Requirements Engineering for Product Service Systems

A State of the Art Analysis

Marina Berkovich, Jan Marco Leimeister, Helmut Krcmar

Business & Information Systems Engineering (2011) 3 (6)

Appendix (available online via http://springerlink.com)

Appendix 1 – Research Methodology

This section completes Sect. 2, by explaining the research-methodical proceeding in detail.

1.1 Definition of the research question

Initially, the research questions were defined that needed to be answered in the literature analysis. The central question is: "To what extent are the selected domain-specific approaches of requirements engineering suited for PSS, and which of these can be used in the development process of PSS?"

1.2 Definition and validation of the analysis protocol

The analysis protocol comprised the criteria for evaluating the literature. In the literature analysis the criteria facilitated the answering of the research questions by describing the relevant characteristics for the suitability of an RE approach for PSS in detail, and by providing a basis for evaluating the adequacy of the approaches.

For the derivation of the criteria, a framework was developed, the aim of which was to describe RE concepts in a general way so that older and future research could be structured (Goeken and Patas 2010). The framework (Sect. 5.1) connected the life cycle and characteristics of PSS with the main tasks of RE. Based on the gained results, the criteria for the literature analysis were derived (Sect. 5.2).

In order to validate the analysis protocol, a test run with the "procedure model of Munich" (Lindemann 2006) known from earlier research (see Berkovich et al. 2009a) was performed. The criteria were adjusted accordingly, by focusing on the PSS-specific aspects (e.g., consideration of the requirements and their affiliation to different domains in the search for conflicts). Subsequently, the analysis protocol was reviewed for correctness and completeness.

1.3 Identification and selection of relevant scientific sources

In this step, all sources relevant for the literature analysis were identified. Due to the different understanding of RE in product, software, and service engineering, the literature search was performed for each domain.

Literature search. For searching the relevant literature, the portal "Google Scholar" was chosen, covering a major part of the journals and electronic publications in computer science as well as in engineering. Meier and Conkling (2008) have shown that 90% of the publications in engineering, published after 1990, are announced in "Google Scholar." In the field of computer science, important publishing companies (e.g., Elsevier) and non-profit organizations (e.g., ACM or IEEE) are indexed in Google Scholar (Meier and Conkling 2008; Jacsó 2008). The keywords that were used are introduced in the following paragraph.

- In product engineering, we searched for “product engineering” and “product development.” Then, the keywords “requirements engineering” and “management” were added in order to receive specific results of RE.
- Since in software engineering RE is regarded as a discipline, the sole keywords for this domain were “requirements engineering,” “requirements management,” and “Anforderungsmanagement.”
- The results of the search in service engineering, completed with the terms “service engineering” and “service development,” were subsequently combined with “requirements engineering” and “requirements management”.
- In addition to individual concepts, other approaches covering the issue of an integrated development of PSS were also considered. To get relevant literature in this area, the search terms “hybrid products” and “product service system” were entered in “Google Scholar”.

Selection of relevant literature. All hits of the search portal were sorted by the frequency of their citations. For each keyword, the 100 most cited articles were examined and assigned to one of the following categories.

(a) Articles that constituted a generic approach/procedure model:

The generality and spread of the respective article comprised the selection criteria for the first category (see Goeken and Patas 2010). Further, only those sources were considered that made a major contribution to RE, that is, they dedicated at least one subject area (e.g., a book chapter or a section of an extensive journal article) to RE. By selecting generic approaches, a superordinate structure of knowledge and a meta-level were established for the classification of detailed approaches appearing in conferences and journals (Goeken and Patas 2010). The assignment of the specific publications to generic approaches (mostly in the form of books) thus gave an overview of the RE in product, software, and service engineering.

For each domain, five generic approaches were presented that were based on expert opinions as well as on appraisals of the authors. In order to provide a high degree of validity, the selection of the articles was coordinated by all authors. Since the generic approaches showed certain similarities, it was sufficient to analyze five of them for each domain (see Table 1).

(b) Articles that implied a specific topic of RE:

This category included journals and conference articles, addressing specific themes of RE, such as activities and techniques also proposed in the generic approaches. Due to the resulting large number, a detailed presentation could not be included in this paper'. However, some articles are cited in section 6.1 and section 10 in the description of the RE approaches.

The publications found in the conferences and journals were assigned to the different domains: 24 referred to product engineering, 47 to software engineering, 14 to service engineering, and 14 to an integrated development of PSS.

(c) Articles that did not consider RE:

Papers that did not primarily deal with the theme of RE were depicted in this category. Due to the full text search of the portal, articles that solely mentioned RE in the text (e.g., in the introduction) were used.

1.4 Data collection and analysis

The data needed for answering the research questions were extracted and analyzed according to the criteria defined in the protocol. In order to ensure the quality of the literature analysis, the following questions obtained from Kitchenham et al. (2009) were answered: Do the sources cover all relevant aspects of the research questions Is the quality of the analyzed articles sufficient? Is the information required for the literature analysis described adequately in the article?

Appendix 2 – Description of approaches in product, software, and service engineering as well as of integrated approaches for the development of PSS

To gain a better understanding of the subsequent analysis, the selected RE approaches were examined on the basis of the criteria defined in Sect. 5.2 and divided into the different tasks of RE.

2.1 Requirements engineering in product engineering

According to Ehrlenspiel (2002, p. 663), a requirement is qualitative and/or quantitative information about the attributes of, or conditions for, a product.

Requirements Elicitation. As a first step in the development process, the approaches of Ulrich and Eppinger (2003), Ehrlenspiel (2002), VDI-Richtlinie 2221 (1993), as well as Pahl et al. (2006) analyze the future development environment in order to identify possible influencing factors in order to establish the overall objective of the development. On this basis, all stakeholders and their requirements to the solution are determined. Subsequently, the resulting requirements have to be estimated (e.g., according to their relevance for the customer) and concretized. In his case study, Hobday (1998) points out that during the development of PSS, requirements elicitation is especially challenging, as customers cannot consciously express their wishes regarding the product.

Requirements Concretization and documentation. Although requirements concretization is one of the most important tasks of RE, it is not described explicitly and consistently in the analyzed approaches. Since the requirements of the stakeholders are often qualitative and imprecise (Tseng and Jiao 1997), they must be translated into the "language of the developers" by receiving detailed and quantitative information used for product development (Ahrens 2000; Pahl et al. 2006). To connect customer and design requirements (Chahadi et al. 2007), the method of "Quality Function Deployment" (QFD) is frequently quoted (Hauser and Clausing 1988). Its main goal is to evaluate the different possible solutions according to the satisfaction of customer requirements, but not to focus on the systematic derivation of the design requirements from the customer requirements (Ahrens 2000; Pahl et al. 2006). For this reason, the initial and concretized requirements should be listed in a document and adapted to changes. In practice, large and unstructured documents, containing requirements to different subsystems and dependent systems represent a serious problem (Almefelt et al. 2006).

Identification and resolution of conflicts between the requirements. If the requirements are not consistent, conflicts arise (Almefelt et al. 2006) that should be identified and resolved as soon as possible (Ehrlenspiel 2002; Ulrich and Eppinger 2003).

Requirements traceability and change management. Changes in one requirement can lead to further changes in other requirements and in the corresponding components of the product

(Peterson et al. 2007). For this reason, it is necessary to record their interdependencies by using, e.g., a DSM-matrix (Danilovic and Sandkull 2005) that allows a fast evaluation of impacts of changes. Moreover, the specification has to be updated in case of any changes during the entire development process to facilitate requirements traceability. Jiao and Chen (2006) conclude their literature analysis with the statement that a lot of researchers have lately investigated this topic. In practice, however, continuous requirements engineering is still a great challenge.

Requirements validation. Modeling and simulation become more and more important in the development process (Schäppi et al. 2005). The simulation enables early statements of the product features and their fulfillment. Additionally, new knowledge is gained by comparing the different alternatives (Pahl et al. 2006). Consequently, gaps in the requirements can be elicited or impacts of certain requirements can be illustrated for stakeholders.

In product engineering, *modularization* is presented in various forms. Its basic principle is to define modules and their interfaces so that they can be reused for different products (Ehrlenspiel 2002; Schäppi et al. 2005). Thus, new products are based on existing guidelines, such as standard designs, reference architectures or product families that have to be discovered in requirements elicitation. Schäppi et al. (2005, p. 328) propose a method for describing products of a product family. The product family proposes a set of variable characteristics that can be combined individually for each product. At this point, the connection to the RE is visible, but the authors only indicate it.

Many of the analyzed approaches comprise classic procedures following the waterfall model such as, e.g., Pahl et al. (2006). However in practice, iterative procedure models are frequently used. Peterson et al. (2007) purport that iterative procedure models are indispensable, as the requirements in innovative projects change permanently. Nevertheless, the waterfall models depict typical phases in product development that imply the same tasks, regardless of the concrete process design.

All activities of RE are passed iteratively and should be integrated into the development process. For further information about RE approaches in product engineering, see Berkovich et al. (2009c).

2.2 Requirements engineering in software engineering

In software development, requirements engineering constitute a separate discipline. It is described as a process of defining the relevant requirements, by identifying all stakeholders and their needs, and by documenting the requirements in form of a specification that can be used for communication, further analyses, and implementation (Sommerville 2004, pp. 143-144). In this regard, requirements management can be seen as a sub-area of RE (Pohl 2007, p. 493).

According to the analyzed approaches, RE includes requirements elicitation, requirements prioritization, requirements concretization, requirements documentation, requirements validation, requirements negotiation, change management, and requirements traceability. These tasks are seen as essential elements of RE in the framework of Pohl (1993, pp. 39 ff).

Requirements elicitation. During requirements elicitation, the requirements and their sources (Lamsweerde 2009, p. 62; Aurum and Wohlin 2005), such as customers, stakeholders, existing documents, laws and competitors, are identified (Robertson and Robertson 2007, pp. 45-54). By communicating intensively and targeted at stakeholders, a better understanding of the requirements is achieved (Coughlan and Macredie 2002; Al-Rawas and Easterbrook 1996).

Requirements concretization. In this step, a bridge between the initial stakeholder requirements and the detailed design requirements is created (Kotonya and Sommerville 1998, pp. 146-149). By checking the consistency of the requirements, a full description comprising qualitative and quantitative aspects is written afterwards (Hull et al. 2004). According to Jarke (1998), various models serve as a conceptual foundation for requirements concretization. In this context, the identification of the relations between the requirements and their interdependencies should be mapped (Nuseibeh and Easterbrook 2000). Thus, it is important to know the priorities of the requirements influenced by the customer needs, the costs, and the risk level. It is then possible to determinate the importance of the requirements in the implementation phase (Hull et al. 2004, p. 81), as well as to estimate the customer satisfaction (Berander and Andrews 2005).

Requirements documentation. In requirements documentation, essential information about the requirements (e.g., their description, changes in requirements or responsibilities) should be specified (Pohl 2007, pp. 547-549). The most common form of documentation is the natural language, which has the disadvantage of being ambiguous. According to IEEE (1998), the specification should be: correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable, and traceable.

Identification and resolution of conflicts between the requirements. The aim of this activity is to discover and resolve all conflicts between the requirements by finding some compromise (Lamsweerde 2009, p. 14; Cheng and Atlee 2007).

Requirements validation. In this phase, the requirements are checked for inconsistencies, ambiguity and falsity (Jönnson and Lindvall 2005; Kotonya and Sommerville 1998, pp. 87-90). Furthermore, the design requirements are validated for their compliance with the initial requirements by involving all stakeholders (Al-rawas and Easterbrook 1996). In this connection, prototypes are frequently used to simulate and demonstrate different aspects of the developing system (Lamsweerde 2009, pp. 70-72). Prototypes show the effects of the requirements and thus contribute to the extraction of new stakeholder requirements as well as to the validation of requirements, particularly with regard to their fulfillment concerning the expectations of the stakeholders (Pohl 2007, pp. 457-462).

Requirements traceability and change management. The task of change management is to identify changes in the requirements during the entire product life cycle (Cox et al. 2009), to check them for their feasibility by determining their costs and impacts on other requirements, as well as to modify the specification (Kotonya and Sommerville 1998, pp. 143-146). A major precondition for this constitutes requirements traceability that ensures the mapping of all relationships between the requirements, from the initial requirements to the final system properties (Kotonya and Sommerville 1998, p. 160). Therefore, it supports the detection of existing conflicts between requirements (Lucia et al. 2007).

In software engineering, the reuse of modules is known as software product lines. On the basis of a common platform, various domain-specific applications can be created (Käkölä and López 2006). In their scientific work, Käkölä and López (2006, pp. 130 ff) provide an overview of the methods relating to requirements engineering for product lines. Additionally, they propose their own approach that is rested on an artifact model of RE specification results, which gives a description and definition of the steps in the process of RE.

In addition to the RE tasks presented in this section, various procedure models are available that refer to certain problems, indicating in what order existing RE steps should be applied, and what particular aspects should be considered.

2.3 Requirements engineering in service engineering

Services are defined as intangible assets produced for satisfying customer needs by using external production factors (Frietzsche and Maleri 2003, p. 197). The objective of service engineering is to enable a systematic development and design of services by providing various methods, procedure models, and tools (Bullinger and Schreiner 2003). In the literature, several approaches of this discipline, also known as “service development,” can be found.

Requirements elicitation. Initially, essential information about service ideas, key clients, and sources of requirements, namely, the customer and the supplier, is identified (Bullinger and Schreiner 2003). For product-related services, the primary product is determined and extended by different services (Husen 2007). Afterwards, the goals, chances and risks of the developing services should be recognized (Husen 2007; Frietzsche and Maleri 2003). Further, the customer requirements on the services are elicited with the help of interviews, observations, market analyses, and investigations (Husen 2007; Ramaswamy 1996), and are differentiated between wishes and needs (Edvardsson and Olsson 1996).

Requirements concretization and documentation. The initial requirements are prioritized (Ramaswamy 1996) and checked for interdependencies (Husen 2007). Based on the result, the requirements are concretized by assigning them to quantifiable attributes related to the implementation of the services (Husen 2007; Ramaswamy 1996). For this process, the method of QFD is frequently applied (Husen 2007). By classifying the concretized requirements according to the three dimensions of development (potential, process, and result), the result of service provision, as well as the necessary activities and resources, can be identified (Frietzsche and Maleri 2003; Bullinger and Schreiner 2003). This classification underlies the constructional characteristics of services that are defined as the achievement potential of the provider, the immateriality of the service, and the customer integration into the development process. Having completed the task of concretization, the requirements are documented afterwards (Bullinger and Schreiner 2003).

The approaches of service engineering do not focus on the identification and resolution of conflicts between the requirements, change management, and requirements traceability. They mention these activities only briefly and refer to the procedures in software engineering. For further information about RE approaches in service engineering, see Berkovich et al. (2009a).

In service development, the benefits of modularization are recognized. The reuse of undifferentiated service components leads to an increase in profitability (Böhmann et al. 2008). Böhmann and Krcmar (2003) propose an approach to develop modular services, which also includes the RE phases, but only superficially.

2.4 Requirements engineering of integrated approaches for PSS development

The theme of RE with regard to an integrated development of PSS is abstractly discussed in several approaches without going into detail on the activities of RE. Therefore, the following section is not structured according to the different RE tasks.

In the development process of PSS, the hardware, software, and service components are produced in parallel and coordinated by the involved disciplines (Korell and Ganz 2000; Spath and Demuß 2003).

Aurich et al. (2007) propose a procedure model for the life cycle management of PSS. While in its first phase (organizational structures) the organizational conditions are created in order to enable an integrated development, and the requirements are elicited afterwards (PSS planning). Based on that, the ideas for the PSS are generated, analyzed, categorized, and finally realized (PSS implementation). Another model considering the entire lifetime of PSS is described in Lindahl et al. (2007). As an essential step in the development process, the task of RE is to identify the customer needs in terms of goods and services. However, concrete techniques for its realization are not mentioned.

The “hyprid product development” of Spath and Demuß (2003) focuses mainly on the single elements of the whole solution. Initially, the requirements are elicited, analyzed, and used for the derivation of the product structures regarding material and immaterial components. Following this, the product structures are assigned to certain functions and principles that are subsequently implemented by the relating domain. During the entire development, the requirements model has to be updated.

As a basis for the development process, the “framework for the development of PSS” of Botta (2007) considers the requirements with respect to the required product properties. To describe the PSS accurately, its feature structures are derived.

Another concept is delineated in the development process of Thomas et al. (2008) that underlies the methods of product engineering. The first step includes the definition of the requirements to the PSS in order to determine its desired features and characteristics by using QFD. The article concludes that the methods of product engineering are only slightly suited for PSS.

The approach of Böhmann et al. (2008) to the modularization of standardized solutions is particularly tailored to IT services, consisting of hardware, software, or classic services. The requirements to the final product are classified based on four different views. Following this, the standardized partial solutions are identified.

References

Al-Rawas A, Easterbrook S (1996) Communication problems in requirements engineering: a field study. In: First Westminster conference on professional awareness in software engineering. London

Almefelt L, Berglund F, Nilsson P, Malmqvist J (2006) Requirements management in practice: findings from an empirical study in the automotive industry. Research in Engineering Design 17(3):113-134

Berander P, Andrews A (2005) Requirements prioritization. In: Aurum A, Wohlin C (eds) Engineering and managing software requirements. Springer, Berlin, pp 69-94

Chahadi Y, Wäldele M, Birkhofer H (2007) QFD – a link between customer requirements and product properties. In: International conference on engineering design, ICED'07. Paris

Hobday M (1998) Product complexity, innovation and industrial organisation. Research Policy 26(6):689-710

Hauser JR, Clausing D (1988) The House of quality. Harvard Business Review 66(3):63-73

IEEE (1998) IEEE recommended practice for software requirements specifications

Jarke M (1998) Anforderungsmodellierung: Können wir die Brücke zwischen Anwendung und Entwicklung stabilisieren? In: Modellierung 98 GI

Jiao JR, Chen C-H (2006) Customer requirement management in product development: a review of research issues. Concurrent Engineering 14(3):173-185

Korell M, Ganz W (2000) Design hybrider Produkte. Der Weg vom Produkthersteller zum Problemlöser. In: Bullinger H-J, Hermann S (eds) Wettbewerbsfaktor Kreativität: Strategien, Konzepte und Werkzeuge zur Steigerung der Dienstleistungsperformance. Gabler, Wiesbaden, pp 153-159

Lucia AD, Fasano F, Oliveto R, Tortora G (2007) Recovering traceability links in software artifact management systems using information retrieval methods. ACM Trans. Softw. Eng. Methodol. 16(4):13-63

Nuseibeh BA, Easterbrook SM (2000) Requirements engineering: a roadmap. In: 22nd international conference on software engineering. ACM Press, New York, pp 35-46