Pattern-Based Design Research – An Iterative Research Method Balancing Rigor and Relevance

Sabine Buckl[1], Florian Matthes[2], Alexander W. Schneider[2], and Christian M. Schweda[1]

[1] iteratec GmbH
Inselkammerstr. 4, 82008 München-Unterhaching, Germany
{sabine.buckl,christian.schweda}@iteratec.de
http://www.iteratec.de
[2] Technische Universität München (TUM)
Chair for Informatics 19 (sebis)
Boltzmannstr. 3, 85748 Garching bei München, Germany
{alexander.w.schneider,matthes}@in.tum.de
http://wwwmatthes.in.tum.de

Abstract. Researchers in the area of Information Systems (IS) applying the design science paradigm are confronted with the challenge to make theoretical contributions which also help to solve current and anticipated problems in practice. This is often referred to as the rigor and relevance challenge of design science research. To ensure relevance of the research outcome, research projects in IS are often conducted in close cooperation with one or more industry partners. This typically leads to a need for early results and a binding to the specific organizational context of the participating industry partner(s).

In this paper, we propose pattern-based design research (PDR), an iterative design research method consisting of four phases, to overcome this problem. We argue that patterns as early stage design artifacts enable researchers to build innovative artifacts that address current and anticipated problems of practitioners in an organizational context. Building on well-established concepts as patterns, design theories, and the design theory nexus, the proposed research method enables a researcher to theorize and learn from the intervention at the industry partner(s) while performing rigorous and relevant design science research. We illustrate the applicability of PDR by presenting a research project from the area of enterprise architecture management.

1 Rigor and Relevance in Practice-Driven Research

In the field of Information Systems (IS) research a pluralism of paradigms, methods, and research approaches is prevalent. While behavioral science traditionally plays a major role in IS research, the approach of design science is meanwhile also accepted by the scientific community. Design science approaches (cf. [1-3]) target the creation of a novel *artifact*, i.e., a solution to a relevant problem. Researchers designing an artifact need to account for two important criteria: *rigor* and *relevance* (cf. [3-5]). On the one hand, rigor can be achieved by applying sound methodologies [6]. On the

J. vom Brocke et al. (Eds.): DESRIST 2013, LNCS 7939, pp. 73–87, 2013.

other hand, relevance can be achieved by addressing the needs of using practitioners. Nevertheless, there is an on-going debate on the topic of rigor and relevance. Although many scientists agree that both rigor and relevance have to be achieved simultaneously [7, 3], another school of thought exists with the perception that rigor and relevance each compromise the other [8]. It, in this sense, remains challenging to account for both rigor and relevance [9].

Enterprise Architecture (EA) management is a field, susceptible to practice-driven resign research. While this field has been researched for more than 10 years [10], yet the diversity of the management challenges and the differing organizational contexts have hampered the development of single and embracing management approach. Moreover, the management function in general and modeling of the EA in particular has to be tailored to the concerns and context of the using organization [11]. With no well-established and sufficiently detailed management approach at hand and in the light of the diversity of the management concerns to be addressed, the development of an organization-specific EA management is far from a routine design activity. Especially, the design of the modeling languages is a challenge that is of interest for researchers and practitioners as well.

In a similar line as the aspect of relevance is gaining more and more importance for the field of IS research (cf. [5]), the number of research projects that are conducted in close cooperation with an industry partner or which even are industry-funded increases. Such cooperations on the one hand open the door for developing and extracting case studies (cf. [2, 12]) by employing an intrinsically motivated industry partner but on the other hand are typically subject to the partnering organization's pace. The demand for an early delivery of results and the methodological rigor required for academic contributions (cf. [13]) mirrors the tension between rigor and relevance. This leads to a situation in which researchers are challenged to ensure that their research does not degenerate into "routine design" that according to [3] must be distinguished from design science. In contrast, the close cooperation can be used to contribute to design by providing empirical evidence.

[14] coined the term *community determined output* to delineate that the expected level of abstraction in the research outcome is determined by the members of the participating community. This in particular applies to research performed in close cooperation with an industry partner. To address the different objectives of this community, consisting of practitioners – seeking for practical solutions – and academics – searching for theories – knowledge on different levels of abstraction has to be documented as research findings.

The situation in which scientists cooperate with industry can be described by further characteristics. In IS research often so called *wicked problems* (cf. [15, 16]) are tackled. Wicked problems occur, if the specific situations of different industry partners relate to asymmetric criteria determining the different solutions [17]. In order to increase the relevance of the developed design artifact in terms of importance and suitability (cf. [5]) often an iterative approach is applied. Thereby, each iteration – with a single or multiple industry partners – allows learning and accordingly the evolution of the designed artifact in order to provide a sustainable solution.

Findings originating from research projects that have been conducted in close co-operation with an industry partner should naturally account for the aspect of relevance. Therefore, we subsequently present the pattern-based design research (PDR) approach that addresses the aforementioned challenges accounting for the aspect of rigor in practice-driven research projects. It makes use of patterns, design theories, and a design theory nexus which are introduced in Section 2. By adding a dedicated phase for evaluation and learning to the existing methods an iterative approach – PDR – is created in Section 3. To illustrate the idea of PDR, a concrete example from the domain of Enterprise Architecture (EA) management is presented in Section 4. We use patterns for EA information models to show how they can be used to derive design theories and allow for a controlled nexus evolution by incorporating adjusted as well as new solutions to enhance the general design theories.

2 Contributing Perspectives and Approaches

The role of patterns for IS research is subject to controversies (cf. [18]). In other domains, like software engineering, patterns are well accepted artifacts. Similarly, the role of theories as artifacts for IS research is heavily discussed (cf. [19]). Patterns in contrast to design theories represent best practices that are bound to a specific context in which the provided solution has proven to work. Similarly a design theory nexus interlinks alternative design theories and details the context in which a specific design theory can be applied. The three concepts *pattern*, *design theory*, and *design theory nexus* are introduced in the following to prepare the presentation of PDR.

2.1 Patterns

[20] and [21] introduced the idea of patterns in the area of construction and urban planning. According thereto, a pattern documents a solution for a recurring problem given a specific context. [22] applied the pattern idea to document solutions within the domain of software engineering. They elicited a more explicit pattern structure extended with additional sections, namely *consequences, known uses*, and *related patterns*. The concept of patterns has also been used to document software architectures [23]. Accordingly, documenting good practice solutions to recurring problems in a specific context as patterns is a commonly accepted way to facilitate knowledge dissemination in design-intensives domains. The knowledge abstracted in these patterns is knowledge on operational principles in the sense of [14], i.e., is intended to be applied "as-is". If a solution to a single problem is too complex to be documented by a single pattern or the resulting pattern would be too specific, a pattern language can be used instead. A pattern language decomposes the complex problem/solution description into several self-contained patterns [24]. Since each pattern solves a specific problem within the shared context of the language references between the independent patterns are required. Such references according to [25] can be used to, e.g., identify a smaller pattern that is *used by* a larger pattern, define *variants* of patterns, or a *sequence of elaboration*, i.e. a sequence of patterns from the simple to the complex.

2.2 Design Theory

[26] were the first to introduce the notion of prescriptive theories in the field of IS research. The publications of [27-29], or [30] show, that prescriptive theories – also called *design theories* – generated special attention in the IS community. Such theories provide knowledge support to design activities. Hence, they are considered as "theorized practical knowledge" [31]. Accordingly, the development of design theories requires a close cooperation of scientists and practitioners. Figure 1 presents a conceptual framework for the activities of theorizing in design research according [29] and [32]. The two activities *observation* and *experimentation* call for research that is conducted in close cooperation with industry.

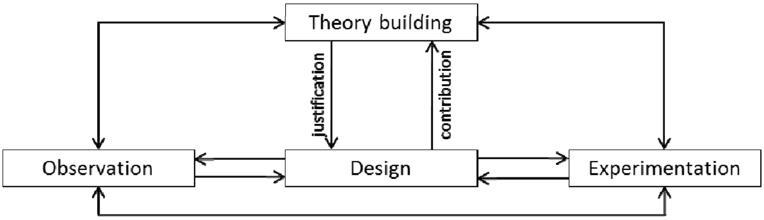

Fig. 1. Elements of theorizing in design research (Source: [32])

Different perspectives on the components of design theories have been taken by [26-27], and [28]. [30] presents a synthesized perspective, further influenced by the idea of pattern-based theory building. We adopt this perspective encompassing the components shown in Figure 2.

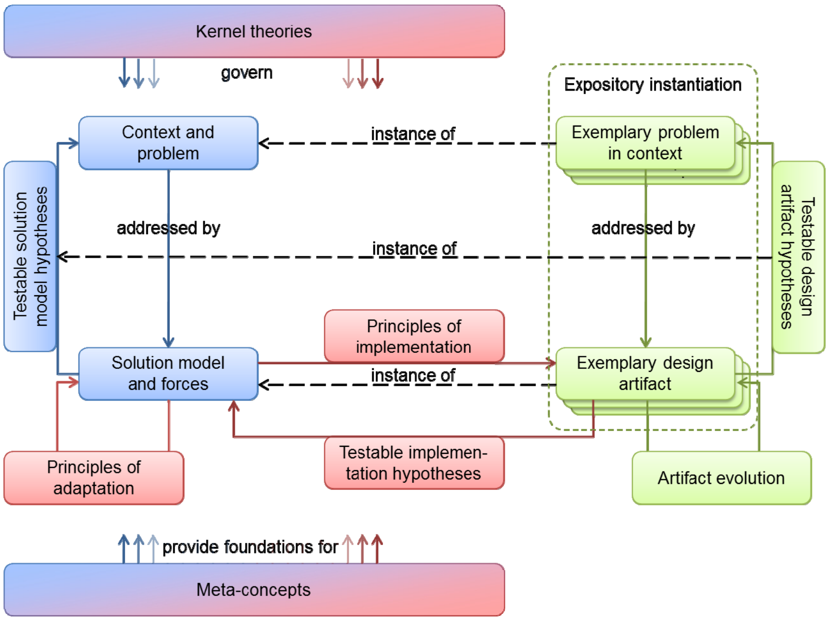

Fig. 2. Components of a design theory (Source: [30])

2.3 Design Theory Nexus

[17] presents the idea of a design theory nexus to address the challenge of wicked problems, i.e., of problems that are characterized by asymmetric criteria. In consequence, possible solutions can only be evaluated in terms of 'good' or 'bad'. A nexus connects these alternative solutions, i.e., design theories. Using the construct of a design theory nexus best practice solutions that evolve from research cooperation projects with industry can be combined to a knowledge base that helps "decision makers in choosing which of the theories are most suitable for their particular goals in their particular setting" [17]. Figure 3 gives an overview about the concept of a design theory nexus that connects different theories and defines the constraining goals and environmental contexts in which the different theories have proven to work good.

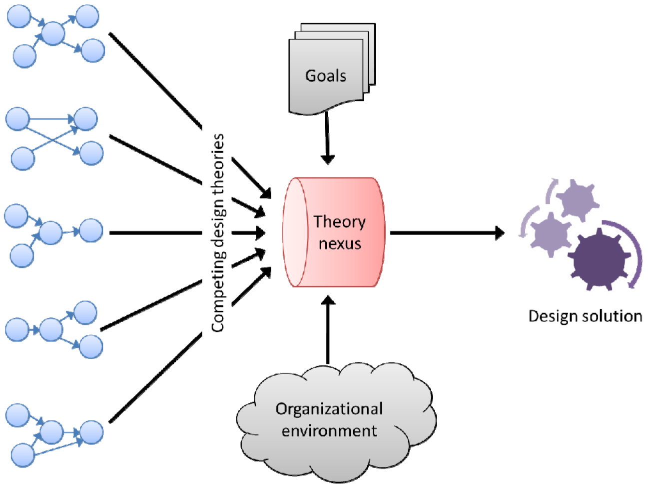

Fig. 3. Components of a design theory nexus [17]

To construct a design theory nexus, [17] detail a five step approach: (1) *identify approaches*, e.g. via a literature analysis, (2) *analyze approaches* to identify explicit or implicit conditions, i.e. context and problem descriptions, that must hold, (3) *formulate assertions* conditions are assessed for practical relevance and reformulated, (4) *develop decision making process* that builds on the assertions, and (5) *develop tool* that supports the evaluation regarding the fit for each design theory in a given situation.

3 Pattern-Based Design Research

Patterns can be understood as early stage design science artifacts observed in practice. Based on this understanding, we propose a pattern-based design research (PDR) method that outlines an approach to balance rigor and relevance in IS research. Following the *activity framework for design science research* as proposed by [33] our research method consists of four main activities: *observe & conceptualize* representing the problem diagnosis, *pattern-based theory building & nexus instantiation*, which enables the abstraction of observed solutions to better theories in terms of [14], *solution design & application*, representing the creation of an IS artifact, and *evaluation & learning* closing a direct feedback loop from practice to academia. Figure 4 gives an overview about our pattern-based design research method and illustrates the interplay between theory and practice.

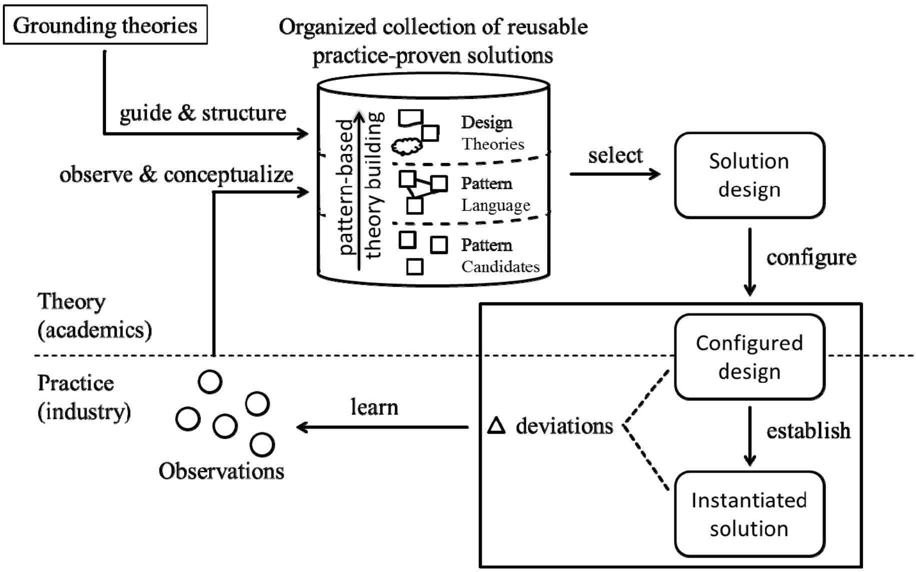

Fig. 4. Pattern-based design research

3.1 Observe and Conceptualize

Patterns are commonly accepted means to document good practice solutions to recurring problems in a specific context. Patterns are further accepted as way to facilitate knowledge abstraction and dissemination in design-intensive domains. Therein, patterns are operational knowledge in the sense of [14] gained from practice. Following this understanding, patterns are neither invented nor developed, but observed best practice solutions. *The rule of three*, established in [34], gives account to this fact: a documented pattern must provide reference to at least three known uses in practice to ensure the re-usability of the provided solution. The different patterns that have been published over the years, mostly follow a typical pattern format according to [23] consisting of a description of the addressed problem, the solution, and the intended application context for the solution. A pattern further identifies driving forces, denotes known usages as well as consequences, and makes relations to other patterns explicit.

In the *observe & conceptualize* activity of the PDR method, good practices from industry are observed and documented following a typical pattern structure. Thereby, the observer which can either be a researcher or practitioner describes at least the following concepts: *problem* to be addressed, *solution* that has proven to work good, *context* in which the solution can be applied, and *forces* that frame the solution space. Depending on the application domain, the conceptualization might be further detailed (see Section 4) by using grounding theories to elicit a pattern terminology for the application domain. The documentation of the pattern candidates is refined in cooperation with the industry partner during so-called pattern workshops. During the conceptualization and documentation of the pattern candidates, known patterns in the field are revisited for relevant terms. In particular, synonyms and homonyms used in the description are identified, and resolved, where possible. Unresolved terminological issues remain to be documented in the next phase via pattern relationships for linguistic compatibility.

3.2 Pattern-Based Theory Building and Nexus Instantiation

The *pattern-based theory building & nexus instantiation* activity of the PDR method describes how pattern candidates mature and evolve to competing design theories for a specific application domain that are interconnected in a design theory nexus. Patterns, which can be regarded as coherent and self-contained design entities describing a solution to a specific problem, can in line with [35] be understood as elementary design principles. In this sense, we interpret patterns as potential building blocks for a design theory. A design theory itself shows typical characteristics of a pattern, i.e. a design theory contains a context description in which the provided solution is applicable as well as the problem to which it can be applied (see Figure 2).

The construction of the design theory nexus from pattern candidates is a possible result from a research cooperation carried out between practice and academia. Thereby, a pattern candidate evolves over time to a pattern, a part of a pattern language, to a part of a design theory as illustrated in Figure 4 by the different levels of the organized collection. If a pattern candidate matures through three successful known uses in practice to a pattern, the researcher integrates the new pattern into the organized collection of patterns by defining relationships to the already existing patterns. Relationship types that can be used are the ones introduced in Section 2. Reflecting the inherent interdisciplinary of IS research, the researcher accounts for terminological compatibility by introducing two new types of relationships between patterns – *linguistically compatible* and *linguistically diverse* – to indicate that two patterns employ compatible or conflicting terminologies.

Building design theories from patterns might at a first glance be an easy to accomplish task, as some constituents of a design theory as introduced in Section 2.2 can be directly mapped to parts of the pattern description, e.g. *context and problem* as well as *solution model and forces*. Patterns however typically vary in respect to the level of granularity and abstraction on which they are described due to their observational nature. This in particular becomes obvious, when the patterns are to be composed into a design theory for a certain research field. The different relationship types of a pattern language support the researcher in addressing this problem and in doing so facilitate the construction of a design theory from a pattern language. Exemplifying this, we subsequently discuss selected relationship types and their roles in pattern-based theory development:

- **Used by** describes that and how a larger pattern employs another patter to solve a sub-problem of the larger pattern. Building on this relationship the researcher can aggregate solution building blocks into a comprehensive solution for a coarse grained problem.
- **Refined by** represents the inverse relationship of used by in which a refining pattern targets a similar problem and context as its "larger" pattern but provides a more detailed solution model or outlines a broader variety of forces to be balanced. Relationships of that can be used by the researcher to define *principles of adaptation* in the design theory as they sketch possible trajectories for refining the design artifact.
- **Variant** relates to a pattern with a similar or closely related problem and context as the initial pattern, providing a solution that only slightly differs from the original one. Relationships of that type may help the researcher to refine the context and problem description by both broadening the scope of the corresponding classes and by raising further dimensions of distinction.
- **Sequence of Elaboration** relates different patterns that describe more and more elaborate solutions for a similar problem and context. Patterns connected by that type of relationship may be understood as contribution to the *principles of adaptation* by providing possible ways to evolve the design artifact and to bring it to a more 'mature' level.

With the alternative or competing design theories developed from pattern-based theory building, the researcher has successfully processed step 1 and 3 of the construction process of a design theory nexus instantiation (see Section 2.3). The final steps 4 and 5 deal with the development of a decision making process and its implementation within a tool to support usability. The researcher is supported during these final steps by the common terminology established by the meta-conceptualization.

3.3 Solution Design and Application

The design theory nexus instantiation is applied in this activity of the PDR method by the researcher to construct a situated design artifact. Therefore, the *exemplary problem in context* is used as input to the decision making process whose output is the *exemplary design artifact*. Creating an expository instantiation, the researcher applies the principles of implementation in practice, i.e., in the environment of an organization willing to use the design theory nexus instantiation. Accounting for terminological aspects, the *solution design* must be configured and adapted to the terminology of the using organization. Finally the configured design is established as new solution in the organization under consideration.

3.4 Evaluation and Learning

While time passes, the instantiated solution may evolve and deviations from the originally configured design may arise in practice. These deviations represent the artifact evolution in terms of a design theory and can be ascribed to the ongoing change of environments, contexts, and goals of the using organization. Following the PDR approach these derivations observed by the practitioner can be used by the researcher to evolve the design theory nexus. Thereby two main types of deviations can be distinguished that result in different kinds of learning:

- Deviations in the instantiated solution that represent minor changes with respect to the configured design. These deviations can typically be traced back to the corresponding design theory, organizational context, or problem.
- Major changes in the instantiated solution that do not match a design theory, organizational context, or problem from the configured design. These deviations typically represent newly observed best practices.

To identify the above deviations a formal review process of the instantiated solution needs to be set up by the practitioner and researcher. If new best practices are observed they have to be documented as pattern candidates as described in the *observe & conceptualize* phase. Minor changes typically result in a rework of existing patterns in terms of changes with respect to context, problem, and solution interplay. Similarly, minor changes might also result in new relationships of the pattern language. In particular the relationships refined by, variant, and sequence elaboration are therefore used.

4 Pattern-Based Design Research Example _ BEAMS IBBs

The research described in this section is an *exemplary instantiation* of the pattern-based research design (PDR) method presented in the previous section. The example is based on previous research in the field of Enterprise Architecture (EA) management. EA management establishes a holistic perspective on all elements forming the architecture of an enterprise, e.g. business processes, applications, information, hardware, and their interrelations. For EA management, descriptions of the EA are created and used. A conceptual model, called "information model" or "meta-model", is usually used to define the structure of an EA description. Figure 5 shows an exemplary excerpt of such a model. Based on this perspective, the respective management function aims to increase transparency about the enterprise [36] and to control the enterprise evolution [37].

4.1 Observe and Conceptualize

In 2007 researchers from the Technische Universität München started to observe identical solutions to EA management problems during their various industry projects. With the idea of software design patterns [22] in mind, an extensive catalog of patterns for EA management has been published [38]. It includes patterns for EA management methods (M-pattern), patterns for EA management viewpoints (V-patterns), and patterns for information model fragments (I-patterns) [39].

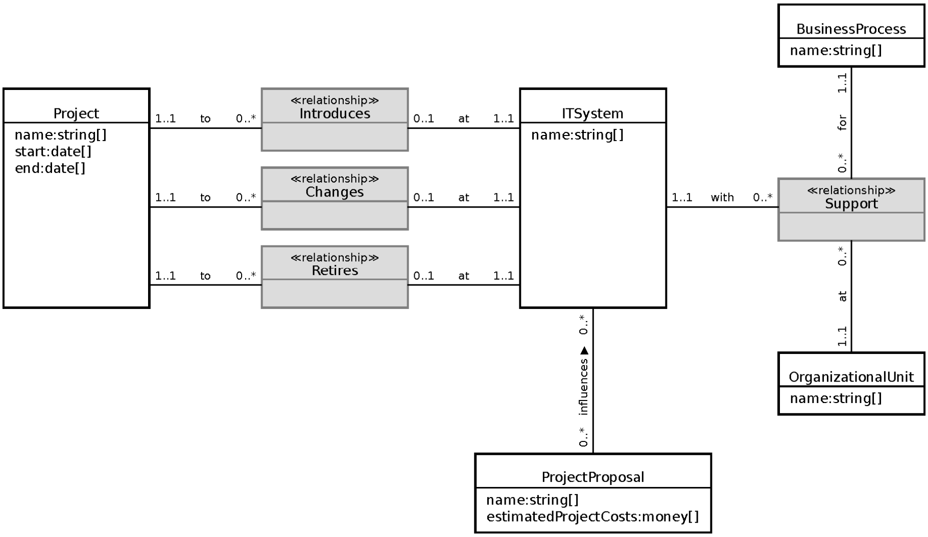

Fig. 5. Exemplary enterprise architecture information model

4.2 Pattern-Based Theory Building and Nexus Instantiation

Based on the context descriptions of the EA management patterns, the Building Blocks for Enterprise Architecture Management Solutions (BEAMS) [40, 41] demonstrated how patterns can be used to derive design theories for EA management. These theories were derived as reusable and configurable building blocks based on the initially observed patterns, for example as information model building blocks (IBBs). As outlined by [42] and visualized in Figure 3, a design theory nexus consists of goals, environment, theory nexus, and design solution. The five step process for a nexus construction suggested by [17] has been executed as follows.

1. The available approaches in the area under consideration, i.e. EA management, have been examined by pattern documentation.
2. Competing theories, i.e., patterns, have been investigated for explicit or implicit conditions, e.g. by determining their different goals.
3. Assertions based on prominent characteristics of competing approaches have been derived which resulted in the organizational context description of the nexus (environment). Centralized vs. decentralized IT organization is one of them.
4. Based on a fitting matrix including the identified EA management goals, the organizational context assumptions and the various solutions, a decision-making process for selecting one or more appropriate approaches for designing an EA management function has been developed. Therefore, these constraints, i.e., goals and organizational context, determine whether a competing EA management approach succeeds or fails.

5. A tool supporting the utilization of the presented nexus has been described by [43]. It begins with a characterization of the situation followed by a selection of building blocks and concludes with their assembly.

4.3 Solution Design and Application

With the characteristics of a specific organizational context and concrete problems as input, the introduced design theory nexus for EA management is able to design an EA meta-model suitable for a specific enterprise. Based on assembled building blocks, i.e the solution design, researchers and practitioners adapt the outcome to additional organization-specific requirements. This adaption includes, among others, the renaming of the general concepts provided by IBBs. Such renaming comprises, e.g. the adaptation of model element names to fit the terminology used by the implementing organization. The general concept of an *IT System*, as depicted in Figure 5, might be renamed to *Business Application* if this term is commonly accepted within the participating linguistic community. Another adaptation might extend or replace values of an enumeration to describe the organizational requirements in more detail. For example, if an IBB provides that a *Project Proposal* can either be in state *accepted* or *rejected*, another state *to be revised* could be added if necessary. As a result, this activity of PDR method provides an applicable EA meta-model which can be implemented by software tools and used for documenting an EA.

4.4 Evaluation and Learning

After the solution design and its organization specific adaption, the information model backing an EA management function is tailored to cover the concerns that are known at the time of its design. Changes in stakeholder composition, in the organizational context or the enterprise context, and the rising maturity of the enterprise-level management functions require EA management governance to take the adequate measures. Relevant measures given the perspective of pattern-based learning are measures that adapt the information model. Two kinds of such measures can be distinguished:

- Adaptations in which **new IBBs** are selected in response to changing concerns.
- Adaptations in which the information model is changed **without using IBBs**.

Adaptations of the latter kind in particular are sources for learning and for evolving the design theory nexus. To enable pattern-based learning, a minimum level of formality regarding the EA management governance (or the part of information model adaptations) is required: for each adaptation at least the responsible enterprise architect has to be documented. Additional information on the concern that caused the particular adaptation as well as on the stakeholders requiring the concern to be addressed is beneficial for the subsequent step of the learning-process, but not mandatory.

The different adaptations occurring in the information models of different industry partners are reviewed on a regular basis. A consulting company, the vendor of an EA management tool, or an organized body of practitioners can perform the necessary reviews. During these reviews, the group of reviewer groups and classifies singular adaptations, i.e., singular changes to an information model, into larger logic units of change. This grouping is performed either based on the documented concerns or based on the results of interviewing the responsible enterprise architects. The change units form “pattern candidates”. For each of the pattern-candidates information following the structure proposed by [24] and the relationships to IBBs constituting the configuration previous to the adaptation are documented.

The mandatory elements (context, problem, solution, and related patterns) of describing an artifact of re-use do not only apply to newly identified pattern candidates, but are also applied to the validated artifacts, i.e., the patterns and the design theories. The mesh of relationships established by denoting related artifacts of re-use complement the perspective of the singular artifact to the whole of a pattern language in terms of [24], or a design theory nexus in terms of [17], respectively. The IBBs, i.e., the theories of the nexus, and the patterns and pattern candidates are thereby linked by two kinds of relationships:

- **Content-relationships** that express that an artifact of re-use (IBB, pattern or pattern candidate) depends on another artifact of re-use, being a prerequisite, a generalization or a foundation in another way. Content-relationships are the usual kind of relationships forming the core of a pattern language.
- **Learning-relationships** that express that an artifact of re-use has been learned from an artifact with lower formalization. A pattern for example is learned from at least one pattern candidate and provides the basis for learning at least one IBB. Learning-relationships are thereby used to document the evolution of the learning design theory nexus.

The conceptual model outlined in Figure 4 summarizes the understanding of pattern-based learning. On the more abstract level, the model describes the core structure of an organized library of re-usable artifacts, concretizing this structure to pattern candidates, patterns and design theories. On this concrete level, the different kinds of artifacts are understood as participating in a learning and formalization process.

5 Outlook

The pattern-based design research method is currently applied without dedicated tool support. The activities of pattern documentation and evolving the design theory nexus could well be supported by a tool. A related circumstance is already discussed by [26] and [17]. Tool support for the design phase of the PDR method, exemplified for the field of EA management, is further outlined in [43]. In this article, especially the need for tool support in comparing information models and in tracking model changes is highlighted.

Although we demonstrated a successful instantiation of a large part of the PDR method in the previous section including the first three phases at least the evaluation & learning phase still remains subject to evaluation in cooperation with practitioners. Such evaluation could benefit from available tool support. The evaluation of this phase nevertheless requires a longer-time span of research in order to be carried out. In the context of EA management a suitable time-span according to [44] would be five years. In addition, the PDR method needs still to be evaluated within IS disciplines other than EA management.

A researcher can assume different roles during the research activities carried out in cooperation with practitioners. Accounting for these roles, the PDR method can be further detailed. Especially during the application phase of PDR the participating researcher can inhabit different roles. Like in case studies, the researcher can act as an observer without intervening or like in action research the researcher can be an actor influencing the solution design instantiation. Furthermore, if design and evaluation are processed in parallel as suggested by [45] action design researchers are expected to share their knowledge of theory and technological advances. Accordingly, the appropriate degree of influence still needs to be found and may depend on the actual context.

References

1. Simon, H.A.: The Sciences of the Artificial, 2nd edn. MIT Press, Cambridge (1981)
2. Aken, J.: Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules. Journal of Management Studies 41(2), 219–246 (2004)
3. Hevner, A., March, S., Park, J., Ram, S.: Design science in information systems research. Management Information Systems Quarterly 28(1), 75–105 (2004)
4. King, J.L., Lyytinen, K.: The Future of the IS Field: Drawing Directions from Multiple Maps, pp. 345–354. John Wiley & Sons, West Sussex (2006)
5. Rosemann, M., Vessey, I.: Toward improving the relevance of information systems research to practice: the role of applicability checks. Management Information Systems Quarterly 32(1), 1–22 (2008)
6. Benbasat, I., Weber, R.: Rethinking 'diversity' in information systems research. Information Systems Research, 389–399 (1996)
7. Benbasad, I., Zmud, R.W.: Empirical research in information systems: The practice of relevance. Management Information Systems Quarterly 23(1), 3–16 (1999)
8. Davenport, T.H., Markus, L.M.: Rigor and relevance revisited: Response to benbasat and zmud. Management Information Systems Quarterly 23(1), 19–23 (1999)
9. Staw, B.M.: Repairs on the road to relevance and rigor: Some unexplored issues in publishing organizational research, pp. 85–97. Sage, London (1995)
10. Langenberg, K., Wegmann, A.: Enterprise architecture: What aspect is current research targeting? Technical report, Laboratory of Systemic Modeling, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (2004)
11. Aier, S., Kurpjuweit, S., Riege, C., Saat, J.: Stakeholderorientierte dokumentation und analyse der unternehmensarchitektur. GI Jahrestagung (2), 559–565 (2008)
12. Eisenhardt, K.M., Graebner, M.E.: Theory building from cases: Opportunities and challenges. Academy of Management Journal 50(1), 25–32 (2007)
13. Gallupe, R.B.: The tyranny of methodologies in information systems research. SIGMIS Database 38, 20–28 (2007)
14. Vaishnavi, V., Kuechler, W.: Design research in information systems (2004), `http://desrist.org/design-research-ininformation-systems`

15. Churchman, C.W.: Guest editorial: Wicked problems. Management Science 14(4), 141–142 (1967)
16. Rittel, H., Webber, M.: Dilemmas in a general theory of planning. Policy Sciences 4, 155–169 (1973)
17. Pries-Heje, J., Baskerville, R.: The design theory nexus. Management Information Systems Quarterly 32(4), 731–755 (2008)
18. Winter, R., vom Brocke, J., Fettke, P., Loos, P., Junginger, S., Moser, C., Keller, W., Matthes, F., Ernst, A.M.: Patterns in business and information systems engineering. Business & Information Systems Engineering, 468–474 (2009)
19. Straub, D.W.: Editor's comments: Does mis have native theories? MIS Quarterly (36), iii–xii (2012)
20. Alexander, C.: Notes on the Synthesis of Form. Harvard University Press, Cambridge (1972)
21. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S.: A Pattern Language. Oxford University Press, New York (1977)
22. Gamma, E., Helm, R., Johnson, R., Vlissides., J.M.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional (1994)
23. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-oriented software architecture: a system of patterns. John Wiley & Sons, Inc., New York (1996)
24. Meszaros, G., Doble, J.: A Pattern Language for Pattern Writing, pp. 529–574. Addison Wesley (1998)
25. Noble, J.: Classifying relationships between object-oriented design patterns. In: Australian Software Engineering Conference (ASWEC), pp. 98–107. IEEE Computer Society, Los Alamitos (1998)
26. Walls, J., Widmeyer, G., El Sawy, O.: Building an information system design theory for vigilant eis. Information Systems Research 3(1), 36–59 (1992)
27. Gregor, S., Jones, D.: The anatomy of a design theory. Journal of the Association of Information Systems 8(5), 312–335 (2007)
28. Schermann, M., Böhmann, T., Krcmar, H.: Fostering the evaluation of reference models: Application and extension of the concept of is design theories. Wirtschaftsinformatik 2, 181–198 (2007)
29. Gehlert, A., Schermann, M., Pohl, K., Krcmar, H.: Towards a research method for theory-driven design research. business services: Konzepte, technologien, anwendungen. In: 9. Internationale Tagung Wirtschaftsinformatik, pp. 441–450 (2009)
30. Buckl, S., Matthes, F., Schweda, C.M.: Utilizing patterns in developing design theories. In: 2010 International Conference on Information Systems, ICIS 2010 (2010)
31. Goldkuhl, G.: Design theories in information systems - a need for multi-grounding. Journal of Information Technology Theory and Application 6(2) (2004)
32. Nunamaker, J., Chen, M., Purdin, T.: Systems development in information systems research. Journal of Management Information Systems 7(3), 89–106 (1991)
33. Venable, J.R.: The role of theory and theorising in design science research. In: Design Science Research in Information Systems and Technology (2006)
34. Coplien, J.O.: Software Patterns: Management Briefs. Cambridge University Press, Cambridge (1996)
35. Markus, M.L., Majchrzak, A., Gasser, L.: A design theory for systems that support emergent knowledge processes. MIS Quaterly 26(3), 179–212 (2002)
36. Hanschke, I.: Strategic IT Management: A Toolkit for Enterprise Architecture Management. Springer (2009)
37. Murer, S., Bonati, B., Furrer, F.J.: Managed Evolution. Springer (2010)

38. Buckl, S., Ernst, A.M., Matthes, F.: Enterprise architecture management pattern catalog. Technical report, Technische Universität München, Munich, Germany (2008)
39. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F., Schweda, C.M.: Enterprise architecture management patterns - exemplifying the approach. In: The 12th IEEE International EDOC Conference (EDOC 2008), München, pp. 393–402 (2008)
40. Buckl, S.M.: Developing Organization-Specific Enterprise Architecture Management Functions Using a Method Base. PhD thesis, Technische Universität München, München, Germany (2011)
41. Schweda, C.M.: Development of Organization-Specific Enterprise Architecture Modeling Languages Using Building Blocks. PhD thesis, Technische Universität München (2011)
42. Buckl, S., Matthes, F., Schweda, C.M.: A situated approach to enterprise architecture management. In: IEEE International Conference on Systems, Man and Cybernetics (SMC 2010), pp. 587–592. IEEE (2010)
43. Buckl, S., Matthes, F., Schweda, C.M.: A method base for enterprise architecture management. In: Ralyté, J., Mirbel, I., Deneckère, R. (eds.) ME 2011. IFIP AICT, vol. 351, pp. 34–48. Springer, Heidelberg (2011)
44. Ross, J.W., Beath, C.M.: Sustainable it outsourcing success: Let enterprise architecture be your guide. MIS Quarterly Executive 5(4), 181–192 (2006)
45. Sein, M.K., Henfridsson, O., Puraro, S., Rossi, M., Lindgren, R.: Action design research. MIS Quarterly (2) (2011)