

Please quote as: Dickhaut, E.; Li, M. M.; Janson, A. & Leimeister, J. M. (2021):  
Developing Lawful Technologies – A Revelatory Case Study on Design Patterns. In:  
Hawaii International Conference on System Sciences (HICSS) (Vol. 54).

## Developing Lawful Technologies – A Revelatory Case Study on Design Patterns

Ernestine Dickhaut  
University of Kassel, Germany  
ernestine.dickhaut@uni-kassel.de

Mahei Manhai Li  
University of Kassel, Germany  
mahei.li@uni-kassel.de

Andreas Janson  
University of St.Gallen, Switzerland  
andreas.janson@unisg.ch

Jan Marco Leimeister  
University of Kassel, Germany & University of St.Gallen, Switzerland  
leimeister@uni-kassel.de & janmarco.leimeister@unisg.ch

### Abstract

Higher legal standards regarding the data protection of individuals, such as the European General Data Protection Regulation, increase the pressure on developing lawful systems. In the development of technologies, not only developers are involved. It also requires knowledge from other stakeholders, such as legal experts, that lack technical knowledge but are required to understand IT artifacts. We see two strings that can benefit from the use of design patterns: first, the well-known use of design patterns to support developers in case of recurring problems. Second, we see potential that legal experts, who have to interact with and understand complicated, novel technologies, benefit from the same patterns. We conduct a revelatory case study using design patterns to develop and assess a smart learning assistant. We scaffolded the case interpretation through the human-centered view of socio-materiality and provide contributions concerning the use of design patterns in the development and assessment of lawful technologies.

### 1. Introduction

The benefits of design patterns to design systems have been verified by various studies over the last five decades [8, 44]. Design patterns offer solution approaches for recurring problems, which originate from the seminal work by Alexander et al. in the field of architecture [1]. Today, design patterns play an essential role in software engineering [11] and human-computer interaction (HCI).

With the rising number of information systems (IS) that permeate our everyday life, the requirements for developing such systems become increasingly complex. Nowadays, system development converges a multitude of different disciplines. Thus, not only software developers are required for the implementation of IS, but rather different stakeholders

need to be involved as well. Consider, for example, the development of COVID-19 tracing apps or the design of smart assistants, such as Amazon's Alexa: both system development examples demonstrate that privacy-friendly development processes are key to both user acceptance and market success [34]. Higher legal standards with regards to the data protection of individuals, such as the European General Data Protection Regulation (GDPR), are increasing the pressure on developers of IT artifacts. Nonetheless, developers often lack the necessary legal expertise [8]. In practice, many important measures necessary to launch a system on the market are often only considered at the end of the development process. For developing a lawful system, it would be more prudent to involve the legal experts much earlier, such as during the design phase. This ensures that the legal expertise and legal knowledge can be leveraged much earlier to develop a lawful IS that considers legal aspects and societal consequences. In this context, the aforementioned design patterns could be a feasible way to improve the design of IT artifacts, ensure the legality of norms such as the GDPR, but also contribute to sustainable IS. Apart from the actual development of an IS, the assessment of its lawfulness is also a decisive step in determining whether a system may be market ready and sustainable through compliance and lawfulness.

Up until now, knowledge related to the two-fold value of design patterns during either artifact design or artifact evaluation has mostly been looked at separately. This is an important gap, since we see two strings that can benefit from the use and further understanding of design patterns, practically and theoretically. Similar to system development, design patterns can be used to impart design knowledge to both developers and other-disciplinary stakeholders. On the one hand, it is well known that the use of design patterns supports developers and designers in case of recurring problems. On the other hand, legal experts have to interact with and understand complicated,

novel technology and could also benefit from the same design patterns. The goal of our paper is to present an approach that uses a theoretical lens to consider the use of design patterns for two different application scenarios and is based on the following research question (RQ):

**RQ:** How beneficial are design patterns in different application contexts from the development to the assessment of technologies?

To answer our research question, we conduct a revelatory case study and scaffold the case interpretation through the human-centered view of socio-materiality. The used patterns include design knowledge from both disciplines, IT and law. In the development phase, they can provide solutions for recurring problems, and in the assessment phase they offer details for the technical implementation and corresponding explanations. The case focuses on both the development and use of the developed smart learning assistant in a university course. We accompany the case from the beginning and are able to investigate both the development of the learning assistant and its subsequent four legal assessments, which are conducted by professional lawyers and judges in simulated court cases.

## 2. Theoretical Background

### 2.1 Socio-Materiality of Formulating Design Knowledge

When referring to the understanding and use of design knowledge in IS research, we consider the design science research (DSR) paradigm, which focuses on the development and evaluation of new technologies. As vom Brocke et al. [39, p. 5] highlight, the "*goal of DSR is to generate knowledge on how to effectively build innovative solutions to important problems*". In this context, rules and concepts are applied, such as design theories, design principles and design patterns, which can be used to map and support design processes [29]. Design knowledge, namely knowledge to design a system, consists of methods and constructs for designing systems [13]. Good design should not only be used for a "single success story" [6]. The reusability of and learning from the design knowledge is critical to the success of DSR projects and beyond [39].

To make design knowledge reusable in the future, it must be codified. The codification of design knowledge requires special methodologies. There are already various approaches to codify design knowledge like knowledge maps [41], mind maps [41], conceptual maps [45], wikis [30], prototypes [42], design principles [36], cheat sheets [37], and

design patterns [11]. To codify design knowledge, it is important to highlight the properties of their formulation [7]. For this purpose, socio-materiality [18, 25, 26] is a powerful instrument when considering the theoretical basis of IT artifacts in practice [27, 35]. Socio-materiality focuses interactions between humans and nonhumans, which [22]. Thus, socio-materiality is a prime candidate when researching the interface of IT artifact design knowledge and human centrality. The result of the interactions between people and nonhumans, which are often technologies, memories, or intentions, is the social part [10]. Materiality, as a characteristic of technology, is part of socio-materiality, which represents a set of activities with institutions, norms, discourses, and everything that can be defined as "social" [23]. The action and use of a materiality differ depending on the context without changing the materiality [23].

Socio-materiality can be used to explain how knowledge, the intellectual owner, and known representations, subjects, and objects are jointly related [15]. Fenwick et al. [10] use socio-materiality in education to show what effects "things" such as teachers, learners, learning activities, and knowledge representations as texts, curriculum content, and pedagogy have on each other.

Kruse et al. implicitly apply socio-materiality to design principles and relate categories that characterize design knowledge: 1) focus on the technology, 2) the use of technology, and 3) focus on the technology and the use of technology [5]. While the first category, according to socio-materiality theory, focuses on materiality, the second category relates primarily to action, and the third category includes both [5]. Similarly, design patterns follow the same categories. First, the materiality part consists of system properties, specifically represented by all system functionalities. Second, the action part contains the final causes of the system, such as the purpose of the material system properties. Following Leonardi's [19] thoughts, in which a materiality can be applied in different contexts without changing, design patterns can be used in different contexts.

After outlining the foundations of design knowledge, we lead over into the following to design patterns as concrete objects of analysis for the revelatory case study presented afterwards.

### 2.2 Design Patterns for legal IS Development

Patterns are, for instance, used in HCI to teach design concepts [4, 8, 17, 40]. In that sense, patterns are not only used to present solutions to problems but also to pass on knowledge. In the development and

design of lawful systems, patterns have already been applied by mapping legal knowledge to patterns. Thereby, approaches are followed that should enable developers to extract legal knowledge for the development of lawful systems through patterns [14, 33, 44].

Design patterns present solutions to solve recurring problems and challenges [1] and are established tools to make complex knowledge accessible and applicable. Patterns contain templates to describe (design) information in oftentimes tabular form and represent established instruments to make complex knowledge accessible and applicable to developers [2]. Furthermore, patterns document known and proven solutions to recurring problems. Therefore, design patterns are often applied as formalizations of design knowledge. A pattern defines the basic structure of a solution for a specific problem. Still, it does not yet describe a complete solution to a given problem, thus leaving a certain degree of flexibility related to the solution space and not restricting design creativity by enabling abstraction of the pattern. Hence, design patterns respond to the projectability issue of design knowledge and design knowledge reuse as proposed by vom Brocke et al. [39].

By capturing complex design knowledge, patterns also make legal knowledge accessible to IS developers. Especially data protection and data processing legal requirements have drastically changed in recent years [31]. Therefore, legal norms, as part of the social system, are becoming increasingly important in user-centered system development [38]. In this context, lawfulness means that the legality of a system is the minimum legal requirement for it to be approved on the market. To achieve this, measures for the protection of personal data must be implemented.

It is important to note that even if a system has been developed according to the best legal knowledge available, in practice this does not automatically guarantee that the system does not constitute any violations of the law in the future. In practice, once someone suspects a violation of the law, the technology must be subject to legal court cases and during the negotiations and the legal discourse a judge decides upon the lawfulness of the system. The legal practice in negotiations argues the technology's lawfulness based on the advocates' knowledge to represent facts on behalf of their clients [3]. For this purpose, lawyers apply their legal knowledge to the information and knowledge that they receive from their clients by using documents, such as contracts or documentation [21]. Socio-materiality frames this notion of lawful until proven otherwise, by providing

a frame for explaining the mechanisms of the legal sensemaking using design patterns during its legal assessment.

### **3. Methodology**

#### **3.1 Research Design and Case Selection**

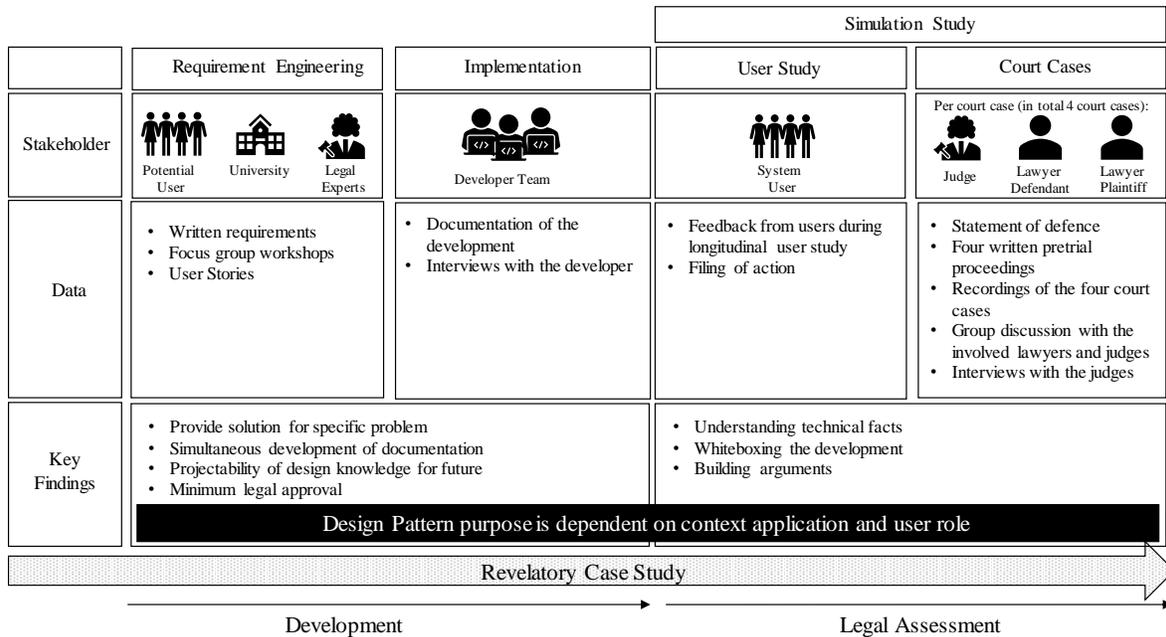
For investigating our research question, we draw on a revelatory case study approach that aims to investigate the use of design patterns in a new application context, the legal assessment of technologies. Referring to Yin [43], our case offers the opportunity to observe and analyze a phenomenon previously inaccessible to social science inquiry.

We have accompanied a development project for a voice-based intelligent learning assistant in which we had the opportunity to gain insights, first, into the development with interdisciplinary design patterns and, second, into the legal assessment of the developed learning assistant. The latter includes court cases in which the lawfulness of the learning assistant is negotiated. We were able to accompany the entire development of the learning assistant, the use of the learning assistant by its user base, and the resulting legal disputes arising from user complaints, which must be clarified in court.

According to Yin's [43] five components of case studies, a research method is especially useful when (1) the main research questions are "how" or "why" questions, (2) the researcher has little or no control over behavioral events, (3) the focus of the study is a contemporary phenomenon, and (4) the logic links the data to the propositions and (5) the criteria for interpreting the findings. This is true in our case: (1) with our RQ we strive to derivate insights into the interdisciplinary use of design patterns; (2) we have no influence on behavioral events, since we do not interfere in the interactions between the developers and the lawyers and judges in the court cases, (3) the development of lawful systems is becoming increasingly important, especially in Europe, where innovations in the GDPR have created stricter legal requirements for technologies; (4) the linking of our data to the proposition is done by pattern matching according to Yin [43]; and (5) we use the strategy to identify, address, investigate, and (if appropriate) reject rival explanations to our findings.

#### **3.2 Data Collection and Analysis**

Our data collection can be divided into two phases: (1) the development of the learning assistant and (2) the use of the technology by real users resulting in the



**Fig. 1** Overview and Key Findings of the Revelatory Case Study

legal assessment of the learning assistant. Figure 1 depicts the key parts of the case and highlights the rich data sources to enable data triangulation to support our findings afterwards. To pursue our research question, we used insights from different sources: (1) literature from DSR and socio-materiality theory to gain an overview and identify relevant concepts related to the codification of design knowledge, (2) documentation of the development, (3) interviews with the developers, (4) feedback from the users of the learning assistant, (5) four written pretrial proceedings, (6) the recordings of four court cases, followed by a (7) group discussion with the involved lawyers and judges, and (8) interviews with the two judges. The aim of the design patterns is to develop lawful technologies. This includes the protection of privacy but goes further beyond that.

To evaluate the lawfulness of the learning assistant, a law simulation study was conducted. The simulation study is a well-known evaluation method among law researchers for capturing the legal compatibility of IT artifacts [30]. The study is carried out in court cases according to European law. In a simulation study, court cases with real lawyers and judges, and a legal dispute are simulated to clarify the state of facts. To capture possible conflicts with the law, a user study was conducted beforehand. The simulation study includes four court cases in which the legality of a digital smart learning assistant was negotiated. In simulated court cases under real conditions with real judges and lawyers, we had the unique opportunity to look "behind the scenes" and to question the involved

persons about their argumentation and judgment formation in the discourse. This would not be possible in a real court case for reasons of data protection. Like in reality, written negotiations between the plaintiff, the defendant, and the judge were conducted before the negotiations. The entire correspondence was made available for our analysis. Besides that, we were allowed to take part in the hearings and document them with sound recordings so that we could evaluate them afterwards. Finally, in a focus group and interviews further questions could be asked to the participants to extract more in-depth insights and subjective evaluations as well as ascertain the need for the design patterns used. To analyze our data and to gain insights into the use of the design patterns, we conduct a structured qualitative content analysis according to Mayring [20]. The coding corresponds, on the one hand, to Yin [43], and, on the other hand, our coding was open for novel insights emerging from our data [28].

#### 4. Case Narrative & Socio-material Perspective on Design Patterns

Inspired by the socio-material lens, we look at the data of our revelatory case study. We consider two fields of application in which design patterns are particularly useful. Figure 2 shows an exemplary

Processing Emotional Data		Time in Development Process	
		<input type="checkbox"/> Interaction Pattern	<input type="checkbox"/> Architecture Pattern
		<input type="checkbox"/> Learning Pattern	<input checked="" type="checkbox"/> Data Processing Patterns
<p><b>Goal</b></p> <p>Users should receive dialogues that are adapted to their emotions. Nevertheless, data that allow conclusions to be drawn about the user's emotionality should neither be processed nor stored or used for profile building.</p>			
<p><b>Law</b></p> <ul style="list-style-type: none"> <li>• Non-linkability</li> <li>• Processing of sensitive data only with permissions</li> <li>• No discriminatory decisions</li> </ul>	<p><b>Requirements</b></p> <ul style="list-style-type: none"> <li>• No processing of intimate data</li> <li>• No complete user profile</li> <li>• Setting options for users</li> </ul>	<p><b>Technical Aspects</b></p> <ul style="list-style-type: none"> <li>• Friendly dialogues</li> <li>• Human dialogues</li> <li>• Interpretation of and reaction to emotions</li> <li>• Avoiding sensitive issues</li> </ul>	
<p><b>Law</b></p> <ul style="list-style-type: none"> <li>• Protection of personal data</li> <li>• Protection of intimacy and privacy</li> <li>• No profile creation</li> </ul>	<p><b>Consequences</b></p> <ul style="list-style-type: none"> <li>• Personal configuration</li> <li>• Dialogues appropriate to the current emotional situation</li> </ul>	<p><b>Influences</b></p> <ul style="list-style-type: none"> <li>• Empathy</li> <li>• Data minimization</li> <li>• Aim binding</li> <li>• Protection of privacy and intimate sphere</li> <li>• Non-discrimination</li> </ul>	
<p><b>Solution</b></p> <ul style="list-style-type: none"> <li>• Emotion recognition on the device through an emotion ontology <ul style="list-style-type: none"> <li>Three steps to recognize the emotions of the user: <ol style="list-style-type: none"> <li>1. signal processing: digitalization of the acoustic/visual signal</li> <li>2. feature calculation: a feature selection algorithm selects the most important features of emotions from the signal</li> <li>3. comparison of the characteristic with the database, assignment of the characteristic to a specific emotion</li> </ol> </li> </ul> </li> <li>• Link to typical signal words: Based on frequency and probability a categorization is made</li> <li>• Additional factors can be provided by speech recognition</li> <li>• Generation of an emotionally adequate response takes place on the user's terminal device</li> </ul>			
<p><b>Important data protection regulations</b></p> <ul style="list-style-type: none"> <li>• Article 5 sect. 1 lit. b (aim binding), lit. c (data minimization) (here, if necessary, opening clauses such as article 6 sect. 3 GDPR and member state regulations based thereon in the BDSG, HDSIG and HHG must also be observed, especially for data processing by public bodies)</li> <li>• Article 9 GDPR (processing of special categories of personal data) (here, if necessary, opening clauses in Article 9 sect. 4 GDPR and member state regulations based thereon in the BDSG, HDSIG and HHG must also be observed, especially for data processing by public bodies)</li> <li>• Article 22 GDPR (automated decisions in individual cases including profiling), (here, if applicable, Member State regulations based on the opening clause of Article 22(2)(b) GDPR must be observed)</li> </ul>			

Fig. 2 Selected Exemplary Design Pattern "Processing Emotional Data"

design pattern (see [9]), which was used for both development and legal assessment. First, we have a look at the development of a learning assistant using design patterns. Second, we look at the legal assessment of the learning assistant, in which the design patterns support legal experts.

#### 4.1 Using Design Patterns to Develop Lawful IS

The development of the smart learning assistant starts with its requirements analysis. The learning assistant is used for a university course and should support the lecturer during supervision and exam preparation. The technology must comply with the legal requirements of the university and is therefore subject to strict data protection regulations.

Based on the collected requirements, three programmers develop the learning assistant. To meet the requirements of both users and legal experts, the developers are provided with the aforementioned design patterns to support them. During development, the programmers were free to choose their work environment and their development approach while supplementing the requirements documents (including the legal requirements) with the design patterns. We extract three major benefits of applying design patterns in the development of lawful technologies for developers.

The Challenge of the Meaning of Law: Data protection regulations lead to strict legal requirements for the learning assistant. In addition to specifications for data storage, the data processing of the system is also decisive for its compliance with the university. Especially in interdisciplinary problems, where legal knowledge is required, the developers use the design patterns to point out possible solutions. There is a mismatch between understanding the law and its practice, which lead to somewhat difficult decisions in the development for the developers due to a certain degree of perceived vagueness. More precisely, the legal requirements for the system are rather inaccessible to developers, making the transfer from law (legal requirement) to practice (implementation) challenging. The developers use design patterns to build solutions that meet the strict legal requirements of the university (all following quotes translated to English):

*"I have little knowledge of data protection information and often did not know how to implement the requirements in practice. The design pattern showed me approaches to solutions that I could use as orientation (Developer 2)"*

The application of the design pattern requires the developers to apply the abstract design pattern to the concrete program. Thereby the design patterns provide a direction for a solution but not a complete solution.

Design Knowledge Reusability: All developers use the design patterns frequently to get a range of possible solutions. The design patterns are always used if no

solution is found for the problem to be solved. By using already proven solutions, the developers get a feeling of security. The patterns make that design knowledge accessible to future developers and create an opportunity to impart knowledge.

The developers compare the design patterns with the technical documentation, which is a "mandatory obligation" for the developers. In comparison to the documentation, the design patterns provide a benefit and thus an added value for the final development and achievement of the goal, whereas the documentation is very time consuming and does not necessarily add value to the system.

*"Now we can simply use the design pattern as documentation and have killed two birds with one stone. On the one hand, we get help in the development and on the other hand we save ourselves the tiresome documentation". (Developer 3)*

However, the design patterns do not completely replace the technical documentation of the development because they offer the user a solution scope through their abstraction in which he finds the concrete solution for the system to be developed.

**Nature of Benefit:** Besides the frequency of use of the design patterns, the nature of the solution provided by the pattern can be observed. In the observation of the development as well as in the interviews, we see how the developers start to recognize the relevance and importance of the legal requirements by using the patterns.

*"Finally, the legal requirements make sense, and I understand the purpose for which they have to be implemented." (Developer 2)*

The way the design patterns help does not refer to pure instructions according to the simple scheme of "if [this occurs], do [that]" but provide explanations and further information about the problem space and possible solutions. They offer a direction to follow and think about the problem. The developer uses existing knowledge, reflects upon it via the design pattern, and finally finds a suitable solution for the problem at hand. Therefore, design patterns hint the user to possible solutions and build specific domain knowledge. Therefore, design patterns indicate the reasons for the necessity to do something:

*"I get further explanations and hints for each approach, so I can understand what the individual specifications in the pattern are necessary for." (Developer 1)*

But there is not always a suitable design pattern for every problem that occurs, which means that for more novel problems, the developers need to invest more time in finding a comparable pattern-solution match.

We have thus shown how design pattern can be used to for the development of lawful IS. Now, let us have a look at how the design patterns are used for assessing its lawfulness.

## 4.2 Using Design Patterns to Assess the Lawfulness of IS

The use of the learning assistant has led to legal disputes in which users have turned to their lawyers due to, for example, unnecessary data storage or discrimination in the application process due to poor performance between the users and the university. To assess the lawfulness of the developed learning assistant a simulation study was conducted, which is a well-established methodology for testing technologies among legal experts [32]. The simulation study includes a user study in which the system users, in our case students, use the learning assistant under real conditions. Based on this use, possible legal infringements arise, which are then negotiated by lawyers. In our case, the assessment was a legal assessment of the previously developed learning assistant. This took place in four simulated court cases, which assessed the lawfulness of the learning assistant.

**Building Arguments:** The design patterns were first introduced in the written pre-negotiation. The defendant's lawyer used the patterns in his statement of claim for his evidence. The design patterns are used as evidence for the practical implementation in the system. Based on the patterns, the lawyer shows how, for example, data protection regulations were taken into account by using the design pattern "Data protection-friendly user profile". Thus, the lawyers and judges also received the design patterns for their argumentation, search for evidence, and the formation of judgments.

Not only were the lawyers able to form arguments on the basis of the design patterns, but the judges also used the design patterns to clarify the state of facts.

*"You can finally understand for what purpose the data is needed and no longer have the feeling that the data storage is carried out without any purpose." (Judge 1)*

In addition to the statements of both parties, an expert is asked to confirm the implementation of the design patterns. Although the design patterns are accepted as evidence, they do not confirm the actual implementation in the technology. On the one hand, the design patterns allow insights into the development, but, on the other hand, without written confirmation the judge requires the expert's statement of the implementation.

**White-boxing Development:** The design patterns were always used as soon as technical details were negotiated, and certain implementations were not clear. The design patterns were used to gain an understanding of the development process. Any court hearing aims to clarify the state of facts. Both sides contribute their evidence and argumentation. The

design patterns are used to refer to technical details into the lawyers' argumentation. For this, the defendant uses the details of the design pattern, which provides the possibility to use expert knowledge in understandable language:

*"Whenever I was at a loss with my arguments, I could find technical details of the programming in the design patterns and use them for my arguments." (Lawyer defendant 1)*

To use the patterns to form arguments, it is crucial that the content of the patterns can be understood and can also be used by laymen. According to this, the patterns represent a possibility to impart knowledge and technical understanding (also to users from other domains).

*"The technical information in the pattern is easy to understand [...]" (Lawyer defendant 2)*

*"They offer background information about the development details" (Lawyer defendant 1)*

**Legal Approval through Design Patterns:** The use of design patterns, at the very least, document that the developers have taken the legal aspects seriously and made the effort to adhere to previously considered lawful design knowledge. The judge who tries to understand the implementation of the learning assistant refers to the patterns in his questions to both parties. The questions are primarily further questions that challenge whether the descriptions of the patterns have been implemented in practice as described. In the focus group discussion, the judge underlines the observation:

*"The fact alone that the pattern has been taken into account in the development shows the importance of the protection of personal data." (Judge 2).*

This leads to the fact that the defendant's lawyer has to introduce a few technical details into the court case and, in case of ambiguity, refers to the design pattern and its application.

### 4.3 Theoretical Sensemaking of the Case

Socio-materiality helps us with understanding how design knowledge is both externalized and applied. Design knowledge starts as tacit knowledge of developers, which might be understandable but oftentimes not perceivable for others [24]. Design knowledge is currently still residing on a nonmaterial level. Once written down and formalized, knowledge becomes perceptible, thereby transferring that tacit knowledge through inscribing it into something external, such as a design pattern. Design patterns allow design knowledge to be externalized and codified. This moves the codified design knowledge from a nonmaterial level to a material level. The design pattern is, thus, the conduit or medium that allows intelligible knowledge to become perceptible

[15]. The codified design knowledge (nonmaterial) of design patterns (material) is inscribed into the fabric of technological artifacts (material). Hence, design patterns are used to develop specific systems.

The design patterns provide a combination of action-oriented guidelines, as well as explanations and effects for the developers to find a solution. Therefore, the patterns bridge the gap between initially unsolvable problems (problem space) and the suitable design knowledge externalized in the design patterns (solution space). For bridging the gap, the patterns provide an approach to share (interdisciplinary) domain knowledge. In the socio-material language, we can say by using design patterns, we are able to transfer unknown (domain) knowledge by inscribing it into the software. Due to the abstract nature of the pattern, the user must develop a suitable solution for the concrete application context. The patterns indicate a possible solution space. If we call the application of the design pattern by a user the *social part* of socio-materiality and the pattern the *material part*, the concrete solution in the learning assistant only emerges through the interaction of both components, the unfolding of *socio-materiality*.

In addition to the use of the design patterns in the development process, design patterns can be used for the assessment of a technology. They represent a supplement to the documentation and provide insights into the development which would otherwise not be visible. By applying the patterns in the court case, design patterns uncover technical details of the system and provide additional information that can help to understand and assess the system. In socio-material language, we can say: by the application of the design patterns in a court case, *the social part* of the design patterns unfold the technical details, which represent the *material part*, and other facts that were considered during the development. In other words, the subject system of both development and assessment is itself of socio-material nature. Legal experts (or us as researchers) come in contact with the system and equipped with our pre-conceived knowledge of all kinds, both social, technological or legal, we create the bidirectional connection with the system to try and project a specific nonmaterial aspect and its relation to the law. To help with this projection of the system's always changing socio-materiality, design patterns are used as material means to direct the user to the target direction. Simply put, the legal experts act upon the subject system, and, together with the design patterns, they try to figure out how the technical aspects of the system are related to the legal requirements. The re-application reverses the process of unfolding the inscribed knowledge by using the design patterns

during development and makes the knowledge accessible to the legal expert.

## 5. Discussion & Conclusion

In accordance to Leonardi [19], we observe that design patterns (materiality) exist independently of the user, but their affordances and constraints do not. Because the developers and the legal experts come to materiality with diverse goals, they distinguish between different possibilities for action. Thereby we observe how the affordances of patterns can change across different contexts even though the design patterns do not.

The kind of benefit and the purpose of design patterns depend on the context of the application and the user role. While developers use design patterns to develop a system and to find solutions to problems, exactly the same design patterns offer other user groups, in our case legal experts as an umbrella term for lawyers, judges, expert opinions, and other legal experts, a completely different benefit. Legal experts use the design patterns to understand software systems in the legal assessment. The design patterns lead to more precise assessment and evaluation for lawyers' legal perspective by making concrete implementations in the development recognizable and understandable.

The design patterns lead to the projectability of design knowledge for future developers and researchers. Thus, design patterns help to achieve one of DSR's core objectives, namely the transfer of design knowledge for future projects [39]. By using the design patterns to access the design knowledge of the development, the knowledge becomes perceptible [15]. The acquired knowledge can now be used for further work. The design patterns are abstract and can, therefore, be used in various application scenarios. Thereby one design pattern offers the explanation for many problems. Thus, design patterns can counteract one of the problems identified by Brocke et al. [39] in the reuse of design knowledge. The seemingly material patterns are actually of socio-material nature, which provides a basis to understand that the software is also of socio-material nature, which has the norms and other legal aspects embedded in its essence. Thus, design patterns act as a bridge between what the software system is (development) and the understanding of its legal practice (assessment).

By providing additional information, the developer creates a common understanding of the necessity of nonfunctional (primary legal) requirements. By providing additional information that goes beyond the description of the goal state, the understanding of the context is fostered. The design patterns provide information related to the codependency of

interdisciplinary details such as possible consequences of implementation for both disciplines. From a DSR perspective [39], this means that the context of the problem space is already part of the design patterns and linked to suitable solutions.

The use of the design patterns for the developer leads to the simultaneous development of a means of documentation. Usually, the documentation of the development is an unpopular part. The design patterns are used as a supplement to the documentation and, at the same time, offer support during the development. This saves the developer time and work that would otherwise flow into the documentation. Thus, the application of design patterns in the development shows a further benefit for the developer, who is now supported by one tool, the design patterns, both in the presentation of possible solutions and in the documentation of the development. However, the design pattern does not completely replace the expert's statement and the technical documentation.

In the legal assessment the design patterns offer the benefit that the developers can show that they have tried to address the legal requirements with a solution that has addressed at least a similar legal challenge elsewhere. Without considering each legal context, it is an indicator that can be used to argue for an intent to address the legal issues. This can be used in favor of the technology in court cases. The use of design patterns signals legal experts that the legal requirements (problem space) and tried and tested technical solutions (solution space) were at least attempted to be matched during the development (problem solution) [39]. This means that within the assessment, the legal experts do not have to discuss whether legal requirements were not considered but rather focus on how they were instantiated. In other words, it is clear that legal requirements have been considered, and argumentation is based on technical details as evidence as starting points for the discussion in the court. If there are open questions about the technology, the lawyers can refer to the design pattern, which reduces or even in our case completely eliminates the need to consult the developers or an expert. Nevertheless, design patterns remain new territory for legal experts and require openness. While some lawyers have based their argumentation largely on the design pattern, others have only consulted the pattern in an absolute impasse of argumentation.

Our analysis suggests that design patterns in terms of content can be classified into the categories according to Kruse et al. [5], i.e. a) action-oriented, b) materiality-oriented, and c) action- and materiality-oriented, but contain information that goes beyond these categories. By applying and using the design patterns, more than an instruction to do something and

an explanation of how to do it is created. The design patterns act as a bridge between design knowledge and the developed technology. In our case, either as support for developers or as support for the assessment of the developed system. By applying the patterns in two different scenarios, namely the development and legal assessment, we have seen that design patterns contain another characteristic. In comparison to design principles, the information in the design patterns extend the action-oriented and materiality-oriented information of design principles. It is precisely this content that distinguishes design patterns from design principles. Design principles can be defined as a rule or standard of conduct [16] that gives precise descriptions of how an artifact must be built [12]. Design patterns, on the other hand, contain information that justifies the purpose for which something must be implemented. Therefore, socio-materiality shows us that a design pattern is much more than just an established solution.

## 6. Future Research & Limitations

Our study has certain limitations that provide directions for future research. First, by having evaluated the use of the patterns in the simulation study, the results can only be generalized to a limited extent. Nevertheless, we have provided a first insight into the holistic use of design patterns from the development to the evaluation of a technology.

Second, with our case study we consider two fields of application. Further work should look at other application scenarios, where other challenges can be solved by design patterns.

Third, the feedback from the simulation study, as well as further findings from the literature, should be used to revise the design pattern. By reflecting the use of design patterns for different user groups with a socio-materiality lens, we have theoretically considered the practical application of design patterns. The reusability of design knowledge through codification in design pattern should in any case be considered and pursued theoretically. In the future, design patterns should be written so that they can be used in several application contexts, such as development and the legal assessment of technologies. In addition to these two application scenarios, others can also be discovered.

Fourth, so far, few design patterns have established themselves internationally in practice. To integrate the benefits of patterns in practice, further work should deal with requirements and challenges and uncover the reasons for them as well as find solutions.

## 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] Alexander, C., *A pattern language: towns, buildings, construction*, Oxford University Press, 1977.
- [2] Alexander, C., *The timeless way of building*, 24th edn., Oxford University Press, New York, 1979.
- [3] Bellucci, E. and J. Zeleznikow, "Managing Negotiation Knowledge with the goal of developing Negotiation Decision Support Systems", *ACIS 2005 Proceedings*, 2005.
- [4] Borchers, J., "Teaching HCI design patterns: Experience from two university courses", *CHI*, 2002.
- [5] Chandra, L., S. Seidel, and S. Gregor, "Prescriptive Knowledge in IS Research: Conceptualizing Design Principles in Terms of Materiality, Action, and Boundary Conditions", *HICSS*, 2014, pp. 4039–4048.
- [6] Chandra Kruse, L. and J.V. Nickerson, "Portraying Design Essence", *51st HICSS*, 2018.
- [7] Chandra Kruse, L., S. Seidel, and S. Purao, "Making Use of Design Principles", *DESRIST*, 2015, pp. 37–51.
- [8] Compagna, L., P.E. Khoury, F. Massacci, R. Thomas, and N. Zannone, "How to capture, model, and verify the knowledge of legal, security, and privacy experts", *International conference on Artificial intelligence and law*, 2007, pp. 149–153.
- [9] Dickhaut, E., A. Janson, and J.M. Leimeister, "Codifying Interdisciplinary Design Knowledge through Patterns – The Case of Smart Personal Assistants", *DESRIST*, 2020.
- [10] Fenwick, T.J., R. Edwards, and P. Sawchuk, "Emerging approaches to educational research: Tracing the sociomaterial", 2011.
- [11] Gamma, E., Helm, Richard, Johnson, Ralph, and J. Vlissides, "Design Patterns: Elements of Reusable Object Oriented Software", AddisonWesley Professional, 1994.
- [12] Gregor, S., "The nature of theory in information systems", *MIS Quarterly* 30 (3), 2006.
- [13] Gregor, S. and A.R. Hevner, "Positioning and presenting design science research for maximum impact", *MIS Quarterly*, 2013, pp. 337–355.
- [14] Hafiz, M., "A collection of privacy design patterns", *Proceedings of the 2006 conference on Pattern languages of programs*, 2006.
- [15] Hassan, N.R., "Editorial: A brief history of the material in sociomateriality", *ACM SIGMIS Database: the DATABASE for Advances in Information Systems* 47(4), 2016, pp. 10–22.

- [16] Hevner, A.R., S.T. March, J. Park, and S. Ram, "Design Science in Information Systems Research", *MIS Quarterly*, 2004, pp. 75–105.
- [17] Koukouletsos, K., B. Khazaei, A. Dearden, and M. Ozcan, "Teaching Usability Principles with Patterns and Guidelines", 2009, pp. 159–174.
- [18] Leonardi, "When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Agencies", *MIS Quarterly*, 35(1), 2011, p. 147.
- [19] Leonardi, P.M., "Materiality, sociomateriality, and socio-technical systems: What do these terms mean? How are they different? Do we need them.: Social interaction in a technological world", *Materiality and organizing: Social interaction in a technological world* 25, 2012.
- [20] Mayring, P., *Qualitative content analysis: theoretical foundation, basic procedures and software solution*, 2014.
- [21] Morcón, C.R., J.P. García, and J.A.S. Pizarro, "Knowledge management in a law firm", *Journal of knowledge management*, 2000.
- [22] Moura, E.O. de and M.d.S. Bispo, "Sociomateriality: Theories, methodology, and practice", *Canadian Journal of Administrative Sciences*, 2019.
- [23] Nardi, B.A., J. Kallinikos, and P.M. Leonardi, *Materiality and organizing: Social interaction in a technological world*, Oxford University Press, Oxford, 2012.
- [24] Nonaka, I. and R. Toyama, "The knowledge-creating theory revisited: knowledge creation as a synthesizing process", *Knowledge Management Research & Practice*, 1(1), 2003, pp. 2–10.
- [25] Orlikowski, W.J., "Sociomaterial Practices: Exploring Technology at Work", *Organization Studies*, 28(9), 2007, pp. 1435–1448.
- [26] Orlikowski, W.J. and C.S. Iacono, "Research Commentary: Desperately Seeking the "IT" in IT Research—A Call to Theorizing the IT Artifact", *ISR*, 12(2), 2001, pp. 121–134.
- [27] Orlikowski, W.J. and S.V. Scott, "10 Sociomateriality: Challenging the Separation of Technology, Work and Organization", *The Academy of Management Annals*, 2(1), 2008, pp. 433–474.
- [28] Ozanne, J.L., A. Strauss, and J. Corbin, "Basics of Qualitative Research", *Journal of Marketing Research*, 29(3), 1992, p. 382.
- [29] Peffers, K., T. Tuunanen, M.A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research", *JMIS*, 24(3), 2007, pp. 45–77.
- [30] Phuwanartnurak, A.J., *Interdisciplinary Collaboration through Wikis in Software Development*, IEEE, Piscataway NJ, 2009.
- [31] Politou, E., E. Alepis, and C. Patsakis, "Forgetting personal data and revoking consent under the GDPR: Challenges and proposed solutions", *Journal of Cybersecurity*, 4(1), 2018.
- [32] Pordesch, V., A. Roßnagel, and M. Schneider, "Simulation Study Mobile and secure communication in Healthcare", *DuD*, 1999, pp. 76–80.
- [33] Rossi, A., R. Ducato, H. Haapio, S. Passera, and M. Palmirani, "Legal Design Patterns: Towards A New Language for Legal Information Design", *Internet of Things. Proceedings of the 22nd International Legal Informatics Symposium IRIS*, 2019, pp. 517–526.
- [34] Security Week, "www.securityweek.com/zooms-security-and-privacy-woes-violated-gdpr-expert-says", accessed on June, 25st 2020.
- [35] Seidel, S. and N. Berente, "Toward "Third Wave" Information Systems Research: Linking Sociomaterial Practice with Broader Institutional Logics", *ICIS*, 2013.
- [36] Seidel, S., L. Chandra Kruse, N. Székely, M. Gau, and D. Stieger, "Design principles for sensemaking support systems in environmental sustainability transformations", *EJIS*, 27(2), 2018, pp. 221–247.
- [37] Song, Y., Y. Guo, and D. Thuente, "A quantitative case study on students' strategy for using authorized cheat-sheets", *IEEE Frontiers in Education*, 2016, pp. 1–9.
- [38] van der Sype, Y.S. and W. Maalej, "On lawful disclosure of personal user data: What should app developers do?", *RELAW*, 2014, pp. 25–34.
- [39] vom Brocke, J., R. Winter, A. Hevner, and A. Maedche, "Accumulation and evolution of design knowledge in design science research—A journey through time and space", *JAIS*, 2019.
- [40] Wania, C., "Exploring Design Patterns as Evaluation Tools in Human Computer Interaction Education", *MWAIS*, 2019.
- [41] Wheeldon, J. and J. Faubert, "Framing Experience: Concept Maps, Mind Maps, and Data Collection in Qualitative Research", *International Journal of Qualitative Methods*, 8(3), 2009, pp. 68–83.
- [42] Winkler, M., T. Huber, and J. Dibbern, "The Software Prototype as Digital Boundary Object: A Revelatory Longitudinal Innovation Case", 2014.
- [43] Yin, R.K., *Case study research and applications: Design and methods*, SAGE, Los Angeles, 2018.
- [44] Yskout, K., R. Scandariato, and W. Joosen, "Do Security Patterns Really Help Designers?", *International Conference on Software Engineering*, 2015, pp. 292–302.
- [45] Zenkert, J., A. Holland, and M. Fathi, "Discovering contextual knowledge with associated information in dimensional structured knowledge bases", *International Conference on Systems, Man, and Cybernetics*, 2016, pp. 923–928.