A Pattern-Based Approach for Constructing Design Theories with Conceptual Models

Michael Schermann
Santa Clara University, michael.schermann@in.tum.de

Tilo Boehmann
Technische Universitat Munchen, boehmann@in.tum.de

Helmut Krcmar
Technical University Munich, krcmar@in.tum.de

Follow this and additional works at: http://aisel.aisnet.org/ecis2007

Recommended Citation
http://aisel.aisnet.org/ecis2007/181
A PATTERN-BASED APPROACH FOR CONSTRUCTING DESIGN THEORIES WITH CONCEPTUAL MODELS

Michael Schermann, Chair for Information Systems, Technische Universität München, Boltzmannstrasse 3, 85748 Garching, michael.schermann@in.tum.de

Tilo Böhmann, Chair for Information Systems, Technische Universität München, Boltzmannstrasse 3, 85748 Garching, boehmann@in.tum.de

Helmut Krcmar, Chair for Information Systems, Technische Universität München, Boltzmannstrasse 3, 85748 Garching, krcmar@in.tum.de

Abstract

In this paper we outline a new approach for constructing design theories in IS research by formulating theoretical statements as conceptual models. In recent history the philosophical and methodological aspects of design research have been in the focus of many research groups in IS research. We argue that applying conceptual models in the theory-building process enables design researchers to express hypotheses and underlying assumptions more accurately. Furthermore, our approach allows researchers to specify empirically refutable statements. We ground our work in the concept of IS design theories as proposed by Walls et al. (1992) and extend their idea twofold: first, based on an analysis of the output types of design research as proposed by March & Smith (1995) we incorporate conceptual modeling to formulate theory statements. Second, to facilitate developing concise testable theory statements, we apply the idea of patterns as proposed by Alexander (1973). Overall, we propose a detailed framework that integrates conceptual modeling in the process of theorizing in design-oriented IS research. Thus, we present an important step towards building “own” theories of IS research.

Keywords: design research, design science, IS design theories, patterns, conceptual modeling.
1 INTRODUCTION

The quest for theoretical foundations in Information Systems Research is as old as the discipline itself. For instance, Weber criticized the lack of theoretical orientation in IS research (Weber, 1987). He argues that IS researchers have not yet agreed on paradigms. Paradigms provide a commonly shared goal of a certain scientific community (Kuhn, 1998). In line with Kuhn, he argues that theories are at the center of scientific paradigms (Weber, 1987). Subsequently, Weber demands that IS research should focus on the development and rigorous testing of theories. Weber identifies three main areas for improvement – among them the scientific process of designing and implementing artifacts (Weber, 1987). Furthermore, Weber argues that IS researchers have to develop own paradigms and thus own theories. In the same line, many researchers have expressed the importance of design-oriented research for the progress of IS research (e.g. March & Smith, 1995; Hevner et al., 2004; Nunamaker & Chen, 1991). For instance, Orlikowski and Iacono (2001) argue that the focus of IS research should be the IT artifact and its performance in practice. Thus, we argue that design theories may be the ideal building ground for paradigms in IS research. To facilitate constructing design theories we develop a framework to integrate conceptual models in the development process of design theories.

While the word theory is often used, little agreement can be found when it comes to the structure and components of theories (Gregor, 2006). According to Popper, a theory is a set of statements claiming universal validity (e.g. all ravens are black). Universal statements can then be transformed to prognoses within a context (e.g. the raven in the yard should be black). Following Gregor (2006), theories in design research express prescriptive statements. However, little is said e.g. about how such statements about design should be expressed to facilitate the testing and improvement of design theories.

Thus, we analyze the concept of design theories as proposed by Walls et al. (1992) to answer our first research question: What is the structure of an IS design theory? In the course of answering this question we will develop a first draft of a meta-model that is specifying the elements of an IS design theory.

The overall goal of IS design research is to design and implement new artifacts, e.g. new information systems, that facilitate attaining human goals more efficiently or more effectively (Simon, 1969). March and Smith (1995) argue that design research produces four types of output: constructs, models, methods, and instantiations. In this paper we argue that models are the primary theoretical output of design research. When using the term model we refer to conceptual models, i.e. “representation[s] of selected phenomena in some domain” (Wand & Weber, 2002 p 363). Furthermore, we argue that conceptual models, often constructed in a semi-formal graphical language, facilitate a higher degree of non-ambiguity in expressing the statements of a theory. Hence, our second research question is: How does conceptual modeling facilitate the process of theory development in IS design research? To answer this question we will review research on conceptual modeling and show how conceptual modeling can support the process of theory development.

According to Popper, science is the process of trying to refute theories (Popper, 2002). Since we want to explicate design theories by using conceptual models we have to evaluate them to substantiate or refute the underlying design hypotheses. We argue that ultimately evaluating the prescriptive statements of a conceptual model requires developing a corresponding instantiation. Since IS artifacts tend to be large applications in a complex socio-technical environment, evaluation is impeded by a large number of confounding factors. Hence, Markus et al. (2002) structure design theories by forming concise design principles. To transfer the concept of design principles into conceptual models, we propose applying the idea of patterns, as described in Alexander (1973). Hence, our third research question is: How does a pattern-based approach support developing theories in IS design research? To answer this question we extend the meta-model of research question one with the constructs required by the pattern approach.
The remainder of this paper is organized as follows. In the second section we review existing research on design theories in IS research. The result of that section is a meta-model that represents the components of IS design theories. In the third section we incorporate conceptual models in the development process of design theories. For this purpose, we introduce the pattern idea as outlined in (Alexander, 1979; Alexander, 1973). In the fourth section we apply our approach by explicating three aspects of a theory of IT service data management systems. The paper closes with a critical appraisal of our work and an outlook to future research activities. Figure 1 depicts the line of arguments of this paper.

Figure 1. The line of arguments in this paper

This paper is of exploratory and conceptual nature. Hence, we provide argumentative support when answering our research questions. However, we base our arguments upon available empirical and conceptual research results.

2 IS DESIGN THEORIES

In this section we discuss existing research on design theories in IS research. Secondly, we analyze the structure of design theories and summarize it as a meta-model. Finally, we identify two weaknesses in the current structure of design theories. First, no guidelines are available on how to structure concise design principles. Second, current design theories are explicated using natural languages.

2.1 Design Theories in Information Systems Research

Designing and developing new information systems to improve business processes or to enable new ways of doing business is an integral part of work for both IS researchers and IS practitioners (Mertens, 1995; Hevner et al., 2004). From a design research perspective, the pivotal research objectives of IS researchers is to provide theories that guide the development of efficient and effective information systems (Gregor, 2006).

The term design theory is not without discussion within design-oriented IS research. On the one hand, March and Smith (1995) and Hevner et al. (2004) reserve the term theory to natural and social sciences. A theory is described as a set of hypotheses, which claim to explain or predict phenomena (Popper, 2002). From this perspective, design research is about applying theories from natural and social sciences to solve perceived problems. On the other hand, various researchers recognize the importance of theoretical work in design research (Gregor, 2006; Walls et al., 1992).

Generally, the process of design is understood as planning, specifying, and subsequently implementing artifacts (Simon, 1969). As design research aims at providing solutions to perceived problems (Hevner et al., 2004), the central focus is to support the specification and development of future artifacts (Walls et al., 1992; Frank, 1998). Thus, design theories aim at providing guidance on how to solve a specific problem by claiming that “if acted upon, [they] will cause an artifact of a certain type to come into being” (Gregor, 2006 p 619). Hence, formulating design specifications is similar to formulating theories. Furthermore, design theories are then refuted or substantiated by instantiating the design specification within the intended domain (Walls et al., 1992).
As the designed artifacts are going to be deployed in a certain environment, the ability of attaining the goals is determined by the governing natural and social laws of that environment. Hence, many researchers demand a multidisciplinary approach in design research (Nunamaker & Chen, 1991; Gregor, 2006). Thus, developing design theories requires considering existing theories, e.g. explanatory, predictive and normative theories from natural or social sciences: “The prescriptive plane [of design theories] provides the common ground for integrating these different types of theories” (Walls et al., 1992 p 41). The constraints and intended applications of these underlying theories influence the properties of the resulting artifact and provide the base for evaluating the quality of the artifacts and thus the design theory itself.

2.2 Structure of Design Theories in IS Research

In this section, we analyze the structure of IS design theories as proposed by Walls et al. (1992). Various authors have applied design theories as a vehicle for capturing and formulating design principles that describe how information systems should be build (Markus et al., 2002; Moor, 2005; Jones & Gregor, 2006).

Design theories consist of two types of design propositions (see Figure 2): first, the design product specifies the properties the artifact has to possess to meet certain requirements, as well as propositions on how to test the quality of this relationship. Second, the design process describes the sequence of activities that is required to design an artifact in the way that it meets the stated requirements (Walls et al., 1992).

The design product itself consists of four elements: In the element class requirements the design theory developer specifies the problem and subsequently the goals the design theory is supposed to attain. In the element class design, the theory developer specifies the structural and functional properties and characteristics of the intended artifact. The element kernel theories references existing theories, e.g. from social science or mathematics, that constrain class requirements. The final element of the design product is a collection of testable hypotheses that allow evaluating the capability of the class design to meet the class requirements.

The second component design process consists of three elements: The element design method is specifying the process of designing the intended artifact in a way that the artifact meets the specified requirements. The element kernel theories again refers to existing theories that determine or influence the design process. The element testable design process hypotheses refer to propositions that can be derived from the design process and their underlying kernel theories and allow evaluating whether applying the design method results in the intended artifacts.

---

1 Please refer to Walls et al. (1992) for the following paragraphs.
2 Walls et al. (1992) use the prefix meta for requirements and design to denote that both aspects refer to a class of artifacts instead of a specific artifact (e.g. retail information systems versus the retail information system for company ABC). However, we argue that using the prefix meta is misleading as both requirements and design refer to a class of systems. Thus, in the remainder of this paper we will refer to both elements as class requirements and class design.
As Figure 2 shows, design theories capture design knowledge and experience on both the artifact and its construction process. Design theories are the composition of “…user requirements, a type of system solution (with distinctive features), and a set of effective development practices” (Markus et al., 2002 p 180).

Overall, our analysis of the structure reveals two weaknesses of IS design theories. First, the structure of kernel theories, requirements, design, and hypotheses remains abstract. There are no guidelines on how to structure concise design principles (Markus et al., 2002). Furthermore, existing design theories do not incorporate existing classifications of IS artifacts (e.g. March et al., 2000).

3 FRAMEWORK FOR DEVELOPING DESIGN THEORIES USING CONCEPTUAL MODELS

In this section, we first argue that conceptual models are theoretical design artifacts and provide semi-formal representation techniques for design theories. Second, we introduce the idea of patterns to contribute twofold: to provide guidance when developing design principles and to reduce the complexity of conceptual models. The result of this section is a framework for constructing IS design theories based on conceptual models and the idea of patterns.

3.1 Conceptual Models as the Theoretical Artifact in Design Research

As we have discussed above, design theories consist of both an artifact and its design process. Concerning the design process Nunamaker & Chen (1991), for instance, proposes five phases of design-oriented research: construction a conceptual framework, development of a system architecture, analysis and design of the system, implementation of the system, and evaluation of the system (see Figure 3). Concerning the classification of design products, March and Smith (March & Smith, 1995) distinguish between four types of artifacts: constructs, methods, models, and instantiations. However, Walls et al. argue that design research results “can be proven only by construction of the artifact” (Walls et al., 1992 p 38).

Hence, we argue that all four types of research outputs are necessary to develop IS artifacts. Constructs are necessary to describe certain aspects of a problem domain. Models depict problems and solutions in the domain: “they are set[s] of propositions or statements expressing relationships among constructs” (March & Smith, 1995 p 256). In the context of design research, models represent problem solutions and thus theoretical statements on design. Instantiations are the realization of a model: “[i]ntstantiations operationalize constructs, models, and methods” (March & Smith, 1995 p 258). Although, methods are necessary to develop all types of artifacts, we omit the discussion of methods,
as they are a composite artifact of a language and a process model (March & Smith, 1995). Instantiation of methods are processes within organizations (Greiffenberg, 2004).

Figure 3 depicts a mapping of IS design artifacts to the phases of design research. As it can be seen, constructs and models map to conceptual phases in design research. Models depict a solution, i.e. a prescription of what to do to attain a certain goal. Hence, models abstract form the necessary adaptations, which have to be incorporated when developing and introducing an instantiation. Therefore, we argue that conceptual models are the central theoretical artifacts in IS design research.

Additionally, there are also pragmatic reasons for expressing design theories by employing conceptual modeling. Conceptual models are usually constructed in a semi-formal graphical modeling language. Thus, they provide a restricted vocabulary. Furthermore, often conceptual modeling incorporates a multi-perspective approach to facilitate communication between stakeholders (Wand & Weber, 2002).

In sum, we argue that expressing design theories by employing conceptual models is a beneficiary approach.

3.2 Patterns as a Useful Way to Structure Design Theories Based on Conceptual Models

Conceptual models tend to be very complex, as the work of Becker & Schütte (2004) or Scheer (1998) demonstrates. Furthermore, conceptual models generally focus on providing complete design proposals. To employ conceptual modeling in the development process of design theories it is necessary to decompose conceptual models to small and concise entities. Therefore, we apply Alexander’s pattern approach (Alexander, 1973). Alexander’s foundational conceptualization of design is that “good” design resolves perceived misfits within a context (Alexander, 1973). Originally developed in the field of architecture, patterns have been applied to many domains (Schumacher, 2003; Gamma et al., 1994). To facilitate good design, design requirements are deconstructed in a hierarchical manner. A certain aspect of the design solution will meet each requirement. Alexander’s main argument is that design issues can be solved by combining coherent and modular solutions to specific problems (Alexander, 1973). These coherent solutions are called patterns (Alexander, 1973; Alexander, 1979). A pattern generally comprises the following elements (Buschmann et al., 1998).

- The context comprises causes which lead to the problem described in a pattern and the conditions under which the problem occurs. The context should support assessing the relevance of a pattern.
- The problem describes contradictions causing the perceived misfits in the context of the pattern. These aspects of the problem are often called forces (Buschmann et al., 1998).
- The next section of a pattern explains the proposed solution by describing how to dissolve the forces described before. An illustration of consequences of applying the pattern is given as well.
- The closing section of a pattern is composed of references to related patterns.

As patterns are rarely used independently, Alexander broadens the pattern idea to a system of interrelated patterns that he calls a pattern language (Alexander, 1979). The semantic power of such pattern languages is determined by the references between patterns, which consequently allow capturing solutions for more complex problems.

Overall, patterns are coherent design entities that describe a solution to a specific problem in the sense of design principles as proposed by Markus et al. (2002). Decomposing design theories into patterns enables identifying the design principles. Furthermore, concise patterns allow evaluating each design principle (i.e. construct of the pattern language) separately and thus derive more concise and elementary evaluation objectives. Therefore, the idea of pattern languages facilitates reconstructing dependencies between different elements of a design theory and thus supports the identification of core elements. By stating consequences for application of the pattern solution, patterns facilitate explaining the impact of applying a design principle as well as formulating hypotheses on its benefit.
3.3 Framework

Based on the concept of patterns we can now extend the meta-model of section 2.2 to incorporate conceptual modeling in design theory development.

A design theory consists of patterns. These patterns reference each other and thus form a pattern language. Please note that these references can point to design principles of other theories as well. A design pattern consists of a context, a problem, and a solution, i.e. a design proposition expressed by a conceptual model. The context refers to kernel theories that apply to the specific pattern.

![Figure 4: Structure of pattern-based design theories](image)

As Figure 4 reveals, it is not required to assign kernel theories. Walls et al. (1992) argue that in information systems it might not be possible to identify appropriate kernel theories. Hence, Markus et al. broaden the definition of kernel theory to include theories-in-use (e.g. Sarker & Lee, 2002). The problem describes forces that are the result of user requirements. The conceptual model depicts a solution (either by specifying an artifact or a method) and resolves these forces and propose certain consequences when applying it. These consequences, either good or bad, are the basis for testable hypotheses.

The process of constructing patterns can be found e.g. in Schumacher (2003). Please note that the concept of patterns can also be used to describe common analysis and design processes (Köhne, 2005). Thus, patterns can also be used to describe the design process section of design theories.

In sum, this framework enables specifying design theories as conceptual models. The pattern approach facilitates deriving concise design principles. Thus, we have answered research questions two and three on how to facilitate developing design theories.

4 TOWARDS A DESIGN THEORY FOR IT SERVICE DATA MANAGEMENT SYSTEMS

In the following we apply our framework by explicating a design theory for IT service management systems.
4.1 Introduction to the IT Service Management

The IT services industry will likely have a worldwide market volume of about US$ 760bn. by 2009 (Hale et al., 2005). As IT services (i.e. services that rely on information technology) become more and more complex, systematic development and efficient delivery of IT services is an important requirement (Bullinger et al., 2003). IT service providers face challenges similar to that of industrial enterprises: establishing an integrated management of services throughout their lifecycle across different stages of the service value chain (Da Rold et al., 2005). Hence, an integrated view on all aspects of service engineering and delivery is needed. We call this view IT service data management (Böhmann, 2004; Böhmann et al., 2004).

4.2 Aspects of a Design Theory for IT Service Data Management

Figure 5 summarizes three fundamental aspects of the IT-SDM design theory. The objective of IT-SDM is to facilitate requirements determination for IT service data management by specifying key domain concepts and their relationships. In the following, we focus on three key patterns that define building blocks for mass-customized IT solutions: the Service Architecture, the Service Module, and the Service Level Agreement. These patterns form three design principles of our design theory. The following tables show these design principles in detail.

![Figure 5: The SDM design patterns](image)

### Service Architecture

<table>
<thead>
<tr>
<th>Context (with theoretical references)</th>
<th>Problem (with forces)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similar to industrial products, IT services are complex systems of various functionalities that are provided by many internal organization units and sub-providers. In industrial production industry, product architectures are used to componentize product elements (Scheer, 1998) and thus reduce coordination costs (Coase, 1937). This idea also has been transferred to software engineering (Pohl et al., 2005).</td>
<td></td>
</tr>
</tbody>
</table>

- Mass-customization for IT services (Böhmann, 2004) requires standardized service elements that can be combined.
- Many stakeholders, e.g. marketing, sales, and engineering, have different views on IT services.
- Especially managing long-term IT services requires considering existing service contracts and their impact on the service infrastructure.

---

3 A more detailed description of the patterns would go beyond the scope of this paper.
The differentiation in architecture, catalogue, and configurations reduces coordination costs between stakeholders in IT service engineering and delivery.

Service architectures allow mass customization of IT services.

Service architectures enable tracking of impacts of possible changes in the service capabilities.

<table>
<thead>
<tr>
<th>Context</th>
<th>Modern IT services are complex sets of functionalities and rely on technical, organizational, and human resources. Thus, services can be characterized as complex systems (Bunge, 1977; Ropohl, 1979). Efficient management of such complex systems requires mechanisms to reduce complexity (Baldwin &amp; Clark, 2000).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td>Decomposing service functionality requires describing visible and accessible characteristics. The dependencies between service functionalities have to be identified and documented.</td>
</tr>
<tr>
<td>Conceptual Model (ERM)</td>
<td><img src="image" alt="Diagram of Service Architecture and Module" /></td>
</tr>
<tr>
<td>Consequences (testable hypotheses)</td>
<td>IT service can be decomposed into service modules (Böhmann, 2004). It is possible to develop standardized definitions of IT services by specifying an interface. It is possible to develop service products from standardized service module interfaces.</td>
</tr>
<tr>
<td>References</td>
<td>Meta-model of BWW-constructs (Rosemann &amp; Green, 2002) (external prerequisite)</td>
</tr>
</tbody>
</table>

**Table 1: The pattern Service Architecture**

Efficient service delivery has to provide the contracted service functionality at the agreed quality (Bullinger et al., 2003; Sturm et al., 2000). However, services generally do not exhibit characteristics that customers can inspect prior to acquiring a service (Böhmann, 2004). Furthermore, services rely on the integration of external factors, e.g., input of the service customer (Burr, 2004).

<table>
<thead>
<tr>
<th>Context</th>
<th>Efficient service delivery has to provide the contracted service functionality at the agreed quality (Bullinger et al., 2003; Sturm et al., 2000). However, services generally do not exhibit characteristics that customers can inspect prior to acquiring a service (Böhmann, 2004). Furthermore, services rely on the integration of external factors, e.g., input of the service customer (Burr, 2004).</th>
</tr>
</thead>
</table>

**Table 2: The pattern Service Module**


<table>
<thead>
<tr>
<th>Problem (with forces)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration of external factors requires definition of responsibilities of service provider and service client.</td>
</tr>
<tr>
<td>Contracting services require defining the outcome of the service contract.</td>
</tr>
<tr>
<td>As services change over time, the quality definitions have to change as well.</td>
</tr>
<tr>
<td>Services have various states that result in different quality requirements.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conceptual Model (ERM)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Consequences (testable hypotheses)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service quality can be described as a set of objectives that are measured and assigned to specific parties.</td>
</tr>
<tr>
<td>It is possible to measure each service quality criterion.</td>
</tr>
<tr>
<td>All types of IT services have distinct states, e.g. maintenance, operating, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Service Level Agreements (Ludwig et al., 2003) (external prerequisites)</td>
</tr>
<tr>
<td>Service Module (prerequisite)</td>
</tr>
</tbody>
</table>

Table 3: The pattern Service Level Agreement

### 5 CONCLUSION

In this paper, we have first analyzed the concept of design theories. Secondly, we identified conceptual models as the central theoretical artifact and subsequently proposed expressing IS design theories using conceptual modeling. To facilitate the development of concise design principles we have incorporated the idea of patterns. Thus, our framework enables researchers to realize the following benefits:

- Decomposing existing conceptual models into coherent patterns reduces the complexity of the resulting design theory. Patterns can be evaluated individually by testing the provided hypotheses. For instance, the modularization of IT services has already been applied successfully in (Böhmann, 2004). Thus, this hypothesis has been substantiated.
- The references between patterns help to analyze the immutable core of the theory: the “deep structure” (Weber, 1987 p 13). The patterns show existing links to design theories and hence help to build up a theory network (Balzer et al., 1987).
- Patterns can be applied individually and reduce the overhead of learning and adaptation. Thus, the individual utility of a design theory can be determined more easily. Results from evaluating design principles will lead to local changes in the patterns. Thus, our approach facilitates the incremental enhancement of design theories.

However, there are some limitations to our approach as well:

- We could not yet identify any patterns describing the design process, which is necessary to formulate a complete design theory. Existing approaches on how to develop management information systems could be analyzed and adapted for the specific requirements of the IT service industry.
- As we have discussed, evaluating design principles requires instantiating and adapting the model and test the instantiation. As models are interpreted and then implemented in a technical environment, many confounding factors may apply. However, recent approaches in software engineering such as model driven development may provide a solution for that problem.
• Our framework represents a semi-formal approach to specifying design theories. To facilitate a more formal theory development, meta-theoretical programs from philosophy of science could be applied. For instance, philosophical Structuralism could facilitate formalizing the structure of design theories (Balzer et al., 1987).

• Currently, we apply conceptual models to depict the solution sections of the design principles. However, they could be used to describe other sections as well. For instance, Rossi et al. proposed a modeling approach for rationales, which could be applied to model the forces of the patterns (Rossi et al., 2004).

However, these limitations do not corrode our approach. Thus, future work will include addressing the above-mentioned limitations as well as developing a more fine-grained design theory of IT service data management.

References
Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Publishing Company, Reading, MA, USA.


