UTILIZING PATTERNS IN DEVELOPING DESIGN THEORIES

Completed Research Paper

Sabine Buckl
Chair for Software Engineering of Business Information Systems
sabine.buckl@mytum.de

Florian Matthes
Chair for Software Engineering of Business Information Systems
matthes@mytum.de

Christian M. Schweda
Chair for Software Engineering of Business Information Systems,
Technische Universität München
Boltzmannstr. 3, 85748 Garching, Germany
christian.m.schweda@mytum.de

Abstract

Aside from specialized design artifacts solving 'wicked' problems in a distinct organizational context, prescriptive (design) theories for solving classes of problems are well-accepted results of design science research. Especially for intricate problem fields, as IT service management or EA management, the development of such theories remains a highly complex research endeavor spanning a longer period of time. In this paper, we discuss how patterns, i.e. practice-proven, re-usable solutions to recurring problems, can be used to ease the establishment of a design theory, and to provide practice-relevant intermediary results prior to theory completion.

Keywords: Design theories, design research, patterns

Motivation

There is an ongoing discussion on the importance of theories as results of information systems (IS) research. In her recent work Gregor (2006) provides a framework for classifying theories in IS research, namely theories for a) *analysis and description*, b) *explaining*, c) *predicting*, d) *explaining and predicting*, and e) *prescription*, sometimes also alluded to as "design theories". With the bordering position of IS between social sciences (especially management sciences) and engineering sciences (e.g. software engineering), theories of all natures may apply in the field. Nevertheless, especially in the IS community, as opposed to the design-oriented discipline of 'Wirtschaftsinformatik' in German speaking countries, prescriptive theories often do not reach dissemination in leading journals. Applying her taxonomy Gregor (2006) shows that the vast majority (over 60%) of theories as published in the journals *Management Information Systems Quarterly* and *Information Systems Research* from March 2003 to June 2004 are explanatory and predictive theories. This fact well reflects an ongoing discussion in design science research, where March and Smith (1995) and Hevner et al. (2004) promulgate an understanding of theories restricted to explanatory and predictive ones, while pure *design theories* are not alluded to as "theories".

Irrespective the aforementioned controversies on theorizing in design science research, the concept of *design theory* enjoys a renaissance in the metanarrative of IS research, as recent publications of Gregor and Jones (2007), Schermann et al. (2007), or Gehlert et al. (2009) show. There may be many different explanations for the fact that such theories are increasingly en-vogue today, while they have been 'wallflowers' for a long period since their first appearance in 1992 in the work of Walls et al. (1992). To us, the following statement of Friedman (2003) provides a worthwhile indication towards a possible explanation:

> There comes a moment in the evolution of every field or discipline, when central intellectual issues come into the focus as the field and the discipline [... shifts ...] to an area of reasoned inquiry. At such a time, scholars, scientists, researchers and their students begin to focus articulate attention on such issues as research methods, methodology [...and philosophy of science] through which a research field takes shape. In many fields today, this also entails the articulate study of theory construction.

In 2009 Steininger et al. (2009) conducted a literature analysis in leading journals of IS research and 'Wirtschaftsinformatik', respectively. They were able to show that activities in the topic areas of corresponding research fields show typical temporal patterns, of which two prominent ones are, the patterns *recurring mode* and *oscillating topic*. Abstaining from in-depth considerations on the distinction between these two pattern types, we summarize the findings of Steininger et al. in the sense that 70% of topics in IS research and 34% of topics in Wirtschaftsinformatik-research are *recurring*, i.e. have undergone at least one previous period of heavy research engagement. A related analysis undertaken by Baskerville and Myers (2009) discusses effects of repeated *upswing* and *downswing* with respect to topics in IS research, further calling for a pro-active management of what is called "fashion waves" there. Reflecting these findings against the statement of Friedman (2003) it is sensible to assume that quite a few topics in IS research have evolved over a longer period of time and may thus be in the midst of that aforementioned very moment, that "entails the articulate study of theory construction".

As more and more IS researchers are entering the field of design theory development, the metanarrative of IS research can and in turn must provide novel and more detailed responses to the question how methods for building design theories should look like. This applies against the background of the increasing openness and permeability of the 'wall' separating the design-oriented research tradition in German 'Wirtschaftsinformatik' from the quantitative research orientation in IS research, see Niehaves (2007). In this light, methods for developing design theories must specifically account for the aspects of *rigor* and *relevance* that according to Hevner et al. (2004) shape the environment in which design science research takes place. The methods must nevertheless also relate to the pragmatics of scientific research in practice-relevant and possibly industry-funded research projects. On the one hand, the close interaction with an organization willing to practically apply the prescriptions of the design theory opens the door for *developing case-studies* in line with van Aken (2004) employing an intrinsically motivated industry (practice) partner. On the other hand, an industry-funded research project usually underlies the partnering organization's pace, and may hence be obliged to deliver results early, which effectively aggravates the development of comprehensive theoretical underpinnings. In this sense, the advantage of having an industry partner is likely to turn into a disadvantage for the research endeavor, at worst reducing the endeavor to 'routine design' that in line with Hevner et al. (2004) must be distinguished from design research. As a consequence, theory building often accompanies but does not directly provide input to the industry-funded project, which in turn is later re-used as

extracting case-study in terms of van Aken (2004). In the light of the above discussions, the research gap that this paper targets can be summarized to the following research question:

> *How does a method for building design theories in close interaction with an organization (willing to apply these theories) look like?*

Illustrating the field that above research question is aiming at, we quote a figure from Gehlert et al. (2009) that presents a conceptual framework of the elements of theorizing in design research in accordance to Nunamaker et al. (1991). In this framework (cf. Figure 1) we highlight the area of research that this paper targets. More precisely, we aim at providing pattern-based mechanisms for theory building, i.e. for deriving theories from actual design.

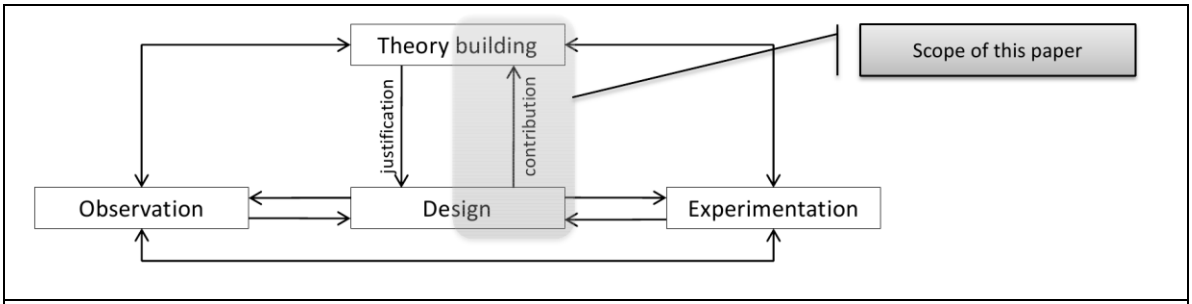

Figure 1. Elements of theorizing in design research, see Nunamaker (1991), and scope of this paper

Approaching the research question, the article revisits the metanarrative of IS research in respect to frameworks as well as development methods for design theories in Section *Design Theorizing in IS Research* and points out aspects relevant for devising a pattern-based development method for design theories. Complementing, an introduction into the field of pattern development is given and aspects of research via patterns as well as pattern languages are discussed in Section *Pattern-Based Development of Design Theories*. These aspects are then revisited to further describe a pattern-based development method for design theories. In final Section *Critical Reflection and Outlook*, we critically summarize advantages and shortcomings of the development method, and give indications on topics for future research.

Design Theorizing in IS Research

IS research as well as its German pendant "Wirtschaftsinformatik" are often alluded to as disciplines with a multi-methodological background. More precisely, a plethora of research methods is applied in IS research, a fact that may be ascribed to the philosophical positioning of the discipline in the midst between management and social sciences on the one hand, and engineering sciences on the other hand. Wilde and Hess (2007) investigate the pluralism of methods and compare the situation in "Wirtschaftsinformatik" to the one in IS research. While the comparison shows a significant difference in methodological preoccupation, both disciplines present a broad spectrum of research methods ranging from empirical to engineering ones. Narrowing down the focus to design research in the IS context, the methodological plurality does not significantly change. Hevner et al. (2004) show that various methods are applied to design and evaluate IS research outcomes. In the light of this variability, it is not surprising that multiple methods for developing design theories are investigated in literature.

One cannot discuss design theories, at least in IS research, without referring to Walls et al. (1992), who were the first to introduce the notion of prescriptive theories into the research field. Central to their considerations is a conceptual framework outlining the constituents of a design theory: *kernel theories*, *meta-requirements*, *meta-design*, *design method* as well as *testable hypotheses* targeting the design product and process, respectively. In this framework, the important principle of *metaization* is devised, especially as a principle aiming at abstracting *situated* problems to the level of *general classes* of problems. Further, Walls et al. describe that design goals are contingencies for design theories, i.e. they are intrinsic to prescriptive theories. In this respect the authors outline how explanatory laws and rules can be reformulated to read as design prescriptions helping to fulfill design goals. Beyond the aforementioned basic considerations, Walls et al. (1992) discuss selected aspects of theory development as reflected in the design processes. Prominent in these discussions is the role of information systems development,

which is regarded an essential cornerstone in devising a design theory. More precisely, Walls et al. introduce the generalized conception of the *information system generator*, a configurable information system that incorporates the essence of an underlying design theory. By configuring such a generator, the researchers apply the design theory in the situated context, i.e. instantiate the theory's meta-design into an actual design.

The work of Walls et al. has been furthered ever since in different publications, e.g. by Gregor and Jones (2007), by Schermann et al. (2007), or by Markus et al. (2002) as well as revisited by the authors themselves (cf. Walls et al. 2004). The former two publications are of crucial importance for our pattern-based development method for design theories and are alluded to in detail in Section *Pattern-Based Development of Design Theories*. To avoid unnecessary repetitions, we abstain from discussing these approaches here. Markus et al. (2002) refine the conceptual framework of Walls et al. and revisit the constituting elements. Especially, the notion of the meta-design is subjected to a different understanding as a *type of system solution*. Central to the considerations of Markus et al. is the question, how design theories can be evaluated for their truthfulness. As answer to the question, the authors devise that the utility of the type of system solutions in respect to solving problems from the theory's class of problems provides an indication for the truthfulness of the theory. Nevertheless, the considerations remain abstract and general, such that applying researchers have to find an operationalization of this *utility conception of truth* themselves.

Further discussions on the principle of *metaization* are undertaken in the work of Vaishnavi and Kuchler (2004), where they describe the different levels of *metaization* – called *abstraction* – that are applied to get from situated implementations (solutions) to design theories. Figure 2 summarizes their point of view. They explain that the expected level of abstraction in the research outcome is determined by the community that is conducting the research. Linking back to the research question of this paper, especially the intermediary level of *knowledge as operational principles* may be of interest in respect to practice-driven research. While we cannot expect that the practitioners abstract their implementations towards 'full-blown' theories, knowledge sharing and communication processes within a group of practitioners may lead to abstracted representations of implementations on the level of operational principles. In this sense, theorizing in close cooperation with practitioners can harvest this knowledge as operational principles instead of developing and abstracting theories from situated implementations.

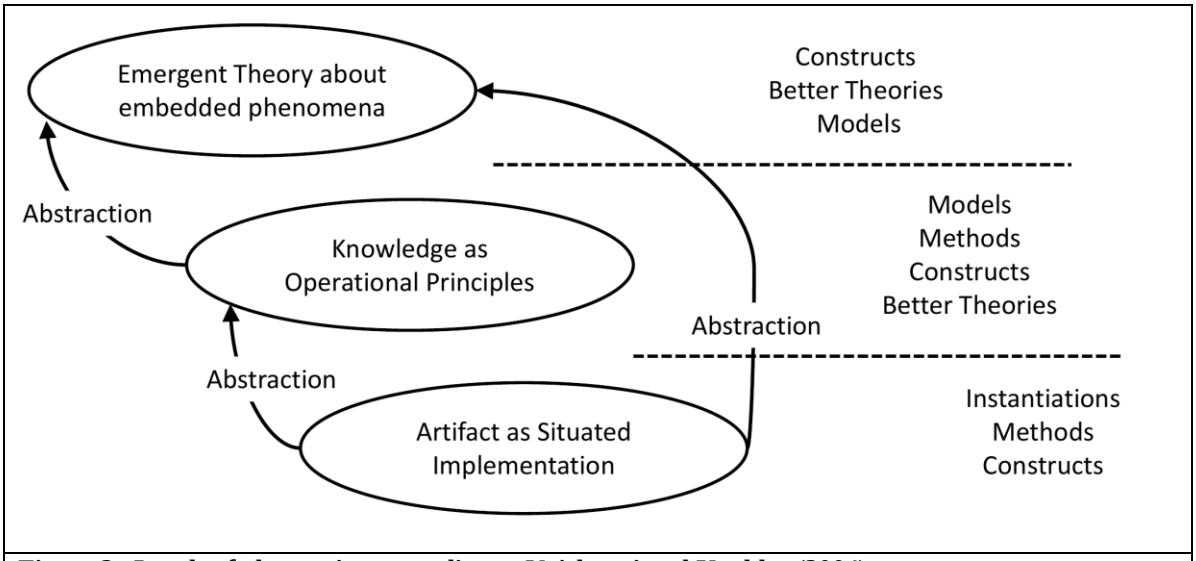

Figure 2. Levels of abstraction according to Vaishnavi and Kuchler (2004)

Carlsson (2007) sheds a slightly different light on IS design science research, focusing on the addressees of design knowledge as well as the types of knowledge that such research can create. As addressees of design knowledge that means as community to evaluate the relevance of design theories, Carlsson (2007) names "IS professionals and managers responsible for IS-supported and enabled processes". In this sense, a design theory must address a problem relevant to this group in a way that the theory's prescriptions are accessible to the group's members. Necessarily, this

raises some terminological and linguistic issues, as theory representation should employ prevalent termini in the practitioners' community. These termini nevertheless frequently lack a precise definition and are often used ambiguously. Put in other words, design science research is *norma normata*, when it comes to communicating its results to the addressees, while the parts of the results must also be *norma normans* refining and providing disambiguation for the termini of the universe of discourse. The latter fact is mirrored in another statement of Carlsson (2007), where he calls IS education the second important group that design knowledge addresses. Detailing on the types of design knowledge, Carlsson (2007) states that such knowledge is "practical knowledge for the design and realization of IS initiatives [or for] the improvement of the performance of existing IS". This position seems to us to shape the very core of design theories, while Hevner et al. (2004) would refute "performance improvements" to be genuine design knowledge linking back to the question of relevance. We pick up the argument of Carlsson (2007), that such knowledge is very important for practitioners and should hence be targeted in design science research. Further detailing on the types of design knowledge, Carlsson (2007) stresses the notion of the *design proposition* which is built around a concept for solving a class of problems by actions. Two types of propositions can thereby be distinguished, namely *algorithmic design propositions* and *heuristic design propositions*. Propositions of the former type are strongly formalized and assure a good design outcome, while heuristic propositions are most likely better suited for addressing non-technical issues. With respect to heuristic design propositions Carlsson (2007) further emphasizes the difficulties of proving their effects conclusively.

Venable (2006) discusses how theorizing relates to design science research, coining the general statement that "theorizing occurs before, during, throughout and at the end, and as a result of design science research". In this vein, Venable sees theories as link between different research activities over time. More precisely, theorizing can be understood as a central activity providing input to and receiving input from *problem diagnosis*, *technology invention*, and *technology evaluation*. In all these activities, the situated outcomes and artifacts can be abstracted to general problem or solution descriptions in the light of underlying kernel theories. This sheds a different light on the general statement of Venable (2006), especially when theory building throughout design science research is taken into account. In line with this argument, a theory can be developed iteratively in a sort of hermeneutic cycle. A preliminary theory thereby provides the conceptual basis and framework to understand the situated problem that should be solved next. In contrary, the newly devised solution contributes to the theory's body of knowledge, and prepares the next cycle. As a matter of fact, Venable (2006) leaves open the question of frequency in respect to the iterations, i.e. does not make prescriptions on the duration of a single iteration. From this, we advocate a development method of 'small steps', i.e. opt for an iterative development of a theory with frequent application-abstraction cycles. Emphasizing on the problem of evaluating design theories during the iterative development, Venable (2006) offers a promising and sophisticated 'turn of perspectives'. Central to this turn is the understanding of design theories as utility theories, leading to a reformulation of their prescriptions from "to achieve A do B" to "in order to achieve A it is useful to do B". What at first sight seems to be simple rephrasing, in fact opens design theory evaluation to the variety of evaluation techniques in IS research that focus on utility. Further, the utility perspective allows accounting for linguistic and terminological aspects as discussed above.

The paper of Eisenhardt and Graebner (2007) is devoted to *theory building from cases*, a topic which is increasingly prominent. The authors motivate their discussions with the fact that albeit this recent prominence, papers pursuing theory building from cases may fall in disgrace during review, as the applied methods are often misunderstood. In response to this, Eisenhardt and Graebner (2007) provide hints and suggestions that should be helpful in overcoming this difficulty. At this point a reader might argue that theory building from cases is not the only choice for developing prescriptive theories, we nevertheless regard this research strategy to be the natural choice, especially in research projects with strong industry and practice ties. Against that background, we summarize some key suggestions and arguments of Eisenhardt and Graebner (2007), although there might be other ways of theory development, as e.g. *natural inquiry*, for which these suggestions may only be of minor relevance. Further, we focus on *phenomenon*-driven research in terms of Eisenhardt and Graebner (2007), i.e. a type of research that addresses broadly scoped research questions around the barycenter of an IS-relevant phenomenon. First step of a corresponding research endeavor would be an inquiry into and argumentative justification of the phenomenon's importance, as well as an exposition showing that no viable theory on that phenomenon is given in literature. In the second (main) step of research, different practice cases are sampled to build and develop the body of the theory. As Eisenhardt and Graebner (2007) state, these cases do not have to be *representative* for a population, but can be selected in a way that they provide theoretical insights, as "revelations of an unusual [aspect of the] phenomenon". In a final step, the (emergent) theory has to be described in a way that an interested and willing reader is able to understand the theory as well as the underlying practical evidence it is based upon. This leads to the question how to report on the manifold cases in multiple-case research. Eisenhardt and Graebner (2007) argue that in multiple-case

research the concise and complete rendering of the cases' stories is not feasible without 'ballooning' the text and confounding the reader. In this vein, they advocate to use extensive "construct tables" and "other visual devices" to summarize the case evidence underlying a theory or more precisely self-contained parts thereof. We fully agree with latter argument and will show in Section *Pattern-Based Development of Design Theories*, how selected structuring principles used in describing patterns resemble such "visual devices".

In their paper Gehlert et al. (2009) focus on methodical guidance for the process of theorizing in design research. Revisiting existing approaches, most notable the ones discussed above, the author diagnose that all approaches fail to provide this guidance, more precisely guidance on the question on how to relate design decision to theoretical justification. In this sense, the *research gap* addressed by Gehlert et al. (2009) is the 'justificatory counterpart' of the gap addressed in this paper (see Figure 1). To close the aforementioned gap, Gehlert et al. (2009) provide guidelines how design decisions can be documented against the justificatory background of a theory[1]. Central to the documentation approach is the decomposition of design goals into smaller goals and further to a level, where concrete requirements can be derived. The entirety of requirements devised in this way reflects the plethora of (possibly conflicting) forces framing the design space. An actual design must resolve the conflicts between the forces by selecting one possible position reflected in the forces. To facilitate the resolution of conflicts, theory-driven design can call upon the theoretical contribution that introduces constructs as relevant conceptualizations of the design space and formulates *indicators*, i.e. properties, to them. These indicators exhibit distinct positions that can be selected during the design process. Put in other words, a theory describes relevant elements in the design space and the options to "configure" them during design. Linking these configuration options – positions in terms of Gehlert et al. (2009) – to the corresponding forces and transitively to the design requirements and goals can help to justify the design decisions. Picking up this educated method for theory-supported or theory-based design of Gehlert et al. (2009) one can conversely derive that a 'good' prescriptive theory must

- provide appropriate conceptualizations of the solution space, i.e. constructs and their properties,
- make the forces framing the solution space explicit, and
- describe the relationships between the constructs and their properties on the one hand and the corresponding forces on the other hand.

In the subsequent section, we revisit the fundamentals of patterns and pattern languages, and show to which extent basic and well-established principles in that domain can be applied to support design theory development.

Pattern-Based Development of Design Theories

Documenting good practice solutions to recurring problems in a specific context, i.e. knowledge on *operational principles* in the sense of Vaishnavi and Kuchler (2004), by so-called *patterns* is a commonly accepted way to facilitate knowledge abstraction and dissemination in design-intensive domains. The idea of patterns originates from construction and urban planning and was first introduced by Alexander et al. (1972, 1977). Related fields, which have since then successfully adopted the idea of patterns, are e.g. software engineering by Gamma et al. (1994), software architectures as presented by Buschmann et al. (1996), or project management by DeMarco et al. (2008). In all these disciplines patterns present operational knowledge harvested from practice, i.e. patterns are neither invented nor developed, but observed. Giving account to this fact, Coplien (1996) establishes, what has become widely known as the *rule of three*: a documented pattern must provide reference to at least three *known uses* in practice. Against the background of being re-usable solutions, pattern are often demanded to provide illustrative *examples* of the problem that they are intended to address. Over the years additional structuring principles for pattern descriptions have developed, leading to a typical *pattern format* according to Buschmann et al. (1996) consisting of a description of the *addressed problem* and the intended *application context* for the *solution*. A pattern further identifies *driving forces*, denotes *known usages* and *consequences*, and makes *relations* to other patterns explicit.

While the importance of patterns in disciplines, as software engineering, is unquestioned, the role of patterns in IS or "Wirtschaftsinformatik" research is subject to controversies. In the discussion and opinion section of the Business & Information Systems Engineering journal, researchers and practitioners recently outlined their attitudes towards

[1] The work is thereby not confined to design theories, but can draw design justification from theories of any nature in the sense of Gregor (2006).

patterns for IS research (Winter et al. 2009). Thereby, special emphasis was put on advantages and disadvantages from the practical nature of pattern-based research. Discussants of non-academic provenience, as e.g. Keller, outline that the rule of three is especially in the academic context regarded as a lack of originality, such that patterns are not understood as 'true' research. Challenging former statements, Fettke and Loos argue that nowadays no distinction between *reference modeling*, which according to Becker et al. (2007) is a classical topic of IS research, and *pattern identification* should be made. While this position seems fully acceptable to us, Fettke and Loos go even further in Winter et al. (2009), stating that the terms "reference model" and "pattern" have become interchangeable. In the light of the original understanding of patterns as introduced by Alexander (1972), we respectfully disagree as reference models are invented and developed, while patterns are observed and documented. This latter understanding of what a pattern is reverberates through the remainder of this article, as we see no need to amalgamate two terms that are rooted in different cognitive backgrounds[2].

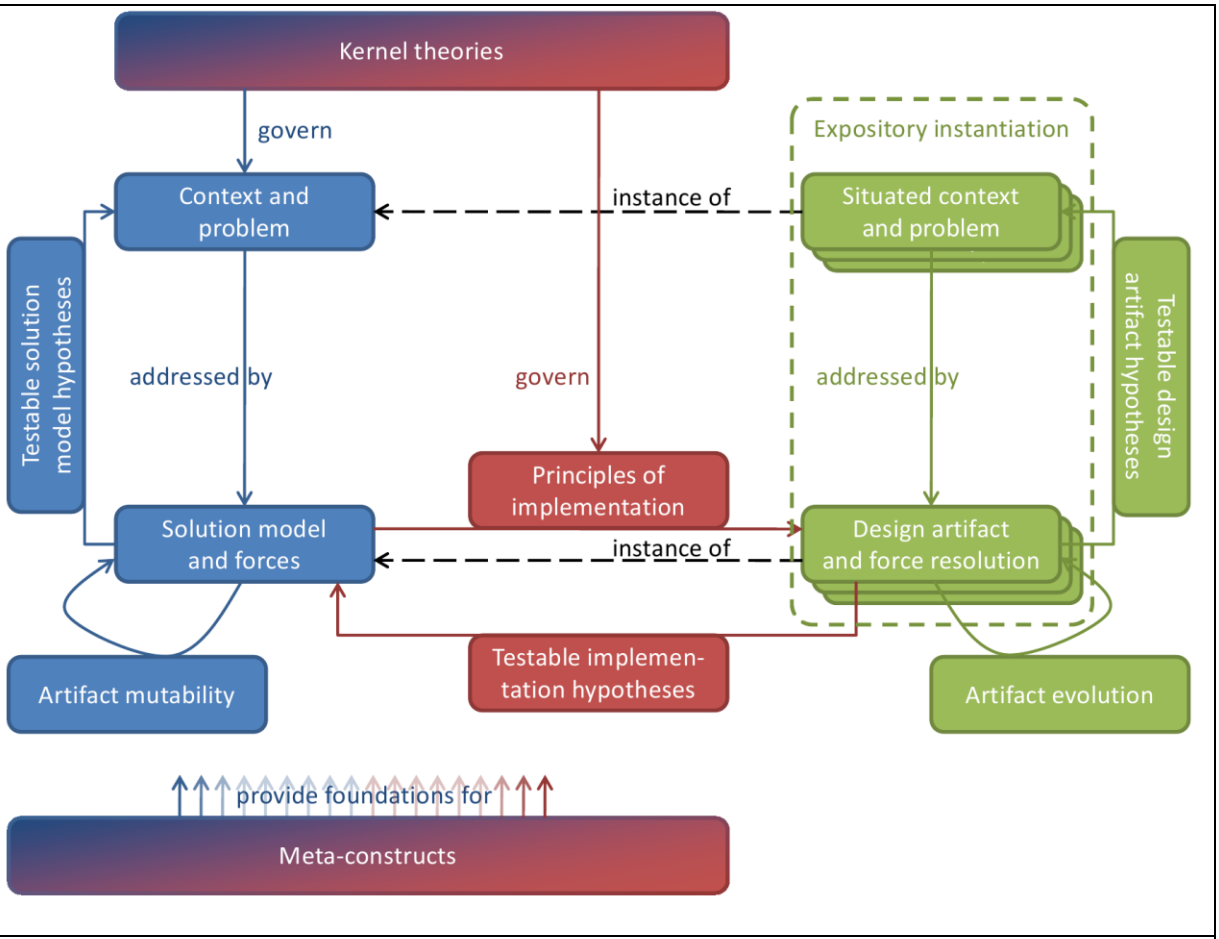

Figure 3. Conceptual framework for a pattern-based design theory

Patterns, which can be regarded as coherent and self-contained design entities that describe a solution to a specific problem, can be understood as elementary design principles in line with Markus et al. (2002). In this sense, it seems reasonable to interpret patterns as potential building blocks for a design theory. Support for this interpretation can be found by Schermann et al. (2006), who sketch how patterns can be employed in describing design theories. Prior to revisiting their ideas as starting point for devising our development method for prescriptive theories, we provide a conceptual framework for design theories. This framework is grounded in the work of Walls et al. (1992), but is strongly influenced by Gregor and Jones' (2007) inquiry into the anatomy of design theories. Figure 3 outlines the framework's core constituents, which are further detailed subsequently. Thereby, we distinguish process-related

[2] Admittedly, the backgrounds, while distinct, are strongly interrelated.

constituents of the theory (red symbols), artifact-related ones (blue symbols), and elements resulting from the instantiation, i.e. application of the design theory (green symbols).

Kernel theories, named originally so by Walls et al. (1992) provide the underpinnings on which the other constituents of the design theory, especially the *context* and *problem* description and the *principles of implementation* are grounded. We stick to the original term "kernel theory" instead of applying the term "justificatory knowledge" used by Gregor and Jones (2007) or "theory references" used by Schermann et al. (2006). Thereby, we want to emphasize that kernel theories not necessarily have to be justificatory, but may also have an analytical, explanatory, or predictive character or may even be other design theories providing a valuable basis to build the design theory upon. Especially theories of the latter kind may be of high importance in a pattern-based design theory, where other pattern-based design prescriptions may serve as kernel theories.

A design theory is dedicated to certain classes or problems, which are addressed by the theory, as well as dedicated usage contexts. Introduced with the term "meta-requirements" in the work of Walls et al. (1992), and further referred to as "purpose and scope" of a design theory in the work of Gregor and Jones (2007), Schermann et al. (2006) give the term "context and problem" to describe application area and intended environment of a design theory. Latter term (**context and problem**) seems to us most appropriate for pattern-based design theories in line with the context-problem-dichotomy of re-usable solution building blocks, aka patterns.

The classes of artifacts, which are hypothesized to address the problems in the corresponding contexts, are called "meta design" by Walls et al. (1992). Gregor and Jones (2007) utilize the term of "principles of form and function". We resort to the term **solution model and forces** in extension of the terminology of Schermann et al. (2006). Thereby, we emphasize on the fact that the solutions devised by the design theory should in line with the discussions of Gehlert et al. (2009) also allude to the forces framing the solution space. Again, the structure of the conceptual framework mirrors the *pattern format* described by Buschmann (1996).

Rounding up the artifact-related constituents of a design theory, we allude to the concept of **artifact mutability**. This concept was not part of the initial framework of Walls et al. (1992), but has later been proposed by Gregor and Jones (2007) in response to the fact that IS artifacts most likely have to adapt to their ever changing environment. In this respect a design theory for such an artifact according to Gregor and Jones should encompass information how to evolve and refine the artifact. In a pattern-based design theory, the relationships (as the *refined by* relationship, see below) linking different patterns can provide worthwhile information on how to evolve a design artifact.

Central design process-related constituent of a design theory is, what Walls et al. (1992) call "design method". This method describes the procedure for constructing the design product from the solution model provided by the theory. Venable (2006) argues that the term *construction* should in this context be interpreted more abstract applying to a broad range of activities used to develop a particular design artifact. In line with this argumentation, we apply the term **principles of implementation** as used by Gregor and Jones (2007) instead. In a pattern-based design theory, these principles may be fairly general, and show up as rules and guidelines for selecting and integrating patterns.

Coming to the 'instantiation side', recent publications on design theories, as e.g. the one of Gregor and Jones (2007) argue that the term of *design product* is of limited generality. Resorting to the more abstract notion of *artifact* as also put forward by March and Smith (1995) and quoted by Hevner et al. (2004), Gregor and Jones (2007) propose the generic term **design artifact** as the outcome of applying a design theory. This proposal is gratefully adopted in the conceptual framework that we devise here, especially in the light of the variability of design outcomes. A theory-supported design endeavor needs not to target a product, as e.g. an information system, but can also provide more generic outcomes as *constructs*, *models*, *principles*, and *methods*.

Contrasting the initial framework of Walls et al. (1992), Gregor and Jones (2007) demand an **expository instantiation** to be part of a design theory. This instantiation exemplarily applies the principles of implementation to devise, based on the solution model and forces, a design artifact addressing the *situated context and problem*. Aforementioned demand reflects a statement of Hevner et al. (2004) saying that "design research must produce a viable artifact in the form of a construct, model, method, or instantiation". In the context of a pattern-based design theory, such expository instantiation can at least partially be mirrored in the examples given in the single patterns.

By adding the concept of the **artifact evolution**, we have applied another structural change. It is of minor relevance for the design theory itself but added for reasons of completeness. It complements the *artifact mutability* that introduces the capability to adapt the design artifact to a changing environment with the actual trajectory of adaptations that occurs on the instance, i.e., implementation level. Furthermore, different types of **testable hypotheses** are used to relate concepts in the design theory. These hypotheses, that differing to the model of Gregor

and Jones are distinguished into distinct types, are of crucial relevance, during theory evaluation. In contrast, during theory building, we can expect the hypotheses, especially the **testable solution model hypotheses** to be a by-product of pattern documentation. More precisely, each pattern itself is a solution model hypothesis, stating that a specific class of problems situated in a distinct context can be addressed by a heuristic solution.

Concluding the exposition of our conceptual framework for design theories, we pick up a statement of Gregor and Jones emphasizing the critical omission of "constructs" in the original framework of Walls et al. (1992). Such constructs are necessary to build a consistent and shared terminology for the design domain (cf. *norma normans* argument in Section *Design Theorizing in IS Research*) used throughout the exposition of the theory. To avoid an ambiguous term, which could be mixed up with the construct concept[3] from March and Smith (1995), we utilize the notion of **meta-constructs** instead. The formulation of a consistent set of meta-constructs is a central challenge in developing a pattern-based design theory. While after careful revision, one can expect that a single pattern employs a consistent terminology, inter-pattern consistency of terms relies on the notion of pattern relationships.

The aforementioned conceptual framework lays the basis for pattern-based design theory development in a twofold way: firstly, the design theory itself shows typical characteristics of a pattern as e.g. visible in the context, problem, and solution description. Secondly, as discussed along the constituents of the theory, many of these constituents can be composed from multiple patterns, if the patterns are interrelated. This links back to an idea presented by Alexander et al. (1977), the idea of a *pattern language*, i.e. a system of interrelated patterns. The general applicability of this idea in the context of design theory construction has been explored by Schermann et al. (2006). Nevertheless, they focused on *untyped* relationships, denoted as "see also" in Alexander et al. (1977) or "related patterns" in Coplien (1996), while semantically richer relationship types between patterns have been devised in literature. Noble (1998) provides the following list of relationship types:

Table 1. Relationship types of patterns according to Noble (1998)	
Used by	A smaller pattern is used by a larger pattern
Refined by	A general pattern is refined by a specific pattern
Variant	A variant pattern refines a more well-known pattern
Variant Uses	A variant of one pattern uses another pattern
Similar	A pattern is similar to another pattern
Combines	Two patterns combine to solve a single problem
Requires	A pattern requires the solution of another pattern
Tiling	A pattern uses itself
Sequence of Elaboration	A sequence of patterns from the simple to the complex

Relationships of the above types can contribute to the formulation of a pattern-based design theory in manifold ways. Exemplifying this, we discuss selected relationship types and their roles in theory development subsequently:

- **Used by** The usage relationship describes that and how a larger pattern, sometimes alluded to as 'umbrella pattern', employs another pattern for solving a sub-problem of the 'umbrella problem'. Building on the usage relationship, it is possible to aggregate solution building blocks into a comprehensive solution for a coarse grained problem.
- **Refined by** A refining pattern targets a similar problem and context as its 'parent', but provides a more detailed solution model or outlines a broader variety of forces that have to be balanced. Relationships of that type contribute to the artifact mutability in the design theory, as they sketch possible trajectories for refining the design artifact.

[3] This construct concept would contrastingly be part of the design artifact.

- **Variant** A pattern variant targets a similar or closely related problem and context as the initial pattern, providing a solution that not far differs from the original one. Relationships of that type may help to refine the context and problem description by both broadening the scope of the corresponding classes and raising further dimensions of distinction.
- **Sequence of Elaboration** A sequence of patterns that describes more and more elaborate solutions for a similar problem and context may be understood as contribution to artifact mutability by providing possible ways to evolve the design artifact and to bring it to a more 'mature' level.

Due to their observational nature patterns typically vary in respect to their level of granularity and abstraction. This especially becomes obvious, when the patterns are to be composed into a prescriptive theory for a certain research field. Utilizing *umbrella patterns*, it is possible to address granularity variability, which allows to develop a usage hierarchy of patterns. In its most extreme form, establishing umbrella patterns is likely to lead to one embracing pattern, which – in line with the above established conceptual framework – would comprise a description of the design theory itself. Revisiting terminological issues as repeatedly alluded to in Section *Design Theorizing in IS Research*, we propose an extension to above list of relationship types, namely two relationships **linguistically compatible** and **linguistically diverse** to indicate that two patterns employ compatible or conflicting terminologies, respectively. In contrast to the relationship types outlined by Noble (1998), one cannot expect the two linguistic relationships to emerge during the maturation of the pattern language. Therefore, we demand that a researcher conducting pattern-based design theory development keeps these relationships up-to-date, while the pattern language evolves.

The above considerations on design theorizing with patterns summarize to a development method[4] consisting of the activities **observe and document design patterns**, **elicit pattern terminology**, **devise pattern-relationships**, and **derive terminological compatibility relationships**. With exception of the last one, these activities can be conducted in close cooperation with the industry partner of the research project. This is especially true, as each activity outputs worthwhile intermediary results that the partnering organization can readily re-use under the premise that the research projects targets a problem of relevance for the partner. With the emerging pattern language on the one hand and the indications of the patterns' utility gained from the application in the context of the partnering organization on the other hand, a researcher has a solid and sound basis to finally devise a comprehensive design theory. At least during the final step of theory formulation, the formerly documented terminological issues can be resolved in favor of a well-defined and grounded terminology.

The framework and method introduced above can be utilized to develop design theories based on patterns in a variety of different application domains, which a plurality of different research methods. Nevertheless, it makes some epistemological and ontological assumptions, which a research based on this framework should share. The explication of ontological and epistemological assumptions is of special importance since, the internationalization of IS research in recent years (Niehaves 2006) and the methodological and paradigmatic diversity (Chen and Hirschheim 2004) yields new challenges for researchers. To facilitate collaborative work between researchers and between academia and industry as well as evaluation of research in respect to the appropriateness of the research method choice and application, the epistemological and ontological assumptions underlying the research findings have to be made explicit. The pattern-based development of design theories presented above is based on ontological and epistemological assumptions, which are detailed subsequently alongside the epistemological framework developed by Becker and Niehaves (2007). The frameworks rises the questions of i) *ontology*, ii) *subject-object-relationship*, iii) *concept of truth*, iv) *origin of cognition*, and v) *methodology*. Answering the first of the aforementioned questions, a researcher utilizing the approach presented above should assume a Kantianism perspective. Taking possible fields of investigation in IS research into account, we assume that things exist, which are independent from human consciousness (so-called "noumena") as well as things, which contrastingly depend on human consciousness (so-called "phenomena") (Kant 2008). Furthermore, we determine the relationship between cognition and the object of cognition to be determined by the subject, following a *constructivist* assumption. Truth is assumed to be determined through *consensus theory of truth*, i.e. a statement is true if it is rationally acceptable under ideal and optimal conditions for a community or group. In accordance with the relevance of collaboration between industry and academics for IS research, we assume a *Kantianism* position regarding the origin of cognition considering both, intellect and experience as complementing sources thereof. As the objects of investigation in IS

[4] Speaking more precisely, we should talk of a *meta-method*, as it does not make prescriptions on the actual implementation of the single activities.

research are often intricate and comprehensive problems fields, we regard the fundamental methodology aspects of *hermeneutics* as presented by Gadamer (1975) to be beneficial.

Critical Reflection and Outlook

This paper contributes to the metanarrative of IS research, especially to the field of developing prescriptive theories, i.e. design theories. In our opinion, new methods have to be added to the methodical canon of the field in response to the recently increasing interest in design theorizing. The method presented in this paper is developed to address the challenges of developing design theories by resorting to methods and techniques from the field of pattern development. These techniques are related to well-established and -grounded methods from design science research leading to a new methodical framework that accounts for rigor and relevance in the potentially theory-adverse environment of an industry-funded application-centric research project. Along the way of devising this development methods, some "mythical" statements on the role of patterns in IS research are critically reflected and a clear understanding of the "can-do's" and the "cannot do's" of patterns in design science research is advocated for.

In the light of aforementioned achievements, we can say that this paper provides a starting point, but more challenges are await in the field. While having laid strong emphasis on the pattern perspective on design theory development, the guidelines and prescriptions put forward in the paper remain on a general and abstract level. This may be ascribed to both the embracing nature of the field of theory building and the fact that practical applications of the proposed guidelines are yet to be successfully completed. With the completions being on the horizon, we weighted the options to present preliminary guidelines or a *theory holon*, and preferred to contribute to a relevant discussion now instead of providing more rigorous results. Once again refraining the words of Friedman (2003) as quoted in Section *Motivation*, our early action might be justified with the fact that our "research discipline shifted to an area of reasoned inquiry", such that we had to "focus articulate attention on such issues as research methods [...] through which a research field takes shape". Against that background, we felt urged to subject our ideas to discussions, thereby seeking to collaborate further the guideline and canonize them into the metanarrative of design theory development.

Above discussions of the liabilities of this paper also sketch the way that is to go next. Upon the completion of our current design theorizing research projects, the experiences gained during the conduction have to be revisited and best-practice methods have to be documented as detailed guidelines grounded in research methodical kernel theories. In this sense, the maturation of the theory outlined in this paper itself is an application of the theory, i.e. exerts characteristics of *autopoiesis* in theorizing. We thereby complement the experience-based refinement of the development method itself as well as the justification from relevant research theories. Future research could revisit comprehensively documented design research endeavors that aimed at developing design theories. With these 'war stories' as *extracting case studies* in terms of van Aken (2004), more best-practice methods for theory development could be devised. Most preferably the findings of the extractions would further be inquired during interviews with the researchers that initially conducted the research endeavors to complement mostly positive summaries with hands-on experiences that might also point towards possible anti-practices (*anti-patterns*) for design theorizing.

References

Aken, J.E. 2004. “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), pp. 219-246.

Alexander, C. 1972. *Notes on the Synthesis of Form*, Harvard University Press, Cambridge, MA, USA.

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S. 1977. *A Pattern Language*, Oxford University Press, New York.

Baskerville, R.L. and Myers, M.D. 2009. “Fashion waves in information systems research and practice” in *MIS Quarterly*, (33:4), pp. 647-662.

Becker, J., Delfmann, P. and Knackstedt, R. 2007. “Adaptive reference modeling. Integrating configurative and generic adaptation techniques for information models”, in *Reference Modeling - Efficient Information Systems Design Through Reuse of Information Models*, Physica, Berlin, Germany, pp. 23-49.

Becker, J. and Niehaves, B. 2007. “Epistemological perspectives on is research: a framework for analysing and systematizing epistemological assumptions” in *Information Systems Journal* (17), pp. 197-214.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. and Stal, M. 1996. *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons, Inc., New York, NY, USA.

Carlsson, S.A. 2007. "Developing knowledge through is design science research: For whom, what type of knowledge, and how?" in *Scandinavian J. Inf. Systems* (19:2), pp. 75-86.

Chen, W. and Hirschheim, R. 2004. "A paradigmatic and methodological examination of information systems research from 1991 to 2001" in *Information Systems Journal* (14:2), pp. 197-235.

Coplien, J. 1996. *Software Patterns: Management Briefs*. Cambridge University Press.

DeMarco, T., Hruschka, P., Lister, T., Robertson, S. and Robertson, J. and McMenamin, S. 2008. *Adrenaline Junkies and Template Zombies - Understanding Patterns of Project Behavior*, Dorset House.

Eisenhardt, K.M. and Graebner, M.E. 2007. "Theory building from cases: Opportunities and challenges" in *Academy of Management Journal* (50:1), pp. 25-32.

Friedman, K. 2003. "Theory construction in design research: criteria: approaches, and methods" in *Design Studies* (24:6), pp. 507-522.

Gadamer, H.-G. 1975. *Wahrheit und Methode - Grundzüge einer philosophischen Hermeneutik*. J.C.B. Mohr, Tübingen, 3rd edition.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J.M. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software (Addison-Wesley Professional Computing Series)*. Addison-Wesley Professional.

Gehlert, A., Schermann, M., Pohl, K. and Krcmar, H. 2009. "Towards a research method for theory-driven design research" in *Business Services: Konzepte, Technologien, Anwendungen.* H. R. Hansen, D. Karagiannis, and H. G. Fill (eds.), 9. Internationale Tagung Wirtschaftsinformatik, volume 1, Österreichische Computer Gesellschaft, Wien, Austria, pp. 441-450.

Gregor, S. 2006. "The nature of theory in information systems" in *MIS Quarterly* (30:3), pp. 491-506.

Gregor, S. and Jones, D. 2007. "The anatomy of a design theory" in *Journal of the Association of Information Systems* (8:5), pp. 312-335.

Hevner, A.R., March, S.T., Park, J. and Ram, S. 2004. "Design science in information systems research" in *MIS Quarterly* (28:1), pp. 75-105.

Kant, I. 2008. *Critique of Pure Reason*. Penguin Group, New York, NY, USA.

March, S.T. and Smith, G.F. 1995. "Design and natural science research on information technology" in *Decis. Support Syst.* (15:4) , pp. 251-266.

Markus, M.L., Majchrzak, A. and Gasser, L. 2002. "A design theory for systems that support emergent knowledge processes" in *MIS Quaterly* (26:3), pp. 179-212.

Niehaves, B. 2006. *The Reective Designer - Designing IT-Consulting Processes*. Phd thesis, Westfälische Wilhelms-Universität Münster.

Niehaves, B. 2007. "On epistemological pluralism in design science" in *Scandinavian Journal of Information Systems*, 19, pp. 93-104.

Noble, J. 1998. "Classifying relationships between object-oriented design patterns" in *Australian Software Engineering Conference (ASWEC)*, IEEE Computer Society, Los Alamitos, CA, USA, pp. 98-107.

Nunamaker, J.F., Chen, M. and Purdin, T.D.M. 1991. "Systems development in information systems research" in *J. Manage. Inf. Syst.* (7:3), pp. 89-106.

Schermann, M., Böhmann, T. and Krcmar, H. 2007. "A pattern-based approach for constructing design theories with conceptual models" in *ECIS2007: Processding of the European Conference on Information Systems*, pp. 1368-1379.

Schermann, M., Böhmann, T. and Krcmar, H. 2007. "Fostering the evaluation of reference models: Application and extension of the concept of is design theories" in *Wirtschaftsinformatik* (2), A. Oberweis, C. Weinhardt, H. Gimpel, A. Koschmider, V. Pankratius, and B. Schnizler (eds.), Universitaetsverlag Karlsruhe, Karlsruhe, Germany, pp. 181-198.

Steininger, K., Riedl, R., Roithmayr, F., and Mertens, P. 2009. „Moden und Trends in Wirtschaftsinformatik und Information Systems" in *Wirtschaftsinformatik* (51:6), pp. 478-495.

Vaishnavi, V. and Kuechler, W. (2004/5). "Design Research in Information Systems" January 20, 2004, last updated August 16, 2009. URL: http://desrist.org/design-research-in-information-systems.

Venable, J. R. 2006. "The role of theory and theorising in design science research" in *Design Science Research in Information Systems and Technology*, S. Chatterjee and A. Hevner (eds.), CGU, Claremont, CA, USA.

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

Walls, J.G., Widmeyer, G.R. and El Sawy, O.A. 2004. "Assessing Information Systems Design Theory in Perspective: How Useful was our 1992 Initial Rendition? In *Journal on Information Technology Theory and Practice* (6:2), pp. 43-58.

Wilde, T. and Hess, T. 2007. „Forschungsmethoden der Wirtschaftsinformatik“. *Wirtschaftsinformatik* (49:4), pp. 280-287.

Winter, R., vom Brocke, J., Fettke, P., Loos, P., Junginger, S., Moser, C., Keller, W., Matthes, F. and Ernst, A. 2009. “Patterns in business and information systems engineering” in *Business & Information Systems Engineering*, pp. 468-474, 2009.