

Please quote as: Dickhaut, E.; Janson, A.; ... Leimeister, J. M. (2020): Codifying Interdisciplinary Design Knowledge through Patterns – The Case of Smart Personal Assistants. *Design Science Research in Information Systems and Technology (DESRIST)* (bll 114–125). Springer, Cham.



Codifying Interdisciplinary Design Knowledge Through Patterns – The Case of Smart Personal Assistants

Ernestine Dickhaut¹(✉), Andreas Janson², and Jan Marco Leimeister^{1,2}

¹ University of Kassel, Kassel, Germany

{ernestine.dickhaut, leimeister}@uni-kassel.de

² University of St.Gallen, St.Gallen, Switzerland

{andreas.janson, janmarco.leimeister}@unisg.ch

Abstract. Smart personal assistants (SPAs) are proliferating into our daily lives and are a ubiquitous platform for providing digital services. However, when designing such innovative IT artifacts, interdisciplinary domain knowledge is often needed. For example, SPAs utilize a plethora of sensors and cloud-based computation of data to deliver high-quality services, but those services, for example, may conflict with current regulations of law, e.g., with the Data Protection Regulation (GDPR) in Europe. In that sense, approaches are needed to overcome the limited domain knowledge of developers. Thus, we propose in our study a pattern-based approach to codify interdisciplinary design knowledge. For this purpose, we derive theory-motivated requirements, develop design principles for patterns, and evaluate the utility of the patterns. Our results from a 2×2 fully randomized field study show that the provision of patterns for SPA development supports interdisciplinary design through better consideration of both service quality and law compatibility. Thus, we provide design contributions concerning how we effectively codify and communicate design knowledge and provide practical guidance for supporting interdisciplinary IS development.

Keywords: Smart personal assistants · Legal compatibility · Design knowledge

1 Introduction

Smart personal assistants (SPAs) such as Amazon Alexa are more popular than ever before. One key success factor of SPAs is the adaptability to the user, which offers new support in everyday life. Nonetheless, this is only possible by collecting and evaluating personal user data. Conversely, the protection of one's own data and legal aspects is becoming increasingly important when considering recent privacy scandals. Higher legal standards increase the pressure on developers of IT artifacts. However, developers often lack the necessary domain expertise to develop a legally compatible IT artifact such as SPAs. They do not correctly understand the contents of legal requirements and cannot take them into account satisfactorily in the development.

Therefore, we present a pattern-based approach for system development to improve the legal compatibility of an SPA while also considering the quality aspects of these IT artifacts. We provide a way that enables developers to, on the one hand, gain knowledge from other disciplines and, on the other hand, present concrete solution approaches during the development process. In this context, we also pay attention to the cognitive processes of the pattern user, i.e., the system designer. To achieve this goal, we map elements of cognitive load theory to the development and design of our design patterns. Based on relevant literature, we identify requirements for the codification of knowledge and derive design principles for the development of patterns considering cognitive load theory. Finally, our patterns are a manageable way to support developers in designing legally compatible SPAs but also to acquire expertise and use it for other purposes. The goal of our paper is to present a theory-driven design approach to provide a set of design principles for building design patterns that, ultimately, support designing legally compatible SPAs and are based on the following two research questions (RQ):

RQ: How should we codify interdisciplinary design knowledge in patterns to support SPA developers, and what are the effects of those patterns on IS development?

To answer our research question, we evaluate our developed patterns in an experimental user study. We pay particular attention to two aspects. On the one hand, we examine the extent to which a legally compatible SPA can be developed with the help of our patterns. On the other hand, we focus on the usefulness, comprehensibility, and cognitive processing of each design pattern.

2 Related Work

2.1 Smart Personal Assistants

SPAs can support everyday life in many ways, such as on smartphones, in cars, in service encounters, in smart home environments, or as support for elderly or impaired people [1]. According to Knotte et al., SPAs can be further differentiated into five archetypes: Adaptive Voice (Vision) Assistants, Chatbot Assistants, Embodied Virtual Assistants, Passive Pervasive Assistant, and Natural Conversation Assistants [1].

To develop SPAs, we have to pay more and more attention to the requirements of various disciplines. Key aspects of SPAs relate, for example, to their usability and user experience, which we can sum up as the overall service quality. Service quality is frequently being paid much attention during the development process. However, there is also growing skepticism and concern that these systems, for example, “listen” without being activated by a wake word [2], thus showing that quality perceptions are difficult to achieve with these devices. More importantly, legal requirements are often only addressed to a minimum extent in order to be compliant with the minimal requirements of law [3]. However, the protection of one’s data is also becoming important, especially when taking new legal regulations and user fears into account. In this context, system developers often lack the necessary domain expertise to be able to implement legal requirements for developing a legally compatible SPA. Furthermore, there is a lack of and great uncertainties in research on how to support developers in their design process of user assistance systems such as SPAs [4].

2.2 Codification of Design Knowledge

During the development process, requirements from unfamiliar disciplines are often addressed sparsely by system developers, e.g., ethical or legal requirements. One possible reason for this is the lack of expertise related to those domains [5]. Nonetheless, we know that humans need familiarity, even in system development processes. However, this familiarity is often not provided by other domains, e.g., legal regulations. Therefore, design knowledge and solution approaches for system development problems should be codified in a way that supports system developers. By reusing knowledge, effectiveness and efficiency can also be achieved, especially related to the knowledge transfer of solutions to new use cases [6].

The codification of design knowledge is gaining importance in the DSR field, also enabling transferability of design knowledge to other fields [6]. Design knowledge is used in DSR projects for various purposes. Gamma et al. [7] found that most knowledge is used for designing artifacts. Therefore, it is crucial to have the right design knowledge in the most helpful presentation form for the specific application. Different approaches, such as wikis and documentations [8], are already used to capture knowledge. These approaches do not offer a solution to the problem of codifying knowledge in a way that is understandable for disciplines outside the field. Facts such as missing structure, using technical terms, and incompleteness lead to low usage of these tools. It needs to be guaranteed that the design knowledge is formulated in clear, unambiguous, accessible language, without inconsistencies and contradictions [9].

In this context, requirement and design patterns are a solution to solve recurring problems and challenges [10]. In addition to the previously used application areas, they can now be used to enable a broad understanding of periphery disciplines. Patterns contain templates to describe information in tabular form [11] and represent established instruments to make complex knowledge accessible and applicable for system developers [12]. A pattern thus defines the basic structure of a solution for a specific problem but does not yet describe a complete solution to the problem. This approach supports the developer in solving the interdisciplinary problem but does not restrict their creativity by abstracting the pattern [12]. The use of patterns has been established in many different disciplines and has become an integral part of the basic education of a software developer.

3 Designing Patterns for Smart Personal Assistants' Design

3.1 Design Science Process and Methodological Considerations

We use the Design Science Research (DSR) process by Peffers et al. [13] for designing and evaluating our design patterns (see Fig. 1). In specific, we follow a problem-centered approach that centers around the lack of interdisciplinary design knowledge to develop a legally compatible SPA.

Therefore, we derive design requirements from the literature based on which we create design principles for patterns. The patterns should support developers to design a legally compatible SPA because there are a lot of law elements that must be taken into account when designing a legally compatible SPA, such as the protection of personal data and the transparency of data storage. To develop a satisfactory system for the user, our patterns also take quality into account [6].

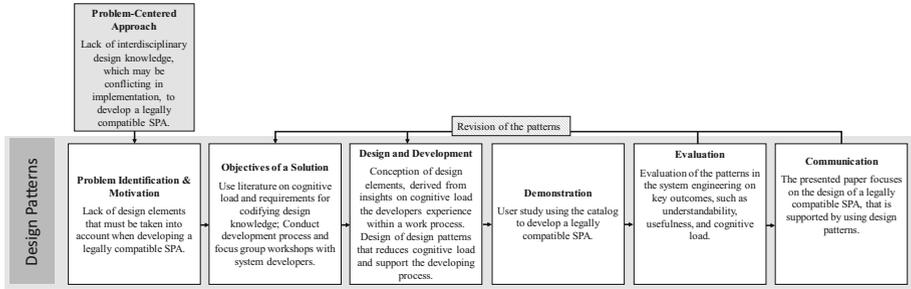


Fig. 1. Design Science Research Approach (adapted from [13])

3.2 Kernel Theories for Scaffolding Requirements Derivation

Design patterns are intended to support the user in the development of the specific system. We do not want the users to have the patterns at hand again and again during the implementation, but rather we want them to learn and expand their knowledge by using them. To derive our requirements for codifying design knowledge in design patterns, we draw on theories related to conceptualizing design knowledge, the representation of design knowledge, and refer in this paper especially to cognitive load theory (CLT). With its origins in educational psychology [14], CLT provides a theoretical framework for studying how individuals process information during learning and problem-solving processes while at the same providing guidance on how to structure information for better learning results [15]. We argue that using design patterns in interdisciplinary IS development is also a problem-solving process, where developers acquire design knowledge and apply it to complex problems, i.e., when conflicts occur that arise from conflicting requirements. Since it is important that patterns have a long-term learning effect on the users, we use CLT as the kernel theory.

Since the cognitive resources of an individual are limited, it is essential to manage cognitive load related to the pattern presentation when taking design advice offered by design patterns into account [16]. We derive in the following theory-based requirements to codify design knowledge in design patterns. The patterns are built on CLT as a kernel theory but also involve other concepts, such as research on codifying design knowledge.

3.3 Derivation of Theory-Driven Requirements

To ensure that our patterns have the highest possible added value for the user – in our specific case, SPA developers – we first conducted a literature research including various literature dealing with the codification and representation of design knowledge (see Table 1). By this means, we derive requirements from theory for codifying design knowledge to develop SPAs from an interdisciplinary perspective, thus addressing the first part of the objectives of the solution phase of the design science approach by Peffers et al. [13].

In addition to requirements for the structure and design of the patterns, the content is particularly important when creating the patterns. It is important to not only deal with the content of the interdisciplinary knowledge in the patterns but also to consider the

Table 1. Theory-driven Design Requirements

	Design requirements	Source
R1	A pattern should provide implementation guidance but should not restrict the pattern users in their creativity	[18]
R2	A pattern should present relationships between the involved disciplines and any necessary explanations and adjustments	[19]
R3	A pattern should contain a description of why it is useful to use the pattern to illustrate the importance and the added value of the pattern	[20, 21]
R4	The core content of the pattern should enable knowledge transfer, document knowledge, and provide good repositories for knowledge dissemination	[22]
R5	A pattern should provide additional domain-specific information and knowledge to support the implementation	[22]
R6	A pattern should account for different knowledge levels when considering domain-specific knowledge needed for SPA design	[22]
R7	The procedural steps of a pattern should be adjusted by a clear subdivision of the content to reduce the extraneous load	[22]
R8	The content of the pattern should be presented in a compact and clear way without inducing unnecessary extraneous load, i.e., by avoiding split-attention effects	[17]

structure of the patterns. Both factors are essential for the success of the patterns in practice. There are various requirements for elements, which must be presented and offered in any case in order to offer a practical use for the users of the patterns. That is why the patterns should contain not only a problem solution but also details about an introduction in the problem-solving context, information about actions, and concrete implementation examples. Doing so refers especially to the process of scaffolding problem-solving processes that lower the cognitive load by pointing out important aspects relevant to the task.

In order to guarantee general validity, the patterns should have general applicability. Chandra et al. assume that design knowledge is codified for the purpose of reuse, and even though reusing is sometimes associated with repetition, reuse has been observed in contexts that strive toward innovation and customization [17].

However, for the success of the patterns, not only the content and its preparation are important but also the presentation form. A clear presentation form increases the quality and usefulness of the patterns for the users. Chandra et al. highlight the importance that the design knowledge presented should be as compact and clear as possible [17]. Less extraneous load through a clear presentation form facilitates the work with the patterns [22]. Although the information is often codified in written form, the long continuous text makes the application of patterns more difficult. As such, there is a lack of an overview and the possibility to get the information you are looking for as quickly as possible. Therefore, the design knowledge presented should be supplemented by graphic representations [17].

3.4 Design Principles for Design Patterns

We identified in Sect. 3.2 requirements by analyzing literature to extract design-relevant knowledge from previous work, which helped us to address design principles for our case, the development of design patterns for the implementation of a legally compatible SPA (see Fig. 2). In addition, we also ground our design principles in literature concerned with scaffolding problem-solving, which has also proven to manage cognitive load [23]. To ground our artifact in practical relevance, we conducted a focus group workshop (n = 8) to justify the design principles derived from the literature. The participants in the workshop had both expertise in software engineering and knowledge of legal compatibility to assess the efficiency of the derived principles to solve the practical problem. In the workshop, requirements for patterns for the special field of application, the development of a legally compatible SPA, were jointly sought, and then possible approaches for solving conflicts between service quality and legal compatibility were elaborated.

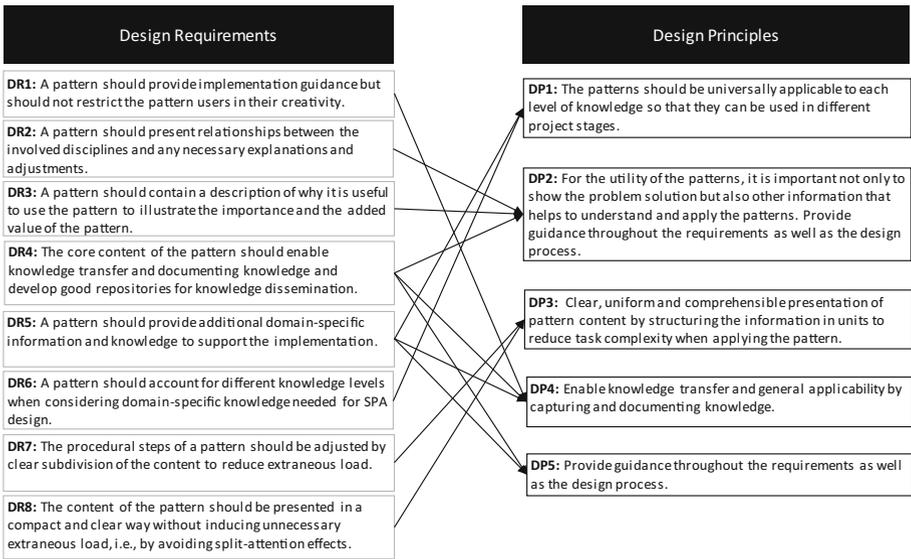


Fig. 2. Design Requirements and Design Principles for Interdisciplinary Design Patterns

To make sure our design patterns bring the highest possible added value for many different users, special attention must be paid to the different levels of knowledge that developers have. The design incorporates both technical knowledge and artistry and occurs as a reflective conversation between designers, their actions, and their situations [22]. Thus, we suggest our first design principle:

DP1: The patterns should be universally applicable to each level of knowledge so that they are reusable and applicable.

To support the user in solving the problem, a concrete example for the implementation is necessary [21]. The solution approach should include requirements, the relationship

between constructs and domain objects, explanation, and justification [19]. By adding further content and information, not only the comprehensibility of the problem and the possible solution are improved but also the expertise of the user is increased to solve related problems and engage to work on domain knowledge shortcomings [23]. This results in our second design principle:

DP2: For the utility of the patterns, it is important not only to show the problem solution but also other information that helps to understand and apply the patterns.

The content is not only of great importance for the applicability and added value of the pattern but also for the way the design knowledge is presented. For this purpose, a clear presentation method should be used in order to keep the cognitive process to a minimum [21, 24]. Thus, we suggest our third design principle:

DP3: A clear, uniform, and comprehensible presentation of pattern content by structuring the information in units to reduce task complexity when applying the pattern.

In addition to the actual problem-solving, information should be provided that allows the user to understand the context of the problem [21] and critical task aspects [25]. In this context, a pattern should also provide information about the actions of the artifacts, and according to boundary conditions, to enable knowledge transfer [21]. The patterns must be as generally valid as possible to be generally applicable. Hence, we propose:

DP4: Enable knowledge transfer and general applicability by capturing and documenting knowledge.

In order to have the greatest possible impact on the learning effect of the patterns from a cognitive point of view, they should be used throughout the development process [22]. Thus, we suggest our fifth design principle:

DP5: Provide guidance throughout the requirements as well as the design process.

3.5 Patterns for Designing Smart Personal Assistants

We developed patterns with the help of our design principles. For the development, we have decided on a procedure that consists of different methods, like workshops, expert interviews, and literature research.

First, we created requirement patterns as a prerequisite for design patterns by conducting a literature search and identified higher-level goals related to achieving service quality and legal compatibility (**DP5**). In the next step, we compared the goals of legal compatibility and service quality in order to identify possible connections and conflicts. The requirement patterns do not yet solve any concrete implementation problems in the development process but initially serve as support for the specification of legal and service quality requirements. Therefore, we need patterns to support the development process, namely design patterns (see Fig. 3 for an example design pattern). Based on **DP2**, our patterns start with an introduction, including links to relevant background information in the problem-solving context. That is why each pattern starts with its individual goal and, as with the requirement pattern, represents the best possible final

state for the developed system. As the design patterns are based on the already presented requirement patterns and their high-level goals, the influences on the respective patterns are presented in each design pattern.

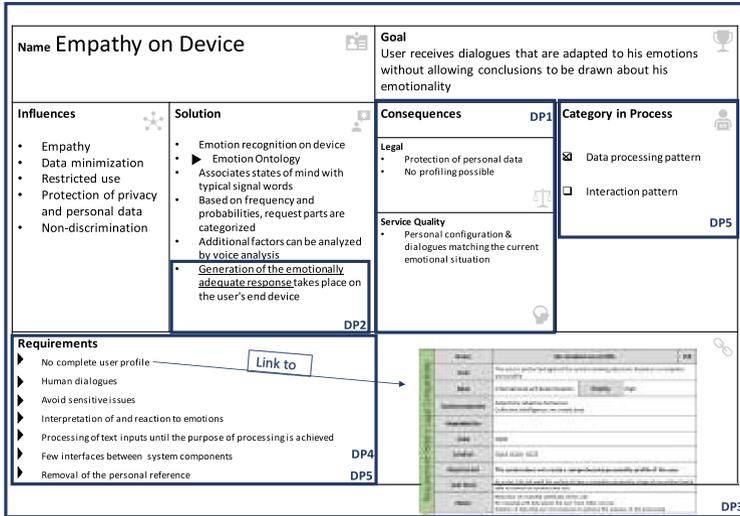


Fig. 3. Realization of the Design Principles in the Design Patterns

Thus, legal influences and service quality influences, as well as their priorities, are considered in the patterns (DP4). At the core of the pattern are possible solution approaches. They demonstrate concrete, reusable solutions and explain their positive and negative consequences (DP1). We have taken great care to ensure that all patterns follow the same structure to guarantee the highest possible added value through use (DP3). Because we have already paid special attention to cognitive load theory when creating the design principles, our design patterns also consider cognitive processing.

4 Evaluation

With the following evaluation, we address the phases of demonstration and evaluation as depicted in the design science cycle [13] by evaluating the patterns and their utility to scaffold interdisciplinary IS development projects. For our evaluation, we drew on the framework for evaluating DSR [26] and chose a formative ex-post evaluation approach [26, 27] that will help us to form the design of our method early in the development process.

4.1 Methodological Aspects

The data collection took place in the university course “Business and Information Systems Engineering” in a Western European university and was embedded in a fully randomized between-subject experiment. The patterns were applied by students, who were

trained in requirements engineering as well as systems design; thus, they are a suitable sample for evaluating the utility of our artifact. Overall, the task was to prototypically design a chatbot application. In total, 22 groups of two persons each, who all studied in business with a major in information systems, participated in the voluntary task that was incentivized with three extra credits for the final exam. To ensure the seriousness of participation, extra credit could only be obtained when individuals followed the tasks as described.

The subjects were provided with a mockup and received information regarding the case, such as interview documents of the clients and customers. In addition, they pursued the overall task of developing a chatbot prototype of an SPA for higher education that supports teaching. In order to meet the requirements of university teaching and the latest data protection regulations, it was necessary to make the SPA as legally compatible as possible but still ensure high service quality. The participants were completely randomly divided into two groups. Concerning the experimental evaluation, we designed a 2×2 experiment with two experimental manipulations that correspond to the provision of (1) requirement patterns ($n = 6$), (2) design patterns ($n = 6$), and (3) requirement patterns as well as design patterns ($n = 6$). In that sense, the experiment also included a control group ($n = 4$) without the support of patterns during the development process of the prototype (see Fig. 4).

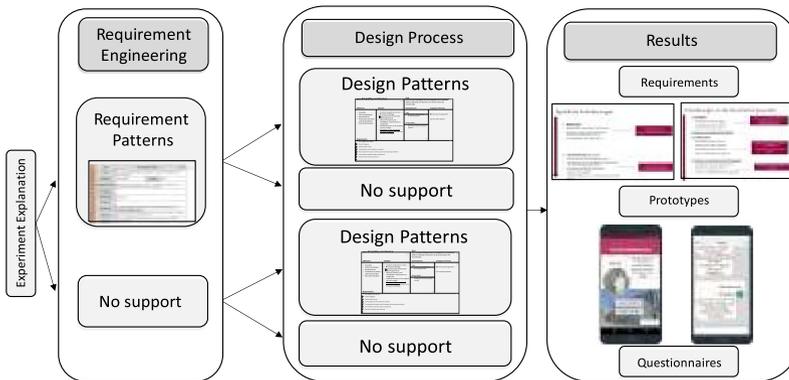


Fig. 4. The Experimental Procedure and Core Results

We used the developed prototypes for a rating and evaluation procedure by five subject matter experts experienced in digital service design as well as the legally compatible design of IT artifacts concerning the two outcome variables of interest, service quality and legal compatibility. The selected experts did not know the used patterns. All experts have scientific publications either in the field of service quality ($n = 3$) or legal compatibility ($n = 2$).

4.2 Results

The experiment was conducted in groups of two persons each. The control group (CG) counted four groups. The treatment groups (TG) consisted of 18 groups in total, of which

six groups were subject to a different treatment. All subjects were about the same age, with a range from 21 to 26 ($\bar{x} = 24.34$). We conducted the field experiment with two goals in mind: first, we investigate how the design principles provide the benefits and added value. Second, we examine how interdisciplinary knowledge through patterns is used by system designers. Both aspects are evaluated by the expert panel assessment of the developed prototypes. The results of the expert evaluation are presented in the table below. The results (see Table 2) show that the subjects who have used both the requirement and design patterns have also had significantly better ratings in service quality and legal compatibility in the expert evaluation. In the groups that used the pattern, more attention was paid to the implementation of legal criteria.

Table 2. Comparison between Control and Treatment Groups’ Prototypes

Legal Compatibility & Service Quality			Groups	Rating (1-5)	Post hoc tests		
	Group	Support	N	Mean (SD)	RP	DP	RPDP
	CG	No Support	4	2.63 (.95)	.88	.45	.02*
	TG 1	Requirement patterns (RP)	6	2.66 (1.07)		.36	.01*
	TG 2	Design patterns (DP)	6	2.38 (1.03)			<.05*
	TG 3	Requirement and Design patterns (RPDP)	6	3.16 (1.05)			

In addition to the expert evaluation, we have conducted further surveys, in the form of questionnaires, concerning the use of the patterns. The questionnaires contained, on the one hand, the content and, on the other hand, the form of presentation of the used patterns. We used the already proven questionnaire for evaluating requirement patterns from [28], which we adapted to our evaluations. An exemplary item examined the degree of abstraction and the concreteness of the content associated with it. Items that deal with the structure and presentation of the patterns achieve particularly good results. The clarity of the content is particularly well received.

Both evaluations show that the form of presentation and the content of the interdisciplinary expertise in the patterns lead precisely to the desired goal: namely, the design of an SPA that is both legally compatible and still includes high service quality, which in practice often leads to conflicts during development.

5 Discussion and Implications

To support developers to design a legally compatible SPA, we derived requirements based on the literature on codifying design knowledge [17–22]. With the help of the requirements, we were able to generate design principles that are applied to generate requirements and design patterns for interdisciplinary knowledge [11]. We followed the design science research approach of Peffers et al. to extract our design principles [13] for designing patterns.

We hypothesized that interdisciplinary patterns in the development process better support SPA developers. Therefore, especially for interdisciplinary projects, there often

arise conflicts between different requirements, which led us to consider the codification of knowledge first. Our results confirm that the codification of knowledge in requirement and design patterns leads to better results regarding the legal compatibility and service of SPAs.

Our design principles for patterns are not without limitations that, at the same time, present avenues for future research. Moreover, we had the design principles tested by potential users to rely on these insights to improve the patterns further. Regarding our pattern design, our next steps will be to improve the patterns by integrating the feedback of the study subjects. Based on this feedback, we can revise and improve our design principles for pattern design. In addition to the preparation and the presentation of the content, a particular focus is given to the ability to learn through the patterns in the following improvement. In this context, the findings of cognitive load theory play a key role and need to be further integrated. An overall goal of the patterns is the extraction of interdisciplinary knowledge through the use of our patterns. Future research could also validate if improving the learning processes of information system developers leads to an improvement of knowledge transfer to other development projects. This would enable us to capture the utility of patterns beyond single development projects. The method was evaluated partially with students. We remark that professional IS developers would have potentially applied the method differently, which would lead to different results. Especially the evaluation of the patterns could be more comprehensive to show the implications of practical software development and the learning from patterns. We evaluated the patterns and the developed prototypes but not directly our design principles. Nevertheless, by evaluating the patterns, conclusions can be made about the quality of the design principles.

To answer our RQ, we would like to summarize: first, we provide insights into the understanding and difficulties of design knowledge codification in interdisciplinary projects, which, in practice, is particularly difficult to implement. Second, our patterns contribute to the ongoing discussion in design science research on providing real guidance for how to design IT artifacts [29] and how to accumulate design knowledge.

Acknowledgements. This paper presents research that was conducted in context of the project “AnEka” (project number:348084924), funded by the German Research Foundation (DFG).

References

1. Knote, R., Janson, A., Söllner, M., Leimeister, J.M.: Classifying smart personal assistants: an empirical cluster analysis. In: Proceedings of the 52nd HICSS (2019)
2. <https://www.wired.com/2016/12/alex-and-google-record-your-voice/>
3. Hoffmann, A., Schulz, T., Zirfas, J., Hoffmann, H., Roßnagel, A., Leimeister, J.M.: Legal compatibility as a characteristic of sociotechnical systems. *Bus. Inf. Syst. Eng.* **57**(2), 103–113 (2015). <https://doi.org/10.1007/s12599-015-0373-5>
4. Maedche, A., Morana, S., Schacht, S., Werth, D., Krumeich, J.: Advanced user assistance systems. *BISE* **58**, 367–370 (2016)
5. Knote, R., Söllner, M., Leimeister, J.M.: Towards a pattern language for smart personal assistants. In: PLoP 2018 (2018)

6. vom Brocke, J., Winter, R., Hevner, A., Maedche, A.: Accumulation and evolution of design knowledge in design science research—a journey through time and space. *J AIS* (2019)
7. Sturm, B., Sunyaev, A.: A good beginning makes a good ending: incipient sources of knowledge in design science research. In: *ICIS 2019 Proceedings* (2019)
8. Phuwanartnurak, A.J.: *Interdisciplinary Collaboration through Wikis in Software Development*. IEEE, Piscataway (2009)
9. Lukyanenko, R., Parsons, J.: Design theory indeterminacy: what is it, how can it be reduced, and why did the polar bear drown? *J. Assoc. Inf. Syst. (JAIS)* 1–59 (2020)
10. Alexander, C.: *A Pattern Language: Towns, Buildings, Construction*. Oxford (1977)
11. Knote, R., Baraki, H., Söllner, M., Geihs, K., Leimeister, J.M.: From requirement to design patterns for ubiquitous computing applications. In: *EuroPlop 2016*, pp. 1–11 (2016)
12. Thies, L., Knote, R., Jandt, S., Söllner, M., Roßnagel, A., Leimeister, J.M.: *Anforderungs- und Entwurfsmuster als Instrumente des Privacy by Design*. Springer (2018)
13. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. *JMIS* **24**, 45–77 (2007)
14. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* **63**, 81–97 (1956)
15. Sweller, J.: Cognitive load during problem solving: effects on learning. *Cogn. Sci.* **12**, 257–285 (1988)
16. Kirschner, P.A., Ayres, P., Chandler, P.: Contemporary cognitive load theory research. The good, the bad and the ugly. *Comput. Hum. Behav.* **27**, 99–105 (2011)
17. Chandra Kruse, L., Nickerson, J.: *Portraying Design Essence* (2018)
18. Ahrens, J., Sankar, C.: Tailoring database training for end users. *MIS Q.* **17**(4), 419–439 (1993)
19. Heinrich, P., Schwabe, G.: Communicating nascent design theories on innovative information systems through multi-grounded design principles. In: Tremblay, M.C., VanderMeer, D., Rothenberger, M., Gupta, A., Yoon, V. (eds.) *DESRIST 2014. LNCS*, vol. 8463, pp. 148–163. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06701-8_10
20. Garud, R.: On the distinction between know-how, know-what, and know-why (1997)
21. Chandra, L., Seidel, S., Gregor, S.: Prescriptive knowledge in IS research: conceptualizing design principles in terms of materiality, action, and boundary conditions. In: *HICSS*, pp. 4039–4048 (2014)
22. Chandra Kruse, L., Seidel, S.: Tensions in design principle formulation, pp. 180–188 (2017)
23. Janson, A., Söllner, M., Leimeister, J.M.: Ladders for learning: is scaffolding the key to teaching problem solving in technology-mediated learning contexts? *AMLE* (2019)
24. Doering, A., Veletsianos, G.: Multi-scaffolding environment: an analysis of scaffolding and its impact on cognitive load and problem-solving ability. *JECR* **37**, 107–129 (2007)
25. Reiser, B.J.: Scaffolding complex learning: the mechanisms of structuring and problematizing student work. *J. Learn. Sci.* **13**, 273–304 (2004)
26. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: a framework for evaluation in design science research. *Eur. J. Inf. Syst.* **25**, 77–89 (2016)
27. Sonnenberg, C., vom Brocke, J.: Evaluation patterns for design science research artefacts. In: Helfert, M., Donnellan, B. (eds.) *EDSS 2011. CCIS*, vol. 286, pp. 71–83. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33681-2_7
28. Janzen, A., Hoffmann, A., Hoffmann, H.: *Anforderungsmuster im Requirements Engineering*. Information Systems, Kassel University (2013)
29. Gregor, S.: The nature of theory in information systems. *MIS Q.* **30**(3), 611–642 (2006)