

Please quote as: Hoffmann, A.; Schulz, T.; Hoffmann, H.; Jandt, S.; Roßnagel, A. & Leimeister, J. (2012): Towards the Use of Software Requirement Patterns for Legal Requirements. In: 2nd International Requirements Engineering Efficiency Workshop (REEW) 2012 at REFSQ 2012, Essen, Germany.

Towards the Use of Software Requirement Patterns for Legal Requirements

Axel Hoffmann¹, Thomas Schulz², Holger Hoffmann¹, Silke Jandt², Alexander Roßnagel², and Jan Marco Leimeister¹

¹Information Systems, Kassel University, Germany

{axel.hoffmann, holger.hoffmann, leimeister}@uni-kassel.de

²Public Law particular Environmental Law and Technology Law, Kassel University, Germany
{t.schulz, s.jandt, a.rossnagel}@uni-kassel.de

Abstract. Laws and regulations play an increasingly important role for requirements engineering and system development. The challenge of interpreting the law to elicit legal requirements for a novel application calls for legal expertise. In this paper, we investigate if the effort of compiling a list of legal software requirements can be reduced by reusing recurring legal requirements. Therefore, we collected legal requirements that are stable concerning changes due to their origin in fundamental, higher-ranked laws, and derived software requirement patterns from them. This paper contributes by presenting those software requirement patterns consisting of the name, the goal and the pre-defined requirement template. We argue that under certain circumstances they can be used as a lightweight approach to specify legal requirements in system development projects and hence reduce the need for legal advice.

Keywords: Software Requirement Patterns, Requirements Reuse, Legal Requirements, Laws, Regulations

1 Introduction

The need for system developers to create systems compliant to legislature has been identified as a challenging and important problem in the requirements engineering (RE) community [1, 2]. This trend can be seen, for example, in the finance and healthcare domain, but is also getting more important in other domains of system development [3]. During the design of information systems in particular, one needs to consider: the EU Data Protection Directive, the basic rights to informational self-determination, confidentiality and integrity of information technology systems, the secrecy of telecommunications, as well as the data and consumer protection law. Infringement of any such laws and regulations can lead to high costs, e. g., in the form of compensations or penalties. These litigation-related costs are rising faster than the costs covering all the other aspects of software development; they even outgrow the cost for programming [4]. Only considering laws and regulations and complying with them enables legitimate information system development [5].

Developing legally compliant systems is very challenging. Laws and regulations contain numerous ambiguities, cross-references, and domain specific definitions. Furthermore, they are frequently amended via new regulations and judicature. Although the access to laws and regulations has become easier for system developers in the age of the Internet [2], the problem of the complexity of applying laws is not resolved. Even the identification of relevant laws, and especially the derivation of requirements for the technical system from laws, can hardly be accomplished without legal expertise. Despite the knowledge of specific legal terms and legal reference techniques [6], requirements analysts need to recognize the correlation between the different rules, as well as comprehend the statements of laws relating to technology. Thus, the challenge is already in the development process of interpreting the law and deriving system requirements from them.

Researchers are providing engineers with techniques and tools for specifying and managing software requirements for legally compliant systems [2]. However, these techniques are very laborious and require experience with laws and legal texts. Only a few requirements analysts have such legal expertise. Further, many system development projects cannot afford a comprehensive legal requirements analysis.

The purpose of our research is to help requirements analysts in specifying legal requirements (LRs). We thus compare the results of LRs specifications and derive software requirement patterns (SRPs) [7] that can be (re)used by requirements analysts in system specification. The LRs specifications we used as source material were derived by legal experts with KORA, a method used in German legal research. The acronym KORA stands for “**K**onkretisierung **r**echtlicher **A**nforderungen” (concretization of LRs) [8], and denotes a procedural method which allows the consideration of LRs in the design of information technology. This method has been evaluated several times in legal research [9-14] and derives requirements from the (stable) purpose of law, rather than handling detailed (changing) regulations. In our study, we have chosen the legal purpose of personal data protection and have derived six SRPs supporting it. These SRPs cannot replace LRs analysis in law-critical domains, but they can serve as a lightweight approach to consider legal purposes in RE.

The remainder of the paper is organized as follows. We first give an overview of the related work with LRs and SRPs. Next, we briefly describe KORA to show why the results are appropriate to create reusable SRPs. After a description of the research design in section 4, we present six SRPs for LRs in section 5. This is followed by the discussion and conclusion.

2 Related Work

2.1 Specifics of Legal Requirements

Laws are normative provisions that describe what is forbidden or allowed. The way in which laws are formulated differs fundamentally from the way in which requirements are specified [15]. As developers of technical systems usually have no legal training, specialists need to be incorporated into the development process to analyze

LRs of law [1]. In determining the LRs for a technology, there are the following basic challenges [1]:

- Choice of laws
- Extraction of relevant obligations and rights from laws
- Abstract and technology neutrality laws
- Dynamics of law

Due to the large number of laws, it is hard to assess which laws, along with their LRs, need to be considered for the development of a specific information system. Given today's global distribution of technical systems, laws of different countries can be relevant. Additionally, there is a prevailing legislative hierarchy: in Germany, for example, the constitution, laws at the federal and state level, and regulations. Thus, the developer is faced with a multitude of laws, some of which are parallel, but which may also occur secondarily [1, 2]. The legal analysis is complicated by the fact that not only the (written) laws, but the interpretation by the courts, as well as that in the literature, must also be taken into account [2]. These are harder to obtain than legislation, and can sometimes produce a more mixed picture, with decisions regarding specific cases taken independently. It is precisely challenging for this reason to identify relevant sources of LRs.

After the relevant LRs have been identified, the next challenge awaits. From the often very long laws or legal interpretations, relevant rights and obligations need to be extracted in order to provide LRs. It is common for technical systems that only a small part of existing legislation is relevant. In addition, dynamic and static references in the laws make the related interpretation more difficult [2, 16].

Laws set particular legal consequences for an unlimited number of individual cases, and must generally be formulated abstractly. This requires laws to be interpreted before they can be applied to specific cases. Further, laws often provide a margin of their interpretation, since names and phrases can be ambiguous [2]. In RE, this is referred to as a defect of natural language. In legal literature, interpretations can be found that do not meet the intent of the laws. Additionally, there are often varying legal views [2]. Laws usually address issues that have occurred in the past, such as problems that were caused by economic or social changes. For advanced information systems, the relevant specific details in laws are missing because economic or social changes have not yet taken place in practice, and legislature has not yet intervened [6]. Moreover, it is not possible for the legislature to adapt the laws at the same pace as that which technology development moves. This issue is largely met with abstract and technology-neutral regulations that target only specific risks. However, it is possible that regulations are missing for certain legal risks of new technology [1].

Laws are not necessarily time-consistent and changeless; rather, they are subject to continuous changes [2]. Especially laws and detailed regulations at the lower levels of hierarchy may change quickly, or are supplemented by additional regulations. Further, interpretation of laws by judgments is often necessary for the sake of legal security, but this is a very time consuming process [17]. Thus, for advanced information systems, concrete points of reference may possibly not yet be available. Compounding matters, the interpretation of laws can change over time [18], and the law dynamics

require technical system dynamics. These need to be adjusted throughout the life cycle with respect to changing or new laws [1]. This traceability of requirements raises a further problem area in RE. If the system has to be adapted, it needs to be documented which design decision is influenced by which (legal) requirement.

2.2 Prior Work on Legal Requirements

In software engineering, different efforts have been made to deal with LRs. An in-depth survey of work within the computer science and artificial intelligence domains in handling legal texts for system development has been carried out by Otto and Anton [2] to aid requirements analysts to better specify, monitor, and test information systems for compliance. This section provides a brief overview of approaches that help requirements analysts in acquisition and analysis of LRs.

Siena et al. [15] recommend the transition of LRs into stakeholder goals, and that they should be considered in goal-oriented RE. The described approach corresponds with the explanations of Ishikawa et al. [17], in which they stress the transition between legal goals and the stepwise refinement of technical goals. As laws are often very abstract and general, it is essential for a business organization to derive its own concrete measures to be taken. However, these legal regulations do not comprise the goals of RE; rather, they equal the concept definitions that require further refinement. As described by Ishikawa et al. [17], goal refinement and the refinement of concept definitions are related to each other. Guarda and Zannone [19] deal with LRs in a goal-oriented way, as they derive goals directly from law and consider them in the later requirements analysis. Problems arise when there are no laws or regulations that can be interpreted and used directly by requirements analysts.

Moreover, there are approaches that translate laws into abstract models [6]. It is therefore possible to formally examine an application in terms of legal conformity. However, this translation of requirements into abstract models requires an exact formulation that regulations often lack, as they are in many cases too general and non-technical [2]. Even if these regulations were to offer a sufficient level of accuracy, there would still be the complexity of translating abstract legal concepts into requirements [15]. Methods for the interpretation of these regulations are not sufficiently advanced, concentrating more on specific aspects [1]. Thus, only explicit guidelines allow applying requirements modeling to legal regulations in order to obtain requirements for the system. Abstract laws need to be concretized in advance.

Toval et al. [5] have set up a LR catalog regarding security and personal data protection which serves as a source of documents and interpretations for system development teams. The catalog enables requirements analysts to incorporate LRs into specifications, and thus build compliance into new systems. This approach, however, still faces the problem of dynamics in legislation and associated changes [2].

2.3 Requirements Reuse and Software Requirement Patterns

Reuse is an established practice in software engineering [20, 21]. In RE, reuse can help requirements analysts to elicit and document software requirements. SRPs are a

worthwhile approach to reuse requirements [22]. A pattern, in general, describes a problem which occurs over and over again, and then describes the core of the solution to that problem, in such a way that it can be used a million times over, without ever doing it the same way twice [23]. SRPs are used for the software analyses stage. There are different approaches that differ in scope, notation and application [22]. Recent approaches using SRPs for writing software requirement specifications can be found in the work of Withall [7] and in the Pattern-based Requirements Elicitation (PABRE) by Renault, Mendez-Bonilla, Franch, and Quer [24, 25].

A pattern based approach can reduce the effort of acquiring requirements for many development projects. The possible benefits for requirements analysts are not only the reduction of time spent to perform the elicitation of the requirements, but also the improvement of the quality of the requirements book obtained [25]. For this reason, the reusability of SRPs is the prerequisite for their applicability in practice.

Summarizing, the challenges with LRs analysis evident from: choice of laws, extraction of relevant obligations and rights from laws, abstract and technology neutrality laws, and dynamics of law demand specific knowledge and considerable effort in RE. We seize the suggestion of LR reuse [5] and implement it with SRPs [7, 24] to face the named challenges. With the use of LRs that are stable concerning changes in detailed regulations due to their origin in fundamental, higher-ranked laws, we reduce flaws existing in prior LR reuse. In order to generate SRPs, we use specifications containing LRs homogeneously created with the KORA-Method.

3 KORA – Concretization of Legal Requirements

KORA is a method that has been used in German legal research to derive LRs for technical systems for nearly 20 years [8-14]. KORA is performed by legal experts and is not meant to be performed by requirements analysts. Nevertheless, we used the results of various applications of KORA to identify SRPs for LRs. For ease of understanding, we briefly describe the basics of KORA (with the specific terminology) in the following section.

3.1 Deriving Legal Requirements from Higher-Ranked Laws

For the consideration of legality of systems in computer science, the concept of IT compliance has been established. To this end, laws are analyzed for containing direct or indirect LRs - a step which must be considered in the design of technology. Examples are the Digital Signature Act and the Data Protection Act. From them, legally binding technical requirements can be obtained directly, as failure of implementation could result in legal consequences. For this circumstance, the understanding of laws and other LRs as constraints has emerged.

The minimum requirements for a socially responsible technology design can be found in the law. These serve both the constitutionally guaranteed free democratic basic order of the state and the protection of fundamental rights of individual citizens. Some laws, such as the data protection legislation, contain explicit guidelines for the

design of data processing information systems. In addition, there are design requirements in other laws that regulate only indirect information technology, such as in accordance with § 312g of the German Civil Code (BGB), regarding entrepreneurs fulfilling legal duties in the electronic exchange. Therefore, KORA obtains technical requirements from the purpose of legislation [8, 9]. This is called being legally compatible. For the purpose of the secrecy of telecommunications, e. g., a communications technology where communication is encrypted automatically is more legally compatible than one that is not automatically encrypted; albeit, the unencrypted technology is not in any case unlawful. Further, by permanently validating laws and their purposes, it is not necessary to adapt the systems as a result of legislative changes. At the time of development, loopholes in detailed rules are also irrelevant [26].

In the development of technical systems - similar to the task of a judge in determining the facts of the case - developers have to derive specific technical requirements from the legal provisions. However, this task has to be carried out before there is a finished information system. With KORA, the legal concretization is achieved through a four-step process (Fig. 1) [12].

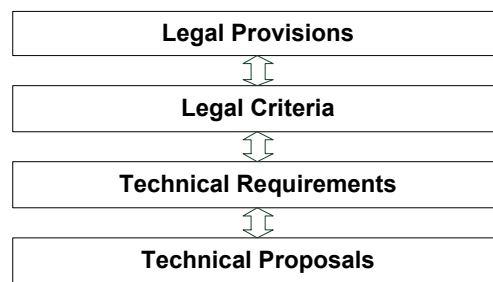


Fig. 1. Levels of KORA

3.2 Application of KORA in Legal Research

KORA starts from existing constitutional and other legal norms, which can be specific legal rules. If there are no specific legal provisions applicable to the planned information system, or if they are subject to short-term changes, KORA starts from steady higher-ranked legal rules, such as can be found, for example, in the constitution [9, 12]. On the basis of the purpose and the knowledge of social chances and risks inherent in the information systems, legal provisions for the planned information system are developed from the constitutional and other legal norms on the first level. Hence, the legal provisions apply to the specific project. By focusing on higher-ranked legal rules, the number of laws to be examined is narrowed down, which simplifies the selection of relevant laws [26]. Furthermore, the differences between the laws to be considered in different jurisdictions are far greater on the lower-ranking level. If an information system is used worldwide, it must indispensably be aligned with general provisions.

Legal criteria are identified by analyzing how the legal provisions that have been developed on the first level can be qualitatively assessed with regard to the information system [9]. The criteria describe rather abstract solutions to fulfill the legal provisions which are in principle legal and non-technical, but certainly can be technical. Legal criteria can also be developed on the basis of the reasoning given by judges in legal cases in which the same legal norms are applied [9]. Sometimes the criteria can already be incorporated as design demands in detailed legislature.

Technical requirements for the design of the technology are abstractions of specific characteristics of the technology. As the objective of KORA is not only a lawful but also a legally compatible design of information systems, the technical requirements for design are requirements which can enhance the legal compatibility of information systems. A high degree of legal compatibility ensures sustainable lawfulness and lawfulness in different jurisdictions [26]. If they are adopted in the system development, there will still remain considerable scope for the implementation by designers. For complex systems, further technical concretization should take place afterwards.

On the last level of KORA, technical proposals for the design of the technology are developed on the basis of the technical requirements [9]. Technical proposals for the design are performance characteristics which constitute technical functions. For a new information system, technical characteristics are developed from the technical requirements for the design.

3.3 KORA-Results as Foundation of Software Requirement Patterns

We argue that the results of KORA are suitable to be used as source of SRPs. Due to their origin in general and stable legal rules, they are most suited for requirement reuse because it ensures a long life period of the SRPs. Since these rules are often recognized internationally, the patterns can often also be used for systems that are intended for an international market. Thereby one should orientate by the strictest rules if possible. By using German law for privacy purposes, there is a high likelihood that the information system is legally compliant with other jurisdictions. Further, due to the focus on legally compatible systems rather than just archiving the minimum standards of law, it ensures legal compliance even when detailed laws are tightened.

KORA results in requirements on different levels of abstraction. Technical proposals (level 4) provide design recommendations for the technical system. Nevertheless, for SRPs, we need solution-free requirements [27] that can be found in the technical requirements (level 3), which are related to the basic functionality of the information system. We extract the technical requirements from the LR specifications and use them as source requirements to generate SRPs.

4 Research Design

Results from LR specifications, all which were archived with KORA, served as our source material. Some of the documents were available in public [11-14], while others were not designated for public use, but were provided for our research.

Given the documents, we followed the systematic approach of Withall [7] to find candidates for SRPs, and scanned a sample of seven LR documents. These documents contained about 30 to 50 LRs. We listed all requirements in a spreadsheet. If a requirement was similar to one we already had on the list, we noted that and moved on. In the end, we filtered the list for all requirements that were mentioned in more than one specification. For the identified recurring requirements we formulated SRPs.

5 Results

With today's technology, it is especially the protection of personal data that often plays an outstanding role. Accordingly, this paper focuses on SRPs which are particularly relevant for the protection of personal data. These patterns are not exhaustive, and should serve only as examples to illustrate reusable SRPs.

We have selected natural language to formulate the SRPs. Non-technical experts, such as legal practitioners, prefer natural language requirements for reading, analysis and discussion [5]. However, the software requirements specifications we used as a source were also written in natural language. This is in line with recent approaches using SRPs for writing software requirements specifications [7, 24].

To illustrate the pattern for LRs, we use the following attributes that are components of the recommended structure of a SRP in [22]:

- Goal: The goal has the role of the problem part of a pattern. It has an important role since it will help to decide whether the pattern is applicable to the software [25]. This is determined by the planned functionality of the software.
- (Fixed Part) Template: The fixed part template is the core of the solution, stating that the software has to achieve the goal of the SRP, but not indicating how this goal can be achieved. Since the fixed part of a form is abstract, it is possible to provide some extra-information or constraints in the extension part about how to achieve the goal of the SRP [25].
- Sources: The sources usually comprise the source documents. For our purposes, we provide the legal provisions from which the requirements were derived and cite LRs specifications in which the derivation is described.

The example patterns are ascribed to the informational self-determination [11-14]. The right to informational self-determination is a special manifestation of the right of development and protection of one's personality, which is established in Art. 2 (1) read in conjunction with Art. 1 (1) of the German Constitution. This right was acknowledged in 1983 by the German Federal Constitutional Court [28]. By the right to informational self-determination, the individual is protected from unlimited dealing with personal data. Individuals need to decide for themselves when, and within which limits, personal life issues should be revealed. Today, the right to informational self-determination has a big impact, especially on the data protection acts.

The following are a few examples of SRPs that are particularly relevant for the protection of personal data (Table 1).

Table 1. Software Requirement Patterns

1	Confidentiality of the Communication Channels	
	Goal	Ensure protection and confidentiality of personal data during transmission.
	Template	The system shall prevent spying out personal data by unauthorized third parties during transmission.
	Source	[28]; derived in, e.g., [11-14].
2	Divide of Different Personal Data	
	Goal	Limit the usage of personal data to the dedicated purpose.
	Template	The system shall divide personal data according to different purposes and coherences of use.
	Source	[28]; derived in, e.g., [11, 12].
3	Control about Storage Medium	
	Goal	Ensure protection and confidentiality of personal data during storage.
	Template	The system shall store personal data on a storage medium that is exclusively controlled by the user.
	Source	[28]; derived in, e.g., [11-13].
4	Access Control	
	Goal	Ensure protection and confidentiality of personal data during storage.
	Template	The system shall ensure that only authorized users gain access to the service.
	Source	[28]; derived in, e.g., [11-13].
5	Limitation of Storage Time	
	Goal	Limit the usage of personal data to the dedicated purpose.
	Template	The system shall delete personal data if they are no longer necessary for system operations.
	Source	[28]; derived in, e.g., [11, 12].
6	Documentation of Processing with Personal Data	
	Goal	Ensure transparency of personal data usage.
	Template	The system shall record processing with personal data.
	Source	[28]; derived in, e.g. [12].

6 Discussion

There are many pitfalls when formulating legal SRPs in order to ensure the applicability of the result in more than just one system development project; fortunately, there are also some advantages. Legal SRPs satisfy the need of requirements analysts in three situations. First, they can help if no detailed laws or regulations are applicable. Second, they are very useful if the requirements analysts do not have any exper-

tise to work with laws and regulations. Third, they are essential if there are too few resources to conduct a comprehensive LRs analysis.

To reduce the disadvantages of LRs reuse, we considered specifics in LRs engineering. Usually, a LR catalog requires updates each time the law changes. This is also true for legal SRPs. But with the selection of requirements worked out with the KORA-Method, we take advantage of the specifics. The KORA-Method that derives the requirements from general and stable legal rules ensures a long life period of the SRPs even without permanent updates. Further, due to the focus on legally compatible systems rather than just archiving the minimum standards of law, it ensures in all likelihood legal compliance even when detailed laws are tightened.

The traceability between the derived requirements and the sources in law are ensured by specifying the legal sources mentioned in the analyzed source specifications. Further, the full trace from the LRs to the legal sources can be found, if necessary, in the KORA specifications.

When a pattern is to be used, it first has to be examined whether this pattern is relevant for the design of the information system at all [24]. If, for example, a system does not gather, process or utilize personal data, a pattern which only purposes the protection of such data must not be adopted. After identifying all relevant SRPs, the requirements analyst can assemble the requirements document [24].

The effort for selecting and adapting SRPs is much less than a full requirements analysis. According to the domain, while the search, extraction and translation of regulations into requirements took up to several weeks, the selection and adaption of SRPs can be done in four to five man-days [25].

With LRs, there is always the problem that the legislature can change them at any time. For patterns which are deduced from LRs, this means that they can be deprived of their legal basis. This is especially a problem in dealing with relatively detailed laws, since these can change frequently. Fundamental legal provisions, however, remain very stable. Patterns are therefore more stable when they are deduced from more stable law. For the use of patterns in practice, their stability is very important. For this reason, we developed patterns which can be ascribed to fundamental, higher-ranked laws.

For use in practice, it is also important that the patterns are reusable. Only in this way is the considerable effort to create patterns worth. To ensure the reusability, we developed patterns by means of technical requirements derived in different projects for different systems. A further challenge in the development of such patterns is that they implement legal provisions, but should be used by engineers. This assumes that the patterns are formulated in a language that can be understood by engineers. For this reason, our patterns were formulated in technical language. It could thus be ensured that there were no misunderstandings with the use of patterns due to linguistic differences between the legal and technical languages.

Not the least of the challenges, the patterns must also be legally correct. Patterns which should implement legal provision should be evaluated with the cooperation of jurists; accordingly, jurists were involved in the design and evaluation of each pattern. Thus, the derived SRPs are conform to today's detailed laws.

7 Conclusion

The fulfillment of LRs cannot be reached by supplementing individual software components or modules to a system, as they affect the whole software, compared to a cross-cutting concern of aspect-oriented programming [17]. LRs resulting from the laws must therefore be considered in the early phases of RE in order that the legally compliant technology design can be ensured at early stages of development [15]. Early consideration of LRs does not take place in most current development projects. For example, requirements of informational self-determination which have already been established by comprehensive data protection legislation are as important as functional requirements when designing information systems; however, they are not elicited, analyzed and taken systematically into account during implementation [19].

SRPs offer a solution for requirements analysts to factor LRs directly into the information system design. These patterns are generalizable, which leads to reusability. We created the patterns from LRs that were deduced from stable higher-ranked laws, resulting in the development of stable patterns. Additionally, we formulated the patterns in a technical language to guarantee that even requirements analysts without a legal background could work with them. With our patterns, requirements analysts have a lightweight approach to incorporate LRs into system specifications. It can improve the productivity of requirements analysts, as they can start from a set of pre-defined SRPs in a technical language. The quality of the specification can also be enhanced because the SRPs are evaluated by legal experts.

Our future plan is to integrate the requirement patterns within a SRP catalog. Further, we want to parameterize some parts to allow more detailed choices by each analyst applying the pattern and make it easier to adapt the patterns.

8 References

1. Kiyavitskaya, N., Krausova, A., Zannone, N.: Why Eliciting and Managing Legal Requirements Is Hard. In: Requirements Engineering and Law, pp. 26-30. (2008)
2. Otto, P.N., Anton, A.I.: Addressing Legal Requirements in Requirements Engineering. In: 15th IEEE International Requirements Engineering Conference, pp. 5-14. (2007)
3. Hoffmann, A., Söllner, M., Fehr, A., Hoffmann, H., Leimeister, J.M.: Towards an Approach for Developing socio-technical Ubiquitous Computing Applications. In: Sozio-technisches Systemdesign im Zeitalter des Ubiquitous Computing, Berlin (2011)
4. Cosgrove, J.: Software engineering and the law. *Software*, IEEE 18, 14-16 (2001)
5. Toval, A., Olmos, A., Piattini, M.: Legal requirements reuse: a critical success factor for requirements quality and personal data protection. In: 10th IEEE International Requirements Engineering Conference, pp. 95-103. (2002)
6. Breaux, T.D., Anton, A.I., Boucher, K., Dorfman, M.: Legal Requirements, Compliance and Practice: An Industry Case Study in Accessibility. In: 16th IEEE International Requirements Engineering Conference, pp. 43-52. (2008)
7. Withall, S.: *Software requirements patterns*. Barnes & Noble (2008)
8. Hammer, V., Pordesch, U., Roßnagel, A.: KORA—Eine Methode zur Konkretisierung rechtlicher Anforderungen zu technischen Gestaltungsvorschlägen für Informations- und Kommunikationssysteme. *Infotech/I+ G* 21–24 (1993)

9. Hammer, V., Pordesch, U., Roßnagel, A.: Betriebliche Telefon- und ISDN-Anlagen rechtsgemäß gestaltet. Springer, Berlin (1993)
10. Jandt, S.: Vertrauen im Mobile Commerce – Vorschläge für die rechtsverträgliche Gestaltung von Location Based Services. Baden-Baden (2008)
11. Gitter, R.: Softwareagenten im elektronischen Geschäftsverkehr – Rechtliche Vorgaben und Gestaltungsvorschläge, Baden-Baden (2007)
12. Steidle, R.: Multimedia-Assistenten im Betrieb – Datenschutzrechtliche Anforderungen, rechtliche Regelungs- und technische Gestaltungsvorschläge für mobile Agentensysteme, Wiesbaden (2005)
13. Ranke, J.S.: M-Commerce und seine rechtsadäquate Gestaltung – Vorschläge für vertrauenswürdige mobile Kommunikationsnetze und -dienste, Baden-Baden (2004)
14. Idecke-Lux, S.: Der Einsatz von multimedialen Dokumenten bei der Genehmigung von neuen Anlagen nach dem Bundesimmissionsschutz-Gesetz, Baden-Baden (2000)
15. Siena, A., Mylopoulos, J., Perini, A., Susi, A.: From Laws to Requirements. In: Requirements Engineering and Law, pp. 6-10. (2008)
16. Maxwell, J.C., Antón, A.I., Swire, P.: A Legal Cross-References Taxonomy for Identifying Conflicting Software Requirements. In: 19th IEEE International Requirement Engineering Conference. (2011)
17. Ishikawa, F., Inoue, R., Honiden, S.: Modeling, Analyzing and Weaving Legal Interpretations in Goal-Oriented Requirements Engineering. In: Requirements Engineering and Law, pp. 39-44. (2009)
18. Massey, A.K., Otto, P.N., Anton, A.I.: Prioritizing Legal Requirements. In: Requirements Engineering and Law, pp. 27-32. (2009)
19. Guarda, P., Zannone, N.: Towards the development of privacy-aware systems. Information and Software Technology 51, 337-350 (2009)
20. Berkovich, M., Esch, S., Leimeister, J.M., Krcmar, H.: Requirements engineering for hybrid products as bundles of hardware, software and service elements – a literature review. 9. Internationale Tagung Wirtschaftsinformatik (WI 2009), Wien, Österreich (2009)
21. Berkovich, M., Leimeister, J., Krcmar, H.: Requirements Engineering for Product Service Systems. Business & Information Systems Engineering 3, 369-380 (2011)
22. Franch, X., Palomares, C., Quer, C., Renault, S., De Lazzer, F.: A Metamodel for Software Requirement Patterns. Requirements Engineering: Foundation for Software Quality 85-90 (2010)
23. Alexander, C.: The timeless way of building. Oxford University Press, USA (1979)
24. Renault, S., Mendez-Bonilla, O., Franch, X., Quer, C.: PABRE: Pattern-based Requirements Elicitation. In: Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on, pp. 81-92. (2009)
25. Renault, S., Mendez-Bonilla, O., Franch, X., Quer, C.: A Pattern-based Method for building Requirements Documents in Call-for-tender Processes. International Journal of Computer Science and Applications 6, 175 - 202 (2009)
26. Hoffmann, A., Jandt, S., Hoffmann, H., Leimeister, J.M.: Integration rechtlicher Anforderungen an soziotechnische Systeme in frühe Phasen der Systementwicklung. In: Mobile und ubiquitäre Informationssysteme, Kaiserslautern (2011)
27. Firesmith, D.G.: Engineering security requirements. Journal of Object Technology 2, 53-68 (2003)
28. BVerfGE (anthology of the judicial decisions of the German Federal Constitutional Court): 65, 1 - court ruling of the 15th December, 1983 - 1 BvR 209/83 et al., (1983)