: (1) initial CRM; (2) IT-focused CRM; (3) mature CRM; (4) selective CRM. Each of these approaches is characterized by a unique vector of realization degrees with respect to four CRM design factors and represents an exemplary, generalized way of how organizations deal with CRM. These insights are particularly useful for the construction of situational CRM artifacts within DSR.

**Keywords:** Customer Relationship Management, Design Science Research, Contingency Approach, Situational Framework.

## 1 Introduction

Design science research (DSR) is a problem solving paradigm focusing on the development of useful artifacts with which information systems (IS) problems can be solved [23]. The artifact types of March & Smith [23], i.e. constructs, models, methods and instantiations, have been established as artifacts of the DSR discipline (e.g. cf. [15; 32; 34]). For the development of such artifacts contingency approaches – also called situational approaches – become more and more important (e.g. cf. [5; 34]). Situational construction approaches allow for the development of situational artifacts that are developed for a class of problems but can be adapted to a certain problem situation which the artifact user is confronted with.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 349–364, 2010.

DSR can be applied to multiple fields of the IS discipline, such as enterprise resource planning (ERP), business process management (BPM), knowledge management, customer relationship management (CRM), etc. In this paper we put our focus on CRM as one particular field of DSR application.

Also in the field of CRM it has been recognized that "[d]ifferent organizations are approaching CRM in different ways" [25, p. 84] and the request for adaptable models and methods that support the establishment and development of CRM has been formulated (e.g. cf. [36]). However, this aspect has hardly been addressed in literature yet. Within a brief literature analysis no CRM artifact could be identified which addresses the aspect of being adaptable to a certain problem situation. That is why this paper aims at a contribution to close this research gap by answering the following research question:

> *How can existing CRM approaches be differentiated, i.e. which distinct CRM realization approaches are pursued by real-world organizations?*

In order to answer this research question we conduct an empirical exploratory analysis. Exploratory analyses are used especially if the established knowledge in respect of the object under investigation is scarce. This is the case for the concept of situativity in the field of CRM. Our contribution to answer the above mentioned research question results in a situational framework for CRM. With the help of such a situational framework it will be possible to characterize CRM approaches implemented in real-world organizations. This will be the basis for the development of new or the adaptation of existing CRM artifacts which take these situational specifics into account and better support the users in solving their problems.

The paper is structured as follows: In the next section, the theoretical foundation is laid. First, the concept of situativity is explained in more detail. Second, the results of our brief literature review are presented by giving an overview about artifacts already developed in the field of CRM. This is followed by the main part of our paper: We explain the research design and the results of our exploratory analysis. Furthermore, we present the interpretation of these results by introducing the situational framework for CRM we derived. Finally, we summarize our findings and present some issues for further research.

## 2 Theoretical Foundation

### 2.1 The Concept of Situativity within DSR

Within DSR, contingency or situational construction approaches have gained more and more importance in recent years. This can be recognized in particular within the research areas of method engineering and reference modeling (see e.g. [27] and [33; 34]). Method engineering aims at the development of construction processes for the development of situational methods. Reference modeling proposes construction principles for the design of reusable conceptual models.

The increased importance of situational construction approaches can be attributed to the high complexity within the field of DSR. Furthermore, it has been realized that artifacts which are used within different problem situations without being adapted to the specific problem characteristics often do not lead to a successful outcome. This aspect is not new in scientific theory and has been taken up by organizational theory first (see e.g. [8; 19]). Following Fiedler [8] there is no "best way" of managing an organization. Instead, the effectiveness of an organization is influenced by various internal and external factors. Thus, the organizational design must be contingent upon those factors and solutions have to be developed which address the specific characteristics of an individual problem situation (see also [14]) – so called situational solutions or situational artifacts. For the development of such situational artifacts the description of the problem situation and the identification of those situational factors which influence the application of the artifact or which are influenced by the artifact itself is necessary [26].

Although construction processes for the development of situational artifacts have been proposed in DSR – especially in method engineering (see e.g. [2; 17; 28]) and in reference modeling (see e.g. [34]) – there are hardly any contributions that deal with the description of different problem situations and the identification of situational factors [37]. For those contributions that exist in literature, two different types of describing a problem situation can be differentiated: (1) Approaches belonging to the first group present pre-defined situational factors such as "importance of the project", "number of stakeholders" or "technology used" (see e.g. [21]). (2) The second group comprises contributions that do not suggest standardized, pre-defined situational factors but that describe specific concepts that can be used to identify situational factors for each and every problem situation individually (see e.g. [3; 24]). In order to identify situational characteristics, (exploratory) empirical analyses can be used. This technique has already been applied successfully to other fields of DSR (see e.g. [4]); we will therefore revert to it in this paper, too.

## 2.2 Review of Related Customer Relationship Management Research

A good long-term customer relationship is seen as the key to an organization's success [25]. Originally driven by software vendors, the concept of customer relationship management (CRM) has evolved into a management approach which helps organizations to manage their customer interactions more effectively and build profitable, long-term relationships with their customers [20; 22]. Next to conceptual and strategic aspects, CRM comprises a set of processes and enabling information systems in order to implement its management approach in an organization [22; 30][1].

In DSR, CRM processes and CRM information systems are of special interest because DSR artifacts can support the implementation of these CRM aspects. Thus, we want to characterize CRM processes and CRM information systems first and complete our review of related CRM research with the introduction of CRM artifacts. In doing so, we also want to analyze whether these artifacts already incorporate situational factors or not.

In literature, the distinction between the following CRM processes has become generally accepted (see e.g. [9; 25; 31]):

[1] More detailed information about the term CRM and its development can be found in [38].

- *Operational CRM processes:* Those processes which have direct contact to the customer include campaign management, sales management, service management, complaint management and customer loyalty management [9; 30]. Furthermore, they collect data for analytical CRM processes. In turn, findings from these analyses are passed on to the operational CRM processes in order to improve their effectiveness [30].
- *Analytical CRM processes:* Those processes analyze customer information that was collected in other CRM processes [30]. Statistical methods are used to analyze e.g. customer segments, behavior and value [25]. Analytical CRM processes may include lead management, customer segmentation, customer evaluations/profiling etc. [9].
- *Collaborative CRM process:* This process is mainly understood as multi-channel management which concentrates on the customer interaction using a coordinated mix of interaction channels [9; 25]. Geib [9] goes one step further and puts forward "management CRM processes" instead of "collaborative CRM process": In his opinion "management CRM processes" include multi-channel management, CRM strategy development and CRM process controlling.

Equivalent to the distinction of CRM processes, CRM information systems can be separated into (see e.g. [6; 30]):

- *Operational CRM systems:* These systems support operational CRM processes and improve their efficiency. They include solutions for campaigns/marketing, sales and service automation. [30]
- *Analytical CRM systems:* These systems store and analyze customer information which was captured from operational CRM. Such systems include, but are not limited to, data warehouses, data marts or data mining systems. [6; 30]
- *Collaborative CRM systems:* These systems manage all customer interactions and embrace all costumer "touch points" (communication channels such as telephone, fax, e-mail, internet, etc.). [6; 30]

In the context of CRM processes and CRM information systems, several DSR artifacts have been developed. In the following, we will present such CRM artifacts which we identified in our review of CRM research. Next to artifacts that address the design of CRM processes or the development of CRM information systems we also identified other artifacts – mostly methods – that support the implementation or rollout of CRM solutions in organizations. The literature review does not claim to be exhaustive.

With respect to CRM processes, Reichold [29] develops best practices in terms of reusable conceptual models. Her focus is on analytical CRM processes but she also develops process models for operational CRM and for the management of CRM, i.e. for the process of CRM strategy development in particular. In her contribution situational aspects are not considered. In the contribution of Berger et al. [1], best practices for CRM processes in retail banking are developed. First, a process landscape is presented – a model which contains CRM core processes in retail banking. Second, for one of the core processes of the process landscape – the channel assignment – a detailed process description in the form of a reusable conceptual model is presented. For both artifacts situational characteristics have not been taken into consideration.

Regarding CRM information systems, DSR artifacts in terms of reusable conceptual models have been developed as well. The first artifact identified has been presented by Cunningham and Song [7] who propose a model with which the design of CRM data warehouses is supported. In their model situational factors are not taken into account. On the contrary, Geib [9] does not focus on the architecture of a singe system, but presents IS architectures for analytical CRM systems, operational CRM systems and collaborative CRM systems. Also in this contribution situational characteristics are not included into the developed artifacts.

Geib et al. do not only suggest best practices for application architectures; instead they also derive procedure steps which should support the establishment of CRM in business networks in the financial service sector [10]: Their procedure consists of six challenges that have to be addressed while establishing CRM. The authors also identify one situational aspect which is derived from five case studies that build the basis for their contribution. It was recognized that there are differences between intra-organizational settings (universal banks) and inter-organizational settings (financial services alliances) with respect to the degree of standardization and the role of data privacy. Unfortunately, it is not becoming perfectly clear how this situational factor and the resulting differences have been identified. Furthermore, these aspects are hardly addressed in the developed procedure steps for the CRM implementation. Another CRM methodology is developed by Colomo Palacios et al. [6]. Their methodology integrates aspects of a CRM customization approach and aspects of a systems development method of the European Space Agency. Again, situational aspects are not taken into account. Mishra and Mishra [25] also provide some guidance on the implementation of CRM solutions by presenting the results of a case study. Even though the authors realize that "[d]ifferent organizations are approaching CRM in different ways" [25, p. 84] their contribution is limited to the implementation steps of this single case study. Finally, Zablah et al. [38] present an implementation method consisting of five steps which should support the successful implementation of CRM solutions. Here as well, no situational characteristics are taken into consideration.

This small review gives an impression about the state-of-the-art with regard to CRM artifacts and situational characteristics. Although it is not a complete literature review, it nevertheless shows that although the demand for situational CRM artifacts has been recognized, hardly any CRM artifact that takes situational aspects into consideration has been developed so far.

## 3 Exploratory Analysis

### 3.1 Research Design and Description of the Data Set

Data for the exploratory analysis was collected by means of a questionnaire that was distributed at the DW 2008 "Synergies through integration and information logistics" conference held in Switzerland in late 2008. The questionnaire was designed to assess design factors of CRM. For this purpose, several statements were formulated, and the respondents were asked to indicate the current realization degree of each statement within their organization. To this end, a five-tiered Likert scale was provided. Before distributing the questionnaire at the conference, a pre-test was conducted by both academics and practitioners resulting in a revision of the questionnaire in terms of completeness and comprehensibility. During the conference the questionnaire was briefly explained to the conference participants.

A total of 61 questionnaires were returned. After the elimination of incomplete questionnaires, 53 valid observations remained that entered the analysis. The interviewed organizations are primarily large and medium-sized companies (50.9 % have more than 1,000 employees, and another 18.9 % have more than 200 employees) from Switzerland, Germany and Austria. The main industry sectors represented in the sample are banking, financial services and insurance, software and technology as well as telecommunication.

Amongst questions relating to demographic characteristics as well as to customers and distribution channels, the questionnaire contains 32 items describing the design of CRM in an organization. These variables can be grouped into seven categories:

- *Development and organizational establishment of CRM*: The maturity of a CRM approach is influenced by its organizational establishment and anchoring. Thus, the data set comprises e.g. information on the extent to which the CRM strategy is derived from the business strategy, which different analyses are used to control CRM activities and whether or not a dedicated CRM department exists.
- *Customer evaluation/profiling techniques*: The identification and calculation of customer evaluations/profiles is an important part of each CRM approach which can be realized with the help of qualitative or quantitative techniques. Furthermore, companies conduct customer segmentations in order to identify different customer groups/clusters that possess distinct requirements regarding products and services. These customer segmentations are proposed once and/or updated regularly. Due to the importance of customer evaluations/profiles, the data set covers the presented variables.
- *Use of customer evaluations/profiles*: Based on the fact that customer evaluations/profiles can be used for different reasons, four items (e.g. the identification of value-creating or value-destroying customers, an instrument to monitor and forecast the achievement of business objectives or to control whole customer portfolios) relating to these reasons were collected in the data set.
- *Campaign management*: CRM approaches incorporate different operational processes, campaign management being one of them [9, p. 35]. If companies carry out campaigns they can choose from individual and segment-specific campaigns. In addition to these variables, the data set comprises information on the extent to which a selection of a target group as well as the evaluation of a campaign is conducted.
- *Sales and service management*: With the help of various CRM activities, sales and service management can be supported actively, especially by conducting lead management, submitting individual product and service offers to customers, submitting offers to customers that contain individual price conditions, including customer evaluations in service management, introducing complaints management, and by supplying the continuous improvement process with customer complaints. Therefore, the data set covers these variables.

- *Customer loyalty*: Next to campaign, sales and service management, customer loyalty management is another operational CRM process [9, p. 35]. A total of four items relating to customer loyalty (e.g. use of a bonus/loyalty program, collection of information about customers who drift away, modeling of customer drifts and establishment of counter measures) were collected.
- *CRM process outsourcing and IT support for CRM*: Finally, statements regarding the outsourcing of CRM processes such as new client acquisition, customer loyalty management and customer win-back management, and the use of a dedicated IT system to support CRM activities were proposed.

Using this data set as a basis, we conducted an exploratory analysis in order to attain scientific knowledge regarding the design factors and realization approaches of CRM that are observable in practice. At first, a factor analysis was conducted in order to identify different CRM design factors. Building upon the calculated factor values, we then classified the 53 observations with the help of a cluster analysis algorithm. The clustering was aimed at the identification of distinct realization approaches of CRM. In contrast to hypothesis-testing analyses, an exploratory, i.e. hypothesis-generating, approach is used when research relating to the topic in question is scarce or non-existent. This is the case for design factors and realization approaches of CRM.

### 3.2 Course of Analysis

Design factors of CRM were identified by means of principal component analysis. This is a statistical technique that is generally applied for the extraction of a small number of mutually independent factors from a larger number of indicator variables. The question: "How can all the indicator variables loading on a particular factor be summarized by means of a collective term?" is tried to be answered based on the analysis results [13].

Principal component analysis was performed using a reduced data set, covering 21 indicator variables from the questionnaire. The measure of sampling adequacy ("Kaiser-Meyer-Olkin criterion") for this reduced data set is equal to 0.886; this is an outstanding value for a socio-scientific study. The Kaiser-Meyer-Olkin criterion represents the extent to which the input values of a factor analysis belong together; it therefore allows for the determination whether or not this statistical technique can reasonably be performed or not. Kaiser and Rice apprise a value of at least 0.8 as "meritorious" and a value of 0.9 or more as "marvelous" [16]. Consequently, the underlying data set can reasonable be considered to be highly appropriate for the application of a factor analysis technique.

Four factors were extracted by means of principal component analysis; these four factors jointly explain about 71.6% of the total variance. Both the Kaiser criterion and the scree plot point to this four-factor-solution. The resulting component matrix was rotated using the Varimax method with Kaiser normalization in order to enhance and ease the interpretation of the items' assignment to factors. Split up into the four design factors of CRM as identified in our study, the rotated component matrix and the corresponding factor loadings are shown in Table 1through Table 4. An item is normally assigned to a factor if its factor loading amounts to a value of at least 0.5 [13]. As an add-on to this general rule, we assigned an item to the factor on which it loads highest in the event that this particular item shows loadings above threshold value with respect to two or more factors.

The four design factors of CRM can be interpreted as follows:

- *Factor 1: "Coverage of customer process".* Six items were found to have significant impact on the first design factor of CRM. Their common denominator is that they relate to the single steps of the customer process – from the management of customer acquisition to customer service, customer care, customer retention and finally, if need be, to the drifting away of customers. Thus, this first design factor of CRM describes the degree to which the customer process is covered by adequate measures and procedures; it thus reflects whether or not the customer relationship is managed in a well-defined, structured way within a given organizational context. The higher the value of this design factor for an organization, the higher is the coverage of the customer process through CRM.

**Table 1.** Results of Factor Analysis – Factor 1 ("Coverage of Customer Process")

| Item Description | | Factor 1 | Factor 2 | Factor 3 | Factor 4 |
|---|---|---|---|---|---|
| 1.1 | Lead management activities are conducted, i.e. customer requests are consolidated, reviewed and prioritized. | **0.634** | 0.116 | 0.460 | 0.397 |
| 1.2 | Based on the CRM activities, individual price conditions are offered to the customers. | **0.756** | 0.282 | 0.204 | 0.009 |
| 1.3 | CRM-related information (e.g. customer evaluation) is used for service management. | **0.741** | 0.223 | 0.233 | 0.186 |
| 1.4 | Complaints management is an integrated part of the corporate CRM approach. | **0.667** | 0.227 | 0.118 | 0.205 |
| 1.5 | Information from customer complaints flows into a continuous improvement process. | **0.757** | 0.212 | 0.035 | 0.270 |
| 1.6 | Information about customers migrating to competitors is collected and analyzed (e.g. point in time; reasons; new supplier). | **0.636** | 0.228 | 0.315 | 0.127 |

- *Factor 2: "Usage of CRM instruments".* The second factor is also made up by six items; these items account for the variety of tools and instruments that are used to manage the customer relationship. Among these instruments, qualitative and quantitative methods for the determination of customer evaluations, techniques for customer segmentation and procedures for the controlling and monitoring of the behavior of customers/customer groups are distinguished. Thus, this design factor provides information on the "method portfolio" of CRM that is applied within a given organization. This portfolio is the more extensive, the higher the value of this design factor is.
- *Factor 3: "Professionalizing of CRM".* Five items show high loadings on the third factor. These items relate to various organizational as well as managerial aspects of CRM, such as the level of experience with respect to CRM, the strategic anchoring of CRM, the design of CRM processes, the organizational embedding of the CRM initiatives as well as the IT support of CRM. Thus, this third design factor represents the level of professionalism of the CRM program within a given organization; this characteristic is directly linked to the magnitude of this factor.

**Table 2.** Results of Factor Analysis – Factor 2 ("Usage of CRM Instruments")

| Item Description | | Factor 1 | Factor 2 | Factor 3 | Factor 4 |
|---|---|---|---|---|---|
| 2.1 | Qualitative methods are used to calculate customer evaluations (e.g. analysis of customer satisfaction). | 0.172 | **0.797** | 0.115 | 0.138 |
| 2.2 | Quantitative methods are applied to determine customer evaluations (e.g. customer lifetime value; customer break-even analysis). | 0.256 | **0.716** | 0.331 | 0.138 |
| 2.3 | The existing customer segmentation is evaluated and adjusted on a regular basis. | 0.407 | **0.649** | 0.303 | 0.143 |
| 2.4 | Customer evaluations are used to identify value-creating customers. | 0.437 | **0.584** | 0.362 | 0.184 |
| 2.5 | Customer evaluations are executed to monitor and to forecast the achievement of business objectives. | 0.569 | **0.633** | 0.070 | 0.175 |
| 2.6 | Not only single customer relationships are controlled but also the whole customer portfolio is monitored (e.g. portfolio segmentation; size of the segments). | 0.294 | **0.644** | 0.173 | 0.351 |

**Table 3.** Results of Factor Analysis – Factor 3 ("Professionalizing of CRM")

| Item Description | | Factor 1 | Factor 2 | Factor 3 | Factor 4 |
|---|---|---|---|---|---|
| 3.1 | CRM has been in use for many years. | 0.165 | 0.253 | **0.814** | 0.259 |
| 3.2 | The CRM strategy is based upon the business strategy. | 0.364 | 0.302 | **0.657** | 0.286 |
| 3.3 | Analyses are conducted and reports are generated in order to supervise CRM activities. | 0.282 | 0.453 | **0.654** | 0.120 |
| 3.4 | CRM is conducted by a dedicated CRM department. | -0.031 | 0.520 | **0.666** | -0.030 |
| 3.5 | An IT system is used to support the CRM activities. | 0.331 | -0.022 | **0.710** | 0.305 |

- *Factor 4: "Application of campaigns"*. Finally, four items were found to have significant impact on the fourth factor. Campaigns, i.e. the realization of target group-orientated marketing actions, are the common ground of these items. Both the general use of campaigns and their basic principles and design – be it individual or segment-/group-specific campaigns; be it based on previous marketing activities or not – are covered by this factor. Thus, the fourth design factor of CRM informs about the prevalence and relevance of campaigns within a given organizational context. The higher the value of this design factor, the higher is the importance of campaigns.

**Table 4.** Results of Factor Analysis – Factor 4 (“Application of Campaigns”)

| Item Description | | Factor 1 | Factor 2 | Factor 3 | Factor 4 |
|---|---|---|---|---|---|
| 4.1 | Campaigns are conducted, i.e. target group-orientated marketing actions, that make use of different media and channels of distribution. | -0.019 | 0.345 | 0.118 | **0.851** |
| 4.2 | Individual campaigns are very important. | 0.409 | -0.062 | 0.284 | **0.701** |
| 4.3 | Segment-specific campaigns are very important. | 0.293 | 0.195 | 0.194 | **0.818** |
| 4.4 | Past campaigns are evaluated in order to learn for future campaigns. | 0.444 | 0.303 | 0.280 | **0.569** |

To identify distinct realization approaches of CRM, cluster analysis was applied on the data set, using the calculated factor values of the four previously identified design factors as input data. We decided to use the Ward algorithm [35] and the squared Euclidean distance as fusion algorithm and distance measure, respectively.

The Ward algorithm is among the most popular fusion algorithms in cluster analysis; it is widely accepted and has been used extensively in various disciplines. The algorithm determines categories by minimizing the clusters’ variances [11; 35]; it therefore tends to assign observations to less populated clusters while calculating the hierarchy and often results in relatively homogeneous groupings.

In the course of the present exploratory analysis, we have experimented with a variety of algorithms in order to solve the clustering problem at hand. It has turned out that – for our purposes – the Ward algorithm yields the “best” results in terms of producing a consistent and well-interpretable partitioning of the underlying observations. Many authors, e.g. Kaufman and Rousseeuw [18] as well as Han and Kamber [12], suggest that experimenting with different algorithms and choosing the one which yields clustering results that “best” suit the data set and the purpose of the analysis at hand is a valid and generally accepted procedure: “The choice of clustering algorithm depends both on the type of data available and on the particular purpose of the application. If cluster analysis is used as a descriptive or exploratory tool, it is possible to try several algorithms on the same data to see what the data may disclose.” [12, p. 400]

For the present analysis, the dendrogram heuristic suggests that grouping the observations, i.e. the surveyed organizations, into four clusters is the most reasonable solution. These four clusters represent four distinct realization approaches of CRM. Table 5 exhibits the standardized arithmetic means of the four calculated factor values for each of the four clusters. Depending on these standardized values, we distinguish between low parameters (smaller than -0.5; dark grey shading), medium parameters (between -0.5 and +0.5; medium grey shading) and high parameters (greater than +0.5; light grey shading). Based on this information as well as on a more accurate and comprehensive analysis of the clustering results, the four realization approaches of CRM will be discussed in detail in the next sub-section of this paper.

**Table 5.** Results of Cluster Analysis

| Cluster Description | Factor 1 | Factor 2 | Factor 3 | Factor 4 |
|---|---|---|---|---|
| ***Cluster 1:*** "Initial Customer Relationship Management" | -0.755 | -0.075 | -1.370 | 0.149 |
| ***Cluster 2:*** "IT-Focused Customer Relationship Management" | -0.785 | -1.379 | 0.983 | 0.640 |
| ***Cluster 3:*** "Mature Customer Relationship Management" | 1.330 | 0.592 | -0.007 | 0.668 |
| ***Cluster 4:*** "Selective Customer Relationship Management" | 0.210 | 0.862 | 0.395 | -1.457 |

### 3.3 Interpretation of the Analysis Results

Fig. 1 illustrates the standardized arithmetic means of each of the 21 indicator variables for each of the four clusters, grouped in accordance with the four factors identified by the factor analysis. These profile lines detail the information depicted in Table 5; they allow for an analysis on the level of each indicator variable rather than on the factor values representing these indicator values.

From both the detailed profile line illustration and the summary numerical representation of the cluster analysis results, it can be seen that there is a rather obvious, significant partitioning between clusters 1 and 3 ("initial CRM" versus "mature CRM"). These two clusters cover the vast majority of all observations that entered the exploratory analysis; 42 out of 53 organizations are grouped into either one of these two clusters. Moreover, these two clusters appear to be very different: the organizations grouped into cluster 1 show rather low parameter values with respect to the indicator variables whereas those represented by cluster 3 exhibit rather high degrees of performance. Clusters 2 ("IT-focused CRM") and 4 ("selective CRM") are somewhat "in the middle"; their characteristics are distributed more heterogeneously.

Taking all this information into account, the four different types of how organizations deal with customer relationship management (i.e. the four distinct realization approaches of CRM) can be described and interpreted as follows:

- *Cluster 1:* A total of 20 organizations were combined in the first cluster. These organizations show particularly low parameter values with respect to design factors 1 and 3, i.e. with respect to the coverage of the customer process as well as regarding the professionalizing of CRM. As to the design factors 2 and 4 ("usage of CRM instruments" and "application of campaigns", respectively), at the most medium parameter values are displayed. This can be interpreted as a sign for a first, rather crude and therefore yet immature approach to CRM that is pursued by these organizations: Some CRM tools and techniques (e.g. complaints management (item 1.4); quantitative measures for the determination of customer evaluations (item 2.2); monitoring/forecasting of the achievement of business objectives by means of customer evaluations (item 2.5); campaigns, especially segment-specific ones (items 4.1 and 4.3)) are already applied; the professionalizing of CRM in the organization is, however, still very poor and the coverage of the customer process is yet incomplete. We propose to refer to this particular realization approach as *“initial customer relationship management”*.

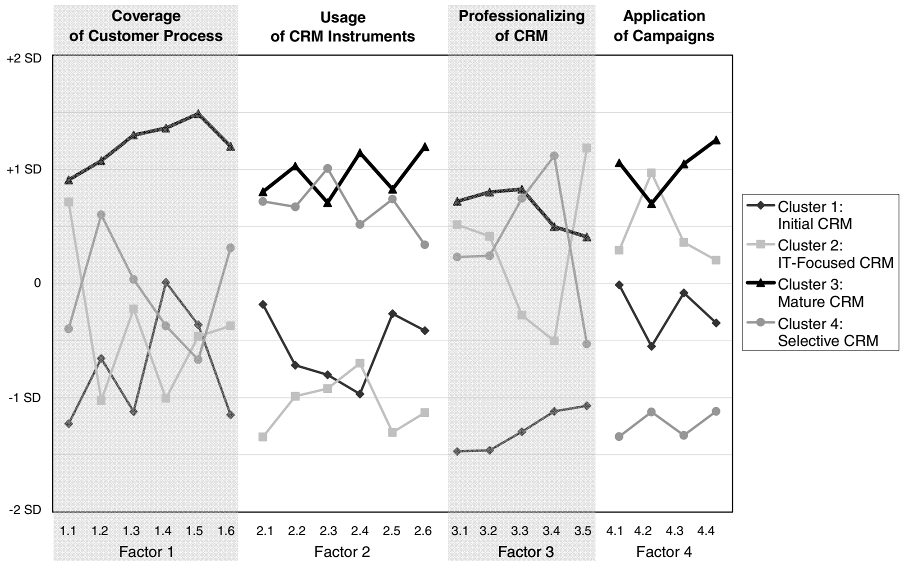



**Fig. 1.** Profile Lines for the Four Clusters

- *Cluster 2:* The second cluster merges a total of six organizations. These organizations are characterized by a rather low realization of design factors 1 and 2, that is, both the coverage of the customer process and the usage of CRM instruments are developed quite sparsely. In contrast, design factors 3 (“professionalizing of CRM”) and 4 (“application of campaigns”) are in high gear. Most outstanding are the assessments regarding items 3.5 (usage of an IT system to support CRM), 4.2 (application of individual campaigns) and 1.1 (conduction of lead management activities). Therefore, we propose that the organizations grouped into this cluster follow an *“IT-focused customer relationship management”* approach.
- *Cluster 3:* Those 22 organizations that were grouped into the third cluster exhibit – in contrast to those in cluster 1 – rather high parameter values with respect to the design factors of CRM. The coverage of the customer process (design factor 1) is complete; the distance to the other realization approaches becomes most obvious when looking at this criterion. The same is true for the CRM tools and techniques portfolio (design factors 2 and 4); here, those organizations grouped into the third cluster share the top positions with those organizations represented by clusters 2 (regarding the application of campaigns) and 4 (with respect to the usage of CRM instruments), though. Factor 3, the professionalizing of CRM, scores somewhat lower in the average (both clusters 2 and 4 are above). This is, however, due only to some “outlier” assessments: cluster 2 scores extremely high with respect to item

3.5 (usage of an IT system to support CRM); cluster 4 shows a very high valuation with regard to item 3.4 (CRM is conducted by a dedicated CRM team). On the average, cluster 3 exhibits the most homogeneous high assessment regarding the professionalizing of CRM. We therefore denote this realization approach as *"mature customer relationship management"*.

- *Cluster 4:* Five organizations were grouped into the fourth cluster. This cluster is characterized by very heterogeneous assessments, ranging from high appraisals (factor 2, "usage of CRM instruments") to very low appraisals (factor 4, "application of campaigns"). The coverage of the customer process (design factor 1) and the professionalizing of CRM (design factor 3) are somewhere in the middle range. The organizations grouped into this cluster share a medium to high maturity of CRM; this is not least shown by high assessment of item 3.4 (CRM is conducted by a dedicated CRM team). However, the other item scores are very heterogeneous; this might point towards the fact that the organizations represented in this cluster follow a rather individual CRM approach. Thus, we propose that this realization approach should be designated as *"selective customer relationship management"*.

## 4 Summary and Outlook

Based on literature as well as on practical experience, we have shown that there is a strong need for situation-specific approaches to CRM. DSR can contribute to this field through the development of situational CRM artifacts such as situational models or methods for the implementation and advancement of CRM in real-world organizations.

To enable and support the construction as well as the application of such artifacts, we have derived a situational framework for CRM based on the results of an exploratory survey. This situational framework differentiates between four distinct realization approaches of CRM which are based on four essential characteristics. These characteristics are referred to as CRM design factors and are assumed to have significant influence on the design of the CRM approach that is adopted by a given organization. Based on our empirical evidence, it seems reasonable to infer that the design of an organization's CRM approach is, among other things, dependent on those factors and therefore also on the underlying items.

As a result of the findings that are laid out in the present contribution, there are at least three major opportunities for further research:

- *Examination, discussion and validation of the proposed situational CRM framework.* The first research opportunity consists in the careful, thorough validation of both the design factors and the realization approaches of CRM that have been derived in this article. The need for more detailed analyses relates especially to the completeness and consistency of the CRM design factors on the one hand and to the stability as well as the persistence of the CRM realization approaches on the other hand. This could be accomplished either quantitatively by means of an extended, representative empirical study that is designed to test the hypothesized situational CRM framework, qualitatively by use of case study and/or action research approaches, theory-driven based on an extensive literature research of by means of a so-called mixed method approach (triangulation of research results).

- *Proposition of rules on how to use the situational CRM framework.* Based on this validated (and potentially extended) situational CRM framework, heuristics should be developed that link the CRM realization approaches to CRM artifacts. These rules describe the implications for the adaptation of CRM artifacts when a given set of situational characteristics is present.
- *Engineering of situational CRM artifacts.* Finally, the third opportunity for further research pertains to the engineering of situational CRM artifacts – especially models and methods – that are directed at supporting the implementation and advancement of CRM. This can be accomplished either by enhancing existing CRM artifacts (cf. e.g. section 2.2 of this paper) to meet the requirements of situational design, or by designing new situational CRM artifacts from scratch.

The contribution of the present article would be marginal if it neither stimulated nor facilitated the engineering of appropriate situational CRM artifacts, taking the findings of our empirical study into account. Correspondingly, the implications for managerial practice from the research at hand are at least threefold:

- *Assess the current CRM realization approach of your organization.* Based on the situational framework presented in this paper, organizations have the possibility to assess their current positioning and their approach to CRM.
- *Identify potential fields of activity to advance your CRM approach.* Moreover, organizations have a tool to analyze their particular approach, to compare it to others, and thus to identify potential measures to modify and adapt the current approach in order to become more effective and efficient in the field of CRM.
- *Make use of the situational CRM artifacts resulting from further DSR.* This transition from one CRM approach to another should be supported through the application of adequate situational CRM artifacts. Managerial practice should make use of corresponding DSR results in order to implement and/or advance customer relationship management.

## References


1. Berger, S.C., Gensler, S., Skiera, B., Wiesel, T.: Customer Relationship Management Processes in Retail Banking – Best Practice and Business Value Evaluation. In: Proceedings of the 42nd Hawaii International Conference on System Sciences, Hawaii, pp. 1–10 (2009)
2. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38(4), 275–280 (1996)
3. Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R.: Situational Method Engineering – On the Differentiation of "Context" and "Project Type". In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering – Fundamentals and Experiences, pp. 33–48. Springer, Boston (2007)
4. Bucher, T., Winter, R.: Classification of Business Process Management Approaches – An Exploratory Analysis. BIT – Banking and Information Technology 7(3), 9–20 (2006)
5. Bucher, T., Winter, R.: Dissemination and Importance of the "Method" Artifact in the Context of Design Research for Information Systems. In: Vaishnavi, V.K., Baskerville, R. (eds.) Proceedings of the Third International Conference on Design Science Research in Information Systems and Technology (DESRIST 2008), Atlanta, pp. 39–59 (2008)

6. Colomo Palacios, R., Gómez Berbís, J.M., García Crespo, Á.: An Integrated Methodology for Customer Relationship Management Customization. Journal of Information Systems and Technology Management 4(3), 287–300 (2007)
7. Cunningham, C., Song, I.-Y.: Designing Data Warehouses to Support Customer Relationship Management Analyses. In: Poster at the IST Research Day 2007, Drexel University, College of Information Science and Technology (2007), `http://144.118.25.24/bitstream/1860/1605/1/2007021030.pdf`
8. Fiedler, F.E.: A Contingency Model of Leadership Effectiveness. Advances in Experimental Social Psychology 1, 149–190 (1964)
9. Geib, M.: Kooperatives Customer Relationship Management – Fallstudien und Informationssystemarchitektur in Finanzdienstleistungsnetzwerken. Deutscher Universitäts-Verlag, Wiesbaden (2006)
10. Geib, M., Kolbe, L.M., Brenner, W.: Customer Relationship Management in Business Networks: Lessons from the Financial Services Industry in Germany and Switzerland. MIS Quarterly Executive 4(1), 247–260 (2005)
11. Hair Jr., J.F., Anderson, R.E., Tatham, R.L., Black, W.C.: Multivariate Data Analysis. Prentice Hall, Upper Saddle River (1998)
12. Han, J., Kamber, M.: Data Mining – Concepts and Techniques. Morgan Kaufmann, San Francisco (2006)
13. Härdle, W., Simar, L.: Applied Multivariate Statistical Analysis. Springer, Berlin (2003)
14. Hauswirth, I.A.: Effective and Efficient Organisations? – Government Export Promotion in Germany and the UK from an Organisational Economics Perspective. Physica, Heidelberg (2006)
15. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
16. Kaiser, H.F., Rice, J.: Little Jiffy, Mark IV. Educational and Psychological Measurement 34(1), 111–117 (1974)
17. Karlsson, F., Ågerfalk, P.J.: Method configuration: adapting to situational characteristics while creating reusable assets. Information and Software Technology 46, 619–633 (2004)
18. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data – An Introduction to Cluster Analysis. John Wiley & Sons, New York (1990)
19. Kieser, A., Kubicek, H.: Organisation. de Gruyter, Berlin (1992)
20. King, S.F., Burgess, T.F.: Understanding success and failure in customer relationship management. Industrial Marketing Management 37(4), 421–431 (2008)
21. Kornyshova, E., Deneckère, R., Salinesi, C.: Method Chunks Selection by Multicriteria Techniques: an Extension of the Assembly-based Approach. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering – Fundamentals and Experiences, pp. 64–78. Springer, Boston (2007)
22. Ling, R., Yen, D.C.: Customer relationship management: An analysis framework and implementation strategies. The Journal of Computer Information Systems (April 2001), `http://www.allbusiness.com/technology/computer-software-customer-relation/975920-1.html`
23. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)
24. Mirbel, I., Ralyté, J.: Situational method engineering: combining assembly-based and roadmap-driven approaches. Requirements Engineering 11(1), 58–78 (2006)
25. Mishra, A., Mishra, D.: Customer Relationship Management: Implementation Process Perspective. Acta Polytechnica Hungarica 6(4), 83–99 (2009)

26. Olle, W.T., Hagelstein, J., Macdonald, I.G., Rolland, C., Sol, H.G., van Assche, F.J.M., Verrijn-Stuart, A.A.: Information Systems Methodologies – A Framework for Understanding. Addison-Wesley, Wokingham (1988)
27. Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.): Situational Method Engineering – Fundamentals and Experiences. Springer/International Federation for Information Processing, Boston (2007)
28. Ralyté, J., Rolland, C.: An Approach for Method Reengineering. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 471–484. Springer, Heidelberg (2001)
29. Reichold, A.: Prozesse des Analytischen CRM – Fallbeispiele aus der Finanzdienstleistungsbranche, Architekturvorschlag und Methodenelemente. PhD thesis. Universität St. Gallen, St. Gallen (2006)
30. Reichold, A., Kolbe, L., Brenner, W.: Performance Measurement of CRM in Financial Services. In: Proceedings of the Fourth International Conference on Performance Measurement and Management, Edingburgh, pp. 859–866 (2004)
31. Teo, T.S.H., Devadoss, P., Pan, S.L.: Towards a holistic perspective of customer relationship management (CRM) implementation: A case study of the Housing and Development Board, Singapore. Decision Support Systems 42(3), 1613–1627 (2006)
32. Vahidov, R.: Design Researcher's IS Artifact: a Representational Framework. In: Chatterjee, S., Hevner, A.R. (eds.) Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, pp. 19–33 (2006)
33. vom Brocke, J.: Design Principles for Reference Modelling. Reusing Information Models by Means of Aggregation, Specialisation, Instantiation, and Analogy. In: Fettke, P., Loos, P. (eds.) Reference Modelling for Business Systems Analysis, pp. 47–75. Idea Group, Hershey (2007)
34. vom Brocke, J., Buddendick, C.: Reusable Conceptual Models – Requirements Based on the Design Science Research Paradigm. In: Chatterjee, S., Hevner, A.R. (eds.) Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, pp. 576–604 (2006)
35. Ward Jr., J.H.: Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association 58(301), 236–244 (1963)
36. Winter, R.: IT-Unterstützung für CRM-Prozesse: Konzeptionelle Lücken kann auch die beste (Standard-)Software nicht schließen. Wirtschaftsinformatik 45(2), 243–245 (2003)
37. Winter, R., Gericke, A., Bucher, T.: Method versus Model – Two Sides of the Same Coin? In: Albani, A., Barijs, J., Dietz, J.L.G. (eds.) Advances in Enterprise Engineering III: 5th International Workshop, CIAO! 2009, and 5th International Workshop, EOMAS 2009, held at CAiSE 2009, Amsterdam, pp. 1–15 (2009)
38. Zablah, A.R., Bellenger, D.N., Johnston, W.J.: An evaluation of divergent perspectives on customer relationship management: Towards a common understanding of an emerging phenomenon. Industrial Marketing Management 33(6), 475–489 (2004)

# The Instance-Based Multilevel Security Model


Jeffrey Parsons[1] and Jianmin Su[2]

[1] Faculty of Business Administration, Memorial University of Newfoundland
St. John's, NL A1B 3X5 Canada
jeffreyp@mun.ca
[2] Department of Computer Science, Memorial University of Newfoundland
St. John's, NL A1B 3X5 Canada
jianmin.su@mun.ca



**Abstract.** Managing data securely is a critical issue in modern organizations. Multilevel database systems offer one approach to security that assigns various security levels or clearances to the data in a database and to users of a database. The objective of multilevel databases is to share data that have been assigned different security levels, while preventing unauthorized access to data by a user with a given clearance level to data at a higher security classification. Current models store information in different security levels separately to prevent unauthorized access to data by users at different levels. However, leakages could still occur in cases such as those involving null values and those where sensitive data is included in the key field(s) of a relation. In this research, we design a novel security model, the instance-based multilevel security model, to solve problems associated with existing security models. We define the model, propose data access and integrity rules, and develop a two-layered access control method. Finally, we prove that the model is secure and identify areas for future research.



**Keywords:** Access control, confidentiality, instance-based security model, security, multilevel security, polyinstantiation.


## 1 Introduction

Multilevel database systems have been proposed to address the increased security needs of database systems. *Multilevel* means there are multiple clearance levels. A multilevel database is intended to provide the security needs for databases that contain data at a variety of security classifications and serve a set of users having different clearances [1]. In multilevel databases, higher-level security users can access lower-level security data but not vice versa. If lower-level security users can use any means to access higher-level security data, directly or indirectly (e.g., by guessing), then the security system is termed ***broken***. The means to allow a lower-level security user to access higher-level security data is referred to as a ***covert channel*** [2]. The objective of multilevel database systems is to enable data sharing between different clearance levels while preventing covert channels between levels. This increases the usefulness of data in the database while preserving the security needs of the organization.


R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 365–380, 2010.

Many multilevel security database models have been proposed. For example, the Bell-LaPadula model [3] stresses two basic needs of multilevel security systems: (1) a lower level user cannot read any higher level data; and (2) a higher-level user cannot update any lower-level data. The SeaView security model [1] provides an applied multilevel security database system by extending the standard relational model. Several extensions of the SeaView model have been proposed, including the multilevel relational (MLR) data model [4] and the belief-consistent multilevel secure (BCMLS) relational model [5]. These models were developed to improve security or efficiency. For example, polyinstantiation integrity introduced in MLR was used to improve security needs at different levels, and belief label introduced in BCMLS was used to reduce storage redundancy.

Despite advances in database security, existing models have problems that limit their usefulness. In this paper, we contend that a fundamental problem with existing security models is that they are linked to the class-based assumption underlying the relational data model – namely, that data instances (e.g., records or objects) belong to classes and that security is managed at the class level. We propose an instance-based multilevel security model (IBMSM) as a design artifact in the domain of data management, propose a set of data access and integrity rules, show that the model addresses common problems with existing models, and prove that the model is secure.

## 2 Problems with Traditional Security Models

Multilevel database systems face many challenges. A multilevel database system is governed by the two restrictions in the Bell-LaPadula proposal. The purpose of these restrictions is to prevent covert channels between security levels. However, when applying the two restrictions in a database, there is a trade-off between the objectives of sharing data and ensuring security. Some problems arise as a result.

**Data redundancy.** The SeaView security model and its derivatives define a rule, called *entity polyinstantiation integrity*, which allows a database to store the same record (i.e., having the same key attribute value(s)) at different security levels to achieve the goal of protecting the higher security data. However, entity polyinstantiation integrity is not designed to support data sharing. The basic principle of the entity polyinstantiation integrity rule is to store data at several security levels. Figure 1 illustrates an entity polyinstantiation example in a multilevel security database. Name is the primary key of the relation and security classifications are assigned at the granularity of individual data elements. In addition, TC (tuple class) (namely TS, S, U) is the security classification of each tuple of each record (entity).

| Name | Weight | Age | TC |
|---|---|---|---|
| John U | 180 U | 28 U | U |
| John U | 180 U | 28 U | S |
| John U | 180 U | 28 U | TS |

**Fig. 1.** Entity Polyinstantiation Integrity Example

As we can see in the relation, the three records express the exact same real world thing, a customer John. However, the polyinstantiation integrity rule splits data to different security levels. Of course, this approach increases the data stored in database, resulting in data redundancy. The MLR security model introduces the "borrow" concept and stores pointers, rather than real data, in the higher-level to deal with this problem. Even then, we still need to store three records to represent the single real world entity depicted in Figure 1. The BCMLS model can solve the problem only if the redundancy of all attributes in an entity belongs to the same security level, as is the case in Figure 1. The records can be stored as in Figure 2 in BCMLS. However, if we have data as in Figure 3, BCMLS still needs to store three records; thus, the redundancy cannot be reduced.

| Name | Weight | Age | TC |
|---|---|---|---|
| John U S TS | 180 U S TS | 28 U S TS | U S TS |

**Fig. 2.** One Record Belongs to Three Levels in BCMLS

| Name | Weight | Age | TC |
|---|---|---|---|
| John U S TS | 180 U TS -S | 28 U S -TS | U -S TS |
| John U S TS | 181 S -U TS | 28 U S -TS | S -TS |
| John U S TS | 180 U TS -S | 26 TS | TS |

**Fig. 3.** Three Records Belong to Three Levels in BCMLS
[Note: a user at a security level preceded by "-" does not believe the attribute value]

**Null value inference problem.** The second problem faced by existing multilevel security models is inference when dealing with sensitive data. For example, if we have a set of data similar to what is shown in Figure 4(a), a query from an unauthorized (U level) user may result in null values, as shown in Figure 4(b). The null values generated may result in security risks of inference. The lower level user may infer that there is a value for John in Weight that is not accessible. In some cases, the unauthorized users can breach the security system and obtain the sensitive information by executing a pre-designated sequence of queries.

| Name | Weight | Age |
|---|---|---|
| James U | 180 U | 32 U |
| John U | 225 S | 28 U |

(a)

| Name | Weight | Age |
|---|---|---|
| James U | 180 U | 32 U |
| John U | Null | 28 U |

(b)

**Fig. 4.** The Inference Problem

The null value inference problem can be reduced if tuple-level labels are added and each tuple-level label is set to at least the highest security level of its components. However, since lower-level users cannot access tuples with higher tuple-level labels, additional tuples have to be created for the lower-level users to access. In Figure 5, a new tuple (the third tuple) needs to be created for U level users. In general, for each lower level, we may need to add a tuple. However, this will result in data redundancy as previously described.

| Name | Weight | Age | TC |
|---|---|---|---|
| James U | 180 U | 32 U | U |
| John U | 225 S | 28 U | S |
| John U | 220 U | 28 U | U |

**Fig. 5.** Data Redundancy Problem

**Sensitive key value problem.** In traditional multilevel security models, the polyinstantiation integrity rule, intended to protect sensitive data from lower level subjects (users), allows only non-key attributes to have different values at different security levels. However, these models leave a problem unsolved: how to deal with sensitive data in the key attribute(s). Since the relational model uses key attribute(s) to identify records (instances), when sensitive data is included in the key attribute(s), the polyinstantiation integrity rule must be relaxed and therefore cannot protect the sensitive data. Figure 6 demonstrates this problem. In the PERSON table, the attribute Name is a key attribute. Three records are assigned to different security levels. In the first record, the value James in Name attribute is assigned to the unclassified level (U level). The second record, the value is still James for secure level (S level); however, in the third record, the value for Name attribute has changed to John and is assigned to the top security level (TS level). This will not be a problem if the records contain data of two real world people. However, suppose that in this case, the three records represent one real world thing, possibly a government agent, whose real name is John. In this case, the highest level's subjects (TS level users) will access all three records as in Figure 6. However, without further information they would not recognize that the first two records are masks that protect the last record from lower level subjects.

PERSON table

| **Name** | **Weight** | **Age** | **TC** |
|---|---|---|---|
| James U | 160 U | 32 U | U |
| James S | 170 S | 30 U | S |
| John TS | 170 S | 31 TS | TS |

**Fig. 6.** The Sensitive Key Value Problem

**Summary.** The difficulties with multilevel security models arise from the basic limitations of class-based models such as the relational model [6], rather than the multilevel security models themselves. In a class-based database, a schema comprises the global view of data. Because of the existence of such a schema, any data (and/or users) have to be assigned to a certain (global) security level; this is where problems originate. Since the designer has to decide security levels in advance to match the schema, a number of questions also arise: How many security levels will be sufficient for the application?

Should users with the same security level but different needs access the same portion of the dataset? For example, does the sensitive information in a security department at a university share the same security level as an academic department? How can administrators modify the authorizations of users after the system is built? If a user is initially assigned with medium security level, how can we allow this person to access sensitive data beyond his/her security level?

Based on its theoretical principles, the multilevel security model should be a reliable and convenient method for protecting sensitive information. However, the basic concepts of class-based models have limited the applications of this security control methodology, hence its success. In this paper, we propose a security model based on an alternative approach to data organization – the instance-based data model.

## 3 The Instance-Based Data Model

The instance-based data model (IBDM) [6, 7] is based on a philosophical ontology [8] and on classification theory in cognitive psychology [9]. It does not assume "inherent classification," which is fundamental to class-based models such as the relational and OO models. The following are some fundamental concepts of ontology and classification theory on which the instance-based model depend.

*Representation Principle 1*: The world is viewed as made of things that possess properties.

*Representation Principle 2*: Classes are abstractions created by humans in order to describe useful similarities among things.

*Corollary 1*: Recognizing the existence of things should precede classifying them.

*Corollary 2*: There is no single "correct" set of classes to model a given domain of instances and properties. The particular choice of classes depends on the application.

Based on the two principles and their corollaries, the instance-based model proposes a two-layered approach to information modeling, each layer assuming responsibility for representing different aspects of a domain. The instance layer represents instances and their properties. The class layer describes how the things are classified for certain purposes. Each layer stores information and implements operations as follows:

*Instance layer*: consists of the instances and their properties necessary to model a particular domain. The operations on the instance layer provide the capability to create, maintain, and examine information about the domain of instances.

*Class layer*: consists of classes that describe similarities among instances in the instance layer in terms of their shared properties. The operations on the class layer provide the capability to create, maintain, and examine the classes in the class layer. Some of these operations may invoke operations on the instance layer.

Class-based data models implicitly assume two conditions: (1) We identify every thing by a specific class to which it belongs; (2) There exists a preferred set of classes to describe a domain. In a class-based data model, all information about instances is logically organized in classes. There are no instances that remain unclassified and all instances in a class possess the same properties specified in the class definition. However, in the instance-based model, although there are also instances and classes, the conceptualization is not the same. A class is only defined by some common properties of a set of instances. So there are two basic differences between the two models. First, in the instance-based model, instances are independent of classes. Second, instances in a class may possess different properties. The two basic differences produce very different results between the instance-based model and class-based models. For example, in the instance-based model it is possible both that some instances do not belong to any class and other instances belong to more than one class. In contrast, in class-based models an instance must belong to a class, and instances generally do not belong to more than one class unless those classes are subclasses of a common superclass.

## 4 The Instance-Based Multilevel Security Model (IBMSM)

People recognize a thing (instance) by recognizing its properties. However, since the knowledge that people (subjects) have about the real-world things (objects) is different, the level of understanding, or level of recognition, of the subject is different as well. For example, a young child may describe the sun (a thing) as a bright sphere. However, an astronomer will characterize the sun in much more detail (e.g., by mass, temperature and photospheric composition). In the multilevel security model, the ability of subjects to recognize an object in terms of its properties is directly related to their security levels. In the context of the above example, the astronomer can be considered to be in a higher security level than the young child. To recognize this hypothesis, we propose the following about views of an instance.

*Proposition 1 (Property views)*: Subjects in different security levels recognize an object (instance) by recognizing its properties. Different levels of subjects have different capability to identify a property, hence different views of the same real property. We call these views of the real property.

Proposition 1 assumes that all subjects in the same security level have the same capability to identify a property. However, they may not have the same interests in the objects. For example, in a company a production manager may be interested in production equipment; however, a business manager may not have interests in the equipment used to produce products, but in promoting and merchandising company products. So, we offer another proposition to deal with this situation.

*Proposition 2 (Class View)*: Different subjects may be interested in different sets of instances (objects); each set of instances could be (recognized as) a class, which expresses all the common aspects of the instances.

Following the above propositions, the instance-based multilevel security model consists of three parts – the instance, the class and the control models. The definitions of the three parts are:

*Definition 1:* A ***view of an instance*** at a security level $S_j$ is denoted by $\boldsymbol{i}$ $\{(P_i, S_j) \mid P_i \in \mathbf{P}$ and $S_j \in \mathbf{S}\}$, where $\boldsymbol{i}$ is an instance identifier, $P_i$ is a property view over the set of all properties (**P**), $S_j$ is a security level over all security levels, **S**, in a database. A pair $(P_i, S_j)$ indicates that a property's view, $P_i$, belong to the security level $S_j$.

An instance may have a different view in different security levels. For example,

*Instance1{(Name James, U), (Weight 160, S), (Weight 160, U), (Age 30, TS), (Age 32, S)}*, where U<S<TS, has the following view at the S level:

*Instance1{(Name James, U), (Weight 160, S), (Weight 160, U), (Age 32, S)}.*
However, in the U level, the instance will have the view:

*Instance1{(Name James, U), (Weight 160, U)}*

As we can see, since the lower level user cannot access the higher level data, the instance view at the U level includes less information than at the S level.

Note that a higher level user can see lower level data according to Definition 1, which is the same as in traditional multilevel security models (for example, the MLR model). In the IBMSM, an instance might not have a view of a property at a higher security level but several views of the same property at lower levels. However, in traditional multilevel security models, if a higher level record needs to access values of a lower level record, it has to borrow a value from a certain level record before higher level records can be inserted into a database (this is called the belief-based assumption in traditional multilevel security models). We will discuss the problem of the belief-based assumption in the next section.

*Definition 2*: A ***class*** is denoted by Class_ID ($\{p_i\}$,$\{u_i\}$), where Class_ID is a class identifier, $\{p_i\}$ is a subset of properties of all properties (**P**), and $\{u_i\}$ is a subset of subject (user) identifiers over all the system.

A class contains two pieces of information. First, it includes information about which instances (objects) should be included in the class. Second, it includes information about which users (subjects) can access this class. For example, if we define a class *Class1({Name, Age}, {user1, user3})* then an instance *Instance1{(Name James), (Weight 170), (Age 30)}* belongs to the class. However, an instance *Instance2{(Name John), (Weight 170)}* does not belong to the class. Meanwhile, user1 and user3 can access this class. However, other users cannot access this class.

*Definition 3*: A view of an instance at security level $S_j$, which is ***i*** $\{(P_i, S_q) \mid P_i \in \mathbf{P}, S_q \leq S_j, \text{ and } S_q, S_j \in \mathbf{S}\}$, belongs to a class $C(\{p_k\},\{u_i\})$ if and only $\{p_k\}$ is a subset of $\{p_i\}$. A subject (user) U can access a class $C(\{p_k\},\{u_i\})$ if and only if $U \in \{u_i\}$.

As indicated, an instance may have different views at different security levels. Thus, an instance may belong to different classes at different levels. For example, if we define two classes; Class1({Name, Age},{user1, user3}) and Class2({Name, Weight},{user2, user3}), then an instance, Instance1{(Name James, U), (Weight 160, S), (Weight 160, U), (Age 30, TS), (Age 32, S)}, belongs to Class1 and Class2 at the TS and S levels. However, it only belongs to Class2 at the U level. At the U level, the instance only has a view Instance1{(Name James, U), (Weight 160, U)}. At this level it does not have the properties {Name, Age} in the definition of Class 1.

To enhance system security, we propose a rule for the IBMSM model originating from the two Bell-LaPadula restrictions for the instance-layer.

*Rule 1*: A subject u at a certain security level S (designated as $u_S$) can read a property (a view of a property) of an instance at a security level $S_j$ (we express this property as $p_{Sj}$) if and only if $S \geq S_j$. However, $u_S$ can update $p_{Sj}$ if and only if $S=S_j$.

Rule 1 indicates that the data that a user at a certain level security can read consists of two portions: One is the data in the same security level as that of the user, and the other is the data in security levels lower than that of user. The latter can be updated by lower-level users who have the same security level with the records. In other words, a subject (user) can update the data in the same security level as itself (the user); it cannot update data in any lower security levels even though it can read them.

## 5 Data Interpretation in IBMSM

For all instances $i$ $\{p_i | p_i \in \mathbf{P}\}$, in the IBMSM, data are interpreted in two parts: the instance part and the property part. We describe each of them as follows:

**Property $P_i$ and its Security Level** $S_j$. An instance possessing a property view, $p_i$, at a security level $S_j$ is denoted as a pair ($p_i$, $S_j$) as in Definition 1. However, since an instance may possess the same property's view in more than one security levels, ($p_i$, $\{S_j\}$) is used to denote more than one pair ($p_i$, $S_j$) (e.g., ($p_i$, $S_1$), ($p_i$, $S_2$) … ($p_i$, $S_{10}$)) and vice versa, if an instance possesses more than one property views of different real properties in the same security level S, they are denoted as ($\{p_i\}$, S).

**An Instance's View and its Security Levels**. An instance identifier $i$ identifies an instance in the database. $i$(S) denotes that an instance $i$ possesses some properties at security level S. To represent that an instance possesses properties that belongs to more than one security level i.e., an instance $i$ possesses some properties in security levels $\{S_j\} \in \mathbf{S}$, the notation $i(\{S_j\})$ is used.
If an instance possesses a property at a security level S, the notation ($p_i$, $S$), represents that an S level user has created a property, $P_i$, of the instance. Instances $p$ and $q$ are identical at a security level S, if and only if they have the same view at the security level, that is: if, for all $P_i$, $(P_i, S) \in p$ if and only if $(P_i, S) \in q$.

For example, two instances, Instance1{(Name James, U), (Weight 160, S), (Weight 160, U), (Age 30, TS), (Age 32, S)} and Instance2{(Name James, U), (Name John, S), (Weight 160, U), (Age 32, TS), (Age 30, S)}, are identical at the U level since they have the same property view, {(Name James, U), (Weight 160, U)}, at that level.

## 6 Data Access and Integrity Rules

We define several rules to guarantee that data in an IBMSM database is secure and consistent. These rules are related to data access and integrity in the IBMSM.

*Rule 1* (Instance view integrity): An S level view of an instance $i$, which is $i$(S), can exist in an instance-based multilevel security database if and only if no identical view of another instance $j$, $j$(S), exists in the same level in the database.

Rule 1 ensures that no duplicate objects exist in any level of a database. It guarantees the semantics of the instance identifier. Since an instance identifier includes all the features of the instance, and no instance should have completely the same features as other instances in the real world, if a subject discovers that any objects have completely duplicated properties, the objects are identical. The original idea of Rule 1 comes from the semantics of the instance identifier in the instance-based model; however, we extend the rule to each security level.

*Rule 2* (Property Integrity): Instance $i\{(p_p, S_j)|p_p \in \mathbf{P}$ and $S_j \in \mathbf{S}\} \in \boldsymbol{i}$ satisfies property integrity if and only if, for any pair of $(p_p, S) \in i$, and $(p_q, S) \in i$, $p_p \neq p_q$, the expression $p_p \cap p_q = \phi$ is always true.

Rule 2 states that a user (subject) at security level S can create a property $p_p$, which is $(p_p, S)$, of an instance $i$, if and only if the instance does not have property $p_p$ at the security level S and there is no other property of the instance $i$ at the level S that is a compatible property[1] of $p_p$. Rule 2 is also a rule extended from the instance-based model to the IBMSM.

Note that the property integrity rule only applies to properties in the same security level. If two properties of an instance belong to different security levels, then it does not matter whether they have an intersection.

*Rule 3:* A user (subject) at security level S can read a property $p_i$ of an instance at security level $S_j$, that is $(p_i, S_j)$, if and only if $S \geq S_j$.

*Rule 4:* A user (subject) at the security level S can only create (or update) a property $p_i$ of an instance at the security level S (not higher, not lower).

Rule 3 and Rule 4 extend the basic Bell-LaPadula rules, which is no read up and no write down, to the instance-based setting.

*Rule 5:* (Association integrity) An association (a mutual property) of two instances, $i\{(p_i, S_j)|\ p_i \in \mathbf{P}$ and $S_j \in \mathbf{S}\ \}$and $i'\{(p_{i'}, S_{j'})|\ p_{i'} \in \mathbf{P}$ and $S_{j'} \in \mathbf{S}\ \}$, at a certain security level $S_{ij}$ exists only if

(a) $i$ and $i'$ exist in the database. That is $i \in \boldsymbol{i}$ and $i' \in \boldsymbol{i}$.
(b) The security level of the association, $S_{ij}$, should belong to both $\{S_j\} \in i$ and $\{S_{j'}\} \in i'$. that is $S_{ij} \in \{S_j\} \cap \{S_{j'}\}$. Rule 5(a) just follows the referential integrity of the instance-based model, but (b) is new to the security model. Rule 5(b) indicates that instances can be associated in a security level if and only if they both can be updated in that level. For example, assume there are two instances, instance 1{(Name John, S), (Age 21, TS), (Weight 120, S), (Sex M, S)} and instance 2 {(Name Alice, U), (Age 20, S), (Sex F, U)}, in an IBMSM database. If the two instances associate together to form a higher level thing, e.g., they are married and form a couple, then such association can only be formed at the S level. Although instance 1 belongs to the TS level and users at the TS level can read instance 2, instance 2 does not belong to the TS level. Following Rule 4, users at the TS level cannot update any information about instance 2. Adding an

---

[1] By Bunge's ontology, two properties P1 and P2 are ***incompatible*** over a set $T \subseteq S$ of substantial individuals iff possessing one of them precludes having the other. They are mutually compatible over T iff they are not incompatible over T.

association between instances 1 and 2 at the TS level means to update information of both instances at the TS level, so the operation is not allowed.

*Rule 6:* A subject, u, can access data through a class, C({$p_i$},{$u_j$}), iff u∈ {$u_j$}.

Rule 6 represents the basic idea of two-layered access control in the IBMSM that plays an important role in the security control. We discuss it in more detail next.

## 7 Two-Layered Access Control

The instance-based multilevel security model uses two layers to control access to data. The class layer governs the range of objects accessible to a particular subject. The instance layer controls access to the sensitive data in a class by a subject (user).

Figure 7 illustrates such control in the two-layered approach. The dataset constitutes the instance layer; the classes represent the class layer. The original dataset is divided into different sections, each section containing both sensitive and non-sensitive data. To access the data in each section, either non-sensitive or sensitive, users have to be able to access different classes first. As shown in the figure, in order to access the dark section (sensitive data), users have to be authorized to access class 1 first. Both User 1 and User 3 may access data in the dark data section since they both have ability to access Class 1; however, User 2 cannot do so since it does not have authorization to access this class. The ability of users to access the same class does not necessarily mean that they share the same ability to access the data in the data-section through this class. For instance, an unclassified (U) user, User 3, can only access the non-sensitive data but not the sensitive portion, even if he/she is authorized to access the same classes as the authorized (S) user, User 1.

SeaView and its derivatives use views to control access as well. However, the two-layered security approach in the instance-based multilevel security model is different from the original multilevel security models in class-based models. For example, in MLR model, even if a real object (thing) has several records in different security levels they belong to the same class (table). However, in the IBMSM, an instance may belong to different classes on different security levels. For example, an instance *{(Name John, U), (StudentID 2000001, S), (Birthday 93/05/06, TS)}* is a student (defined by *student{StudentID}*) on S level, but it does not belong to the class student on U level. This approach increases the security of the IBMSM.

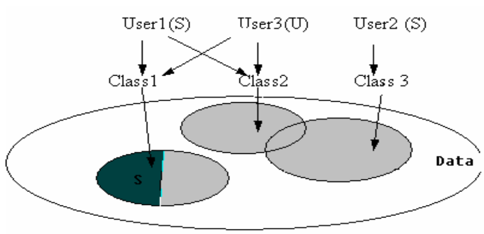



**Fig. 7.** Two-layered Controls

## 8 Security

IBMSM utilizes a two-layer approach to control access, the class and the instance layers. In the class layer, the accessibility of a class to a user is not determined by user's security level. Users at different security levels may access the same class (the definition of the class). Also, sensitive information is not accessible by users in the class layer. Any data in the class layer is maintained by system administrators. A user may query a class definition if he/she can access the class. Therefore, the formation of inference channels is impossible on the class level. The following proof focuses on the instance level.

*Notation*:
*U*: the set of all users with varying security levels.
*I*: the set of all instances with varying views in all security levels.
*P*: a property with varying security level views in an instance.
*s*: a certain security level.
*UL*(s): the set of users with security levels lower than or equal to *s*.
*UH*(s): the set of users with security levels higher than *s*.
*IL*(s): a set of views of instances with the security level *s* or lower.
*IH*(s): a set of views of instances with the security level higher than *s*.
*PL*(s): a set of views of properties of instances with the security level equal to or lower than *s*.
*PH*(s): a set of views of properties of instances with security level higher than *s*.
From the above notation, six equations can be obtained:

$$UL(s) \cup UH(s)=U \tag{1}$$

$$UL(s) \cap UH(s)=\phi \tag{2}$$

Equations 1 and 2 mean that: all users are in a security level either higher than (or equal to) *s* or lower than *s*; no user is in a security level both higher than (or equal to) *s* and lower than *s*.

$$IL(s) \cup IH(s)=I \tag{3}$$

$$IL(s) \cap IH(s)=\phi \tag{4}$$

Equations 3 and 4 mean that: all views of instances have their security level either higher than (or equal to) s level or lower than s level; no view of any instance has its security level both higher than (or equal to) s level and lower than s level.

$$PL(s) \cup PH(s) =P \tag{5}$$

$$PL(s) \cap PH(s) =\phi \tag{6}$$

Equations 5 and 6 mean that: (1) all views of properties are in a security level either higher than (or equal to) s or lower than s; and (2) no view of any property is in a security level that is both higher than (or equal to) s and lower than s.

Note that in IBMSM, if an instance, *i* is in *s* security level, *i*(s), this means that it has at least one view of at least one property of the instance *i* at the security level *s*.

A database is an organized collection of data. A database state is a collection of all instances of a database at a particular time. A secure state of a database is a state such that from that state any operation on the database will be finalized (complete or failure), and no covert channel could exist between the state and the final states. A ***secure data model*** is a database model that takes a database from one secure state, through a number of operations, to another secure state. Figure 8 illustrates this idea. Goguen and Meseguer suggest a security data model [10], which defines several concepts for security data models. We will also use their definition for a secure database. For example, for any security level s, a command to delete any data issued by users at a higher security level does not affect the view of data to any user at the lower security levels.

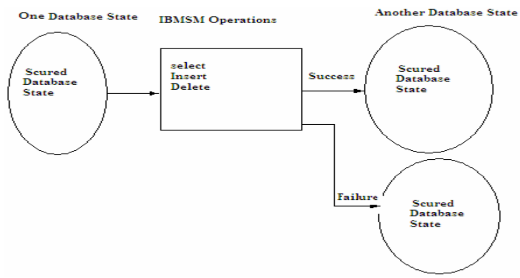



**Fig. 8.** IBMSM Database State Transformations

**Theorem:** The IBMSM is a secure data model.

As shown in Figure 8, a database is modified from the initial to the final state by a series of user operations. To prove that IBMSM can only go from one secure state to another secure state through a sequence of operations, we only need to prove that all IBMSM operations are secure, as these are the only operations allowed.

In the IBMSM, all database operations are issued by users on a certain security level. First, we prove that database operations at some security level will not affect another level; that is, it will not change the database state at another security level. For example, a higher security level user operating on data should not affect any lower security level users. Also, a lower security level user operating on data should not affect any higher security level users. Thus a direct path from either left to right or right to left in Figure 9 is not possible.

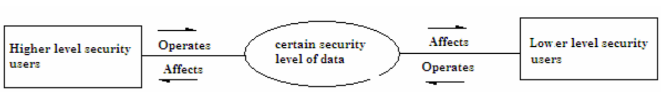



**Fig. 9.** Different Security Levels of Users Affecting Each Other

In fact, a user at a certain security level, *s*, may operate on two types of data: data at security levels equal to or lower than *s*; and data at the security levels higher than *s*. So, whether users affect each other in Figure 9 can be expressed as Figure 10.

We prove Theorem 1 by following two steps. First, we prove that any higher level data change will not affect any lower level user in IBMSM.

*Lemma 1:* For any security level s, changing data at higher level security views of instances, IH(s), will not affect any users u∈UL(s).

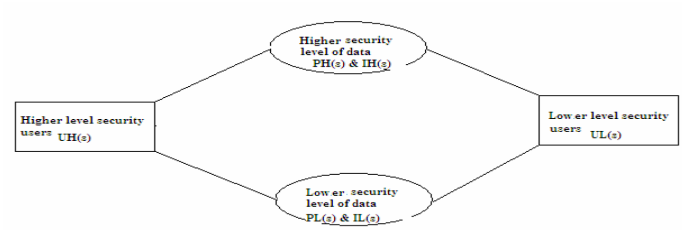



**Fig. 10.** Users Affect Each Other on Different Security Level Data

*Proof:* A user u at security level s' can use several database operations: *select*, *insert*, and *delete*. A user, u, at security level s', s' ≤ s, means u∈UL(s). For the *select* operation, the user *u* can access any view of an instance, *i*(s')∈IL(s') that is equal or less than the level s'. IL(s') is a set of views of instances whose security level is in s' level. Only p(s') or lower security level properties of instances, or *PL(s')*, can be accessed by the user *u*. Views of instances at levels which are higher than s' level are not accessible by user u. Since s'≤s, PH(s')⊇PH(s) and IH(s')⊇IH(s). We know that IL(s')∩IH(s')=$\phi$ and PL(s')∩PH(s') = $\phi$. So, IL(s')∩IH(s)=$\phi$ and PL(s')∩PH(s)=$\phi$. That is, any changes in PH(s) and IH(s) will not affect PL(s') and IL(s'). Therefore, changes of PH(s) and IH(s) do not affect s' level users u∈UL(s) with s'≤ s.

For *insert* and *delete* operations, any operation either succeeds or fails. However, since IBMSM only allows *u* to operate on the data with these operations at its own security level, successful operations only modify with the data at the security level *s'*. Changes at security levels other than *s'* will not affect the users at those levels. That is, changing PH(s) and IH(s) will not affect any *s'* level users u∈UL(s) with s'≤ s. Several factors can results in failures of operations.

(1) An *insert instance* command issued by a *s'* level user, *u*, to insert an instance *i* fails if and only if:
   (a) There are two properties $p_i$ and $p_j$, ($p_i$, s')∈ *i* and ($p_j$, s')∈ *i*, and $p_i \cap p_j \neq \phi$;
   (b) There is an instance, instance *j*, which i(s')=j(s') for any s'∈ ***S.***
(2) An *insert mutual property* command issued by a s' level user, u, to insert a mutual property of two instances, *i* and *j*, fails if and only if:
   (a) *i*(s')∉***i*** or *j*(s')∉***i***;
   (b) There are two property pairs ($mp_i$, s') and ($mp_j$, s') such that $mp_i$ and $mp_j$ are compatible (see footnote 1).
(3) A *delete instance* command issued by a *s'* level user, *u*, to delete an instance view at the s' level, *i*(s'), fails if and only if a combined instance (mutual property) is formed by the instance at the s' level.
(4) A *delete mutual property* command issued by a *s'* level user, *u*, to delete a mutual property view at the *s'* level, *mp*(s'), fails if and only if a combined instance (mutual property) is formed by the mutual property at the s' level.

All these situations deal with *s'* level data (instance view, property view, etc). Data at the s' level (p(s'), mp(s'), or *i*(s')) belongs to PL(s') or IL(s'). Since s'≤s, PH(s') ⊇PH(s) and IH(s')⊇IH(s). So, p(s'), mp(s')∉PH(s')⊇PH(s), i(s')∉ IH(s')⊇IH(s). That is, changes in PH(s) and IH(s) do not affect any s' level users u∈UL(s) with s'≤ s.

Second, we show that a higher level user will not affect lower level data.

*Lemma 2:* For any security level s, higher level security user u, u∈UH(s), changing data does not affect any data PL(s) or IL(s).

*Proof*: A user may change data in two ways: insert data into or delete data from an IBMSM database.

(1) An *insert instance* command issued by the *s'* level (s'>s) user *u*, u∈UH(s), to insert an instance view *i*(s'), can only add a set of property views of the instance *i*, {p(s')}∈PH(s), to the database. Since s'>s, the added data {p(s')}∉ PL(s).
(2) An *insert mutual property* command issued by the *s'* level (s'>s) user *u*, u∈UH(s), to insert a *mutual property* view of two instances, *i* and *j*, can only add a mutual property view of the instances *i* and *j*, mp(s')∈PH(s), to the database. Since s'>s, the added data mp(s')∉PL(s).
(3) A *delete instance* command issued by the *s'* level (s'>s) user u, *u∈UH(s),* to delete an instance view at the s' level, *i*(s'), can only delete a set of property views of the instance *i*, *{p(s')}∈PH(s),* from a database. Since s'>s, the deleted data *{p(s')}∉PL(s).*
(4) A *delete mutual property* command issued by the *s'* level user, *u*, to delete a mutual property view at the s' level, *mp*(s'), can only delete a mutual property view *mp(s')∈PH(s),* of instances formed to a database. Since s'>s, the deleted mutual property *mp(s')∉ PL(s).*

From the above operations, if the *s'* level user, s'>s, changes any data, this does not affect any data in *PL(s)* or *IL(s).*

From the above two lemmas, neither path from higher level users to lower level data nor higher level data to lower level users is applicable. Figure 11 shows the results of the two lemmas.


**Fig. 11.** Effects on Different Levels

Finally, Theorem 1 can be proven. As shown in Figure 11, since U=UL(s)∪UH(s) and ϕ=UL(s)∩UH(s), no user is in between UL(s) and UH(s). Because I=IL(s)∪IH(s), ϕ=IL(s)∩IH(s), P=PL(s)∪PH(s), and ϕ=PL(s)∩PH(s), the intersection between two sets of data, {PH(s) & IH(s)} and {PL(s) & IL(s)}, is empty as well . Thus, no connection can be made between either different security levels of users or different security levels of data. Therefore, the IBMSM is a secure model.

## 9 Advantage of IBMSM

The semantics of data in the IBMSM model are different from the class-based model, resulting in several advantages of the IBMSM relative to traditional models. In particular:

1. In IBMSM, the requirement that an instance should belong to any class (schema) is eliminated. So, the greatest lower bound to define a view of an instance at any security level, common in all class-based multilevel security models, is unnecessary. Eliminating this assumption in the IBMSM enhances the security of the model. For example, the null value problem does not exist in the IBMSM.
2. An instance's views at different security levels may belong to different non- hierarchical classes. In the IBMSM model, subjects recognize an instance by recognizing its properties. Subjects at different security levels have different abilities to recognize properties of an instance. Since a higher-level subject (user) can access any lower-level data, the higher-level subjects (users) may recognize that an instance belongs to a class that the lower-level subjects (user) may not recognize. For example, we define an overweight class: a person (instance) is overweight if his weight is more than 300lb. Then an instance {(Name James) U, (Weight 280lb) U, (Weight 305lb) S, (Age 21) U} belongs to the overweight class at the S level since weight is 305lb at the S level. However, it does not belong to the overweight class at the U level since the U level users only recognize its weight as 280lb.
3. The absence of a property of an instance for subjects at a security level means that this property is not present at the security level. However, the absence does not reflect the rejection of this property. Subjects in the security level may define the property later. For example, in the relational model any instance (record) must belong to a table (class). We may use Null if the value of the attribute of the instance is missing or if the instance does not have this attribute. Thus, the semantics of the Null value is unclear. On one hand, the Null value itself tells that we do not know whether the instance has this attribute. On other hand, the schema indicates that any record in a table must have the same set of attributes. In the instance-based model, this problem is solved using an "open world" assumption. If subjects recognize a property of the instance, they just add it to the instance.
4. In class-based security models, an object could have several records (views) in different security levels. For example, most class-based security models combine several key attributes of a table and a security level as the real key to identify records in the table. Since the key attributes identify objects in the relational model, it is possible that several records (as many as there are security levels) could refer to one object. However, in IBMSM, any object is described by its instance identifier. An object only has one identifier however many security levels it might belong to. This is the biggest advantage of the IBMSM model compared to class-based security models.

## 10 Summary

In this paper, we reviewed several MLR database security models and indicated the problems faced by current MLR models. We claim that the problems come from a class-based view of the world, which widely used in traditional data models. We then proposed a new multilevel security model, IBMSM, using instance-based concepts. We define data interpretation and access rules for the model, and use two layers to ensure complete security control compared to the traditional multilevel security control methods. Finally, we proved that the model is a secure model. The essential advantage of the IBMSM is that it provides a way to store only one instance (record) even if the instance belongs to more than one security level. Therefore, the IBMSM achieves the goal of multilevel security models - to maximize sharing of data at different security levels. Since the semantics of the IBMSM is based on the instance-based model, we believe the concepts of the model are simple to understand. However, further research is needed. For example, work is needed to determine how best to implement the model. Since the IBMSM is different from the class-based models, research is needed to design data structure to store data so that the processing operation is efficient and secure. In addition, methods are needed to permit concurrent access for the IBMSM.

## References

1. Lunt, T., Denning, D., Schell, R., Heckman, M., Shockley, W.: The SeaView Security Model. IEEE transaction on software engineering 16(6), 593–607 (1990)
2. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems. Addison Wesley Longman, Inc., Amsterdam (2000) ISBN 0-8053-1755-4
3. Bell, E., LaPadula, L.: Secure computer systems: Unified exposition and multics interpretation. Technical Report MTR-2997, The Mitre Corporation, Burlington Road, Bedford, MA, USA (1973)
4. Sandhu, R., Chen, F.: The Multilevel Relational Data Model. ACM transaction on information and system security 1(1), 93–132 (1998)
5. Pranjic, M., Jukic, N., Fertalj, K.: Implementing belief-consistent multilevel secure relational data model: issues and solutions. Journal of Computing and Information Technology 11(3), 225–232 (2003)
6. Parsons, J., Wand, Y.: Emancipating Instances from the Tyranny of Classes in Information Modeling. ACM Transactions on Database Systems 25(2), 228–268 (2000)
7. Su, J., Parsons, J.: Analysis of Data Structures to Support the Instance-based Database Model. In: Proceedings of Design Science Research in Information Systems and Technology (DESRIST), Claremont, CA, pp. 107–130 (2006)
8. Bunge, M.: Treatise on Basic Philosophy: Ontology I: The Furniture of the World, vol. 3. Reidel, New York (1977)
9. Parsons, J., Wand, Y.: A question of class. Nature 455(4216), 1040–1041 (2008)
10. Goguen, J.A., Meseguer, J.: Security policies and security models. In: Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 11–20. IEEE Computer Society Press, Los Alamitos (1982)

# A Negotiation Based Approach for Service Composition

Sherry X. Sun, Jing Zhao, and Huaiqing Wang

Department of Information Systems, City University of Hong Kong, Hong Kong, China
{sherry.sun,iswang}@cityu.edu.hk, zhaojingwhut@126.com

**Abstract.** In service oriented environments, complex applications can be constructed from invoking a number of atomic service components. Given that many services provide the same functionality and differ in quality of service (QoS), e.g., availability and execution time, the critical challenge in service composition is to dynamically identify the available service components to meet the user's QoS requirements and preference. In this paper, we propose to tackle this challenge through incorporating the negotiation mechanism into service composition. The negotiation mechanism allows for both service consumers and providers to exchange proposals and counter proposals to reach agreements on QoS attributes dynamically at runtime.

**Keywords:** QoS, Service Composition, negotiation.

## 1 Introduction

The software industries have witnessed an increasing use of Service-Oriented Architecture (SOA) recently [29]. In an SOA environment, software components are packaged as independent services and can be accessed without the knowledge of the implementation platform [11, 17]. Thus, as an emerging framework for distributed applications, the SOA approach allows integration of service components developed independently into complex business processes and applications to meet the users' needs and to offer extra business value.

Due to the growing number of services provided by different service providers, many services offer the same functionality and differ in quality of service (QoS), such as price, response time, reliability, and reputation, it has become an important challenge to ensure the QoS requirements in forming new value-added applications through service composition. For example, a travel planner service may combine attraction searching, flight booking, hotel booking, car rental services, and real-time weather forecasting. Different users of this travel planner service can have different budget limits and different requirements for accessing the real-time weather forecasting information. As such, the composite service needs to ensure that different requests from users can be serviced properly.

The existing works in service composition mainly focus on the methods for selecting component services with regards to the QoS requirements [1, 28]. Given a service composition request that includes a set of tasks and a list of functionally equivalent service candidates for each task, the service selection methods attempt to find one service for each task to optimize the global QoS. Essentially, the selection is made

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 381–393, 2010.

based on the QoS properties of each service pre-defined by service providers. It is difficult for a service provider to offer the service with the QoS properties customized to different requests from consumers. In addition, the QoS properties may not be precisely defined in advance since services are usually distributed across the Internet and the QoS properties can be easily affected by the environmental factors such as the network connections. As a remedy, negotiation can be added to service composition to allow for both service consumers and providers to exchange proposals and counter proposals to dynamically agree on some QoS criteria at runtime, thus providing a flexible way for service composition.

In this paper, we propose a negotiation based approach to meet the QoS constraints in service composition. In order to make negotiation possible, the global QoS constraints for a composite service need to be decomposed into the QoS constraints for each component service. Further, the negotiation algorithm we propose uses Bayesian learning to maximize the probability for an offer to be accepted. Our approach is significant in that it enables automatic service composition with a satisfaction of the QoS requirements. The remainder of this paper is organized as follows. Section 2 reviews related literature on service composition and negotiation. Section 3 describes the design artifact. Section 4 formulates the service composition problem. Section 5 presents a negotiation based method for service composition. Finally in Section 6, we conclude this paper and point out the future work.

## 2 Literature Review

### 2.1 Service Composition

Zeng et al. [28] have identified two approaches for the selection of service candidates: local selection and global optimization. The local approach [4, 6] selects optimal service candidates for each individual task without considering the global constraints for a composite service. A utility function is used to evaluate the values of the different QoS criteria and the service with the maximum utility value is selected. Although this approach is very efficient in terms of time complexity, it is not suitable for QoS-based service composition, with end-to-end constraints, for it can only guarantee the QoS criteria for each individual task.

The global approach considers QoS constraints assigned to the whole composite service. The set of services that satisfy the global constraints, while maximizing the aggregated utility function are selected. This approach guarantees global optimization while the computation complexity is high for the solution. Integer linear programming is proposed [1, 28] to study end-to-end QoS-aware service composition. Similar to this approach, Ardagna and Pernici [5] extends the linear programming model to include local constraints. However, linear integer programming is only suitable for small-sized problems as the complexity of the algorithm increases exponentially with the increasing problem size. In order to overcome the exponential time complexity of the proposed integer programming, Yu et al. [26] studies the problem in two ways: multi-dimension multi-choice knapsack problem (MMKP) and multi-constraint optimal path (MCOP) and designs efficient heuristic algorithms to find a near-to-optimal solution more efficiently in polynomial time.

Furthermore, a hybrid approach [2] that combines global optimization and local selection is proposed to take the advantages of both of the approaches. Alrifai et al. [3] proposes an approach to decompose the global constraints into local constraints with a focus on the sequential workflow process model. However, all the studies discussed above consider the QoS to be deterministic. Actually, the QoS provided by service providers are not always deterministic, and under some circumstances are negotiable.

Paurobally et al. [24] implements actual negotiation components for web services. Napoli [21] selects services candidates based on certain temporal constraints at run time and coordinated negotiation is applied to reach agreement with service providers for the temporal constraints. In this paper, we attempt to utilize negotiation for service composition while at the same time guaranteeing the end-to-end QoS constraints for a composite service.

### 2.2 Negotiation

Negotiation can be understood as a process for different parties to reach a final agreement on one or more common interest. It has been a hot topic in game theory and artificial intelligence. Under the assumption that the complete information about the opponent is available, classical game-theoretic negotiation models analyze and estimate the optimal outcomes. However, most of the negotiations are carried out with incomplete information since the information of the opponent is usually confidential and private. Therefore, one of the crucial issues is to learn the negotiation histories of the opponent to understand its behavior and improve negotiation outcomes based on such understanding. Several prediction techniques have been proposed and machine learning is one of the most popular mechanisms. Existing machine learning methods include Bayesian learning, evolutionary computation, case-based reasoning, and reinforcement learning.

Zeng and Sycara [27] presented a sequential negotiation model for updating negotiation orders between two intelligent negotiation agents in bilateral negotiations in which Bayesian belief was used as the underlying learning mechanism. A similar research [8] has also been proposed to incorporate Bayesian classification technique into negotiation agents. These studies assume that the preferences of the opponent are available. Lau et al [18] have developed a novel knowledge discovery method and a probabilistic negotiation decision making mechanism which can continuously mine the preferences of the opponents based on the negotiation histories.

Oliver [22] applied the genetic algorithm based learning technique to develop adaptive negotiation agents. In this research, agents are modeled as chromosomes while the parameters of the negotiation model are modeled as genes in the chromosome. After iterations of cross-over and mutation, the optimal chromosomes will be chosen as the solution. Instead of using the evolutionary approach to develop an agent's decision making model, Matos et al. [19] used genetic algorithm to learn optimal negotiation strategies given a particular negotiation situation that is time-dependent, resource-dependent, or behavior-dependent.

Brzostowski and Kowalczyk [7] proposed a case-based reasoning approach to select an appropriate negotiation partner by investigating previous negotiations. This possibilistic case-based model is based on the assumption that the more similar the situations are, the more possible that the outcomes are similar. Oprea [23] proposed a feed forward artificial neural network as a learning ability to model the other agent negotiation strategy. A similar research was conducted [9] to predict the opponent's offers during negotiation process. Cheng et al [10] has applied fuzzy logic to develop intelligent negotiation agents in e-marketplace. The agent's strategies are represented by fuzzy rules to generate trade-offs for quantitative issues.

# 3 Design Artifact

In this paper, we have followed the design science paradigm of Hevner et al. [14] to create a design artifact by proposing a negotiation based approach for service composition while maximizing the utility of composite services and guaranteeing end-to-end QoS constraints. Figure 1 describes the design artifact, which consists of the decomposition of the global utility and constraints into local utility and constraints, negotiation for local selection, and adjustment of the local constraints to improve the solution.

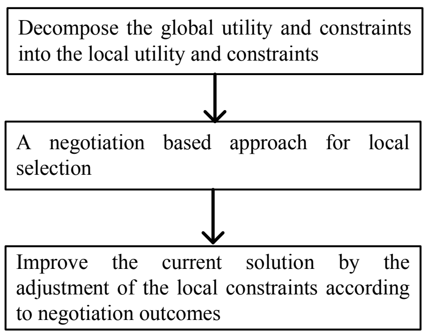



**Fig. 1.** Design artifact of this paper

The significance of our work lies in the application of negotiation to automate the process of service composition while satisfying the user's QoS requirements. Given the wide adoption of service oriented architecture, our framework is of great business relevance. Moreover, our study expands the application of the negotiation technology.

# 4 System Notations and Parameters

## 4.1 Composition Model

A composite service requested by a consumer includes a set of tasks. Each task corresponds to a service class which is a collection of functionally-equivalent service candidates. The task needs to be accomplished by one candidate from the service class. A composite service can be constructed with several component services in different structures. There are four basic structures: sequential, parallel, conditional and loop structures. In a sequential structure (Fig 2-a), a task $S_{i+1}$ in a process is not enabled until the preceding task $S_i$ is completed. In a parallel structure (Fig 2-b), all tasks

($S_1$,…,$S_n$) in parallel branches are executed concurrently. In a conditional structure (Fig 2-c), one execution path $S_i$ is chosen among many alternative paths with the probability of $PR_i$. In a loop structure (Fig 2-d), a task $S_i$ is executed n times before the next task is executed. Other complex structures can be considered as a combination of the sequential, parallel, and conditional structures. In this paper, we focus on the basic sequential model. Other models will be invested in future work.

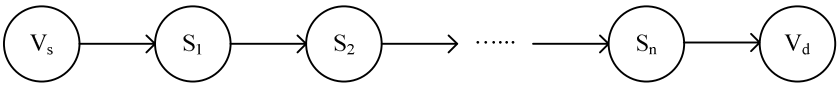



**Fig. 2-a.** Sequential structure

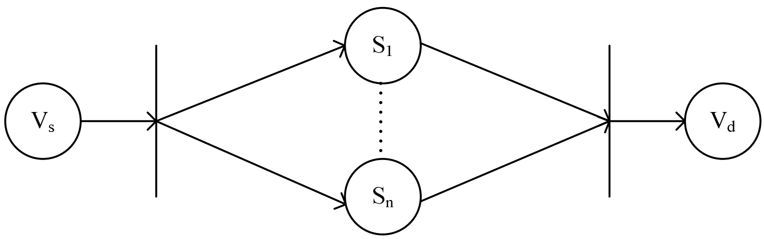



**Fig. 2-b.** Parallel structure

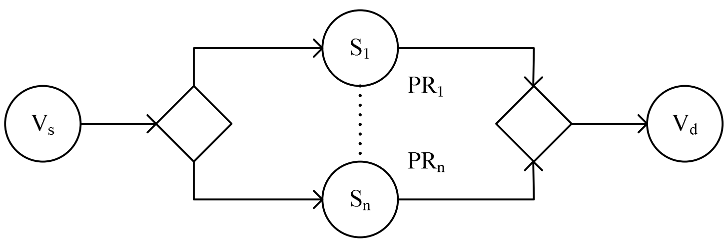



**Fig. 2-c.** Conditional structure

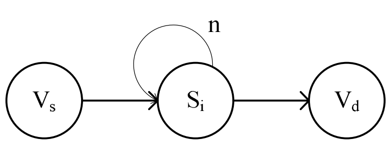



**Fig. 2-d.** Loop Structure

## 4.2 QoS Model

A composite service is composed of several component services and each component service can be invoked by a set of service candidates. To differentiate the service candidates, their non-functional properties need to be considered. A set of QoS

attributes (such as response time, cost, availability, reliability, throughput, and reputation) has already been discussed by Menasce [20], Ran [25] and Zeng et al. [28]. To compute the QoS attributes for a composite service, we need aggregation functions for each attribute. Assume $q_k(CS)$ denotes the aggregated value of the $k$-th attribute for the composite service and $q_k(S_i)$ denotes the value of the $k$-th attribute for the component service $S_i$. In this sequential model, two kinds of attributes are considered.
(1) Additive attributes
For this type of attribute, e.g. cost and response time, the aggregated value of an attribute for the composite service is the summation of the attribute value of all component services. Thus, for additive attributes, the aggregation is

$$q_k(CS) = \sum_{i=1}^{n} q_k(S_i) . \tag{1}$$

(2) Multiplicative attributes
For this type of attribute, e.g. availability, the aggregated value of an attribute for the composite service is the product of the attribute value of all component services. For multiplicative attributes, the aggregation is

$$q_k(CS) = \prod_{i=1}^{n} q_k(S_i) . \tag{2}$$

### 4.3 Utility Function

In order to evaluate a given service, a utility function is used to map all the QoS attributes into a single value [16]. A Simple Additive Weighting (SAW) [15] technique is applied to normalize the aggregated values of quality. There are two steps in applying SAW.

Firstly, we need to scale the values of the QoS in order to measure the multi-dimensional service attributes independent of their units and ranges. The set of QoS attributes can be classified into two subsets: positive and negative ones. The values of positive ones need to be maximized while the values of negative ones need to be minimized. In the normalization phase, positive and negative attributes are scaled in different ways, as defined in (3) and (4), respectively.

For negative attributes,

$$V(s,k) = \frac{Q_{\max}(s,k) - q(s,k)}{Q_{\max}(s,k) - Q_{\min}(s,k)} . \tag{3}$$

For positive attributes,

$$V(s,k) = \frac{q(s,k) - Q_{\min}(s,k)}{Q_{\max}(s,k) - Q_{\min}(s,k)} . \tag{4}$$

Secondly, we need to assign a weight value to each QoS attribute and sum up their weighted values as the utility. The utility is computed as

$$U(s) = \sum_{k=1}^{m} V(s,k) * w_k \,. \tag{5}$$

A utility function for a component service is a local utility function, while a utility function for a composite service is a global utility function. Before we compute the local utility function and global utility function, we make some simplifications about the QoS attributes. Firstly, there are both positive and negative QoS attributes. In this paper we only consider the negative attributes because the positive attributes can be transformed into negative attributes by multiplying their values by -1. Secondly, among the attributes, the aggregation functions for the multiplicative attributes are nonlinear functions and we can use the following formula to transform them into linear ones.

$$\ln q(CS) = \ln \prod_{i=1}^{n} q(S_i) = \sum_{i=1}^{n} \ln q(S_i) \,. \tag{6}$$

Thus, the local utility function and the global utility function are computed as follows.

$$U(s) = \sum_{k=1}^{m} \frac{Q_{\max}(i,k) - q_k(s)}{Q_{\max}(i,k) - Q_{\min}(i,k)} \times w_k \,, \tag{7}$$

$$U(CS) = \sum_{k=1}^{m} \frac{Q'_{\max}(k) - q_k(CS)}{Q'_{\max}(k) - Q'_{\min}(k)} \times w_k \,, \tag{8}$$

where $Q_{max}(i,k)$ and $Q_{min}(i,k)$ are the maximum and minimum value of the $k$-th attribute for service class $S_i$ and $s \in S_i$. $Q'_{max}(k)$ and $Q'_{min}(k)$ are the maximum and minimum aggregated value of the $k$-th attribute for the composite service, namely,

$$Q_{max}(i,k) = max\ q_k(S_i) \,, \tag{9}$$

$$Q_{min}(i,k) = min\ q_k(S_i) \,, \tag{10}$$

$$Q'_{\max}(k) = \sum_{i=1}^{n} Q_{\max}(i,k) \,, \tag{11}$$

$$Q'_{\min}(k) = \sum_{i=1}^{n} Q_{\min}(i,k) \,, \tag{12}$$

where $i$=1,...,$n$, $k$=1,...,$m$, and $w_k$ is the weight of the k-th attribute under the constraints that $w_k \in R_0^+$ and $\sum_{k=1}^{n} w_k = 1$.

### 4.4 Problem Statement

For a composite service CS with n component services $\{S_1, \ldots, S_n\}$ and with $m$ QoS constraints $\{c_1, \ldots, c_m\}$, the goal of negotiation-based service composition is to find service candidates such that

(1) The utility for the composite service $U(CS)$ is maximized
(2) The aggregated QoS values meets the global constraint: $q_k(CS) \le c_k$, $(k=1,\ldots,m)$.

## 5 A Negotiation Based Method for Service Composition

In this section, we present the negotiation based method for service composition. In our method, to create a composite service, a service broker negotiates with every service candidate in each service class and evaluates each service candidate individually. As such, we decompose the global utility function and the global QoS constraints of a composite service to a local utility function and the local QoS constraints for each service class using the algorithm proposed by Alrifai et al [3].

When a service broker negotiates with a service candidate, after a certain time period, the service broker selects the best available proposal. Then, the broker gathers all the negotiation information of each service class and makes some adjustments on local constraints to increase the utility. The adjustments are made based on the probability for the adjusted value to be accepted by a service provider. The probability is estimated using the Bayes theorem. After the adjustment, the service broker will negotiate with the service candidates again. This negotiation process is repeated until no further improvement can be made. Figure 3 summarizes the method we propose.

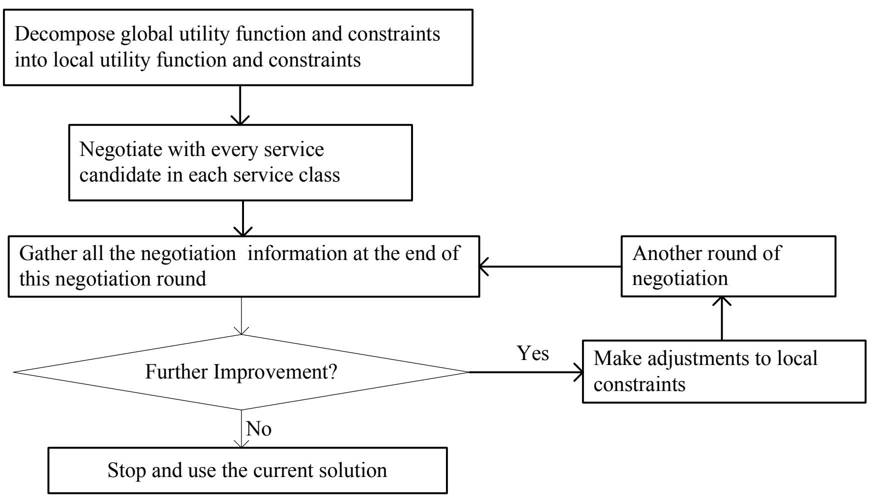



**Fig. 3.** A general method for negotiation based service composition

### 5.1 Prelimilary -- Decomposition of Global Utility Function and Global Constraints

We decompose the global QoS utility and constraints based on the method proposed by Alrifai et al [3]. For the sequential model, we have

$$
\begin{aligned}
U(CS) &= \sum_{k=1}^{m} \frac{Q'_{\max}(k) - q_k(CS)}{Q'_{\max}(k) - Q'_{\min}(k)} \times w_k \\
&= \sum_{i=1}^{n} \sum_{k=1}^{m} \frac{Q_{\max}(i,k) - q_k(S_i)}{Q'_{\max}(k) - Q'_{\min}(k)} \times w_k \quad , \\
&= \sum_{i=1}^{n} U'_{local}(S_i)
\end{aligned}
\tag{13}
$$

where $U'_{local}(S_i)$ is a modified local utility function for service class $S_i$.

$$
U'_{local}(S_i) = \sum_{k=1}^{m} \frac{Q_{\max}(i,k) - q_k(S_i)}{Q'_{\max}(k) - Q'_{\min}(k)} \times w_k \, , \tag{14}
$$

The utility function for a service class $S_i$ can be computed locally using the above formula. As the total utility of a composite service is a summation of the modified local utility of all service classes, the total utility is increased when the local utility of a service class increases.

A global constraint $c_k$ for the $k$-th attribute can be decomposed as follows:

1. Compute the difference between the maximum aggregated value of the composite service and the global constraint value.

$$
d_k = Q'_{max}(k) - c_k \, , \tag{15}
$$

2. Set the local constraint based on the relative distance between the local maximum and minimum QoS value. $c_k(i)$ represents the local constraints of the $k$-th attribute for service class $S_i$.

$$
c_k(i) = Q_{\max}(i,k) - d_k \times \frac{Q_{\max}(i,k) - Q_{\min}(i,k)}{\sum_{i=1}^{n} (Q_{\max}(i,k) - Q_{\min}(i,k))} \quad , \tag{16}
$$

where $i = 1,...,n$, $k=1,...,m$.

### 5.2 A Probabilistic Negotiation Model for Local Selection

In our method, a service broker negotiates with each service candidate individually. When a negotiation process starts, a service broker sends an initial proposal to all the service candidates in a service class. The initial proposal is a proposal with the maximum utility determined from the information available in the service registry. Every time when the broker receives a counter proposal from a service provider, the broker generates another alternative proposal and sends it to the provider until the proposal is accepted[1] or the deadline has approached for this round of negotiation. An alternative proposal sent by the broker is produced to increase the utility value and the acceptance probability.

[1] Note that a proposal that a service provider agrees to accept in a negotiation round may not be final. The final agreement is reached at the end of negotiation as discussed in Section 5.3.

The acceptance probability of a proposal is estimated according to Bayes theorem [18] as follows:

$$\Pr(\text{accept}|\text{o}) = \Pr(\text{o}|\text{accept}) \times \Pr(\text{accept})/\Pr(\text{o}) \,, \tag{17}$$

where $Pr(accept)$ is the prior probability that any proposal is accepted; $Pr(o|accept)$ is the conditional probability of seeing the proposal $o$ in all the accepted proposals; $Pr(o)$ is the marginal probability of the proposal $o$; and $Pr(accept|o)$ is the posterior probability for proposal $o$ to be accepted.

Suppose there are $m$ attributes to negotiate in a proposal. We use $o=[q_1(o),\ldots,q_m(o)]$ to represent a proposal and $q_k(o)$ to represent the $k$-th attribute of the proposal $o$. Then,

$$\Pr(o \mid accept) = \prod_{i=1}^{m} \Pr(q_i \mid accept) \,, \tag{18}$$

$$\Pr(o) = \Pr(o \mid accept) \times \Pr(accept) + \Pr(o \mid reject) \times \Pr(reject) \,, \tag{19}$$

$$\Pr(accept \mid o) = \frac{\prod_{i=1}^{m} \Pr(q_i \mid accept) \times \Pr(accept)}{\prod_{i=1}^{m} \Pr(q_i \mid accept) \times \Pr(accept) + \prod_{i=1}^{m} \Pr(q_i \mid reject) \times \Pr(reject)} \,, \tag{20}$$

When a counter proposal from a service provider is received by the broker, the broker uses the strategy of either concession or trade-off [12, 13] to generate a set of feasible proposals that meet the constraints $q_k(o) \le c_k(i)$ of all the local QoS attributes $k$=1, …, $m$. The feasible proposals are ranked based on both the utility and the acceptance probability according to

$$\text{Rank(o)} = \text{w}_\text{o} \times \text{U(o)} + \text{w}_\text{p} \times \Pr(\text{accept}|\text{o}) \,, \tag{21}$$

where $w_o$ is the weight of the utility value of a proposal $o$ and $w_p$ is the weight of the acceptance probability of the proposal $o$ and $w_o + w_p = 1$. The proposal with the highest rank is used as the new alternative proposal.

The utility of a proposal $o$ in a service class $S_i$ is computed as follows:

$$U(o) = \sum_{k=1}^{m} \frac{Q_{\max}(i,k) - q_k(o)}{Q'_{\max}(k) - Q'_{\min}(k)} \times w_k \,, \tag{22}$$

### 5.3 Adjustment of Local Constraints

A negotiation process can have many rounds. We have discussed the negotiation mechanism for a single round in Section 5.2. Now we present the method for adjusting local constraints at the end of each around.

We assume that at the end of each round, for each service class, at least one service provider has agreed to accept one proposal sent by the broker. In order to extend the negotiation space for the next round, adjustments can be made to the local constraints of an attribute when a constraint of the attribute in one service class is easier to achieve than that in another service class. The local constraints for an attribute in some service classes can be tightened while the corresponding constraints in another service classes can be loosened to keep the global constraints unchanged.

The adjustment is made as follows. A difference between a constraint and the corresponding attribute of an accepted proposal is considered as a saving. The total saving $\sigma_k$ of the $k$-th attribute for the composite service is computed as follows [3].

$$\sigma_k = \sum_{i=1}^{n} (c_k(i) - q_k(o_i)) \ , \tag{23}$$

where $k=1,...,m$ and $o_i$ is the best proposal proposed by the broker for service class $S_i$ that is accepted by a service provider.

We make adjustments on every attribute based on the probability that the constraint is acceptable to a service provider after being adjusted. Thus, we revise current local constraints as follows:

$$c_k(i) = q_k(o_i) + \sigma_k \times \frac{\Pr(accept \mid q_k(o_i) < \mathrm{q}_k(o) \le Q_{\max}(i,k))}{\sum_{i=1}^{n} \Pr(accept \mid q_k(o_i) < q_k(o) \le Q_{\max}(i,k))} \ , \tag{24}$$

where $\Pr(accept|q_k(o_i) < q_k(o) \le Q_{max}(i, k))$ represents the probability for a proposal to be accepted again by the same provider on the condition that the $k$-th attribute of the proposal falls into the range from the accepted value to the maximum value in service class $S_i$.

The above adjustments loosen the local constraints difficult to satisfy in a service class in order to increase the number of the feasible proposals in the next round of negotiation and thus the likelihood for the proposals with a better utility value.

Note that the negotiation process ends when no further improvements can be achieved for the local utility or the global utility has reached a predefined threshold.

## 6 Conclusions

Advances in Service-Oriented Architecture have presented a new paradigm for distributed computing. With Service-Oriented Architecture, complex business processes can be automated through orchestration of services to offer new business value. Given that many services are available with a same function and different QoS properties and service users have different QoS preferences, it is important to orchestrate services dynamically to meet the individual request from users.

In order to inject dynamics and flexibility into service composition, we have proposed a negotiation based method in this paper. Our method consists of several steps. First, the global utility functions and constraints are decomposed into a number of local ones for each service class. A Bayesian based model is applied in generating alternative proposals. Then the QoS constraints are adjusted based on the information of local negotiation in order to improve the negotiation outcome.

The main contribution of this paper lies in incorporating negotiation into service composition and proposing an algorithm to achieve a maximum utility by iterative adjustment of local constraints based on the outcomes of negotiation. Our approach is significant in that it enables more flexible service composition to meet different QoS requirements of different users.

We are in the process of extending our work in several directions. We will evaluate our method through simulation experiments. Furthermore, we will extend our method to support more complex composition model consisting of not only the sequential structures but also parallel structures, conditional structures, and loops.

## Acknowledgment

This research is supported substantially by City University of Hong Kong Start-up Grant (Project Number 7200102), Strategic Research Grant (Project Number 7002257), and partially by a General Research Fund (Project Number 9041467) of Research Grant Council, Hong Kong SAR Government.

## References

1. Aggarwal, R., Verma, K., Miller, J., Milnor, W.: Constraint driven Web service composition in METEOR-S. In: Proceedings of the IEEE International Conference on Service Computing, pp. 23–30 (2004)
2. Alrifai, M., Risse, T.: Combining global optimization with local selection for efficient QoS-aware service composition. In: Proceedings of the 18th international conference on World Wide Web, pp. 881–890 (2009)
3. Alrifai, M., Risse, T., Dolog, P., Nejdl, W.: A scalable approach for QoS-based web service selection. LNCS, vol. 5472, pp. 190–199. springer, Heidelberg (2009)
4. Ardagna, D., Pernici, B.: Global and Local QoS Guarantee in Web Service Selection. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 32–46. Springer, Heidelberg (2006)
5. Ardagna, D., Pernici, B.: Adaptive service composition in flexible processes. IEEE Transactions on Software Engineering 33(6), 369–384 (2007)
6. Benatallah, B., Sheng, Q.Z., Ngu, A.H.H., Dumas, M.: Declarative composition and peer-to-peer provisioning of dynamic web services. In: Proceedings of the International Conference on Data Engineering, pp. 297–308 (2002)
7. Brzostowski, J., Kowalczyk, R.: On possibilistic case-based reasoning for selecting partners for multi-attribute agent negotiation. In: The 4th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2005), pp. 273–279 (2005)
8. Bui, H., Venkatesh, S., Kieronska, D.: Learning other agents' preferences in multi-agent negotiation using the Bayesian classifier. International Journal of Cooperative Information Systems 8(4), 275–293 (1999)
9. Carbonneau, R., Kersten, G.E., Vahidov, R.: Predicting opponent's moves in electronic negotiations using neural networks. Expert Systems with Applications 34, 1266–1273 (2008)
10. Cheng, C.B., Chan, H.C.C., Lin, K.C.: Intelligent agents for e-marketplace: Nego-tiation with issue trade-offs by fuzzy inference systems. Decision Support Systems 42, 626–638 (2006)
11. Erl., T.: Service-Oriented Architecture: Concept, Technology, and Design. Prentice Hall, Englewood Cliffs (2005)
12. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Journal of Robotics and Autonomous Systems 24(3–4), 159–182 (1998)

13. Faratin, P., Sierra, C., Jennings, N.R.: Using similarity criteria to make issue trade-offs in automated negotiations. Artificial Intelligence 142(2), 205–237 (2002)
14. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. Management Information Systems Quarterly 28(1), 75–106 (2004)
15. Hwang, C.L., Yoon, K.: Multiple Attributes Decision Making. Lecture Notes in Economics and Mathematical Systems. Springer, Heidelberg (1981)
16. Keeney, R., Raiffa, A.: Decisions with Multiple Objectives: Preference and Value Tradeoffs. Wiley, New York (1976)
17. Krafzig, D., Banke, K., Slama., D.: Enterprise SOA: Service Oriented Architecture Best Practices. Prentice-Hall, Englewood Cliffs (2005)
18. Lau, R., Li, Y., Song, D., Kwok, R.: Knowledge discovery for adaptive negotiation agents in e-marketplaces. Decision Support Systems 45(2), 310–323 (2008)
19. Matos, N., Sierra, C., Jennings, N.R.: Determining successful negotiation strategies: An evolutionary approach. In: Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS 1998), pp. 182–189 (1998)
20. Menasce, D.A.: QoS Issues in Web Services. IEEE Internet Computing 6, 72–75 (2002)
21. Napoli, C.D.: Software agents to enable service composition through negotiation. Knowledge Processing and Decision Making in Agent-Based Systems 170, 275–296 (2009)
22. Oliver, J.R.: A machine learning approach to automated negotiation and prospects for electronic commerce. Journal of Management Information Systems 13(3), 82–112 (1997)
23. Oprea, M.: An adaptive negotiation model for agent-based electronic commerce. Studies in Informatics and Control 11(3), 271–279 (2002)
24. Paurobally, S., Tamma, V., Wooldrdige, M.: A framework for web service negotia-tion. ACM Transactions on Autonomous and Adaptive Systems 2(14) (2007)
25. Ran, S.: A model for web services discovery with QoS. ACM SIGecom Exchanges 4(1), 1–10 (2003)
26. Yu, T., Zhang, Y., Lin, K.-J.: Efficient algorithms for web services selection with end-to-end QoS constraints. ACM Transactions on the Web 1, 1 (2007)
27. Zeng, D., Sycara, K.: Bayesian learning in negotiation. International Journal of Human–Computer Studies 48(1), 125–141 (1998)
28. Zeng, L., Benatallah, B., Dumas, M., Kalagnamam, J., Chang, H.: QoS-aware mid-dleware for web services composition. IEEE Transactions on Software Engineering 30(5), 311–327 (2004)
29. Zhao, J.L., Tanniru, M., Zhang, L.J.: Services computing as the foundation of enter-prise agility: Overview of recent advances and introduction to the special issue. Information Systems Frontiers 9(1), 1–8 (2007)

# Usability through System-User Collaboration*
## Deriving Design Principles for Greater ERP Usability

Tamara Babaian, Wendy Lucas, Jennifer Xu, and Heikki Topi

Bentley University, Waltham, MA 02452, USA
{tbabaian,wlucas,jxu,htopi}@bentley.edu

**Abstract.** Enterprise Resource Planning (ERP) systems have become essential in industry, yet the potential value created through system use can be illusive due to poor usability. Extensive interviews with users revealed that the underlying complexity of these systems manifests itself in unintuitive interfaces that are challenging to use. Given the lack of progress made with traditional design approaches, we propose a different tactic based on a system-user collaborative approach. This entails that the system acts as a collaborative partner by sharing knowledge, providing task-specific support, and adapting to user behaviors. Based on this collaborative view, we derive a set of principles for guiding the design of ERP systems and provide concrete examples demonstrating (1) how a lack of collaborativeness contributes to various usability problems, and (2) how our proposed design principles can be used to enhance the collaborativeness and, hence, the usability of ERP systems.

## 1 Introduction and Motivation

Enterprise Resource Planning (ERP) systems are widely employed in industry to integrate various business processes. While this integration has the potential to provide tremendous operational value, using these systems can be a challenge for novices and even experienced users. ERP interfaces are typically unintuitive, presenting an abundance of information reflecting the underlying complexity of the processes around which they are built. The poor usability of these systems has been noted in industry reports [13,14] and field studies on usage [15,25,7].

The lack of progress in addressing the usability of ERP systems has motivated our interest in this topic. The prevailing theme in user interface design is the human-centered paradigm, with its emphasis on knowing the user. While user-based methods work well for uncovering usability problems [8], they typically focus on a narrow scope of specific features of the existing implementation. This tends to lead to localized fixes rather than system-wide alterations of the design [19]. This is particularly problematic for ERP systems, whose broad scope and

* This material is based in part upon work supported by the National Science Foundation under Grant No. 0819333. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 394–409, 2010.

integrated functionality require the use of multiple system features by multiple users for achieving comprehensive goals.

We propose that viewing system-user interactions through a "collaboration lens" affords a novel perspective that is advantageous for improving usability. The human-computer collaboration paradigm specifies that the system must act as a partner to its users by supporting them in the increasingly complex environments of modern applications [11]. This changes the dynamic from the user being the only one with responsibilities and knowledge about the process to one in which the system is called upon to do its part. Note that this approach is different from Computer-Supported Cooperative Work (CSCW), which is concerned with computing technology that supports human collaboration.

The work presented here is part of a multi-method research project for addressing ERP system usability that uses collaboration theory as a unifying framework. Components of this project include conducting field studies for identifying usability issues, modeling usability based on collaboration theory [7], developing software artifacts for addressing usability limitations identified in field studies [2], and designing an infrastructure that supports input logging for use in evaluating proposed design interventions [1].

In this paper, we make the following contributions:

- demonstrate how to use the theory of collaboration to derive novel design principles for improving the usability of enterprise systems,
- highlight usability problems in ERP systems based on findings from our interviews and observations in the field, and
- link the collaborative properties of a system to usability using empirical data and theory.

The next section of this paper describes our theoretical framework and related work. We then examine the link between usability and collaboration as revealed by concrete examples from a field study of ERP users at three organizations. Next, we derive a set of design principles based on characteristic properties of ERP systems and illustrate how they can be applied for achieving greater usability. We conclude with a discussion and directions for future work.

## 2 Theoretical Framework and Related Work

### 2.1 Human-Computer Collaboration

The collaboration paradigm of human-computer interaction (HCI) [24] views the interaction between a system and its user as a process in which they work together to achieve shared goals. There are various philosophical accounts [23,4] and computational frameworks (e.g. [12,6,17]) of collaboration involving humans and/or computer agents. Terveen's review article [24] summarizes several different approaches to modeling collaboration in interfaces. Terveen identifies the following key issues as being present in virtually all of these approaches:

1. An agreed-upon goal of collaboration (often referred to as the collaborative activity). The specification of the goal may not be complete at the onset. During the collaboration, the parties gradually explore and decide on the essential details.
2. Plans for performing the activity, division of tasks between the parties, and coordination. As with the goal, such plans may be only partially specified initially and evolve with time.
3. Shared context. The parties must be aware of the progress towards the goal.
4. Communication. The parties must share information and communicate to establish their goals, allocate tasks, etc. Observation of the other partners' activities and behaviors is also essential.
5. Adaptation and learning. Effective collaboration leads to partners learning about each other and adapting to each other in order to maximize the success of their joint efforts.

Terveen distinguishes human-complementary from human-emulation approaches for implementing system-user collaboration. The latter is focused on developing human-like abilities in the system's interface by, in particular, communicating via natural language and modeling and recognizing the mental state of the human user, including his beliefs, goals, and plans. The human-complementary approach, on the other hand, recognizes the fundamental difference in the natural strengths of humans and computers and aims to make "the computer a more intelligent partner" (page 73) by means that leverage the natural strengths of each party. Our work presented here falls into the class of human-complementary solutions.

Grosz [11] distinguishes two ways in which the formal theoretical frameworks of collaborating agents are applied in the design of software: (1) using the theoretical framework directly as a formal specification that prescribes the constraints on the system's behavior, and (2) as a design guide that provides an "insight" into relevant aspects of successful system-user collaboration at the design stage. Grosz argues that effective human-computer collaboration does not require human-like abilities in the interface but can be brought upon by different mechanisms. She calls for investigating approaches in design that strengthen the collaborative properties of system interfaces and solving computational research problems that arise in implementing such approaches.

The theory of collaboration guides our design approach by serving as a lens through which system-user interactions are viewed. In particular, we follow the philosophical view of Bratman [4] and the SharedPlans [12] mathematical model of collaborative action expressed in the form of a logic. Throughout this paper, when we refer to the theory of collaboration, we are referring to these two theories.

Grosz summarizes the SharedPlans model in [11, page 536]:

> "Translated into English, the definition states that for a group activity to be collaborative, the participants must have (1) intentions that the group perform the group activity; (2) mutual belief of a recipe; (3) individual or group plans for the constituent subactions of the recipe; and (4) intentions that their collaborators (fellow group members) succeed in doing the constituent subactions."

A *recipe* refers to how to perform the activity, as agreed upon by all participants. The specification implicitly requires that participants communicate as necessary to share their knowledge in order to establish mutual belief in the overall recipe, form individual and group plans, or provide helpful information. Partners must also maintain some knowledge of the context of their interaction for that interaction to be efficient. Clause (4) above also implies that subtasks are assigned according to the collaborators' capabilities, and that partners must be committed to helping each other when the success of their joint activity requires it [12,11]. Note that the collaboration we consider in this work involves only two partners, the system and its user, and no subgroups are involved.

The SharedPlans formulation is consistent with the principles Bratman identifies as required for a joint activity to be a collaboration. His requirement of *commitment to the joint activity* implies that the parties have intentions to successfully perform the activity together. It also captures intentions to refine the group and individual plans for the activity, share information whenever necessary and, overall, act in a way that leads to the success of the collaborative enterprise. The *mutual responsiveness* requirement states that the partners must adjust their own behaviors based on the actions and intentions of their collaborators in a way that facilitates achieving their joint goal. *Commitment to mutual support* further requires that all parties be ready to help a partner who is having difficulty with her portion of the activity if they can provide such help.

The key parameters summarized by Terveen are present in both SharedPlans and in Bratman's account, though "packaged" differently. This allows us to refer to Terveen's concepts in our exposition throughout this paper, while employing the more elaborate and nuanced specifications of SharedPlans and Bratman where necessary. We must note that those two theories also address important collaboration-related phenomena that extend beyond Terveen's list of five.

Software artifacts that embody collaborative behavior in some form have been implemented for a variety of domains (e.g.[21,5,3]). However, we are not aware of any applications of the collaboration paradigm to large-scale multiuser organizational systems, such as ERP systems, other than our own work.

### 2.2 Usability and Design Principles for Enhancing Usability

We rely on one of the most widely accepted definitions of usability, which is based on the ISO standard 9241-11. It defines usability as the "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." [16, page 2]. While there are other definitions of usability (e.g. [9, page 4], [10, page 300], [22]), they are fundamentally consistent with the core elements of the ISO definition.

The three core terms are defined as follows: effectiveness is specified as "accuracy and completeness with which users achieve specified goals;" efficiency refers to "resources expended in relation to the accuracy and completeness with which users achieve goals;" and satisfaction is "freedom from discomfort, and positive attitudes towards the use of the product." For example, if the user's goal in a specific ERP context is to complete purchase requisitions, effectiveness refers to the extent to which the finished purchase requisitions reflect the intended purchases accurately and are complete; efficiency refers to the number of purchase requisitions completed within a unit of time; and satisfaction refers to the extent to which the user is able to complete the task without discomfort and with positive attitudes regarding the process of using the system. Three other elements of critical importance to this definition of usability are "specified users," "specified goals," and a "specified context of use." Any concrete way to measure effectiveness, efficiency and satisfaction will produce different results depending on the user – goal – context of use combination.

A great deal of literature – both academic and practitioner – exists on design guidelines and principles for enhancing usability. This work can be divided into two streams: one focusing on characteristics of the artifact and the other on the design activity itself. Since our work concerns the artifact, we consider only the former stream here.

Polson and Lewis [20] present a set of design principles for applications that would allow "successful guessing." These highly regarded principles were based on their CE+ theory concerning the learnability of a system and are aimed at walk-up-and-use interfaces. While we are also deriving design principles from theory, a key difference is our focus on interfaces for handling complex processes.

The work that is most relevant to our own is Nielsen's [18] time-tested and well-known set of usability heuristics, which can be viewed as both design guidelines and a way to define systems in terms of concrete indicators of specific characteristics. Nielsen analyzed seven sets of well-known usability heuristics and used principle components factor analysis to extract nine factors, essentially integrating the heuristics into a set of design principles based on their ability to "explain" usability problems. Each heuristic is listed below and followed by a description:

1. Visibility of system status: provide users with feedback regarding the status and progress in task performance.
2. Match between the system and the real world: use the vocabulary of terms and follow the conventions with which users are familiar.
3. User control and freedom: allow users to redo or undo actions; do not limit the actions that users can take at a specific time.
4. Consistency and standards: define and present the same things in the same way across the system.
5. Error prevention: reduce the opportunities for users to make mistakes.
6. Recognition rather than recall: make objects and options available and visible for users.
7. Flexibility and efficiency of use: allow experienced users to use accelerators of action (such as shortcuts).
8. Aesthetic and minimalist design: include only relevant information in dialogs.
9. Help users recognize, diagnose, and recover from errors: be able to detect errors, identify their sources, and provide constructive solutions.

Nielsen's usability heuristics have been widely accepted in the field of HCI and frequently cited in usability evaluation studies. It has been reported that these heuristics could help find serious usability problems that are likely to cause "major delays or [prevent] the users from completing their task" [18, page 154], resulting in low effectiveness and efficiency.

Like Nielsen, our purpose in designing guidelines is to capture the most frequently occurring usability problems and to lay a foundation for designing more usable systems. Our principles, however, are derived theoretically from collaboration theory, which can be used to explain Nielsen's principles and also to provide another view of human-computer interactions that may reveal a wider range of usability issues.

## 3 Lessons From the Field

In this section, we present lessons learned from our field studies. We interviewed 33 employees at three organizations located in a northeastern U.S. state during fall 2008 and spring 2009. These organizations represented different industry sectors (IT, property management, and medical device manufacturing), used different ERP systems, and had varying levels of system experience. The interviews were semi-structured and conducted with a set of interview questions that were designed to learn users' perceptions of and experience with ERP systems through the lens of collaboration. Interviews were audio recorded and then transcribed and coded. Using examples from these interviews, we reveal the link between collaboration and usability by demonstrating how the absence of key properties of system-user collaboration [24] can be viewed as violations of several of Nielsen's heuristics [18].

**Shared goal.** Successful system-user collaboration requires that each party in the collaborative activity knows the shared goal. Since ERP systems are designed to assist users, the shared goal between the system and a user is often equivalent to the user's goal. Thus, it is important for the user to be clear about what the business goal is before performing any task. This is illustrated by the following comment from a superuser (an experienced user who has a strong understanding of how the data and processes are related and routinely uses different components of the system):

> *User 1: And the first thing is [to] forget about the system for a sec. Keep your keyboard away, discuss as a group, individually, collectively, whatever, what are you trying to do conceptually. Then execute.*

**The plan to achieve the goal.** Even though users know the business goal and the logical steps in the plan for achieving it, those steps may not be easily mapped to the functions provided by ERP systems. The systems' interfaces, which are designed for a wide range of business processes, contain various functions, menu items, instructions, clickable icons and buttons, etc. The intrinsic complexity of the interrelated business processes coupled with complex interfaces makes it difficult for users to perform even a very simple action like locating a function.

Moreover, many tasks involve a series of steps, screens, functions, forms, and data. The ERP systems in our study did not provide navigational or procedural guidance through these processes:

> *User 2: It [the system] doesn't tell you what steps to take next. You have to basically know what the next step is for your process, for what your job title is to do.*

As a result, users spend a significant amount of time in training, communicating with colleagues, and using trial and error approaches to learn the steps for their tasks. These steps are described by users as "unintuitive," as they often do not match the logical steps that users associate with the business processes. As a result, users frequently create "cheat sheets" that document the procedures and steps required for a task. With practice, users may no longer need these notes, but they continue to rely on them for performing non-routine tasks:

> *User 3: I have a little checklist, so when I do ACH payments, I just have screen charts and just little directions that I need to go back in and redo it. I have just directions on step-by-step with the screen chart. This was just so much easier.*

The need for memorization and notes is in violation of several of Nielsen's [18] usability heuristics. The match between ERP systems and the real world is not particularly good (violating heuristic #2), and, clearly, ERP systems require recall rather than supporting interactions based on recognition (in violation of heuristic #6). In addition, the lack of navigational and procedural guidance violates heuristic #1: system status is typically not available, leaving users unsure of their progress in performing a task.

**Shared context.** In an ERP system, the shared context includes the business context for the task. That contextual information may not be explicit and immediately clear to users:

> *User 4: If I had a new person in purchasing, I'd need to tell them what the company code was and how our GL chart of accounts worked and what our cost-center structure is – a lot more details in order for them to be able to enter just one invoice. And then it's spider webs off of that as far as whether it's a fixed asset or pre-paid, etc. So there's a lot more information that needs to be shared there if we had a new employee in any one of those areas.*

Actually, the system maintains the business contextual information and could easily present it to users if it had been designed to do so. This example suggests another violation of Nielsen's heuristic #2: the match between the system and the real world is not as good as it could be. This example also points to a contribution of the collaborative theory-based approach to viewing system interactions, i.e., the emphasis on sharing information between the system and its users, which is not part of Nielsen's heuristics.

Information on progress and feedback to the user are also important context that the system should provide in general and are essential when the user needs help. The system must first be able to detect and recognize that need for help, but ERP systems typically play a passive role:

*User 5: Now, the thing is that in this case, the system is not reaching out to you saying that you obviously need help. It's me having to go find it there. Just to go back to that GL account scenario, rather than just telling me you had to put something in, if it knew automatically what that one was supposed to be, and once you failed, say three times, or X times putting in the wrong one and then at that point, it would query you – you obviously need help here. And then it would send you to a help desk function.*

Furthermore, ERP systems often fail to utilize the contextual information they possess regarding the organization, user, business process, and task. An error message may simply report that there is a problem without offering any diagnoses or suggestions, or even isolating where that problem exists. While some error messages provide possible solutions, users often find them to be too general to be helpful:

*User 6: No, it doesn't tell me detail, but it tells me that it cannot be performed at this time.*
*User 2: It's just the [dinging] sound, yes! Nothing comes up and you know that you're looking at the wrong order in the wrong [location]. It doesn't come up with a pop up screen that says this is the wrong order.*

Some users try to seek solutions by reading system-provided help documents, but this is usually not a productive use of time, as the documents are often not specific to the task at hand and do not consider the context of the activity. As a result, users typically ask someone else (coworkers, superusers, IT staff) for help:

*User 5: I would just call someone because again, I have spent time trying to figure it out and go through the menu path, and I feel like I always get more lost and I'm just trying to save time, so I just usually pick up the phone and call someone.*

These examples illustrate violations of Nielsen's heuristics #1 and #9: in an error situation, the users find it impossible to use the system to identify needed status information, and the system is often unable to help users recover from errors.

**Communication.** A collaborative activity will likely fail if the parties do not maintain good communication. Communication requires sharing knowledge, which, in ERP systems, includes business data, the procedure for a task, status and progress reporting, and context. To communicate and share knowledge, the system should speak the users' language and use the vocabulary of terms with which they are familiar. However, the terminology used by some ERP systems is drastically different from the users' and little or no explanation is provided about what terms mean:

*User 5: I don't know how it's chosen that for vendors it's XK, and for purchase orders it's ME prior to the numbers. I do know, obviously, the numbering system as far as O1 is for creation, O2 is for change, and O3 is for display. But no idea what it means to the actual function!*

*User 7: Sometimes when I get an error message and I don't know why I'm getting it then that's when it's questionable about what's going on. Because usually it's in codes, and I don't understand that.*

Incomprehensible terms and error messages are in direct violation of Nielsen's heuristic #2 (the match between the system and the real world), which includes components such as "Speak the user's language" and "Contains familiar terms and natural language."

**Adaptation and learning.** To maximize the long-term success of a collaboration, each party needs to learn from and adapt to the others. In interactions with ERP systems, users are the ones who must do the adapting. An alternative would be for the system to also adjust to its users' behaviors by taking into consideration their previous actions. This would enable the system to automatically populate previously entered data, list functions in the order of frequency of use, offer an option of repeating a frequently performed task, etc. However, such capability is currently lacking:

*User 7: I don't think it does something to make it easier due to the replication of me doing something. So, if the system had enough intelligence that it noticed that I am always printing the details for all the items that are on the overall report, and then it would say let me offer you, do you want to print all of this? You seem to be doing this always.*

Nielsen's heuristics do not include any with a direct match to such behavior. He does identify "use of default values so that the user does not have to re-enter information" [18, page 153] as one of the heuristic candidates that was not included in the final set of nine. Use of default values is, of course, more narrowly defined than the broader consideration of previous actions, and this is clearly one of the areas where our approach extends those heuristics.

The above examples illustrate the lack of collaboration between ERP systems and their users, and the frequent violations of Nielsen's usability heuristics can be framed in terms of the non-collaborative behavior of these systems. A true collaboration aimed at enhancing usability requires a partnership, the absence of which is best summarized by a superuser in this vivid way:

*User 8: So with the system, it's somebody that just smirks at you. And when you make mistakes, it looks at your with the same kind of dopey look on its face . . . And it starts forcing you to kind of work around and work over and work under. And that's the frustrating part about it that again is the biggest pain in the backside. So, I would say, it's not a good partner, I feel like it's an impassive sometimes uncooperative coworker.*

# 4 Design Principles

In this section, we first analyze the central constructs of collaboration theory as they apply to system-user interaction in the ERP domain. We then derive design principles for making the system a better collaborative partner, based on our extensive field study of ERP usage and the theoretical framework outlined in section 2.1.

## 4.1 System-User Collaboration in the ERP Domain

A shared goal and intentions towards it are the prerequisites to any collaboration; thus, we review these concepts in the context of the ERP domain. The overall **goal** of this system-user collaboration is to automate the management of data related to the business processes within an organization for achieving greater organizational efficiency. This high-level goal can be decomposed into smaller, more specific activities (such as fulfilling a customer order) and even further down into transactions involving an individual user and the system working on a particular task. While the set of tasks an individual user is exposed to is limited, they are part of a chain of tasks that correspond to the components of a business process.

**Intentions towards collaboration.** We view the process of users collaborating with ERP systems as being similar to the collaboration that occurs between co-workers in an organization. While co-workers can choose not to participate in the organization's processes, this option is rarely taken due to the likely negative consequences. Similarly, employees are motivated to use the ERP system for both contractual reasons as well as for benefits derived from such use. A further simplification is that, unlike a human co-worker, the system does not have any competing intentions; all of its time and resources are devoted to its users.

Given the proper intentions towards the shared goal, it is important to consider the **knowledge** and **abilities** of the collaborators for the optimal division of labor according to each partners' strengths. In terms of **knowledge** and **abilities**, an ERP system is an embodiment of widely generalized organizational practices. It has superb capacity for storing, organizing, retrieving, and visualizing organizational data. An employee has partial knowledge of the organization's operations and business transactions, business practices, and associated data. It is important to realize that, while this knowledge depends on the employee's role, it is always incomplete.

## 4.2 Designing for Collaborative System-User Interactions

**Knowing the plan and communication.** To engage in a successful collaboration in performing an activity, the system and its users must be aware of the *overall recipe*. The system's *communication* to the user regarding the steps to be taken includes textual and pictorial labels on input fields and buttons; components used for navigation, such as menus and lists of transactions; and instructions and other text provided in dialogs and error messages. When users are familiar with the steps involved in completing a task, they are quite efficient at using the system to do so. However, the learning process is lengthy and characterized by negative terms such as “brutal” and “intimidating.” This is due to multiple factors, including the mismatch between the users’ and the system’s vocabularies and the generic nature of the interfaces, which do not reflect the practices with which users are familiar. From the collaborative standpoint, the user is the one forced to take on the burden of learning to speak the system’s language, utilize the necessary functionality, and navigate to the appropriate interfaces.

The complexity of the learning process and the overall effort expended by the user would be greatly reduced if the system took part of this process on itself. For example, having labels on menu options, transaction names, input fields, etc. be consistent with the organizational vocabulary would greatly improve the user’s understanding and confidence. Furthermore, the graphical interface could be customized to include only those input components that are essential for the organization. This latter kind of optimization is sometimes done in practice but is typically avoided because of initial costs and, more importantly, incompatibilities with later versions of the ERP software that will incur future costs. These considerations lead us to our first design principle (henceforth abbreviated DP):

*DP1. The user interface should provide a mechanism for customizing the vocabulary of terms used by the system in its communication to the user, the composition of business transactions, and the content of the system’s informational output to match the practices of the organization. There should be a mechanism for incorporating the customizations from an earlier version of the system to a later one.*

This design principle does not prescribe a particular method of customization. For example, it can be done using machine learning techniques that draw on the history of system-user interactions, or can be performed manually, or can be achieved using some combination of the two. While we are working on developing effective methods (algorithms, representations) and design sketches for implementing our design principles, they are beyond the scope of this paper.

To further aid a novice or an infrequent user in understanding the steps required to complete a task, the system should provide navigational support and information on progress in completing a process. From the collaboration standpoint, this sort of explanatory guidance is required, since the system is the one with complete knowledge of the relationships between the data, the process, and the interface components, and must share any knowledge that the human partner needs to perform her part.

*DP2. The system should provide navigational and progress guidance to a user performing a transaction, indicating the broader context of each interaction in terms of the related business process components and specifying the completed and remaining parts. A sufficiently competent user should be able to turn off this guidance if it becomes a distraction.*

**Commitment to helping a partner in need.** When something goes wrong during an interaction and an error is signaled by the system, users often experience difficulties understanding and resolving the problem. The poor quality of the error message can be a factor, but even if the message is reasonably descriptive, the user's difficulties are often due to the following:

- Data-to-process and process-to-process relationships play a critical role in defining ERP system functions, but they are too numerous to be known in their entirety. Because of this complexity, the structures and relationships between processes and data are typically hidden from the users, leaving them unable to diagnose the causes of many even trivial problems, such as an incorrectly specified code.
- Even when a user is familiar with the business context and operation of the interfaces being used, another hurdle in diagnosing problems stems from the fact that ERP systems involve multiple users working on different parts of time-extended business processes. The processes affect different but related portions of business data, but individual users often lack an understanding of how their tasks relate to the broader process in which they are taking part. This greatly impedes their ability to diagnose and correct errors that resulted from the actions of other users in a related task interface.

As a result of the above plus the user's perception of the ERP systems as "intimidating," the most common error diagnosis and resolution strategy involves asking another person (colleague, superuser, consultant, etc.) for help. The system can be so obscure that even users who have encountered the same error before often cannot recall how to overcome it. The help function is regarded as a waste of time due to the lack of context of the information that is presented and the effort required by the user to "connect the dots." In our field studies, we have invariably encountered stories about errors that took days to diagnose.

Collaboration presupposes a commitment to completing the joint activity and helping a partner who is having a problem performing her part. When a system signals an error, it is a clear sign that it is aware of the fact that something has gone wrong. Typically, the ERP system's involvement in diagnosis and correction of errors stops at the reporting stage. In many cases, the system has access to contextual data that would explain the cause of the problem. Sometimes the solution or a set of possibly helpful actions are readily available and identifiable, but the system usually takes a passive role, leaving the burden of diagnostic discovery to the user. This behavioral pattern is primarily due to the lack of focus at the design stage to providing error diagnostics and recovery functions in the system interfaces.

Two examples illustrate our point. The first is a simple, real life case in which a user has entered a shipment date into a field. The system rejects the user's input and generates an error message stating that the date is in an incorrect format. The system waits passively for the user to correct the error. A more collaborative response would be for the system, when displaying the error message, to also bring up its calendar feature from which the date can be selected, or at least suggest the use of this feature.

An example with a less obvious solution is an error that resulted from a mismatch between the parameters of a business process entered by different users. This is the type of error that would typically be sent to a superuser to diagnose. Instead, the system could aid in the diagnosis by providing the broader context of the interaction within which the mismatch occurred: i.e., display a list of the related transactions and provide easy access to views of the related data.

Not all error situations are due to the actions of the user; for instance, a storage device failure may prevent the system from saving data. Commitment to the success of the collaborative activity requires that the system not give up until it has explored other avenues for problem resolution and consulted with the user when his agreement to a solution is required.

*DP3. When the system detects a problem, it should identify the possible causes and ways of resolving it. If the fix is obvious, the system should inform the user and perform it. If it isn't obvious, the possible causes and resolution scenarios should be presented to the user and be readily executable. If the system is unable to identify resolution strategies, it should present the user with the relevant data and transactions.*

Deciding whether to proceed with a fix to a problem with or without engaging the user depends on the nature of the problem and, sometimes, the particular user's preferences. These items should be carefully considered during the system design stage to make error resolution and diagnosis effective.

**Other helpful behaviors.** Helpful behaviors are those that increase the effectiveness and efficiency of the collaborative efforts of the parties and increase the likelihood of the success of the joint activity. As theory states, such behaviors stem from the partners' commitment to the success of the joint activity. Humans often do things that are helpful for a group effort without being explicitly asked. For example, when going to a business meeting, they take their calendar with them, knowing the group may need to schedule the next meeting. In doing so, people are using their knowledge of the task, the environment, their partners, and their commonsense reasoning abilities. There are many opportunities for designing helpful behaviors into the system based on both a priori analysis of the tasks, environment, and users, as well as the data collected during system use. An example is displaying those currencies most frequently selected by the user in prior interactions at the top of the list of world currencies.

A less straightforward example involves the wide variety of search interfaces common to ERP systems. To find a code for a specific material, for example, one can search through the entire material master, or by material by plant, or by material group. If a user invokes a search interface, instead of blindly offering a collection of tabs for all possible search options, the system can use the contextual information available to it to rank-order the options and to fill in any known details. If the user is working on an order form and has specified the plant for which the order is to be placed, a search for material by plant can be highlighted and the search interface should include the specified plant number.

*DP4. In presenting selection choices, the system should utilize what it knows about the user, the organization, the task, and the context, and provide faster access to the more likely choices than the less likely ones. Where the choice of data or action is obvious, the system should have an option of not waiting for the user to enact it. The user should have an option to replace/cancel the system's provided choice of data/action.*

## 5 Discussion

The above design principles for achieving greater usability of ERP systems by improving their collaborative strength were derived using the theory of collaboration and findings from our field studies. Quotations from those studies, presented in Section 3, highlight collaborative weaknesses of ERP systems. They also provide evidence of usability problems, many of which can be explained using Nielsen's usability heuristics.

As shown in Table 1, our proposed design principles encompass Nielsen's usability heuristics, thus supporting usability. However, the principles go far beyond merely restating and aggregating those heuristics: they provide a unified theory-based perspective that explains their utility in terms of human-computer collaboration. Furthermore, what sets our work apart from other design principles for usability is its emphasis on the system's role in using its capabilities and knowledge to maximize the effectiveness and efficiency of its use in service to the user's goals.

**Table 1.** Design principles, implied usability heuristics and underlying collaboration requirements

| Design principles | Nielsen's heuristics | Collaboration requirements |
|---|---|---|
| DP1 | 2,7,5 | (Terveen) Communication; Adaptation and Learning<br>(Bratman) Commitment to joint activity; Mutual responsiveness<br>(SharedPlans) Mutual belief of recipe |
| DP2 | 1,6,7,5 | (Terveen) Shared Context; Determining goals; Communication; Planning, allocation of responsibility and coordination<br>(Bratman) Commitment to joint activity; Mutual responsiveness<br>(SharedPlans) Mutual belief of recipe |
| DP3 | 5,9,6,7 | (Bratman) Commitment to joint activity; Commitment to mutual support<br>(SharedPlans) Intention that the collaborators succeed |
| DP4 | 6,7,8,3,5 | (Bratman) Mutual responsiveness<br>(SharedPlans) Intention that the collaborators succeed |

To illustrate our claim, we consider DP4. Relative to Nielsen's heuristics, a system that implements DP4 would enable recognition rather than recall and would have a minimalist design, in order to make likely choices easy to reach and de-emphasize or remove the irrelevant ones. The shortcut property of heuristic #7 is manifested by the system enacting the obvious choice, while the user freedom property of heuristic #8 is preserved by allowing the user to undo the system's choice and follow up with her own. Easy access to the most relevant choices reduces the chances of the user selecting the wrong one (heuristic #5). What collaboration theory *adds* to this formulation is the system's responsibility to bring to bear all its knowledge of the users, tasks, organizational practices, and context of the interaction in order to produce the most useful choices and present or enact them in an effective way.

The set of the principles presented here is not intended to be exhaustive. User interface design for usability involves considerations of varying granularity: from general design of the interaction sequences to minor details of layout and style. In our analysis, we deliberately focused on the "big picture," highlighting the aspects of the interaction that we found particularly problematic for the users. However, the principles have broad applicability and demonstrate how the theory of collaboration can be used as a design guide to address design issues at various levels of granularity.

## 6 Conclusions

The human-computer collaboration paradigm employed here has been applied in several domains, with the implicit goal of creating software that is more effective, efficient and pleasant to work with and that behaves like a user's partner. However, the link between the collaborative properties of such systems and usability has not been formally addressed before this work. Quotes from actual ERP users presented in this paper highlight the relationship between poor usability and the collaborative weaknesses of a system. We have presented design principles for improving the usability of ERP systems, derived from collaboration theory and field studies, and outlined how they address the usability shortcomings in terms of Nielsen's usability heuristics and our field observations.

While some practices that implement collaborative behaviors exist in commonly used interfaces, developing new, effective methods for the particular properties of the ERP domain is part of our on-going investigations. We are currently developing algorithms and representations to support the design principles described here and are implementing artifacts to demonstrate the application of those principles.

## References

1. Babaian, T., Lucas, W., Topi, H.: A data-driven design for deriving usability metrics. In: Proc. of ICSOFT 2007, pp. 154–159 (2007)
2. Babaian, T., Lucas, W., Topi, H.: Visualizing the process: A graph-based approach to enhancing system-user knowledge sharing. In: Proc. of ICEIS 2005, pp. 122–128 (2007)
3. Babaian, T., Grosz, B.J., Shieber, S.M.: A writer's collaborative assistant. In: Proc. of IUI 2002, pp. 7–14. ACM Press, New York (January 2002)

4. Bratman, M.E.: Shared cooperative activity. Philosophical Review 101(2), 327–341 (1992)
5. Breazeal, C., Hoffman, G., Lockerd, A.: Teaching and working with robots as a collaboration. In: Proc. of AAMAS 2004, pp. 1030–1037. IEEE Computer Society Press, Los Alamitos (2004)
6. Cohen, P., Levesque, H.: Teamwork. Nôus 25, 487–512 (1991)
7. Cooprider, J., Topi, H., Xu, J., Dias, M., Babaian, T., Lucas, W.: A collaboration model for ERP user-system interaction. In: Proc. of HICSS-10 (2010)
8. Dumas, J.S.: User-based evaluations. The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, 1093–1117 (2003)
9. Dumas, J.S., Redish, J.C.: A Practical Guide to Usability Testing. Intellect, Ltd., Bristol (1999)
10. Gould, J.D., Lewis, C.: Design for usability: Key principles and what designers think. Commun. ACM 28(3), 300–311 (1985)
11. Grosz, B.G.: Beyond mice and menus. Proceedings of the American Philosophical Society 149(4), 529–543 (2005)
12. Grosz, B.G., Kraus, S.: Collaborative plans for complex group action. Artificial Intelligence 86(2), 269–357 (1996)
13. Herbert, L.: Put business applications to the usability test. Forrester Research (2006)
14. Hestermann, C.: Key issues for enterprise resource planning. Gartner (2009)
15. Iansiti, M.: ERP end-user business productivity: A field study of SAP & Microsoft: Keystone strategy, http://download.microsoft.com/download/4/2/7/427edce8-351e-4e60-83d6-28bbf2f80d0b/KeystoneERPAssessmentWhitepaper.pdf (downloaded 12/21/2009) (2007)
16. ISO 9241-11: Ergonomics requirements for office work with visual display terminals, part 11 -guidance on usability. International Standards Organization (1998)
17. Kinny, D., Ljungberg, M., Rao, A.S., Sonenberg, E., Tidhar, G., Werner, E.: Planned team activity. In: Castelfranchi, C., Werner, E. (eds.) MAAMAW 1992. LNCS (LNAI), vol. 830. Springer, Heidelberg (1994)
18. Nielsen, J.: Enhancing the explanatory power of usability heuristics. In: CHI, pp. 152–158 (1994)
19. Norman, D.A.: Human-centered design considered harmful. Interactions 12(4), 14–19 (2005)
20. Polson, P.G., Lewis, C.H.: Theory-based design for easily learned interfaces. Hum.-Comput. Interact. 5(2), 191–220 (1990)
21. Rich, C., Sidner, C.L., Lesh, N.: Collagen: applying collaborative discourse theory to human-computer interaction. AI Mag. 22(4), 15–25 (2001)
22. Rosson, M.B., Carroll, J.M.: Usability engineering: scenario-based development of human-computer interaction. Morgan Kaufmann Publishers Inc., San Francisco (2002)
23. Searle, J.R.: Collective intentions and actions. In: Cohen, P.R., Morgan, J., Pollack, M.E. (eds.) Intentions in Communication, pp. 401–415. MIT Press, Cambridge (1990)
24. Terveen, L.G.: Overview of human-computer collaboration. Knowledge-Based Systems 8(2-3), 67–81 (1995)
25. Topi, H., Lucas, W., Babaian, T.: Identifying usability issues with an ERP implementation. In: Proc. of ICEIS 2005, pp. 128–133 (2005)

# Instruction Manual Usage: A Comparison of Younger People, Older People and People with Cognitive Disabilities

Abdusselam Selami Cifter and Hua Dong

Brunel University, School of Engineering and Design,
Uxbridge, UB8 3PH, Middlesex, United Kingdom
selami.cifter@gmail.com, hua.dong@brunel.ac.uk
http://www.inclusivedesignresearch.org/

**Abstract.** When people are faced with new products for the first time or require assistance using features, the instruction manual is a key information source and therefore the design of instruction manuals is as important as the design of the product itself. There are often situations where the design embedded in the product is not sufficient to express its usage to the user. In addition, users differ significantly from each other in terms of their needs, expectations and capabilities. The main question is "are instruction manuals accessible enough and do they consider a variety of user groups?" This paper investigates the differences between three user groups (i.e. younger people, older people and people with cognitive disabilities) regarding their approach to understanding of instruction manuals. An experimental study was carried out testing thirty volunteer participants from the aforementioned user groups, using two digital products from two different market segments and their instruction manuals.

**Keywords:** Instruction Manuals, Older People, Disabled People, User Characteristics.

## 1 Introduction

When we are faced with new products there are two main information sources to understand the usage; the first one is the product itself and the second one is instruction manuals. According to the Department of Trade and Industry [29], there are three ways to make the products safer and convenient to use, and two of them were found to be relevant to most of the everyday products. The first one is designing the products in a way that will not have any potential threat or inconvenient feature for the users. However this option is not always possible for all type of products. The other option is "to provide adequate safety information and instructions for proper use and allow for foreseeable misuse" [29].

Design of the instruction manual and design of the product itself are equally important; both of them can prevent the user from making mistakes or can lead them to misuse. However it is not always possible to predict all the usage problems through the design process, due to the complexity of users. According to Smith

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 410–425, 2010.

[28], "too often, consumers act in a way that is inconsistent with what the manufacturer intended." For example some people may not prefer to use instruction manuals and might try to understand the usage through their own understanding [28]. Well designed instruction manuals can grab the user's attention and improve the possibility of successful usage of the products. However why do we require instruction manuals to use products? Why is preparing a good instruction manual as important for both users and manufacturers? The answers to these questions were sought through this study.

### 1.1 Complexity: Technology, Designers and Users

Today designers have the opportunity to make products smaller, smarter, and cheaper than before [4]. However the devices that we use are becoming more and more sophisticated [7] and as a result people have already started to complain about complexity of interfaces [30].

Maybe the main problem is, as suggested by Margolin [24], users have become a central theme of design discourse, though there is still a large gap in the knowledge of designers in this area. Most of the designers are designing products and systems on the basis of their own taste and capabilities [22]. However users are not heterogeneous and are highly variable. Due to this diversity, they have different needs and expectations. For example, they may be suffering from age or disability related physical, perceptual or cognitive problems [14][33].

Heskett argues that [16], thanks to the developments in electronics, manufacture and software systems, today products are possibly more flexible to meet the needs of specific user groups. However technology itself is not sufficient to make products usable for these people. Products should be designed in a way that considers the user's needs, expectations and capabilities. It is not only the products, but also instruction manuals that should be designed accessibly. As suggested by the Department of Trade and Industry [29], knowing the people who are likely to use the instruction manual is crucial to produce them intelligible. Therefore designers should be aware of the diversity of the users. An inclusive design approach is a way to address these problems.

### 1.2 What Is Inclusive Design?

According to the British Standards Institute [5], "inclusive design is comprehensive, integrated design which encompasses all aspects of a product used by consumers of diverse age and capability in a wide range of contexts, throughout the product's lifecycle from conception to final disposal." By means of inclusive design, products can be designed in a way which can be accessible by people who are likely to be excluded (i.e. older people or people with disabilities) by designers.

Demographical changes towards 'aging' and recognition of 'disabled people' are the two major drivers of inclusive design at the international level [26]. Due to designing for mass production in the second half of the 20th century, an incorrect understanding was developed by designers towards standardising people to create the 'universal type' of user rather than understanding them as individuals [7]. This resulted in shortcomings in terms of user and design compatibility and inclusive design can be seen as a response to this situation [7]. Today companies have started to realise the benefits of designing products or systems which aim to include a wider variety of users, rather than designing products or systems focussed on younger users. In terms of the business case, inclusive design can provide “a better understanding of changing consumer needs, lifestyles, expectations and aspirations which can expand the consumer base, extend product lifecycles and develop brand loyalty” [5]. However there is still a lack of awareness of inclusive design and its benefits [9].

“By determining the capability demands of a product on users, it is possible to identify and quantify those who have difficulty with, or cannot use it” [5], therefore knowing the users is crucial. So what are the differences of older people and disabled people in terms of their product usage characteristics? Before describing the instruction manuals, let us consider this question.

### 1.3 Why Older People and Disabled People?

Today we can see more and more older people in the end user market particularly in developed countries. “The rapid increase in numbers of individuals who are older is also starting to provide a market "pull" towards more accessible products” [32]. Reduced birth rates and longer life expectancies are the two main factors of this situation [6]. It is expected in the UK that, by 2020 half of the adults will be over 50 or over, unless birth rates increase or younger people immigrate [5].

Older people are unique in terms of their characteristics, and their capabilities can vary significantly. They are likely to have visual impairments, low dexterity or limited mobility, also they can have decline in their cognitive capabilities regarding rapid assimilation and analysis of new information [19].

Due to the fact that older people are likely to spend most of their time at home, they are targets of home-based technologies [10] such as home-use medical devices [14]. However the multi-functionality and the complexity of the interfaces alienate older people [2]. Hence they are less likely to use technology when compared with younger people [10] due to the fear of the unknown and getting lost in confusion [15].

Another important user group is disabled people. In the UK, particularly after the Disability Discrimination Act in 1995, people have started to give more consideration on the rights of disabled people, and as a result, by means of new regulatory changes and laws, the rights of disabled people have been improved significantly [5]. These people can have congenital disability, or impairment can happen anytime, e.g. a stroke or an accident [11]. We can have temporary disabilities like, injuring or broking one of our limbs. Having or developing a permanent capability loss also brings devastating psychological or social effects, because the needs and expectations of that person changes dramatically [11].

People are generally considered in two separated groups as able bodied or disabled people by designers, which drive them to design different products for each group [6].

However if the capability demand of the product is considered carefully then the final design can be usable for both of these user groups [6].

To summarise, older people and disabled people are unique in terms of their characteristics and generally they are treated as special cases by designers. However, if the capability-demand relationship of the product is considered carefully during the design process, then the final product is more likely to include both groups of users.

### 1.4 What about Instruction Manuals?

According to Smith [28], instruction manuals are 'tools' for the users to use for carrying out tasks and they should be carefully designed to be useful tools. Instruction manuals can be considered critical to the required usage of products. This is because they are not only providing information about the functions of the product, but also safety information for users. They are part of a system which includes all product related elements, including the physical design of the product, and as a complete system they must support each other [28]. However according to Horen et al [17][18], many people end up with complaining about the manuals when they first time try to use an electronics device. They also argue that instruction manuals are inaccessible and difficult to use for many user groups particularly for older people due to their impaired capabilities [17][18].

When preparing instructions it is important to know what type of users are likely to use it, hence their characteristics should be taken into account [29]; otherwise the final product may fail in terms of the intended way of usage. When preparing instructions it is important to know what type of users are likely to use it, hence their characteristics should be taken into account [29]; otherwise the final product may fail in terms of the intended way of usage. For example regarding home-use medical devices, during the preparation of instructions one should take into account that the end-user can be: older or on medication that may interfere with memory or can have disabilities such as poor vision or hearing [1].

But what is the current situation? Are the instruction manuals designed inclusively to address the needs of different user types? Or are they only designed for the general population? To answer these questions an experimental study was carried out.

## 2 The Study

The experimental study focussed on understanding the behavioural differences of three types of user groups with respect to their instruction manual usage. The desired output of the study is:

- To present the current situation by exemplifying two instruction manuals belonging to two different off the shelf products from two different market areas.
- To understand if there are any differences between the user groups observed in terms of their approach to the understanding and use of instruction manuals.

### 2.1 Methodology

The study involved three groups of people:

- 10 able-bodied young people (Aged between 18-64)
- 10 healthy older people (65+)
- 10 people with cognitive disabilities (Aged between 18-64)

Disabled people vary significantly in terms of their capabilities. For this study only people with cognitive disabilities were considered, because they are the most likely group to experience difficulties in terms of their interaction with the products and also their instruction manuals. The disabilities of the participants were not shared with the PhD researcher in detail because all the participants were recruited through organisations. They were all mentioned as having a learning disability where some of the participants were found to be more severe than others. For instance one participant with Down's syndrome was found with severe cognitive disability, because in many situations the communication between the researcher and the participant was hard to establish. However six of the disabled participants are employees of a charity, and three of them successfully completed most of the tasks (refer to Table 1) due to their self confidence. All the other participants had moderate learning disabilities as they experienced difficulties in maintaining attention, reading and understanding, and sometimes had communication problems. The 10 older participants also demonstrated physical and sensory impairments during their interaction with the products. These impairments are age related or health related, i.e. impaired vision, hearing and dexterity, arthritis, heart problems, blood pressure problems and diabetes.

The study was conducted as product interaction trials which involved the completion of given tasks by the volunteer participants through interacting with two selected digital devices and their instruction manuals. The products used in the user observation study are a digital camera (Sony DSC-S730) and a digital automatic blood pressure monitor (Omron R7).

The study was largely descriptive, so observation was used as a primary method for capturing the outputs of the users [27]. According to the British Standards Institute [5], observation is highly effective where the aim is to identify user difficulties with products. Where some sets of human actions are complex and difficult for a single observer to describe comprehensively, video technology proves a viable method of recording [3]. Video recording methods also gave the opportunity to capture facial expressions which reflected the feelings of the participants during the study.

A pilot study was conducted with five people using convenience sampling. For the main study, quota sampling [8] was used. Whenever it proved difficult in finding enough volunteering participants, snowball sampling [27] or convenience sampling was utilised.

To recruit the participants for the study, several organisations were contacted. The organisations that agreed to provide support for this study included: Brunel

University, Charles Curran House, Yateley Industries, Age Concern Hillingdon and Mencap.

The study has been approved by the ethics committee of the School of Engineering and Design, Brunel University. An information sheet and a consent form were disseminated to the participants prior to their active participation. The participants were informed that they were free to withdraw from the study at anytime without giving reason for their decision.

### 2.2 Explanation of the Study

The product interaction trails were conducted in a quiet room with one participant each time. The typical setting is shown in Figure 1.

The participants were invited to fill in a general questionnaire before they started the trial. This questionnaire asked about their age range, gender, education level and contact details. Then they were given a task list and asked to complete the tasks by using the devices provided along with their instruction manuals.


**Fig. 1.** The typical setting of the product interaction trials

There were seven tasks in total (refer to Table 1); the first three related to the Blood Pressure Monitor (BPM); and the rest related to the digital camera. The tasks were designed to capture data which are likely to reflect the user characteristics of the participants and their experienced difficulties during the product interaction.

Once the participants had finished the tasks, they were given another questionnaire to capture the thoughts of their experience during the study. The participants were also encouraged to give any verbal feedback about their experience (e.g. thoughts, feelings about the tasks, products and their expectations). This session was video recorded to capture the participants' emotional expressions.

It was observed that older users tended to have very fragile motivation and often withdrew from the study when they encountered difficulties. Hence, after the pilot study with older participants, it was decided for the 'hidden task' (refer to Table 1) to be removed from the study involving older users.

**Table 1.** The task list

| Blood Pressure Monitor (BPM) | |
|---|---|
| **Task 1:** | **Prepare the device to be used.** *(open the protective case of the device and take the monitor device out, take the batteries out and then insert them into the device.)* |
| **Task 2:** | **Measure your blood pressure and write down the score.** *(attach the device to your wrist in the correct position as specified in the instruction manual, then switch on the device. The participants were supposed to use their elbow as a fulcrum and take the device to their heart height till hearing the beeping sound, indicating the correct height has been reached and a measurement has started. During the measurement they were expected to obtain the correct posture and sustain their position until the device deflates. Then they were asked to write their scores down.)* |
| **Task 3:** | **Switch off the device as if it will not be used for a long time.** (Turn off the device, take the batteries out and put the device back to the case.) |
| **Digital Camera** | |
| **Task 4:** | **Prepare the device to be used.** *(insert the batteries and the memory stick and then switch on the device)* |
| **Hidden Task:** | *This task was designed only for younger participants. The memory stick used for the study was left full hence the participants were expected to create space in the memory stick by erasing the pictures or formatting the card to be able to continue the following tasks. The purpose of this task was to enable the observation of the response of the participants when they encounter an unexpected situation.* |
| **Task 5:** | **Take your own picture reflected in the mirror provided. Please try to take at least one good picture.** *(direct the digital camera to the mirror and then take a picture of your own reflection. Flash was left on to motivate the participant to interact with the buttons and the digital interface. Due to the fact that flash will spoil the picture, the participants were asked to try to take at least one good picture. This will allow the PhD researcher to observe the participants' reaction to an unexpected situation. The decision was up to the participants therefore when they thought that the picture was good then they were free to pass to the next task.)* |
| **Task 6:** | **Take a picture of the toy car. Please try to take at least one good picture.** *(take the picture of the small toy car provided. The participant has to frame and focus well in order to take a good picture).* |
| **Task 7:** | **Erase the unwanted pictures and switch off the device.** *(the participants were asked to leave two pictures in the camera: one from Task 5 and the other from Task 6). If they had taken more than one picture on any task, they were asked to erase one of them.)* |

### 2.3 An Overview of the Instruction Manuals Used in the Study

The instruction manuals used in the study is shown in Figure 2.

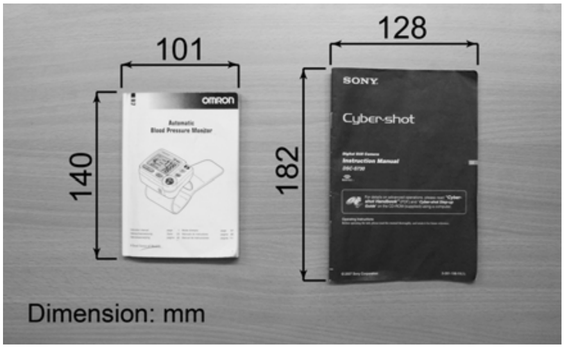



**Fig. 2.** The instruction manuals of the products and their dimensions

**The Blood Pressure Monitor.** The layout is black and white. There are 132 pages in total. It includes foreign languages (English, German, Dutch, French, Italian and Spanish), although English is the first part and covers pages 1 to 22. The explanations include text, symbols, figures and tables. Due to the fact that this product is a medical device, in some parts of the manual it gives some medical information (e.g. what is blood pressure?) which includes terminology and description. Sans serif typeface is used and the font size of the text based explanations is 6 point.

**The digital camera.** The layout is black, white with tones of blue. It is 30 pages in total and is written in English only. The explanations include text, symbols, figures and tables. Both Sans Serif and Serif Typefaces are used. For the headings a Sans Serif Typeface is used however for the explanations, both Sans Serif and Serif Typefaces are used. A Sans Serif Typeface is used for writing the main actions and to give more information about how to do it, a Serif Typeface is used in body text (Figure 5). Font sizes vary, however 7 and 8 point font sizes are used for the text based explanations.

## 3 Results

As can be seen from Figure 3, the results differ for each user group. Overall, the younger participants proved to have the most successful results with both of the devices tested. However, the test also showed that they also had better results with the digital camera tasks when compared with the BPM ones. Task 2 (refer to Table 1) was identified as the least successful task for them with only 4 successful completions out of the sample of 10.

In contrast, older participants have better results with the BPM; however they displayed the poorest performance with the digital camera in all of the user groups. As mentioned previously, the ‘hidden task’ was removed for the older user group due to motivational issues. Task 7 (refer to Table 1) for the older user group is identified as the least successful task with only 1 successful completion.

Disabled participants also have better results with the digital camera. However they were the worst in interacting with the BPM among all the user groups. None of the disabled participants could complete Task 2 (refer to Table 1) successfully.

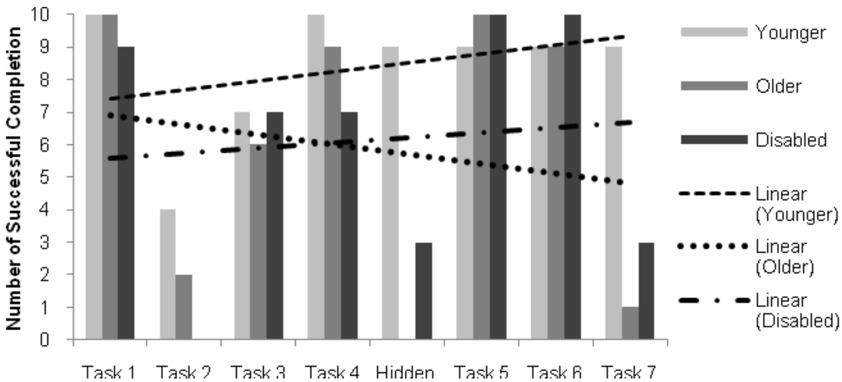



**Fig. 3.** Number of successful completion for each task and the user groups

Participants also showed differences in term of their preference to either using the instruction manuals or not for each of the tasks. Figure 4 shows the number of participants that used the instruction manuals for each task and from each of the user groups.

Most of the younger participants did not prefer to use the instruction manuals during the digital camera tasks; however as can be seen from Figure 3, they displayed the most successful results. Even though task 2 was the least successful task for the younger participants, it showed the highest score for the task where the instruction manual was used the most.

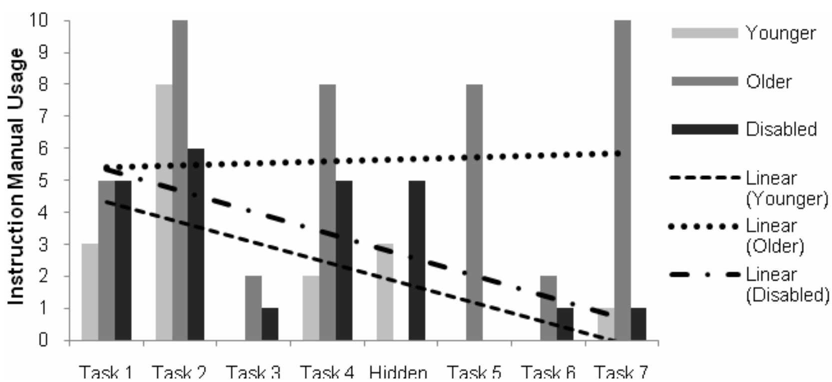



**Fig. 4.** Number of people referred to the instruction manuals during the tasks

Older participants used the instruction manuals more than all the other user groups. Particularly during Task 2 and Task 7, all of the older participants preferred to use the instructions, even though these were their two least successful tasks, particularly task 7. During task 5 (refer to table 1), a total of 8 older participants used instruction manual where none of the younger participants and disabled participants referred to the manual.

Half of the disabled participants did not prefer to use the instruction manuals for most of the tasks although they couldn't complete them. They mainly preferred to use the instruction manual during Task 2, where none of them could complete the task successfully. Similarly, most of them did not prefer using the manual through the digital camera tasks.

### 3.1 User Characteristics

It was observed that different user groups reflected different characteristics. This affected the way of their interaction with the instruction manuals.

**Younger Participants.** It was observed that the most successful group was the younger participants in terms of their approach to using the instruction manuals. Rather than reading the manual beforehand, most of them preferred to use it when it was deemed necessary. They were skilled in finding the relevant parts to the tasks easily from the manual. As is discussed by Langdon et al [23], it was observed that prior experience generally has a positive effect, which can be seen from the results of the digital camera tasks. However it was observed during the BPM tasks that if the participants had used a similar device before, they were likely to not refer to the instruction manuals and adopt a 'trial and error' approach. As a result, some of the participants misused the product due to the variations in terms of the usage regarding the specific brand. For example, five of the younger participants used a BPM before and three of them failed due to their previous experience with different brands. They were also asked in the questionnaire why they did not prefer using the instruction manual and some of the answers of the participants who misused the device are given below:

- *"Because I am familiar to these kind of devices. I have used them many times in my life...that is the reason I didn't use the instruction manual." (Y3)*
- *"Let's just say I am familiar to them."(Y4)*

**Older Participants.** The main difference of the older participants was found to be their motivation. Most of them were totally unmotivated to use digital devices, particularly the digital camera. Only four of them had used a digital camera prior to their participation in the study. They also demonstrated their lack of motivation frequently by blaming themselves:

- *"It is not about that this (showing the manual) not good. I think it is just me!" It doesn't matter what the product is, I have to read the instructions then I cannot find the bit I am looking for.(O2)*
- *"I am afraid it is beyond me."(O3)*
- *"It is a shame really because it is all there, isn't it?" (Showing the manual)(O10)*

They are less familiar with the concepts, visual language and the interface metaphors of digital devices [10]. For example, none of the older participants could guess the way to see the pictures in the memory card by means of the symbols on the buttons. The instruction manual also led them to confusion, because the figure which indicates the button that functions the 'zoom' is displayed as, "(playback zoom) button". As a result four of the older participants were confused by this and pressed the zoom button to go into playback mode. It was also observed that they experienced difficulty in understanding the explanations which includes symbols and text together, for example, only one older participant could carry out the action shown in Figure 5. Four older participants could not understand which button is the 'round' button because there are five round shaped buttons on the product, however none of them have a round symbol as shown in Figure 5.

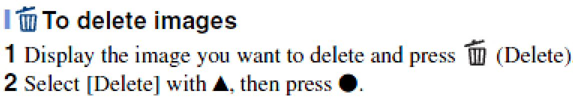

To delete images
1 Display the image you want to delete and press (Delete).
2 Select [Delete] with ▲, then press ●.

**Fig. 5.** How to delete images [20]

Due to the small font size, older participants experienced difficulty in reading the instruction manuals. Generally, 14-point type is recommended to be used in instruction manuals for older people or visually impaired people [1][28]. The questionnaires and the task list was prepared with 14-point type and all of the participants were able to read them easily, however 7 out of 10 older participants experienced difficulty in reading the instruction manual of the BPM and 4 of them also experienced the same problem with the digital camera's instruction manual. Below are some of their comments:

- *"I did find this (The instruction manual of the BPM) a bit difficult to read with my eye sight. You know a bigger print will help. Because, well you see I got glasses but still I can't see small print very well." (O4)*
- *"I found it is difficult to follow the instruction manuals and I also think the print is too small. It should be much larger, perhaps sort of this size (Showing the questionnaire where 14-point type used)." (O3)*

**People with Cognitive Disabilities.** Disabled participants were unmotivated to use the instruction manuals. As the result 5 participants did not use instruction manuals on any task for both of the devices. Only one of those participants was able to complete the digital camera tasks successfully (due to previous experience with digital cameras) however all of the others experienced various difficulties during the tasks which resulted in their failure to complete the tasks.

It was observed that participants with cognitive disabilities can have problems due to their reading skills, comprehension, impatience and attention. They frequently experienced difficulties in carrying out the actions written in the manuals. Some of their comments are given below:

- *It was quite difficult to follow the instructions. (D1)*
- *I can't always follow instructions very well, that's all. When it is that type of books I find it difficult. (D10)*

Similarly with the older participants, it was observed that most of the disabled participants were not familiar with digital devices. For example seven of the disabled participants experienced difficulty in understanding the symbols on the digital camera and in the instruction manual.

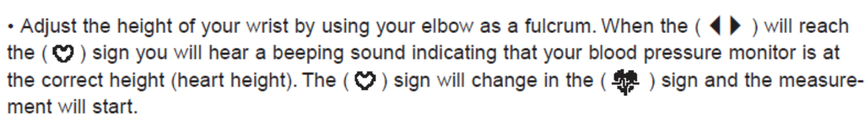
It was observed that all the disabled participants who used instruction manuals experienced difficulty in understanding the explanations. Differently than other user groups they also experienced difficulties in understanding the figures. For example four of the participants wrapped the device to their wrist in wrong position. Even though they checked the figure, shown in Figure 6, they were still not able to perform this routine correctly.

**Fig. 6.** the correct position of the BPM [21] – left; and the faulty positions performed by the participants (middle and right)

Finding the correct section within the instruction manuals was another difficulty for them, 2 out of 6 participants for the BPM and 3 out of 5 participants for the digital camera experienced this problem (where others did not prefer to use the manual for both of the devices).

### 3.2 Common Problems for All User Groups

**Explanation leading to confusion.** Figure 7 shows the explanation about how to use the BPM which is written in the instruction manual. Normally the participants are supposed to switch on the device and bring it to the heart height by lifting their hand using their elbow joint. During this process the arrow sign on the screen moves towards the heart symbol and the device gives a beeping sound. After they heard the beeping sound, they were supposed to maintain their position and to not move until the measurement taking is complete by the device (Figure 7).



• Adjust the height of your wrist by using your elbow as a fulcrum. When the ( ◀▶ ) will reach the ( ♡ ) sign you will hear a beeping sound indicating that your blood pressure monitor is at the correct height (heart height). The ( ♡ ) sign will change in the ( ) sign and the measurement will start.

**Fig. 7.** How to measure blood pressure [21]

As a result:

- 3 out of 8 younger participants (2 younger participants did not use the manual for this task)
- 5 out of 10 older participants
- 4 out of 6 disabled participants (4 disabled participants did not use the manual for this task)

related the arrows to the ‘arrow buttons’ on the product and hence, they pressed the arrows and waited for some time for the device to work without holding the device at their heart level. One of the younger participants could not understand the explanation and description given within the manual for this task even after reading it several times, however coincidentally managed to make the device work and completed the task 2. Three of the older participants could not complete this task because they could not figure out what they supposed to do. Three of the disabled participants coincidentally made the device work and the remaining disabled participant could not manage to get a reading from the device. A total of 12 participants experienced this problem which accounted for 50% of the participants referring to the instruction manual during this task.

**Wording problem.** As Figure 7 shows, the description is to “adjust the height of your wrist by using your elbow as a fulcrum”. With this, most of the participants (the majority of the participants’ first language is English) expressed that they did not know the meaning of the word ‘fulcrum’.

6 out of 8 younger participants stated that they had never heard of it before whilst one of them did not mention anything whether or not he knew the meaning of this word and the remaining 1 younger participant said:

– *“The meaning of fulcrum as I remember from physics is a turning point, something that you use for leverage of an object?” (Y1)*

6 out of 10 older participants also mentioned that they had never heard this word before and the remaining 4 did not mention anything. All of the disabled participants expressed that they did not know what the meaning of ‘fulcrum’ was. All of the older and the disabled participants were British.

– *“No, I don’t know. What is that mean?” (O5)*
– *“I am not sure what it means.” (D5)*

In total, 22 of the 30 participants expressed that they did not know the meaning of the word ‘fulcrum’. 7 participants did not mention anything at all. One participant was able to tell the meaning of this word from her knowledge of physics.

## 4 Discussion

The younger participants performed better than the other user groups. The majority of the younger participants used their previous experience as their primary information source, and they referred to the instruction manuals when it was deemed necessary to do so. As a result they were more successful and confident in using the digital camera when compared with the BPM.

As suggested by the literature, older people encountered more problems in using the instructions due to their impaired capabilities [17] [18]. According to Wright [34], older people are likely to experience difficulties regarding three domains of cognitive change, i.e. memory, attention and comprehension. The effects of decreased memory was obvious where they were asked to recall an action done before, e.g. when they were asked to switch off the device by using power button again. Their attention was very fragile particularly when they required switching between the product and the manual to execute the action explained in the manual. In addition they were unmotivated to use the digital devices, particularly when faced with the digital camera tasks. They experienced difficulty in understanding the functions and the symbols used on both of the devices and within the instruction manuals. Even though they used the instruction manuals more than all the other user groups, they still experienced various difficulties in understanding the explanations within the manuals. Due to their lack of motivation, they had a tendency to give up and as a result blame themselves on failing to complete the tasks. They experienced difficulty in reading the instruction manuals due to the small font size and frequently complained about it.

Regarding the disabled participants, their reading skills, comprehension, impatience and attention issues did prove to hinder them in their understanding of what was described within the manuals, which, in turn unmotivated them to use the instruction manuals. According to Valett [31], repeated failure can have negative motivational effects on people with learning disabilities. This was also observed from the comments of the participants during their use of the instruction manuals and this can be the reason of why some of the participants did not use instruction manuals.

Most of the disabled participants who used instruction manuals experienced reading difficulties [12] and reading comprehension deficiency was prominent [12][13]. This was demonstrated through their understanding of the text explanations and occasionally they had difficulty in understanding the figures. Attention deficits were also frequently observed as suggested by McKinney [25], especially when they were trying to find the correct sections within the manuals, which turned into a challenge for the disabled participants. However the participants with previous experience of using digital cameras performed better than others without experience.

Some of the explanations caused confusion in all user groups. Even though they read the explanations several times, some of the users from all user groups could not manage certain actions explained within the manuals. This was more prevalent with the older participants. Wording was another problem, sometimes participants experienced difficulty in understanding the explanations due to the terminology used, as with the example of the word ‘fulcrum’ described above.

## 5 Conclusion

Instruction manuals are critical for the product usage. Preparing good instruction manuals is important poorly designed ones can mislead the users. In addition it is up to users to decide whether or not using the instruction manuals, hence the design of the instruction manual should be appealing for the users.

All the different user groups presented different characteristics regarding their approach to using instruction manuals. Younger participants were found to be the most successful user group in using the instruction manuals. Therefore it is critical to understand the diversity of the users before preparing an instruction manual.

The design of the instruction manuals for the two products was found to be more appropriate for the younger user group. Some of the older participants and disabled participants experienced various difficulties regarding their understanding of the visual and text based explanations which resulted in their failure. Some of the older users were excluded by the design of the instruction manuals due to the small font size.

The effects of the prior experience were generally positive for all of the users, however during the BPM tasks some of the younger participants misused the device because they did not refer to the instructions due to their previous experience with similar products.

Usage of the symbols in the text based explanation was found to be confusing. If the users were not familiar with the products or the symbols, they were likely to be misguided, especially the older and disabled user groups.

It was observed that the explanations and the wordings should be tested with different kinds of users during the preparation of the instruction manuals. If the instruction manuals are designed inclusively, then the products can be usable by a broader range of user, thus resulting in better performances.

## References

1. Backinger, C.L., Kingsley, P.A.: Write It Right: Recommendations for Developing User Instruction Manuals for Medical Devices Used in Home Health Care. FDA (1993)
2. Baskinger, M., Hanington, B.: Sustaining Autonomous Living for Older People Through Inclusive Strategies for Home Appliance Design. In: Langdon, P., Clarkson, J., Robinson, P. (eds.) Designing Inclusive Futures, pp. 117–126. Springer, London (2008)
3. Bauer, M., Gaskell, G.: Qualitative Researching with Text, Image and Sound. Sage Publications, London (2007)
4. Braddock, D., Rizzolo, M.C., Thompson, M., Bell, R.: Emerging Technologies and Cognitive Disability. Journal of Special Education Technology 19, 49–56 (2004)
5. BS 7000-6:2005: Design Management Systems-Part 6: Managing Inclusive Design-Guide, British Standards Institute (2005)
6. Clarkson, J., Coleman, R., Hosking, I., Waller, S.: Inclusive Design Toolkit, Kall Kwik Cambridge (2007)
7. Coleman, R., Clarkson, J., Dong, H., Cassim, J.: Design for Inclusivity. Gower Publishing Ltd., England (2007)
8. Czaja, R., Blair, J.: Designing Surveys: A Guide to Decisions and Procedures. Sage Publications, Pine Forge Press, London (2005)
9. Dong, H.: Barriers to Inclusive Design in the UK. PhD Thesis, Cambridge, Department of Engineering (2003)
10. Eisma, R., Dickinson, A., Goodman, J., Syme, A., Tiwari, L., Newell, A.F.: Early User Involvement in the Development of Information Technology-Related Products for Older People. Universal Access in the Information Society, vol. 3, pp. 131–140 (2004)
11. French, S.: Dimensions of Disability and Impairment. In: French, S. (ed.) On Equal Terms: Working with Disabled People, pp. 17–34. Butterworth-Heinemann Ltd., Oxford (1994)
12. Gardill, M.C., Jitendra, A.K.: Advanced Story Map Instruction: Effects on the Reading Comprehension of Students with Learning Disabilities. The Journal of Special Education 33, 2–17 (1999)
13. Gersten, R., Fuchs, L.S., Williams, P.J., Baker, S.: Teaching Reading Comprehension Strategies to Students with Learning Disabilities: A Review of Research. Review of Educational Research 71, 279–320 (2001)
14. Gupta, S.: Design and Delivery of Medical Devices for Home-Use: Drivers and Challenges. Cambridge, Department of Engineering (2007)
15. Hawthorn, D.: Interface Design and Engagement with Older People. Behaviour & Information Technology 26, 333–341 (2007)

16. Heskett, J.: Design: A Very Short Introduction. Oxford University Press, Oxford (2002)
17. Horen, F.M., Jansen, C., Meas, A., Noordman, L.G.M.: Manuals for the Elderly: Which Information Cannot Be Missed. Journal of Technical Writing and Communication 31, 415–431 (2001)
18. Horen, F., Jansen, C., Noordman, L., Maes, A.: Manuals for the Elderly: Text Characteristics That Help or Hinder Older Users. In: 2005 IEEE International Professional Communication Conference Proceedings, International, July 10-13, pp. 334–342 (2005)
19. Huppert, F.: Designing for Older Users. In: Clarkson, J., Coleman, R., Keates, S., Lebbon, C. (eds.) Inclusive Design: Design for the Whole Population, pp. 88–103. Springer, London (2003)
20. Instruction Manual DSC-S730 Sony Cyber-shot Digital Camera: Sony Corporation (2007)
21. Instruction Manual R7 Omron Automatic Blood Pressure Monitor. Omron Healthcase. Co. Ltd
22. Keates, S., Clarkson, J.: Design Exclusion. In: Clarkson, J., Coleman, R., Keates, S., Lebbon, C. (eds.) Inclusive Design: Design for the Whole Population, pp. 88–103. Springer, London (2003)
23. Langdon, P., Lewis, T., Clarkson, J.: The Effects of Prior Experience on the Use of Consumer Products. Universal Access in the Information Society, vol. 6, pp. 179–191 (2007)
24. Margolin, V.: Getting know the user. Design Studies 18, 227–236 (1997)
25. McKinney, J.D.: The Search for Subtypes of Specific Learning Disability. Journal of Learning Disabilities 17, 43–50 (1984)
26. Myerson, J.: A Growing Movement. In: Coleman, R., Clarkson, J., Dong, H., Cassim, J. (eds.) Design for Inclusivity, pp. 23–31. Gower Publishing Ltd., England (2007)
27. Robson, C.: Real World Research: A Resource for Social Scientists and Practitioner-Researchers. Blackwell Publishing, Malden (2002)
28. Smith, T.P.: Manufacturer's Guide to Developing Consumer Product Instructions. US Consumer Product Safety Commission (2003)
29. The Department of Trade and Industry: Instructions for Consumer Products: Guidelines for Better Instructions and Safety Information for Consumer Products. Her Majesty's Stationery Office, London (1989)
30. Thompson, C.W.: Rizzolo: Smart Devices and Soft Controllers. IEEE Internet Computing 9, 82–85 (2005)
31. Valett, R.E.: Programming Learning Disabilities. Fearon Publishers (1969)
32. Vanderheiden, G.: Fundamental Principles and Priority Settings for Universal Usability. In: Proceedings of the ACM Universal Usability Conference, Washington, DC, pp. 32–38 (2000)
33. Wiklund, M., Wilcox, S.: Designing Usability into Medical Products. CRC Press, Boca Raton (2005)
34. Wright, P.: Supportive Documentation for Older People. In: Westendorp, P., Carel, J., Punselie, R. (eds.) Interface Design & Document Design, pp. 81–100. Rodopi, Amsterdam (2000)

# FASTS: FAcets Structured Tag Space – A Novel Approach to Organize and Reuse Social Bookmarking Tags

Sudha Ram and Wei Wei

Department of MIS, Eller College of Management
The University of Arizona, Tucson, AZ 85721, USA
ram@eller.arizona.edu, wwei@u.arizona.edu

**Abstract.** Social bookmarking tools are generating an enormous pool of metadata describing and categorizing web resources. The value of these metadata in the form of tags can be fully realized only when they are shared and reused for web search and retrieval. The research described in this paper proposes a facet classification mechanism, and a tag relationship ontology to organize tags into a meaningful and intuitively useful structure. We have implemented a web-based prototype system to effectively search and browse bookmarked web resources using this approach. We collected real tag data from del.icio.us for a wide range of popular domains. We analyzed, processed, and organized these tags to demonstrate the effectiveness and utility of our approach for tag organization and reuse[1].

**Keywords:** tag, social bookmarking, facet, semantics, ontology, del.icio.us.

## 1 Introduction

Social bookmarking services such as del.icio.us, Connotea, Flickr, and CiteULike are rapidly rising in popularity. Fueled by Web 2.0 technologies, these systems are used by individuals to bookmark interesting web resources via a web-accessible "Favorites" list. More importantly, each bookmarked resource can be annotated with a set of tags. A **tag** is a keyword or term assigned to a resource mainly for categorization purposes. Tags help describe a resource and allow it to be located again by browsing, searching, and navigating. Through tagging, web information consumers collaboratively create a **tag space**, a web resource taxonomy also known as *folksonomy*. From a social and collaboration viewpoint, tags are useful if they are shared and reused to help locate items of interest. This requires software systems that can effectively facilitate the reuse process based on the meaning of the tags, i.e., the semantics of the tags and the structure of the tag space. Currently, reuse of tags is at a very primitive level, i.e., using keyword based search. This kind of search has both low precision and recall due to problems such as homonyms, synonyms, different lexical forms or alternate spellings, and misspellings of tags. There are some techniques to create tag clouds that provide a visual depiction of tags based on their popularity. These can be used to find collections

---

[1] This research is supported in part by research grants from the National Science Foundation (#EF-0735191) and the Science Foundation of Arizona.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 426–438, 2010.

of web resources. However tag clouds have been found to be suboptimal when used for searching. It is still not feasible to locate relevant and specific web resources by formulating meaningful ad hoc queries on the tag space [12]. Contributing to this deficiency are the facts that folksonomy lacks inherent structure and hierarchy, and there are no explicitly represented tag semantics and relationships among tags. The objective of our study is to develop and validate a mechanism to explicitly represent the structure and the relationships inherent in the tag space in an intuitive and user-friendly manner. This will help harness, represent, and organize the hidden 'wisdom of the crowds' in a meaningful way to benefit web information searches.

The rest of the paper is organized as follows. In section 2, we describe the challenges of reusing tags for web resource retrieval. In section 3, we examine and analyze relevant work and justify the need for a new approach. In section 4 we describe our research methodology and propose a faceted approach and its core elements. In section 5, we introduce a prototype system we have implemented and describe its use and benefits. We conclude with a discussion of future research directions in section 6.

## 2 Challenges in Reusing Tags

Social bookmarking tools are readily available and they are easy to use. These low entry barriers directly contribute to the challenges in reusing tags. Tags generated using these tools are untidy and imprecise, i.e., they have inherent ambiguity caused by polysemy, homonymy, synonymy, and basic level variation [9], and they are often overly personalized, and not machine interpretable. In addition, the inability to annotate relationships between tags greatly restricts the expressiveness of tags and places them out of their contexts. The fact that human taggers need a way to represent relationships and hierarchies is evident in some novel forms of concatenated tags created by many users. For example, "java/programming languages" as a tag, though syntactically counted as one word, actually implies a web resource referring to Java as a type of programming language. Therefore, we as humans know that here "Java" does not refer to coffee, it can be classified under "Programming Languages". Unfortunately, none of this is understandable to a machine and as a result has minimal value in tag reuse. Adding hierarchy and structure consisting of relationships to the flat tag space seems to be a plausible solution. However, this creates a conflict between the "freedom" advocated by the social web and the "control" that can be achieved by using taxonomies and ontologies. Gruber in [10] argues that we cannot really consider the Semantic Web and folksonomy as two opposite ends of a spectrum, rather, there is increasing value in applying Semantic Web technologies to the data from the social web. Ontologies are necessary to solve problems associated with reusing user generated metadata such as tags. Based on this argument, a major question that arises is, "What approach may be used to augment the tag space with semantics and how feasible and valid is it to operationalize this approach." We attempt to answer this question in our research.

## 3 Relevant Work

Realizing the importance of augmenting a tag space with semantics, researchers have been working on two major approaches. The first approach is to extract semantics including the meaning of tags and relationships between tags from the tag space *post hoc* i.e. *after* the tags are created. This extraction can be performed using statistical analysis of tag co-occurrence to identify clusters of related tags; or using a knowledge-intensive approach where semantics of tags are obtained by aligning them to a knowledge source, or combining both. There are a lot of statistical techniques for tag co-occurrence analysis [3, 17, 20, 22]. Studies such as [1] utilize knowledge sources including online ontologies and WordNet. All of these techniques have some drawbacks: (i) The techniques focus on finding groups of related tags, but do not identify how these tags are related to each other, i.e. the exact relationship semantics are not extracted or labeled; (ii) The accuracy of these techniques depends on the intended meaning of the tag which may be ambiguous in some cases; (iii) The diversity of the tags makes it impossible to find knowledge sources that are comprehensive enough to enable semantics extraction with acceptable accuracy and coverage.

The second approach is to capture semantics *during* the process of tagging. One method is to enable taggers to create tags as a triple i.e. <subject, predict, object> [25]. Taggers can describe and categorize web resources using triples with less ambiguity and more semantics. Methods such as those described in [23] allow taggers to annotate bookmarking intent that goes beyond the bookmarked content, thus the generated "purpose tags" are expected to provide additional insights for content search and browsing. Other methods allow taggers to add semantics using resources such as WordNet [15], or the Wikipedia [14]. Some social bookmarking systems[2,3,4] allow taggers to clarify semantics during tagging. The success of these approaches is dependent on the tagger's willingness to follow the suggested practices. This is an elevated entry barrier and in many cases a serious problem to overcome.

To augment a tag space with machine-understandable semantics, a certain level of control and structure are required, which can be achieved by ontologies. One such ontology is the Tag Ontology [18] which models the relationship between a tagger, a web resource, and one or more tags. The SCOT (Social Semantic Cloud of Tags) [13] ontology models the social network among taggers using social web ontologies such as FOAF [7]. Normally, tagging is modeled using three elements: Tagger, Resource, and Tag. MOAT (Meaning Of A Tag) [19] proposes the addition of "Meaning" as the fourth element. The meaning is generated by a tagger to clarify the semantics of the tag by referring to other web resources. These ontological approaches focus on the key elements of a tagging system and provide little help in augmenting the tag space with meaningful hierarchy and structure which are necessary for effective reuse of the tag data. These ontologies are incapable of capturing and representing the various relationship semantics inherent in the tag space other than very high level super-sub class relationships. In reality, many relationships other than IS-A [6], including part-whole [8], and verb-based generic relationships [4] implicitly exist in a tag space. All these types of relationships are yet to be exploited and incorporated for tag analysis and organization. Our approach incorporates these semantic relationships into tag organization.

[2] www.zigtag.com
[3] www.faviki.com
[4] www.fuzzzy.com

# 4 A Faceted Approach for Organizing Tags

## 4.1 Research Methodology

In this research we take a design science approach as suggested by Hevner et al in [11]. Our artifacts in this research include a three-dimensional faceted model to categorize social bookmarking tags; a relationship ontology to define and categorize various relations existing implicitly in the tag space; heuristics for organizing tags into facets; and a prototype system implemented as an instantiation of our proposed faceted framework. In terms of problem relevance, our research provides an answer to an important question arising from the "social web", i.e., how can value be created from the enormous amount of "social web user generated metadata". In addition, the model proposed in this work is empirically evaluated using real social bookmarking tag data for effectiveness and completeness. Rigor of this research lies in the formal modeling for the tag space and the faceted structure construction methodology.

## 4.2 Our Proposed Approach

Our method to organize and structure tags is based on two major components: (i) A model to categorize tags into three high level facets, i.e. Resource Type, Resource Value, and Resource Content; and (ii) A relationship ontology to organize tags within these facets, which provides rich semantics beyond a simple hierarchy.

In our research we collected a large set of social bookmarking data from del.icio.us. as summarized in Table.1.

**Table 1.** del.icio.us Tag Dataset

| Domains | Art | Blogs | Business | Design | Food | Science | Software | Travel |
|---|---|---|---|---|---|---|---|---|
| NOU | 91903 | 82471 | 86412 | 138103 | 72929 | 64336 | 105847 | 80190 |
| NOUU | 56122 | 58385 | 49831 | 65964 | 47066 | 35298 | 55112 | 51841 |
| NOT | 511867 | 377905 | 517211 | 784039 | 302706 | 341227 | 641098 | 316293 |
| NOUT | 27434 | 28258 | 29880 | 26970 | 18533 | 21609 | 27181 | 20261 |

This data includes bookmarked URLs, the tagger who created the bookmark, date and time when the bookmark was created, the tags created by the user as well as any notes. We collected this data for 8 different long-term popular domains—art, blogs, business, design, food, science, software, travel. For each domain, we collected a large number of tags over a five week period to ensure a sizable tag dataset. Table 1 shows the details of our dataset with Number of URLs (NOU), Number of Unique URLS (NOUU), Number of Tags (NOT), and Number of Unique Tags (NOUT). In each domain, NOUU is a subset of NOU with duplicate URLs removed.

## 4.3 Facets for Organizing Tags

In folksonomy, a web resource carries various labels in the form of tags because taggers often view the resource from different perspectives for diverse purposes. This social practice makes conventional taxonomic classification inadequate because a web resource can fall under many interesting categories. Faceted classification, on the other hand, allows an object to have more than one classification. Organizing web resources using a faceted classification scheme makes it possible for users with disparate intents to search, browse, and navigate through resources.

A facet is a clearly defined property or characteristic of a class or specific subject. To come up with tag facets, we analyzed our collected del.icio.us dataset. The analysis result confirms previous findings on purposes of tagging i.e. to "organize" and "communicate" [16]. In addition, we discovered that in order to fulfill these two purposes, people often create tags to: (i) describe domain of the resource; (ii) indicate the type of the resource; and (iii) highlight the value of the resource, i.e., describe the benefits of a resource. This finding gives us justification to propose a three-facet model as shown in Fig. 1 for web resource classification using tags.

The Resource Type Facet describes the presentation format of the resources. For example, a web resource may be textual such as a blog entry, or an e-magazine. It may be a graphic such as paintings, photos, illustrations, or multimedia such as an audio or a video clip. The Resource Value Facet describes the reason that made the resource worth tagging. For instance, a resource may provide tips to save money and/or effort by providing services and tools. Some resources help the user acquire problem-solving skills. Other tagged resources provide opinions or reviews in various forms. These two facets are domain-independent and the collected tag datasets suggests they remain stable across domains. For example, our collected del.icio.us tag datasets indicate that for every resource domain, the "top" (popular) tags often include Resource Type tags such as "blog" and "graphics"; and Resource Value tags such as "howto", "free", "gadget", and "tutorials".

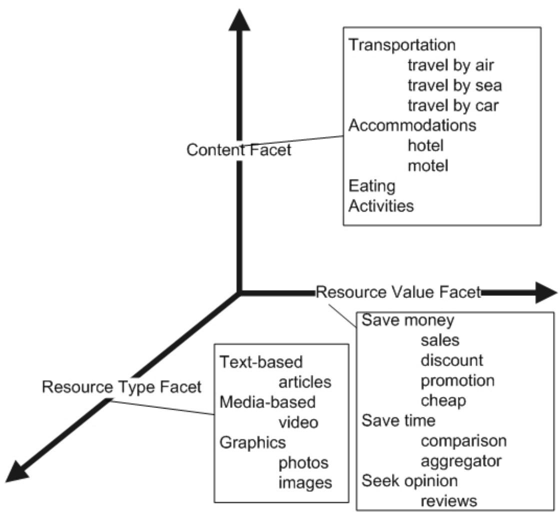



**Fig. 1.** Facets of Social Bookmarking Tags

The Content Facet is domain-dependent. For each resource domain, tags in the Content Facet and its subfacets describe various aspects of the application domain. For example, in the domain of "travel", tags are often used to annotate web resources in terms of transportation and accommodation, i.e., these are the two things people care about most when they travel. Furthermore, tags often describe the types of accommodation and transportation, such as air transportation which in turn has elements including "ticket", "airfare", and "airline"; car travel may include terms such as "rental". The details of the Content Facet can be extracted and derived from the tag space. Information about tags such as tag frequencies, and popularity ranking of tags can help suggest the components of the content facet. Tag co-occurrence analysis can then be used to derive the structure of all facets. We explain how this is actually operationalized in section 5.

### 4.4 Relationship Ontology

Another component of our approach is the relationship ontology. The tag space is a pool of distilled knowledge generated by taggers. A prerequisite to using this knowledge is to clearly represent the structure of this body of knowledge, i.e., the concepts as well as their relationships. Relationships between concepts/entities are implied by the fact that certain tags are used together for a specific annotation. An analysis of the tag dataset revealed that the set of relationship types implicitly embedded in the tag space is stable across resource domains. Based on classical work in knowledge representation [21] and findings from our empirical study, we propose a relationship ontology which has the ability to represent the semantics of relationships between tags as shown in Fig.2.

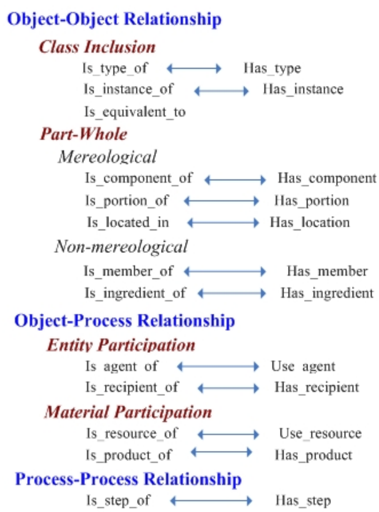



**Fig. 2.** Relationship Ontology

In this ontology, 'Object' and 'Process' are used to represent the topics about which people make annotations, i.e., when they see interesting resources about certain objects (or entities) and activities, they create tags to denote them. Some tags are grouped into sets because the tagger considers them related, therefore, the relationships can be between objects, between processes, or between object and process. Relationships that are inverse of each other are connected by two-sided arrows.

The relationship between objects can be of two general types: specialization/generalization relationships that describe the hierarchy among entities; and part-whole relationships that specify the composition of various entities of different nature. Within the Part-Whole category, mereological part-whole relationships are transitive, as are their inverses. The relationship between processes can be used to describe the decomposition of a process into finer granularity. The relationships between object and process help represent how certain activities take place, what people and things participate in the process, and the manner in which they are related. In traditional categorization schemes, hierarchy is the major relationship. In our approach, we enrich the relationship type pool with our relationship ontology so that more relationship semantics can be represented. The flat tag space can thus be organized into a faceted structure using this ontology. Tag frequencies suggest facets, i.e., topics of interests. Tag co-occurrence analysis can be used to generate related tag sets, the relationships from our proposed ontology can be used to augment and describe how tags in each set are related. Fig.3 is an example of such a faceted tag structure in the domain of "travel".

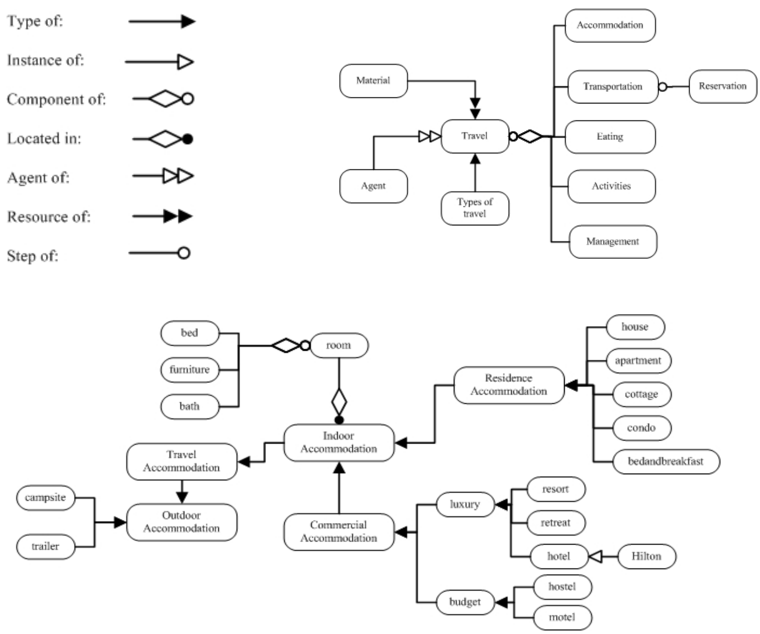



**Fig. 3.** Using Relationship Ontology to Organize a Tag Space into Faceted Structure

### 4.5 Faceted Model of the Tag Space

A faceted tag space is defined as $TS \subseteq U \times T \times R$, where $U$ is the set of social bookmarking service users, $T$ is the set of tags used for resource annotation, and $R$ the is set of tagged resources. Facets defined in our tag categorization model—Content Facet, Resource Type facet, and Resource Value facet are represented by $F_c$, $F_{rt}$, and $F_{rv}$ respectively. These three facets are orthogonal, i.e., $F_c \cap F_{rt} = \varnothing$, $F_c \cap F_{rv} = \varnothing$, and $F_{rv} \cap F_{rt} = \varnothing$. For each of the facets, there exist hierarchies which can be viewed as tree structures. Subfacets are the nodes which can be located through tree traversal. The location of a node is denoted as $N\langle i,j,k,\ldots\rangle$ where $i,j,k,\ldots$ are positive integers used as indexes. The hierarchy depth is the length of the index list and the numbers in the list indicate the node indexes at each hierarchy level. For example, $N\langle\rangle$ with no index means it is at the highest level of the hierarchy, while $N\langle 1,2\rangle$ is the second subfacet of its parent facet, which is the first subfacet of the top level facet. Similar to other hierarchical structures, subfacets have the property of disjointness and nesting [5]. $\forall i,j, \forall p: i \neq j \Rightarrow N\langle p,i,\ldots\rangle \cap N\langle p,j,\ldots\rangle = \varnothing$, all nodes whose notation share an initial prefix are disjoint if the first index after the common prefix is different. $\forall p,j: N\langle p,j,\ldots\rangle \subseteq N\langle p\rangle$, Given a node with prefix $p$, all nodes represented by a longer index list starting with the prefix $p$ are subsets of the first node. For each bookmark, relationships in the tag set can be modeled using our relationship ontology, $t_i r\, t_j \Rightarrow r \in RO$ where $t_i$ and $t_j$ are the tags, $r$ is the relationship between them, and $RO$ is the set of relationships defined in our relationship ontology.

## 5 Constructing the Facet Structure

We have implemented a prototype system called FASTS (**FA**cets **S**tructured **T**ag **S**pace) to demonstrate how the faceted structure and relationship ontology can be constructed and used to organize a tag space. We also use this prototype system to demonstrate the utility of the organized tag space structure for web search and retrieval.

### 5.1 Tag Data Facet Structure Creation Using Tag Data

Figure 4 describes the major steps we undertook to reform a flat tag space into a faceted structure. Starting with tags collected from del.icio.us, for each domain, we had to cleanse the data before further analysis and facet construction. Tags that contained two or fewer letters or any special symbols were removed. Tags that were used less than 10 times were deleted. In addition, tags that were obviously meaningless, non-English, and overly personal were also deleted. Furthermore, overlapping tags such as plural/singular forms or upper/lower case were sorted and combined—our rule of thumb was to use plural form and lower case representation whenever possible.

The construction of the faceted structure requires generating facets and their corresponding subfacets as well as organizing these facets and subfacets into a meaningful structure. The first step was conducted in a data-driven approach and the second step used our relationship ontology. Tags and their structures for the Content Facet vary by domain and therefore the construction for each domain was as follows. We first used the dataset to create a tag co-occurrence network. Using a network analysis technique, we then conducted a p-core decomposition [2] of the co-occurrence matrix which was very sparse. This helped identify major cohesive groups of tags; each group had tags that were much more closely linked to each other than to tags outside the group. The tags in each group served to generate major topics in the various domains. We then did hierarchical clustering using Ward's method [24] (Normalized corrected Euclidean as a measure of dissimilarity) to identify possible subgroups within each group of tags. The relationships between subgroups and the relationships between subgroups and their parent groups were then labeled using our relationship ontology. This procedure was repeated for each of the subgroups and a comprehensive structure of tagged topics in the domain was constructed with relationship semantics explicitly defined.

Finally orphan tags that did not belong to any of the subgroups identified by p-core decomposition were allocated to the content facet by domain experts facilitated with some knowledge bases such as well-established domain ontologies, their relationships were also defined using the relationship ontology where applicable. This process helped define the structure of the content facet. This analysis and construction process also confirmed that our relationship ontology was capable of capturing the relationship semantics between most tags in our dataset.

As stated earlier, we confirmed by empirical analysis of the dataset, that tags describing the Resource Type and Resource Value facets and their structures did not vary across domains. Therefore, the structure of Resource Type and Resource Value facet were generated empirically and reused for every domain.

Once the facet structures were defined, we automatically assigned each bookmarked resource and its tags to one or more specific position(s) in the faceted structure. For example, a travel website with tags that indicated airfare comparison and hotel reviews was allocated to multiple positions in the faceted structure: Resource Value→ Save Money, Seek Opinion; Resource Type→Website; Content Facet→Air travel→Airfare; and Content Facet→Accommodation→Hotels.

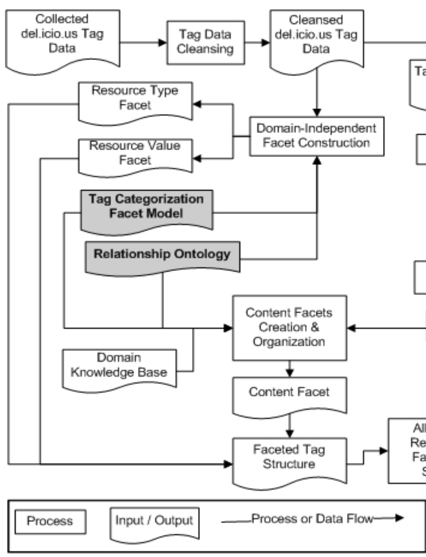



**Fig. 4.** Faceted Tag Structure Construction

### 5.2 FASTS Prototype System Implementation

Our collected tag data and derived tag facet structure were loaded into an open source software system called Flamenco [26]. Developed in Berkeley, Flamenco (FLexible information Access using MEtadata in Novel Combinations) provides a framework and infrastructure for storing and structuring metadata. It also provides a mechanism to develop a query front-end to search for data using the harvested metadata. It allows users to both refine and expand their search queries, while maintaining a consistent representation of the collection's structure. The use of metadata is also integrated with free-text search, allowing the users to follow links, add search terms and follow more links, without interrupting the interaction flow. Figures 5 and 6 show the faceted structure and the user interface of our prototype FASTS system.

Within FASTS, users can browse through the faceted structure of the tag space distilled from social bookmarking sites. Using the facet structure navigation, hierarchy expansion and drilling mechanisms built into FASTS, users can construct complicated queries based on the underlying relationships defined in our ontology. For example as shown in Figure 5, users can view the faceted structure of the "food" domain. There are query previews behind each facet that reveal the number of qualified resources for that facet and avoid navigation to "dead ends". Subfacets, if applicable, can be viewed by pointing the cursor on a facet.

Clicking on a facet helps narrow down the users' search and browse choices. For example, a user who is interested in locating recipes using a specific cooking method will be able to find out that the collection of "food" related web resources contains more than 2000 recipes bookmarked with "cooking method" tags and that it covers 9 different cooking methods. This set of over 2000 resources are mapped to the facet structure from where the user can drill down to find out that there are 21 recipes under the cooking method "stir fry", as shown in Figure 6; while 36 of the recipes are presented in multimedia—such as a video clip. Thus the user is able to narrow down her

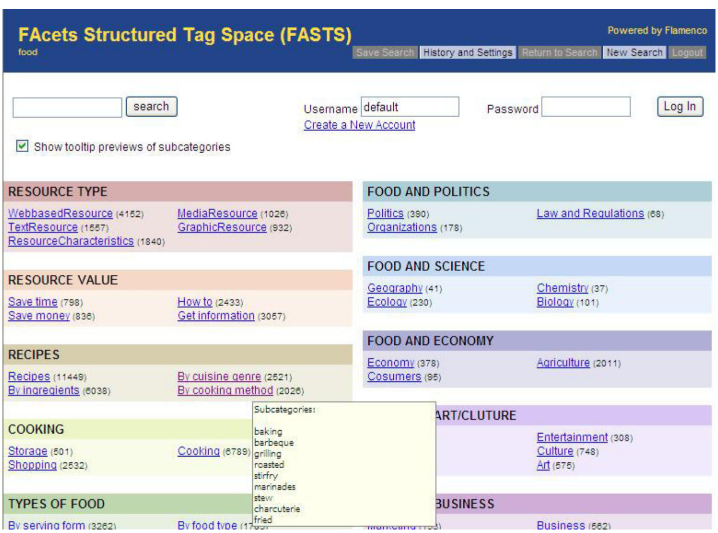



**Fig. 5.** FASTS Front Page for the Food Domain

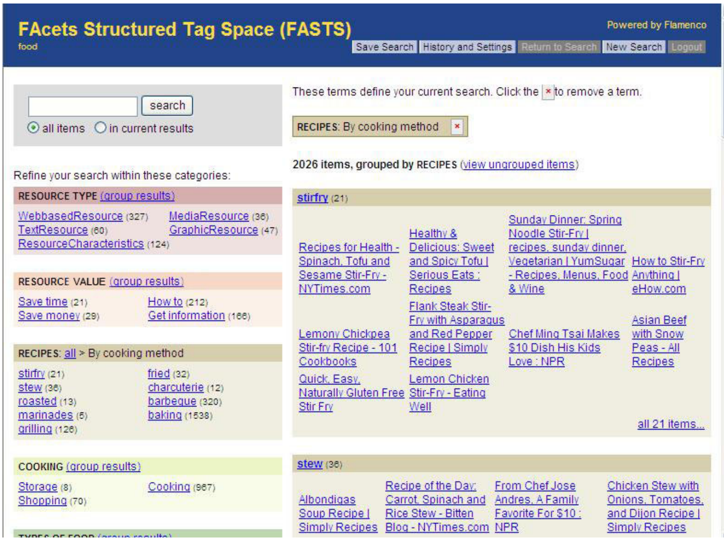



**Fig. 6.** FASTS Intermediate Page for the Food Domain

selection using one or more facets to enhance her search capability. In the process, some tag disambiguation is also achieved. For example, a resource in the domain of "food" (tagged by 'food) with tag 'java' is interpreted as describing a type of coffee instead of a programming language.

## 6 Conclusion and Future Research

In this research, we propose a novel approach to reorganize the flat social bookmarking tag space into a faceted structure. Using an empirical study we verified that our proposed faceted structure can be derived from the tags and our proposed relationship ontology is capable of covering a wide variety of domains. A prototype system was implemented based on our proposed approach to search and browse web resources in various popular domains. We believe that organizing web resources into a faceted structure extracted from tags is an effective solution to current challenges we face in tag reuse. Keyword based search can be replaced by flexible query formulation and revision to save users from tedious manual search. Our techniques also help in resolving the ambiguities in tag semantics and uses tag relationship semantics to generate the facet structure. This has the potential to increase information retrieval precision.

We are in the process of developing techniques to increase degree of automation of the facet construction process. We believe domain ontologies will prove useful for this. We are also interested in exploring the use of the facets in identifying and/or refining social networks or groups generated from the social bookmarking community.

## Reference


1. Angeletou, S., Sabou, M., Motta, E.: Semantically Enriching Folksonomies with FLOR. In: 1st International Workshop on Collective Semantics: Collective Intelligence & the Semantic Web (CISWeb 2008), European Semantic Web Conference, pp. 65–79 (2008)
2. Batagelj, V., Zaversnik, M.: Generalized Cores (2002), http://arxiv.org/abs/cs.DS/0202039
3. Begelman, G., Keller, P., Smadja, F.: Automated Tag Clustering: Improving search and exploration in the tag space. In: The Collaborative Web Tagging Workshop at WWW 2006, Edinburgh, Scotland (2006)
4. Bergholtz, M., Johannesson, P.: Classifying the Semantics of Relationships in Conceptual Modeling by Categorization of Roles. In: 6th International Workshop on Applications of Natural Language to Information Systems, pp. 199–203. GI (2001)
5. Bonino, D., Corno, F., Farinetti, L.: FaSet: A Set Theory Model for Faceted Search. In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT 2009, pp. 474–481 (2009)
6. Brachman, R.J.: What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks. Computer 16(10), 30–36 (1983)
7. Brickley, D., Miller, L.: FOAF Vocabulary Specification 0.9 (2007), http://xmlns.com/foaf/0.1/
8. Chaffin, R., Herrmann, D.J., Winston, M.: An Empirical Taxonomy of Part-Whole Relations: Effects of Part-Whole Relation Type on Relation Identification. Language and Cognitive Processes 3(1), 17–48 (1988)
9. Golder, S.A., Huberman, B.A.: Usage Patterns of Collaborative Tagging Systems. Journal of Information Science 32(2), 198–208 (2006)
10. Gruber, T.: Ontology of Folksonomy: a Mash-up of Apples and Oranges. International Journal on Semantic Web and Information Systems 3(1), 1–11 (2007)
11. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information System Research. MIS Quarterly 28(1), 75–105 (2004)
12. Heymann, P., Koutrika, G., Garcia-Molina, H.: Can Social Bookmarking Improve Web Search? In: International Conference on Web Search and Web Data Mining, Palo Alto, California, USA, pp. 196–205. ACM Press, New York (2008)
13. Kim, H.L., Yang, S., Song, S.-J., Breslin, J., Kim, H.: Tag Mediated Society with SCOT Ontology. In: The 5th Semantic Web Challenge Workshop at the 6th International Semantic Web Conference (ISWC 2007) (2007)
14. Krotzsch, M., Vrandecic, D., Volkel, M., Haller, H., Studer, R.: Semantic Wikipedia. Web Semantics: Science, Services and Agents on the World Wide Web 5(4), 251–261 (2007)
15. Laniado, D., Eynard, D., Colombetti, M.: Using WordNet to Turn a Folksonomy into a Hierarchy of Concepts. In: Semantic Web Application and Perspectives-The 4th Italian Semantic Web Workshop, Bari, Italy, pp. 192–201 (2007)
16. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT06, tagging paper, taxonomy, Flickr, academic article, to read. In: Proceedings of the Seventeenth Conference on Hypertext and Hypermedia, Odense, Denmark, pp. 31–40. ACM, New York (2006)
17. Mika, P.: Ontologies Are Us: A Unified Model of Social Networks and Semantics. Journal of Web Semantics 5(1), 5–15 (2007)
18. Newman, R.: A Tag Ontology, http://www.holygoat.co.uk/owl/redwood/0.1/tags/tags.n3

19. Passant, A., Laublet, P.: Meaning of A Tag: A Collaborative Approach to Bridge the Gap Between Tagging and Linked Data. In: WWW 2008 Workshop Linked Data on the Web (LDOW 2008), Beijing, China (2008)
20. Schmitz, C., Hotho, A., Jaschke, R., Stumme, G.: Mining Association Rules in Folksonomies. In: Batagelj, V., Bock, H.-H., Ferligoj, A., Ziberna, A. (eds.) Studies in Classification, Data Analysis, and Knowledge Organization, pp. 261–270. Springer, Heidelberg (2006)
21. Sowa, J.F.: Knowledge Representation : Logical, Philosophical, and Computational Foundations. Brooks/Cole, Pacific Grove (2000)
22. Specia, L., Motta, E.: Integrating Folksonomies with the Semantic Web. In: 4th European Semantic Web Conference, Innsbruck, Austria, pp. 624–639 (2007)
23. Strohmaier, M.: Purpose Tagging: Capturing User Intent to Assist Goal-Oriented Social Search. In: The 2008 ACM Workshop on Search in Social Media, Napa Valley, California, USA, pp. 35–42. ACM, New York (2008)
24. Ward, J.H.: Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association 58(301), 236–244 (1963)
25. Yang, J., Matsuo, Y., Ishizuka, M.: An Augmented Tagging Scheme with Triple Tagging and Collective Filtering. In: IEEE/WIC/ACM International Conference on Web Intelligence, Silicon Valley, USA, pp. 35–38 (2007)
26. Yee, K.-P., Swearingen, K., Li, K., Hearst, M.: Faceted Metadata for Image Search and Browsing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Ft. Lauderdale, Florida, USA, pp. 401–408. ACM, New York (2003)

# Twitter Me: Using Micro-blogging to Motivate Teenagers to Exercise

Mark Mingyi Young

School of Information Systems & Technology
Claremont Graduate University
Claremont, CA 91711 U.S.A.
mark.young@cgu.edu

**Abstract.** A micro-blogging site for the persuasive technology is developed along with cell phone application that helps motivate teenage girls to exercise by exploiting their social desire to stay connected with their peers. We choose teenage girls because they are more responsive to health behavior interventions and they think exercising is the easiest way to improve health. The purpose of this study is to provide real-time information sharing in order to modify the behaviors of girls and ultimately lead to improved health habits. Our study investigated how collecting, sharing and comparing personal fitness information impacts activity level and health awareness.

**Keywords:** Persuasive technology, health improvement.

## 1 Introduction

Today persuasive technologies are ubiquitous; we are surrounded by digital products designed to change what we think and do. Persuasive technology experiences come to us through the web (from commerce sites to social networking) and mobile phones (e.g., health applications for iPhone and commercial texting services) [5]. Therefore, Fogg argues that we no longer have to invent new persuasive solutions out of whole cloth. Instead, we can focus on existing persuasive technology products and techniques, varying those systems to understand the dynamics and principles of persuasive design [4]. The target group of people in this research is teenage girls; this design aims to motivate them to exercise and improve their health awareness. The intent of this design is to provide information in the form of short text messages to modify their exercise behaviors. Our study will investigate how sharing and comparing personal exercise and fitness information impacts activity level and health awareness.

Many of the health challenges faced with in the western society can only be solved by infusing humans with the motivation to make long-lasting lifestyle changes. For instance, obesity, alcoholism, internet addiction, compliance and corrective behavior technologies, social support, and digital interventions provide a multitude of interesting research questions [15]. The prevalence of overweight adolescents in the United States has tripled in the past 20 years due to poor dietary habits and a lack of physical activity [3]. Designing persuasive systems that could resolve even some small parts of

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 439–448, 2010.

these problems and aid in true long-term sustainable change would provide to be very valuable. Although computers cannot communicate in the same way as humans, there are studies that suggest that computer-human persuasion may utilize some patterns of interaction similar to social communication [12], whereas computer-mediated persuasion means that people are persuading others through computers, e.g. discussion forums, e-mail, instant messages, blogs, or social network systems.

A micro-blogging site that emulates functions of Twitter will be designed as the artifact for this study. However, only partial functions of Twitter that are deemed related to this research will be implemented. The reason for using a site to emulate Twitter is that we could have total control over the interface design and the site usage. For example, we need to show group statistics and achievement level on the site.

The persuasion design principles of this study are based on the PSD (persuasive systems development) framework suggested by Oinas-Kukkonen and Harjumaa [14]. This design research tries to investigate using micro-blogging site like Twitter and cell phone technology, as the persuasive tool to alter teenager's motivation to exercise. Little study has been done regarding using micro-blogging as a persuasive technology to change teenagers' behaviors. This research will provide new contribution in helping to understand the utility of micro-blogging as a persuasive technology.

## 2 Micro-blogging as the Persuasive Technology

Micro-blogging site like Twitter is a new form of communication in which users can describe their current status in short posts distributed by instant messages, mobile phones, email or the Web. Micro-blogging like Twitter encourages users to share information about anything they are seeing or doing, the motivation facilitated by the ability to post brief text messages through a variety of devices. Twitter, the most popular micro-blogging tool, is exhibiting rapid growth [10].

Java et al. found that people use micro-blogging to talk about their daily activities and to seek or share information [8]. Micro-blogging has been widely adopted by users as an effective means to capture and disseminate their thoughts and actions to a larger audience on a daily basis. Interestingly, daily chatters of a user obtained from her micro–blogs offer a unique information source to analyze and interpret her context in real-time – i.e. interests, intentions, and activities [1]. Java et al. found that most posts on Twitter talk about daily routine or what people are currently doing [8].

Zhao and Rosson [16] found that by staying aware of others' ongoing updates with Twitter, people are able to keep in touch with friends and maintain social relationships; this is especially important for contacts that are not part of their daily life or work activities. In their study they found that micro-blogging was viewed as a quick and easy way to share interesting and fun things happening in daily life activities; it lets users keep in touch with friends and colleagues. Information posted by a person the reader has deliberately selected to follow is perceived as useful and trustworthy. Zhao and Rosson [16] also indicate that Twitter has potential impacts on informal communication. They suggest that Twitter is useful for increasing awareness of what is on each others' mind; this in turn implies that it may help to generate more common ground.

### 2.1 Object of Study

This study focuses on teenage girls. A study on inactive adolescent girls shows that the effectiveness of interventions aimed at increasing physical activity among adolescent girls might be enhanced by engaging support from friends, family, and caring adults; addressing real and perceived time constraints; and helping adolescent girls feel more confident about themselves and their ability to engage in physical activity [13]. Tosco et al. reported that teenage girls thought exercising is the easiest way to improve health when compared with dietary changes [15]. According to Gortmaker et al., girls were found to be more responsive to health behavior interventions in a two-year study targeting adolescent obesity [7].

### 2.2 Behavior Modification

Behavior modification is the cornerstone of any persuasion program. Fogg [6] asserts that for a person to perform a target behavior, he or she must (1) be sufficiently motivated, (2) have the ability to perform the behavior, and (3) be triggered to perform the behavior. These three factors must occur at the same moment; else the behavior will not happen. Lacroix et al. [9] studied the role of behavioral regulation, motives, and self-efficacy for physical activity behavior. They argue that when properly accommodated to the three cognitions underlying an individual's behavior, tailored interventions can be promising in realizing behavior change. Therefore, they conclude that in order to enhance the persuasive power of technology-based physical activity interventions, programs should be tailored to accommodate these underlying cognitions. In particular, they should aim to induce and gradually internalize the three cognitions that seem to be powerful in realizing an active lifestyle. Our design concept leverages persuasive technology to change the behaviors of teenage girls.

## 3 Research Questions

Our specific goal is to determine if peer pressure and technology can be combined in a way that increases physical activity in teenage girls. Cialdini [2] describes "social validation" or peer pressure as one that drives human behavior. He describes "liking" or the feeling of being connected as a factor in changing behavior.

Our research is concerned with how micro-blogging site like Twitter and cell phones combined change teenagers' exercise habits. Our specific goal is to determine if peer pressure and persuasive technology can be combined in a way that increases physical activity in teenagers [3, 15]. A further research question is whether the girls will develop an increased awareness and understanding of the relationship between exercise and health.

## 4 Theoretical Background of the Design

The persuasion design principles of this study are based on the PSD framework suggested by Oinas-Kukkonen and Harjumaa [14].

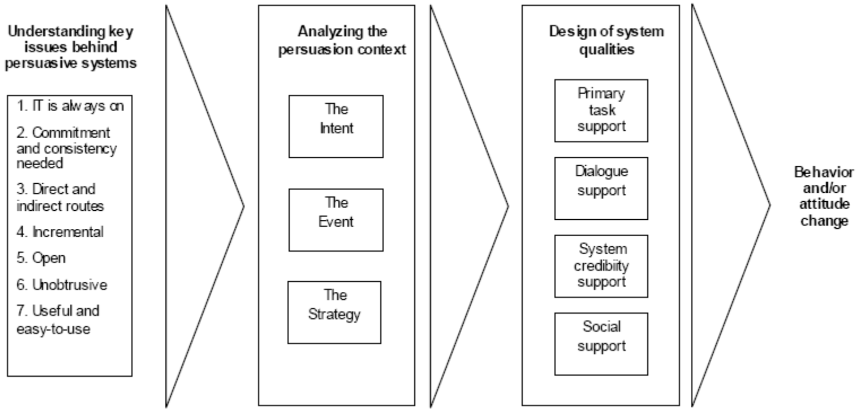



**Fig. 1.** PSD Design Framework

### 4.1 Understanding Key Issues behind Persuasive Systems

From the seven postulates the following postulates are most useful in this design.

The first postulate states that information technology is never neutral. Rather it is "always on", influencing people's attitudes and behavior in one way or another. Moreover, people are constantly being persuaded. The postulate supports that the combination of Twitter-like micro-blogging and cell phone should constantly keep users informed of others' status.

The second postulate states that people like their views about the world to be organized and consistent. If systems support the making of commitments, users will more likely be persuaded. For example, a user may express greater confidence in his or her decision to exercise regularly after having bought a gym membership card. The idea of commitment also implies that persuasive systems could provide means to make private or public commitments to performing the target behavior. This can be implemented, for example, by offering an easy way to send a text message or email to one's relatives, friends, or colleagues. The postulate supports that we employ an achievement system as the goals for users to achieve, and we will send praise messages for performance.

The fourth postulate states that persuasion is often incremental. In other words, it is easier to initiate people into doing a series of actions through incremental suggestions rather than a one-time consolidated suggestion. This implies that a persuasive system should enable making incremental steps toward target behavior. The postulate supports that we create different step achievement levels. The text messages are sent indicating the group performance, including individual achievement level achieved.

The fifth postulate states that persuasion through persuasive systems should always be open. It is very important to reveal the designer bias behind of the persuasive system. For instance, simulations may bear great persuasive power, but if the designer bias remains unclear for the users the simulations may either lose some of their persuasiveness or they may end up misleading their users. Moreover, content that is based on untruthful or false information does not fit with the overall goal of users' voluntarily changing attitudes or behaviors.

The sixth postulate states that persuasive systems should aim at unobtrusiveness, i.e. they should avoid disturbing users while they are performing their primary tasks with the aid of the system. In this manner, the system is capable of fulfilling users' positive expectations. The use of persuasive features at improper moments may result in undesirable outcomes. Since the text messages can be viewed either from the Twitter-like web site or from a cell phone, users have all the freedom to choose an opportune time to access these messages.

The seventh postulate, persuasive systems should aim at being both useful and easy to use, i.e. at really serving the needs of the user. This includes a multitude of components, such as responsiveness, ease of access, lack of errors, convenience, and high information quality, as well as positive user experience, attractiveness, and user loyalty. Quite understandably, if a system is useless or difficult to use, it is unlikely that it could be very persuasive. The postulate supports that both Twitter-like micro-blogging and cell phone are easy to use and quite accessible to teenagers.

### 4.2 Analyzing the Persuasion Context

The persuasion context of this design research will be carefully analyzed using the following framework:

The Intent: The design of this persuasion aims at behavior change.

The Event: Both the use context and the user context will be defined by way of pilot study. A questionnaire will be administered to the participants. Interviews of ethnographic study of teenagers will also be conducted. In this pre-test survey we will find out the current health status and exercise habits of the participants. We will also gather information for recommendations of the design by way of prototype.

The Strategy: After analyzing the message for the persuasion, a set of pre-determined, specially designed text messages will be used for the persuasion. Considering the proper route to be used in reaching the user, since the teenagers are possible to carefully evaluate the content of the persuasive message, a direct route will be used.

### 4.3 Design of System Qualities

The categories for persuasive system principles suggested in this article are primary task, dialogue, system credibility, and social support. The following principles in each category will be appreciated by this design research:

Primary Task Support: Users only have to input minimum messages (reduction); the system provides pedometers for teenagers (tunneling); the system provides information regarding progress and historical data (tailoring); the system provides customized micro-blogging interface (personalization); the system provides interface for users to check their status (self-monitoring); data of burnt calories are presented (simulation); pedometers are administered to participants (rehearsal).

Dialogue Support: the system provides praise as feedback (praise); the system shows stars as reward (rewards); the system reminds users of their target behavior during the use of the system (reminders); the system suggests that users carry out be haviors during the system use process (suggestion); slang names are used in the interface (similarity), the system has a look and feel similar to Twitter (liking).

System Credibility Support: the system provides unbiased praise and physical achievement information (trustworthiness); information is updated frequently and there are no out-of-date data (expertise); there are no ads on the web site; only information related to this study is shown on the interface (surface credibility); the accuracy of site information is verified through peers (verifiability).

Social Support: the system provides social comparison, normative influence, social facilitation, cooperation, competition and recognition for the social support.

# 5 Proposed Design

The participants engaged in a friendly competition. In order to incorporate an element of competition we created different step achievement levels [11] in the form of a star scale which is from one star (basic level) to five stars (highest level). The text messages were sent indicating the group performance, including individual achievement level (stars) achieved. The content of the messages was specially formulated for persuasive purposes. With these text messages, one could notify her followers of the current exercise one is engaging. One could also notify her followers of one's current health improvement, like weight loss. Text messages were also sent offering praise for reaching individual step goals.

We then designed the Twitter-like micro-blogging site for a group of four friends to engage in a friendly competition where the group's walking statistics are tracked. Each girl had a slang name which is familiar to their friends. Each girl had to enter her daily pedometer reading manually as her "update" on the web site. The pedometer kept track of the number of steps that were taken each day. Automated text messages were re-formatted and sent real time to the group cell phones indicating the group performance, individual level achieved, and calories burnt. Text messages were also sent offering praise for reaching individual step goals. Tosco et al. reported that teenage girls are extremely interested in staying connected with their friends [15].

## 5.1 Scenario

Mash, Joker, Tip, and Jeans are the slang names of four teenagers who want to improve their lifestyles. They decide to use the Twitter-like site as a fun way to get more exercise. They start out slowly, each walking between 1000-3000 steps per day. They manually key in their number of steps which is broadcast to others' cell phones by the site. They can get praise messages, group daily activity, and group achievement on the cell phone. They can check how many stars they get for their group achievement and historical statistics. After a few days, Tip decides that she would like to increase her exercise and aims for 5000 steps per day. Her accomplishment inspires Mash and Joker to do the same when they note that Tip is walking much more than they are. Jeans, on the other hand, takes a few days off. The other girls notice that Jenny is bringing down their group average and encourage her to try harder. They decide that it might be more fun to walk together rather than separately and send each other text messages to plan a 'walking date.' This gives Jeans the extra motivation she needs and the next day she walks 7000 steps.

## 5.2 Prototype

The teenagers were interviewed to gather information for recommendations of the design. Design recommendations based on these findings are used in the prototype design. The recommendations include a Twitter-like layout, easy access to all statistics, and simplicity. We moved forward to design and implement a working prototype for the micro-blogging site and the cell phone interface. The prototype, written in Visual C# and ASP.NET, was developed with consideration for future cell phone implementation in terms of storage limitations and interface design (e.g. size of the display). The web site prototype includes user registration and set-up, short message entry, group progress report, and historical statistics (Figure 2).

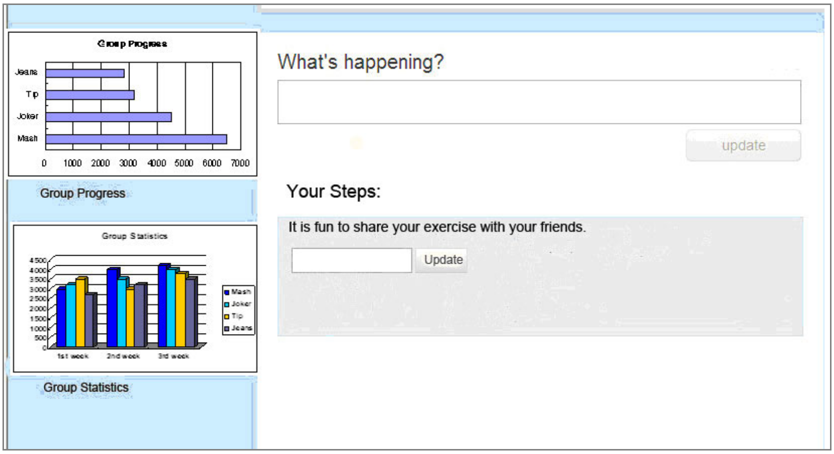



**Fig. 2.** Web Site Interface

The messages entered at the web site are integrated with other information and reformatted for mobile device. The data is then sent out to the cell phone and presented on the cell phone interface (Figure 3). On the cell phone interface users receive information like group progress, her current level of achievement, the praise, and group step counts. The cell phone also provides an application for users to calculate the calories (Figure 4).

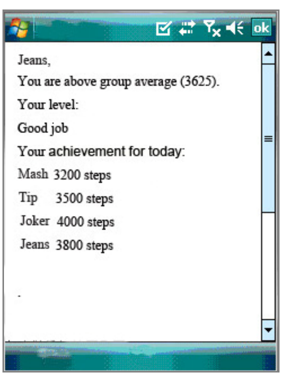



**Fig. 3.** Cell Phone Interface

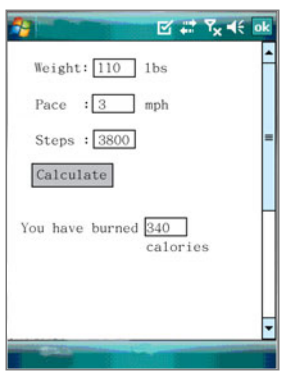



**Fig. 4.** Cell Phone Calorie Calculator

## 6 Method

We opted for a fashionable pedometer to measure the steps the teenage girls take. A group of four friends were recruited to participate in the study. Our group consists of four teenage students including two 15-year-old girls and two 18-year-old girls. Human Subjects Committee approval was obtained and parental consent secured for each participant. Each of the girls was given a pedometer, a cell phone and the web site prototype to use for four weeks. They were required each day to communicate their step counts and other updates using the micro-blogging web site.

In a pilot study, a questionnaire was administered to the participants. In this pre-test survey we found out the current health status and exercise habits of the participants. We also gathered information for recommendations of the design. Design recommendations based on these findings were used in the initial design.

## 7 Evaluation

We conducted a user study, using Twitter-like micro-blogging and pedometers to evaluate the effectiveness and acceptability by teenage girls. A post-test user study used a questionnaire to evaluate the effectiveness and acceptability by teenagers. Pre-study and post-study questionnaires as well as post-study interviews yielded the quantitative and qualitative results.

## 8 Results

The pre-study questionnaire showed that all participants were physically healthy, and two of the participants had daily physical activity. We also found that all participants had a cell phone and were Twitter users. None of them ever owned a pedometer before. The girls were very excited to use the web site and the cell phone combined technology. Figure 5 shows that the girls made progress in number of group steps taken over the four weeks.

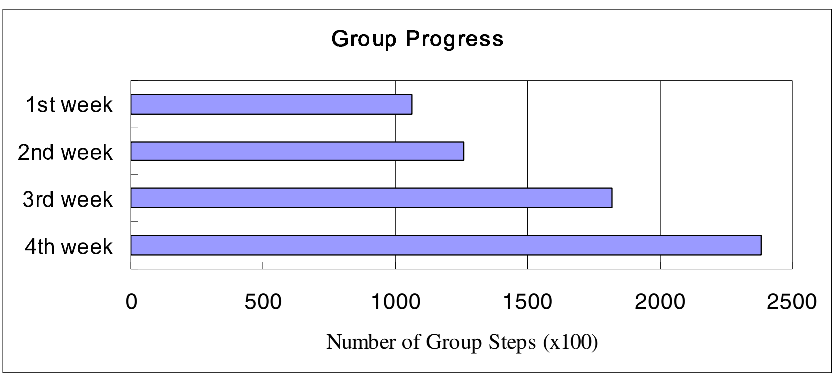



**Fig. 5.** Group Progress

A main finding from the post-study questionnaire was that impact on fitness was rated by the girls as being the most powerful method of changing behavior. Viewing group performance was also rated high in the method of changing behavior. Surprisingly, the importance of calorie information was not as highly rated by the girls. This persuasive technology was also effective in terms of raising the health awareness. However, it was not as effective in changing the behavior in other exercise (Table 1).

**Table 1.** Reponse to Post-study Questionnaire: Likert Scale 1–5 (not–very)

| | |
|---|---|
| Impact on fitness | 4.6 |
| Impact of viewing group performance | 4.4 |
| Importance of calorie information | 3.8 |
| Impact on other exercise | 3.2 |
| Impact on health awareness | 4.1 |

## 9 Conclusion

A Twitter-like micro-blogging site, along with a cell phone interface, is used for the persuasive technology to offer teenage girls a way to collaboratively motivate each other to continue being physically active and enjoy the friendly community. In sum, this persuasive technology changes the exercise behavior by providing a cooperative, supportive process where friends can share personal fitness information and give one another encouraging feedback.

## 10 Research Recommendations

Future research could focus on the durability of the persuasive effects on these teenage girls after the study has finished. Further information is needed to answer questions like: Will they still keep their current physical activity and health awareness after the study? Will their awareness of health diminish or change over time?

## Acknowledgments

I thank all who participated in our study and their parents for their support.

## References

1. Banerjee, N., Chakraborty, D., Dasgupta, K., Joshi, A., Mittal, S., Nagar, S., Rai, A., Madan, S.: User Interests in Social Media Sites: An Exploration with Micro-blogs. In: Proceeding of the 18th ACM conference on Information and knowledge management (2009)
2. Cialdini, R.B.: The Science of Persuasion. Scientific American 284, 2 (2001)

3. Ferraro, K.F., Thorpe Jr., R.J., Wilkinson, J.A.: The life course of severe obesity. Journal of Gerontology 58B, 2 (2003)
4. Fogg, B.J.: Persuasive Technology: Using Computers to Change What We Think and Do. Morgan Kaufmann Publishers, San Francisco (2003)
5. Fogg, B.J.: Creating Persuasive Technologies: An Eight-Step Design Process. In: Proceedings of the 4th International Conference on Persuasive Technology (2009a)
6. Fogg, B.J.: A Behavior Model for Persuasive Design. In: Proceedings of the 4th International Conference on Persuasive Technology (2009b)
7. Gortmaker, S.L., Peterson, K., Wiecha, J., Sobol, A.M., Dixit, S., Fox, A.K., Laird, N.: Reducing Obesity via a School-Based Interdisciplinary Intervention Among Youth. Arch of Pediatric Adolescent Medicine 153, 409–418 (1999)
8. Java, A., Finin, T., Song, X., Tseng, B.: Why We Twitter: Understanding Microblogging Usage and Communities. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis (2007)
9. Lacroix, J., Saini, P., Goris, A.: Understanding User Cognitions to Guide the Tailoring of Persuasive Technology-Based Physical Activity Interventions. In: Proceedings of the 4th International Conference on Persuasive Technology (2009)
10. Lenhart, A., Twitter, F.S.: status updating. Pew Internet & American Life Project (2009)
11. Miller, G.R.: On Being Persuaded: Some Basic Distinctions. In: Dillard, J.P., Pfau, M. (eds.) The Persuasion Handbook: Developments in Theory and Practice. Sage Publications, Thousand Oaks (2002)
12. Nass, C.I., Steuer, J., Tauber, E.R.: Computers Are Social Actors. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating Interdependence. ACM Press, New York (1994)
13. Neumark-Sztainer, D., Story, M., Hannan, P.J., Tharp, T., Rex, J.: Factors Associated With Changes in Physical Activity: A Cohort Study of Inactive Adolescent Girls. Arch. Pediatr Adolesc Med. 157, 803–810 (2003)
14. Oinas-Kukkonen, H., Harjumaa, M.: Persuasive Systems Design: Key Issues, Process Model, and System Features. Communications of the Association for Information Systems 24 (2009)
15. Toscos, T., Faber, A., An, S., Gandhi, M.P.: Chick Clique: Persuasive Technology to Motivate Teenage Girls to Exercise. In: Proceedings of ACM CHI 2006 Conference (2006)
16. Zhao, D., Rosson, M.B.: How and Why People Twitter: The Role that Micro-blogging Plays in Informal Communication at Work. In: Proceedings of the ACM 2009 international conference on Supporting group work (2009)

# Designing for Light-Weight Collaboration: The Case of Interactive Citizens' Advisory Services

Gerhard Schwabe[1], Claudia Bretscher[1], and Birgit Schenk[2]

[1] University of Zurich, Department of Informatics,
Binzmuehlestrasse 14, 8050 Zurich, Switzerland
[2] University of Applied Sciences Kehl, Kinzigallee 11, 77694 Kehl, Germany

**Abstract.** This paper reports on the design of a collaborative system to support citizens' advisory services. Recent research on the adoption of collaborative technologies indicates that: a) successful collaborative technologies diffuse from the private sector to the business sector and not vice versa, b) collaborative processes evolve and therefore cannot be prestructured in detail, and c) creative collaboration can be characterized as creating and sharing mental models. We demonstrate how these insights informed our design of a citizens' advisory system and provide data from an evaluation in a German city. Implications for the design of our collaborative system are offered.

**Keywords:** CSCW, Collaboration Information Technologies, CIT, E-Government, Advisory Support, Citizenship Information.

## 1 Introduction

The history of collaborative technologies is a history of great promises, large failures and surprising successes. Most users have appropriated Lotus Notes only as an advanced mailing and scheduling system, not taking into consideration that it offers a comprehensive collaboration support. Although large productivity effects have reported on the use of Group Support Systems [1], most companies abolish them after an extended initial implementation [2]. On the other hand, "light-weight" collaborative technologies, such as Wikis, Chats and Twitter, have very quickly been adopted in the private sector and from there moved into the corporate IT. It appears that a strategy of bottom up diffusion of "light weight" collaborative tools linked to the users' private lives is a more appropriate strategy than is the classical top-down-diffusion of "heavy-weight" collaboration support [3]. How can we learn from the success of those light-weight collaborative tools? How can their success inform the design of collaboration support that is applicable in an organizational setting? How can designers induce users to appropriate new collaboration practices? These were questions we asked ourselves when we started off designing a tool to establish a new interaction between citizens and their advisors in public administration.

In the subsequent literature section we introduce relevant literature on the adoption of collaborative technologies. Section 3 describes the design, i.e., the design context of citizen advisory, generic requirements, the design methodology and process, and

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 449–460, 2010.

the resulting prototype. Section 4 presents the results of an evaluation. Section 5 ends the paper with our conclusions and implications.

## 2 Adoption of Collaborative Technologies: Field Experiences and Design Implications

Ever since collaborative technologies were invented in the middle of the 1980s, there has been dispute as to how they should be designed in order to be adopted in the workplace. A significant stream of research has focused on the organizational context and incentive systems. Orlikowski [4] shows how a Lotus Notes implementation failed because the use of the system was not in the interest of the users. Grudin first showed that groupware adoption (in his case: shared calendars) could fail if critical masses were not achieved [5], although he later admitted that peer pressure (!) could lead to a comprehensive adoption [6]. There have been numerous studies of successful Group Support System pilots [7], but many sites have since been abandoned. Researchers argue that facilitation skills are a major bottleneck, as skilled facilitators are needed but then are quickly promoted to other jobs when they are successful. Recent research therefore focuses on replacing the facilitator with tested collaboration patterns that the end user can apply without training [8]. While we acknowledge that organizational issues can influence design, this paper concentrates on the collaborational aspects that influence design for our chosen setting. Here we identify three relevant streams of research. In the following, we characterize each stream and introduce the underlying theories and concepts.

*1. Successful collaborative technologies diffuse from the private sector to the business sector and not vice versa.* In an extensive literature analysis on case studies of Collaboration Information Technologies (CIT) diffusion, Shumarova [9] concludes that successful CIT diffuse bottom up rather than being imposed top down by a company management. Tapscott and Williams ([10] p. 253) quote John Seely Brown from Xerox Parc on the example of Wikis: "A lot of corporations are using wikis without top management even knowing it. It's a bottom-up phenomenon. The CIO may not get it, but the people actually doing the work see the need for them." Tim Bray from Sun Microsystems is quoted for the following statement "…the technologies that come along and change the world are the simple, unplanned ones that emerge from the grassroots rather than ones that come out of the corner offices of corporate strategists" ([10] p. 253). Shumarova and Swatman [3] observe that successful CIT applications typically are Shadow CIT, i.e., tools that "are not implemented as part of the organisational IT infrastructure, neither have they received any targeted organisational investment"([3] p.371). These tools are first adopted into the private social life and then are gradually transferred into the business sector (e.g., via professional activities outside the company). Why is this the case? One argument stresses the primacy of the social function of collaboration. McGrath [11] already stressed in 1991 that team work not only has a productivity function, but also it should also support its members and the group well-being. Yet, it may make more sense to view social worlds as units of analysis, because team boundaries fluctuate and reconfigure [12]. If the social function is very important or dominates CIT adoption, it is much easier and more attractive to explore and adopt new technologies in the unregulated private network than in regulated and more rigid business hierarchies. Another argument looks at the nature of the tools: business oriented CIT tend to pre-structure collaborative work, while socially oriented CIT provide only simple structures and allow other structures to emerge. This argument will be discussed in the subsequent section.

*2. Collaborative processes evolve and therefore cannot be prestructured in detail.* CIT originating from the business sector tend to mimic business organizations, more specifically by: a) implementing the organizational hierarchy and role models into the software (most visible in Lotus Notes) and b) implementing the plan-act-control cycle of management. Any collaboration activity is first planned e.g., by setting up the "appropriate" structures (e.g., setting up a team room in Lotus Notes or an agenda in Group Systems), and then these structures are used by the collaborators and result in a trace of data that can be controlled (e.g., a project documents, electronic meeting minutes). The underlying assumption of this engineering approach to collaboration is that collaborative activities can be pre-planned and thus prestructured. In a famous discussion with Terry Winograd, Suchmann [13] argues that content develops during communication, and thus communicators cannot make their intentions explicit before they voice an opinion. The German poet, Heinrich von Kleist [14], talked beautifully in 1805 about the "gradual composition of thought while speaking."[1] Thus, E-Mail-systems that pre-structure communication based on speech acts, in this case the coordinator [15] are doomed to fail. A study on the adoption of CITs indicates that this finding can be generalized to other kinds of CITs. In 2005, Bajwa et al [16]) showed that only E-Mail had reached high utilization levels - the other (formal) CITs had not been adopted in the workplace. On the other hand, online communities, Skype, Blogs, Social networks (e.g., Facebook), and to some extent Wikis, have been adopted diffused in the private sector first and then been introduced into companies. The nature of collaborative work often requires 'technological improvisation' [17] dealing with exceptions, unexpected breakdowns and emerging opportunities. This is particularly the case in creative, design oriented work. "Design […] is more emergent, more continuous, more filled with surprise, more difficult to control, more tied to the content of action, and more affected by what people pay attention to" ([18], p. 61).

*3. Creative collaboration can be characterized as creating and sharing mental models.* Social collaboration is mostly based on communication - and it is no surprise that, given the widespread adoption of mobile phones, E-Mail, instant messaging, and online forums show, communication support has most easily been adopted for private and work life. However, creative collaboration is rarely based on communication only. As Shrage [19] and Schwabe [20] elaborate, it relies on a shared artifact. These artifacts externalize mental models and allow them to be jointly viewed and manipulated. The artifacts need to be flexible in order to represent both the problem space as well as the emerging solutions. Such an effective sharing of information can then enhance group productivity [21].

## 3 Design: Process and Result

### 3.1 The Design Context: Citizens' Advisory

While commercial service industries (e.g., banks [22], [23]) have made significant advances in improving their advisory services, the public sector is lagging behind:

[1] Translation by the author.

there is a significant gap between support offered that consists of simple telephone services and self service information on the Internet, and the information needs of a citizen in complex life circumstances. Pure information is not sufficient if citizens are not able to express their information needs [24]. For example, a pregnant woman may very well describe her situation (i.e. being pregnant), but may not know what specific information to look for (i.e. what support and services could be applicable for her situation). Structuring websites according to life's circumstances may give a good general starting point, but this does not offer sufficient personalized support. Good advice can be found in the public administration offices (local, regional, state and federal), but it tends to be fragmented and distributed among different agencies [25]. Citizens lack orientation about what to ask, whom to address and how to use the information provided. In a prior publication, we have provided evidence of the need for an advisory service in German public administration concerning the circumstances of a pregnant woman [26]. In mystery shoppings in 18 German cities, good advice for pregnant women was found to be rare, and a systematic and integrated search for solutions was lacking in all communities. A good advisory process has (at least) two phases: The information need is determined first, after which the necessary information is aggregated and activities are initiated [27]. In the description of the prototype, we delve into greater detail of the advisory process.

### 3.2 Generic Requirements

In this section, generic requirements for Citizens' Advisory Systems (CAS) from the themes identified in section 2 are derived. Theme one urges to implement CIT bottom up and moving it from a private sector to the business sector. This is not completely possible with an application that has no use in the private sector, but building blocks from the private can be used to a large extent. This leads to the following generic requirements for the CAS:

R1. *Include Web 2.0 and community based information.* This means established basic information systems such as google maps that can be used build collaborative applications on top of it ("mash ups"). But this also includes community based information, e.g., discussion forums for pregnant women or rating sites for birth related services. These offer an additional perspective that a public administration is not willing or not able to provide itself. For some topics, citizens trust other citizens more than a public authority. Communities can be used in order to search for information during the advisory process or to follow up on issues left open during the session.

Theme one furthermore stressed the primacy of the social function over the productivity function of collaboration. Therefore we propose the following two requirements:

R2. *Allow the user to establish a personal relationship during, and develop it after, the interaction.* In a face-to-face interaction, the advisor can establish a personal relationship by a pleasing presentation of herself and a professional communication. In a distributed setup (not discussed further in this paper), providing pictures and some basic information of the advisor will increase the trust of the citizen [28]. A good citizens' advisor develops a personal relationship not only between the citizen and herself, but also with other relevant persons, e.g., peers in a community or other public officers.

R3. *Base the interaction on verbal communication*. Public authorities tend to rely on forms for gathering information. As we argue in [26], forms are of little use as long as the problem is ill-defined and open ended. The advisor can use her background knowledge and empathy with the citizen to uncover hidden information needs and offer advice not explicitly requested [24]. This is particularly valuable in novel life circumstances. A verbal discussion on problems and possible solutions is also a more natural means of collaboration.

Theme 2 suggests that collaborative processes evolve. This leads to the following generic requirements for the CAS.

R4. *Keep work processes simple*. While there is evidence that elaborate advisory processes can be useful (they are recommended in other sectors such as in banks), lack of acceptance of these models in the workplace [29] supports the argument that an elaborate predefined process may be an obstacle rather than a scaffold, if the subject matter is sufficiently complex.

R5. *Support the evolution of process structure during collaboration*. A lack of prescribed structure does not rule out the ongoing structuring of the work by the participants. Rather, some structure is necessary to support mutual understanding of the current status achieved and the upcoming activities at hand. Thus, there should be features for the users to create and develop their structures during their ongoing collaboration.

R6. *The state of collaboration must be transparent at all times*. Advisory issues can be complex. Since there is no standardized process enforced to scaffold the collaboration, the users must be able to understand the state of their collaboration at any time, e.g., which results have been achieved, how they were achieved, which issues still remain open, and how these open issues can be addressed.

Theme 3 argues that creative problem solving activities should be based on externalized mental models. This leads to the following generic requirements for the CAS.

R7. *Support the externalization of mental models*. Externalization of mental models requires a modeling space and modeling language understood by both advisors and a wide range of citizens. Thus, they have to be simple and be based on common-known metaphors. These externalized models should support joint reasoning about the problem and potential solutions. They should serve as boundary objects [30] between the citizens' and the administration.

R8. *Support the flexible sharing of artifacts capturing mental models*. The artifacts must at least be visible for both users; preferably both advisor and citizens are able to work with the artifacts. For the face-to-face setting, the literature on single display collaboration [31] provides more detailed design guidelines.

### 3.3 Design and Implementation Methodology and Process

The generic requirements were instantiated in the system: Citizens Advisory 2.0. They were implemented in a diploma thesis from March to September 2009 [32]. Scenario based design [33] was selected for development, and as an application scenario, the life situation "birth" was chosen. In scenario based design, the scenarios are the focal artifact for developing a shared understanding of developers and users. Scenarios are informal, situated descriptions of usage in natural language. They allow a holistic perspective on the IT usage. In our case, during the requirements analysis, problem scenarios were used to document the initial situation and their problems. In the subsequent design phase, activity scenarios, information scenarios and interaction scenarios were used to describe the solution. On the basis of the developed understanding, a prototype was developed using the Microsoft .net framework, Silverlight and a HP Touchsmart PC.

Empirical data was captured as part of the requirements analysis via mystery shoppings and workshops with citizens and advisors (for details [27]). The prototype was evaluated with 15 pregnant women (or women who had just given birth) and 7 advisors from the city of Sindelfingen in Southern Germany. This number of evaluators is typical for first prototype evaluations in leading design oriented computer science conferences (CHI, CSCW), but the results cannot yet be generalized to different organizational settings. Each test session lasted 20-30 minutes. Data was captured using screen capture software, observations and questionnaires for advisors and citizens. The core of the questionnaire was based on the UTAUT Framework [34]. Specific questions regarding usability and the generic requirements were added. Details on the development process and the questionnaire can be found in [32].

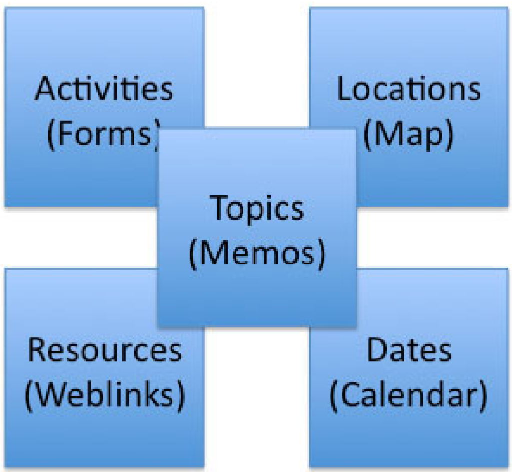



**Fig. 1.** Information Spaces

### 3.4 Prototype Implementation of the Generic Requirements

The system consists of five information spaces (figure 1): In the topic space, the advisor and the citizens establish the problems that need to be addressed, e.g., housing, finding childcare or applying for public support. Under each topic, the locations, activities, dates and resources of solutions can be explored. For example, a pregnant woman searching for childcare can find the location of kindergartens (locations on a map), apply for admission (activities with forms), note when she has to become active (dates on a calendar), and find additional information in the resources information space using web links.

Figure 2 shows the central topics space.

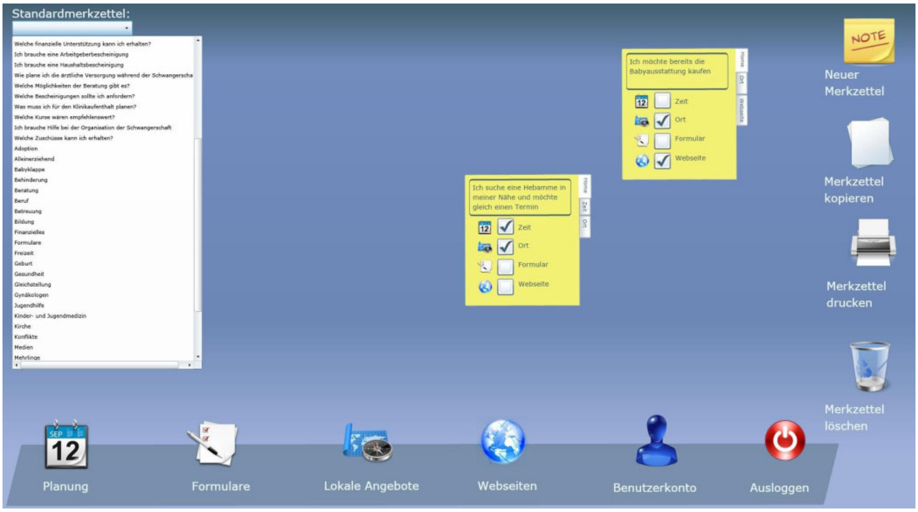



**Fig. 2.** Topics Space

Two memo cards have been created in the discussion between advisor and citizen for two topics (they can be created from scratch or selected from a list of predefined topics on the right side). After defining topics in writing, the pregnant woman and advisor search for solutions by exploring the other spaces. While doing so, they take the memo cards along to the next information space and link all results to it. For this purpose, four link fields are predefined on each card (one for each supporting information space).

Figure 3 shows the locations space as an example of the other spaces. The advisor and citizen can explore a map to find a locality where the solution is placed (e.g., a suitable Kindergarten). The current memo card can be seen on the left side of the screen. At any time, the users can switch to any other information space (= new screen).

Citizens and advisors first define a list of topics and then work through them in any order they wish. For each topic, they aggregate all information (including filling out application forms) needed. As the application is presented to them on a large desktop touch screen monitor, both can view and interact with the application. When they are finished the information is handed over to the citizen as a print out or in electronic form.

As we will see in the evaluation, this approach is sufficient to fulfill a pregnant woman's information needs and to enable her to become active. Key to success is the insight that public advisory is an activity of information aggregation. This leads to the five information spaces as basic architecture. Although it may be advisable that the advisor and the citizen first get an overview of the problems using the topic space and then find solutions for each of the topics, there is no prescribed order in which spaces have to be visited and when. In the test sessions we observed very different work processes, ranging from a very systematic problem solving process to improvized free-wheeling between the information spaces. Thus the work process is simple (R4)

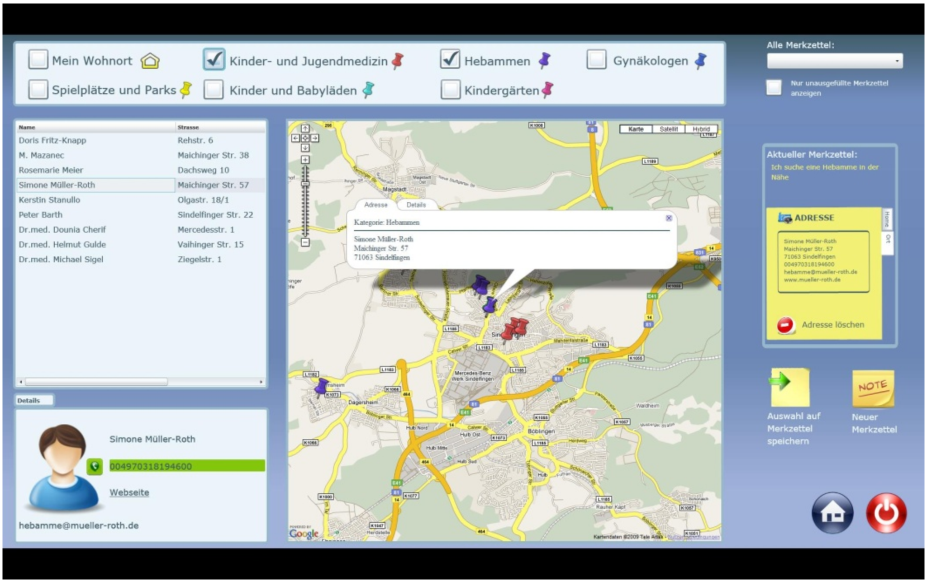



**Fig. 3.** Locations Space

and does not *prescribe* a work process (R5).Rather, the topic space allows the users to structure their work processes themselves as they progress with their work: We have observed users sorting memo cards to piles (prioritizing or sorting them) and then transferring them to other piles once a topic has been finished.

The other key to a successful application is the card metaphor, which is simple enough so that all observed users can externalize their mental models in a desktop layout of memo cards (R7). In combination with well-known web applications, problem solving activities are not hindered by difficulties in understanding the modeling concepts, but they are sufficiently rich to represent the problems at hand. The memo cards are visible in all information spaces and thus establish a context for each activity. Each makes the progress in the information spaces visible (by marking the link to it) and - together with the arrangement of the memo cards on the topic space –they make the work status transparent any time (R6).

The Citizens Advisor 2.0 is meant to be used in a face-to-face setting based on oral communication (R3). While we accept that some asynchronous online advisory may be possible, the face-to-face set-up combines the strengths of human advisory and web-based information in a way that cannot be automated in the foreseeable future. The shared view makes all activities transparent to both advisors and citizens, and the created artifacts can be touched on the screen (R8). The rich face-to-face set-up allows the citizens to establish a personal relationship for discussing critical personal issues (R2). This personal relationship can be extended to other persons as the citizens are made acquainted with relevant user communities in the resource space (R1).

While we are not aware of any advisory system built in a similar manner, our system does build on ideas, concepts and systems discussed in the literature. The basic interaction metaphors (especially interaction aspects not discussed in this paper) benefit from the literature on single display groupware [31].The structuring of collaboration into different information spaces has been discussed by [35]; however, its application to the information aggregation task of an advisor is novel.

Physical memo cards have been used in moderation for gathering and structuring topics for several decades. The cognoter (one of the first CSCW tool created, [36]) transferred the idea to the computer, and Apple popularized the concept of using hyper cards for structuring, linking and storing information. The room metaphor [37] proposed using card-like containers to move information from one information space to another. However, these publications discuss basic technology, and the envisioned user scenarios are different from the face-to-face advisory. Thus, our approach cannot be found in the basic technological building blocks, but it is novel in its purposeful assembly for providing a coherent, theoretically reflected solution on a medium level of abstraction (as typical for design oriented IS research).

## 4 Evaluation

This section presents a selection of the extensive evaluation results. Due to space limitations, we focus on those results most closely related to the requirements. All the evaluation results are available in German in [32]; selected further evaluation results can be found in [27]. The evaluation results indicate that the eight generic requirements have been successfully implemented in the prototype. The following sections present the results from 15 citizens and 7 advisors as a tuple (<citizens' average>/<advisors' average>). If a statement was only presented to one user group, only one number is given. The subjects were presented with a statement and had to indicate their agreement on a scale from 1= "I totally disagree" to "7 = I totally agree." The overall evaluation was very positive: The users clearly agreed with the statements: "I felt, the tested advisory session was productive" (6.2/5.9), "The tested advisory session was an interesting advisory experience" (6.2/6.0) and "I have generally liked the advisory session" (6.3/6.3).The citizens report that they "would use the advisory session as a service" (6.9), and the advisors "regard the system as useful for their work" (6.9). After the session, one pregnant woman asked, "When will the service be available?" and another woman stated, "We have had our children too early - we should have waited!" Thus, the Citizens Advisor 2.0 achieved user acceptance as far as it is possible with a prototype system. What contributed to this success?

The users widely agreed that the resources information space "with web pages on the topic birth ... is convenient to find websites" (6.3/6.1) and "is sufficient to achieve what is desired" (6.2/6.0). Thus, the *community based information* (R1) was implemented in a satisfactory manner. The citizens widely agreed that "the advisory appeared trustworthy" (6.4). Thus, the CAS allowed *the user to establish a personal relationship* (R2), indicating that the *verbal communication* during the advisory session was successful (R3). Although advisors had only half an hour training or no training at all, before the test sessions, the majority of the citizens agreed with the statement, "I could do my tasks with the system even if nobody were available to explain to me what to do" (5.5).This is a surprising result for a system which is meant to be facilitated by an advisor! The advisors agreed even slightly more with the statement above (5.7), and unanimously agreed that "it would be easy for me to become competent in the usage of system" (6.9). This indicates that the *work processes are fairly simple* (R4).

The work processes were primarily structured by the memo cards. Generally, the users overwhelmingly agreed that the "the idea of memo cards is good" (6.5/6.6). The users agreed that "the memo cards enable easy switching between screens" (6.1/5.7), indicating (together with the general high evaluation of memo cards) that there is acceptance of the chosen approach to *support the evolution of process structure during collaboration* (R5). The widespread agreement with the following statement indicates that it provides suitable support for the *externalization of mental models* (R7): "In the tested advisory session, my thoughts and concerns are well made manifest by the memo cards" (6.3/6.4). There is also ample positive user feedback on the *transparency of the state of collaboration* (R6): "The approach to store selections on the memo card ... is useful" (6.4/6.3), "... easy to use" (6.4/5.7)"...is clearly represented" (6.4/6.3) and ".... allows to interrupt work any time and to continue work any time without loss"(6.4/6.4). Further, the users agreed with the overall statement that the "system usage is clear and comprehensible" (6.1/6.3). Several evaluations also indicate that the system sufficiently supports *the flexible sharing of artifacts capturing mental models* (R8). "The joint usage of the screen enabled a productive advisory session" (6.3/6.6). Thus, the citizens "had many possibilities to contribute actively" (6.0/6.4).

## 5 Conclusions

The overall positive user feedback indicates that the prototype development was successful. We regard our most important success that the CAS was approved not only by the advisory clients, but also by the advisors themselves. Advisors tend to be very cautious in accepting technology for the advisory session itself because the application may not be in their interest (e.g., they may be afraid of being controlled) and because they fear losing face in the eyes of the client if they fail to cope with the software [29]. Feedback from representatives of the public administration indicates that the approach demonstrated may be applicable to a wide area of advisory tasks in the public administration.

As the software has not been rolled out to cities, it is premature to draw conclusions on its potential diffusion in the market place. The literature on the diffusion of collaborative technologies suggests that adoptable CIT should be simple, less pre-structured, support the evolution of structures, and be more social. Some of this CIT can be directly diffused from the private sector to the business sector. However, there are many specialized application areas for collaboration that require more specific tools. One such area is advisory software. In these areas, many of the attributes of successful "private-sector" CIT need to be integrated (or "meshed up") with the domain specific features of the application. This can mean that successful applications are integrated into the system (as exemplified by the integration of Google maps for the location search or the integration of communities). It may also mean that the principles of successful CIT need to be transferred. In our case, these principles are the simplicity of the work process and also the lack of prescribed process structure. These principles can then be used to revisit the rich archive of generic CIT tools and to make them more usable by reducing their feature set and prescribed structure to the absolute minimum.

## References


1. Nunamaker, J., Briggs, R.O., Mittleman, D., Vogel, D., Balthazard, P.: Lessons from a dozen years of group support systems research: a discussion of lab and field findings. JMIS 13(3), 163–207 (1996)
2. Briggs, R.O., Nunamaker, J., Tobey, D.: The Technology Transition Model: A Key to Self-Sustaining and Growing Communities of GSS Users. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34), vol. 1, p. 1061. IEEE Computer Society, Washington (2001)
3. Shumarova, E., Swatman, P.A.: Informal eCollaboration Channels: Shedding Light on "Shadow CIT". In: Proceedings of the 21st Bled eConference, Bled (2008)
4. Orlikowski, W.: Learning from Notes: organizational issues in groupware implementation. In: Proceedings of the 1992 ACM conference on Computer-supported cooperative work, pp. 361–369. ACM, New York (1992)
5. Grudin, J.: Why CSCW applications fail: problems in the design and evaluation of organizational interfaces. In: Proceedings of the 1988 ACM conference on Computer-supported cooperative work, pp. 85–93. ACM, New York (1988)
6. Grudin, K., Palen, L.: Why groupware succeeds: discretion or mandate? In: Proceedings of the fourth conference on European Conference on Computer-Supported Cooperative Work, pp. 263–278. Kluwer, Norwell (1995)
7. Fjermestad, J., Hiltz, S.R.: Group Support Systems: A Descriptive Evaluation of Case and Field Studies. JMIS 17(3), 115–159 (2000)
8. Briggs, R.O., De Vreede, G.-J., Nunamaker, J.: Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems. JMIS 19(4), 31–64 (2003)
9. Shumarova, E.V.: Authority-based and Bottom-up Diffusion of Collaboration Information Technologies - Constraints and Enablements. Dissertation. University of Koblenz-Landau (2009)
10. Tapscott, S., Williams, A.D.: Wikinomics: How Mass Collaboration Changes Everything. Portfolio, New York (2006)
11. McGrath, J.E.: Time, Interaction and Performance (TIP): A Theory of Groups. Small Group Research 22(2), 147–174 (1991)
12. Mark, G., Poltrock, S.: Groupware adoption in a distributed organization: Transporting and transforming technology through social worlds. Information and Organization 14(4), 297–327 (2004)
13. Suchman, L.: Plans and Situated Actions: The Problem of Human-Machine Communication. Cambridge University Press, Cambridge (1987)
14. von Kleist, H.: Über die allmähliche Verfertigung der Gedanken beim Reden. Kleist-Archiv Semder, Internet-Editionen (2010), `http://www.kleist.org`
15. Medina-Mora, R., Winograd, T., Flores, R., Flores, F.: The action workflow approach to workflow management technology. In: Proceedings of the 1992 ACM conference on Computer-supported cooperative work, pp. 281–288. ACM, New York (1992)
16. Bajwa, D.S., Lewis, L.F., Pervan, G., Lai, V.: The adoption and use of collaboration information technologies: International comparisons. Journal of Information Technology 20(2), 130–140 (2005)
17. Orlikowski, W., Hofman, D.: An Improvisational Model for Change Management: The Case of Groupware. In: Malone, T., Laubacher, R., Scott Morton, M. (eds.) Inventing the Organization of the 21st Century, pp. 265–282. MIT Press, Cambridge (1997)
18. Weick, K.: Making Sense of the Organization. Wiley-Blackwell, Hoboken (2000)

19. Shrage, M.: Shared Minds: The new Technologies of Collaboration. Random House, New York (1990)
20. Schwabe, G.: Objekte der Gruppenarbeit. Gabler Wiesbaden (1995)
21. Mohammed, S., Dumville, B.: Team Mental Models in a Team Knowledge Framework: Expanding Theory and Measurement across Disciplinary Boundaries. Journal of Organizational Behavior 22(2), 89–106 (2001)
22. Credit Suisse: Der Credit Suisse Beratungsprozess im Private Banking – ein Lern- und Arbeitshandbuch. Nachdruck der 1. Auflage von 2005 Credit Suisse, Zurich (2006)
23. Mogicato, R., Schwabe, G., et al.: Beratungsqualität in Banken - Was der Kunde erwartet. Was der Kunde erlebt. Solution Providers, Dübendorf (2009)
24. Belkin, N.J., Oddy, R., Brooks, H.: Ask for information retrieval. Part 1: Background and theory. Journal of Documentation 38, 61–71 (1982)
25. Brüggemeier, M., Dovifat, A., Kubisch, D., Lenk, K.: Reichard, Ch., Siegfried, T.: Organisatorische Gestaltungspotenziale durch Electronic Government: Auf dem Weg zur vernetzten Verwaltung. Edition sigma, Berlin (2006)
26. Schenk, B., Schwabe, G.: Design IT-gestuetzter kooperativer Buerger-Beratung ("Design of IT-based collaborative citizens' advisory services"). In: Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI), Göttingen (2010)
27. Schwabe, G., Schenk, B., Bretscher, C.: Enabling advisors and citizens through Citizens' Advice 2.0. In: Proceedings of the 14th Annual Conference of the International Research Society for Public Management (IRSPM), Bern (2010), `http://www.irspm2010.com/`
28. Schmidt-Rauch, S., Schär, R., Schwabe, G.: Transforming Telesales to Tele-Advisory Services in Travel Agencies – Preparing for a Multi-Channel Strategy of the Next Generation (Under Review)
29. Schwabe, G., Nussbaumer, P.: Why IT is not being used being used for financial advisory. In: Proceedings der European Conference on Information Systems 2009, Verona (2009)
30. Bannon, L., Boeker, S.: Constructing common information space. In: Proceedings of the fifth conference on European Conference on Computer-Supported Cooperative Work. Kluwer, Norwell (1997)
31. Stewart, J., Bederson, B., Druin, A.: Single Display Groupware: A Model for Co-present Collaboration. In: Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit. ACM, New York (1998)
32. Bretscher, C.: Design und Implementation eines Bürgerberatungstools im Rahmen des E-Government. Diplomarbeit, Universität Zürich, Institut für Informatik (2009)
33. Rosson, M.B., Carroll, J.M.: Usability Engineering: Scenario-Based Development of Human Computer Interaction (Interactive Technologies). Academic Press, Riverport (2002)
34. Venkatesh, V., Morris, M., Davis, G., Davis, F.: User acceptance of information technology: Toward a Unified view. MIS Quarterly 27(3), 425–478 (2003)
35. Streitz, N., et al.: SEPIA: a cooperative hypermedia authoring environment. In: Proceedings of the ACM conference on Hypertext, pp. 11–22. ACM, New York (1993)
36. Foster, G., Stefik, M.: Cognoter: theory and practice of a collaborative tool. In: Proceedings of the 1986 ACM conference on Computer-supported cooperative work, pp. 7–15. ACM, New York (1986)
37. Schwabe, G.: Telekooperation für den Gemeinderat. Kohlhammer, Stuttgart (2000)

# Design for Business Innovation: Linking the Value Chains of Logistics Service and Cargo Insurance Companies by Designing a Collaborative Service Infrastructure

Alexander C.H. Skorna[1,2], Christoph Bode[2], Oliver Baecker[1,3], Jan vom Brocke[4], and Elgar Fleisch[1,2]

[1] University of St. Gallen, Institute of Technology Management, Dufourstrasse 40a, 9000 St. Gallen, Switzerland
{Alexander.Skorna,Oliver.Baecker,Elgar.Fleisch}@unisg.ch
[2] ETH Zurich, Department of Management, Technology, and Economics, Scheuchzerstrasse 7, 8092 Zurich, Switzerland
{askorna,cbode,efleisch}@ethz.ch
[3] SAP Research CEC St. Gallen, SAP (Switzerland) Inc., Blumenbergplatz 9, 9000 St. Gallen, Switzerland
Oliver.Baecker@sap.com
[4] University of Liechtenstein, Institute of Business Information Systems, Martin Hilti Chair of Business Process Management, Fuerst-Franz-Josef-Strasse, 9490 Vaduz, Liechtenstein
Jan.vom.Brocke@hochschule.li

**Abstract.** Both, the logistics and insurance companies rely on software intensive systems and IT-infrastructure to run their core business operational. In recent years IT-improvements have resulted e.g. in better tracking and tracing capabilities for the whole logistics industry. Designing an interface in this case between the logistics and insurance value chain further enhances visibility and transparency on transportation. Though, the design of a large collaborative service infrastructure is a complex task. In this paper, we investigate whether design science supports this. The research follows design science guidelines creating a message hub based on sensor telematics technologies, which physically links the two value chains. The described IT-artefact enables logistics and insurance companies to improve their respective products and solutions with e.g. integrated risk management or active process control. This demonstrates how design science projects eventually facilitate real business innovation within networked enterprises.

**Keywords:** design science, business alignment, enterprise integration, service design, supply chain risk management.

## 1 Introduction

The business world is characterized today by globalization, trade liberalization, fierce competition, increased customer demands, and strict law obligations. Influenced by this environment, customers of logistics and cargo insurance services demand efficient and effective solutions which are fully integrated into their business processes.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 461–474, 2010.

In parallel, the recent major changes in the general insurance sector i.e. deregulation have significantly increased the need for the development and use of information technology (IT) for business innovation [1]. Nowadays customers are more price sensitive, which increases pressure for the whole insurance industry and in consequence leads to further premium erosions. Thus, some insurers as well as logistics companies are looking for new ways to differentiate their service portfolio by offering innovative products to compete with a high value-added service strategy.

IT-related service development in these industries is complex and difficult as usually unique or enterprise-specific solutions are used. However, the design of our collaborative service infrastructure is based on interfaces to existent enterprise application integration (EAI) packages or enterprise resource planning (ERP) software [2,3]. Both solutions often take the form of connecting stovepipe legacy application to interconnect an organization's silo-ed business functions and work practices for streamlining its organizational processes [4]. Regardless of the chosen solution EAI and ERP already are intended to support and facilitate cross-functional business processes [5]. But especially in the logistics industry information sharing about i.e. shipping location or cargo conditions requires manual operations or is only with an extended effort accessible. For insurance companies covering transportation risks such as spoilage or damage the current situation is even worse. Transportation processes carried out by logistics providers and conditions meanwhile are not visible to the cargo insurer, shipper, and consignee. If an insurance claim occurs, the insurance company usually does not assess the claim directly, but a neutral surveyor will be sent to the scene in case of any inconsistencies. This hinders insurers to obtain deeper understandings about claims and their causes which both also lead to an increased "opacity".

This paper aims at designing a cross-enterprise service infrastructure in order to link value chains of logistics and insurance providers. Based on sensor-telematics and localization technology a message hub is used to enhance visibility and transparency of transportation and warehousing processes. The customers will benefit from two different directions if the companies implement the message hub: (1) logistics companies are able to coordinate their supply chain network and to realize an active process control, while (2) insurance companies get direct access to cargo conditions helping them to identify transportation risks and adopting premiums respectively.

This research can be considered as reinvention, which is an innovation - in this case enterprise integration - changed by the adopters e.g. of ERP-systems in the process of adoption and implementation after its original development [6]. The design science approach is used in this case as it aims at the construction of a better IS-related solution and the utility for practice is established as a clear and common measure of its results' relevance [7]. Both, the logistics and insurance industries are not very known as innovation leaders regarding business integration. In contrast, they try to stick as long as possible to their IT-systems as changes in the business processes are complex and may jeopardize daily operation. We assume that the design science process model with its more practitioner-orientated management structure supports innovation within a strong business context and this is going to be proved in the mentioned real-case scenario.

The contribution of this paper is to showcase the value of design science research for practitioners or more general project managers handling complex IT-related projects.

Especially the structured phases of a traditional design science research process make it easy to implement these as blueprints for a real project plan with specific milestones.

Therefore, the paper follows the steps of a design science research process developed by Peffers et al. [8]. First, the problem identification and motivation are explained. The areas of risk exposure in transportation and warehousing are identified based on an analysis of insurance claims data. Second, the objectives of the proposed solution are identified to solve specific pain points of the logistics and insurance industry. Third, we demonstrate the design of an IT-artefact, which links the logistics and insurance providers' value chains. As our proposed solution is still in a conceptual state, the demonstration and evaluation phase is so far just fairly developed.

The implementation of this research is going to take place during the year 2010 by one leading European logistics service provider and one marine cargo insurance company. A multi-case study follows with end-customers from the manufacturing sector, whose usually demand insurance and logistics services in parallel. As part of the communication phase we consider amongst others this paper presentation at DESRIST 2010.

## 2 Problem Identification and Motivation

Global trade has increased with an annual two-digit growth for now 15 years and is already increasing after the economic downturn in the aftermath of the financial crisis. However, damages and losses relatively decline. The world-wide goods transport in containers improves security as the container is turned over instead of the products or pallets inside. But growing global trade leads also to an increasing value concentration and rising insured sums world-wide. Supplier dependency and variability of demands has led simultaneously to an increasing vulnerability of supply chain networks to disruption [9]. A locked and sealed container can be considered as 'black box' for all supply chain parties involved - including the cargo insurance providers. Damages or deteriorations of the cargo are very often only noticed when the container is opened for unloading at the destination port or even later at the consignee. As a result unexpected delays affect the on-carriage and the localization of the incidents or the responsible is practically impossible. Sources of risk in these specific logistics processes are e.g. theft, damage, or spoilage of goods as well as in-transit or customs delays [10]. While many companies are devoting increased resources and attention to security efforts, only little guidance is available to firms seeking to minimize their exposure to unexpected and potentially damaging or disruptive events impacting their supply chains [11].

Cargo insurance companies concentrate mainly on settlements of claims as this is their experienced core business. Logistics and insurance companies are facing high competitive pressure and a high-quality service differentiation is difficult for both. Proactive risk management integrated in transportation and warehousing or offered as integrated product as transportation service including risk engineering as well as proactive claims prevention activities are so far not established [12]. The problem behind is the lack of information about transportation conditions to identify risks globally.

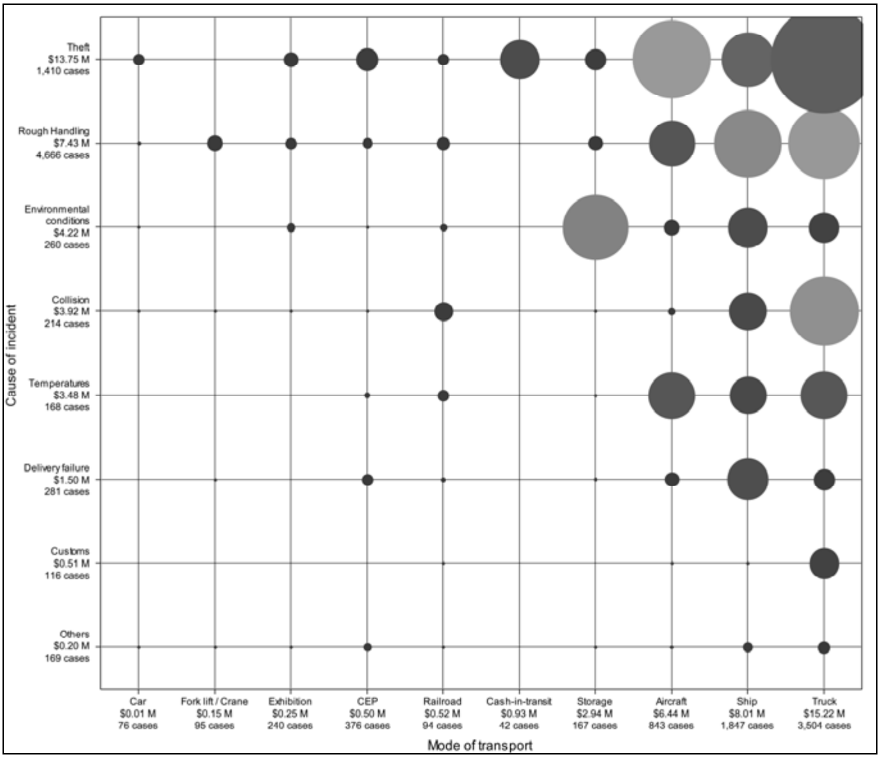



**Fig. 1.** Total loss by means of transport and cause of incident

To study this issue, we analyzed claims data from one of the largest transportation insurance providers in Europe. The sample consisted of 7,284 claims made in the recent four years (2005 – 2008) as a result of incidents in transportation. In this sample the insurance holder was either a logistics company or the shipper. The average loss given incident was US$ 19,265. The five largest incidents accounted for a loss of US$ 4 to 11 Million; all of these five incidents involved trucks and valuable pharmaceuticals. It is important to note that this data set is not representative for the entire transportation insurance industry, as it is certainly affected by the specific customer base of the particular insurance provider. Still, it provides some clues for the identification of the current major pain points in transportation.

We investigated the relationship between the modes of transport and the causes of incidents. Figure 1 visualizes the amount of loss differentiated by the various modes of transport and causes. First, Figure 1 shows that truck, ship, and air cargo transportation operations are most vulnerable to disruptions. Most incidents (both in terms of frequency and total loss) involved these three modes of transportation as the largest bubbles indicate. The average loss given incident, however, was highest for cash-in-transit, followed by storage and air cargo. Second, cargo theft (includes also pilferage), rough handling, and environmental conditions (includes condensation, contamination with fresh or sea water, fire, or natural disasters) are the most salient causes for disruptions in transportation (again, both in terms of frequency and total loss). Changes in temperatures also seem to pose a significant threat to transportation. The average loss given incident was highest for causes by changes in temperatures, followed by collision, and extreme environmental conditions. Third, cargo theft and rough handling are particularly important issues for the modes of truck, ship, and air cargo, while environmental conditions are a significant threat to in-transit storage. An interesting finding is that cargo theft is also a major problem in air cargo business.

The use of IT has permitted the development of faster, more reliable, and precisely timed logistics strategies – but has also lead to information-intensive transportation services. Adopting lean or agile principles [13], firms now require current and immediate information about the location of productive activities and the conditions of supplier goods as well as information linking business operations with available transport opportunities between different sites. In order to keep better control of the sourcing and shipping along with achieving productivity and efficiency gains, companies have started to implement collaborative strategies across their entire network. Inventory levels now are driven by real-time demand, thus synchronizing manufacturing and customer demand. To do so, well functioning and reliable transportation and communication systems are the key [14].

By following the design science research framework to establish and implement innovative cross-enterprise services, companies out of different industries are supported to understand the specific industry need of each other. This guarantees a problem-based solution design and facilitates the collaboration within the innovation process of the involved parties. The illustrative approach of the design science framework helps to understand the problem, objective, and the designed solution for the respective customers. At the end it is the customer who decides over the success of the solution and its convincing demonstration can be easily derived from the results or milestones of each design science phase.

## 3 Objectives of the Solution

One of the industries that might be most depending on real-world services consumable via the Internet and software platforms supporting highly innovative networked businesses is the insurance industry. Insurance is about managing risks with a risk being defined as the probability or threat of damage, injury, liability, loss, or other negative occurrence, caused by external or internal vulnerabilities, and which may be neutralized through pre-mediated action (prevention). Insurance is a risk-transfer mechanism that ensures full or partial financial compensation for the loss or damage caused by events beyond the control of the insured party.

The main asset of an insurance company is its understanding of risks and its ability to price those. In order to do so, insurers hold and analyze large amounts of data and their actuaries are probably able to understand many phenomena going on in the real world second to none. But IT-systems of insurance companies are currently to a large extent isolated 'island applications' not linked to any real world sources. Their understanding of the world is based on mostly manual data feeds done by back-office personnel in policy administration and claims handling. Although insurance companies transfer large risks into their book of business, they have hardly any knowledge about the status of the insured object after underwriting has taken place. It is often only when a claim happens that the insurer gets the next status update on the situation.

And that is not by the insured object itself but by the insured party who is notifying the claim, and this in some cases weeks after the claim originally has occurred.

Technology-based continuous condition monitoring has become a common practice for the transportation of deep-frozen goods or pharmaceuticals; in the European Union this approach is even a statutory requirement. Specialized logistics companies have therefore implemented indicator or logger systems which either show a color indicator (e.g., fade) or track the trend of temperature on physical memory. His helps to reveal rises in temperature or abusive storage conditions.

So far, integrated risk management has focused on claims management and risk transfer through underwriting. Addressing risks in the supply chain requires the identification of triggering events and vulnerabilities while risks are assessed mainly with support of risk management tools (processes shaded in dark gray in Figure 2). Operative risk management principles expand this traditional process chain regarding loss prevention consulting, promotion of risk controlling, and cooperation in the field of technology-supported early intervention avoiding or at least minimizing losses (processes shaded in light gray in Figure 2). Risk prevention should consequently be based on continuous monitoring of the transport and warehousing conditions aiming to confine claims amounts.

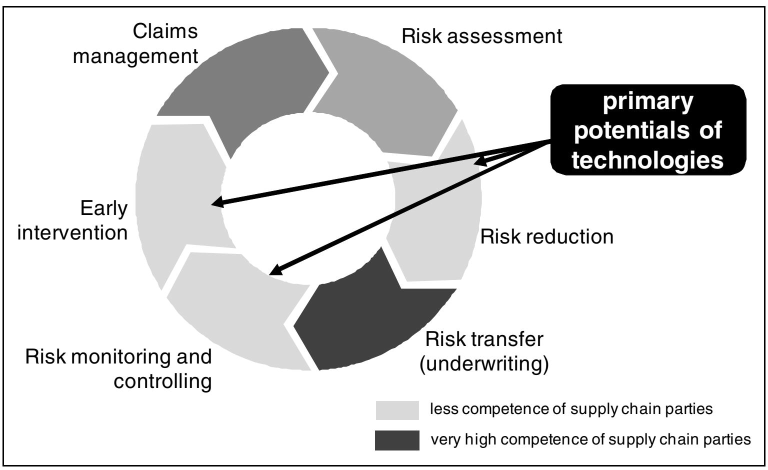



**Fig. 2.** Risk management circle in transportation

Thus, the technology-enabled risk management helps to decrease losses along two different directions: First, recurring risks can be identified based on collected data by sensors and localization systems. Adjusted transport planning optimizes risks on the long-term. Second, continuous condition as well as integrity monitoring of goods and transportation vehicles, containers, and trailers allows responding to unforeseeable exceptions in real-time. If critical values are exceeding a pre-defined range, an alarm would be generated with exact timestamp. Henceforth, counteractive actions can be initiated even before a serious supply chain disruption occurs. Aiming at realizing the above stated operational improvements by enhanced technologies, condition and integrity monitoring systems should consist of the following modules [12]:

- World-wide, high-resolution (i.e., down to street levels) self-contained localization of containers, trailers and other transportation vehicles based on satellite or mobile phone networks featuring a real-time positioning and tracking [15].
- Sensor technology that is capable to monitor temperature, humidity, shocks and gases inside the containers or transport vehicles, and which records the conditions in dedicated intervals. Motion and light detectors as well as door sensors improve transport security and contribute to threat protection [16].
- Communication systems that allow sending sensor and positioning data in case of an exception or alarm. Communication is usually carried out by common mobile phone network derivates or satellite networks. Server or integrated enterprise applications receive the data packages and visualize the raw data user-orientated in web-based portals.

Localization, sensor, and communication systems are all part of ubiquitous computing technologies which connect things in the real world to the internet in order to provide information on anything, anytime, anywhere. Applied to objects such as containers and transportation units, they could thus react and operate in a context-sensitive manner appearing to be “smart” [17]. All technology-related modules function as a core body or message hub enabling the linkage of the two value chains. Communication technology then transfers the collected data to the enterprise applications of the logistics and insurance companies.

The business objectives can be divided into end-customer management and value creation aspects. The latter aspect aims at a reduction of claims due to the higher process transparency. Risks can be actively identified and controlled within the customers' supply chains. Decreased damages and claims lead further to a lesser probability of business interruptions. From a logistic process point of view the technologies allow a significant reduction of cycle times as the shipping can better be coordinated. This leads in consequence to a reduction of safety stocks in the customers supply chain which yields bounded capital. The customer management aspect aims at a differentiation strategy for the insurance and logistics solutions to compete not with the lowest price but with the best value-added service portfolio. Both service providers are able to create a unique selling point when linking their value chains through the message hub. Thus, this case is another good example that technology is a significant tool for differentiation of services [18]. Insurance and logistics companies are capable to sustain their market positions and generate growth. Existing customers of both service providers profit form an increased supply chain quality and integrated proactive prevention solutions.

## 4 Designing a Collaborative Service Infrastructure

The research is theoretically grounded by the two concepts of supply chain risk management (SCRM) and supply chain event management (SCEM). Risk itself is an elusive construct that has a variety of different meanings, measurements and interpretations depending on the academic research field. In this context a hazard-focused interpretation common in risk management is used which presents risk in terms of: Risk = Probability (of a given event) x Severity (negative business impact) [19]. Identifying and assessing likely risks and their possible impact on operations is a complex and difficult task for a single company. However, to properly assess vulnerabilities in a supply chain, firms must not only identify direct risks to their operations but also the risks to all other entities as well as those risks caused by the transportation linkages between organizations [20]. The process of SCRM, as discussed by Closs and McGarrell [21], refers to: "the application of policies, procedures, and technologies to protect supply chain assets from theft, damage or terrorism, and to prevent the unauthorized introduction of contraband, people or weapons into the supply chain." Risk management related to the transportation and logistics chain includes processes which reduce the probability of occurrence and/or impact that detrimental supply chain events have on the specific company [22].

Supply Chain Event Management provides control and the pro-active management of processes. Enterprises manage their processes based on comprehensive planning. However, with growing planning reliability and more complex processes, the sensitivity of planning increases by occurrence of unexpected events. Especially in the era of "on-demand" and "just-in-time", manufacturers have no possibility anymore to react to unscheduled events, e.g. great demand, and to fulfill additional orders. As a consequence, the forecast is stepped up and production capacities are extended [23].

SCEM software enables enterprises to react quickly and to some extent automatically to unexpected events – without re-working the planning completely. SCEM applications achieve this by pro-active messages to the important process participants on the occurrence of certain events respectively on the non-occurrence, e.g. going below minimum inventory, delayed deliveries, etc. Within SCEM applications such as track and trace, online-dispatching of goods or vehicles are bundled together. In short, SCEM tools continuously monitor the logistics processes across the network on the basis of predefined events. If a specific instance occurs too late, unexpectedly, or not at all, the SCEM tool generates a report giving notice of the deviation from the original plan. Ideally, SCEM tools help to identify plan deviations within the supply chain in real time. A company can then respond immediately to sudden bottlenecks or failures with the appropriate adjustment measures [23, 24].

The transparency is the greatest benefit and advantage of SCEM and helps to mitigate transportation risks. However, problems are often detected too late and have to be solved in a laborious and expensive way. Reactive behavior ensures neither the service level nor customer satisfaction. The transparency gained by SCEM primarily offers a continuous fine-tuning of processes and planning. Thus, unscheduled events and exceptions turn into chances to optimize planning, to avoid delayed deliveries, and to reduce costs. In the long term, SCEM software does not only improve efficiency due to a proactive manner but also increases customer satisfaction through a flexibility gain [24, 25].

The envisioned architecture is shown in more detail in Figure 3. At the core is the service delivery platform, through which services are discovered, deployed and orchestrated. This platform will be deployed centrally at the enterprise level, but platform instances at different enterprises can be federated in order to allow the construction of business applications that run across enterprises. The developed interface and linkage between the two enterprises A and B can be interpreted as a software-as-a-service solution platform provided by a system integrator e.g. SAP. As part of a closer connected Business-to-Business (B2B) communication this platform supports the IT-related horizontal integration of information flows between the two enterprise partners.

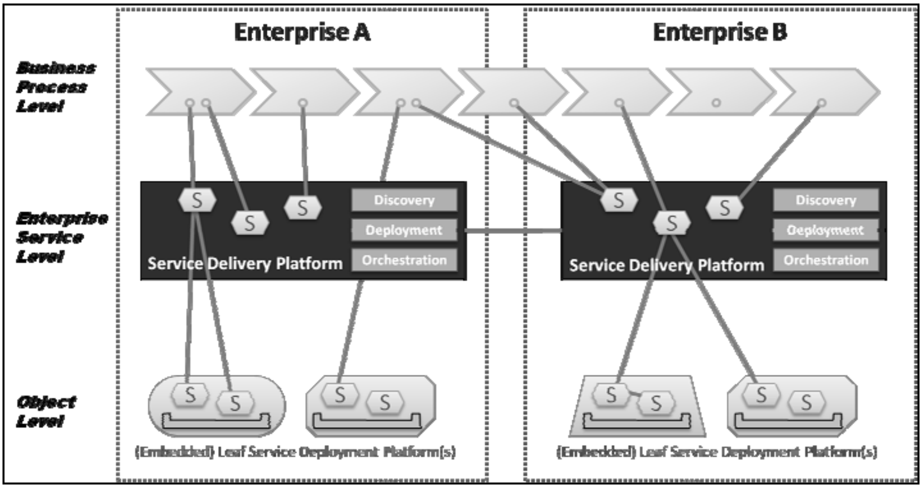



**Fig. 3.** High-level architecture of the service platform

However, not all services will be executed centrally. With increased computational and communication capabilities embedded in physical objects, processing power is shifting towards the edges of the network. Intelligent mechanisms for data aggregation, filtering, fusion and conversion may be deployed to the object level and executed on so-called leaf devices embedded in or attached to the things in the real world. Utilizing the computing power and intelligence on leaf devices allows enterprise systems to propagate data and services to these new, potentially underutilised, run time environments. It enables early aggregation and filtering of raw sensor data, local reasoning and decision making as well as the direct integration of actuators, extending the real world visibility concept with real world interaction. Supply chain visibility is the key enabler to manage risks close to the actual execution. Visibility enables the immediate identification of critical situations and an optimal response to them.

Leaf devices can be embedded systems, sensor nodes, actuators etc. In short, the hardware and protocols used on this level are very heterogeneous, as indicated by the different shapes on the object level in Figure 3. In order to allow the deployment and execution of services on this level, the different hardware platforms must include a leaf service deployment platform adapted for the specifics of the hardware.

Deploying services onto leaf devices and the local execution of software can bring significant benefits, but this comes at a cost of increased management complexity. Because of the increased management complexity it has to be determined for each application on a case-by-case basis if the distributed business logic is appropriate. The advantages and disadvantages of a distributed approach – and hence criteria which approach to choose in a particular scenario – are listed in Table 1.

The proposed solution comprising a message hub linking the value chains of logistics and insurance providers aims at introducing sensor-based telematics technology in transportation which has the potential to reduce theft and damages in transit and subsequent economic loss due to supply shortfalls and business interruptions. The technology contributes to an improved security throughout the whole transportation network and supports anti-terrorism laws as well as quicker customs clearance.

**Table 1.** Advantages and disadvantages of a decentralized approach for business logic execution

| Advantages | Disadvantages |
| --- | --- |
| 1. *Responsiveness* of the overall system, since unnecessary 'expensive' communication round-trips are eliminated.<br>2. *Scalability*, since the execution of business logic is distributed.<br>3. *Network independence*: The system will also work when there is no connection to backend systems, e.g., during transportation or in temporary storage areas. | 1. *Manageability*: A distributed system is more complex to manage; services have to be deployed reliably, business rule and configuration changes need to be propagated.<br>2. *Reliability*: Guaranteeing that all components as well as the system as a whole work as expected is much more difficult. |

Today insurers have only little knowledge about the status of the goods they are insuring. They do not know in what shape the goods are, if they are where they are supposed to be, if they are stolen or not, etc. The deployment of a combination of communication, localization (e.g. GPS, Cell-ID, WLAN, RFID, and Bluetooth), and sensor technology (e.g. temperature, shock, humidity, movement, and door activities) on transport containers or trailers creates the basic collaborative infrastructure to link the tracked data via the service platform with respective enterprise systems, as it is shown in Figure 4. The enterprise systems store and analyze the data in order to recognize problem events like damaged or spoiled goods or misrouted containers. Thus, the message hub has to be considered as shared resource, which links the value chains of logistics and insurance companies by transferring data into each company's business application through a software-as-a-service application. The information then can be used in each value chain to optimize the own products and solutions for the end-customer. This includes early intervention as part of a proactive risk management to avoid damages of the goods to be conveyed based on real-world data, new pricing models, impacts on accumulation of risk (e.g. multiple containers from different sources on the same ship or warehouse), insurance on demand and the like. The logistics company is able to use the sensor data e.g. as part of an improved information management for its customers.

Besides adding customer value the message hub improves internal processes of each company's value chain as well. The insurance companies can further optimize their risk portfolio including the new given transparency into the police management and in terms of risk transfer. Higher resolution concerning cargo conditions also helps to speed up the claims management process. For logistics companies the proposed solution has the potential to optimize e.g. asset management or the whole product management such as better coordinated timetables between air or ocean freight and land transportation.

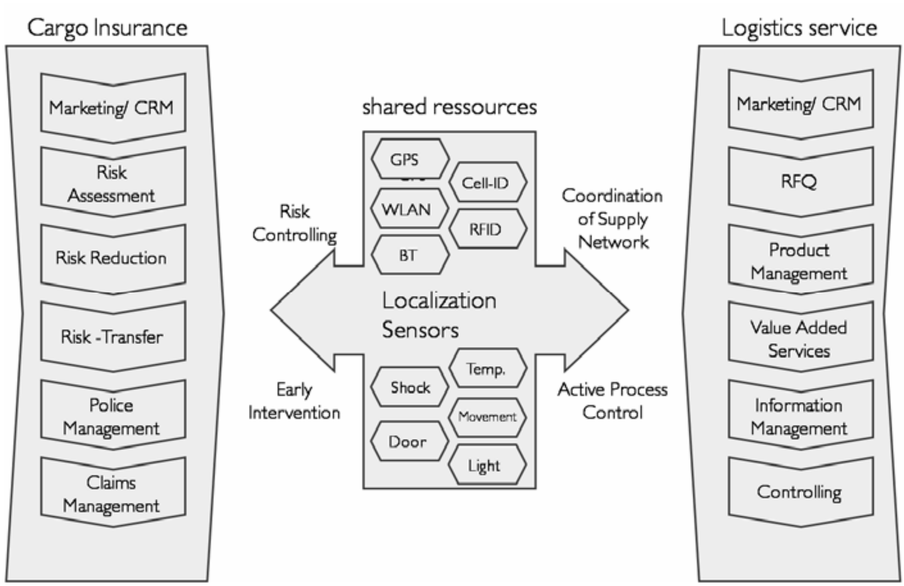



**Fig. 4.** Message hub as linkage element in between of the two value chains

Due to ongoing price pressure and standardization efforts, technology is becoming smaller, more affordable, and more powerful, which creates these enhanced business values. In the field of transportation and warehousing, the use of technology is commonly used today when tracking or tracing goods through the supply chain. Ubiquitous computing is a logical next step in the development of business information technology. Integrated information systems like enterprise resource planning (ERP) systems have linked firm functions and departments, and thus enabled consistent business processes. Internet and e-business systems have extended these processes beyond the boundaries of organizations and have become integral to modern business networks [26]. The concurrence of several technologies e.g, sensor technology, localization and communication allows for the vision of an "internet of things". The platform then integrates various technologies in a vertical manner to achieve more transparency on the companies' value chains with enhanced data availability. Finally, the designed massage hub enables to create a collaborative service infrastructure, which integrates business services horizontally and vertically through only one single platform.

## 5 Evaluation and Discussion

The goal-oriented execution of the claims analysis reconfirms the value of information in supply chain management. Henceforth, the integration of technology comprised in the designed message hub into a distinct risk management concept clearly enables to detect potential weaknesses in supply chains prior to failure or mistreatment. Besides, the potential supply chain security relies on other “softer factors” such as the development and continuance of business relationships among the supply chain parties. Communication between the involved companies allows for sharing information, risks, and rewards were identified as critical factors for effective supply chain risk management [11]. A study by Peleg-Gillai et al. [10] confirms quantifiable benefits of investments in supply chain security including a 38% reduction in theft, loss, or pilferage, a 14% cut in excess inventory, a 49% reduction in cargo delays, and a 29% reduction in overall transit times.

As the proposed service infrastructure hub is not yet deployed in a real-case scenario, only our evaluation criteria can be demonstrated in this section. Our evaluation criteria catalogue triggers four different dimensions:

1. Reduction of claims aims at less damage probability and severity. As consequence the transportation risks decrease.
2. Improving transparency in the pricing of insurance premiums. So far cargo insurance companies usually use the customers' turnover value as reference and offer lump-sum premiums as specific proportion of the turnover. More accessible information through the message hub leads to a more risk-adequate pricing.
3. Improved transportation processes which lead to high quality supply chain network with a better coordination and communication be offered.
4. The linkage of the value chains of insurance and logistics companies enables for the respective parties product and service innovation e.g. an integrated risk management. This allows differentiating with a high service strategy to be competitive in very price-sensitive industries.

The message hub with its comprised technology elements will be deployed by one leading European insurance company in cooperation with the worlds-leading logistics service provider. We therefore plan a profound evaluation process as a multi-case study during the year 2010, as the concept is going to be presented as additional service to existing customers and within request for quotation (RFQ) as well as tender processes by both companies. How effectively a company can quantify the impact may also depend on its ability to identify collateral benefits of various investments in security and resilience. An investment in telematics technology can improve not only security by real-time tracking and monitoring cargo movements but also visibility. A better visibility leads to decreased inventory requirements and improved service levels. Standard operating procedures (SOP) developed by shippers and carriers how to handle and protect goods will benefit from avoiding loss and damage. Moreover, involving marine cargo insurance companies in this concept may lower insurance premiums, which positively affects premium calculation and potentially eases claims administration.

During the design process for this kind of business innovation the design science research framework has been truly supportive. Especially the design science research guidelines underline that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artifact. Following a design science approach for the deployment of enterprise integration solutions is also an effective procedure as some of the conceptual design knowledge gathered in the design process might be reused in other related projects [4]. According to Hevner et al. the objective of research in information systems is to acquire knowledge and understanding that enable the development and implementation of technology-based solutions to heretofore unsolved and important business problems [27]. This helps in particularly to communicate the essence of the IT-artefact to the involved companies. Both project partners come from industries in which IT-systems play an important role for its daily running business. Quick and aggressive changes in the IT-infrastructure are therefore not possible. With the design science approach and adopting established business integration applications in order to link the companies' value chain through the message hub, both are able to create new business opportunities. Summing up, it is the perfect time for the implementation of a technology-enabled risk management achieving higher efficiency and productivity gains, thanks to today's real-time availability of information in case of any disruptions during transport. Shared visibility related to freight conditions allows corrective actions executed by the responsible forwarding agent, which helps to monitor risks as well as to reduce the probability and extent of damages.

**Acknowledgments.** This research is partly funded by the insurance lab (I-Lab), a joint research initiative of ETH Zurich and University of St. Gallen (HSG), which focuses on the deployment of pervasive technologies and the assessment of its business impacts within the insurance and assurance-related sectors. We would like to thank our project partners and the anonymous reviewers for their valuable input.

## References

1. Barrett, M., Walsham, G.: Managing IT for Business Innovation: Issues of Culture, Learning, and Leadership in a Jamaican Insurance Company. Journal of Global Information Management 3(3), 25–33 (1995)
2. Markus, M.L., Tanis, C.: The enterprise systems experience - from adoption to success. In: Zmud, R.W. (ed.) Framing the Domains of IT Management Research: Glimpsing the Future through the Past, pp. 173–207. Pinnaflex Educational Resources, Cinciannati (2000)
3. Lee, J., Siau, K., Hong, S.: Enterprise integration with ERP and EAI. Communications of the ACM 46(2), 54–60 (2003)
4. Umapathy, K., Puaro, S., Barton, R.: Designing enterprise integration solution: effectively. European Journal of Information Systems 17(5), 518–527 (2008)
5. Sharif, A.M., Irani, Z., Love, P.E.D.: Integrating ERP using EAI: a model for post hoc evaluation. European Journal of Information Systems 14(2), 162–174 (2005)
6. Rice, R.E., Rogers, E.M.: Reinvention in the Innovation Process. Science Communication 4(1), 499–514 (1980)
7. Winter, R.: Design science research in Europe. European Journal of Information Systems 17(5), 470–475 (2008)
8. Pfeffers, 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: Chatterjee, S., Hevner, A. (eds.) Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), pp. 83–106 (2006)
9. Wagner, S.M., Bode, C.: An empirical investigation into supply chain vulnerability. Journal of Purchasing and Supply Management 12(6), 301–312 (2007)
10. Peleg-Gillai, B., Bhat, G., Sept, L.: Innovators in Supply Chain Security: Better Security Drives Business Value. The Manufacturing Innovation Series. The Manufacturing Institute, Washington (2006)

11. Autry, C.W., Bobbit, L.M.: Supply chain security orientation: Conceptual development and a proposed framework. International Journal of Logistics Management 19(1), 42–64 (2008)
12. Skorna, A., Bode, C., Wagner, S.M.: Technology-Enabled Risk Management along the Transport Logistics Chain. In: Wagner, S.M., Bode, C. (eds.) Managing Risk and Security, pp. 197–220. Paul Haupt, Berne (2009)
13. Christopher, M., Peck, H.: Building the resilient supply chain. International Journal of Logistics Management 15(2), 1–14 (2004)
14. Viswanathan, N., Littlefield, M.: Lean Manufacturing. Aberdeen Group, Boston (2009)
15. Hillbrand, C., Schoech, R.: Shipment localization kit: an automated approach for tracking general cargo. In: Proceedings of the Sixth International Conference on Management Mobile Business (ICMB 2007), Toronto, Canada, pp. 46–52 (2007)
16. Schaefer, S.: Secure trade lane: a sensor network solution for more predictable and more secure container shipments. In: Dynamic Language Symposium an Companion to the 21st ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications (OOPSLA 2006), Portland, OR, pp. 839–845 (2006)
17. Mattern, F.: The vision and technical foundations of ubiquitous computing. Upgrade - the Online Magazine for the Information Technology Professional 2(5), 3–7 (2005)
18. Sauvage, T.: The relationship between technology and logistics third-party providers. International Journal of Physical Distribution and Logistics Management 33(3), 236–253 (2003)
19. March, J.G., Shapira, Z.: Managerial perspectives on risk and risk taking. Management Science 33(11), 1404–1418 (1987)
20. Jüttner, U.: Supply chain risk management: Understanding the business requirements from a practitioner perspective. International Journal of Logistics Management 16(1), 120–141 (2005)
21. Closs, D.J., McGarrell, E.F.: Enhancing Security Throughout the Supply Chain. Special report to the IBM Center for the Business Government, Washington, DC (2004)
22. Zsidisin, G.A., Panelli, A., Upton, R.: Purchasing organization involvement in risk assessments, contingency plans, and risk management: an explorative study. Supply Chain Management: An International Journal 5(4), 187–197 (2000)
23. Otto, A.: Supply Chain Event Management: Three Perspectives. The International Journal of Logistics Management 14(2), 1–13 (2003)
24. Dießner, P., Rosemann, M.: Supply Chain Event Management: Managing Risk by Creating Visibility. In: Ijioui, R., Emmerich, H., Ceyp, M. (eds.) Strategies and Tactics in Supply Chain Event Management, pp. 83–98. Springer, Heidelberg (2008)
25. Bodendorf, F., Zimmermann, R.: Proactive Supply-Chain Event Management with Agent Technology. International Journal of Electronic Commerce 9(4), 57–89 (2005)
26. Fleisch, E., Dierkes, M.: Ubiquitous Computing: Why Auto-ID is the Logical Next Step in Enterprise Automation. White Paper, Auto-ID Center, University of St. Gallen (2003)
27. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 41–90 (2004)

# A Requirements Engineering Method Designed for the Blind

Tuure Tuunanen[1], Ken Peffers[2], and Simeon Hebler[3]

[1] University of Oulu, Department of Information Processing Science, P.O. Box 3000, 90014 University of Oulu, Finland
tuure@tuunanen.fi
[2] MIS Department, College of Business, University of Nevada Las Vegas
4505 Maryland Pkwy, Las Vegas NV 89154-6034 USA
ken.peffers@unlv.edu
[3] Deloite, Magnusstrasse, 50672 Alstadt-Nord
Köln, Germany
mail@simeon.de

**Abstract.** This paper motivates, designs and demonstrates a method for requirements engineering (RE) with blind users. We motivate the need for the method by pointing out that, although much has been done to accommodate blind users' accessibility to systems, little formal research has addressed the need to include members of this large and growing population in RE activities. We develop objectives for a method to address three problems that potentially affect such participation. We design a method to address each of the problems We demonstrate its use in a RE effort among users in New Zealand and Germany to develop requirements for mobile service applications and features for blind users and validate its use in a follow-up survey. Our theoretical evaluation of the process shows that we were able meet most of the objectives for a blind user RE method. The proposed method should be a beginning for research efforts.

**Keywords:** requirements elicitation, requirement discovery, requirements engineering, blind, design science research.

## 1 Introduction

Approximately 161 million people in the world are vision impaired, according to the World Health Organization [1], including 45 million who are blind. This number is increasing, largely because of increased life expectancy and age-related causes for blindness, such as cataracts and macular degeneration. Clearly, there is a need to insure that information systems are designed to be accessible and usable for this very large population of users and potential users.

Blind people have special requirements for accessing information systems. Computers and mobile devices are largely visual media, since user interfaces are most often designed to accommodate information presentation on visual displays. Special arrangements are required to make the technology accessible for the non-sighted, including consideration of input and output hardware and usable representation of information for applications and content [2].

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 475–489, 2010.

Early on, there was little consideration of how alternative input or output mechanisms could effectively process information for the blind. In recent years, however, voluntary and legislated efforts have begun to address these needs. IT vendors have integrated accessible facilities in operating systems and third-party vendors made special accessibility hardware available to supplement standard hardware. Legislation and regulation required efforts to render systems accessible to disabled persons, including the blind. While these efforts promise the potential to make systems accessible and more usable for blind users, ironically, little has been done to insure that RE activities are effectively accessible to the blind, so that the blind could readily be included among participants in the RE process. Without such participation, blind users can access systems designed for the sighted, but their preferences for functionality cannot readily be incorporated into those systems.

Can we design an RE method that is highly accessible for blind participants? The blind may require more than the obvious, simple accommodation for lack of sight, in order to accomplish effective participation in RE. We have identified four general problem areas, based on Davis [3] and Browne [4]: motivation, information processing, complexity, and communication. We assert that an RE process to include blind persons must accommodate elements in these areas to be effective.

In this study we design a blind user RE method that incorporates accessibility techniques, as well as techniques adapted from the standard repertoire of data collection, to accommodate the needs of the blind with respect to three problem areas that we identified, to allow blind users and potential users able to effectively participate in the data gathering and validation parts of the RE process. We employ the design science research paradigm [5] to identify the problem and objectives of a solution, and to design a process to address the objectives. We demonstrate the use of this method in the context of RE for mobile services and features that would benefit blind users. Finally we evaluate the identified services and features in a follow-up survey of blind users.

This paper makes several contributions to the RE literature. It:

1. makes the first effort in the literature to specifically develop a method for data collection, and analysis of user requirements for blind users;
2. demonstrates the method with the development of requirements for applications and features that would benefit blind users; and
3. theoretically evaluates the efficacy of the method by referring back to design objectives.

This paper is organized as follows. In Section 2 we detail the problems and objectives of a solution to the problem. Section 3 briefly outlines the designed process. Section 4 demonstrates use of the method to develop the requirements for mobile service applications and features for blind users. Section 5 evaluates the efficacy of the designed method by comparing the intended effects of its functionality with the objectives for a designed solution, described in Section 2. Section 6 concludes the paper with a recap of what we accomplished and a brief agenda for future research. The study on which the paper is based follows the design science research methodology [6], although for reasons of brevity, not every step is fully explicated here.

## 2 The Problem and Objectives for a Solution

Researchers have observed four major sources for many of the failures affecting RE: the motivation and ability of RE participants to be effectively engaged in the process; human information processing constraints; the complexity of information system functionality and features; and communication among the parties to RE, including subject participants, analysts, designers, and managers [3, 4]. Here we use three of the four items in this framework to consider how blindness affects the ability of members of this population to effectively participate in RE and we develop objectives for a solution in an RE process[1].

### 2.1 Human Information Processing Constraints

Limitations on working memory affect the ability of participants to perform complex evaluation tasks during RE. Long-term memory limitations affect participants' ability to provide all relevant data about their needs and preferences. RE participants often may not recall system needs and preferences that have not been brought to their attention lately, particularly if they have worked around the needs in their work or daily lives or if they assume that their preferences cannot be met.

Typically, RE techniques rely heavily on visual materials, e.g., lists, scenarios, and documents, as aids to short and long-term memory. The absence of support from visual aids dramatically increases demands on memory for the blind participant. Most particularly such demands affect the participant's ability to engage in activities, such as rank ordering item lists, choosing from lists, or manipulating items in complex relationships. With attention and training, blind persons can compensate for the lack of visual aids with improved working memory skills, but only to a very limited extent [7]. In addition, the absence of visual material support makes it difficult for blind participants to identify missing items from data they have provided to analysts from long-term memory.

**Objective:** An effective RE method for blind users would avoid excess demands on working and long-term memory for participants in order to enable participants to provide complete data about their preferences and needs, as well as to effectively evaluate competing ideas for functionality and features.

### 2.2 Complexity of Information System Functionality and Features

The inherent complexity of information systems is known to lead to high variability in the expressed preferences of RE participants. Participants often may not fully understand the organizational or technical possibilities of the system or application for which their preferences are being sought. This is particularly true where the participant is unfamiliar with the organization or has little or no experience or conceptual understanding of the underlying technology. It may also affect members of the organization who are dealing with new systems for which the scope of activities is outside the experience of the members [8-10].

[1] For reasons of brevity "motivation and ability of RE participants to be effectively engaged in the process" is not discussed here, however, it was included in our project design.

For sighted RE participants the most common way to overcome this constraint is through visually oriented stimuli, such as presentations or prototypes. For blind participants the need to eschew visual materials means that the stimuli necessary to aid in idea generation for functionality and features require substantial adaptation.

**Objective:** An effective RE process for blind users would be designed to elicit needs and preferences for functionality and features from participants who are unfamiliar with underlying processes and technology, without the use of visual materials.

### 2.3 Communication among the Parties to RE

Communication among RE participants and other parties to RE, such as analysts, designers, and managers, is an acknowledged source of incomplete or misunderstood needs. Participants are likely to lack understanding of the professional and organizational language of the analyst and they are unlikely to fully understand the objectives and values of the organization. In addition, they are likely to be members of underlying individual cultures that are different than those of the analyst, because of differences in nationality, socio-economic background, lifestyle, etc. As a result, the structure of their beliefs about how the world works, values, and goals may differ to the extent that the analyst may find it difficult to capture their preferences completely or to express them accurately.

For RE to succeed the needs of the participants must be captured completely and expressed accurately, but also to be represented in ways that are understandable and useful for clients of RE, including managers and system designers.

For blind participants the differences between blind and sighted persons' perspectives must be added to the communication problems described above, i.e., blind persons have all of the cultural, educational, and situational variation of the sighted population, in addition to their blindness, which can be expected to affect their beliefs, values, and goals, as well as the structure of their knowledge [8].

**Objective:** An effective RE process for blind participants would capture and aggregate data in a manner that is structurally very flexible and it would represent data in a manner that is useful and understandable for managers and designers.

## 3 Design of a Solution

Here we briefly outline procedures for the designed RE method for use with blind participants, designed to meet the objectives described in Section 2. Some of the reasoning for specific features is omitted for reasons of brevity, including, notably discussion of the state of the art in accessibility and RE and discussion of the design process. A process description of the designed method is depicted in Figure 1 below.

**Determine project scope and participant characteristics; recruit participants; and collect stimuli.** Recruit participants with an assertive, multi-layered approach that includes communications through blind organizations and/or snowballing, each of which is intended to enhance trust by referral of the solicitation through known, trusted entities, i.e., acquaintances or known organizations. Screen participants for desired characteristics, including "lead user" attributes. Lead users are a minority of savvy users who tend to use new technology early in its life cycle and whose experience with technology can be exploited to predict future trends [11] and they are expected to be less constrained by complexity and new technology than average users. Use interview scheduling as an opportunity to collect stimuli by asking the prospective participants for ideas [12]. The use of stimuli ideas in data collection is intended to reduce demands on long-term memory that result from lack of visual aids. Document and aggregate the collected ideas to get a stimuli list for the laddering interviews. The number of stimuli depends on the study, but often 4-5 is a good number [9] to avoid information processing constraints of participants. Simplify the collected stimuli for oral presentation and to accommodate working memory limitations of oral presentation, by reducing the number of items in the stimuli and the verbal complexity of their presentation.

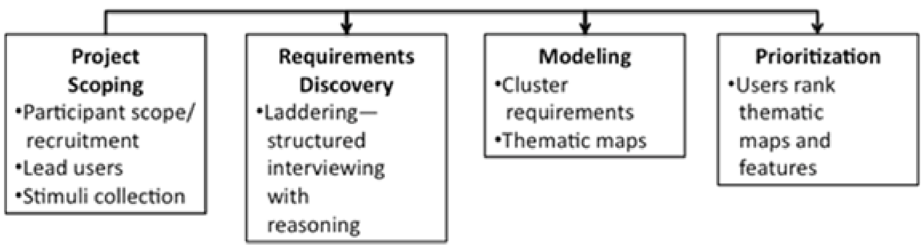



**Fig. 1.** The designed method

**Use laddering for requirements discovery, collect data from the participants, and record them as ladders.** Use one-on-one oral interviews to optimize convenience to participants. Structure the interviews and record the resulting data, using the laddering method [12]. Laddering is based on personal construct theory (PCT) [13]. Kelly found, through his experience as a school psychologist, that if he understood his clients' models of how the world works, he could better understand their behavior. PCT is concerned with peoples' models of the states of the universe, the consequences of those states, and impacts on their individual values. Laddering is a data collection and recording technique that captures states, consequences, and values, in this case preferences for new system attributes, expected outcomes from those attributes, and the expected affects on the participants' values. The laddering technique is useful because it aids communication among stakeholders by collecting preferences and reasoning data, without any other structural constraints.

Starting with the participant's choice of the most appealing stimuli (reducing complexity), ask "how would this work for you" to elicit preferences for specific features. For each preference continue with "why would that be important for you" questions to elicit the steps in a ladder of consequences and values [4, 13]. Continue with the participant's second choice stimulus to repeat the process. Record the data in a manner that preserves the integrity of each ladder, e.g., as cells in a spreadsheet column, and make an audio recording of the interview. Going through two stimuli will likely require 45 to 60 minute interview. After this the participant starts to be tired [9] and it is therefore better to either conclude the interview or consider continuing on another time if you have chosen to use more than two stimuli.

To record participant views on the importance of their ideas, ask each to prioritize his/hers own ideas. In a two step rank-ordering process to reduce working memory demands, ask the participant to identify and rank order the several most important of his/her feature ideas and then separately to rank-order the remaining ones.

**Model requirements by clustering interview data into themes and creating thematic maps.** Code the individual statements in the data as attributes, consequences and values items. Determine a number of themes, ideally no more than seven to accommodate working memory limitations, that represent the data in the ladders and assign each ladder into one or more themes.

Create thematic maps, graphical network representations that depict the links between attribute, consequence and value codes as they occurred in each ladder and are aggregated into themes. A thematic map represents a consensus model consisting of, from left to right, descriptions of desired system attributes, the reasoning and consequences behind them, and associated participant values. The thematic maps connect nodes with association (reasons why) links that may or may not be causal. The horizontal dimension in the maps represent an 'attribute—consequences—value' dimension.

Arrange the thematic maps to show sub-themes, i.e. major feature sets of the application. Sub-themes will reduce information processing constraints. Continue with clusters of attributes, consequences and values, within the sub-themes. These maps provide a graphical picture of all ladders in sub-themes.

**Prioritize requirements according the importance of themes and individual features.** The first step is to determine which themes represent the most desirable or important application bundles. To reduce complexity of the task and working memory demands, create brief summary descriptions that depict a descriptive label for each of the themes and a list of the included features. Present the participants with the summaries, asking them to rate the importance of each theme using a Likert-type scale.

Continue to the feature level by developing scenarios that briefly describe attributes and consequences for groups of features, so as to appeal to a user's imagination about having several features available in a system. Depending on the complexity of each theme, scenarios should contain two or more features. Read the scenarios to participants and ask them to name the features that are most interesting or important to them and record the importance responses as Likert scores and identify the most important features for each participant.

Reduce communication constraints further by asking participant to give reasons for preferences. For a participant's most preferred features ask him/her, "why is this feature important to you?" to elicit reasoning for the choices. This post-analysis feedback instrument will serve as a constructive critique or confirmation of the analysis phase. If necessary, the users' critiques can be incorporated into the refinement of the suggestions.

## 4 Demonstration of the Designed Solution

Here we demonstrate the use of the designed method in a case study where we use it to develop requirements for mobile services for the blind. We carried out the study in

New Zealand and Germany in 2006. We start with data gathering and continue with analysis, presentation and results.

### 4.1 Determine Project Scope and Participants

Target participants for the study included blind lead users. We foreswore the use of participant incentive payments, thus insuring that recruiting participants would be challenging. Our objective was to recruit at least 20 participants, the minimum necessary to make the study results meaningful [9, 10, 14]. To recruit participants in New Zealand we employed communications channels of the Royal New Zealand Foundation of the Blind. We sent several hundred emailed invitations to the foundation's lists, inviting list members to take a short screening survey. We posted an announcement on the foundation's telephone oral newspaper service. In addition, we asked willing participants to nominate other likely participants and contacted them, either by email or through the referring participant. After four weeks, these efforts yielded five participants.

We continued recruitment by making a presentation at a foundation-training center, where we explained our research objectives in the classroom. This yielded five more participants. Foundation staff contacted some people directly, yielding three more and one staff member agreed to participate. In all, we recruited 14 potential participants in New Zealand.

Next we turned our attention to Germany, where *Trierische Tonpost* publishes a monthly "spoken magazine" for 850 blind and vision-impaired subscribers across Germany. A solicitation in this medium yielded nine, for a total of 23 participants.

To screen for lead users, we asked a series of questions, derived from a screening device used elsewhere [10], to assess the use of new technologies. Part 1 contained six statements about the use of mobile and assistive technologies. Part 2 posed two questions about mobile and adaptive technology the participants have used. The survey was available as an Internet questionnaire and by telephone. One participant was screened out of the study in this process.

This left us with 22 participants, for whom table 1 shows sample demographic data. Note that the age distribution is more heavily weighted with people who are less than 40 years old and the sample was male dominated.

**Table 1.** Participant demographic profile

| **Age Distribution** | | **Sex** | |
|---|---|---|---|
| 18-29 years | 39% | Male | 78% |
| 30-39 years | 22% | Female | 22% |
| 40-49 years | 28% | | |
| 50+ | 11% | | |

### 4.2 Collect Data from Participants

All participants that qualified for participation were telephoned to invite them to participate in individual interviews. At the end of each phone call they were asked to give one idea for a potential system that could be of interest to them. Some participants contributed three or more ideas. After the first seven invitations we developed a preliminary list of stimuli for use with the first interviews. This was gradually extended and refined with new ideas for the remainder of interviews. This allowed us to start with the interviews before all participants had committed to taking part.

We conducted the interviews, lasting an average of 35 minutes each, in person or by telephone, using the laddering method [12]. We audio recorded each interview and took notes on an electronic spreadsheet. Telephone interviews served to minimize potential resistance among the participants to unnecessary travel, which is more arduous and causes more anxiety for the blind than for the sighted. At the same time the structure of the interviews was designed to capture rich preference data with reasoning.

Table 2 illustrates the resulting raw interview notes. Shown the stimuli ideas about new mobile services, the participant chose the two most interesting to him/her or volunteered his/her own ideas. Next we asked, "what would be an interesting feature of the service/product?" and recorded the response as an attribute, shown in Table 2 with capital A prefix. Then we asked a series of "why this would be interesting to

**Table 2.** Field notes with example ladders

| | | |
|---|---|---|
| Interview | i06 | i06 |
| Type | Face-to-face | Face-to-face |
| Stimuli | Shopping | Shopping |
| Ladder ID | 46 | 47 |
| Start time | 0:00 | |
| Rank | - | - |
| Feature | A: Guiding service in supermarket | A: Bulletins about special offers sent to your device |
| | C: Can go to unfamiliar supermarket | A: Being able to turn this off / Customized |
| | C: You could be more pro-active | C: More information about offers on special |
| | C: Better shopping experience | V: Save money |
| | C: Customers around you could help you | V: Greater shopping experience |
| | V: Independence | |
| | V: Empowerment | |

you?" questions, and recorded the responses labeled with a "C" (for consequence) prefix. We continued until the participant volunteered one or more basic values or goals, shown with a "V" prefix in table 2. We repeated this process for the second most interesting stimuli.

To wind up each interview, we asked the participant to evaluate the ideas that they had contributed. To avoid working memory overload, we first asked the participant to identify the three most interesting ideas in rank order. Then we collected a rank ordered list of additional items that they thought interesting.

### 4.3 Model and Aggregate Requirements

The interpretive analysis process involved four steps. First, we determined seven "themes," or concepts, that would cover all chains and we assigned each chain to one of these themes. Next, within each theme, we grouped participants' statements by attribute clusters, consequence clusters and value clusters, to highlight clusters, or subthemes, within each theme. Finally, for each theme, we drew a value map, a graphical representation that shows preferences and reasoning of the aggregated chains that it contains.

Two researchers independently examined the data to develop conceptual themes that would incorporate all of the chains; one developing ten and the other seven. They reached a consensus on seven themes by discussion.

Next the two researchers independently assigned each of the chains to one of the themes. They initially agreed for 86% of the chains, a high level of agreement, compared to similar studies [10, 14], and resolved differences through discussion, for six chains including them in two themes each and placing one chain in three themes. One chain was dropped from the analysis due to feasibility concerns.

Each participant expressed his/her preferences and reasoning using different language. Consequently, to interpret the data within themes, we clustered the chains by attributes, consequences and values, assigning a common label to each of similar attribute, consequence and value statements. The names followed the language used by participants to the extent appropriate. By assigning these labels it was going to be easier to get a bird's eye view of the themes and their features and to depict links among attributes, consequences and values in the themes. By sorting by the labels, this allowed us to see clusters within the themes.

We copied each ladder into a spreadsheet and labeled it according to the theme to which it was assigned. Then we grouped the ladders into clusters, stepwise, by attributes, consequences and values. Where ladders included more than one attribute each, we copied them into more than one ladder, where necessary, to cluster. Next, we arranged the codes theme-wise in a graphical representation, to form thematic maps, creating one map for each theme on partitioned sheets with areas for attributes (left), consequences (centre) and values (right).

With the aggregated models of user preferences in hand we proceeded to getting post-analysis feedback from the users. In the initial interviews we grounded user's individual preferences for mobile services. In this phase, we gathered feedback on what every user thinks about the features all participants contributed.

The feedback was gathered in short phone interviews. All participants were called with the objective to have a 10-minute feedback chat. Three participants could not be reached by telephone. In total, we successfully contacted 18 for the post-analysis feedback session. The three steps to elicit user feedback are further described below.

**Selecting most Important Themes.** The initial challenge was to find a representation that gets the user familiar with each theme and its associated features. We did not want to overwhelm the users with descriptions that were too verbose. To limit complexity, we wrote a theme description that included a refined name of the theme, followed by a summarized enumeration of the related features. We read out the theme summaries one by one and clarified the specifications if necessary. After each summary we asked the participant to indicate, on a scale between 1 (not important) and 10 (very important) how important the particular theme was to him/her.

**Selecting most Important Features.** We broke down each theme into written scenarios to summarize groups of features. The narratives aimed to introduce participants to features, of which they might never have heard. Each scenario started off with *"Imagine…"*, then named the features and explained them if necessary. The scenarios also stated what one could accomplish with a particular feature or features. These constitute the *consequences* that manifest themselves when having the feature available. For example, one scenario read, "*Imagine you had a rather bulky phone with, large, raised and well spaced-apart keys, with square-shaped number keys and differently shaped function keys, arrow keys instead of joysticks, more wheels and switches, and a dot on the 5. This would result in a more accurate input.*" Participants provided importance ratings for the scenarios in a manner similar to that above for the themes.

**Compile Rankings.** Using the data from the feedback interviews, we used the Likert ratings for themes and feature groups, normalized for the number of items rated, to compute rank order ratings for each theme and feature cluster.

### 4.4 Prioritize Importance of Requirements

We evaluated the results of requirements elicitation and analysis in two ways. Firstly, we aimed to clarify which of the ideas proposed by individual participants were most valued by others. Secondly, we sought to verify whether the system suggestions derived from interpretive analyses, which were carried out by the researchers, accurately represented users' needs. To evaluate the efficacy of the selection, ranking and rating rounds we here compare the outcomes of these consecutive rounds to each other.

During the original individual laddering interviews we had asked the participants to rank order their top three ideas. This provided us with a first round of data, a portion of which is depicted in the left column of Table 3. The second column presents data collected in follow-up telephone interviews after the creation of the thematic maps. It presents a tally of Likert-type rating scores for the ideas that lie within the three top themes. The top three themes, out of seven, *'navigation & routing'*, *'traffic & public transport assistant'* and *'shopping assistant'* accounted for 70% of all rating scores. These rating were derived during the feedback-gathering phase of the study.

The results show that the preferences most highly rated by individual participants in the original data collection, were also the most highly rated by all participants, following the intervening analysis. That the same themes where highly rated suggests some support for the aggregation and modeling of the analysts, i.e., that the preferences and reasoning were well captured and interpreted. The rest of the themes

**Table 3.** Theme level preferences comparison before/after data analysis

| Pre-analysis (standardized sum of inverted ranking scores for features per theme) | | Post-analysis (standardized sum of rating scores for features per theme) | |
|---|---|---|---|
| Navigation & Routing | 173 | Navigation & Routing | 161 |
| Traffic & Public Transport Assistant | 153 | Shopping Assistant | 155 |
| Shopping Assistant | 147 | Traffic & Public Transport Assistant | 154 |

followed this trend within the data. Note that comparisons of the values of the ratings across the two columns are not meaningful.

Table 4 presents more in-depth analysis of the results within the navigation & routing theme. Here there is more variation between pre- and post-analysis results, but we can see that the five top-ranking feature groups, out of twelve, are largely the same even though the order among them differs. These results also suggest that the analysis has well captured the meaning of the data provided by the participants. Here also, the actual numerical values are not comparable across columns.

**Table 4.** Feature group user preferences comparison before/after data analysis

| Pre-analysis ranking (sum of standardized, inverted ranking scores) | | Post-analysis feedback (sum of standardized Likert scores) | |
|---|---|---|---|
| City guide | 102 | Bookmark & manage routes | 30 |
| Show points of interest | 46 | City guide | 28 |
| Bookmark & manage routes | 36 | Underground reception | 25 |
| Underground reception | 9 | Locate peers | 25 |
| Locate peers | 8 | Single-handed control | 23 |

## 5 Theoretical Evaluation of the Method

The designed method was intended to address the objectives for an RE method, described in Section 2. To evaluate the design, we observed how the design addressed each of the objectives in its intended functionality and in its use in the demonstration. Here is how the designed method addressed each of the objectives developed in Section 2. This evaluation is summarized in Table 5.

Our objective with respect to information processing was to avoid excess demands on participant working and long-term memory that might ensue from elicitation and prioritization activities, without the benefit of visual aids. The use of stimuli developed from other participants enabled the pooling of long-term memory among participants to elicit more items. Individually interviewing participants on their two most preferred stimuli, using the laddering interviewing technique, minimized working memory demands by focusing on one or a small number of concepts at a time. The two-step rank ordering process for needs evaluation reduced the number of concepts considered at a time. In model evaluation, the talk-aloud scenarios chunked concepts into a smaller number of scenarios to create smaller working memory loads. Thus the need to limit reliance on blind participants' information processing abilities was supported by features in both of the requirements discovery and prioritization phases.

Our objective with respect to complexity was to elicit needs and preferences for functionality and features from participants, who may be unfamiliar with underlying processes and technology, without the use of visual materials. Selecting lead user members of the target population, i.e., members who are interested in exploring the use of new technologies, limited potential negative effects in elicitation resulting from lack of knowledge about unfamiliar technologies. In needs elicitation, the laddering technique broke down participant constructs to consider one element at a time. In model aggregation, structuring the expressed preferences as attribute/consequence/values chains provided a very generalized structure for expressing participant preferences that didn't depend on the structure of the participant constructs. In model evaluation, the use of aggregated rank ordering for preferences was a simple, general concept easily understood by participants. Thus the need to mitigate potential excess complexity for blind participants was supported by features in all four phases of the RE process.

Our objective with respect to communication was to capture data in a manner that was structurally very flexible and that would represent results of analysis in a manner that was useful and understandable for managers and designers. The collection of stimuli from participants initiated the requirements elicitation with ideas that are expressed in terms of the participants' knowledge. The laddering interviews allowed participants to use their own knowledge structures to present reasoning data. The thematic clustering preserved these participant structures, while the model evaluation technique overlaid this with a structure of priorities, for use by managers. The use of the thematic maps, which presented aggregated preferences in terms of attributes, consequences, and values, presented reasoning for the expressed preferences, for the benefit of a potential technical and managerial audience for the analysis, in a manner that accommodated the participants' own knowledge structures. Thus, the need to accommodate communication constraints for blind participants was supported by features in all four phases of the RE process.

Overall, the method, described here and demonstrated with New Zealand and German blind participants, explicitly addressed all of the objectives that we developed above for a blind user RE method. The demonstration showed that it could be successfully used to develop preferences for applications and features for the target population. Indeed, the concepts for new applications developed in the demonstration appear at face value to be an interesting set of applications and features for services well targeted to the blind population and which do not already exist. This supports assertions

**Table 5.** Evaluation of the method

| Process / Constraints | Information processing | Complexity | Communication |
|---|---|---|---|
| **Scoping** | | Lead-users as proxies for users | Stimuli collection for laddering |
| **Requirements Discovery** | Selecting attractive stimuli and laddering interviews limit demands on WM and LTM | Use of Laddering interviewing technique breaks down participant construct | Use of Laddering—participants use own constructs |
| **Model Aggregation** | | Use of attribute/feature-consequence-value/goal structure for data | Thematic clustering of requirements—participants' own constructs |
| **Requirements Prioritization** | Two step rank ordering best features<br>Talk-aloud scenarios of feature subthemes chunk items | Use aggregated rank order data of elicited requirements. | Provide numerical data for cross-comparison of features |

in the introduction of this paper of the importance of including members of this population in RE activities, rather than merely accommodating their use *ex post* of applications that are designed for the general sighted population.

## 6 Discussion and Conclusions

Since this is the first published RE method for blind users, there is not another method with which to compare its efficacy. Certainly we neither claim nor think that it represents a best possible method. We expect to pursue the development of RE methodology for use with the blind and we would hope that others would do so also. It should be noted that none of the specific techniques or activities in this process was invented specifically for this use.

All of the techniques used here have been used elsewhere with members of the general population. It is their combination to accomplish the objectives of this process and their adaptation for use with this target population that creates unique value. Nearly everything that we have done here, by way of individual techniques and methods, can be incorporated piecemeal into RE processes, where, in the judgment of the analysts, it would facilitate participation by disabled participants

The results of this method included ideas for new applications with features that would, if developed be very complex software and hardware projects. They were developed to level of specificity where they could, with additional business analysis, be the subject of project proposals. Obviously, the demonstration did not result in requirements specifications that could be used to move ahead to a project design phase. Extending our process to include all of the phases of requirements specifications would be valuable, although it might be argued that niche target population participation is most critical at the pre-project proposal stage, where ideas for new applications and features of particular importance to the population are incorporated into the initial project proposal.

In this study we identified an important RE problem, the need to adapt RE methods to accommodate blind users, so as to be able to effectively determine requirements for systems targeted to these users, and we identified three objectives for a method to accomplish this, aggressive participant recruitment, non-visual data gathering, and accommodation to working memory limitations among participants. We designed a method to accomplish these objectives and demonstrated its use to develop the requirements for mobile services tailored to blind users. Finally, we evaluated the proposed method in terms of how we were able meet the objectives.

We see that the proposed requirements engineering method holds promise for use in research and practice to develop a variety of applications, services, products, and accommodations for use by blind people. In addition, our success in using a design science approach to develop this new method suggests that the same approach might be used to design RE methods tailored to other hard-to-reach populations, such as Islamic women or the learning disabled.

## References


1. `http://www.who.int/mediacentre/news/releases/2003/pr73/en/`
2. Shinohara, K., Tenenberg, J.: A Blind Person's Interactions with Technology. Communications of the ACM 52, 58–66 (2009)
3. Davis, G.: Strategies for information requirements determination. IBM Systems Journal 21, 4–31 (1982)
4. Browne, G.J., Ramesh, V.: Improving information requirements determination: a cognitive perspective. Information & Management 39, 625–645 (2002)
5. Hevner, A.R., March, S.T., Park, J.: Design Research in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
6. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24, 45–78 (2008)
7. Bliss, I., Kujala, T., Hämäläinen, H.: Comparison of blind and sighted participants' performance in a letter recognition working memory task. Cognitive Brain Research 18, 273–278 (2004)
8. Sauerburger, D.: Orientation and mobility for deaf-blind people. American Rehabilitation 23, 9–14 (1997)
9. Peffers, K., Gengler, C., Tuunanen, T.: Extending Critical Success Factors Methodology to Facilitate Broadly Participative Information Systems Planning. Journal of Management Information Systems 20, 51–85 (2003)

10. Tuunanen, T., Peffers, K., Gengler, C., Hui, W., Virtanen, V.: Developing Feature Sets for Geographically Diverse External End Users: A Call for Value-based Preference Modeling. JITTA: Journal of Information Technology Theory & Application 8, 41–55 (2006)
11. von Hippel, E.: Lead Users: A Source of Novel Product Concepts. Management Science 32, 791–805 (1986)
12. Browne, G.J., Rogich, M.B.: An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques. Journal of Management Information Systems 17, 223–249 (2001)
13. Kelly, G.A.: The Psychology of Personal Constructs. W W Norton & Company, New York (1955)
14. Tuunanen, T., Peffers, K., Gengler, C.E.: Wide Audience Requirements Engineering (WARE): a Practical Method and Case Study. Helsinki School of Economics (2004)

# Assessing Project Effort in Requirements Engineering: A Report on Design Research in Progress

Frank Zickert and Roman Beck

Chair of Business Administration, esp. E-Finance and Services Science
Goethe University Frankfurt, Grüneburgplatz 1,
60323 Frankfurt, Germany
mail@frankzickert.de, rbeck@wiwi.uni-frankfurt.de

**Abstract.** In this paper we report on our design research in progress, where we have developed an artifact that assesses project effort resulting from requirements. Based on models used in the goal-oriented requirements engineering method KAOS, the artifact measures system size via function point analysis and analyzes system complexity via structural analysis. In addition, we provide theoretical explanations and empirically validate how size and structural complexity affect project effort. Overall effort depends on counted functions that must be transformed, since software development can be regarded as a transformation process where size matters. Structural complexity matters as well, since software development is also a complex problem, where effort spent depends on the structure of the problem. Insights from empirical evaluation in three software development projects are encouraging, wherefore we believe that the artifact appropriately assesses project effort. Furthermore, our artifact increases the utility of KAOS by providing additional information on project effort.

**Keywords:** Requirements Engineering, Project Effort, KAOS.

## 1 Introduction

Project goals, such as conformance to constraints in time and budget, are frequently used as indicators for measuring project success [1]. What is astonishing is the fact that although their importance is recognized in requirements engineering (RE) [2, 3], it is not well understood how they are affected by requirements. Consequently, assessment of project goal satisfaction is also not supported in RE, rather, its consideration is explicitly excluded [4].

We find exclusion of project goal satisfaction in RE unreasonable. There are sufficient indications to make consideration important. The main driver for concerns that project goals refer to, like time and cost, is project effort [5, 6], which is the work that has to be done in a project [7]. The amount of this work depends, for instance, on the size and complexity of the system under construction. Requirements affect both. There is, for instance, a difference in size between the requirements "the system should provide a basic calculator" and "the system should provide full support for a company's accounting". There is also a difference in complexity for "our batch job-based accounting system

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 490–505, 2010.

should now support real-time operation" and "it should provide more batch jobs". Although requirements are about the system under construction [8], requirements engineers need to be aware of the effort that results from the system they construct. It is not something nice to have. It is essential, because "engineering is not just about solving problems; it is about solving problems with economical use of resources, including money" [9, p. 15].

In this paper, our objective is to report on our design research in progress, where we have developed an artifact that incorporates into RE the assessment of project effort resulting from requirements for software development projects. To achieve that goal, we extend the goal-oriented RE method KAOS by assessment of system size via function point analysis (FPA) and assessment of its complexity via structural analysis. KAOS provides a semi-formal requirement notation language that supports requirement elicitation and allows reasoning about goal satisfaction [8]. We evaluate our artifact by applying it in software development projects conducted within the financial industry.

We also aim at shedding some light on theories explaining the effects requirements have on project effort. Firstly, for explaining the effect that size has, we use the theoretical perspective of software development as a collection of transformation activities [10]. Secondly, for explaining the effect of structural complexity, we utilize the perspective of software development as complex problem solving [11]. We assess the validity of these theories with observations made in our projects at hand. Using these theories in the context of RE we intend to support the reliability of our artifact. They support an understanding of why our artifact is able to gather information about project effort from requirements.

This research is expected to make contributions to professional requirements engineers, IS researchers in RE, and to design science. For professional requirements engineers, we provide our artifact. It is a Microsoft Visio-based RE tool for assessing project effort from models used in the goal-oriented RE method KAOS. By providing this additional information, it increases the utility of the KAOS method and thereby the return on investment for the requirements engineer when preparing these models.

For IS researchers in RE, we assess the validity of theories we derive from existing knowledge and put into the context of RE. These are moderately abstract theories about the effects that requirements have on project effort. As such they can lead to easily testable hypotheses [12]. Since their formulation is made explicit in our research (as requested by Venable [13]) it may provide a reasonable starting point for further theory testing.

For design science researchers, we provide an example of design research that does not solely aim for the creation of an artifact but that has the explicit intention of assessing the validity of theories that increase the artifact's reliability.

The remainder of this paper is structured as follows: In the second section we describe the process that we find most suited for serving our research goal. For presenting the main part of our paper (sections three and four), that is the iterative construction of our artifact and assessment of the theory validity, we adhere to the example given by Peffers et al. [14], as we use the stages of each iteration cycle as structure for communication. Finally, we provide our preliminary conclusions and an outlook on this research in progress in section five.

## 2 Research Perspective and Method

In our research, we adhere to the design science paradigm. More specifically, we apply design research that aims at creating and improving solutions for a specific class of problems [15, 16]. We aim at improvements in the yet unsupported assessment of project effort resulting from requirements. Moreover, we consider design research as most appropriate for our research because the design science goal of utility [17] complies with our goal of providing requirements engineers with additional information, without forcing them to spend work on another method.

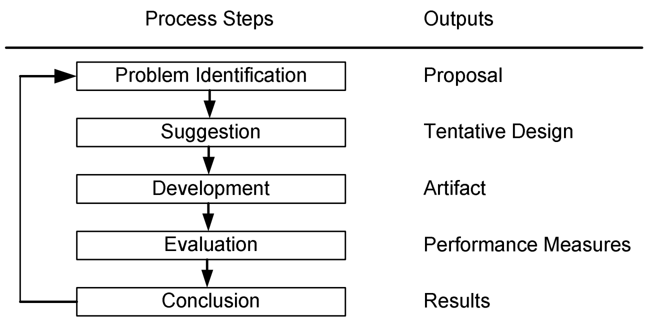



**Fig. 1.** Research method for design research as presented by Vaishnavi and Kuechler

We follow the design research method as presented by Vaishnavi and Kuechler [18] shown in Figure 1. However, we make adjustments that we consider appropriate. We add activities to the suggestion and evaluation steps. In the suggestion step, we additionally propose use of existing theories to explain how requirements affect project effort. In the evaluation step, we assess validity of the propositions with empirical observations we make. Together, theory proposition and validity assessment are intended to increase the reliability of our artifact by supporting a better understanding of theories underlying it.

We planned to build our artifact in an iterative way. Consequently, we pass through all the steps not only once, but repeat them within each cycle. We investigate a single one effect requirements have on project effort with each iteration. This is beneficial (1) for development, because we can selectively incorporate existing methods for refining our artifact; (2) for evaluation, because we can appropriately select projects we find useful for assessing value added by the refinement; and (3) for proposing and assessing the validity of theories that explain the rationale of the respective effect on project effort. Moreover, we expect insights from the evaluation step to be useful for identifying shortcomings of our artifact, which are then addressed in the next cycle to improve the artifact [17]. We have not planned a concrete number of iterations. A decision of whether an additional cycle is required will be based on the results of our empirical investigations [17]. Our stopping criteria are: sufficient confidence in appropriateness of project effort assessments and absence of observed effects that are relevant but have not yet been assessed by the artifact.

In the following sections, three and four, we explain how we applied the method in our design research. In the initial cycle, we address size, which from our perspective is the most obvious effect requirements have on project effort. This cycle has been completed. Insights from evaluation helped us to identify the problem that we address in the second cycle. It is the complexity that arises when the system under construction needs to be integrated into an already existing system. Currently, we are nearly finished with the evaluation step of this cycle, in which we investigate two projects in the financial industry. Thus, insights we gathered from the projects are preliminary.

# 3 First Cycle: Assessing Size

## 3.1 Problem Identification

In RE, most emphasis is put on functional concerns of the system, because building a system is the problem to be solved [4]. However, it is a problem that involves additional conditions, for instance financial constraints [9]. But consideration of such concerns is highly inconsistent in RE. While the importance of some concerns is repeatedly emphasized, like budget constraints, organizational policies [2], or development time [3], others do not mention them [19] or even exclude them from further consideration [4].

For demarcation of the different concerns, we use the classification provided by Glinz. We adopt a high-level differentiation into system, project, and process concerns. We refer to these classes as system-related, project-related, and development process-related. Moreover, we recognize Glinz' second-level distinction of system-related concerns into functions, attributes (performance), and qualities. However, we find that these concerns are not only system-related, but more precisely function-related. Apart from function-related concerns, a system also comprises other concerns such as architectural ones, which do not mention functions but still refer to the system under construction. We refer to these concerns as architecture-related. Figure 2 provides an overview of classes of concerns.

When different concerns are talked about, the unanimous position taken is that functions and function-related attributes are the concerns of interest [19, 20]. There is also some interest in architectural concerns [21, 22]. Together, these represent system-related concerns. In fact, requirements must be about the system under construction [8]. Thus, technically seen, non system-related concerns cannot be addressed by requirements.

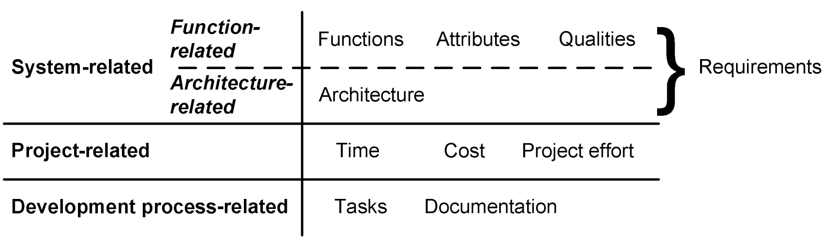



**Fig. 2.** Classes of concerns based on Glinz

However, we believe that not only system-related concerns are relevant when the system is built, but also other concerns. There are recognized conceptual differences [4] between the classes, which, to our knowledge, have not yet received much attention. Thus, they are not well understood in RE. And, sporadic reference raises the suspicion that they are not completely irrelevant [2, 3].

### 3.2 Suggestion

For answering the question of which types of concerns are relevant when building a system, we propose using the notion of software development projects as a collection of transformation activities [10]. We consider the waterfall model [23] as a blueprint of that notion. As shown in Figure 3, it involves the system's content (rectangles) and activities (arrows) that transform content from one state to another. For example, goals are transformed into requirements by refinement, requirements are transformed into specification by derivation, and specifications are transformed into code by implementation.

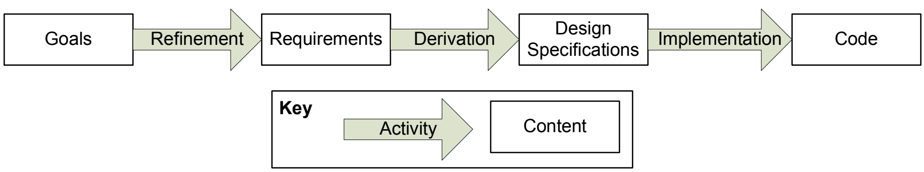



**Fig. 3.** Process model for software development as a collection of transformation activities

In fact, this is an abstract notion of software development. Which specific activities are done at all, whether they are done sequentially or concurrently, as well as other aspects, are all determined by the project model that is applied to a specific project. At this point, recognizing content and activities as different concepts with different interpretations is important. Content is what the system is about. These are in fact system-related concerns. Activities are tasks that have to be done in the project. They are development process-related concerns.

Project-related concerns do not directly refer to content or activities. Rather, they depend on the interaction of both concepts. These concerns, for instance, effort, time and cost, are measures of a specific content's transformation by the respective activity.

On the one hand, effort, time and cost surely depend on the content. With content size increasing, more work has to be spent for a transformation activity. This work in turn takes longer to be completed and staff needs to be paid for that working time. In fact, size was found to relate to development effort [5], time [6], and cost [24]. Requirements are mainly driven by functionality. They describe what the system does, not why or how it does it [20]. Measuring size from requirements is, hence, most appropriately based on functionality. Since FPA is a measure of functional size [25], we suggest using it for measuring size from requirements.

### 3.3 Development of the Artifact

Goal orientation is a major topic of interest in RE research [26] and it has been used in the context of requirements acquisition, relating requirements to organizational context, clarifying requirements, dealing with conflicts, or deriving design [27]. Our artifact is mainly based on KAOS, which is one of the most important goal-oriented RE methods [28]. KAOS comprises four models: (1) a goal model in which goals to be achieved by the system are described; (2) an object model in which objects involved in the system are described; (3) an agent model in which responsibilities are assigned to agents; and (4) an operation model in which input-output relationships among operationalizations of requirements and identified objects are described. Van Lamsweerde [8] provides an in-depth illustration of KAOS.

Our artifact supports the preparation of all four KAOS models. For the purpose of measuring size from requirements, the operation model is used. Here, operations (represented as ovals) refine requirements (parallelograms). Operations can be interpreted as a behavior of the system in a specific situation that has to be implemented by stakeholders (hexagons). Each situation is determined by events (arrowed rectangles) that cause the operation and entities (rectangles) that serve as information input. Similarly, operations may produce either entities, events, or both as outputs. Figure 4 provides an example of a small KAOS operation model.

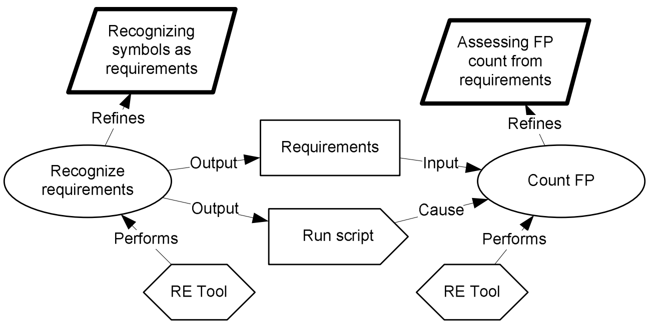



**Fig. 4.** KAOS operation model

Counting function points (FP) comprises two major function classes: data functions and transactional functions. Data functions in FPA are either internal logical files (ILF) or external interface files (EIF). Transactional functions represent external inputs (EI), external outputs (EO), and external inquiries (EQ). All function classes are identified primarily based on the software's logical design [29]. Figure 5 graphically illustrates the five major function classes while Table 1 summarizes the commonly used abbreviations.

Generally, an operation's perspective is applied for mapping structures used in KAOS onto function classes counted in FPA. That means that FPs are counted separately for each operation. From that operation's perspective, transactional functions refer to the operation's input/cause or output connections. Data functions refer to objects (entities or events) that are connected to the operation by the transactional functions.

**Table 1.** Abbreviations used in function point analyses

| Abbreviation and Name | Description |
|---|---|
| EI–External Input | Input that originates outside the system and updates an ILF |
| EO–External Output | Output created by the system and transmitted to an outside component |
| EQ–External Inquiry | Online Input that originates outside the system and results in intermediate system response |
| ILF–Internal Logical File | Logical group of data that resides within the system |
| EIF–External Interface File | Logical group of data that resides outside the system but is used by the system |

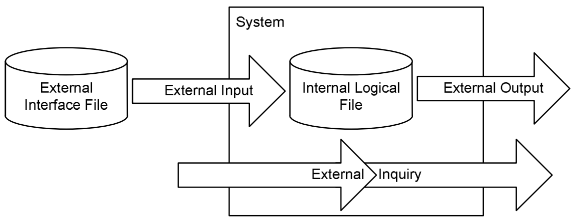



**Fig. 5.** Function classes used in function point analyses

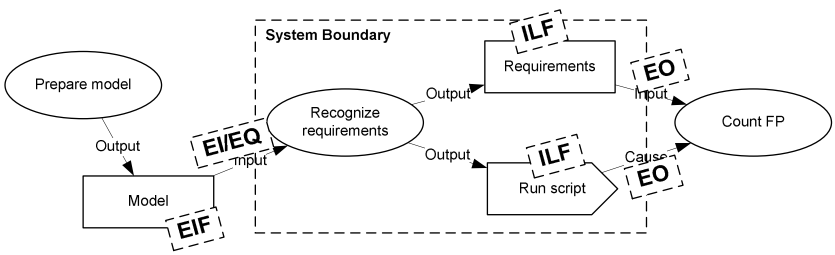



**Fig. 6.** Pattern identification in KAOS

An EO is an output connection from the operation. It connects to an object that is an ILF for the operation. Indeed, this respective output connection is not yet an EO unless the connected ILF serves as input/cause for another operation.

Similarly, an EI is an input connection to the operation. It connects to an object that is an EIF for the operation. Again, an additional condition for the input connection to be counted as EI is that it is an output of another operation.

Although it is advised to distinguish between EQ and EI, the structures that are counted are almost similar. The major difference is whether an ILF is updated (EI) or not (EQ). Since direct connections between operations are not allowed in KAOS, there is no equivalent to EQ and thus, they cannot be distinguished. However, since both EQ and EI are treated equally in FPA (use of the same weighting factors, [30]), we accept this missing differentiation.

The IFPUG manual [29] provides a detailed description of these structures counted in FPA. Figure 6 provides a graphical representation of all patterns as they are identified in KAOS by our artifact.

### 3.4 Artifact Evaluation and Assessment of the Suggestion

We chose a single case for first time evaluation of our artifact. Its main purpose was to assure that our artifact is applicable in practice and that information provided by it is not completely off in calculation. However, we did not want to rely on a single comparison of calculated and actual effort. For that reason, we chose to investigate a multi-stakeholder software development project that was carried out in a large financial institution. It offered us the opportunity to calculate and evaluate modularized components that different stakeholders were responsible for.

The project's purpose was extending an existing front-end system by a new interface for entering payment orders and connecting it with a recently built payment processing system. Due to the institution's organizational structure, this project involved a customer department, two internal information technology departments, one external vendor, and a process quality group. Each of these stakeholders stated different goals that we classified as function-related, architecture-related, project-related, or development process-related depending on the mentioned concerns.

We observed two situations in which alternative requirements were negotiated among stakeholders, since one stakeholder rejected requirements that another supported and vice versa. We concentrated evaluation on these two situations, because these were extensively analyzed by stakeholders in the project for argumentation and, consequently, the most detailed information could be gathered. Sources of information comprised discussions with stakeholders about their perceptions on satisfaction of stated goals and various documents that had been prepared within the project, such as decision documents, meeting minutes, and concepts. Data was both partly quantitative and partly qualitative.

We base evaluation of our artifact on its correct assessment of system-related and project-related goal satisfaction. Our artifact utilizes KAOS that is very suited for modeling content—that is system-related goals. Respective goal satisfaction can be determined by a comparison of content to be and content as is [8]. Moreover, by the mapping of KAOS models onto structures counted in FPA, our artifact also provides size in terms of FP counts. Since stated project-related goals concerned project effort and development time, we expect our measured size, if correct, to relate to effort [5] and time [6].

We also gathered information for assessing the validity of our proposition that software development can be regarded as a collection of transformation activities. In the analyzed situations, requirements were negotiated among stakeholders. For these negotiations we assume that all stakeholders supported the one alternative that better satisfied their goals [31]. If the transformation perspective is valid, we expect that in these negotiations about content, only system-related goals and project-related goals matter, because requirements cover system-related concerns that inherently relate to system-related goals and affect project-related goals. Development process-related goals, however, are expected to not matter, because satisfaction of these goals is determined by conformance to the tasks given by the project model [32]. Descriptions of tasks such as ‘refining goals into requirements’ or ‘deriving design specifications’ should remain unchanged regardless of any discussed content.

While we already assessed system-related and project-related goal satisfaction for artifact evaluation, we required in addition to measure development process-related goal satisfaction for assessing the proposition’s validity. For this, we directly used the results of reviews performed by the process quality group in the project.

### 3.5 Results

Artifact evaluation is based on the correct assessment of system-related and project-related goal satisfaction. For increasing the reliability of our data, we reduced all data to a qualitative appraisal of which of the discussed alternatives better satisfied each stated goal. We compared these relative assessments based on the data with respective assessments based on information provided by our artifact. We found consistent statements of which alternative better satisfied (1) function-related goals in 3 out of 3 cases, (2) architecture-related goals in 4 out of 4 cases, and (3) project-related goals in 4 out of 4 cases. We are thus confident that our artifact supports correct assessment of these types of goal satisfaction within given limits in precision due to reduction to a qualitative comparison of two alternatives. Moreover, our artifact supported assessment of satisfaction of another project-related goal, whose satisfaction could not be assessed within in the case.

The validity assessment of our proposition is based on the relevance of each type of goal for negotiation of system-related requirements. We utilized two types of information for this. Firstly, we evaluated whether a stakeholder’s stated preference regarding the alternatives discussed corresponded to the satisfaction of goals in this alternative. Secondly, we analyzed our data to ascertain whether the goal’s state of satisfaction had affected the stakeholder’s preference and thus, found to be relevant. We calculated the goal relevance ratio by dividing the number of goals found relevant by the total number of goals. As summarized in Table 2, with the exception of development process-related goals, we found all other types of goals highly relevant for stakeholders formulating their preferences.

The empirical observation that project-related goals are relevant, but that stakeholder preferences did not always correspond to higher satisfaction, sounds reasonable, since their satisfaction is affected by requirements, but these goals are not the main reason for carrying out a project. Hence, it is understandable that if a stakeholder is interested in both system-related and project-related goals, the stated preferences are rather oriented towards satisfaction of system-related goals. Altogether, our observations support the notion of software development as a collection of transformation activities.

**Table 2.** Results of the relevance of different goal types

| Type of goal | Number of goals | Corresponded to preference | Found relevant for preference | Relevance ratio |
|---|---|---|---|---|
| Function-related goals | 3 | 3 | 3 | 100% |
| Architecture-related goals | 4 | 4 | 4 | 100% |
| Project-related goals | 5 | 3 | 4 | 80% |
| Development process-related goals | 2 | 1 | 0 | 0% |

# 4 Second Cycle: Assessing the Complexity of System Integration

## 4.1 Problem Identification

The project in our first case aimed at re-engineering a self-contained part within an existing system. Most goals concerned deliveries of the newly developed part, for instance, how order entry had to be guided by the new interface. However, the new part needed to be integrated into the existing system using some interfaces. In fact, conflicts in the project were about which specific interfaces should be used. Since the discussed alternatives in both situations differed significantly in all classes of relevant goal satisfaction, including project effort, the effect that the existing system had on project effort seemed worthwhile to consider.

In RE, interfaces of the system under construction and its environment, which also includes an existing software system, receive recognition [10, 20]. In fact, properties of that environment are used in RE for progression towards the final design [33]. Moreover, since software integration is recognized as a source of inconsistencies [34, 35], the importance of interoperability has been emphasized [36].

However, most of this recognition from the field of RE refers again to system-related concerns. Consideration of interoperability issue effects on project-related concerns, including project effort, can rather be found related to other topics, such as integration of commercial-off-the-shelf (COTS) components [37] or metrics on software reuse [38]. But, although these techniques are useful for make-or-buy or reuse-or-redevelop decisions, they are not intended to support RE. Since in RE both concerns matter, system-related and project-related, there is a lack of integrated support for addressing project effort that is required by an existing system.

## 4.2 Suggestion

For a deeper investigation of the effects that an existing system has on the integration of new requirements we propose that software development be not only regarded as a transformation process, but also as a complex problem [39]. Complex problems are frequently represented as strings of input parameters to a function [40]. The output of that function is the performance. Solving the problem requires finding a configuration which results in the desired performance output. Desired performance is determined by a goal. In fact, if there are different goals, there is a performance function for each goal. Goals often depend on the same or at least partly overlapping parameters.

If there is an existing system, some parameters are already set in a way that lets them serve a purpose. This purpose is the goal that reasoned building the system in the first place.

When faced with the problem of new or changed requirements in the context of an existing system, there are two options for requirements engineers. The first is to rely on re-usage and adjustment of the existing system. It implies changing parameters that have already been set. The second is to newly develop an extension to the system leaving existing parameters untouched as much as possible. There are, however, some changes required to the set parameters since the result should be an (at least partly) integrated system.

Both strategies have advantages and disadvantages. Re-using existing parameters decreases the size of the problem, because some parts are already in place. There may be only a few changes necessary for solving the new problem, as stated by new requirements. However, changing parameters can affect the resulting performance regarding the system's current purpose. In fact, even small changes may significantly decrease performance [41]. Recovering the prior performance then requires other parameters to be changed as well [42]. These changes can obviously yet again affect performance of other goals or purposes, requiring more changes, and finally, it can end up in a cascade where significant parts of the existing system need to be changed.

On the contrary, deciding to instead rely on building extensions to the existing system by using new parameters limits the number of reused parameters for required interfaces. It thereby decreases the chance of cascade effects. However, on the one hand, this results in a larger system to be developed, including respective effects on project effort. On the other hand, the number of configurations that have to be considered for solving a problem exponentially increases with the number of parameters [11]. This again drives effort to be spent on searching for a configuration with the desired performance [43].

We suggest assessing the structure of software development problems by analyzing the extent of reliance on and changing of the existing system and also by assessing the extent of reliance on new development.

### 4.3 Development of the Artifact

The structure of a problem is determined by the composition of its components. These are properties of the existing system as well as new requirements. For analyzing the composition, the context of each component must be assessed. The context of a component is determined by two characteristics. Firstly, the component of interest itself matters. It is the distinction between properties and requirements that KAOS models already support.

Secondly, distinction is based upon other components that the component of interest is connected to. Requirements can be connected to other requirements or properties. Requirements that are not connected to properties are not of concern for issues on interoperability with the existing system. They refer to new development. Requirements that are connected to properties are glue code. The purpose of glue code requirements is the translation of any messages or files sent between new components and the existing system [36].

Similarly, properties can be connected to other properties or requirements. Properties connected to other properties are not concerned with any change at all. They describe the existing system that is expected to remain unchanged. They are persisting properties. Properties connected to requirements are interface properties. These are usually required components of the existing system that glue code can connect to [36]. They may be affected by changes. Again, we utilize the KAOS operation model for assessing connections. Operations that refine a requirement or a property produce outputs in the form of events and entities that are used by other operations as input. Through this, connected components can be identified.

We integrated context assessment into our artifact by an analysis script that identifies each component's context. In fact, components are usually connected to more than one other component. Thereby, they can have different contexts. A requirement, for instance, that is connected to another requirement and a property comprises both characteristics of new development and glue code. Combining the context analysis with the mapping of KAOS onto structures counted in FPA not only allows us to separate a component's different contexts but also allows assignment of the software's assessed size to the contexts. Figure 6 provides a representation of all four contexts that our artifact distinguishes.

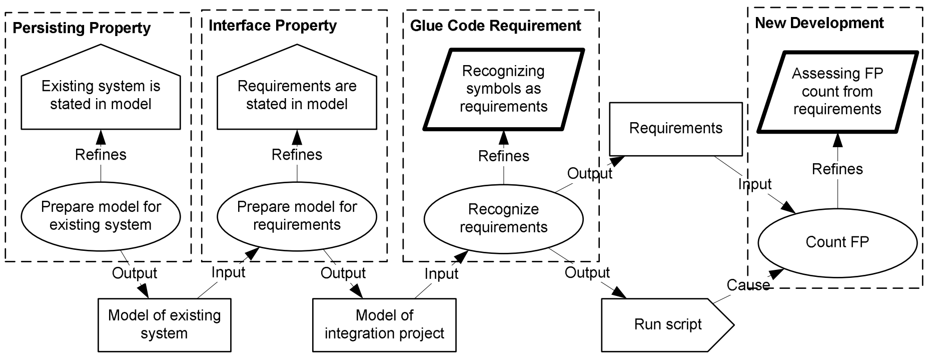



**Fig. 6.** Different contexts identified in KAOS

### 4.4 Artifact Evaluation and Assessment of the Suggestion

We currently are investigating two software enhancement projects where new functions are being integrated into an existing system. For the complete RE process (e.g. elicitation of new requirements, change of requirements, refinement of requirements) we assess size and context for all involved components. Data sources comprise first hand information like concepts, meeting minutes, software documentation, and discussions with other experts from development, quality assurance, or project management. We do not hand over KAOS models or any information produced by the artifact to the requirements engineer, because we intend to gather independent effort estimations from the requirements engineer.

In the first project, messages sent by new hardware added to a client terminal have to be integrated into and evaluated by the server system that is used for monitoring the terminals. The major task here is adaptation of the new message specificities to fit into the terminals' existing messaging mechanisms.

In the second project, the same monitoring system is extended by a function that allows selective deactivation of terminals. Since terminals are programmed for the highest possible availability, they autonomously re-activate and connect to different server systems. The challenge is a purposeful adjustment of the routine that accounts for activation and connection. While deactivation must be assured once a terminal has been disabled, reliability of the activation routine in default cases is just as critical.

In this cycle, we base evaluation of our artifact on correct assessment of project effort. We believe that the respective preconditions have been met. On the one hand, models are not prepared for parts but for the complete system under construction. On the other hand, projects at hand are integration intensive and thus, fit well to what our artifact is supposed to measure. For parts newly developed, we directly rely on calculated effort based on size measured by FPA. For both integration parts, glue code and integration properties, we utilize the respective parts of COCOTS [37] that serves as an effort estimation tool for COTS integration. It has to be noticed that integration properties are not supposed to change and thus, not supposed to result in effort. However, there is an increased risk of changes that may become reality if glue code requirements turn out to have not correctly accounted for all details.

Moreover, we not only assess the accuracy of measures based on final effort, but we also benchmark them with expert judgments that are made at the same time as effort is calculated using our artifact. This is intended to control for unexpected influences within the project that are out of the scope of our artifact.

For assessing the validity of our proposition that software development is a complex problem, we explicitly concentrate on gathering rich information on what the effects of re-usage are for building the system. We expect the building of the system to be accelerated when using existing components, as long as changes of integration properties do not affect the current purpose of the existing system. If they do, we expect negative effects that consume gained acceleration and that, from a certain degree of required changes onward, software development is decelerated.

### 4.5 Preliminary Results

Artifact evaluation in our cases is twofold. On the one hand, it is based upon comparison of effort estimations based on information provided by our artifact with estimates given by the professional requirements engineer. Until now, all estimated efforts using the artifact are within comparable ranges of expert judgments. On the other hand, once final data is available, we will compare these estimates with the actual effort.

Validity assessment of our proposition is mainly based on purposeful observations. Up to now, we have made two interesting observations in this regard. Firstly, existing components supported problem solving. Existing components were willingly reused for satisfying goals, because they were known to work fine. In fact, problem solving took place as a search for existing components that system-related goal satisfaction could best be based upon.

Secondly, why and how some components specifically worked in detail was not always known. If these components needed to serve as integration properties and subsequently required even marginal adjustments, there was reluctance to using them. Requirements that were completely fine apart from the effect they had on integration properties were given up. In fact, in a situation where they were not given up right away but were further refined a few more steps, it turned out that a change of the integration properties would have resulted in a whole cascade of changes in the system. We find these observations supporting our proposition.

## 5 Preliminary Conclusion and Outlook

In this design research in progress, our goal is the development of an artifact that supports assessment of project effort resulting from requirements.

Our first contribution is our artifact. It is a modeling tool based on the KAOS method. It currently gathers information on project effort from assessment of software size and structural complexity arising from contextual composition of properties and requirements. Results already gained are promising in that project effort can be correctly assessed by our artifact. In our first case, we found that all statements concerning which alternative better satisfied a project-related goal based on measured size were consistent with the data. In our two cases in the second cycle, effort estimations using our artifact are already comparable to expert judgments. We are thus confident that our artifact increases the utility of KAOS models because it provides a basis for effort estimations that otherwise would need to be conducted separately.

Our second contribution is a validity assessment of theories that explain effects that requirements have on project effort. While they are general theories put into the context of RE, the initial assessment here may provide a reasonable starting point for theory testing approaches. We explicitly invite any such attempt.

Our third contribution is this instance of design research that explicitly aims at assessing the validity of theories that increase the artifact's reliability. Our marginally extended approach on design research has proven workable for us in the first cycle.

Moreover, we find our results so far to be an inducement for looking beyond system-related concerns in RE. Other concerns, for example project-related ones, represent an under-researched niche that might offer room for valuable insights. While we initially shed some light on the high potential that well known theories offer on understanding project-related concerns, there is much more potential to be exploited. For instance, we find ourselves having much interest in the different classes of concerns relevance. We would highly appreciate a requirements acceptance model that explains in detail why stakeholders accept or reject requirements.

A broader scope of what is considered in RE, such as the one we have in this paper, can show ways for requirements engineers to better control their influence on the political ecology in which requirements are negotiated [44]. It is they who finally determine which goals are addressed by requirements that are put up for discussion. This would address a major challenge in RE, which is not to overcome resistance to requirements but to avoid it [45].

A major limitation of this research in progress is its current empirical foundation. Until now, we were able to investigate three software development projects. The results indicate the usefulness of our artifact for a broader audience, but this is not yet proven. Further steps in this research will address this limitation. Besides finishing the second cycle, we are currently planning a third cycle where we intend to hand over our artifact to a pilot user. Central to that cycle clearly is to increase the utility of the artifact so that we can increase the number of users in a fourth cycle.

## References

1. Capgemini Group: Studie IT-Trends 2006 – Unterschiedliche Signale (2006), `http://www.de.capgemini.com/m/de/tl/IT-Trends_2006.pdf` (Accessed 2009-08-01 )
2. Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. John Wiley and Sons, New York (1998)

3. Gross, D., Yu, E.: From Non-Functional Requirements to Design through Patterns. Requirements Engineering 6, 18–36 (2001)
4. Glinz, M.: On Non-Functional Requirements. In: Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE 2007), pp. 21–26 (2007)
5. Albrecht, A.: Measuring application development productivity. In: Proceedings of IBM Applications Development Symposium, pp. 83–92 (1979)
6. Cheung, Y., Willis, R., Milne, B.: Software Benchmarks using Function Point Analysis. Benchmarking: An International Journal 6, 269–279 (1999)
7. Abdel-Hamid, T., Madnick, S.: Software Project Dynamics. Prentice-Hall, Englewood Cliffs (1991)
8. van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour. In: Invited Paper for RE 2001 - 5th IEEE International Symposium on Requirements Engineering, Toronto, pp. 249–263 (2001)
9. Shaw, M.: Prospect for an Engineering Discipline of Software. IEEE Software 7, 15–24 (1990)
10. Davis, A.M.: Software Requirements. Prenctice-Hall, Englewood Cliffs (1993)
11. Newell, A., Simon, H.A.: Computer science as empirical enquiry: Symbols and search. Communications of the ACM 19, 113–126 (1976)
12. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30, 611–642 (2006)
13. Venable, J.: The Role of Theory and Theorising in Design Science Research. In: DESRIST 2006, Claremont, CA, February 24-25 (2006)
14. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 8, 45–77 (2008)
15. 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)
16. Winter, R.: Design Science Research in Europe. European Journal of Information Systems 17, 470–475 (2008)
17. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
18. Vaishnavi, V., Kuechler, W.: Design Research in Information Systems (2005), `http://desrist.org/design-research-in-information-systems`
19. Mylopoulos, J., Chung, L., Nixon, B.: Representing and Using Non-Functional Requirements: A Process-Oriented Approach. IEEE Transactions on Software Engineering 18, 483–497 (1992)
20. Jackson, M.: Software Requirements and Specifications; a lexicon of practice, principles and prejudices. Addison-Wesley, New York (1995)
21. Chung, L., Gross, D., Yu, E.: Architectural Design to Meet Stakeholder Requirements. In: Donohue, P. (ed.) Software architecture. Kluwer Academic, San Antonio (1999)
22. van Lamsweerde, A.: From System Goals to Software Architecture. In: Bernardo, M., Inverardi, P. (eds.) SFM 2003. LNCS, vol. 2804, pp. 25–43. Springer, Heidelberg (2003)
23. Royce, W.W.: Managing the Development of Large Software Systems. In: Proceedings of IEEE Wescon, pp. 1–9 (1970)
24. Boehm, B.: Software Engineering Economics. IEEE Transactions on Software Engineering 10, 4–21 (1984)
25. Fetcke, T.: A Generalized Structure for Function Point Analysis. In: International Workshop on Software Measurement, IWSM 1999 (1999)

26. van Lamsweerde, A.: Requirements Engineering in the Year 2000: A Research Perspective. In: Proceedings of the 22nd International Conference on Software Engineering, Limerick (2000)
27. Yu, E., Mylopoulos, J.: Why Goal-oriented Requirements Engineering. In: Proceedings of the 4th International Workshop on Requirements Engineering: Foundations of Software Quality (1998)
28. van Lamsweerde, A.: Goal-Oriented Requirements Enginering: A Roundtrip from Research to Practice. In: 12th IEEE International Requirements Engineering Confererence (RE 2004), pp. 4–7 (2004)
29. IFPUG: Function Point Counting Practices Manual Release 4.1. International Function Point Users Group, Westerville (1999)
30. Kemerer, C.F.: Reliability of Function Points Measurement. Communications of the ACM 36, 85–97 (1993)
31. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
32. Krishnan, M.S., Kriebel, C.H., Kekre, S., Mukhopadhyay, T.: An Empirical Analysis of Productivity and Quality in Software Products. Management Science 46, 745–759 (2000)
33. Seater, R., Jackson, D., Gheyi, R.: Requirement progression in problem frames: deriving specifications from requirements. Requirements Engineering 12, 77–102 (2007)
34. Robinson, W.: Integrating Multiple Specifications Using Domain Goals. In: Proceedings 5th International Workshop on Software Specification and Design, Pittsburgh, pp. 219–226 (1989)
35. Yakimovich, D., Bieman, J., Basili, V.: Software Architecture Classification for Estimating The Cost Of COTS Integration. In: Proceedings from the 21st International Conference on Software Engineering, pp. 296–302 (1999)
36. Lauesen, S.: COTS tenders and integration requirements. Requirements Engineering 11, 111–122 (2006)
37. Abts, C., Boehm, B., Bailey Clark, B.: COCOTS: a COTS software integration cost model. In: Proceedings ESCOM-SCOPE 2000 Conference (2000)
38. Frakes, W., Terry, C.: Software Reuse: Metrics and Models. ACM Computing Surveys 28 (1996)
39. Benbya, H., McKelvey, B.: Toward a complexity theory of information systems development. Information Technology & People 19, 12–34 (2006)
40. Kauffman, S.A.: At Home in the Universe. Oxford University Press, New York (1995)
41. Levinthal, D.A., Warglien, M.: Landscape Design: Designing for Local Action in Complex Worlds. Organization Science 10, 342–357 (1999)
42. Kauffman, S.A.: The Origins of Order. Oxford University Press, New York (1993)
43. Campbell, D.J.: Task Complexity: A Review and Analysis. The Academy of Management Review 13, 40–52 (1988)
44. Bergman, M., King, J.L., Lyytinen, K.: Large Scale Requirements Analysis Revisited: The need for Understanding the Political Ecology of Requirements Engineering. Requirements Engineering 7, 152–171 (2002)
45. Markus, M.L.: Power, politics, and MIS Implementation. Communications of the ACM 26, 430–444 (1983)

# Systematic Development of Business-Driven Requirements – Using Next-Generation EIS Design as an Example

Jörg H. Mayer and Frederik Marx

University of St. Gallen, Institute of Information Management,
Müller-Friedberg-Strasse 8, 9000 St. Gallen, Switzerland
{joerg.mayer,frederik.marx}@unisg.ch

**Abstract.** Despite many improvements to IT support for executives, they still complain that executive information systems (EIS) bear little relevance to their management task and fail even more to accommodate their working style. This indicates that business issues should more strongly drive requirements for next-generation EIS. The article contributes to such an EIS design by systematically developing requirements criteria that are more business driven than the state-of-the-art. To do so, requirements lists of EIS, structural models of user satisfaction and technology acceptance are evaluated with criteria derived from the requirements engineering discipline. The findings show a dual gap: as the rigor of the models increases, they become less relevant for practice. In comparison, the requirements lists demonstrate relevance, but do not evidence strong rigor. To bridge this gap, this article applies the principle of economic efficiency to balance scientific rigor with relevance for practice. A case demonstrates a first implementation and helps to evaluate the results of this article by using the same criteria as for the state-of-the-art reflection. The findings should lead to better next-generation EIS design and should also be applicable to IS in general.

**Keywords:** requirements engineering, next-generation executive information systems (EIS), principle of economic efficiency.

## 1 Introduction

Managing companies, especially large, international ones, is not feasible without IT support. It therefore seems surprising that *IT support for executives* is far less common than at the working level [1, 2]. Although many companies have executive information systems (EIS) in place meant to help top management, acceptance for these is often missing.

The literature provides extensive definitions of EIS. Four characteristics are important: First, their overall aim is "… to help an organization carefully monitor its current status, its progress toward achieving its *goals*" [3]. Second, they should enable "… *nontechnical senior executives* to navigate through strategic information culled from several company databases" [4]. Third, EIS are *computerized systems* with access to internal and external information that is relevant to executives' decision making [5]. Young, Watson [6] specify that as "direct and hands-on".

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 506–521, 2010.

Many executives complain that EIS still bear little relevance to their management task (functional requirements) and fail even more to accommodate their working style (*design requirements*) [7, 8]. This article focuses on the latter issue.

Managing a company requires comprehensive content, but executives face constraints in terms of time [9]. As a result, they have to set priorities. *Next-generation EIS* must therefore synthesize and present information in a condensed format [10]. Furthermore, the technology itself is not of concern to executives; instead, easy-to-use system handling largely determines their acceptance [1]. Next-generation EIS should thus focus on the business side of information systems (IS) while still bearing IT capabilities in mind.

In terms of timing, the present moment seems fortuitous: today's executives grew up with IS and should have a positive attitude toward IT [1]. Furthermore, technical progress has been made in the domain of corporate business intelligence. In particular, better front-end interfaces should make system handling much easier to use. On the research side, beside others, the homo economicus theory has given way to a more social perspective on the individual that accommodates diverse types of IS users and placing greater emphasis on socio-technical alignment [11].

To give next-generation EIS design a starting point, a new examination of *requirements* should be helpful. Requirements analysis for EIS is part of IS research which is mainly characterized by two paradigms. Behavioral research aims to develop and verify descriptive theories that explain how individuals and organizations use such systems. *Design research,* in turn, focuses on the development of innovative, generic solutions for practical problems, and thus on utility [12, 13].

A misalignment often exists within EIS requirements examination: On the one hand, behavioral research contains scientifically sound structural approaches such as the User Satisfaction Model or the Technology Acceptance Model. But, they are often not applicable in practice [14]. On the other hand, requirements lists driven by design research focus on relevance, but they are not particularly rigorous. The objective of this article is therefore to contribute to better next-generation EIS design by developing a *catalog of requirements criteria*.

According to Hevner et al. [12] and March, Smith [13], the outcomes of a construction process under the design research paradigm can be classified as constructs, models, methods, and instantiations. The catalog to be developed can be categorized as a *model.* It aims at balancing the needs of practitioners and those of researchers by developing applicable requirements without sacrificing scientific rigor. While the focus is on EIS design, the findings should also aid to better IS success in general.

To develop artifacts, several models for the construction process have been proposed [12, 13, 15]. The process described by March, Smith [13], which specifies ‘build’ and ‘evaluate’ activities, is predominant in the literature [12]. This article focuses on the *build* part using the current literature. As a result, it does *not* include a substantial evaluation process or the subsequent design of next-generation EIS itself. These areas should be the subject of further research.

Chapter 2 gives an overview of requirements engineering to derive a schema for properly identifying requirements. Chapter 3 describes the IS user satisfaction and technology acceptance research and evaluates several requirements lists and structural models in terms of that schema. The systematical development of a more business-driven approach is the subject of Chapter 4. Chapter 5 shows a first implementation and the evaluation of this new model of more business-driven requirements criteria. Finally, Chapter 6 concludes the article with an outlook and a proposal for further research.

## 2 Requirements Engineering

*Requirements* can be defined as prerequisites, conditions or capabilities needed by users (individuals or systems) to solve a problem or achieve an objective [16, 17]. Requirements thus specify desired objectives. In computer science, they describe functions and features of IS.

The discipline of *requirements engineering* (RE) aims at increasing the quality of system development by providing systematic procedures for collecting, structuring, and documenting requirements. These procedures ensure that the requirements are distinct and collectively exhaustive, preventing faulty IS design. Therefore, RE must incorporate the relevant stakeholders and ensure their commitment regarding the final requirements [18]. From formal perspective, RE should help to align the IS design costs and timeline.

The importance of RE in IS design is documented in several surveys. For example, the Standish Group names insufficient RE as a main reason why projects fail [19].

RE processes consist of three stages, which are shortly described in the following [16]. Combined with Hevner et al.'s [12] seven guidelines for design science, these meta requirements make it possible to evaluate diverse requirements lists and models of IS success (see table 1).

The first phase, *requirements identification,* involves defining the scope of the IS and demarcating the system from its environment. The available sources must also be determined. The focus should be on the intended user group and other relevant stakeholders, but the literature and comparable IS should be taken into account as well. Finally, the requirements themselves are collected by analyzing the identified sources using multiple methods (e.g. creativity techniques, literature analysis or empirical methods). The focus of this first RE phase is set on *completeness*.

In the second phase, *requirement analysis and specification*, these unstructured requirements are classified first [18]. Relevant stakeholders help to eliminate overlapping and competing requirements [17], and those remaining have to be prioritized. Then the requirements are put into a standard form. At a minimum, this standard form governs predefined attributes of the requirements; they may also cover the complete structure of the requirements catalog [17, 18]. Meta languages and models often have an advantage here due to the fact that they are more compact and precise. The focus of the second phase is the *distinctiveness* of each requirement.

During the third phase, *requirements validation,* a decision is made which requirements to use in subsequent design activities (build, realize, and test). First, each requirement is reviewed for *scientific rigor* (see Hevners et al. guideline 4: research contribution, [12]) e. g., validity, reliability, and generalizability. Second, the stakeholders must reach a consensus about the IS requirements and whether they effectively represent their expectations [16]. In other words, this phase is concerned with the criteria of *relevance* as well (see Hevners et al. guideline 2: problem relevance, [12]).

In conjunction with RE, *requirements management* (RM) tracks how requirements changes during the course of the project. *Traceability* and *transparency* should therefore be an additional meta requirement. The *handling* of the documentation should guarantee that the requirements are continuously available and provide selective access to the underlying information. Furthermore, RM administer the activities constituting the RE process [16, 19], so, a final meta requirement is an estimate of the *effort* needed in terms of cost and time.

## 3 State-of-the-Art Review

Determining prerequisites for successful EIS is the topic of two different schools of research: the *IS user satisfaction*[1] and *technology acceptance* discipline [20]. Both schools work with approaches that involve simply generating lists of criteria and more complex structural models. Chapters 3.1-3.4 provide an overview in terms of approaches' scope, the principle for structuring the requirements, and available EIS criteria. In Chapter 3.5 they are finally evaluated according to the criteria outlined in Chapter 2 and their ability to support next-generation EIS design requirements.

### 3.1 Requirements Lists of EIS

Focusing on IS success, several researchers have been working on EIS requirements in ways that do not require complex research models (see Chapter 3.2.-3.4). Table 1 (see next page) provides an overview of selected contributions by the two research schools to show the range of the research methods.

These list approaches are dominated by one principle: potential requirements are collected based on literature research and authors' own experience. Surprisingly, most of the approaches do not make use of an overall structuring principle or second-level structuring dimensions. So, the EIS requirements selected vary in terms of their number and level of abstraction. E.g., Young, Watson [6] and Poon, Wagner [21] provide hands-on variables for EIS design, whereas the variables used by Vandenbosch [22] and Jiang et al. [23] remain at a high level and are not directly applicable.

Most of these studies do not specify why certain requirements or dimensions are included. An exception is Stein [24], in which EIS requirements are derived top-down using the IS success factor method.

The approaches themselves are as diverse as the requirements for EIS they provide. The desire to be relevant for practice dominates the need for scientific rigor. Without methodological structuring, the exhaustiveness of these approaches is in doubt.

### 3.2 User Satisfaction Models

A more structured, better grounded approach is provided by the *D&M-IS success model* [26, 27]. Due to its overall IS scope, it is applicable for EIS as well, and has been already used for this purpose.

Based on extensive literature research[2], the model systematically incorporates different dimensions of relevant requirements for EIS. An updated model based on empirical testing and iteration by other authors evaluates IS success using *information quality, system quality,* and *service quality* as structuring dimensions (for extensions

[1] Earlier named just as IS success, referencing the D&M-IS success model, see Chapter 3.2.
[2] The sources reviewed by DeLoan und McLean identified IS success with IS user surveys.

[28, 29]). These characteristics affect IS *use*, *intention to use,* and *user satisfaction*. Furthermore, using the system produces certain *net benefits.* They influence user satisfaction and further use of the IS, completing a closed loop.

**Table 1.** Overview of requirements lists

| | Scope | Structuring principle | Available EIS determinates (input variables) |
|---|---|---|---|
| EIS user satisfaction | Stein [24] – action research with two cases: determination of EIS success factors using five-step IS success factors method | Top-down derivation of EIS success factors – individual list | Task-oriented requirements , based on<br>• business goals<br>• business strategy<br>• leadership strategies<br>• information success factors<br>• No design principles available |
| | Vandenbosch [22] – field study: antecedents affecting information retrieval behavior of executives for better EIS design | Two-level approach to structuring requirements | • Individual differences (tolerance for ambiguity, locus of control, degree of innovativeness)<br>• System characteristics (differentiation, integration, flexibility)<br>• Organizational context (social influences, perceived environmental uncertainty, job characteristics) |
| | Poon, Wagner [21] – action research with six cases: identifying championship, availability of resources and link to organization as meta success factors for IS design for executives | Single-level approach – list of 10 critical success factors | • Committed and informed executive sponsor<br>• Operating sponsor<br>• Appropriate IS staff<br>• Appropriate technology<br>• Management of data<br>• Clear link to business objectives<br>• Management of org.l resistance<br>• Management of system evolution and spread<br>• Evolutionary development methodology<br>• Carefully defined information and system requirements |
| EIS technology acceptance | Young, Watson [6] – survey: examination of the determinates of EIS acceptance | Two-level approach to structuring requirements | • EIS characteristics (ease of use and number of features)<br>• EIS support staff characteristics (staff-to-user ratio, face-to-face time with users, proximity of staff to users, reporting relationship, and staff qualifications) |
| | Jiang et al. [23] – survey: link between reasons for resistance and IS types | Single-level approach | • No direct determinates<br>• Strategies to promote acceptance based on people-oriented and system-oriented reasons; reasons backed by interaction theories |
| | Hung [25] – experiment: effects of expertise on executive support systems | Single-level approach | • Computer self-efficacy<br>• Expertise<br>• Task type<br>• System functionality |

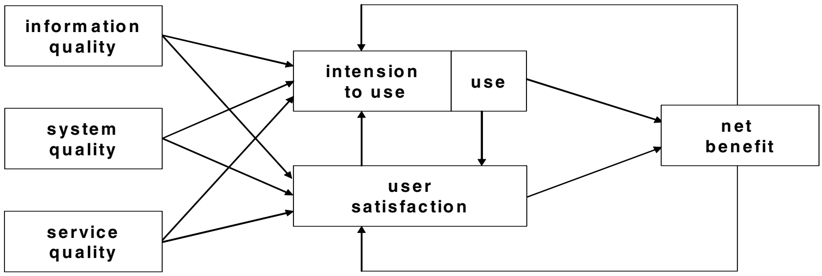



**Fig. 1.** The updated D&M IS success model [27]

In cases of the model's application (e.g. [30]), the dimensions of information, system, and service quality proved to be too abstract to use in practice. As a result, a consensus arose that specification are needed. So, information quality was specified in terms of information timeliness, accuracy, completeness, or comprehensibility, while system quality was specified in terms of reliability or response time. A number of external variables for measuring IS success have been documented accordingly [27].

But for EIS the cases are limited. One evaluates the D&M IS success model in the context of implementation, separating user computer experience, EIS team business skills, EIS team communication skills, user participation, user involvement, and user attitude towards the EIS as external variables. Another case [31] specifies net benefits in terms of individual and organizational impact. Neither article, however, is suitable to defining next-generation EIS characteristics itself.

Despite this criticism, the D&M-IS success model became the dominant IS evaluation framework in research, possibly due to its understandability and simplicity. But for practical purposes the external variables need to be specified. In general, it is doubtful whether practitioners will always understand structural models. But the method of using the literature to identify various IS categories and then testing them empirically attracted many followers.

### 3.3 Technology Acceptance Models

In terms of recognition, the Technology Acceptance Model (TAM) [32, 33] is on par with the User Satisfaction Model [34]. TAM is based on the Theory of Reasoned Action. The objective was to develop a theory to explain *user behavior* while bearing the variability of IS technology in mind.

Within TAM, IS use is determined by the system's perceived usefulness and perceived ease of use (see Fig. 2). Both are behavioral beliefs expressing individuals' different personal attitudes. *Perceived usefulness* is defined as the perceived probability that using a specific IS will improve the performance of work. *Perceived ease of use* is the extent to which prospective users think that using the system requires little effort. An individual's actual *use* of the systems in the work context is influenced by his or her *intention to use* [35].

Perceived usefulness and perceived ease of use are determined by external factors, which were not specified in the original literature. But in TAM2 und TAM3 external variables are added influencing 'perceived usefulness' and 'perceived ease of use'

[36]. They include *subjective norms* such as image or job relevance, *anchors* such as computer self-efficacy, and *other adjustments* such as perceived enjoyment or objective usability.

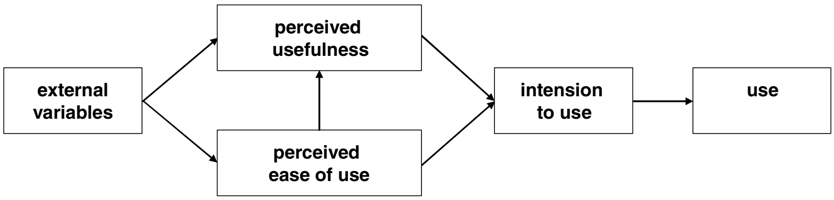



**Fig. 2.** Technology acceptance model [35]

TAM has been applied and evaluated several times (e. g. [1, 33]). Their extensions specified external variables and improve the model's applicability to IS design. However, the insights they provide have more to do with user attitudes than concrete requirements for a to-be EIS design.

### 3.4 Integrated Model

TAM is often criticized as providing bad advice on IS design [33]. Wixom, Todd [20] argue for merging TAM with findings from user satisfaction research, which should supply the missing IS characteristics.

The resulting Integrated Model structures the external variables into the two categories *information quality* and *system quality* of 4 to 5 variables each (see Fig. 3). These specific external variables are derived from the literature: four articles were compared to identify the most cited characteristics.

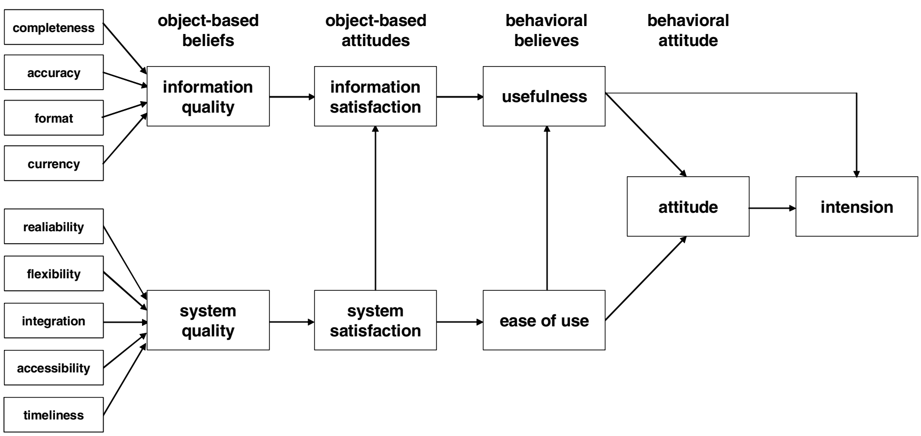



**Fig. 3.** Integrated Model of User Satisfaction and Technology Acceptance [20]

The Integrated Model backs the concepts ‘usefulness’ and ‘ease of use’ with external variables. However, it is important to note that the selection of these characteristics appears somewhat random – not following the structure of an overall design principle. Instead, these antecedents were selected based on their widespread use, representativeness, and relevance to the IT content to be explored. Furthermore, the link between these variables and the behavioral beliefs of usefulness and ease of use is achieved via the information satisfaction and system satisfaction – two additional, interacting constructs. The cause-and-effect chain between these variables and the user's attitude is therefore modeled only indirectly, making the direct impact of interventions at that variable level questionable as well.

### 3.5 Gap analysis

As previously stated, design research focuses on accomplishing utility. This section evaluates the approaches presented in Chapters 3.1-3.4 according to the RE criteria (Chapter 2). The results are summarized in **Fig. 4**, using a *5-step rating scale*.

Requirements lists cover just a few variables, deduced from the literature – based solely on the authors’ experience. As the *completeness* of this method of identifying requirements is always questionable, a negative assessment is justified. The User Acceptance Model provides a more structured approach using predefined dimensions, which the TAM model lacks [1, 33]. As a result, the User Satisfaction Model is rated as ‘good’. Because TAM focuses more on attitudes than requirements, its rating is slightly more negative. In contrast, the Integrated Model, which incorporates variables within the TAM framework, received a positive rating.

In terms of *distinctiveness,* no significant differences in the requirements lists are evident. As all such approaches lack a method for structuring requirements, they are rated ‘bad’. The User Satisfaction Model and the Integrated Model use predefined dimensions that ensure distinct variables, merely self-explanatory. As a result, these models are rated ‘somewhat’. The TAM lacks these features and therefore earns a ‘bad’. The attitudes and beliefs they employ as determinates of acceptance, e. g. ‘perceived fun/enjoyment’ [1], are not easy for practitioners to distinguish.

More interesting are the differences regarding (scientific) *rigor*. The requirements lists drawn on some literature research and criteria are selected based on the authors’ experience. That leads to a rating of ‘bad’. The User Satisfaction Model is empirically sound, so its scientific rigor seems to be ensured. However, the fact that different sets of external variables exist [20] lowers the assessment of its rigor from a perfect score to ‘good.’ TAM is judged ‘good’ for the same reasons. The Integrated Model, in turn, integrates proven external variables and thus earns a ‘very good’ evaluation.

No easy way exists to judge the *relevance* of the requirements lists. Young, Watson and Poon, Wagner [21] provide hands-on variables for the EIS design, whereas the variables used by Vandenbosch and Jiang et al. [22] remain on a high level (see Chapter 3.2). Bearing in mind the fact that truly applicable approaches exist, we give them a rating of ‘somewhat’. Considered solely as structural models, the User Satisfaction Models, TAM and the Integrated Model lack direct relevance for practice. For example, the surrogate variables they employ cannot be used directly in IS design. Only the external variables specified in the Integrated Model provide fundamental orientation for practitioners. Due to its structuring dimension, the User Satisfaction

Model receives a rating of 'somewhat'. The external variables used in the Integrated Model lead to an assessment of 'good'. The TAM model is judged with a 'bad', as attitudes and beliefs are not easy for practice to use.

*Transparency* can be considered in terms of the transparency of the model itself and of the model development. The underlying structural models of the User Satisfaction Model, TAM, and Integrated Model ensure transparency, which is judged as 'good.' Furthermore, the use of statistics provides more or less significant results on what variables lead to what results. The requirements lists, in turn, are easy to understand as they simply list with EIS criteria. But the fact that the selection of input variables is not intersubjectively traceable justifies a rating of 'somewhat'.

In terms of *handling*, requirements lists provide a short list of unilateral means-end relationships resulting in a rating of 'good'. Based on their structuring dimensions alone, the User Satisfaction Model and the Integrated Model would earn a 'good' evaluation, but the structural models add complexity that is not easy for practitioners to handle. This provides a downgrade of their ratings to a 'somewhat'. Finally, the internal links are very difficult to understand in the TAM and the Integrated Model, so the TAM receives a 'bad' evaluation and the Integrated Model's is reduced to 'bad'.

The use of surveys to identify unilateral means-end relationships or perform qualitative research is considered as a 'normal' level of *effort in terms of time and cost*. In comparison to that, the requirements lists receive a 'good'. The fact that they use just a few external variables and easy-to-understand statistical methods justifies this evaluation. The User Satisfaction Model, TAM, and the Integrated Model require large surveys to verify their proposed structural equation models, so they receive a bad rating of cost- and time adequancy.

○ ◔ ◑ ◕ ●
very bad - bad - somewhat - good - very good

| | | evaluation criteria | requirements lists | User Satisfaction Models | Technology Acceptance Models | Integrated Model |
|---|---|---|---|---|---|---|
| requirements engineering | requirement identification | completeness | ◔ | ◕ | ◑ TAM 3 | ◕ |
| | requirement analysis and specification | distinctiveness | ◔ | ◑ | ◔ | ◑ |
| | requirement validation | (scientific) rigor | ◔ | ◕ | ◕ | ● |
| | | relevance | ◑ | ◑ | ◔ | ◕ |
| requirements management | | transparency | ◔ | ◕ | ◕ | ◕ |
| | | handling | ◕ | ◑ | ◔ | ◔ |
| | | effort (cost and time adequancy) | ● | ◔ | ◔ | ◔ |

**Fig. 4.** Evaluation of requirements lists and structural models

In summary, the User Satisfaction Model, the TAM, and the Integrated Model can be classified as research methods. They provide a rigorous understanding of external variables and user satisfaction or technology acceptance. However, their direct application for next-generation EIS design faces several obstacles:

- Despite the existence of the Integrated Model, there is a *lack of external variables* to provide concrete advice how to make an IS design successful. Some extensions have tackled this issue [37], but especially for EIS they are rare in literature.
- An extensive survey is always needed to demonstrate the proposed relationships between requirements, surrogate and dependent variables. Often, the results fail to prove the assumed relationships. The external variables provide by the TAM in particular have more to do with attitudes (Chapter 3.4) than *concrete requirements* for an EIS.
- Overall, EIS cases are limited (e. g., for the Integrated Model). Moreover, the cases focus on the individual and organizational impact of EIS or identify levers for successful EIS implementation, such as business skills (e. g., Chapter 3.1.), and cannot serve as criteria for next-generation EIS characteristics themselves.

In contrast, the requirements lists have practitioners thanks to their clear information and system antedecents and their ease of use. However, we assessed such approaches negatively in terms of requirement completeness and, more obviously, (scientific) rigor. A promising solution would therefore be to develop a method that balances these different approaches. The result should be an applicable list of requirements criteria for EIS success, but, these criteria have to be deduced in a more rigorous and transparent way.

## 4 Alternative Approach: Deriving Requirements Criteria for EIS Design from the Principle of Economic Efficiency

Rigor of the structural models described before is based on the methodology of induction: statistical significance of huge surveys was used to apply a conclusion from an individual case to a general one. We will follow Popper [38] and take the opposite approach. Using the principle of *deduction*, we will derive requirements criteria for next-generation EIS design from the principle of economic efficiency, bearing in mind the mechanisms of user satisfaction and technology acceptance.

### 4.1 Principle of Economic Efficiency

In terms of IS design, we propose that only those principles should be taken into account that are aligned with the overall organizational design. In business research, the *principle of economic efficiency* [39] is a generally accepted paradigm. It addresses the ratio between cost and benefit: in our case, it means that a design process must be oriented what is economically feasible.

Even the cost of information (and IS design) are identified to some degree. With the current scientific research, the ability to quantify the profitability of delivered information is limited [40]. So, *surrogate requirements* of IS success are needed.

A next step is to express economic efficiency in a system of *basic criteria*. Following the 'black box' method from mechanical engineering [41] they can be differentiated into *system output* and the *system input* to generate the output. This leads to Kirsch et al. [42], who separate the characteristics of IS into their degree of generality and (solution) power. The first is about for how many issues IS are designed. That is not of concern for this article, because the domain of EIS is set. The latter refers to the objective of IS, users, or technology, etc. It can be separated as follows [43]:

- The *solution capabilities* cover, how IS output is relevant to users.
- The *resource requirements* cover the input needed to generate the output, such as primary information, methods and models, or manpower.

### 4.2 First Level of Specification: Design Criteria

In the IS success models (Chapter 3.3), the IS solution capabilities are determined by identifying information and system characteristics that are relevant for users. The *information support process* should bring them into a proper structure to be complete and distinct (see Chapter 2). The process consists of information need analysis, information synthesis and presentation [40]. Fig. 5 summarizes the design criteria derived. To ensure investment in the most important design topics, the criteria could be *weighted,* e. g. by the later IS users.

The first step, information need analysis, determines the information demand. Applicable here is the design criterion of *completeness*, which encompasses the scope and the structure of the information to be provided to IS users. 'Scope' refers to completeness in its quantitative sense. 'Structure' focuses on its qualitative aspect.

Second, the information supply must also meet users' demand in terms of information synthesis and presentation. The design criterion of *user orientation* applies here [44]. It captures the need to adapt IS to users' working styles.

To allow IS users to work with the information, formal criteria must be considered as well: executives' tasks often changes, e. g. from internal strategic leadership to external communication and vice versa [8], so an EIS must be *flexible* to adapt to these changes. Even if information are presented carefully, they will not be useful, if they are not up-to-date and delivered on time (see the design criterion of *time conformity*). The same is true if its *accuracy* is questionable [23].

The first criterion addressed by the resource requirements is the *data input* needed to generate the required information. The second input criterion should be the *handling* of needed methods and tools to design IS [6]. Last but not least, IS design must be verified regarding their *effort* (cost and time efficiency to design an EIS).

### 4.3 Second Level of Specification: Evaluation Criteria

The outlined design criteria are merely not directly measurable. Therefore, they have to be specified with *evaluation criteria*. Fig. 5 shows the result. The extent to which requirements are called for EIS ('to-be') or existing EIS already fulfills these criteria ('as-is') can be measured, e.g., using a *five-point ordinal rating scale* – in a group of potential users and stakeholders with the arithmetic mean and standard deviation.

The relevance of the issues can then be measured with the difference between 'as-is' and 'to-be' status (see arrows, Fig. 5). If the criteria are weighted, the difference should be multiplied by that figure for a sound list of prioritized design issues.

Starting with the criterion of 'completeness' the scope of executive support can be detailed in terms of how it *covers objective and subjective information needs.* Regarding executives' tasks [6], *strategic information* and *information for regulatory compliance* can detail the information structure (see the evaluation criteria 1-4).

The ‘user orientation’ criterion must address executives’ scarce management time and limited ability to absorb huge data amounts [6]. This can be addressed with the *aggregation level* and *verifiability* of next-generation EIS, which should provide relevant periodic executive information in a compact, hierarchically structured layout.

Executives still tend to be technology-averse [23] and they have a cognitive working style. Easy-to-use system handling will determine their IT acceptance. As a result, another dimension to specify the user orientation of a system is the *quality of its information presentation*, *user interface design and dialog control,* and *predefined features and functions*, e.g., analysis and links with upstream systems.

The *flexibility* refers to the ability of IS to respond promptly to changing circumstances. *Timeliness* indicates the extent information is delivered on time and updated. Finally, *accuracy* can be specified in terms of content and conformity to formal standards (see the evaluation criteria 14-15 of *correctness* and *reliability*).

Details on data input can be provided regarding its *availability in general* and its potential to be *handled with system support* [44]. How well methods and tools to design IS can be handled is indicated in terms of *ease of use* [5]. *Transparency* covers how results are generated and changes are tracked. Because of its comprehensive content EIS design often requires several methods and tools to *interact*, e. g. an information need analysis with success factors, information synthesis with a Balanced Scorecard, etc. This ability should be a third handling criterion. Last, but not least, the effort to design EIS should be verified in terms of *cost and time adequancy*.

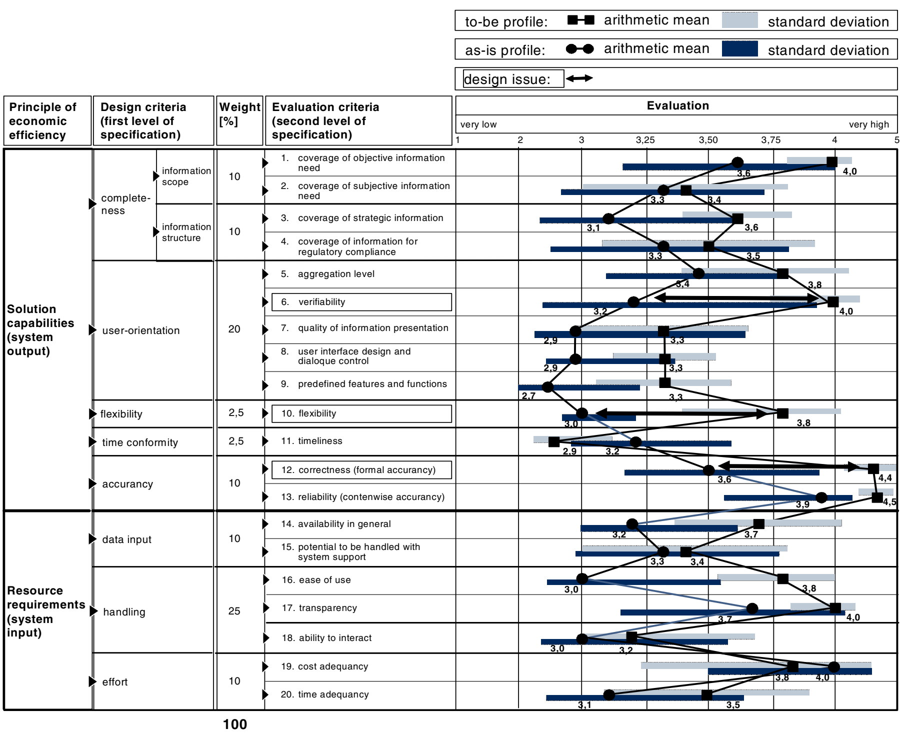



**Fig. 5.** Weighted criteria for next-generation EIS with an ‘as-is/to-be’ profile (example [43])

## 5 Evaluation

Comparing the findings of a first implementation with the comments to requirements lists and the structural models, the method proposed here has the following advantages to be discusses (for the comparison see the criteria in Table 1):

- The principle of economic efficiency enjoys *broad acceptance* – both in business-management research and practice. As a reliable, frequently applied design principle, it should provide a generally accepted starting point for EIS design and the related requirements analysis.
- From a conceptual perspective, deducing basic criteria, design criteria, and evaluation criteria from principle of economic efficiency is scientifically *rigorous*.
- The chosen procedure should lead to a catalog of requirements with a good level of *completeness and distinctiveness* for next-generation EIS. The optional weighting offers an added opportunity for differentiation.

This method offers greater rigor than the requirements lists, making systematic development of requirements possible. Advantages over structural models also exist:

- The *handling* of the catalog of requirements criteria on hand should be easier. Requirements can be defined criterion-by-criterion for a 'to-be' EIS definition. The same is true for evaluating existing IS ('as-is'). A statistically significant sample is not needed to define the structure within the catalog. The *information support process* should bring the output criteria into a complete and distinct structure.
- In addition, the catalog can be used to structure an 'as-is/to-be' profile to identify and especially visualize design gaps and focus on the most important design issues – broken down by every evaluation criterion and *traceable* for third parties.

For these reasons, the catalog of requirement criteria developed with this method should meet executives' requirements for EIS design better than the requirements lists and the structural models mentioned above. However, in terms of scientific rigor: using ordinal-scaled evaluation criteria cause most of the more complex statistical methods to drop out. Another limitation is, that the evaluation always entails some *subjectivity* (the other side of not using huge samples to generate findings).

## 6 Outlook and Directions of Future Research

The objective of this article was to develop a business-driven catalog of requirements criteria and thus contribute to better EIS design. To do so, list approaches and structural models were evaluated using criteria derived from requirements engineering discipline. To bridge the gap between the scientific rigor of the structural models and the relevance of the requirements lists, the article demonstrated the systematic development of business-driven requirements applying the *principle of economic efficiency*. The resulting criteria catalog foregrounds practitioners' perspective without neglecting rigor. A case demonstrates a first implementation and helps to evaluate the results.

Looking ahead, we expect that while functional requirements for EIS are unlikely to change, innovation will continue in terms of their *design principles*. The younger management generation, familiar with IT, will be particularly interested in new, more communicative ways to perform their day-to-day work. In response, EIS should, for example, integrate the working modes of individual desk research, collaborative problem solving, joint decision making in board meetings, and providing information to act upon in close to real time – tasks that until now have been separated [6].

To design such next-generation EIS, the catalog of requirements criteria developed here could be used to evaluate state-of-the-art of EIS design and its implementation status in practice. To do so, the pattern of each design criterion have to be surveyed to create the 'to-be' profile. The end product should be an 'as-is/to-be' comparison to prioritize current design issues. Fig. 5 offers an example of what such a comparison could look like. The information is presented in a structure and style executives will understand, but, with the cost of some statistical limitations due its ordinal scale and a certain amount of *subjectivity.*

Overall, the results should necessarily be applicable to other IS as well and thus contribute to improving requirement analysis in IS design research in general. In terms of the philosophy of science, the rejection of the homo economicus theory we currently see should drive this socio-technical design focus.

## References


1. Pijpers, A.G., Bemelmanns, T.M., Heemstra, F., van Montfort, K.A.: Senior Executive use of information technology. Information and Software Technology 43, 959–971 (2001)
2. Watson, H.J., Wixom, B.H.: The Current State of Business Intelligence. Computer 40, 96–99 (2007)
3. Kelly, J.N.: Executive Information Systems. Patricia Seybold's Office Computing Report 11, 77–83 (1988)
4. Pappas, K.: Executive Information Systems Help Senior Exes Manage More Effectively. PC Week 16, 16 (1988)
5. Nord, J.H., Nord, G.D.N.: Executive Information Systems: A Study and Comparative Analysis. Information & Management 29, 95–106 (1995)
6. Young, D., Watson, H.J.: Determinates of EIS acceptance. Information & Management 29, 153–164 (1995)
7. Rainer, R.K., Watson Hugh, J.: What does it take for succesful executive information systems? Decision Support Systems 14, 147–156 (1995)
8. Mayer, J.H., Winter, R.: Emerging from the economic crisis with a clearer view – next-generation executive information systems. University of St. Gallen, St. Gallen (2009)
9. 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)
10. 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)
11. Brousseau, É., Glachant, J.-M.: New Institutional Economics. Cambridge University Press, Cambridge (2008)
12. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
13. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251–266 (1995)

14. Urbach, N., Smolnik, S., Riempp, G.: The State of Research on Information Systems Success - A Review of Existing Multidimensional Approaches. Business & Information Systems Engineering 4, 315–325 (2009)
15. 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, CA, pp. 83–106 (2006)
16. IEEE: IEEE Recommended practice for software requirements specifications. Institute of Electrical & Electronics Engineers, New York (1998)
17. Rupp, C.: Requirements Engineering und Management: Professionelle, iterative Anforderungsanalyse für die Praxis, vol. 4. Hanser, München (2007)
18. Sommerville, I.: Software Engineering. Addison-Wesley, Harlow (2008)
19. Standish Group: Standish Report, `http://www.standishgroup.com` (accessed 05.01.2010)
20. Wixom, B.H., Todd, P.A.: A Theoretical Integration of User Satisfaction and Technology Acceptance. Information Systems Research 16, 85–102 (2005)
21. Poon, P., Wagner, C.: Critical success factors revisited: success and failure cases of information systems for senior executives. Decision Support Systems 30, 393–418 (2001)
22. Vandenbosch, B., Huff, S.L.: Searching and Scanning: How Executives Obtain Information from Executive Information Systems. MIS Quarterly 21, 81–107 (1997)
23. Jiang, J.J., Muhanna, W.A., Klein, G.: User Resistance and strategies for promoting acceptance across system types. Information & Management 37, 25–36 (2000)
24. Stein, A.: Re-engineering the executive: The 4th generation of EIS. Information & Management 29, 55–62 (1995)
25. Hung, S.-Y.: Expert versus novice use of the executive support systems: an empirical study. Information & Management 40, 177–189 (2003)
26. DeLone, W.H., McLean, E.R.: Information Systems Success - The Quest for the Dependent Variable. Information System Research 3, 60–95 (1992)
27. DeLone, W.H., McLean, E.R.: The DeLone and McLean Model of Information Systems Success - A Ten-Year Update. JMIS 19, 9–30 (2003)
28. Seddon, P.B.: A Respecification and Extension of the DeLone and McLean Model of IS Success. Information Systems Research 8, 240–253 (1997)
29. Ballantine, J., Bonner, M., Levy, M., Martin, A., Munro, I.P. P.: The 3-D model of information systems success: the search. IRMJ 9, 5–14 (1996)
30. Seddon, P.B., Kiew, M.-Y.: A Partial Test and Development of Delone and Mclean's Model of IS Success. Australian Journal of Information Systems 4, 90–109 (1994)
31. Roldán, J.L., Leal, A.: A validation test of an adaptation of the DeLone and McLean's model in the Spanish EIS field. In: Cano, J.J. (ed.) Critical reflections on information systems: a systemic approach, pp. 66–84. IGI Publishing, Hershey (2003)
32. Davis, F.D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quartely 13, 318–340 (1989)
33. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User Acceptance of Information Technology: Toward A Unified View. MIS Quarterly 27, 425–478 (2003)
34. Lee, Y., Kozar, K.A., Larsen, K.R.T.: The Technology Acceptance Model: Past, Present, and Future. Communications of AIS, 752–780 (2003)
35. Venkatesh, V., Davis, F.D.: A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. Management Science 46, 186–204 (2000)
36. Venkatesh, V., Bala, H.: Technology Acceptance Model 3 and a Research Agenda on Interventions. Decision Sciences 39, 273–315 (2008)

37. Taylor, S., Todd, P.: Understanding Information Technology Usage: A Test of Competing Models. Information Systems Research 6, 144–176 (1995)
38. Popper, K.R.: Logik der Forschung. Mohr, Tübingen (1982)
39. Samuelson, P.A.: Foundations of Economic Analysis, Enlarged Edition. Harvard University Press, Harvard (1983)
40. Berthel, J.: Betriebliche Informationssysteme. Schaeffer-Poeschel, Stuttgart (1975)
41. Matek, W., Muhs, D., Wittel, H.: Maschinenelemente - Normung - Berechnung - Gestaltung. Vieweg, Braunschweig (1987)
42. Kirsch, W., Bamberger, I., Gabele, E.: Betriebswirtschaftliche Logistik. Systeme, Entscheidungen, Methoden. Gabler, Wiesbaden (1973)
43. Marx, F., Mayer, J.H.: Report Logic: A User-oriented Reference Model for Corporate Management Systems. University of St. Gallen, St. Gallen (2009)
44. Krcmar, H.: Informationsmanagement. Springer, Berlin (2005)

# Systematization of Maturity Model Extensions

Gerrit Lahrmann and Frederik Marx

University of St. Gallen, Müller-Friedberg-Str. 8,
9000 St. Gallen, Switzerland
{gerrit.lahrmann,frederik.marx}@unisg.ch

**Abstract.** In order to identify and explore the strengths and weaknesses of particular initiatives, managers in charge need to assess the maturity of their efforts. For this, a wide range of maturity models has been developed, but there is no detailed methodical guidance how to extend these models. Therefore, we present a systematization of maturity model extensions.

**Keywords:** maturity model, maturity model extension, systematization.

## 1 Introduction

An established means to identify strengths and weaknesses of certain domains of an organization are maturity models (MMs) [1]. They consist of multiple, archetypal levels of maturity of a certain domain and can be used for organizational assessment and organizational development [1]. Despite the fact that iteration and further development of MMs are central assumptions of maturity modeling, there is no detailed methodical guidance for MM evolution or extension [2, 3]. The goal of this paper is to bridge this gap by systemizing MM extensions.

The paper is structured as follows. In Section 2, we outline methodical foundations of MMs and MM construction. In Section 3, we derive mechanism of the extension of MMs. Concluding, we summarize our findings and offer suggestions for future work.

## 2 Maturity Models and Maturity Model Construction

Maturity is a "state of being complete, perfect or ready" [4]. To reach a desired state of maturity, an evolutionary transformation path from an initial to a target stage needs to be progressed [5]. MMs are used to guide this transformation process. Initially proposed in the 1970's [6], more than a hundred MMs have been published in the field of IS up to date [1, 2]. As these high numbers led to a certain arbitrariness of the design process [1-3], methods for the design of MMs were developed.

A MM consists of several levels (also called stages) of maturity and a number of structuring dimensions. Each level has a distinguishing descriptor clearly providing the intent of the level and a detailed description of its characteristics. Dimensions are capability areas structuring the field of interest. Each dimension is further specified by a number of elements, activities, or measures at each level [3, 5].

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 522–525, 2010.

MMs can be *one-* or *multi-dimensional* and even *hierarchical* by the use of sub-dimensions. Dimensions structure the field of interest, assist in the development of measures and the presentation to the audiences, and should therefore be exhaustive and distinct [1, 3]. Hierarchical MMs are more complex and require a formal architecture of measures. They offer the possibility of separate maturity assessments for discrete areas [3]. MMs can be *continuous* or *staged*. Continuous MMs allow a scoring of activities at different levels. Therefore, the level can be either the (weighted) sum of the individual scores or the individual levels in different dimensions. Staged models require the compliance with the all elements of one level [5]. They specify a number of goals and key practices to reach a predefined level. Staged MMs reduce the levels to the defined stages, whereas continuous MMs open up the possibility of specifying more situational levels. Mettler et al. [1] propose a situational MM that recommends different levels for different types of organizations, so called configurations. This approach specifies different maturity profiles according to defined contingencies [1]. Therefore, the number of audiences of a MM should be differentiated, e. g. industry-specific maturity recommendations. The assessment of maturity can be done *qualitative* using descriptions or *quantitative* using Likert-like scales [5]. Table 1 summarizes the properties of MMs.

**Table 1.** Fundamental characterization of maturity models

| Criteria | Characteristics | | |
|---|---|---|---|
| Dimensions | One-dimensional | Multi-dimensional | Hierarchical |
| Maturity principle | Continuous | Staged | |
| Number of audiences | Single | Multiple (configurations) | |
| Assessment approach | Qualitative | Quantitative | |

The extensibility of a MM depends on the underlying construction principle. Different development processes have been proposed, following a basic design process [1-3]. De Bruin et al. [3] propose a process consisting of the phases of scope, design, populate, test, deploy, and maintain. Becker et al. [2] propose a similar process emphasizing the use of existing MMs and an iterative development. The *scope phase* defines the focus and identifies the relevant stakeholders and targeted audiences of the model. It determines the balance between complex reality and model simplicity. The *design phase* addresses the requirements-based design and outlines the principle concept of maturity, the structure of levels, dimensions, and sub-dimensions (the meta-model). Based on this first design decision, the descriptor of the levels and their definitions are outlined. Thereby, the design process can follow a top-down or a bottom-up approach. A top-down approach first specifies the levels and their descriptions. In the following *populate phase*, the corresponding characteristics are determined. The bottom-up approach first defines dimensions and characteristics representing maturity, and then derives descriptions from it. Different dimensions may have an unequal number of distinguishable characteristics. The populate phase also defines the maturity assessment. This includes the specification of assessment instruments and appropriate assessment questions. Once designed and populated, the constructed model has to be *tested* concerning content completeness and accuracy with regard to the intended scope of the model. Furthermore, the assessment instruments need to be tested regarding validity and reliability.

Next, the model should be *deployed* to the initial stakeholders as well as to an independent community. Finally, the model needs to be *maintained* to ensure its evolution.

For defining and populating MMs, different research methods and combinations of these methods are proposed. Commonly mentioned are literature analysis, Delphi studies, case studies, and focus groups [2, 3]. Quantitative methods are less frequently used for constructing MMs [5], but they can be used for identifying configurations [1]. Testing is also mostly done using qualitative methods. The choice of the research method is influenced by the scope, stakeholders, and targeted audiences [1].

## 3 Systematization of Maturity Model Extensions

The described procedure models for the development of MMs propose an iterative development process [2]. Furthermore, they propose a phase of maintaining the model ensuring the evolution of the model as the domain knowledge changes over the years [3]. Further reasons are technological or management progress which ask for a revision of the model. Also experiences with the usage of the MM in business practice and research necessitate improvements or a change of the MM. Therefore, redefinitions or aging of MMs itself are central assumptions of maturity modeling.

Using the outlined characteristics of MMs (cf. Table 1), conceptual extension mechanisms of MMs can be derived, systemizing the evolution of MMs. These mechanisms are specializations of the general adaptation mechanisms or design principles used e.g. in reference modeling and method engineering. Table 2 contrasts new model development with model evolution approaches.

**Table 2.** Systematization of maturity models (further) development

| New model (innovation) | Model extension/evolution (version) |
|---|---|
| • Development from scratch<br>• Combination of maturity models towards a new model<br>• Transfer of structure and/or content of existing maturity models toward new areas of interest | • Redefinitions or update by maintaining the meta-model of the maturity model<br>• Addition of a new level<br>• Addition of a new (sub-)dimension<br>• Addition of a configuration<br>• Formalization of the model |

The first four model extension/evolution mechanisms (cf. Fig. 1) address updates and extensions of MMs, therefore ensuring relevance of the model. As they focus on specific parts of the model they do not challenge the existing model fundamentally. They are restricted by the content (maturity concept and scope) and the methodical characteristics of the previous model version. For example, a new dimension is bound to existing maturity stages and the description granularity. Furthermore, a new dimension has to be aligned to the same maturity concept, e. g. process efficiency and not process flexibility [1]. Also testing could be restricted to the area of concern. For example, the update of a sub-dimension in a hierarchical model should be possible without challenging the overall model. Beside the model, the assessment instruments also need to be adjusted to the new content. The fifth MM extension/evolution mechanism encompasses methodical aspects. It addresses the degree of formalization of the MM, which may need to be increased because of unsatisfactory model usage in practice. Therefore, formalization aims at ensuring rigor. This includes the movement from a simple meta-model towards a hierarchical structure and the switch from qualitative to quantitative assessment instruments.

In principle, the development of a MM extension can be based on the basic design approaches and procedure models for new model development [2, 3]. The formalization of the whole model may ask for a fully loaded development cycle.

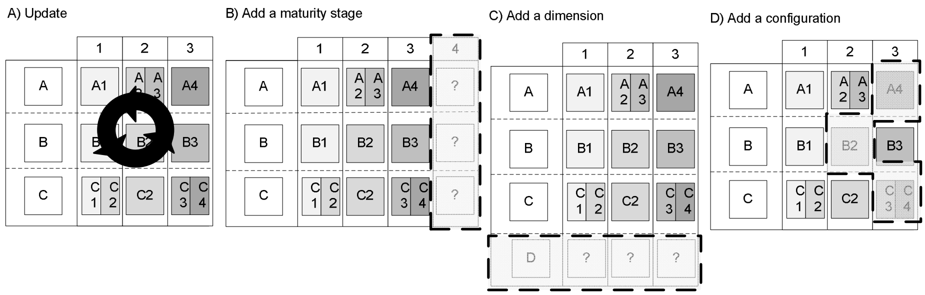



**Fig. 1.** Maturity model evolution mechanisms

## 4 Conclusion and Outlook

Many MMs has been developed, but there is no detailed methodical guidance how to extend these models. Therefore, we systemized MMs extensions. The maturity concept and the scope of the MM have been identified as central antecedents for extensions. A sufficient documentation of the development process and the MM itself is necessary to avoid any mismatch between the extension and the original MM. Methodically, there should be more detailed guidance for each extension mechanism, e. g. by proposing procedure models which especially address extension mechanisms, as the process of adding a dimension is different from adding a level.

## References

1. Mettler, T., Rohner, P.: Situational Maturity Models as Instrumental Artifacts for Organizational Design. In: Proceedings of DESRIST (2009)
2. 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)
3. de Bruin, T., Rosemann, M., Freeze, R., Kulkarni, U.: Understanding the Main Phases of Developing a Maturity Assessment Model. In: Proceedings of ACIS (2005)
4. Simpson, J.A., Weiner, E.S.C.: The Oxford English Dictionary. Oxford University Press, Oxford (1989)
5. Fraser, P., Moultrie, J., Gregory, M.: The Use of Maturity Models/Grids as a Tool in Assessing Product Development Capability. In: Proceedings of IEMC, pp. 244–249 (2002)
6. Gibson, C.F., Nolan, R.L.: Managing the four stages of EDP growth. Harvard Business Review 52, 76–88 (1974)

# Process Design as Basis for Comprehensive Process Modeling

Stefan Jablonski and Stephanie Meerkamm

Chair for Applied Computer Science IV, University of Bayreuth
Universitätsstr. 30, 95447 Bayreuth, Germany
{stefan.jablonski,stephanie.meerkamm}@uni-bayreuth.de

**Abstract.** Today, with process management generally established as a management tool, there is a stringed interest in process modeling. Although there is a multitude of standard process modeling techniques available, often the modeling results are not satisfying. One of the reasons could be that the so called design phases including a requirement analysis and the implementation in an appropriate modeling language and tool is neglected. Thus, we want to offer a framework which focus more on the design and gives the process modeler the option to design an adequate process modeling language and tool.

**Keywords:** process management, process design, process modeling, meta model hierarchy, specification of process modeling languages.

## 1 Introduction

In order to motivate our concern to put more emphasis on the design phase of the process management life cycle [2], [4], [7] we start with an example: Produce a part of metal with certain granularity, an appropriate tool is necessary. "Appropriate" means that the tool offers the functionality/ technique to realize the fine tuning with the desired level of granularity. Besides, the parts must be between 10 and 11 cm long. Transferring the idea into the context of process management the following analogy comes up: for the definition of a process model an *adequate process modeling language* must be provided that is capable to describe all features of the process. Criteria for "good" process modeling called *modeling constraints* and *requirements related to process execution* have to be set (DESIGN). Afterwards, a process modeling tool is used to create the process model considering the posted requirements (MODELING). Then, these processes can be executed.

We criticize that in process management design with an explicit requirement analysis of the process model and the related tool is not adequately considered. Very often standard modeling languages/ tools are taken so that most probably not all requirements can be implemented. We do not want to blame standard process modeling languages and it is obvious that the usage of a domain specific language also has its drawbacks: the development is time and cost consuming, above all portability and exchangeability of process models is reduced. Thus, it has to be balanced whether *accuracy* (offered by domain specific tools) or *portability* (offered by standard tools) is more relevant for a project.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 526–529, 2010.

The overall aim of our research is to focus on the production of qualitatively good process models. This means, that all characteristics of a process are modeled. The most important factor is the initial requirement analysis in the design phase, as wrong decisions in this phase cannot be compensated in the following phases of the process life cycle any more (cf. [5], [6]). Furthermore it must be done independent of the discussion how these requirements can be implemented. Thus we offer a methodology to highlight the design phase. We base our approach on a meta model hierarchy defining a framework for the definition and usage of processes modeling languages, where the requirements analysis and the whole design phase can be related to.

In Section 2 we introduce the meta model hierarchy as basis for our approach. The integration of design and modeling into this method is explained in Section 3, presenting our new ideas more into detail in Section 4. Section 5 concludes the paper.

## 2 Basis Method: Meta Model Hierarchy

As foundation for our ideas we use a meta model hierarchy [1], [3] which is functionally comparable to MOF (Meta Object Facility). Within the meta model hierarchy a *language paradigm*, *process modeling languages, process models* and *process instances* are defined at different levels of abstractions, namely *M3, M2, M1, M0.*

In the following Section we explain in which way design and modeling actually affect this hierarchy and in which way it can integrate and structure the design.

## 3 Design and Modeling in the Meta Model Hierarchy

We now take the basis method from Section 2 and map the process life cycle to it.

**Traditional Interpretation:** *Modeling* is associated with layer M1 of the meta model hierarchy. *Design* is mostly seen as preparation of modeling by selecting a modeling language and defining modeling constraints. Thus it is mainly located on M1, too. Requirements demanding domain specific model elements are mostly neglected indicated by referring the design phase to M2 only to a small extend. On M0 requirements referring to process execution are considered.

**Extended Interpretation:** Since requirements should not only influence the definition of process models but also the functionality of a process modeling language, *design* is relevant to M1, to M2 but to a much greater extend as it is actually the case; and to M3 as it might be the case that a new modeling paradigm is necessary. Thus, design tackles the whole hierarchy and each design level can be assigned to the appropriate modeling level (see fig. 1). In view of our research goal to improve the quality of the process models, mainly the design phase on M2 should be focused more, which allows an individual development of a modeling language.

In the next Section we discuss the design issues for the meta model hierarchy and show what modeling tasks are derived from them.

## 4 The Layers of the Meta Model Hierarchy

In this Section we explain the framework and the tasks of design and modeling. Access is possible on each hierarchy level; the procedure is not as linear as illustrated, but is characterized by many loops.

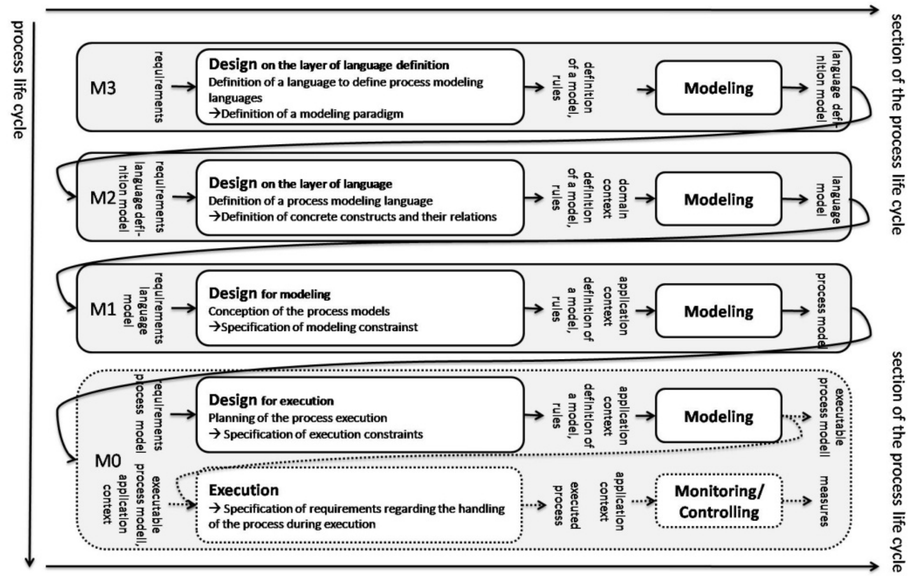



**Fig. 1.** Meta Model Hierarchy

**M3:** On M3 the basic paradigm of the language has to be defined, which finally results in a language definition model. First of all *features of the modeling language* have to be defined, e.g. a directed graph comprising nodes and edges. Abstract *modeling and execution tools* have to be defined (DESIGN). The usage of the abstract modeling language results in an abstract meta model (= *language definition model)* (MODELING). The term "abstract" means that there is no reference to a use case; it serves as theoretical fundament for the further development on the lower levels.

**M2:** Based on the language definition model of M3 a model for a process modeling language is defined. For this, *constructs* are derived from the abstract features of M3. Aspects of visualization are stated, e.g. processes are quadrates, flows are continuous lines. The *modeling tool* has to be adapted on a conceptual level and the *execution tool* is made more concrete (DESIGN). It finally results in the *concept for a modeling language* (MODELING). All this is necessary in case the functionality, grammar or content of a given language is not sufficient.

**M1:** A concrete process model for a special use case has to be generated using a process modeling language defined on M2. Firstly, a *decision about a modeling language* is necessary. *Modeling rules* have to be defined, the modeling language has to be integrated into a *modeling tool* and the *execution tool*, has to be installed (DESIGN). Then the business activities and their aspects are described in a *process model* (MODELING). Nevertheless, the scope of action is limited to the functionality of the modeling language/ tool, which can reduce the quality of the process models. Modeling states a compromise between functionalities of the language and the actual circumstances. Thus, we focus at M2 for an appropriate preparation.

**M0:** M0 does not belong to the modeling environment anymore; the process models are executed. To a small extend design can be assigned to this level, too, referring mostly to *organizational or execution constraints*; the process model has to be integrated into the *execution tool* (DESIGN). This states the transition to the execution, where monitoring and controlling already starts. (EXECUTION, MONITORING/ CONTROLLING). Experts spent lots of time and effort to optimize the processes "on the road" due to unpredictable events. Nevertheless, many issues can be traced back to missing aspects in a process model or a modeling language. Thus it should be placed more emphasis at the design to implement the requirements.

## 5 Conclusion

We are aware that regarding this paper the presentation of a research methodology is missing, as well as an evaluation and the discussion of limitations.

In this paper we introduced an extended interpretation of design and modeling. The main goal is to improve the quality of the process models and with this the whole process life cycle. We offered a framework which is based on a meta model hierarchy structuring the design and modeling. With this the process modeler should be able to develop its own process modeling language to guarantee expressive process models. This provides the basic for the quality of the process execution.

## References


1. Aktinson, C., Kühne, T.: The Essence of Multilevel Metamodeling. In: 4th International Conference on the Unified Modeling Language. Springer, Berlin (2001)
2. Becker, J., Kugler, M., Rosemann, M.: Processmanagement. Springer, Heidelberg (2003)
3. Henderson-Seller, B., Gonzalez-Perez, C.: A power-type-based metamodeling framework. Software and System Modeling 5, 72–90 (2006)
4. Lee, R.G., Dale, B.G.: Business Process Management: A Review and Evaluation. Business Process Management Journal 4, 214–225 (1998)
5. Recker, J., Indulska, M., Rosemann, M., Green, P.: Business Process Modeling - A Comparative Analysis. Journal of Association for Information Systems 10, 333–363 (2009)
6. Rosemann, M.: Potential Pitfalls of Process Modeling: part A. Business Process Management Journal 12, 249–254 (2006)
7. Weske, M.: Business Process Management - Concepts, Languages, Architectures. Springer, Heidelberg (2007)

# Design Science Research Engagement: Proposal for an Engagement Approach for Company Collaboration

Anita Friis Sommer and Charles Møller

Aalborg University, Fibigerstraede 16,
9220 Aalborg Øst, Denmark

**Abstract.** Design science research is experiencing a comeback within IS research, as there are movements towards more fashion setting and practitioner relevant research. An engagement approach for design science research is proposed to engage in practical collaboration with companies. The approach allows for researchers to actively participate in artifact development and implementation in case companies, which generates firsthand knowledge for following generation of new theories. The proposed approach is relevant for potentially all design science researchers within IS research and is based on generally accepted design science research models.

**Keywords:** Design science research, research paradigm, engagement approach.

## 1 Introduction and Background

During recent years academic research methods in IS has experienced a change of general acceptance from being purely on quantitative and inductive studies to broaden towards more qualitative research approaches. "The dominant research paradigms that we use to produce and publish research […] continue to be those of traditional descriptive research borrowed from the social and natural sciences. We recently accepted the use of interpretive research paradigms, but the resulting research output is still mostly explanatory"[1]. "The dominant research philosophy has been to develop cumulative, theory-based research to be able to make prescriptions"[2].

IS research methods can be divided into two types: Exploration and Explanation research, where explanation research includes the typical deductive research methods, which have been dominant in IS literature. "Exploration research compliments explanation research by producing artifacts that can be used as raw material for evaluation research"[3]. When explanation research is considered the only optional research method, development of artifacts and thereby new fashions or fads is left to practitioners. In this scenario researchers are merely fashion followers or fashion critics [4]. There are sound arguments for IS researchers to engage in both exploratory and explanatory research. Not only can research benefit from linking the two disciplines, but researchers will also be undertaking fashion development as part of the research agenda. "Such 'engaged research' is participative and benefits from differing perspectives. From our perspective, engaged research also provides an avenue for better participation by IS scholars in the fashion-setting process" [4]. In design science research

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 530–533, 2010.

the goal is to combine these research methods, and the approach has been recognized in for instance MIS quarterly in 2009: ”Research methods such as action research, design science, and practice research bring the scholarly research directly into the hands of practitioners. These research methods are now accepted in information systems, but need to be encouraged by Ph.D. programs and editorial boards”[4]. Design science research is a design-oriented or prescriptive research approach. The purpose of design science is to: “… develop knowledge that the professionals of the discipline in question can use to design solutions for their field problems”[5]. At its core, design science is directed towards understanding and improving the search among potential components in order to construct an artifact that is intended to solve a problem [4].

## 2 Design Science Engagement Model

Based on a review of the design science research literature, a design science engagement model is utilized as the underlying model for company collaboration. This model is based directly upon previous theories, especially the work of Holmström and Ketokivi (2009)[3] has inspired to research phases and sub phases. The model presented here differs proposing that not only one but multiple artifacts are generated throughout the research processes [6]. The purpose of the engagement model, depicted in table 1, is to formalize collaboration with companies in developing artifacts both benefiting the company and scientific research.

**Table 1.** Design science engagement model

| Research Phase | Sub Phases | Benefit for Company | Theoretical Contribution | Artifacts | Logical Formalism |
|---|---|---|---|---|---|
| **Solution Incubation** | Detect problem<br>Detect Goals<br>Detect possible solutions | Awareness of problems<br>Goal setting<br>Awareness of solutions | - | Constructs | Abduction<br>Action research |
| **Solution Refinement** | Develop solution<br>Refine solution<br>Implement solution | Solution to problems through artifacts<br>Development and implementation | - | Constructs<br>Models<br>Methods<br>Instantations | Abduction<br>Action research |
| **Explanation** | Develop substantial theory based on previous phases | - | Substantial and/or formal theory based on artifact developed and refined by academic researchers | New Theories | Deduction |
| **Conclusion** | Overview of process | Final Evaluation | | | Deduction |

It is argued that scientists today mainly engage in phases three and four which are explanatory, and that it is left to practitioners to create artifacts thereby developing fashions within IS [4]. In the engaged approach proposed in this paper the scientist is part of the exploratory process as well, which not only makes scientists participate in fashion setting, but also provides better explanatory research [3]. When the researcher is engaged in development of the solution to the problems, the knowledge of the case and the development and implementation process is experienced firsthand. This kind of firsthand knowledge cannot be obtained in other ways than a close engagement in the process. Hence researchers only relying on artifacts developed by others will always have a poorer foundation for their research as the knowledge is secondhand.

## 3 Engagement Approach Applied to a Case Company

Research practice at Aalborg University includes extensive engagement in companies with a problem oriented [8] and participatory approach [9]. Inspired by this knowledge base in combination with the design science model described above, an engagement approach is proposed. Due to limit of space in the article format, only a fraction of the approach is presented here.

**Step1: Solution incubation:** Company formulated problems are investigated and described. To identify if the perceived problems might be symptoms to underlying problems in the company, the investigation includes an analysis of the perceived problem based on collected case study data. The goal is not to prove or disprove perceived problems but rather to investigate and understand perceived problems in a larger context. When performing these investigations, it seems evident that a pattern will always turn up (like in detective work [6]) and through the patterns a new light is shed on the initial problem.

**Step 2: Solution Refinement:** Includes all steps from developing a solution to implementation of the chosen solution is finalized. These steps are not specified further here. Essential at this stage is participation of the company in development of artifacts. Ownership of the artifact is important to establish prior to implementation and thus the company must have some involvement in artifact development and refinement.

**Step 3: Explanation:** The final outcome of this process is evaluation of theories based on collected data. Compared to traditional IS research, where there is no development and implementation of artifacts, theories are not picked for evaluation prior to the research study. The rich data developed and collected in the previous steps is used to evaluate theories, which have been pragmatically selected as lenses during the previous exploratory steps.

**Step 4: Conclusion:** The final step is to conclude on the process together with the company. This is in practice an important step that clearly defines when the project is finalized, and benefits for both parties can be examined. For future collaboration it is essential to be able to conclude on results from previous joint projects.

## 4 Discussion

Many authors state that IS research in the future should be more relevant for practitioners and participate in fashion setting within IS [3-5]. The purpose of using a design science engagement approach is to develop artifacts, which are potentially trend setting and relevant for practitioners, while evaluating theories and take part in the academic debate. Thus the engagement approach meets the current demands for new methods within IS research. The approach presented here in brief can prove to be highly valuable in changing IS research to be closer to fashion setting and more relevant for practitioners.

Design science research engagement creates a unique win-win situation, which not only creates an ongoing flow of interested case companies and supports the relationship between practitioners and academics; it also creates an opportunity for better theory development, through better knowledge of the case, obtained through the abductive reasoning processes. The research facility at Aalborg University, which is the basis case of this paper, is primarily engaged in design science research with focus on close collaboration with the business community. A developed 'win-win-situation' is generally recognized by the surrounding companies, which results in an ongoing flow of companies interested in collaborating with the researchers and their students. This situation is much different than the typical situation for descriptive research facilities, where the main goal is to exploit companies as case studies for academic gains. There are considerations to discuss when utilizing the approach for design science research. The approach is inspired by the problem based learning model used at Aalborg University in collaboration with various companies [7] but it has not been explicitly tested through subsequent research studies. Hence the approach could prove to be more dependent on contingent factors than initially believed.

## References


1. Peffers, K., et al.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2008)
2. Iivari, J.: A Paradigmatic Analysis of Information Systems As a Design Science. Scandinavian Journal of Information Systems, 39–64 (2007)
3. Holmström, J., Ketokivi, M.: Bridging Practice and Theory: A Design Science Approach. Decision Sciences 40(1), 65–87 (2009)
4. Baskerville, R.L., Myers, M.D.: Fashion Waves in Information Systems Research and Practice. MIS Quarterly 33(4), 647–662 (2009)
5. 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)
6. Vaishnavi, V.K., Kuechler Jr., W.: Design Science Research Methods and Patterns: Innovating Information and Communication Technology. Auerbach Publications, Taylor & Francis Group, NW, USA (2008)
7. Lehmann, M., et al.: Problem-oriented and project based learning (POPBL) as an innovative learning strategy for sustainable development in engineering education. European Journal of Engineering Education, 283–295 (2008)

# Design Science in Research Cooperations with the Industry: Findings from Three Prototyping Projects

Sven Weber, Roman Beck, and Robert Gregory

University of Frankfurt, Grüneburgplatz 1
60323 Frankfurt, Germany
{svweber,rbeck,gregory}@wiwi.uni-frankfurt.de

**Abstract.** A challenge that IS researchers face in general is to combine the goals of generating new scientific knowledge while at the same time producing practically relevant research results, e.g., in the form of IT artefacts. To combine rigor and relevance, researchers and practitioners need to collaborate to develop and employ methods that enable both the systematic generation of scientific insights and the knowledge exchange between academia and industry. In this paper, we present the findings of a research project where we entered into an industry-academic collaboration with the financial services industry involving three software development and implementation projects. We adopted a design science research approach to accompany the project and to guide the scientific discovery process. In the course of our research process we developed an innovative research model that integrates our experiences from the research project with existing design science research models.

**Keywords:** Design Science Research, Prototyping, IT Artefact.

## 1 Introduction

In this paper, we present the findings of a research project where we entered into an industry-academic collaboration with the financial services industry involving three Grid prototyping projects. The goal of all three prototyping projects was the implementation of innovative IT solutions to tackle business challenges in the financial services sector. In the very beginning of these projects, the question arises how to combine both rigor and relevance in such a joint research and development projects. When we entered into these three projects simultaneously in the beginning of 2007 we started with a design science research approach as proposed in the current literature [1-5]. In the course of our research process, we soon realized that we had to adapt the guidelines into a specific research model to account for our particular research setting. As Hevner himself stated, the proposed guidelines should not be viewed as a cook recipe and researchers need to take a mindful approach in a design science research project [1]. Accordingly, we constantly iterated back and forth between the three Grid prototype development projects and the design science research literature during our research project to combine rigor with relevance. Due to the setting of our research approach, we were able to improve the quality and innovativeness of the developed prototypes while at the same time generating new insights of the domain of study.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 534–537, 2010.

As a result we present a design science research model that combines both rigor and relevance.

## 2 Design Science Research Model for Industry-Academic Projects

The goal of our industry-academic cooperation was on the one hand to build and evaluate problem specific IT artefacts in form of software prototypes and on the other hand to accompany this process from a scientific point of view. Therefore, Winter's model [5, p. 472] provided us an ideal basis for our design science research project. Unfortunately, Winter's model [5, p. 472] does not encompass the knowledge base (e.g. theories, data analysis techniques, etc.) and the environment (e.g. people, processes, etc.) which had deep impacts in our project. For instance, theories and data analysis techniques helped us to evaluate the prototypes in terms of their functionality. Moreover, simulation and Black-Box testing techniques were also derived from the existing knowledge base. Key factors in our industry-academic project were the involved people and the organizational processes. In the case of the third prototype, a standardised process of the Bank was to define the problem beforehand and to include an IT solutions provider afterwards. People and their intentions (e.g. a quick solution) had deep impacts on the first outcomes of the first prototype. The beta version of the first prototype did not meet the requirements. We discovered that the preceded requirements analysis was not realized sufficiently enough. The intention was to create a quick solution and therefore several mistakes occurred. Moreover, in the case of the third prototype the responsible key account managers changed over time and delayed the implementation of the prototype to a later point in time. Therefore, the knowledge base and environment demonstrated major parts in our project beside the creation and evaluation of the IT artefacts. Hence, Hevner et al. [1, p. 80] provided us the opportunity to include these aspects with his model.

Fig. 1 presents our extended and refined model. At the beginning the researchers and project members enter the field by the use of an specified entry point which is derived from the environment according to Peffers et al. [6]. In our project, the problem identification of the first and third prototypes was not defined beforehand. In contrast, the problem of the second prototype was defined beforehand and a usable solution was already established. The goal was to create an improved version of an existing solution. Our first insight was that the environment is a key factor for an industry-academic cooperation. Additionally, the different stakeholders have to define common interests to analyse the problem at hand. Different goals and interests can lead to significant project delays and problems as it occurred in the third or first prototype. The following 'OR' depicts the choice to process one or both possibilities. Thereby, the left path depicts design science which encompasses the reflection of the IT artefact construction and evaluation according to Winter [5, p. 472]. The process of improving the design science method or creating new methods can be outlined as a search for a suitable design [1, p. 83, Guideline 6]. The outcomes of this search process can also be seen as a new IT artefact, e.g. a new methodologically insight. Therefore, the search process creates rigor for the practical use but at the same time requests relevance from the practice. Thus, a contribution to design science research can be created by improving the design science method or creating new methods in terms of literature background or the evaluation of an IT artefact [1, p. 83, Guidelines 3 and 5]. The improvements of the creation and evaluation step of our three prototypes were influenced by this design science path.

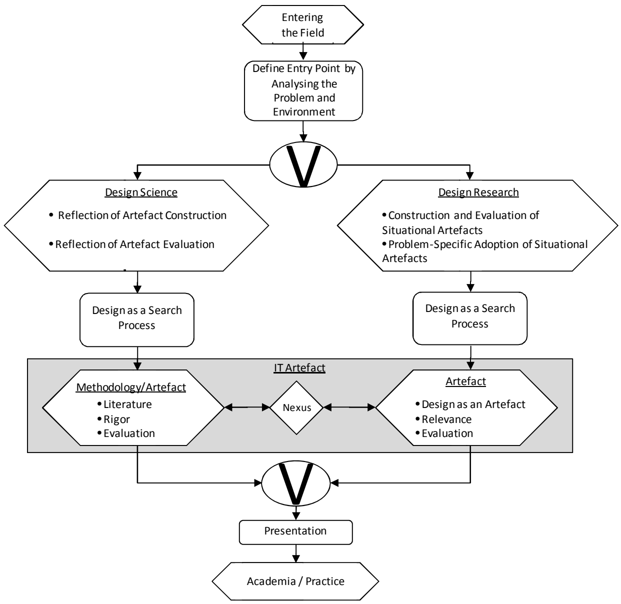



**Fig. 1.** Design science research model

The right path depicts design research which encompasses the construction, improvement, and evaluation of situational IT artefacts and the problem-specific adoption of them [5, p. 472]. Our three developed prototypes represent such design research approaches. Nevertheless, the insights and occurring problems of the prototyping projects influenced our scientific understanding of the construction and evaluation. Thereby, we refined our overall design science research lens.

According to Hevner et al. [1, p. 80], design research encompasses the application in the appropriated environment and thereby creates relevance for the IT artefact. At the same time this path needs scientifically rigor to ensure a theoretical basis for the developed IT artefact. A high emphasis lies on the 'design as a search process' [1, p. 83, Guideline 6] which encompasses both, the building and improvement of an IT artefact. The contribution is presented by the IT artefact itself and its evaluation [1, p. 83, Guidelines 1, 2, and 3, 5]. Therefore, we included the well-defined evaluation process of Hevner et al. [1, p. 86] in the implementation of our three prototypes.

The design research and design science path [5, p. 472] join each other in the last step of our extended model in terms of the presentation and communication to the academia and practice and therewith their justification [1, p. 83, Guideline 7, 6, p. 54].

The major innovation of the presented model is the nexus between the artefact and the methodology. By our industry-academic projects we have found that both paths are important for industry-academic consortiums. On the one hand, the design science path supports the prototyping with helpful guidelines. On the other hand, the design research path provides the relevance for the whole project. In our project, a parallel conduction but mutual improvement led to more rigor and relevance. Moreover, an improved understanding of the design science methodology was established. In fact, the constant iteration of the prototype development and the design science literature leaded to a successfully and satisfactory ending for all stakeholder groups.

## 3 Conclusion

In this paper, we presented the findings of a research project where we entered into an industry-academic collaboration involving three software development projects. We adopted a design science research approach to accompany the development and evaluation of the IT artefacts. Through this process, we developed an extended design science research model which combines rigor with relevance. This is done by integrating elements of design science on the one hand and design research on the other hand into the research process.

We see a nexus between the knowledge base and the environment. Both inform and support the IT artefacts development and evaluation process. On the one hand, the contribution to practice is a structured way to pursue a design science research approach in a development project. On the other hand, the theoretical contribution is that both relevance and rigor can be combined in an industry-academic project.

The limitation of this paper is the focus on industry-academic projects. Moreover, our model has not proven its consistency yet. Future research has to evaluate the model against follow up prototyping projects. Thereby, our research is not an overall recipe but a first effort to combine both perspectives.

## References

1. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
2. McKay, J., Marshall, P.: A Review of Design Science in Information Systems. In: 16th ACIS, Sydney, Australia (2005)
3. Simon, H.A.: The Sciences of the Artificial, 3rd edn. MIT Press, Cambridge (1996)
4. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. ISR 3, 36–59 (1992)
5. Winter, R.: Design Science Research in Europe. EJIS 17, 470–475 (2008)
6. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. JMIS 8, 45–77 (2008)

# Towards a Theory on Collaborative Decision Making in Enterprise Architecture

Agnes Nakakawa[1], Patrick van Bommel[1], and Erik Proper[1,2]

[1] ICIS, Radboud University Nijmegen
P.O. BOX 9010 6500, GL Nijmegen, The Netherlands
[2] CITI, CRP Henri Tudor Luxembourg, Luxembourg
A.Nakakawa@science.ru.nl, pvb@cs.ru.nl, e.proper@acm.org

**Abstract.** Several challenges in enterprise architecture development indicate the need for collaborative decision making to be deployed during architecture creation. However, how this should be achieved remains ad hoc. 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. The first iteration to evaluate the relevance of the concepts in this theory was done using an exploratory survey, and the findings are briefly presented.

**Keywords:** Enterprise Architecture, Collaborative Decision Making.

## 1 Introduction

Some challenges in enterprise architecture development can be addressed by complementing architecture approaches with collaborative decision making. Therefore, this research uses the Design Science research methodology to develop a method for supporting collaborative decision making during enterprise architecture creation. Since the method will specifically enable Collaborative Evaluation of Enterprise Architecture Design Alternatives (CEADA), it is herein referred to as CEADA. Design Science facilitates the creation and evaluation of practical innovative artifacts for solving significant organizational problems [2]. The resultant artifact in this research is the CEADA method. However, there is need to formulate a theory, based on existing theories, that will guide the development of CEADA. This theory can help researchers and practitioners to gain insight into the orchestration of key determinants for collaborative decision making to be successfully realized during enterprise architecture creation. Section 2 presents the evolving theory and section 3 presents the results from the first iteration of the theory as well as the conclusion.

## 2 Joint Decision Making in Enterprise Architecture

Theory refers to the body of knowledge that describes, explains, and increases understanding of a situation in order to predict future occurrences and to lay

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 538–541, 2010.

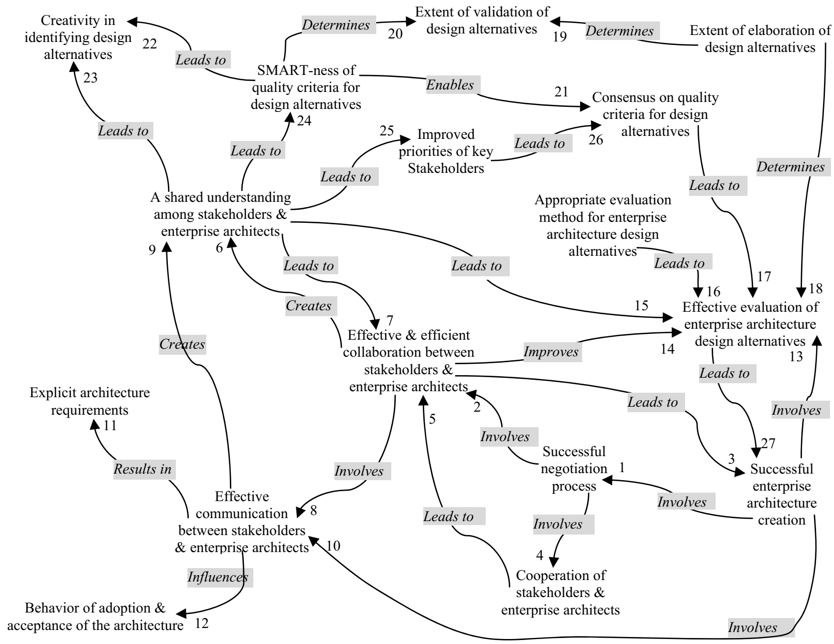



**Fig. 1.** Theory on Collaborative Decision Making in Enterprise Architecture

a foundation for improvement opportunities [3]. The situation of interest is the effective and efficient deployment of collaborative decision making into architecture creation in order to deliver an understandable, feasible, acceptable, and efficient enterprise architecture for an organization. This section shows how the existing knowledge on enterprise architecture creation and (collaborative) decision making is used to explain this situation and predict ways in which it can be addressed. The guidelines for theory formulation in [3,7] have been used to formulate the theory shown in figure 1 and briefly explained below.

Enterprise architecture creation involves activities such as: understanding the purpose of the architecture, creating a shared understanding of the 'as-is' and 'to-be' contexts of the organization, determining possible impacts of the desired transformation, and communicating with stakeholders [9]. Decision making, on the other hand, generally involves: intelligence (investigating an environment for any need for improvement), design (devising possible decision/design alternatives, and choice (selecting the appropriate decision alternative) [1]. Moreover, collaborative (or joint or cooperative) decision making involves several individuals cooperating to arrive at a joint decision, which will yield joint consequences for each individual [6]. From these definitions, collaborative decision making in enterprise architecture creation can be defined to involve enterprise architects and organizational stakeholders *cooperating* to: gain *mutual understanding* of the 'as-is' and 'to-be' contexts of the organization; *identify* and *devise* possible design alternatives for realizing the 'to-be' (or target) organization context; *evaluate* the possible impacts of these design alternatives; and finally *select* the design alternative that is understandable, feasible, acceptable, and efficient. Figure 1 shows the concepts, their relations, and sequences for explaining this definition.

The joint decision in negotiation theory is not only the final decision in a given project; because throughout a negotiation, joint decision opportunities emerge that eventually lead to the final (joint) decision [6]. Relating this to architecture creation, enterprise architecture can be perceived as a collection of joint decisions that have been made throughout the phases of architecture creation. Moreover, development of enterprise (or reference) architecture can be a negotiation process among the units involved [5]. Better still, negotiations help stakeholders understand why all their concerns can not be satisfied. Hence the implication of relation marked 1 and the essence of negotiation theory in architecture creation. Relations 2 and 4 are derived from the definition of negotiation given above. Relations 1 and 2 lead to relation 3 (which is in line with what is recommended, in e.g. [5,9,8], that during architecture creation architects need to collaborate with stakeholders. Relations 1, 2, 3 yield a sequence denoted as 1 – 2 – 3. Since cooperation is when an individual renders the (expected) effort to a group result without intentionally frustrating the efforts of others [10], then the cooperation of individual stakeholders and architects leads to effective and efficient collaboration (hence relation 5). Moreover, since stakeholders provide the organizational resources; determine the requirements and constraints of the architecture; influence others; or are decision-makers, their cooperation is vital for the success of the architecture project [9].

A collaborative environment involves people purposely spending as much time understanding what they are doing as actually doing it, and aiming at creating a shared understanding that didn't exist before [12]. This definition is the basis for relation 6. Moreover, a shared understanding is a basis for effective collaboration [11], and stakeholders' commitment increases as they gain shared understanding of the 'as-is' and 'to-be' aspects [5]. Hence relation 7. From [5], effective communication eliminates ambiguities and this results in explicit requirements for the architecture (see relation 11) as well as positively influencing the acceptance and adoption of the architecture (see relation 12). This results in relations 8, 9, 10. Addressing stakeholders' concerns requires the architect to develop architecture views that show the trade-offs required to resolve conflicting concerns [8]. Such trade-offs are clarified through evaluation of (solution and design) alternatives [9]. Moreover, satisfactory solutions are obtained through evaluating possible (design) alternatives or courses of action [1]. Hence relations 13, 27, 16. For complex problems it can be difficult for an individual to understand and foresee all implications of a given decision, and therefore the best decision requires combining expertise of people from different disciplines [10]. Hence relation 14. Note that sequence 14 – 27 confirms relation 3. Stakeholders' commitment increases as they acquire a shared understanding [5] or a shared goal [10]. Hence relations 25, 26. Consensus on quality criteria for evaluating alternatives will lead to effective evaluation of alternatives (see relation 17). Sequences 25 – 26 – 17 and 7 – 14 imply relation 15. If stakeholders have acquired a shared understanding, then they can unambiguously define quality criteria for design alternatives, and this leads to SMART (Specific, Measurable, Achievable, Realistic, and Time bound) criteria. Hence relations 24, 21, 22, 20, 18, 19. Sequence 24 – 22 implies relation 23.

## 3 First Iteration Using an Exploratory Survey

Data can be used to provide support for a theory [7] or to underpin it [3]. An exploratory questionnaire survey on a sample of 70 enterprise architects, was conducted with the aim of evaluating the relevance of concepts that constitute the theory in figure 1. Findings indicate that although 96% of architects execute architecture development as a collaborative process, 90% of them still face challenges related to acceptance of the architecture results. Examples of these challenges include: some organizations lack a clear decision making unit; difficulty in ensuring that all key stakeholders understand the architecture; the architecture sometimes conflicts with personal ambitions; lack of commitment from stakeholders who were not earlier involved; etc. These are byproducts of the quality of collaboration between architects and stakeholders during architecture creation. These challenges can be minimized through considering sequences 6 – 7, 8 – 9, 12 – 13, 24 – 22, 25 – 26 and relations 11, 19, 21, and 23 in figure 1. The survey also revealed factors that hinder effective collaboration, challenges faced when evaluating architecture design alternatives, and methods architects use to manage collaboration with stakeholders. Due to space limitations these aspects can not be discussed here.

## References


1. Simon, H.A.: The New Science of Management Decision. Harper and Row, New York (1960)
2. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
3. Gregor, S.: The Nature of Theory in Information Systems. MIS Quaterly, 30(3), 611–642 (2006)
4. Lankhorst, M., et al.: Enterprise Architecture at Work: Modelling, Communication, and Analysis. Springer, Heidelberg (2005)
5. Janssen, M., Cresswell, A.: The Development of a Reference Architecture for Local Government. In: HICSS. IEEE Press, Los Alamitos (2005)
6. Raiffa, H., Richardson, J., Metcalfe, D.: Negotiation Analysis - Science & Art of Collaborative Decision Making. Belknap Harvard, Cambridge (2003)
7. Sutton, R.I., Staw, B.M.: What theory is not. ASQ 40(3), 371–384 (1995)
8. The Open Group Architecture Forum. TOGAF Version 9. Zaltbommel. Van Haren Publishing, The Netherlands (2009), ISBN: 978-90-8753-230-7
9. Op ‘t Land, M., Proper, H.A., Waage, M., Cloo, J., Steghuis, C.: Enterprise Architecture - Creating Value by Informed Governance. Springer, Berlin (2008)
10. Kolfschoten, G.L.: Theoretical Foundations for Collaboration Engineering. Delft University of Technology, The Netherlands (2007)
11. van der Raadt, B., Schouten, S., van Vliet, H.: Stakeholder Perspective of Enterprise Architecture. In: Morrison, R., Balasubramaniam, D., Falkner, K. (eds.) ECSA 2008. LNCS, vol. 5292, pp. 19–34. Springer, Heidelberg (2008)
12. Schrage, M.: Shared Minds. Random House, New York (1990)

# In Pursuit of IT Artifact Generality: The Case of Predictive Model for Electronic Negotiation Support

Rustam Vahidov[1], Réal André Carbonneau[2], and Gregory E. Kersten[1]

[1] Department of Decision Sciences & MIS, John Molson School of Business, Concordia University, 1455 de Maisonneuve Blvd W, Montréal, Québec, Canada, H3G 1M8
{rVahidov,Gregory}@jmsb.concordia.ca
[2] GERAD and Department of Management Sciences, HEC Montréal, 3000 chemin de la Cote-Sainte-Catherine, Montréal, Québec, Canada, H3T 2A7
contact@realcarbonneau.com

**Abstract.** Electronic Negotiation Systems (ENS) allow conducting negotiations by parties over the internet. When equipped with analytical tools they also provide means of decision support for the negotiators in analyzing the offers received and preparing new offers. One possibility to enhance the decision support capabilities in ENS is by providing a model for prediction of the next offer by a negotiator's counter-part. This paper describes the project aimed at building such a predictive model. The model had been built initially based on a specific negotiation case using the extensive database of past negotiations conducted through the "Inspire" ENS. The findings supported our anticipation of the effectiveness of offer prediction.

**Keywords:** Electronic Negotiations, Artifact Generality, Offer Prediction, Neural Networks.

## 1 Introduction

Design research must aim at both innovativeness and production of knowledge [1]. The second point implies that the artifact must be applicable to a variety of contexts. Undoubtedly, there is no clear-cut threshold of generality beyond which an artifact could be considered a product of research, rather than plain design. Rather, in many cases there could be a continuity ranging from a narrow context of the artifact through the increasing scope to its most generic applicability.

This paper presents an example of an IT artifact evolution from a concrete and narrow context, to a family of contexts. Specifically, the paper presents two consecutive research projects focusing on the development of a predictive model for electronic negotiations. A key factor influencing the course of negotiations is the ability of a negotiator to assess the possible reaction by the counterpart to a particular candidate offer. The predictive component that could become a part of the analytical toolbox may be of essential value in this respect. In particular, the component could attempt to predict a possible next move by a counterpart in response to the potential offer being considered by the negotiator. To evaluate the feasibility of building such a component we have considered the applicability a neural network-based model to the above problem.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 542–545, 2010.

We have used a large database of past negotiations collected by the Inspire ENS featuring offer exchanges in the context of a specific negotiation case (scenario). In the first project a neural network-based model has been trained using past negotiation instances to predict the opponent moves in the new unseen negotiations. In a subsequent project a new model has been proposed, which is applicable to a family of negotiation cases concerning issues for which the negotiators have similar preference structure.

## 2 Background

Electronic negotiation systems may feature support tools, which help the negotiators to structure his/her preferences, assess the candidate offers, examine the acceptability of the offers from the counterpart, and perform other analytical tasks. In particular, the tools may feature advanced intelligent capabilities, which could further inform the negotiation process, to the point (in a limited range of contexts) of complete automation. Despite some interesting findings related to negotiation automation, in most business negotiation contexts humans need to be involved in the process with the intelligent software possibly playing a supporting role. Research in this area focuses on solutions for assisting human negotiators. For example, in the eAgora negotiation system prototype an agent was included to provide recommendations to the user regarding the acceptability of an offer from an opponent, and construction and critique of the counter-offer [2].

The use of intelligent support tools for offer recommendation could be substantially improved if a possible impact of the candidate offer on the course of negotiations could be assessed. A model for predicting the opponent's next move in response to the offer being considered could also be used by human negotiators to perform "what-if" analysis. Here we set out to investigate the applicability of neural networks to learning the dynamics of negotiations from the past negotiation cases.

## 3 Neural Network-Based Predictive Model

The purpose of our initial project has been to demonstrate the feasibility and evaluate the effectiveness of utilizing a neural-network based model in predicting opponent's next offers during the course of negotiations [3]. At the output layer of a neural network each neuron is associated with one component of the offer, i.e. one negotiation issue. The number of output neurons is essentially the same as the number of the issues in negotiations. The inputs are associated with the past offers and counter-offers and the current trial offer. In general, the more of the history of a given negotiation is fed as inputs the more precise one would expect the predictions to be.

To demonstrate the feasibility and effectiveness of a the predictive model we have used past data collected by Inspire system [4]. Inspire is a web-based ENS, which permits two parties located anywhere in the world, who have internet access, to negotiate on a chosen case. The negotiation case under study features a scenario where a seller and a buyer want to enter a business negotiation. The issues include Price, Delivery, Payment and Returns. The neural model has 39 inputs and 4 outputs and the final dataset for the above required observations results in 6310 observations. In order to test the generalizability of the model we created two separate sets: 5048 (80%) of the observations have been assigned to the training set and 1262 (20%) of the observations have been kept in the testing set.

The total absolute error across all outputs of the testing set for one prediction is 1.68 on the 15 possible ordinal levels, which is approximately 11%. When the outputs were rounded to integers, the rounded error was 1.39 for the total of 15 possible ordinal levels which represented approximately a 9% error.

## 4 Increasing the Generality: The Pairwise Model

Our initial model has demonstrated the applicability of neural networks to opponent's next offer prediction. However, the model has a serious limitation as its inputs and outputs are fixed to the issues contained in the negotiation case. To build a generic model one has to realize the necessity of uncoupling particular issues from the model inputs. One way to achieve this is by focusing on the dynamics between different pairs of issues, rather than on particular issues themselves. Thus, to develop a more flexible and general predictive negotiation counteroffer modeling approach, a pairwise modeling of the negotiation issues is proposed, whereby the issue set in an offer is broken down into pairs.

The main idea of the approach is to predict a value for a given issue while pairing this issue with all the other ones. Subsequently, the average of the pairwise predictions for a target issue is taken as the predicted issue value. Because all pairs are grouped and modeled together and information identifying the individual issues themselves is removed, the pairwise model is only able to learn patterns that are common to all issue pairs, and not those specific to an issue. Thus, at the expense of more complex models and models specific to an issue, the pairwise model is more general and should generalize better across all issues, including predicting new issues, which were not involved in training the network. The actual neural network in this study has 23 inputs and 5 hidden layer neurons resulting in 126 weights.

The results indicate that the non-linear pairwise (NNP) model does not have a higher counteroffer prediction error (9.25%, $p$ = 0.00) than the original model (9.37%). In other words, the less powerful but more flexible pairwise model does not come at the expense of prediction performance. This may represent a lack of patterns that are specific to an issue or a lack of patterns involving interactions between more than two issues. On the other hand, it may be that some modeling accuracy of more complex or issue specific patterns are lost while at the same time offset by a better generalization because of an increased number of observations and/or because the data is confounded across issues.

We further investigated the generality and flexibility of the pairwise model. To this end we have trained it on a dataset while leaving one of the issues out of consideration. Then, we estimated its predictive accuracy for the issue that was previously unseen by the model. This was repeated four times, each time excluding an issue from the training and predicting this unseen issue. The results indicate that the pairwise (NNP) model with an issue omitted during training does not have a significantly higher counteroffer prediction error (9.46%, $p$ = 0.00) for this unseen issue than the pairwise model trained on all the issues (9.25%). Therefore, it suggests that the model could be expanded to other similar cases, as well as the new issues.

## 5 Conclusions

Design research is distinguished from the ordinary design, in particular, by the generality of the produced artifacts. These should be applicable to wider problem contexts, rather than to specifics of the particular settings. The projects described in this paper demonstrate the attempts to pursue higher-level generality of the artifacts. In predicting opponent moves during the negotiation process, our first attempt utilized a neural network-based model with its inputs and outputs strongly coupled with the issues involved in negotiations. In our second project we have proposed an alternative model, based on the principle of decoupling model inputs and outputs from the specific issues. The results suggested that the model displayed a certain level of generality while not compromising on the performance. Future work could be done to promote even higher level of generality. This would involve developing metrics to measure distances between different negotiation cases, collecting negotiation data for a set of dissimilar cases, and then training the neural network on such an expanded dataset.

## References

1. Vaishnavi, V.K., Kuechler Jr., W.: Design Science Research Methods: Innovating Information and Communication Technology. Auerbach Publications, Boca Raton (2008)
2. Chen, E., Kersten, G.E., Vahidov, R.: Agent-Supported Negotiations in E-Marketplace. International Journal of Electronic Business 3(1), 28–49 (2005)
3. Carbonneau, R., Kersten, G., Vahidov, R.: Predicting Opponent's Moves In Electronic Negotiations Using Neural Networks. Expert Systems with Applications 34(2), 1266–1273 (2008)
4. Kersten, G.E., Noronha, S.J.: WWW-based Negotiation Support: Design, Implementation, and Use. Decision Support Systems 25, 135–154 (1999)

# Use Cases for Business Metadata – A Viewpoint-Based Approach to Structuring and Prioritizing Business Needs

Daniel Stock, Felix Wortmann, and Jörg H. Mayer

Institut für Wirtschaftsinformatik, Müller-Friedberg-Straße 8, 9000 St. Gallen
{daniel.stock,felix.wortmann,joerg.mayer}@unisg.ch

**Abstract.** Business metadata plays a crucial role in increasing the data quality of information systems. Despite its importance, business metadata is primarily discussed from a technical perspective, while its business value is scarcely addressed. Therefore, this article aims at contributing to the further development of existing design approaches by explicitly accounting for the use cases of business metadata.

**Keywords:** Business metadata, requirements engineering, user acceptance.

## 1 Motivation

In recent years, "making better use of information" has gained importance and now ranks among the top five priorities of IT executives [1]. This trend is linked to the prevailing significance of Business Intelligence (BI) where data quality is a crucial factor for the perceived net benefits to the end-user [2]. In this context business metadata plays an important role in increasing data quality and therefore the acceptance of BI systems [3].

Despite its increasing relevance to practitioners [4], explicit discussions of the benefits of business metadata and the challenges of implementing respective solutions remain rare in academic literature [5]. This article thus contributes to a structured analysis of business metadata requirements by proposing a framework of potential use cases for business metadata. This framework can be utilized applying all the advantages inherent in a viewpoint-oriented requirements engineering approach [6] to the design of business metadata systems.

## 2 Conceptual Foundations

### 2.1 Business Metadata

"Metadata is data associated with objects which relieves their potential users of having full advance knowledge of their existence or characteristics" [7]. While business metadata is a sub-category of metadata that is used by the business side, technical metadata, in contrast, is used by IT. In literature 7 categories of business metadata can be distinguished [3]: (1) definitional, (2) data quality, (3) navigational, (4) process, (5) audit, (6) usage, and (7) annotations.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 546–549, 2010.

## 2.2 Viewpoint-Oriented Requirements Analysis

A viewpoint is a stakeholder and his or her task-related concerns (requirements) [6]. The idea of viewpoint-oriented requirements analysis is to singling out the concerns to better address the diversity of issues. The Viewpoint-Oriented Requirements Definition (VORD) approach is illustrated in Fig. 1.

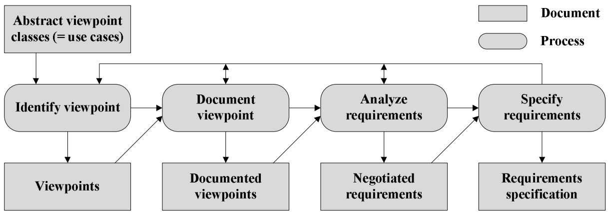



**Fig. 1.** Viewpoint-oriented requirements definition approach [6]

The definition of a viewpoint is intricately linked to the term ‘use case’. The greatest difference is the level of granularity. While a viewpoint describes one stakeholder only, a use case can describe the interaction of several stakeholders. Therefore, we will use the term ‘use case’ to refer to abstract viewpoint classes (see Fig. 1) which subsume stakeholders and their respective concerns.

# 3 Derivation of Use Cases for Business Metadata

In practice three groups of business stakeholders can be made out: data consumers, data managers, and data producers.

**Data consumer:** Each data consumer satisfies his data requirements through two general types of reports: standard and ad-hoc reports. While standard reports anticipate data requirements upfront and are tailored to a specific role, ad-hoc reports rather address spontaneous, role independent data requirements.

Therefore the use cases for standard reports can be categorized according to three different management levels in a business organization: corporate management (by the senior management), corporate performance management (by the middle management), and operational analytics (by the operational management).

Ad-hoc information needs can be either satisfied by each data consumer himself or through the help of dedicated knowledge workers. Therefore two additional use cases can be distinguished: ad-hoc information retrieval (by a data consumer himself) and advanced analytics (by a knowledge worker).

**Data manager:** The Data Management Body of Knowledge (DMBOK) names 10 functions of data management in its functional reference model. Since this framework is a comprehensive list of data management activities within an organization, not all of them lay in the responsibilities of the business side. Therefore we list all relevant activities that name a business role as operational responsible: data governance, data development, data security management, reference & master data management, and data quality management.

**Data producer:** Independently of a specific role, a data producer is responsible for data entry and maintenance. This comprises all activities that lead to the creation, update, or deletion of data. In total, eleven use cases could be identified (see Fig. 2).

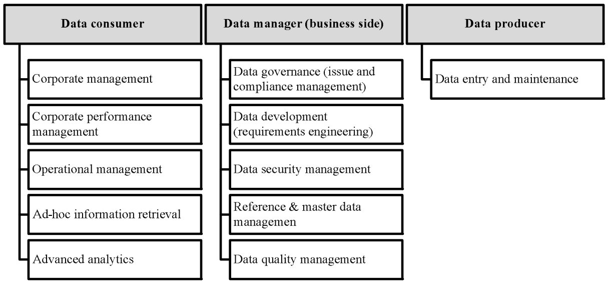



**Fig. 2.** Framework of use cases for business metadata

# 4 Demonstration and Evaluation

European Universal Bank's (EUB) first initiative on business metadata was confronted with much resistance from the business side as no obvious business need could be seen. In order to make the next advance of business metadata a success, EUB was seeking for a more business-driven approach. Therefore, we applied our proposed use case framework in accordance with the presented VORD approach. First, EUB prioritized the use cases. Second, we exemplary defined and documented the viewpoints.

The surveyed employees of EUB named 'data development' as the most relevant for EUB and therefore prime candidate for exemplary viewpoint documentation. In a collaborative workshop of business and IT, the group identified five relevant stakeholders and seven activities, which benefit from business metadata.

**Table 1.** Requirements per activity in use case 'data development' (excerpt)

| Activities per stakeholder | Definitional | Data quality | Navigational | Process | Audit | Usage | Annotations |
|---|---|---|---|---|---|---|---|
| Business units | | | | | | | |
| • Specifying change request | √ | | √ | | √ | | √ |
| • Monitoring implementation | | √ | | | | | |
| … | | | | | | | |

For each activity the workshop participants were asked to assess the usefulness of the seven business metadata categories listed in section 2.1 (see Table 1).

Subsequently we evaluated the demonstrated approach with the participants of the workshop. In a focus group we examined design and utility of the proposed model.

**Design:** The participants challenged the features 'completeness' and 'level of detail' of the model. In order to complement the framework they suggested including a use case 'risk management'. Since risk management is a function which holds special significance in financial institutions, including it as an additional use case would compromise on the frameworks generality. Concerning the second point, further investigation is necessary as to whether the use case 'advanced analytics' is too abstract because it subsumes too many distinct functions or whether the notation is not clear enough.

**Utility:** The participants were generally very satisfied with the utility of the framework and the associated viewpoint-oriented approach. They particularly appreciated that the discussions are concentrated on a business perspective rather than exploring technical details. Even though this is all in all a very positive evaluation result, it does lack representativeness. The participants of the focus group were all from the same company and thus this framework needs further evaluation in different contexts.

## 5 Concluding Remarks

This article has derived a conceptual framework for business metadata use cases. Its applicability was demonstrated in a banking case and subsequently evaluated within a focus group. The evaluation is as yet not representative and must be subjected to future research. Especially the use case 'advanced analytics' needs further investigation in order to comply with the design requirement 'level of detail'.

## References

1. Luftman, J.N., Kempaiah, R., Rigoni, E.H.: Key Issues for IT Executives 2008. MISQ Executive 8, 151–159 (2009)
2. Wixom, B.H., Watson, H.J.: An Empirical Investigation of the Factors Affecting Data Warehousing Success. MIS Quarterly 25, 17–41 (2001)
3. Foshay, N., Mukherjee, A., Taylor, A.: Does Data Warehouse End-User Metadata Add Value? Communications of the ACM 50, 70–77 (2007)
4. Schulze, K.-D., Besbak, U., Dinter, B., Overmeyer, A., Schulz-Sacharow, C., Stenzel, E.: Business Intelligence-Studie 2009. Steria Mummert Consulting AG, Hamburg (2009)
5. Shankar, G., Even, A.: Managing Metadata in Data Warehouses: Pitfalls and Possibilities. Communications of the Association for Information Systems 14, 247–274 (2004)
6. Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. John Wiley & Sons Ltd., New York (1998)
7. Dempsey, L., Heery, R.: Metadata: A Current View of Practice and Issues. Journal of Documentation 54, 145–172 (1998)

# The IT-CMF: A Practical Application of Design Science

Brian Donnellan[1] and Markus Helfert[2]

[1] National University of Ireland,
MaynoothCounty Kildare, Ireland
brian.donnellan@nuim.ie
[2] Dublin City University,
Dublin, Ireland
markus.helfert@computing.dcu.ie

**Abstract.** The IT-Capability Maturity Model [IT-CMF] is a high-level process capability maturity framework for managing the IT function within an organization. The purpose of this paper is to explore and explain the IT-CMF as a " method meta-model" for IT management, emphasizing the novel approach to addressing the application of design processes and design artifacts by means of a very structured use of engaged scholarship and open innovation techniques to the ongoing challenge of managing organization's IT capability.

## 1 Introduction

The research reported in this paper has been developed in the context of the IT-CMF, which presents a high-level process capability maturity framework for managing the IT function within an organization [1], [2], and [3]. The framework identifies a number of critical IT processes, and describes an approach to designing maturity frameworks for each process. The IT-CMF addresses a continuing structural problem in the IT profession and IT industry around managing the returns from IT investments.

Both "method engineering" and "method construction" can be seen as elements of Design Science-oriented information systems research [12]. The purpose of this paper is to explore and explain the IT-CMF as a "method meta-model" for IT management. There has been relatively little published research addressing the *practical* application of design processes and design artifacts in Information Systems Management. This paper addresses this paucity of published research and introduces an innovative solution to the ongoing challenge of managing the returns from IT investments.

## 2 The Application of DSR Principles in the IT-CMF

Design Science creates and evaluates IT artifacts intended to solve identified organizational problems. Such artifacts are represented in a structured form that may vary from software, formal logic, and rigorous mathematics to informal natural language descriptions. The rich phenomena that emerge from the interaction of people, organizations, and technology may need to be qualitatively assessed to yield an understanding of the phenomena adequate for theory development or problem solving [5].

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 550–553, 2010.

As field studies enable behavioral-science researchers to understand organizational phenomena in context, the process of constructing and exercising innovative IT artifacts enable design-science researchers to understand the problem addressed by the artifact and the feasibility of their approach to its solution [8].

Developing innovative artifacts is a central activity in DSR [4]. Such artifacts can be in the form of constructs, models, methods or instantiations [5]. For the construction of such artifacts two basic activities can be differentiated: build and evaluate where building "is the process of constructing an artifact for a specific purpose" and evaluation "is the process of determining how well the artifact performs" [5, p. 254]. The construction of an artifact is a heuristic search process [5]. Within this process an extensive use of theoretical contributions and research methodologies stored in the knowledge base should be made [5]. On the one hand theoretical contributions can come from governance, value based management, risk management, compliance management, etc. to build an artifact, i.e. the situational method. The IT-CMF uses the following DSR patterns proposed in [6]:

- *Different Perspectives:* The research problem is examined from different perspectives, e.g. conceptual, strategic, organizational, technical and cultural.
- *Interdisciplinary Solution Extrapolation:* A solution or solution approach (i.e. methods, instructions, guidelines, etc.) to a problem in one discipline can be applied in or adapted to the integrated IT CMF.
- *Building Blocks:* The complex research problem of IT Management is broken into thirty six critical processes that are examined in turn.
- *Combining Partial Solutions:* The partial solutions from the building blocks are integrated into the overall IT CMF and the inter-dependencies between the building blocks are identified and high-lighted. In order to rigorously demonstrate the utility of the developed artifact, different evaluation methods can be used. Amongst others, the "informed argument" is suggested as an appropriate evaluation method [5].

Maturity models in design oriented research are regarded as being located between models and methods in the form of state descriptions (e.g. the maturity levels) and guidelines [12]. In this sense, maturity models contain two aspects, one capturing the assessment of the current status and another one guiding organizations towards higher maturity levels. In the context of Design Science the first aspect can be described as a model perspective describing various maturity levels (states) of organizations whereas the second aspect describes guidelines to improve the current situation of organizations in form of method components [12]. In order to transform organizations from one maturity level to another, usually the method component is described by "maturity curves" or "maturity profiles".

In our work, we recognize this dual perspective on maturity models and aim to represent both perspectives in the meta-model of the IT-CMF. We also extend the traditional perspective of generic maturity models, in the form of providing guidelines to contextualize maturity models. As such, we combine the recent work on model and method contextualization with our work on maturity models. In order to develop an approach to contextualize the IT-CMF, we follow DSR and apply a method engineering (ME) approach. Recognizing the two perspectives of maturity models - model and methods - we developed a meta-model for the IT-CMF.

Essential to IT-CMF are Critical Processes (CPs) that represent an IT management process. CPs are central to the IT organization and are defined for a particular domain within it. The IT-CMF contains 36 CPs, which are categorized in four macro processes within a high-level overarching process. A CP takes inputs from and provides output to other CPs. Specific characteristics of a CP are further described by Capability Building Blocks (CBB) that are self-contained and completely exhaustive aspects describing the management aspects of a CP.

The IT-CMF Content Development and Review Process is implemented by the IT-CMF development community in the Innovation Value Institute (www.ivi.ie). This community is comprised of university-based academic researchers and industry-based practitioner-researchers drawn from over 40 companies located throughout the world. The IT-CMF development and review processes are based on “engaged scholarship” [13] and “open innovation” principles [14].

Associated with each CP in the IT-CMF is a maturity profile referring to the level of value and assessment elements describing maturity indicators, assessment approaches and metrics. The set of maturity profiles assigned to a CP details the transformation from one maturity level to another. The IT-CMF also contains a defined set of templates describing critical process.

The IT-CMF maturity model (an instance of the IT-CMF meta-model) is then applied to various organizations presenting several distinguished organizational contexts. The organizational context can be differentiated by various factors, for instance; organizational size, sector, coordination form within the organization, decision making structures, organizational structure, communication structure, type of information systems used, task structure, automation level and many more. Generally most maturity models do not explicitly cater for this context adoption, and usually only provide some form of guidelines or best practice. During the application, researchers or consultants adapt the maturity model and select or parameterize certain aspects of the model.

Within the IT-CMF we recognize this contextualization challenge and aim to provide specific guidelines for adapting or configuring the maturity model to a specific problem. In the context of method engineering, several approaches have been suggested to consider situational and contextual factors [9], [10]; [11]. Furthermore Recker et al. [15] suggested an approach contextualizing models in order to facilitate the adaption to specific application contexts. The aim is to produce valuable organizational designs (in form of methods or models) for a set of situations by considering situations in form of contextual influencing factors of the firm. In this sense, we complement the IT-CMF maturity model by providing a process for contextualizing the maturity models. The process configures some element of the IT-CMF meta-model. Thus, the IT-CMF research is a timely response to the malaise in the IS discipline i.e. the tension between rigour and relevance in IS research.

## 3 Conclusions

In contrast to many other maturity models, the IT-CMF maturity model is based on DSR principles. Following a method engineering approach we have presented a development and review process. We have reviewed principles of Design Science and

Method Engineering and applied these principles to the development of an IT-Capability Maturity Framework. Following a rigorous Design Science-based development process we presented a meta-model for describing the IT-CMF maturity model together with a supportive contextualization process. Our work demonstrates the benefits of applying design science principles in a practical setting. By applying a rigorous development process together with a consistent meta-model it can help to improve model and methods as results of design science research.

## References

1. Curley, M.: Managing Information Technology for Business Value. Intel Press (January 2004)
2. Curley, M.: A Value Based IT Capability Maturity Framework. Intel EMEA Academic Forum, Ireland (2006)
3. Curley, M.: IT Innovation, a new Era. In: Proceedings of the International Conference of Computational Science, Reading, UK (2006)
4. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
5. Schön, D.A.: The Reflective Practitioner: How Professionals Think in Action. Basic Books, New York (1983)
6. Vaishnavi, V., Kuechler, W.: Design Research in Information Systems (January 20, 2004)
7. Klein, H.K., Myers, M.: Evaluating Interpretive Field Studies. MIS Quarterly 23(1) (March 1999)
8. Nunamaker, J., et al.: Electronic Meeting Systems to Support Group Work. Communications of the ACM 34(7), 40–61 (1991)
9. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38(4), 275–280 (1996)
10. Gericke, et al.: Situational Method Engineering for Governance, Risk and Compliance Information Systems. In: DESRIST 2009, Malvern, PA, May 7–8 (2009)
11. Bucher, T., et al.: Situational Method Engineering – On the Differentiation of “Context” and “Project Type”. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) IFIP WG8.1 Working Conference on Situational Method Engineering (ME 2007), Geneva, pp. 33–48 (2007)
12. Mettler, T., Rohner, P.: Situational Maturity Models as Instrumental Artifacts for Organizational Design. In: DESRIST 2009, Malvern, PA, May 7–8 (2009)
13. Van de Ven, A.H.: Engaged Scholarship: A Guide for Organizational and Social Research. Oxford University Press, Oxford (2007)
14. Chesbrough, H.: Open Business Models: How to Thrive in the New Innovation Landscape. Harvard Business School Press, Boston (2006)
15. Recker, J., et al.: Model-Driven Enterprise Systems Configuration. In: 18th Conference on Advanced Information Systems Engineering. Springer, Luxembourg (2006)

# Ontology Design for Strategies to Metrics Mapping

Fatemeh "Mariam" Zahedi and Atish P. Sinha

Lubar School of Business, University of Wisconsin-Milwaukee, Milwaukee, WI 53201, USA
{zahedi,sinha}@uwm.edu

**Abstract.** With the growth of enterprise resource planning technologies and automatic data collection, businesses are flooded with data. But the wealth of data has not translated into the knowledge required to articulate strategies at the top levels of the decision hierarchy, because of the lack of connectivity between business strategies and the underlying performance data. This has deprived top-level executives of the ability to connect enterprise strategies to objectives and performance metrics in a systematic fashion. In this paper, we describe the design of an ontology that captures and connects a comprehensive set of business strategies, objectives, and performance metrics by grounding our work in design-science research. It reflects the involvement of a large multi-national manufacturing company and a high-tech provider of services. The ontology we have designed enables the development of strategy support systems for tracking the implementation and assessment of business strategies.

**Keywords:** Design science, ontology, OWL, Protégé, strategies, objectives.

## 1 Introduction

The exponentially increasing computational capabilities have resulted in massive collection of data at various levels of detail. With the growth of enterprise resource planning (ERP) technologies and the automatic collection of transactional data, businesses are now flooded with data. The wealth of available data has rarely been translated into knowledge required to articulate strategies at the top levels of the decision hierarchy, and to alert executives to the need for corrective strategies once the feedback process signals such needs. In this study, we focus on providing a formal and comprehensive connection between business strategies and metrics. We therefore pose the following research question: *How can business strategies be identified, developed, and monitored using metrics from the available data?* In dealing with this question, we propose a framework, which involves developing a comprehensive strategies-to-metrics ontology. The ontology links the performance metrics to enterprise strategies. It could also be used as the basis for regulating and automating the data mining process, as well as the design and implementation of an enterprise strategy support system.

## 2 Strategies-to-Metrics Ontology

In philosophy, ontology is defined as "what exists." With the popularity of the Semantic Web, ontologies for reasoning with machine-readable web-based knowledge

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 554–557, 2010.

have gained acceptance. Ontology is defined as "a formal explicit specification of a shared conceptualization for a domain of interest" ([4], p. vii). To gain a competitive advantage, a firm has to formulate strategies that entail performing different activities from its competitors or performing similar activities in different ways. In order to assist top executives in making decision on strategies and assess their success, it is essential to have a comprehensive ontological knowledge base for connecting the performance metrics to enterprise strategies. Fig. 1 shows the framework for an ontology that connects strategies to performance metrics.

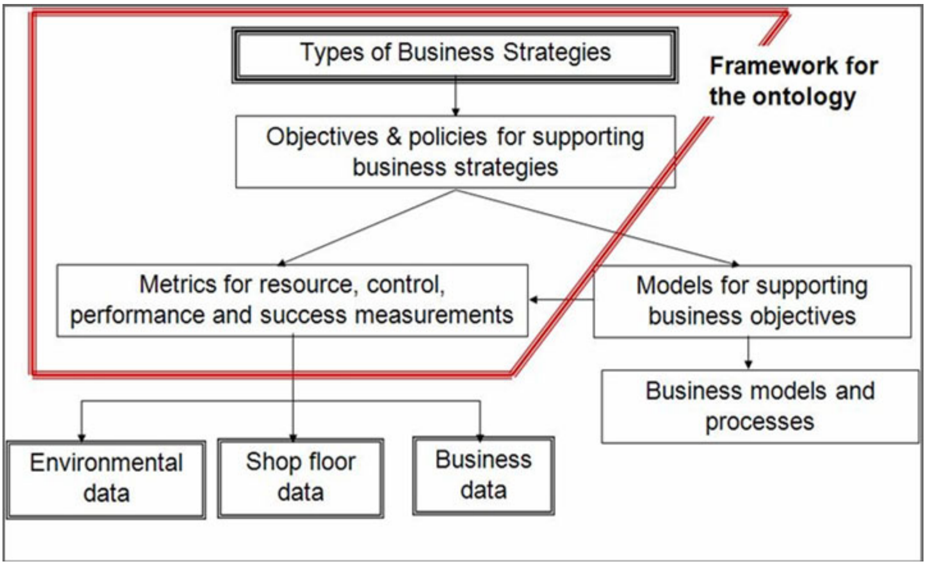



**Fig. 1.** The Framework for the Strategies-to-Metrics Ontology

We used Grounded Theory [1] and design-science research [2] for generating the ontology. Since ontology involves qualitative knowledge, it is common to use qualitative methods to collect and analyze the data (see, for example [3], [5]). The strategies-to-metrics ontology formalizes the mission-critical decisions and actions by top executives, and should provide unambiguous semantics to communicate strategies and objectives to various units, branches, and levels within the enterprise. We collected data from two organizations: one a large multi-national manufacturing company and the other a high-tech service provider. We implemented the ontology using OWL and Protégé. Table 1 summarizes how we applied design-science research principles for developing the ontology.

The completed ontology has a hierarchical structure for strategies. The first level of strategies has 12 distinct types of strategies, each of which has subcategories (see Fig. 2 for first-level strategy classes). There are 231 subcategories of strategies, some as deep as five levels. The ontology has 17 distinct objective classes and 461 subcategories of objectives, some with five levels in the hierarchy of objectives. Each strategy is linked to a number of objectives. An objective could be linked to more than one strategy. There are 405 links between strategies and objectives. Performance metrics were identified for

**Table 1.** Applied Design Science Principles based on Hevner et al. [2]

1. **Design an artifact.**
   Viable artifacts=The ontology implemented in Protégé 3.4
   a. Constructs=Strategies, objectives, metrics
   b. Model: Fig. 1 (additional models were removed due to space limitations)
   c. Instantiation: manufacturing and services
2. **Relevance.** Relevance will be the motivation, direct link between strategy articulations with measures, linking ontology to conceptual schemas.
3. **Evaluation method.** Multiple methods are used in the evaluation process:
   a. Multiple iterations with the expert executive and the president
   b. Used manufacturing ontology to guide the development of the service ontology
   c. Final review presentations for both companies (president in one and top executives in the other), funded by one, supported by both
4. **Research contributions.** This fills the gap articulated both in the literature and by the industry, the ontology development approach is also a contribution, comparative analysis of strategies and objectives in two sectors (manufacturing and service) is another contribution, the artifact is a third contribution
5. **Research rigor.** Synthesis of Grounded Theory and OWL
6. **Design as a search process.** Grounded theory, constant comparison, and categorization bring out the design as a search process, the comparative analysis of manufacturing and service is another aspect of search. Knowledge acquisition was conducted using interviews, document analysis, and literature review.
7. **Communication of research.** Presentation of the final artifact to the President of one company and top executives of another.

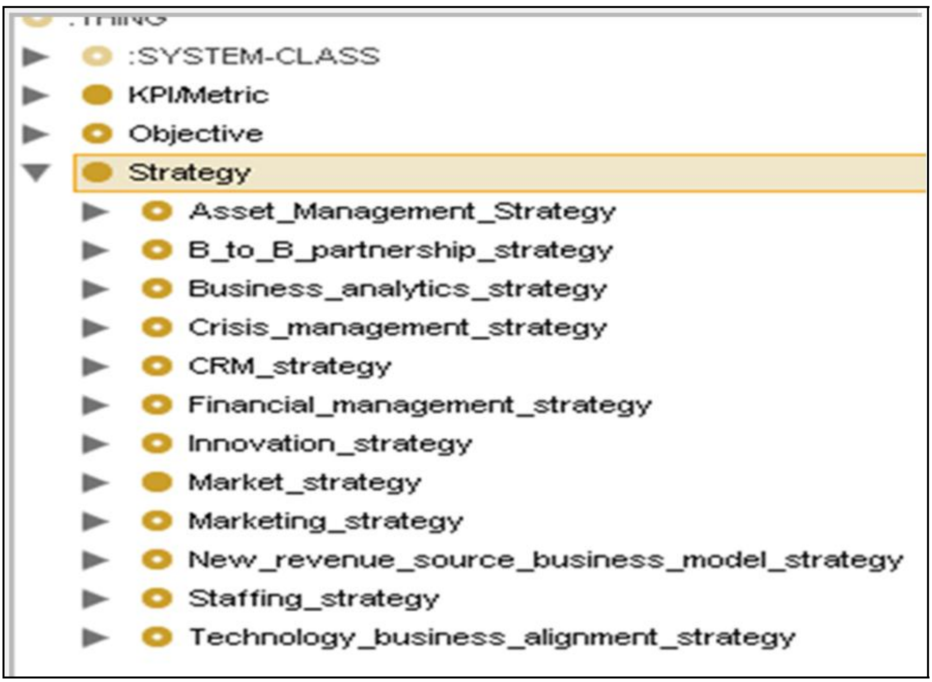



**Fig. 2.** Top Level Strategy Classes

a select number of objectives (16 first-level and 176 subcategories). A complete specification of all metrics for the evaluation of objectives required access to company data repositories—it is not available to outsiders in most companies.

To the best of our knowledge, the strategies-to-metrics ontology is the first ontology linking business strategies to objectives to performance metrics. The ontology is based on the deep knowledge of top executives involved in the development and implementation of enterprise strategies. We followed a rigorous process as prescribed by the design science approach. The ontology can significantly contribute to the support of executives and can enable them to prioritize various objectives based on their roles in enterprise strategies. This ontology can also improve communications across various units of an enterprise, and can enable the development of intelligent support systems for tracking the implementation and assessment of business strategies.

**Acknowledgments.** This work was supported by a Rockwell Catalyst Grant from the UWM Research Foundation. We would like to thank Tim Biernat and John Ische for their active participation in this project.

## References

1. Glaser, B.G., Strauss, A.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldined Publishing Co., Chicago (1967)
2. Hevner, A.R., March, S.T., Park, J.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
3. Mosca, A., Palmonari, M., Srtori, F.: An Upper-level Functional Design Ontology to Support Knowledge Management in MSE-based e-Manufacturing of Mechanical Products. The Knowledge Management Review 24, 265–285 (2009)
4. Staab, S., Studer, R. (eds.): Handbook on Ontologies. Springer, Heidelberg (2004)
5. Wales, R.C., Shalin, V.L., Bass, D.S.: Requesting Distant Robotic Action: An Ontology for Naming and Action Identification for Planning on the Mars Exploration Rover Mission. Journal of the Association for Information Systems 8, 75–104 (2007)

# Naturalistic and Artificial Evaluations of Personas and Role-Based Enterprise Systems

Sven Carlsson and Björn Johansson

Department of Informatics, School of Economics and Management, Lund University,
Ole Römers väg 6, SE-223 63 Lund, Sweden
{sven.carlsson,bjorn.johansson}@ics.lu.se

**Abstract.** Scholars in design science research in Information Systems and Technology (IS&T) argue that evaluation of developed and created artifacts is critical. We describe the evaluations of: 1) the use of personas in the development of ERPs (naturalistic evaluation), and 2) from an individual and organizational effectiveness view what a role-based ERP supports (artificial evaluation).

**Keywords:** Design science research, evaluation, enterprise resource planning, persona, role-based.

## 1 Introduction

The aim of design science research in Information Systems and Technology (IS&T) is to produce viable artifacts [1], to develop information systems design theories [2, 3], as well as to develop information systems design knowledge [4-6]. One important activity is evaluation of created and developed artifacts, design theories and design knowledge. A distinction can be made between naturalistic and artificial evaluation [6]. We present two evaluations related to ERPs: 1) use of personas in ERP development (naturalistic evaluation, Sect. 2), and 2) from an individual and organizational effectiveness view, what is a role-based ERP supporting (artificial evaluation, Sect. 3). The aim of the paper is twofold: 1) to show how evaluation can be done in design science research in IS&T, and 2) show the result from two evaluation studies.

## 2 Developing ERP Systems Using Personas

Microsoft Dynamics (MS) has adapted Personas for development of future ERPs. The approach builds on the "Persona idea". It presents detailed abstractions of fictitious (or real) users of a software product [7]. MS's basic assumption was that personas provide better understanding of potential users. MS's general template for personas contains demographic, work related and psychographic data [8]. All persona information is stored on the persona website available for developers at MS. When developers get work tasks they should directly or indirectly relate them to specific personas, and also be inspired by the personas. A specific persona is continuously enhanced. The persona website and the customer model should provide developers with information so that they better understand for whom they develop and how they should develop a

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 558–562, 2010.

specific functionality. The persona website contains in addition to the persona specification also information about development of that persona and interviews and observations used to create the persona. All this persona-related information has been used for building the "MS Customer Model". It consists of three types of entities: People, Departments, and Work. Relations between the three types are depicted in two parts. "People and Departments" describes general organizational structure of departments, typical roles of people in the departments by means of personas, and that each and every persona belongs to a specific department. "Departments and Work" consists of departments and typical business processes in these departments. There are 32 different business processes, divided into activities, which are divided into tasks that are carried out by various roles (personas). There are 153 activities presented in the departments and work poster, while tasks are not described at all, neither are they described in an extensive way at the persona website.

Personas are presented to developers in several ways. The most comprehensive presentation is the persona website, also holding the customer model, which is a collection of 67 personas that together form the staff of a model-company. The customer model also holds a collection of fixed processes that make the tasks performed in the model company. This collection of processes is called the work model. Each persona in the customer model has direct connections to one or more processes in the work model. Two other main persona artifacts are the work model poster and the persona poster, both displayed to a high extent in corridors, cafés, and offices across MS's development sites.

Based on an analysis of data (interviews, observations and documents) collected at MS a set of evaluation findings emerged.

First, the way persona is used in MS differs from the initial ideas as presented, for example, by Nielsen [9]. She describes persona as primarily a way of understanding end-users' needs. At MS personas are also used between different ERP-developers for sharing knowledge and experience of requirements. Hence, personas are used as a communication tool and as boundary objects.

Second, MS's personas usage differs also in the number of personas used. Both Cooper [7] and Nielsen [9] say that the number of personas should be fairly restrictive with a maximum of five to six personas. In MS the number is considerable higher. The developers do not see this as a problem. However, there are indications that there exist problems with coordination between development groups. It could be questioned if and in what way the personas approach supports coordination between different development groups and teams.

Third, the original idea of using personas was to move away from a task-directed design and development perspective. MS's personas approach focuses on both tasks and goals that users want to be fulfilled. From the personas descriptions it can be concluded that the MS personas approach is a combination of task-directed and goal-oriented, since the personas descriptions focus on both goals and core activities that the users want to be fulfilled.

## 3 Role-Based ERP Systems

The second evaluation is an artificial evaluation and shows a way for evaluating a large and complex role-based ERP system, namely mySAPWP (mySAP Workplace).

In designing an organization-specific Information Portal based on mySAPWP's role-based concept, an organization can create its own roles or use the templates for single roles and composite roles supplied by SAP. Using the templates (“best practice”) an organization modifies these to suit the organization’s requirements. The generic roles, which control access to applications, information, and services, are dived into different functional categories (cross-industry templates) and different industry solutions that SAP supports.

### 3.1 Evaluation Model and Evaluation of mySAP Workplace

In evaluating an artifact some criteria are used and some measurements performed. In general, questions like ”How effective and efficient is this artifact” are asked and answered. In artifact-evaluation one can use a model to evaluate effectiveness and efficiency of an artifact. In order to evaluate mySAPWP we had to develop an evaluation model. The model should be possible to use for evaluating other role-based ERP systems. We chose Quinn and associates' competing values model (CVM) [10-12] as the underpinning theory for our model. The CVM is a framework and model of organizational effectiveness [11]. It acknowledges the existence of simultaneously and conflicting goals, which an organization must attain in order to be effective [13]. In an organization there is a tension between three existing underlying values: 1) focus where internal focus puts emphasis on well being in the organization while external focus puts emphasis on the environment, 2) structure where stability refers to the need of top management to control and flexibility refers to adaptation and change, and 3) ends versus means in effectiveness criteria [10-11]. The values reflect similarities to four organizational models with respect to different constructs of organizational effectiveness. Quinn [12, 14] translated the construct of effectiveness into managerial roles. The four models and eight roles are: human relations model (HR) with its facilitator and mentor roles; open systems model (OS) with its innovator and broker roles; internal process model (IP) with its monitor and coordinator roles; and rational goal model (RG) with its director and producer roles.

We decided to evaluate the "whole package" of mySAPWP roles, using SAP’s web pages to find different roles and descriptions of the roles. In the version we evaluated, we found 433 individually labeled roles. From the 433 roles, we excluded all roles associated with the industry solution of SAP Healthcare and removed doublets and non-classifiable roles and ended up with 329 roles. The evaluation of the 329 roles was done in a four-step process: 1) listing of the 329 roles, 2) categorization of the roles using the value dimensions, 3) mapping of the roles to CVM, and 4) mapping of the roles to hierarchical levels. Two researchers did steps 2-4 independently (intercoder reliability was 80%). Table 1 shows what organizational models, with their associated roles, as well as what organizational levels are supported by mySAPWP’s roles.

Studies within the CVM framework suggest that all models and roles are not equally important and critical. There are changes in the importance of the models and roles in relation to hierarchical levels as well as what state a firm is in.

**Table 1.** Evaluation of mySAP Workplace's role based concept

| CVM | Top | Middle | Operative | Experts | External | Total |
|---|---|---|---|---|---|---|
| Internal Process | 3 | 77 | 83 | 21 | 0 | **184** |
| Rational Goal | 4 | 33 | 46 | 23 | 8 | **114** |
| Human Relations | 1 | 2 | 2 | 2 | 0 | **7** |
| Open Systems | 5 | 8 | 2 | 9 | 0 | **24** |
| **Total** | **13** | **120** | **133** | **55** | **8** | **329** |

Quinn and Cameron [15] found four different states a firm can be in: 1) entrepreneurial, 2) collectivity, 3) formalization and control, and 4) elaboration of structure state. In the entrepreneurial state the roles in the OS model are the critical roles and in the collectivity state the roles in the HR model are the critical roles. In the formalization and control state the roles in the IP and RG models are the critical roles. The elaboration of structure state has a more balanced emphasis of roles. Based on Quinn and Cameron's findings, we can hypothesize that mySAPWP will be more effective in firms in the two latter states.

In another study it was found that there is also a difference in the importance of the roles in relation to hierarchical levels [14]. Two major findings in the study were that: 1) there exists an equal emphasis for the monitor (IP), coordinator (IP), and director (RG) role, and 2) the importance of the two OS-roles increases as we move up the hierarchical levels. Although, mySAPWP supports the IP and RG roles, it does so better for middle and lower level managers than for top-managers. Our study suggests that mySAPWP should be developed to better support top-managers by better supporting the OS roles. Another improvement, important to all levels, would be to enhance the support of the HR roles.

## 4 Conclusions and Future Research

We presented a naturalistic evaluation, using a case-based method, of the use of a personas-based approach for developing ERP systems. The evaluation illustrated the usefulness of the personas-based approach, but it also gave indications on improvements if the approach should fulfill its goals. We also showed how an artificial evaluation of a role-based ERP system could be done. This evaluation shows how to use an underpinning theory for developing an evaluation model and the likely results from using such an evaluation model. Future research will include: 1) evaluation of different role-based and personas-based systems, 2) studies on the personas-based approach, and 3) improvement of the two evaluation methods.

## 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. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems 8(5), 312–335 (2007)

3. 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 (2004)
4. Carlsson, S.A.: Developing Knowledge through IS Design Science Research: For Whom, What Type of Knowledge, and How. Scandinavian Journal of Information Systems 19(2), 75–85 (2007)
5. McKay, J., Marshall, P.: Foundation of Design Science in Information Systems. Sparks working paper, RISO (Research into Information Systems in Organisations), Faculty of Information and Communication Technologies, Swinburne University of Technology, Melbourne, Australia (2008)
6. Venable, J.: A Framework for Design Science Research Activities. In: Proceedings of the 2006 Information Resource Management Association Conference, Washington, DC, USA (2006)
7. Cooper, A.: The Inmates are Running the Asylum: Why High-tech Products Drive Us Crazy and How to Restore the Sanity. Sams Publishing, Indianapolis (1999)
8. Microsoft Dynamics: Roles-based Business Productivity Software Designed for Your People, http://www.microsoft.com/dynamics
9. Nielsen, L.: Engaging Personas and Narrative Scenarios: A Study How a User-centered Approach Influenced the Perception of the Design Process in the e-Business Group at AstraZeneca. Copenhagen Business School, Frederiksberg (2004)
10. Quinn, R.E., Faerman, S.R., Thompson, M.P., McGrath, M.R.: Becoming a Master Manager. John Wiley & Sons, New York (1996)
11. Buenger, V., Daft, R.L., Conlon, E.J., Austin, J.: Competing Values in Organizations: Contextual Influences and Structural Consequences. Organization Science 7(5), 557–576 (1996)
12. Quinn, R.E., Rohrbaugh, J.: A Competing Values Approach to Organizational Effectiveness. Public Productivity Review 5(2), 122–140 (1981)
13. Hart, S.L., Quinn, R.E.: Roles Executives Play: CEOs, Behavioral Complexity, and Firm Performance. Human Relations 46, 543–574 (1993)
14. Quinn, R.E.: Beyond Rational Management: Mastering the Paradoxes and Competing Demands of High Performance. Jossey-Bass, San Francisco (1988)
15. Quinn, R.E., Cameron, K.S.: Organizational Life Cycles and Shifting Criteria and effectiveness. Management Science 9(1), 33–51 (1983)

# Author Index