Wirtschaftsinformatik Proceedings 2007 Wirtschaftsinformatik

2-28-2007

Fostering the Evaluation of Reference Models: Application and Extension of the Concept of IS Design Theories

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

Tilo Böhmann
Technische Universität München, boehmann@in.tum.de

Helmut Krcmar
Technische Universität München, krcmar@in.tum.de

In: Oberweis, Andreas, u.a. (Hg.) 2007. *eOrganisation: Service-, Prozess-, Market-Engineering*; 8. Internationale Tagung Wirtschaftsinformatik 2007. Karlsruhe: Universitätsverlag Karlsruhe

ISBN: 978-3-86644-094-4 (Band 1)
ISBN: 978-3-86644-095-1 (Band 2)
ISBN: 978-3-86644-093-7 (set)

Fostering the Evaluation of Reference Models: Application and Extension of the Concept of IS Design Theories

Michael Schermann, Tilo Böhmann, Helmut Krcmar

Chair for Information Systems
Technische Universität München
Boltzmannstraße 3
85748 Garching b. München
{michael.schermann|boehmann|krcmar}@in.tum.de

Abstract

Despite the imperative to substantiate innovative research results expressed in reference models, little methodical guidance exists for evaluating reference models yet. We propose that IS design theories [WaWE92] can provide theoretical guidance for reference model evaluation since reference models can be formulated as a set of design principles that consist of testable propositions, kernel theories, and intended applications. We show how to facilitate the reconstruction process by applying the idea of pattern languages. Such decomposed reference models allow evaluating each design principle separately and thus formulating a more concise and elementary evaluation objective. We demonstrate the benefits of reconstructing reference models as design theories on the Service Data Management reference model that has been developed by the authors.

1 Introduction, Problem Statement, and Challenges

Since the beginning of the Information systems discipline in German-speaking countries many reference models have been constructed and published [FeLo04b], among them most notably Scheer's Y-CIM or Becker's Retail-H [BeSc04; Sche98]. The term reference model has been adopted by companies in many industries (e.g. Software, Health, Banking) to denote best practices in process design and software design [BeKn02; FeLo04a]. Based on the importance of reference models in the German IS community, reference models and the process of reference modeling have become research objects themselves, e.g. by supporting the adaptation of reference models or facilitating the management of reference models [BeDK04; Thom06].

Choosing reference models and substantiating their claims require sound evaluation. With the number of reference models rising, potential users, e.g. companies, are faced with a problem of choosing references models and hence evaluating the quality and appropriateness of potential usefulness [FeLo04b]. Furthermore, researchers want to evaluate the utility of their reference models and thus substantiate their proposed claims of reference [BöSK06]. The fundamental claim of reference models is that they accelerate model-based development phases, e.g. requirements engineering and system design, by adapting the reference model instead of pursuing individual modeling [BeSc04; FeLo04a]. Thus, reference models usually have a prescriptive notion as they propose how information systems or processes *should* be designed. However, most of the available reference models lack of evaluation results regarding their utility, suitability, and quality [BöSK06; FeLo04b].

Despite the importance of evaluating reference models, little methodical guidance exists for evaluating reference models yet. In their effort to facilitate evaluation, researchers have found that evaluating information models and particularly reference models is especially difficult due to methodological, philosophical, and practical reasons [BöSK06; FeLo03b; Fran00; Fran98a]. First, evaluating the utility of reference models in a positivist understanding would require gaining access to a large number of users that actually have applied a reference model to reduce the impact of confounding factors in the reference model evaluation. Second, reference models are supposed to be adapted to specific needs of the reference model user. Thus, evaluating reference models has to cope with a large number of confounding factors [BöSK06]. Third, constructors of reference models often do not reveal underlying assumptions, theoretical foundations, as well as the immutable core of their reference models [BePf06]. Overall, reference models do not yet provide necessary elements to evaluate the utility and their claims.

Reconstructing reference models as design theories provides the missing link. Design theories have been proposed as scientific method to capture design experience and provide prescriptive information on how to design information systems in specific domains [MaMG02; WaWE92]. We argue that design theories can be seen as a counter piece to reference models. An important aspect of design theories is to reveal underlying theoretical assumptions as well as provide hypotheses that can be refuted or substantiated in empirical analyses. Thus, reconstructing reference models by the structure of design theories facilitates the identification of design propositions in reference models. Furthermore, the methodology of developing design theories requires linking these design propositions to so-called kernel theories that provide the theoretical base for design theories. Reconstructing reference models as design theories also requires the proposition of testable hypotheses and thus facilitate the evaluation of reference models and hence the theoretical and practical advancement of reference models.

Overall, we propose a way of reconstructing reference models as testable theories as it has been demanded, e.g. by Becker and Pfeiffer [BePf06]. Hence, we set the following research questions:

- What are the benefits of applying the concept of design theory to reference models?
- How can we facilitate the process of reconstruction?
- What are the benefits of reconstructing reference models as design theories?

The remainder of this paper is organized as follows. In section 2 we introduce the concept of design theory. As reference models are complex design proposals, it is necessary to decompose them into design principles [MaMG02]. To facilitate the decomposition process, we introduce the idea of patterns in section 3 [Alex79]. We further show that the structure of patterns helps to identify required elements of design theories. *Reconstructing a reference model will result in a pattern language that consists of the design principles proposed by the reference models. Each pattern reflects a design principle that can be evaluated individually.* In section 4 we demonstrate the utility of our approach on the example of the SDM reference model that has been developed by the authors [BWFK04]. The paper finishes with a conclusion of the results and provides an outlook on further research. Figure 1 shows the main arguments of this paper.

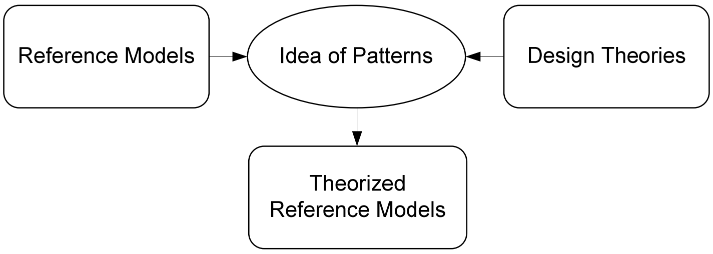


Figure 1: The line of argumentation of this paper

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

2 Design Theories

In this section, we introduce design theories as a methodology of capturing domain knowledge and design experiences in an empirical refutable way. Furthermore, we analyze similarities and differences of design theories and reference models. We will conclude that applying the structure of design theories allows reference model constructors to explicate underlying theoretical assumptions and provide testable hypotheses. This reveals the benefit proposals of the reference model, and thus fosters the academic and practical evaluation of reference models.

2.1 Characteristics of Design Theories

Designing and developing new information systems, e.g. to improve business processes by automation or to enable new ways of doing business is an integral part of work for both IS researchers and IS practitioners [HMPR04; Mert95; Wiss94]. Therefore, one of the pivotal research objectives of IS researchers is to provide theories and practical guidance on facilitating efficient and effective design of information systems. Grounding on the seminal paper by Walls et al. [WaWE92] various authors have used the construct of design theories as a vehicle for capturing and formulating design principles that describe how information systems should be build [MaMG02]. In the following, we will highlight only the main aspects of design theories that are required for our line of argumentation[1].

In the context of design research, designing artifacts means to develop and enhance theories. Generally, the process of design is understood as planning, specifying, and subsequently implementing artificial artifacts [Simo69]. As design research aims at solving problems [HMPR04], the central focus of design science is to support the specification of future artifacts, e.g. new kinds of information systems [Fran97; Fran98a; WaWE92]. Thus, design "…is a set of hypotheses, and ultimately can be proven only by construction of the artifact it describes. The feasibility of a design can, however, be supported by scientific theory to the extent that the design embodies principles of the theory" [WaWE92, p. 38]. Hence, formulating design specifications can be seen as the same process of formulating theories.

Design theories are prescriptive and thus goal-oriented. In contrast to the explanatory and predictive nature of theories in natural science, theories in design science are of prescriptive nature. As design theories aim at providing guidance on how to solve a specific problem: "if acted upon, [they] will cause an artifact of a certain type to come into being" [Greg06, p. 619]

[1] Detailed discussion of design theories is provided in [MaMG02; WaWE92] and the referenced literature there.

Design theories prescribe certain design principles that will lead to applications, which are more effective.

Design theories build upon kernel theories. 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 socials laws of that environment. Thus, Walls et al. argue that developing design theories requires considering existing theories, e.g. explanatory, predictive and normative theories from natural or social sciences: “The prescriptive plane provides the common ground for integrating these different types of theories” [WaWE92, p. 41]. Hence, design theories are composite theories, as they rely on theories, e.g. predictive theories [Greg06; WaWE92]. 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.

Design theories prescribe both the artifact and the process of creating that artifact. Besides defining the properties of the intended artifacts, Walls et al. state that design theories should incorporate the process of designing the artifact. They argue that natural and social laws of the environment also determine the process of designing the artifact. Thus, the design process heavily influences the design result – the artifact [WaWE92].

2.2 Structure of Design Theories

Design theories consist of two components: the *design product* component 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. The *design process* component describes the process that is required to design an artifact in the way that it meets the stated requirements [WaWE92].

The first component *design product* consists of four elements [WaWE92]: In the element *class requirements* the design theory developer specifies the problem and subsequently the goals the design theory is supposed to solve and attain. In the element *class design,* the theory developer specifies the structural and functional properties and characteristics of the intended artifact[2]. The element *kernel theories* specifies existing theories, e.g. from social science or mathematics, that constraint or support statements made in the class-design section. The final element of the

[2] Walls et al. 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) [WaWE92]. We argue, that using the prefix meta is misleading as both elements refer to an instantiation relationship [Stra96]. Thus, in the remainder of this paper we will refer to both sections as class requirements and class design.

design product component is a collection of *testable hypotheses* that allow evaluating the capability of the class design to meet the class requirements. Overall, the component design product specifies the class of artifacts the design theory proposes to facilitate.

The second component *design process* consists of three elements [WaWE92]: The element *design method* is specifying the process of designing the intended artifact from the class design 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.

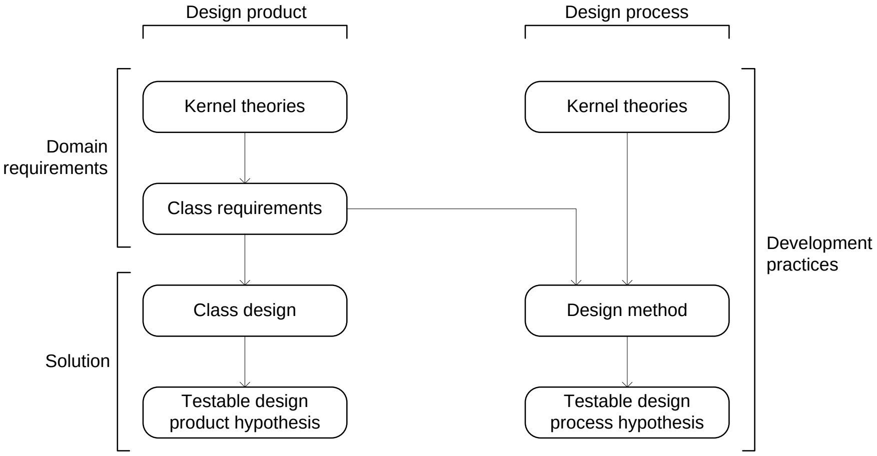


Figure 2: Structure of design theories (according to [MaMG02; WaWE92])

Overall, as Figure 2 shows, design theories capture design knowledge and experiences on both the artifact and the 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" [MaMG02, p. 180].

2.3 Benefits of the Structure of Design Theories for Evaluating Reference Models

Design theories provide a framework for design solution proposals in a theoretical and testable way. We argue that by adopting the framework of design theories for reference models we can realize the following benefits:

- *Design theories require the specification of kernel theories when constructing reference models.* Current reference models often do not reveal their underlying theoretical assumption [BePf06]. Thus, adopting the design theory framework requires reconstructing or disclosing underlying assumptions of the reference models.

- *Design theories provide a framework to reformulate reference models as testable hypotheses.* Design theories result in testable hypotheses that provide the foundation of empirical research on reference models, their utility, and their usage [FeLo04a; WaWE92]. Thus, applying the idea of design theories to reference models facilitate the reconstruction of reference models as genuine theories of IS research.
- *Design theories provide the concept of design principles that guide choosing and adapting reference models.* Design principles can be used to group connected requirements to coherent units that propose a certain utility. By enabling references, e.g. dependencies between design principles [WaWE92] one can identify immutable design principles of reference models. Here, reference models may restrict the adaptations.

In sum, the answer to our first research question is that design theories provide the framework for (re)constructing testable reference models.

3 (Re)constructing Reference Models as Design Theories

To facilitate the reconstruction process we apply Alexander's pattern approach [Alex73].

3.1 Patterns in the Context of Reference Models

Alexander's foundational conceptualization of design is that good design solution resolves perceived misfits in a context [Alex73][3]. To facilitate good design, design requirements are deconstructed in a hierarchical way. A certain aspect of design solution will meet each requirement. The general solution is the combination of all solutions. Overall, the main argument is that design issues can be solved by combining coherent and rather independent solutions to specific problems [Alex73]. These coherent solutions are called *patterns* [Alex73; Alex79]. A pattern generally comprises the following elements [Schu03]: the *context* comprises

[3] Alexander's ideas refer to design issues in the field of architecture. However, the notion of patterns has been applied to many areas in various disciplines, especially information systems development [GHJV94; Schu03].

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 [BMRS98]. The *problem* describes contradictions causing the perceived misfits in the context of the pattern. These aspects of the pattern problem section are often called forces [BMRS98]. The next section of a pattern explains the proposed *solution* by dissolving the forces described before. An illustration of consequences of applying the pattern is given as well [BMRS98]. The closing section of a pattern is composed of *references* to related patterns [Schu03]. In sum, a pattern represents a complex structure of knowledge from an application-oriented perspective. The goal of patterns is to explicate experiences and established expert knowledge [Schu03]. As patterns are rarely used independently, Alexander broadens the pattern idea to a system of interrelated patterns that he called *pattern language* [Alex79]. The semantic power of such pattern languages is determined by the references between patterns, which consequently allow capturing solutions for more complex problems [Schu03].

What is the benefit of applying the idea of patterns for reconstructing reference models? Reference models tend to be very complex [BDKK02; BeSc04; Sche98]. Furthermore, reference models generally focus on providing complete design proposals that have to be adapted. Patterns are coherent entities that describe a solution to a specific problem in the sense of design principles as proposed by [MaMG02]. Decomposing reference models into patterns enables identifying the design principles formulated by the reference model. Hence, reconstructing the reference model as pattern language allows reformulating the reference model as a set of design principles and thus forming a design theory [MaMG02]. Such theorized reference models allow evaluating each design principle (i.e. construct of the pattern language) separately and thus derive more concise and elementary evaluation objectives. Furthermore, the idea of pattern languages facilitates reconstructing dependencies between different elements of a reference model and thus supports the identification of core elements. By stating consequences when applying the pattern’s solution, the pattern concept facilitates explaining the impact of applying a pattern as well as formulating hypotheses on the benefit of the pattern.

3.2 (Re)construction Framework

Overall, the pattern approach facilitates reconstructing reference models as design theories. Based on the concept of patterns we can now develop a framework for reconstructing reference models. This framework is depicting the elements that are required for the reconstruction process. Thus, the framework ensures that a reconstruction process leads to design principles as required by [MaMG02]. As depicted in Figure 3 the basic structure of theorized reference models is derived from the structure of design theories as proposed by Walls et al. [WaWE92].

A *theorized reference models* consists of *patterns*. These patterns have references to each other and thus form a pattern language. We introduce the reference types *prerequisite* and *specialization*. Please note that these references can point to external design principles as well. A pattern consists of a *context*, a *problem*, and a *solution*. The context refers to kernel theories that apply to the specific pattern.

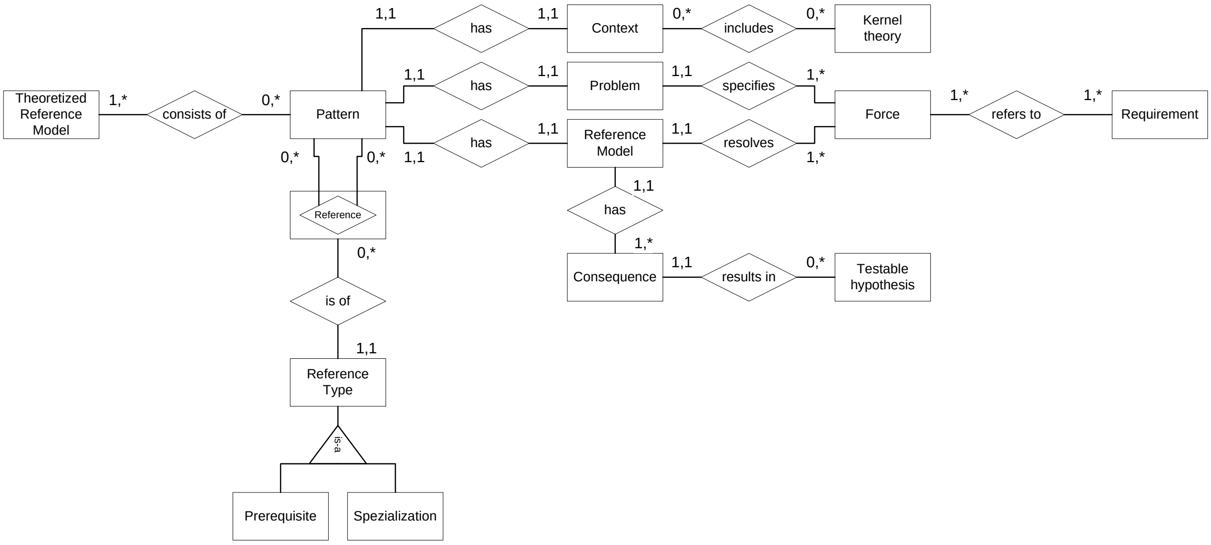


Figure 3: Structure of theorized reference models (according to [Alex79; WaWE92])

Furthermore, figure 3 reveals that it is not required to assign kernel theories. Walls et al. argue that in information systems it might not be possible to identify appropriate kernel theories [WaWE92]. Hence, Markus et al. broaden the definition of kernel theory to include practitioner theories-in-use, e.g. [SaLe02] and theory candidates. The problem analyzes *forces* that are the result of user *requirements*. The *reference model* (or a specific part of it) resolves these forces and has 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 [Köhn05; Schu03]. Please note that the concept of patterns can also be used to describe common analysis and design processes, as explained in e.g. [Köhn05]. Thus, patterns can also be used to describe the design process section of design theories.

In sum, this framework for theorized reference models combines the proposed structure of design theories and patterns and guides the reconstruction process. Thus, we have answered research question two on how to facilitate the process of reconstruction.

4 Demonstration: Reconstruction of the SDM Reference Model

In this section we demonstrate our approach on the Service Data Management (SDM) reference model that has been developed by the authors [BWFK04]. We use the SDM reference model, since we are fully aware of the underlying objective and the intended applications and do not rely on interpretations. Hence, we hope to formulate a more accurate reconstructed reference model.

4.1 Introduction to the SDM Reference Model

The IT services industry will likely have a worldwide market volume of about US$ 760bn. by 2009 [HDLA05]. As IT services (i.e. services that rely on information technology) become more complex, systematic development and efficient delivery of IT services is an important challenge [BuSG03]. 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 [DaJY05]. Hence, an integrated view on all aspects of service engineering and delivery is needed. We call this view service data management [Böhm04; BWFK04]. The objective of the SDM reference model is to capture data structures for service data management.

4.2 The SDM Reference Model as Design Theory

Figure 4 summarizes three fundamental aspects[4] of the SDM reference model and depicts them as patterns: the *Service Architecture*, the *Service Module*, and the *Service Level Agreement*. These patterns form the design principles of our design theory. The following tables show these design principles in detail.

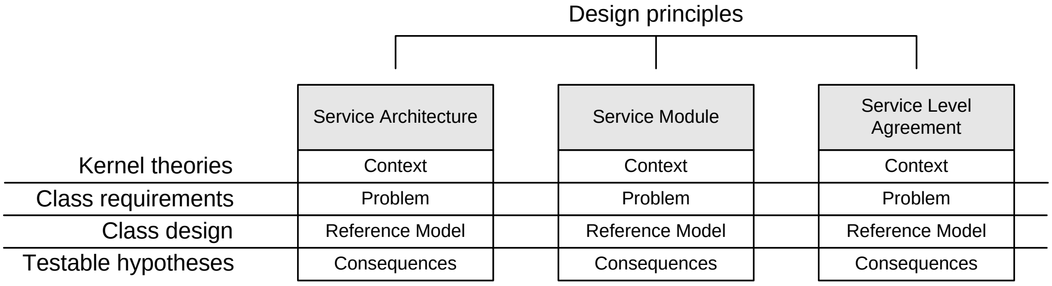


Figure 4: The SDM patterns in the light of the categories of design theories

[4] A more detailed description of the patterns, especially of the section reference model, would go beyond the scope of this paper.

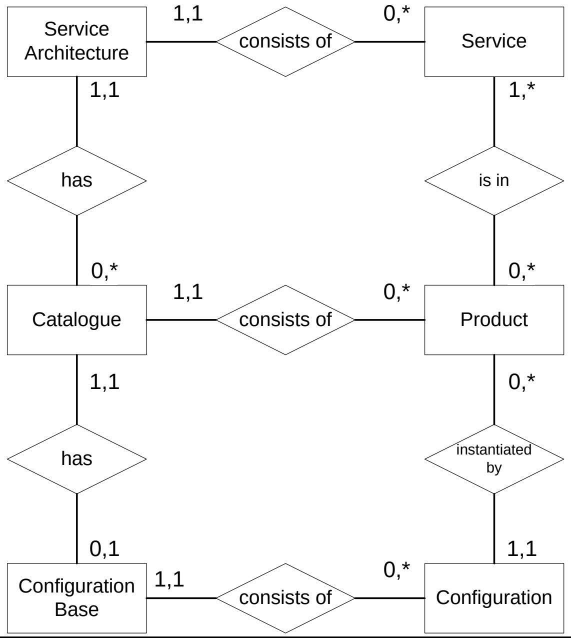
Service Architecture	
Context (with theoretical references)	Similar to industrial products, IT services are complex systems of various functionalities that are provided by many internal organization units and sub-providers. In industrial production industry, product architectures are used to componentize product elements [Sche98] and thus reduce coordination costs [Coas37]. This idea also has been transferred to software engineering [PoBL05].
Problem (with forces)	• Mass-customization for IT services [Böhm04] 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.
Reference Model	 Service Architecture 1,1 consists of 0,* Service 1,1 1,* has is in 0,* 0,* Catalogue 1,1 consists of 0,* Product 1,1 0,* has instantiated by 0,1 1,1 Configuration Base 1,1 consists of 0,* Configuration
Consequences (testable hypotheses)	• 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.
References	• Product Architectures, e.g. [Sche98] (external prerequisite) • Service Module (prerequisite)

Table 1: The pattern Service Architecture

Service Module	
Context (with theoretical references)	Modern IT services are complex sets of functionalities and rely on technical, organizational, and human resources. Thus, services can be characterized as complex systems [Bung77; Ropo79]. Efficient management of such complex systems requires mechanisms to reduce complexity [BaCl00].
Problem (with forces)	• Decomposing service functionality requires describing visible and accessible characteristics. • The dependencies between service functionalities have to be identified and documented.

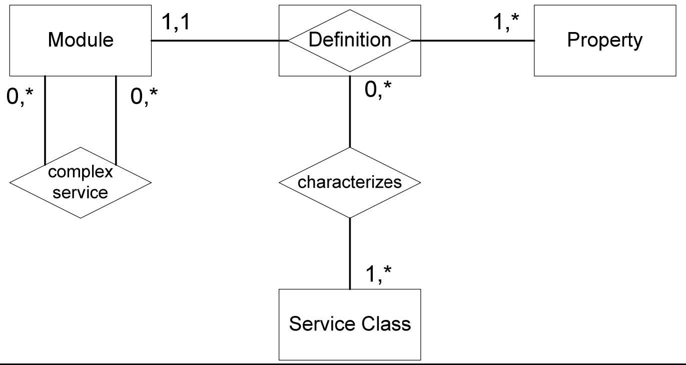
Reference Model	 Module 1,1 Definition 1,* Property 0,* 0,* 0,* complex service characterizes 1,* Service Class
Consequences (testable hypotheses)	• IT service can be decomposed in service modules [Böhm04] • It is possible to develop standardized definitions of IT services by specifying an interface. • It is possible to develop service products from standardized service module interfaces.
References	• Meta-model of BWW-constructs [RoGr02] (external prerequisite)

Table 2: The pattern Service Module

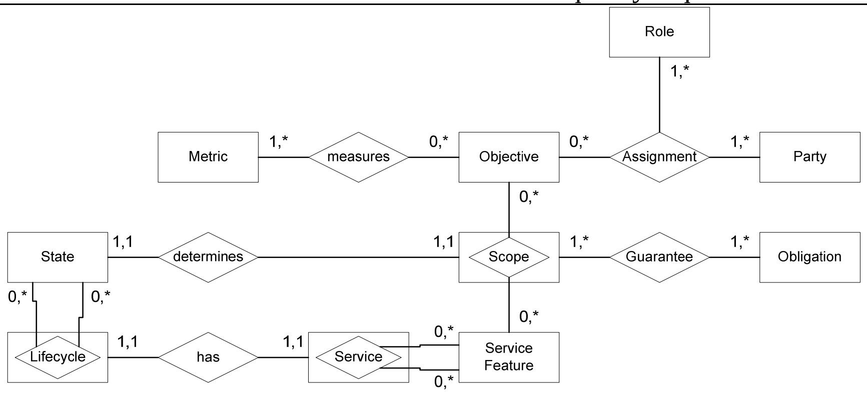
Service Level Agreement	
Context (with theoretical references)	Efficient service delivery has to provide the contracted service functionality at the agreed quality [BuSG03; StMJ00]. However, services generally do not exhibit characteristics that customers can inspect prior to acquiring a service [Böhm04]. Furthermore services rely on the integration of external factors, e.g. input of the service customer [Burr04].
Problem (with forces)	• Integration of external factors requires definition of responsibilities of service provider and service client. • Contracting services require defining the outcome of the service contract. • As services change over time, the quality definitions have to change as well. • Services have various states that result in different quality requirements.
Reference Model	 Role 1,* Metric 1,* measures 0,* Objective 0,* Assignment 1,* Party 0,* State 1,1 determines 1,1 Scope 1,* Guarantee 1,* Obligation 0,* 0,* 0,* 0,* Lifecycle 1,1 has 1,1 Service Service Feature 0,*
Consequences (testable hypotheses)	• Service quality can be described as a set of objectives that are measured and assigned to specific parties. • It is possible to measure each service quality criterion. • All types of IT services have distinct states, e.g. maintenance, operating, etc.
References	• Web Service Level Agreements [LKDK03] (external prerequisites)

Table 3: The pattern Service Level Agreement

4.3 Implications for evaluating reference models

We have proposed design theories as a suitable framework for reconstructing reference models to facilitate the evaluation and thus to enhance the benefit and utility of reference models. So, what benefits can be derived from the example for evaluating reference models?

- Decomposing the reference model into coherent patterns reduces the complexity. Patterns can be evaluated individually by testing the provided hypotheses. As said in the example, the modularization of IT service has already been applied successfully in [Böhm04]. Thus, this hypothesis has been substantiated.
- Referring to existing theoretical foundations, i.e. kernel theories, in the context section allows reference model constructors to reveal underlying assumptions. Furthermore, the context describes intended applications of the specific pattern.
- The references between patterns help to analyze the immutable core (prerequisite patterns) and guide reference model adaptation and configuration. The patterns show existing links to other reference models and hence help to avoid double work [FeLo04b].
- Patterns can be applied individually and reduce the overhead of learning and adaptation. Thus, the individual utility 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 reference models.

Overall, reconstructing reference models as design theories based on the pattern idea provide a beneficiary framework for constructing and evaluating reference models.

5 Conclusion, Limitations, and Outlook

In this paper we have proposed the reconstruction of references models as IS design theories [WaWE92] to substantiate reference models as innovative research outcomes by providing a theoretical and practical foundation for evaluating reference models. To support the reconstruction, we have applied the pattern approach to facilitate formulating the reference models as set of design principles. By the example of the SDM reference model, we have demonstrated the feasibility and utility of reconstructing reference models as design theories.

However, our approach has some limitations:

- We could not identify any patterns describing the design process yet, which is necessary to formulate a complete design theory. Existing approaches on how to use reference models could be analyzed and adapted for the specific requirements of the IT service industry.
- The reconstruction process either has to be done by the authors or relies on the capability of interpreting information models and associated documentation. Here recent research on collaborative reference modeling and “open models” could be applied [Broc04; KoSF06].
- The pattern approach does not support multiple perspectives on reference models [BDKK02; BeDK04]. However, this shortcoming has already been identified in the pattern community and various solutions have been proposed, e.g. as discussed by [Köhn05].
- Managing theorized reference models requires efficient management of their patterns. Approaches for version management of reference models could be combined with approaches for managing pattern languages [Cunn05; Thom06].

Despite these limitations, we conclude that reconstructing reference models as design theories is a promising approach that can foster both the academic and practical utility of reference models. Thus, our future work will include providing tool support for reconstructing reference models as well as addressing the above-mentioned limitations.

References

[Alex73] *Alexander, C.:* Notes on the Synthesis of Form. 7 ed., Harvard Univ. Press, Cambridge, Mass. 1973.

[Alex79] *Alexander, C.:* The timeless way of building. Oxford University Press, New York 1979.

[BaCl00] *Baldwin, C.Y.; Clark, K.B.:* The power of modularity. MIT Press, Cambridge, Mass. 2000.

[BDKK02] *Becker, J.; Delfmann, P.; Knackstedt, R.; Kuropka, D.:* Konfigurative Referenzmodellierung. In: *Becker, J.; Knackstedt, R. (Eds.)*: Wissensmanagement mit Referenzmodellen: Konzepte für die

Anwendungssystem- und Organisationsgestaltung. Physica-Verlag, Heidelberg 2002, pp. 25-144.

[BeDK04] *Becker, J.; Delfmann, P.; Knackstedt, R.:* Konstruktion von Referenzmodellierungssprachen: Ein Ordnungsrahmen zur Spezifikation von Adaptionsmechanismen für Informationsmodelle. In: Wirtschaftsinformatik 46 (2004) 4, pp. 251-264.

[BeKn02] Becker, J.; Knackstedt, R. (2002). *Referenzmodellierung 2002: Methoden - Modelle - Erfahrungen* (90). Münster: Westfälische Wilhelms-Universität Münster, Institut für Wirtschaftsinformatik.

[BePf06] *Becker, J.; Pfeiffer, D.:* Konzeptionelle Modellierung: Ein wissenschaftstheoretischer Forschungsleitfaden. In: *Lehner, F.; Nösekabel, H.; Kleinschmidt, P. (Eds.)*: Multikonferenz Wirtschaftsinformatik 2006. GITO, Berlin 2006, pp. 3-19.

[BeSc04] *Becker, J.; Schütte, R.:* Handelsinformationssysteme. 2 ed., Redline Wirtschaft, Frankfurt am Main 2004.

[BMRS98] *Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M.:* Pattern-orientierte Software-Architektur: Ein Pattern-System. Addison-Wesley-Longman Verlag, Bonn 1998.

[Böhm04] *Böhmann, T.:* Modularisierung von IT-Dienstleistungen - Eine Methode für das Service Engineering. Deutscher Universitäts-Verlag, Wiesbaden 2004.

[BöSK06] *Böhmann, T.; Schermann, M.; Krcmar, H.*: Application-Oriented Evaluation of the SDM Reference Model: Framework, Instantiation and Initial Findings. In: *(Eds.)*: Referenzmodellierung 2006 in conjunction with Multikonferenz Wirtschaftsinformatik 2006 (MKWI '06) 2006. Passau, Germany.

[Broc04] *vom Brocke, J.:* Internetbasierte Referenzmodellierung - State-of-the-Art und Entwicklungsperspektiven. In: Wirtschaftsinformatik 46 (2004) 5, pp. 390-404.

[Bung77] *Bunge, M.:* Ontology I: The furniture of the world. (Bd. 3), Reidel, Dordrecht 1977.

[Burr04] *Burr, W.:* Chancen und Risiken der Modularisierung von Dienstleistungen aus betriebswirtschaftlicher Sicht. 2004.

[BuSG03] *Bullinger, H.-J.; Scheer, A.-W.; Grieble, O.:* Service Engineering: Entwicklung und Gestaltung innovativer Dienstleistungen. Springer, Berlin 2003.

[BWFK04] *Böhmann, T.; Winkler, T.; Fogl, F.; Krcmar, H.:* Servicedatenmanagement für IT-Dienstleistungen: Ansatzpunkte für ein fachkonzeptionelles Referenzmodell. In: *Becker, J.; Delfmann, P. (Eds.)*: Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung. Physica, Heidelberg 2004, pp. 99-124.

[Coas37] *Coase, R.H.:* The Nature of The Firm. In: Economica New Series 4 (1937), pp. 386-405.

[Cunn05] Cunningham, W. (2005). Portland Pattern Repository. http://c2.com/ppr/, Accessed: 2005-Oct-27.

[DaJY05] Da Rold, C.; Jester, R.; Young, A. (2005). *The Future of Outsourcing* (Gartner Research Report). Stamford, CT, USA: Gartner Inc.

[FeLo03b] *Fettke, P.; Loos, P.*: Multiperspective Evaluation of Reference Models - Towards a Framework. In: *(Eds.)*: International Workshop on Conceptual Modeling Quality (IWCMQ'03) 2003. Chicago, Illinois, pp. 80-91.

[FeLo04a] *Fettke, P.; Loos, P.:* Referenzmodellierungsforschung. In: WIRTSCHAFTSINFORMATIK 46 (2004) 5, pp. 331-340.

[FeLo04b] Fettke, P.; Loos, P. (2004). *Systematische Erhebung von Referenzmodellen - Ergebnisse einer Voruntersuchung* (Working Paper 19). Mainz: Lehrstuhl für Wirtschaftsinformatik und BWL, ISYM - Information Systems & Management, Johannes Gutenberg-Universität.

[Fran00] *Frank, U.:* Modelle als Evaluationsobjekt: Einführung und Grundlegung. In: *Häntschel, I.; Heinrich, L.J. (Eds.)*: Evaluation und Evaluationsforschung in der Wirtschaftsinformatik. Oldenbourg, München 2000, pp. 339-352.

[Fran97] *Frank, U.:* Erfahrung, Erkenntnis und Wirklichkeitsgestaltung: Anmerkungen zur Rolle der Empirie in der Wirtschaftsinformatik. In: *Grün, O.; Heinrich, L.J. (Eds.)*: Wirtschaftsinformatik: Ergebnisse empirischer Forschung. Springer, Berlin 1997, pp. 21-35.

[Fran98a] *Frank, U.*: Die Evaluation von Artefakten: Eine zentrale Herausforderung der Wirtschaftsinformatik. In: *(Eds.)*: Workshop Evaluation und Evaluationsforschung in der Wirtschaftsinformatik 1998. Linz, Austria.

[GHJV94] *Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.:* Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Publishing Company, Reading, Massachusetts 1994.

[Greg06] *Gregor, S.:* The Nature of Theory in Information Systems. In: MIS Quarterly 30 (2006) 3, pp. 611-642.

[HDLA05] Hale, K.; De Souza, R.; Lo, T.; Adachi, Y. (2005). *Forecast: IT Services, Worldwide, 2005-2009* (Gartner Research Report): Gartner Inc.

[HMPR04] *Hevner, A.R.; March, S.T.; Park, J.; Ram, S.:* Design Science in Information Systems Research. In: MIS Quarterly 28 (2004) 1, pp. 77-105.

[Köhn05] Köhne, S. (2005). *Didaktischer Ansatz für das Blended Learning: Konzeption und Anwendung von Educational Patterns*. Dissertation, Universität Hohenheim.

[KoSF06] *Koch, S.; Strecker, S.; Frank, U.*: Conceptual Modelling as a New Entry in the Bazaar: The Open Model Approach. In: *Damiani, E.; Fitzgerald, B.; Scacchi, W.; Scotto, M.; Succi, G. (Eds.)*: Second International Conference on Open Source Systems (OSS) 2006. Como, Italy.

[LKDK03] Ludwig, H.; Keller, A.; Dan, A.; King, R.P.; Franck, R. (2003). *Web Service Level Agreement (WSLA) Language Specification 1.0*.

[MaMG02] *Markus, M.L.; Majchrzak, A.; Gasser, L.:* A Design Theory for Systems that support emergent Knowledge Processes. In: MIS Quarterly 26 (2002) 3, pp. 179-212.

[Mert95] *Mertens, P.:* Wirtschaftsinformatik: von den Moden zum Trends. In: *König, W. (Eds.)*: Wirtschaftsinformatik '95: Wettbewerbsfähigkeit, Innovation, Wirtschaftlichkeit. Physica-Verlag, Heidelberg 1995, pp. 25-64.

[PoBL05] *Pohl, K.; Böckle, G.; Linden, F.v.d.:* Software product line engineering: foundations, principles, and techniques Springer, Berlin 2005.

[RoGr02] *Rosemann, M.; Green, P.:* Developing a meta model for the Bunge-Wand-Weber ontological constructs. In: Information Systems 27 (2002), pp. 75-91.

[Ropo79] *Ropohl, G.:* Eine Systemtheorie der Technik: zur Grundlegung der allgemeinen Technologie. Carl Hanser Verlag, München 1979.

[SaLe02] *Sarker, S.; Lee, A.S.:* Using a Positivist Case Research Methodology to Test Three Competing Theories-In-Use of Business Process Reengineering. In: Journal of the Association for Information Systems 2 (2002) 7, pp. 1-74.

[Sche98] *Scheer, A.-W.:* Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse. 2 ed., Springer, Berlin 1998.

[Schu03] *Schumacher, M.:* Security engineering with patterns: origins, theoretical models, and new applications. Springer, Berlin 2003.

[Simo69] *Simon, H.A.:* The Sciences of the Artificial. The M.I.T. Press, Cambridge, Massachusetts, USA 1969.

[StMJ00] *Sturm, R.; Morris, W.; Jander, M.:* Foundations of Service Level Management. Indianapolis, SAMS 2000.

[Stra96] *Strahringer, S.:* Metamodellierung als Instrument des Methodenvergleichs: Eine Evaluierung am Beispiel objektorientierter Analysemethoden. Shaker, Aachen 1996.

[Thom06] *Thomas, O.*: Version Management for Reference Models: Design and Implementation. In: *(Eds.)*: Referenzmodellierung 2006 in conjunction with Multikonferenz Wirtschaftsinformatik 2006 (MKWI '06) 2006. Passau, Germany.

[WaWE92] *Walls, J.G.; Widmeyer, G.R.; El Sawy, O.A.:* Building an Information System Design Theory for Vigilant EIS. In: Information Systems Research 3 (1992) 1, pp. 36-59.

[Wiss94] *Wissenschaftliche Kommission Wirtschaftsinformatik:* Profil der Wirtschaftsinformatik. In: WIRTSCHAFTSINFORMATIK 36 (1994) 1, pp. 80-81.