How to Design Patterns in IS Research – A State-of-the-Art Analysis

Andreas Günther[1] and Robin Knote[2]

[1] University of Kassel, Kassel, Germany
andreas.guenther@gumail.net
[2] University of Kassel, Information Systems, Kassel, Germany
robin.knote@uni-kassel.de

Abstract. Patterns are becoming increasingly prominent in the field of Information Systems (IS). They contain good practice solutions to recurring problems and are therefore especially valuable for systems development. Although a huge amount of patterns for nearly every problem in systems development (and even in many fields of IS such as enterprise information management or security) exist, literature still misses a systemized overview on how such patterns are developed. Approaches in practice exist but are either methodologically imprecise or lack a scientific foundation. This contribution is devoted to review the state-of-the-art on how patterns are designed. The findings of our systematic literature review reveal approaches for pattern development which we consolidate, structure and critically reflect. Since we argue that patterns can be considered design artifacts, we apply a design science research (DSR) lens to these approaches, highlight potential gaps and show needs for future development in theory and practice.

Keywords: pattern, pattern development, sociotechnical systems engineering, design science research

1 Introduction

The ongoing digital transformation causes great technological and societal challenges and opportunities. Consequently, the complexity of systems engineering is exponentially increasing. On the technical side, systems have to be able to communicate with an unknown number of different systems and subsystems, offer individualized services to the user and even change their behavior during runtime based on context information. On the societal side, systems have to be accepted by the users, which require privacy, trust, usability or legal considerations as well as the offering of added value. Therefore, the main challenges for systems engineering are identified as complexity, multidisciplinarity and user focus. Accordingly, systems development, in order to be efficient and effective, has to include heterogeneous and multidisciplinary stakeholders to analyze their requirements and offer proper designs [1]. A profitable approach for eliciting, understanding and applying multidisciplinary knowledge and experience within systems engineering are patterns [2–5]. Patterns

13th International Conference on Wirtschaftsinformatik,
February 12-15, 2017, St. Gallen, Switzerland

Günther, A.; Knote, R. (2017): How to Design Patterns in IS Research – A State-of-the-Art Analysis, in Leimeister, J.M.; Brenner, W. (Hrsg.): Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017), St. Gallen, S. 1393-1404

focus on making good solutions to recurring problems accessible and comprehensible and thus, prevent the applier from 'reinventing the wheel'. Capturing multidisciplinary knowledge related to certain design problems, patterns may decrease cost and effort when solving the problem. Thus, they ensure that already identified requirements and good designs are used (avoidance of potential mistakes), reveal knowledge in an understandable form for all stakeholders and provide a common vocabulary for experts to communicate about design questions. Furthermore, they lead to more transparency of the problem context, as they entail information about structural relations and dependencies that would otherwise probably be neglected [2, 6–8]. Therefore, the use of patterns, has a high impact on the performance of systems, services and organizations, leading to economic benefits and increased competitiveness. Prominent examples of patterns are the Design Patterns by the Gang of Four, which contains patterns about good software engineering [8].

However, although a lot of pattern repositories and catalogues exist and patterns are used in systems engineering practice, the question of how patterns are developed to reach certain design goals (such as quality, applicability or utility) remains unclear. Furthermore, since pattern development is a practical problem and is mostly focused by practitioners or practitioner-oriented communities, existing approaches still lack a scientific foundation. However, we suggest that such a methodological underpinning is necessary in order to have patterns becoming an acknowledged topic for IS and adjacent research fields. We believe that pattern development, underpinned by methodology, can have a strong impact and offer new solutions to existing design problems, as well as increasing their added value through best possible development. From a practical point of view, a clear methodological approach on how to develop patterns is suggested to resolve issues concerning pattern structure, quality and utility.

To solve these problems, we are undertaking a systematic literature review to identify, cluster and structure pattern development approaches. Thus, our work is devoted to structure the field and can be considered a first step towards approaching pattern development from a scientific base.

Our paper is structured as follows. Section 2 gives deeper insights about patterns and pattern development. Our research methodology is described in section 3. The findings of the literature review are given in section 4 and further discussed in section 5. We conclude with our contributions, limitations and future outlook in section 5.

2 Background

2.1 Patterns in IS

The idea to establish and apply patterns originally stems from the field of architecture. Alexander [9] defines patterns as specifications of recurring problems and proven solutions to these problems. Hence, a pattern comprises applied and established knowledge on how to solve a certain design problem. In the mid 1990's, the pattern concept was first introduced for system development by Gamma et al. [10] who described design patterns for object-oriented software development. Since then, the

acceptance and use of patterns has increased tremendously. Nowadays, a huge amount of patterns can be found for different fields of IS, such as human-computer interaction, security, requirements engineering and enterprise integration. Due to the wide circulation of pattern publications, the pattern community is continuously becoming larger. Today, many workshops on scientific conferences (such as RePa[1]) or even entire conferences (e.g., PLoP[2] conferences) are devoted to pattern development and dissemination.

Across all disciplines, the main goal of patterns is reuse of established knowledge for specific problem contexts [11]. Therefore, patterns contain templates for problem specification and other relevant information in tabular form, including a unique pattern name, meta data (e.g., author), conditions of pattern applicability and relations to other patterns [7, 12]. Thus, patterns should contain at least five central elements: a description of *problems* and *forces* to describe the problem and why it is difficult to solve (problem domain), a description of the *solution* and possible *consequences* to demonstrate what happens when the solution is applied (solution domain) and a description of the *context* in which the pattern is applicable [13].

2.2 Pattern Development

Although nowadays many patterns exist for various application domains, literature about how to systematically develop them is scarce. Wellhausen and Fießer [13] introduce the topic by explaining what patterns are and how to identify them with regard to IS disciplines from a practitioners point of view. Furthermore, Withall differentiates between a systematic (inductive) and a theory-based or opportunistic (deductive) approach for pattern development [7]. According to his work, following a systematic approach, is appropriate in case enough satisfactory solutions exist for a problem. In this case, these solutions (e.g., design guidelines) are collected, aggregated and generalized in a pattern format. The opportunistic approach is applicable if no or few solutions exist. Pattern development is then based on relevant experiences and theoretical insights (e.g., empirical data). However, the process itself still lacks a theoretical and methodological foundation which is important for pattern development in a scientifically acknowledgeable manner. Thus, Petter et al. consider the 'scientific' way of pattern development a design science approach, in which the pattern becomes the design artifact [14]. A design artifact is a solution to a problem, which creates value while being used. To ensure the quality of an artifact and therefore its value, it iterates within three different cycles until it creates a satisfactory value. These cycles can be assigned to pattern development for developing satisfactory, value adding, patterns. The Design science research (DSR) cycles can be described as follows [15]. The *relevance cycle* is about identifying the problem

[1] International Workshop on Requirements Patterns held on the International Requirements Engineering Conference. http://re16.org/pages/conference/workshops/#WS08 (accessed October 12, 2016)

[2] Pattern Languages of Programs conferences. http://hillside.net/conferences (accessed October 12, 2016)

domain, eliciting requirements and, after the solution is built, field-testing the artifact. In the *rigor cycle* a common knowledge base is created and other relevant research (e.g., other pattern repositories) is screened to derive design considerations. The goal of the *design cycle* is to build the artifact and validate it against preliminarily defined evaluation criteria. In this work, we follow this logic, since other research present DSR as a suitable way for pattern development [16]. Hence, the conceptual model for our literature review is divided into relevance (identify problems), rigor (include existing knowledge), design (build artifacts) and evaluation (assure value).

3 Methodology

To identify approaches for pattern development, we conduct a systematic literature review [17, 18]. According to the taxonomy given by Cooper, our review is characterized as follows [19]: We focus on approaches for but not the results of pattern development (i.e., the actual patterns). We thereby understand an approach as a technique or type of (collaborative) work, that is used for developing patterns. The goal is to identify pattern development approaches and to integrate them within the DSR framework. The findings will be clustered and organized on a conceptual level. Our target audience are scholars and practitioners. The coverage of this research is representative and should identify common approaches.

The search for relevant articles was conducted in February 2016. For identifying relevant databases and search phrases, a brief research was done based on the aforementioned foundational work including a short forward search. The following databases were chosen, as they cover the PloP conferences, fit the targeted domain of systems engineering and Requirements Engineering: ACM, SpringerLink, IEEE Xplore und AISeL. The predominant phrases in the context of pattern development are: *pattern mining* [7, 20–22], *pattern writing* [7, 22–24] or *pattern authoring* [24, 25]. Unsuitable phrases were *pattern detection* [27, 28] and *pattern discovery* [29], since they are predominantly used in Data Mining or technical pattern detection and hence were excluded from our search. The search was conducted in title, abstract and keywords, which led to 1156 hits. In a first step title, abstract and keywords were screened and unsuitable articles disregarded. In the second step the remaining full articles were read thoroughly to further sort out non-relevant articles. In general, articles were disregarded if they were purely technological, did not mention any approach or were not available in German or English. In the end, a total number of 28 articles were selected for further analysis.

The initial list of possible approaches found in the relevant articles contained 76 items. A four phase process was used to cluster these items into useful and relevant approaches. First, all items were analyzed regarding their abstraction level, which resulted in disregarding very abstract approaches (e.g., Grounded Theory). Second, the meaning of each remaining approach was summarized in a short description- Approaches with similar meaning were clustered in a third step. Fourth, all approaches that did not have at least three hits were disregarded. Since our way of clustering approaches is comparable to coding in qualitative content analysis (i.e.,

interpretative nature), two researchers clustered the findings independently to establish inter-coder reliability [30]. The clustered approaches were finally analyzed regarding their fit into DSR elements as mentioned above.

4 Approaches for Pattern Development

The findings of the literature review are presented in Table 1. In general, it can be observed that inductive approaches (i.e., observing the specifics of a context and moving towards generalization) are more commonly used to identify and design possible patterns, whereas deductive approaches (i.e., starting from generalized insights and moving towards specifics of a context) are used to structure and validate them in different practical contexts [21, 23]. For example, Schadewitz and Jachna used an inductive approach to find reoccurring issues in student teams' interactions and communications as well as a deductive approach to discover possible connections between these issues and to compare them within different contexts [26]. Another general finding is the importance of collaboration and communication. It urges the meaning of the community and inclusion of interdisciplinary stakeholders for actively creating and sharing comprehensible patterns [24, 25].

Table 1. Pattern Development approaches

Approach	Hits	Relevance	Rigor	Design	Evaluate	Sources
development workshop	15	●	○	●	○	[21, 23–25, 31–41]
enhancement workshop	15	○	○	●	●	[21, 23–25, 31–41]
guided development	15	○	○	●	○	[7, 24, 25, 31–33, 36, 42–46]
shepherding	9	○	○	●	○	[21, 23–25, 32–34, 36, 41]
expert interview	8	●	○	○	●	[26, 32, 33, 35, 39, 41, 43, 47]
observation	8	●	○	○	○	[21, 23, 26, 31–33, 40, 43]
open channel	6	●	○	○	●	[21, 24, 25, 36, 41, 43, 44]
literature review	4	○	●	○	○	[32, 32, 36, 42, 48, 49]
collaborative learning and development	3	●	○	●	○	[33, 35, 39]
pattern mapping	3	●	○	●	○	[21, 26, 32]
pattern writing	3	●	○	●	○	[23, 31, 35]

○ = not applicable / unknown; ● = applicable

Shepherding and workshops are state-of-the-art pattern development approaches and are frequently used on pattern conferences [25]. They are both sharing a highly interactive character. A workshop is a cooperation of a group of people, while shepherding is a cooperation between a shepherd (experienced pattern author) and a sheep (pattern author). A general benefit of a workshop is the attendance of multiple participants which leads to a harmonization of different views [32]. The workshop can target a wide range of goals, from creatively coming up with new pattern ideas to finalizing one definite pattern. Accordingly, there is no generic workshop approach. We suggest two different approaches. First, the pattern development workshop, which is used to mine (collect, categorize and summarize) patterns [23]. For example, Iacob uses a set of workshops to confront designers with a set of problems. The design process is analyzed and transformed into design issues, which are then counted. Their degree of recurrence is calculated to identify pattern candidates [21]. The output of the development workshops are pattern candidates or patterns. Hence, development workshops can be used to find relevant issues and design pattern artifacts. Second, the enhancement workshop, which is about giving and getting feedback on a pattern (evaluation) and using this to improve the pattern until its final form. To continue with the example from Schadewitz and Jachna, their design issues were evaluated in workshops with novices and experts. The insights from those workshops were used for further development. Experts can actively interact with design suggestions, while novices can offer great insights about the actual quality of the pattern [41]. One technique, "fly on the wall", is described very often and helps exemplifying the intention of the enhancement workshop: the pattern author is listening to a group of participants which discusses the pattern he developed without being allowed to intervene or rectify. The goal of this approach is to evaluate the understandability of patterns as stand-alone, self-explaining artifacts [6]. Therefore, the enhancement workshop can be used to design and evaluate patterns.

Shepherding is much more focused on pattern specific knowledge and the artifact itself. It usually takes place over several months and is an intense interaction between the shepherd and the sheep. This makes it a very valuable approach for the design phase. In practice, it is often used in advance of pattern conferences, which offer a wide range of approaches to support pattern development. They contain multiple approaches like workshops, discussions, presentations and shepherding.

Collaborative learning and development, is an effective approach to start pattern mining, as it teaches domain experts and novices about patterns and how to develop them. While understanding the nature of patterns and their use, participants start recognizing and can support each other in developing first patterns. This approach raises pattern awareness, which helps finding relevant issues and supports the design of first patterns.

Guided development is mainly used to support designing the actual pattern as it informs about possible formats (even offers tool supported templates), steps to do while writing a pattern or best practices with useful recommendations. It makes particular pattern development knowledge available to be used by authors. The approach comprises elements such as step-by-step guides, pattern templates and checklists.

Expert interviews are interactive meetings of two people or a small group to gather each subject's personal views and experiences within their context. The interviewees are either experts of a certain domain and thus act as source for issues within their expertise (relevance) or future pattern appliers who give feedback on pattern designs (evaluation).

A not necessarily interactive approach for investigating people and/or technology within their realworld environment, is observation. It was also used for first patterns by Alexander [50] and is highly applicable in identifying relevant issues for pattern development. Especially for non-communicable complex situations and contexts this approach is highly effective.

The open channel approach implies communication between all stakeholders of a system. A continuous discussion between stakeholders leads to two main benefits: Already discovered patterns are being continuously evaluated and new issues for pattern development might arise within a simple discussion. Therefore, the open channel approach is inherent in most of the presented approaches. However, its core intention is to focus on being open for discussion at all other phases as well. Fehling et. al. posit that patterns evolve continuously through discussion in a community [44].

Pattern mapping is used to structure codes, elements or domains of patterns with regards to their relationships. This approach supports the design of patterns as it sheds light on interdependencies between different patterns. Additionally, it can be used to identify relevant issues or missing solutions for new patterns as it reflects the structure of the underlying problem domain. [32].

The literature review approach is a systematic research of existing knowledge about existing solutions, a theoretical base or adjacent patterns. A literature review can access a multitude of resources from wikis, scientific and practitioner-oriented outlets or pattern repositories [32]. Pattern repositories, as an important resource for pattern development, comprise different patterns for certain problem domains or applications [48]. However, due to the vast amount of potential sources, a pattern-oriented literature review may be a difficult task.

Pattern writing is the actual process of creating the pattern artifact, but it is also an approach for identifying relevant patterns as it reveals new facts by adding additional experiences or asking the authors to express more of their implicit knowledge [23].

5 Discussion

It can be observed that most of the identified approaches are applicable in the design cycle. Thus, it may be inferred that this is the core to the very practitioner-oriented and less scientific state-of-the-art of pattern development. However, we second the opinion that this phase is the most crucial part of a pattern development process. However, high number of approaches can be applied in the relevance cycle as well, which again highlights that pattern development is practically-driven process. Not many approaches were found applicable for the rigor or evaluation phase. Although a vast amount of pattern repositories exist that can potentially be used in the rigor cycle, the search for patterns relevant to a specific development problem is often a complex

endeavor. This is mainly caused by a lack of structuration as well as nonexistent formatting standards for patterns and pattern repositories. Furthermore, validation of patterns as usually conducted in the design-evaluation phase of DSR has yet widely been neglected. This issue is also manifested in the vague definition of when a recurring solution can be considered a pattern: the pattern community calls for at least three good implementations, without specifying what 'good' means. Hence, based on our findings, we call for rigor in (1) screening existing relevant knowledge that may be useful for pattern design and (2) validate patterns and evaluate them in the problem context. We believe that both are necessary to give pattern development a stronger scientific impact besides its practical relevance.

6 Conclusion and Research Outlook

Our research embodies a first step towards a clearer understanding of pattern development approaches from a design science perspective. To the best of our knowledge, we are the first to structure existing approaches for pattern development by conducting a systematic literature review. With this contribution we aim to make the following contributions: From a scholarly perspective, we made steps towards an operationalization of DSR for pattern development, which leads to the creation of both prescriptive and descriptive design knowledge [51, 52]. From a practical viewpoint, pattern authors can use our findings to enhance their own pattern development. Our findings serve as a guide through necessary phases of pattern development and highlight applicable approaches for each phase. Furthermore, the description provided for each approach supports pattern authors in choosing the best fit for their context. This may lead to patterns with improved quality and more efficient and effective pattern development.

However, since our research is only a first step towards structuring the field it does not come without limitations. Although we are sure in having covered the most relevant approaches with our orientation towards the pattern community, literature search should be extended to more scientific and pattern-related outlets and databases in future research in order to reveal complementary approaches and enhance the presented findings. Furthermore, future research may build up on our results by creating (reference) processes for pattern development. Finally, evaluation of our approach in specific problem contexts is necessary to prove our results in field.

References

1. Pohl, K.: Requirements Engineering. Grundlagen, Prinzipien, Techniken. Dpunkt-Verl., Heidelberg (2008)
2. Towards Interdisciplinary Design Patterns for Ubiquitous Computing Applications. Kassel University Press, Kassel (2014)
3. Hoffmann, A., Hoffmann, H. & Söllner, M.: Fostering Initial Trust in Applications – Developing and Evaluating Requirement Patterns for Application Websites. Utrecht, Netherlands (2013)

4. Hoffmann, A., Söllner, M., Hoffmann, H., Leimeister, J.M.: Requirement Patterns to Support Socio-Technical System Design. In: David, K., Geihs, K., Leimeister, J.M., Roßnagel, A., Schmidt, L., Stumme, G., Wacker, A. (eds.) Socio-technical Design of Ubiquitous Computing Systems, pp. 191–209. Springer International Publishing, Cham (2014)
5. Gebauer, L., Kroschwald, S. & Wicker, M.: Anforderungsmuster zur Förderung der Rechtmäßigkeit und Rechtsverträglichkeit von Cloud Computing-Diensten. Kassel University Press, Kassel (2015)
6. Kohls, C.: The theories of design patterns and their practical implications exemplified for e-learning patterns. Ingolstadt (2014)
7. Withall, S.: Software requirement patterns. Microsoft Press, Redmond, Wash. (2007)
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns. Elements of resuable object-oriented software. Pearson education limited, England (1995)
9. Alexander, C.: The timeless way of building. Oxford University Press, New York (1979)
10. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns - Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1994)
11. Franch, X., Palomares, C., Quer, C., Renault, S., Lazzer, F. de: A Metamodel for Software Requirement Patterns. In: Wieringa, R., Persson, A. (eds.) Requirements Engineering: Foundation for Software Quality, 6182, pp. 85–90. Springer Berlin Heidelberg (2010)
12. Durán Toro, A., Bernárdez Jiménez, B., Ruiz Cortés, A., Toro Bonilla, M.: A Requirements Elicitation Approach Based in Templates and Patterns. Workshop em Engenharia de Requisitos 1999 (1999)
13. Wellhausen, T. and Fießer, A.: How to write a pattern? A rough guide for first-time pattern authors, http://europlop.net/sites/default/files/files/0_How%20to%20write%20a%20pattern-2011-11-30_linked.pdf
14. Petter, S., Khazanchi, D., Murphy, J.D.: A Design Science Based Evaluation Framework for Patterns. SIGMIS Database 41, 9–26 (2010)
15. Hevner, A.R.: A three cycle view of Design Science Research. Scandinavian Journal of Information Systems 19, 87–92 (2007)
16. Hoffmann, A.: Anforderungsmuster zur Spezifikation soziotechnischer Systeme. Standardisierte Anforderungen der Vertrauenwürdigkeit und Rechtsverträglichkeit. Kassel University Press, Kassel, Germany (2014)
17. Vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A. (eds.): Reconstructing the giant: On the importance of rigour in documenting the literature search process (2009)
18. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: Writing a literature review. MIS Quarterly (2002)
19. Cooper, H.M.: Organizing knowledge syntheses. A taxonomy of literature reviews. Knowledge in Society 1, 104–126 (1988)
20. Kohls, C.: The structure of patterns. In: Kohls, C. (ed.) the 17th Conference, pp. 1–10 (2010)

21. Iacob, C.: A design pattern mining method for interaction design. In: Paternò, F., Luyten, K., Maurer, F. (eds.) the 3rd ACM SIGCHI symposium, p. 217 (2011)
22. Hillside Group: The Hillside Group - A group dedicated to design patterns. Home of the patterns library., http://hillside.net/
23. Kohls, C., Panke, S.: Is that true…? In: Wirfs-Brock, R. (ed.) the 16th Conference, p. 1 (2009)
24. Wellhausen, T., Fiesser, A.: How to write a pattern? In: Avgeriou, P., Fiesser, A. (eds.) the 16th European Conference, pp. 1–9 (2011)
25. Fehling, C., Barzen, J., Breitenbücher, U., Leymann, F.: A process for pattern identification, authoring, and application. In: Eloranta, V.-P., van Heesch, U. (eds.) the 19th European Conference, pp. 1–9 (2014)
26. Schadewitz, N., Jachna, T.: Introducing New Methodologies for Identifying Design Patterns for Internationalization and Localization. In: Aykin, N. (ed.) Usability and Internationalization. Global and Local User Interfaces. Second International Conference on Usability and Internationalization, UI-HCII 2007, Held as Part of HCI International 2007, Beijing, China, July 22-27, 2007, Proceedings, Part II, 4560, pp. 228–237. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg (2007)
27. Yu, D., Zhang, Y., Ge, J., Wu, W.: From Sub-patterns to Patterns: An Approach to the Detection of Structural Design Pattern Instances by Subgraph Mining and Merging. In: 2013 IEEE 37th Annual Computer Software and Applications Conference (COMPSAC), pp. 579–588 (2013)
28. Tsantalis, N., Chatzigeorgiou, A., Stephanides, G., Halkidis, S.: Design Pattern Detection Using Similarity Scoring. IIEEE Trans. Software Eng. 32, 896–909 (2006)
29. Gupta, M., Pande, A., Rao, R.S., Tripathi, A.K.: Design Pattern Detection by normalized cross correlation. In: 2010 International Conference on Methods and Models in Computer Science (ICM2CS 2010), pp. 81–84
30. Mayring, P.: Qualitative Inhaltsanalyse. Grundlagen und Techniken. Beltz, Weinheim u.a. (2010)
31. Meszaros, G., Doble, J.: A pattern language for pattern writing. In: Martin, R.C., Riehle, D., Buschmann, F. (eds.) Pattern languages of program design 3. [papers to PLoP '96 and EuroPLoP '96]. Addison-Wesley, Reading, Mass. (1998)
32. Lotz, N., Law, E.L.-C., Nguyen-Ngoc, A.V.: A process model for developing learning design patterns with international scope. Education Tech Research Dev 62, 293–314 (2014)
33. Linden, T., Cybulski, J.: Refining the process of sharing problem-solving experience across domain: A hermeneutic study. AMCIS 2006 Proceedings, 433 (2006)
34. Rising, L., Rehmer, K.: Patterns for sustainable development. In: Kohls, C. (ed.) the 17th Conference, pp. 1–11 (2010)
35. Wesson, J., Cowley, L.: UI Design Patterns: From Theory to Practice. In: Seffah, A., Desmarais, M.C., Gulliksen, J. (eds.) Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle, 8, pp. 331–351. Springer, Dordrecht (2005)

36. Schümmer, T., Haake, J.M.: Shaping Collaborative Work with Proto-patterns. In: Pipek, V., Rosson, M.B., Ruyter, B. de, Wulf, V. (eds.) End-user development. 2nd international symposium, IS-EUD 2009, Siegen, Germany, March 2 - 4, 2009 ; proceedings, 5435, pp. 166–185. Springer, Berlin (2009)
37. Köppe, C.: Using pattern mining for competency-focused education. In: van der Veer, G., Sikorski, M., Sloep, P., van Eekelen, M. (eds.) Second Computer Science Education Research Conference, pp. 23–26 (2012)
38. Deng, J., Kemp, E., Todd, E.G.: Focussing on a standard pattern form. In: Billinghurst, M. (ed.) the 6th ACM SIGCHI New Zealand chapter's international conference, pp. 83–90 (2006)
39. Akado, Y., Kogure, S., Sasabe, A., Hong, J.-H., Saruwatari, K., Iba, T.: Five Patterns for Designing Pattern Mining Workshops. In: EuroPLoP '15, July 08-12, 2015, Kaufbeuren, Germany (2015)
40. Hagge, L., Houdek, F., Lappe, K., Paech, B.: Using Patterns for Sharing Requirements Engineering Process Rationales. In: Dutoit, A.H., McCall, R., Mistrík, I., Paech, B. (eds.) Rationale Management in Software Engineering, pp. 409–427. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg (2006)
41. Hentrich, C., Zdun, U., Hlupic, V., Dotsika, F.: An approach for pattern mining through grounded theory techniques and its applications to process-driven SOA patterns. In: Kohls, C., van Heesch, U. (eds.) the 18th European Conference, pp. 1–16 (2015)
42. Diaz, P., Acuña, P., Aedo, I., Malizia, A.: A Design Patterns Catalog for Web-Based Emergency Management Systems. In: D'Atri, A., Marco, M. de, Braccini, A.M., Cabiddu, F. (eds.) Management of the Interconnected World. ItAIS: The Italian Association for Information Systems, pp. 387–394. Springer-Verlag Berlin Heidelberg, Heidelberg (2010)
43. Välimäki, A., Kääriäinen, J.: Requirements Management Practices as Patterns for Distributed Product Management. In: Münch, J., Abrahamsson, P. (eds.) Product-focused software process improvement. 8th international conference, PROFES 2007, Riga, Latvia, July 2 - 4, 2007 ; proceedings, 4589, pp. 188–200. Springer, Berlin (2007)
44. Fehling, C., Ewald, T., Leymann, F., Pauly, M., Rutschlin, J., Schumm, D.: Capturing Cloud Computing Knowledge and Experience in Patterns. In: 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), pp. 726–733 (2012)
45. Borchers, J.: The Aachen Media Space: Design Patterns for Augmented Work Environments. In: Lahlou, S. (ed.) Designing user friendly augmented work environments. From meeting rooms to digital collaborative spaces, pp. 261–312. Springer, London, New York (2009)
46. Gholami, M.F., Jamshidi, P., Shams, F.: A Procedure for Extracting Software Development Process Patterns. In: 2010 European Modelling Symposium (EMS), pp. 75–83 (2010)
47. Benedicenti, L., Succi, G., Vernazza, T.: From process modeling to domain modeling. SIGAPP Appl. Comput. Rev. 5, 28–32 (1997)

48. Franch, X., Renault, S., Mendez-Bonilla, O., Quer, C.: PABRE: Pattern-based Requirements Elicitation. In: Third International Conference on Research Challenges in Information Science (RCIS), pp. 81–92 (2009)
49. Welicki, L., Manuel, J., Lovelle, C., Aguilar, L.J.: Patterns meta-specification and cataloging. In: Tarr, P., Cook, W.R. (eds.) Companion to the 21st ACM SIGPLAN conference, p. 679 (2006)
50. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M.: A pattern language. Towns, buildings, construction. Oxford Univ. Press, New York, NY (1977)
51. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30, 611–642 (2006)
52. Chandra, L., Seidel, S., Gregor, S.: Prescriptive Knowledge in IS Research: Conceptualizing Design Principles in Terms of Materiality, Action, and Boundary Conditions. In: 2015 48th Hawaii International Conference on System Sciences (HICSS), pp. 4039–4048