

Please quote as: Berkovich, M.; Hoffmann, A.; Leimeister, J. M. & Krcmar, H. (2011): Analysis of Requirements Engineering Techniques for IT enabled Product-Service-Systems. In: Requirements Engineering for Systems, Services, and Systems of Systems (RESS), Trento, Italy.

ANALYSIS OF REQUIREMENTS ENGINEERING TECHNIQUES FOR IT-ENABLED PRODUCT SERVICE SYSTEMS

Marina Berkovich
Information Systems
Technische Universität München
Munich, Germany
e-mail: berkovic@in.tum.de

Axel Hoffmann
Research Center for IS Design
Universität Kassel
Kassel, Germany
e-mail: axel.hoffmann@uni-kassel.de

Jan Marco Leimeister
Research Center for IS Design
Universität Kassel
Kassel, Germany
e-mail: leimeister@uni-kassel.de

Helmut Krcmar
Information Systems
Technische Universität München
Munich, Germany
e-mail: krcmar@in.tum.de

Abstract— Product service systems (PSS) are introduced by many companies to increase their differentiation and to provide integrated solutions to customers. PSS are integrated solutions consisting of physical products, software and services aiming at providing an individualized solution to a customer's problem. The specific attributes of PSS lead to specific requirements for requirements engineering (RE). The goal of this paper is to analyze to which degree the analysis techniques of software engineering are suitable for PSS. We therefore conducted a structured literature review of software engineering techniques. The criteria for assessing the suitability of the techniques were based on the characteristics of PSS and the task of RE in the development process of PSS. We analyzed five textbooks and 144 scientific articles and identified 27 groups of techniques. The result is that there are major gaps in techniques for RE for PSS. Two of ten criteria are not satisfied by any technique. Moreover, for the majority of tasks of RE multiple techniques have to be combined for satisfying the criteria. In summary, the literature review shows that the techniques of software engineering are largely not directly applicable to PSS.

Keywords- requirements engineering, product service system, complex solution, technique.

I. INTRODUCTION

Companies today are challenged by strong competition which causes the need for differentiation. The differentiation by physical products or services alone is no longer sufficient to be successful in the market [1]. Customers now want their problems to be solved by an integrated solution, instead of purchasing single products and services [2]. The companies become "customer-centric" and thus focus on offering solutions to customers' problems instead of offering single products and services [3]. These solutions consist of integrated bundles of technological elements (products)

developed by product and software engineering (PE and SWE) and services developed by service engineering (SE). The customer perceives them as being a single whole item [4]. Such complex solutions are also called hybrid products or product service systems (PSS) [5]. As software provides all basic functionalities such as storage, processing and communication, it plays an important role, particularly in providing innovative functionalities [6]. The following example clarifies what a PSS is according to Berkovich et al. [7]. The customer problem is the desire for a constant room temperature of 21 degrees. The contractor solves the customer's problem by offering a bundle of elements such as radiator (technological elements), control system (software) and maintenance services. Thus, the customer does not receive the elements separately but as a bundle.

PSS have special characteristics that result in special needs for the development and particularly the requirements engineering (RE) of them. At the center of the PSS concept is the enhancement of customer satisfaction [8] and provisioning of an added-value for the customer by adapting the solution to customer needs [1]. Due to the wide range of different stakeholders involved in PSS [9], it is challenging to manage the requirements of all of them. In order to adapt the PSS to customer needs and to integrate them both technically and organizationally into the value-creation process of the customer [10], the requirements emerging from the environment of the customer must be captured and analyzed [11]. The needs to customize the PSS for each customer by understanding the special character of the customers' environment [10] and the need to decompose the PSS into loosely coupled modules that can be standardized and reused [4] require a comprehensive understanding of customer needs and a domain-spanning management of requirements.

For successful RE for PSS, adequate techniques are needed. In practice as well as in the literature, physical products and services are developed separately [2]. Although RE is done systematically in all domains, a common and integrated approach is still missing [12-14]. Due to the special characteristics of PSS, it is unclear whether existing RE techniques are suited for PSS. In this paper RE techniques of software engineering are analyzed with regard to their suitability for PSS. In order to do so, we conduct a structured literature review according to Kitchenham et al. [15]. First, in section II, the context of the development process of PSS and the role of RE therein is explained. Next, in section III, the research design is described: The research question is defined and based on the context, and the process of selecting relevant literature is explained. The criteria for the analysis of the literature are then defined in detail. In section IV the research results – the found literature and the analysis thereof using the previously defined criteria – is presented. The paper concludes with a discussion and outlook for further research.

II. CONTEXT

The development process of PSS consists of task clarification, product conception and a development-specific component design [16]. In the phase of task clarification, the requirements are elicited and analyzed. Then, a first structure of the product is developed which discerns between physical and immaterial parts so that the requirements can be partitioned according to it. In the next phase of product conception, the product structure is augmented by functions and their interdependencies, and the requirements are partitioned again. In the last phase, the development-specific component design takes place, in which the components of the PSS are developed by single domains.

Requirements engineering is seen as an important task in the development process. It has the task of systematically collecting the requirements for a product, and it is essential that the requirements are complete and correct [17]. RE is a critical phase in development [18] because weaknesses in this phase leads to project failures and to cost-intensive changes during later phases [19]. The special characteristics of PSS, as explained in the introduction, make RE especially important to them [20]. In order to achieve an integrated development, all participating domains need to have a common understanding of the customer's problem and should clarify whether the developed solution meets the initial requirements.

In the following, the tasks of PSS (often also called "phases of RE") in the life-cycle of PSS are clarified. Thereby, we explain how the characteristics of PSS influence the tasks RE has to accomplish. First, RE has to determine the sources of requirements and the suited techniques for eliciting them. Next, the requirements are analyzed and concretized, which means that the initial requirements – in the language of the customer – are translated into target-characteristics of the product, in the language of the developer [21]. The analysis has the task of assuring that the requirements are not conflicting and that they are prioritized according to the importance for the customer. Conflicts have

to be solved by negotiating between all stakeholders. The next step is making a comparison between customer and concretized requirements in order to achieve a common understanding among all domains. The requirements are then documented in a requirements specification that serves as input for the following steps of the development process. During the development, the RE has to manage the requirements in order to ensure that the requirements are kept up to date. This includes change management of requirements and providing traceability in order to be able to assess the impact of changes of requirements on other requirements and solution components.

III. RESEARCH DESIGN

The steps of the literature review were planned according to Brereton et al. [22]. The structure of the paper was aligned according to the guidelines for a systematic literature review proposed by Kitchenham et al. [15].

A. Research Question

The goal of this paper is to analyze whether the techniques for RE of software engineering are applicable to PSS. Therefore, the research question addressed in this study is:

“To which degree are the RE techniques of software engineering suitable for PSS?”

First, this research question addresses whether a RE technique of software engineering is applicable to PSS. It is further checked whether the specific characteristics of PSS are handled by the technique in an appropriate manner. The combination of these two factors describes the suitability of the technique.

B. Identification of Relevant Research

For the literature review, we selected textbooks and articles that describe RE techniques in software engineering. A first challenge is that the term “technique” is often used interchangeably with “method” and “tool.” Since commercially available tools are also evaluated in research studies, we will not evaluate them ourselves, but rely on literature doing so. In order to analyze techniques for RE, it is important to have a clear-cut definition of “technique.” Therefore, the related terms “method” and “tool” are defined and differentiated from each other by reviewing definitions of these terms in both software engineering and method engineering. A *tool* is an automatic way to support a part of the development process [11]. According to Brinkkemper [23], Greiffenberg [24], and Nuseibeh and Easterbrook [11], a *method* is an approach that describes the conduction of an entire development process or project. It provides a prescription of how to perform a collection of activities. Thus, a method provides a systematic approach of how to use different techniques. A *technique* can be defined as a part of a method that gives concrete and tangible instructions for how to conduct the work of an activity. Brinkkemper [23] defines a technique as “a procedure, possibly with a prescribed notation, to perform a development activity.” This definition is relevant for our research.

As a starting point in selecting the relevant literature, common textbooks on RE in software engineering were considered. Therefore, the top five selling books on RE according to “amazon.de” and “amazon.com” (accessed on 10.02.2010) were searched for RE techniques. Since a textbook usually describes several different techniques, for each technique it must be decided if it is relevant. A technique is considered relevant if it describes how a task of RE (see section *context*) is conducted.

Regarding journal articles, we decided to search all publications in A-journals and A-conferences according to WI-list [25], as well as in the “Requirements Engineering Journal” of the years 2003 to 2010. An initial selection of the publication was done by automatically searching for the publications for the following key words: “requirements engineering,” “stakeholder,” “requirements [documentation OR elicitation OR verification OR validation OR traceability OR management OR specification OR analysis]” and “[customer OR functional OR non (-) functional] requirements.” For each resulting article, we manually decided whether it was relevant for analysis. As for textbooks, a technique is considered relevant if it describes how a task of RE (see section *context*) is conducted.

C. Data Analysis Method

The techniques found in the literature were analyzed for their suitability for RE for PSS in this study. In general, a technique is suited for PSS if it describes how a task of the RE (see section *context*) is conducted, whereby the technique must be able to handle the special characteristics of PSS. Because the single tasks of RE are very different, a proper set of criteria was developed for each task. In the following, for each task of RE a subsection is introduced that describes the criteria of the respective task.

1) Analysis Criteria for Task “Requirements elicitation”

The task of requirements elicitation is to identify all requirements of all stakeholders. In the context of PSS, this is especially challenging because of the different nature of stakeholders and the resulting interdisciplinarity. Therefore, the techniques must enable the *inter-domain communication* in order to get all requirements without neglecting a certain domain.

Another common challenge in elicitation is to also get implicit requirements from the stakeholders. When eliciting requirements for PSS, the stakeholders do not know which parts the solution will consist of, and thus they express requirements to the solution as a whole. The requirements to the services offered in conjunction with the solution are then usually neglected. This means that the requirements to the services are only implicitly stated [26]. Thus, the techniques must be able to catch *implicitly expressed requirements to services*.

Another important issue for the elicitation in the context of PSS is that the PSS has to be integrated into the system landscape and the business processes of the customer. For a successful integration, it is necessary to elicit the *requirements from the value-creation processes* of the customer. This includes the identification of relevant sources of requirements and the elicitation of the requirements.

2) Analysis Criteria for Task “Requirements analysis and negotiation”

The RE has the task to concretize the requirements initially elicited. The task of *concretization* is part of the requirements analysis. According to Sommerville and Sawyer [27], the goal of the requirements analysis is “to establish an agreed set of requirements which are complete and consistent.” In the context of PSS, the stakeholders express the requirements to the whole solution, but then they are concretized and partitioned according to the single domains which have to realize the requirements. Therefore, the RE must provide techniques to perform the analysis in order to get the solution-requirements in the language of the developer [28]. The analysis techniques should translate the requirements into the “language of developers” so that they can be realized by the domains. These techniques should *assign the requirements to the related domains*.

It is necessary to be able to identify conflicts, identify their causes, and then resolve them [29]. Conflicts between requirements mean that requirements are contradicting each other [11]. In single domains there are techniques available for finding and resolving conflicts, but in the context of PSS, the challenge is to find conflicts between the requirements of different domains. Because of the different nature of the domains’ requirements, it is difficult to identify the conflicts, e.g., conflicts between requirements for software and requirements for services are very challenging to find. The negotiation techniques should be able to detect and resolve the conflicts between the initial requirements and especially *inter-domain requirement conflicts* between requirements belonging to the domains involved. Apart from the negotiation, the techniques for prioritizing requirements according to their importance for the stakeholders are also necessary. According to their importance for the customer, the requirements must be *prioritized without overrating one domain*.

3) Analysis Criteria for Task “Requirements documentation and management”

The requirements documentation has the task to assure a complete, unambiguous, traceable, and continuous recording of requirements [30]. The documentation techniques should cope with the *domain specific characteristics of requirements* of PSS. First, the requirements to services include a resource-, process-, and result-dimension with different characteristics. The resource dimension describes requirements to the resources used by the solution provider, such as human resources. The process dimension describes the requirements of the process of conducting the service, and the result dimension describes the desired outcome of the service. These dimensions describe the immaterial nature of service requirements [31]. The documentation techniques must be able to capture these different forms of requirements. Second, the requirements to products describe characteristics and attributes of physical systems, which need to be documented.

Another important task of RE is the management of the requirements, which has the task of ensuring that the current state of the requirements is available at all times. This aspect

is important because requirements change throughout the development process by increasing knowledge about the product, not only by changing customer wishes, but also by technical constraints [28, 32]. In order to achieve these characteristics, it is necessary to provide *traceability*, which is defined as “the ability to describe and follow the life of a requirement, in both a forwards and backwards direction” [33]. In the context of PSS, the traceability techniques should enable capturing the requirements sources and interdependencies, and they have to explicitly detect interdependencies between requirements of different domains, as well as within one domain.

IV. RESULTS OF THE STUDY

After the description of the search results, we present the results of the analysis. Similar to the analysis criteria in section C, the results are structured according to the tasks of RE. For each task a subsection is introduced describing the fulfillment of each criteria of the respective task. Due to the large number of journal articles found, we will not describe each of them. They were structured into categories; and for each category some journal papers are cited exemplarily.

A. Search Results

The five top-selling books on RE were the following: Lamsweerde [34], Sommerville and Kotonya [35], Hull et al. [21], Pohl [19], and Aurum and Wohlin [28]. Each of these books describes techniques for all tasks of RE, and thus each book was included in all categories. The journal articles found in the search mostly handled one specific topic of RE and were categorized according to the analysis criteria. Table I shows the analyzed textbooks and articles.

TABLE I. NUMBER OF ANALYSED TEXTBOOKS AND ARTICLES

| | Elicitation | Analysis and Negotiation | Documentation and Management |
|------------------|-------------|--------------------------|------------------------------|
| Textbooks | 5 | 5 | 5 |
| Journal articles | 36 | 71 | 37 |

B. Requirements Elicitation

The requirements elicitation is “all about learning the needs of users, and communicating those needs to system builders” [36]. We analyzed 36 articles and the text-books described in the chapter “Research Methodology” that were handling the requirements elicitation. The techniques found

were categorized according to the categories of Nuseibeh und Easterbrook [11]. Table II shows the results.

The traditional techniques (e.g., interviews, written survey, and analysis of existing documents) are flexible, but their results strongly depend on the qualification of the performing person (e.g., [17, 35]). When applying these techniques it is important to be aware of the goal to be achieved [37]. They are partly suited for eliciting implicit requirements [38]; however, it must be ensured that the “right” questions are asked in order to support the inter-domain communication and the identification of requirements from the value-creation process.

The “Group elicitation” techniques (e.g., brainstorming, focus groups and workshops) augment the traditional techniques by using group-dynamic effects in elicitation sessions (e.g. [21, 39]), and are thus able to identify implicit requirements. They are able to identify the requirements for the value-creation process, provided that the right stakeholders are taking part in the group sessions.

Prototyping is a specialized technique which is used to increase the stakeholders’ understanding of possible solutions. Prototypes can be used for visualization and are therefore able to increase the inter-domain understanding. However, prototypes are only useful for concepts that can be realized in some form. Especially for services, this is not the case because they are of immaterial nature. Also value-creation processes and the integration of PSS into the customer’s processes cannot be visualized using prototyping.

The model-driven techniques (e.g., goal-oriented approaches) “provide a specific model of the type of information to be gathered and use this model to drive the elicitation process” [40]. Since each domain must be able to understand the goals of other domains, these techniques do not provide the possibility of supporting the inter-domain communication. Additionally, they can only be used to elicit implicit requirements if the stakeholders are familiar with them.

The cognitive techniques (e.g., protocol analysis, card sorting, and repertory grids) use approaches for the communication with stakeholders using visual representations of requirements (e.g. [17]). They support the inter-domain communication and identification of implicit requirements. For the elicitation of the value-creation requirements, it is necessary to focus on the right aspects and to select the right stakeholders.

TABLE II. ANALYSIS OF TECHNIQUES FOR THE REQUIREMENTS ELICITATION (OWN ILLUSTRATION)

| Criteria \ Techniques | Techniques | | | | | |
|--|------------------------|--------------------------------|-------------|-------------------------|----------------------|-----------------------|
| | Traditional techniques | „Group elicitation“ techniques | Prototyping | Model-driven techniques | Cognitive techniques | Contextual techniques |
| Identification of implicitly expressed requirements | ● | ● | ◐ | ◐ | ● | ◐ |
| Identification of requirements from the value-creation process | ◐ | ◐ | ○ | ◐ | ◐ | ◐ |
| Inter-domain communication | ◐ | ● | ◐ | ◐ | ● | ○ |

● completely met ◐ partly met ○ not met

The success of the contextual techniques (e.g., ethnography, observation, and introspection) depends largely on the skills of the performing expert (e.g. [11]). They are suited for eliciting the value-creation requirements to a certain degree. It is necessary that suitable situations for observations exist and that the requirements analyst asks the “right” questions.

C. Requirements analysis and negotiation

For the topic of requirements analysis, we analyzed 46 conference and journal papers on requirements analysis and 25 papers on requirements negotiation, in addition to the books on RE. The various modeling techniques used in RE were seen as a part of the analysis because modeling requirements forces a deep understanding of them [21]. Even though each modeling technique focused on different aspects, we evaluated each technique for its suitability for PSS. Table III and IV show the results of the analysis.

The data flow diagrams are used to model a functional perspective of the system-to-be [35], and are therefore able to concretize the requirements, but the diagrams cannot be used to divide the requirements according to domains and to consider the different dimensions of services. Use cases describe interactions between external actors and the system in a structured way [41] and help to think about the solution in an abstract way, one that is independent of a possible technical solution. In this way, they allow it to stepwise concretize the requirements until they can be assigned to a domain. Entity relationship diagrams describe structural aspects of the system-to-be [21]. They are used to model the data that is processed by the solution without talking about a technical solution. This helps to concretize and complete requirements and to assign them to the domains step by step. Both the use case and entity relationship diagram are able to

cover requirements to the services. But only the requirements of the resource- and result-dimension can be analyzed using these techniques. The activity diagrams represent courses of actions, whereas interaction diagrams describe the message exchange between actors and the system [42]. These types of diagrams are able to describe the sequences of actions and interactions between actors. Therefore, they can be used for software, hardware and for the process-dimension of services. The class diagrams represent a structural perspective, and state charts describe states of the system and transitions between them.

These diagrams are useful to model certain kinds of requirements and are therefore suitable for the concretization of requirements, but they are not able to divide the requirements according to domains. The initial QFD approach of Akao [43] has been adapted for PSS and enables the translation of initial requirements into characteristics of products [44]. All of these techniques are not suitable for the modularization, but the SCORE-method of Böhmman et al. [4] was developed for that.

For the identification of conflicts and the resolution of these, other techniques were found in the literature. Interviewing techniques [45] help to find conflicting requirements of different stakeholders, especially those caused through non-formal representation. Because of their generality they are also able to find conflicts between different domains, but the completeness of found conflicts depends solely on the interviewers’ abilities. They provide no guidelines to resolve the conflicts. Formal techniques, such as model checking [46, 47], discover conflicts in formal specifications. However, only some special types of requirements can be represented formally, and only certain types of conflicts can be found in this way. Comparison techniques, such as interaction matrices [29], help to check

TABLE III. ANALYSIS OF TECHNIQUES FOR THE REQUIREMENTS ANALYSIS (OWN ILLUSTRATION)

| Criteria \ Techniques | Data-flow diagrams | Use Cases | Entity-Relationship Diagrams | Activity Diagrams | Statechart Diagrams | Class Diagrams | QFD |
|--------------------------------|--------------------|-----------|------------------------------|-------------------|---------------------|----------------|-----|
| Concretization of requirements | ◐ | ● | ● | ● | ◐ | ◐ | ◐ |
| Assign the req. to the domains | ○ | ● | ◐ | ○ | ○ | ○ | ◐ |

● completely met ◐ partly met ○ not met

TABLE IV. ANALYSIS OF TECHNIQUES FOR THE IDENTIFICATION AND RESOLUTION OF CONFLICTS BETWEEN REQUIREMENTS - REQUIREMENTS NEGOTIATION (OWN ILLUSTRATION)

| Criteria \ Techniques | Interviewing techniques | Formal techniques | Comparison techniques | Competing negotiation techniques | Collaborative negotiation techniques | Decision techniques |
|---|-------------------------|-------------------|-----------------------|----------------------------------|--------------------------------------|---------------------|
| Identification of inter-domain requirements conflicts | ◐ | ◐ | ● | ○ | ○ | ○ |
| Prioritizing the requirements without overrating one domain | ○ | ○ | ○ | ● | ● | ◐ |

● completely met ◐ partly met ○ not met

requirements against each other, but they do not offer help in resolving conflicts. Apart from the structured comparison of requirements, they do not offer help in handling requirements of different domains. Competing negotiation techniques representing the classical negotiation with offers, counter-offers and voting [28] are suited to resolve conflicts between different domains. Collaborative negotiation techniques, such as the WinWin approach [48], try to find a satisfying result for every stakeholder, whereas decision making techniques, such as the feasibility check [49], try to find the best requirements solution appropriate to cost, complexity and risk.

D. Requirements documentation and management

Regarding the documentation of requirements, we analyzed 10 papers in addition to the books on RE. Table V shows the results of the analysis. Story techniques, such as play scripts [50], are used to document functional requirements by describing the interaction between the product and the user in natural language. The requirements to services, as well as the required physical characteristics of products, cannot be formulated in this way. Formatted specification techniques define a structure to organize requirements documents in natural languages [19]. Therefore, it is possible to define a structure containing chapters for the service requirements and physical requirements. However, the requirements themselves have to be documented in natural language without further guidance.

Standardized language techniques, such as Partial Reductive Paraphrase [51], offer a well-defined language to avoid misunderstandings. These techniques can be applied for describing the initial and the concretized requirements. For service requirements, these techniques only provide minor benefits because it is not possible to define an exact language covering all possible customer wishes. For the requirements to the physical characteristics of products, it is possible to define a standardized language, covering all requirements.

Formal specification techniques which rely on a formal model of the system to be built in order to document clear-cut requirements. For services where no such model is available, this technique is not applicable, whereas for certain types of physical products, such models are available. However, a full coverage of all physical products is non-existent. Table V shows the results of the analysis.

To analyze traceability techniques, 27 journal articles and conference papers were reviewed. Textual references and hyperlinks are used to describe the relations between input- and output artifacts. Using these techniques, each artifact directly refers to its predecessors and successors, and thus applicable to all types of requirements. Traceability-matrices describe the relations between requirements, and requirements or components of the system in a separate matrix. They are also applicable to all types of requirements. In traceability-graphs, the vertices represent artifacts and the edges relations between the artifacts. This technique is also applicable to all types of requirements. Table VI shows the results of the analysis.

V. RELATED WORK

One part of the related work in the literature focuses on classifying and categorizing techniques for all tasks of RE. Jiang et al. [52] present a methodology for the analysis and selection of RE techniques. They developed criteria for the analysis of techniques, and evaluated the techniques in expert workshops. We used this methodology for selected PSS specific criteria. Hickey and Davis [18, 36] present a method for effective selection of RE techniques in specific projects. Bickerton und Siddiqi [53] developed a framework for the classification of RE techniques, based on hypothesis on organizational development. Macaulay [54] presents a set of techniques that was developed based on identified requirements of a project. Other studies focus only on one special activity of RE; for example, Browne and Ramesh [55] focus on the selection of techniques for the requirements elicitation. Their approach is based on a cognitive model. In

TABLE V. ANALYSIS OF TECHNIQUES FOR THE REQUIREMENTS' DOCUMENTATION (OWN ILLUSTRATION)

| Techniques \ Criteria | Story techniques | Formatted specification techniques | Standardized languages techniques | Formal specification techniques |
|--|------------------|------------------------------------|-----------------------------------|---------------------------------|
| Document the different dimensions of service requirements | ○ | ◐ | ◐ | ○ |
| Document requirements to characteristics and attributes of physical products | ○ | ◐ | ● | ◐ |

● completely met ◐ partly met ○ not met

TALBE VI. ANALYSIS OF TECHNIQUES FOR THE TRACING OF REQUIREMENTS (OWN ILLUSTRATION)

| Techniques \ Criteria | Textual References | Hyperlinks | Traceability Matrix | Graphs |
|-----------------------|--------------------|------------|---------------------|--------|
| Traceability | ● | ● | ● | ● |

● completely met ◐ partly met ○ not met

contrast to our work, these papers present methods only for classifying or selecting techniques, without actually applying them on specific techniques. Furthermore, this related work is limited to RE for software only. The papers focusing on the selection of techniques often provide lists of techniques, without analyzing them. The major difference from our research is that the related work does not make the connection to PSS and they only examine each technique for just one phase of the RE-process. In contrast, our goal was to provide an overview of techniques for RE for PSS.

VI. CONCLUSION

This paper has analyzed RE techniques of software engineering for their suitability for PSS. First, the tasks of RE and the special characteristics of PSS that have to be regarded were explained, after which the method for selecting and analyzing the literature was presented. For each task of RE, a set of proper criteria was developed to assess the suitability of the techniques. For each task of RE, the results of the study describe the degree to which the techniques available for it satisfy the predefined criteria. Since several publications present variations of the same technique, in the analysis these techniques were grouped accordingly. In the following, we summarize the conclusions from this study, discuss the implications, and recommend further research in this area.

A. Discussion

The overall result of this paper (see Table VII) is that the techniques are very unequally suited for PSS. In

requirements elicitation none of the six groups of techniques fulfill all criteria: one criterion is not covered by any technique and the other two are only covered by the minority of techniques. It is important for PSS to cover the implicit requirements of services. Some techniques of software engineering offer general methods that are able to do so. The elicitation of requirements from the value-creation process of the customer is hardly supported by the techniques. The methods of software engineering are only partly applicable to it. The need to support the inter-domain communication when eliciting requirements is only addressed by two techniques. In *requirements analysis*, only Use Cases fulfill all criteria. For the concretization of requirements, Use Cases, Entity-Relationship-Diagrams, and Activity-Diagrams can be used. The task of assigning the requirements to the domains participating in the development of a PSS is only addressed by Use Cases. The need for action regarding the partitioning of requirements to the domains is especially high; the majority of techniques do not support it at all. Out of six groups of technique, none supports both the *identification of conflicts* and their resolution by prioritizing the requirements. For the identification of conflicts, only the simplest technique is applicable, and to be able to find these inter-domain conflicts, all requirements must be compared to each other. In order to resolve the conflicts in software engineering, two groups of negotiation techniques are available that are also applicable in the context of PSS.

Regarding *requirements documentation*, service requirements are a major challenge. None of the four analyzed groups of techniques is fully suitable to document

TABLE VII. OVERALL RESEARCH RESULTS (OWN ILLUSTRATION)

| RE-Process-Phase | Criteria | Techniques met Criteria | | |
|---|--|-------------------------|------------|---------|
| | | Completely met | Partly met | Not met |
| Requirements Elicitation | Implicit requirement identification | 3 | 3 | 0 |
| | Value-creation-process requirement identification | 0 | 5 | 1 |
| | Inter-domain communication | 2 | 3 | 0 |
| Requirements analysis and negotiation | Concretization of requirements | 3 | 4 | 0 |
| | Assign the req. to the domains | 1 | 2 | 4 |
| | Inter-domain requirements conflict identification | 1 | 2 | 3 |
| | Prioritizing Requirements w/o overrating one domain | 2 | 1 | 3 |
| Requirements documentation and management | Traceability | 4 | 0 | 0 |
| | Document the different dimensions of service requirements | 0 | 2 | 2 |
| | Document requirements to characteristics and attributes of physical products | 1 | 2 | 0 |

the service requirements. Half of the techniques support it partly. These techniques propose standardized languages and structures for specification documents that can help. For the documentation of characteristics and attributes of physical products, one of the analyzed techniques can be used. This technique proposes a standardized scheme for requirements that can be adapted to physical characteristics of products. The challenge of *requirements traceability* is fully solved by the analyzed techniques. All techniques are completely applicable to PSS because of their general nature.

B. Implications and Future Work

Although the single domains involved in the development of PSS have developed mature techniques for RE, it has still been largely unclear whether they are suited for PSS. We have presented an analysis which evaluated RE techniques of software engineering regarding their applicability for RE for PSS. Our goal was to find suitable techniques for the different phases of the RE-process. The result is that there are some major gaps. Out of the ten criteria defined for the RE techniques, two are not satisfied by any technique. Moreover, for the majority of tasks of RE, multiple techniques of software engineering have to be combined for satisfying the criteria. Summing up, the study shows that the techniques of software engineering are largely not directly applicable to PSS. However, the combination of different techniques and the adaptation of them may be promising for future work. To enable a purposeful RE for PSS in the future, our future research aims at developing a catalogue of techniques with selection criteria that help to assemble an adequate set of techniques for individual project constellations. Thus, our immediate further research will include the analysis of techniques from product and service engineering.

ACKNOWLEDGEMENT

We thank the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) for funding this project as part of the collaborative research center „Sonderforschungsbereich 768 – Managing cycles in innovation processes – Integrated development of product-service-systems based on technical products. For further information, please visit: <http://www.sfb768.de>. We thank Hesse's Ministry of Higher Education, Research, and the Arts for funding the project as part of the research funding program “LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz”. For further information, please visit: <http://www.iteg.uni-kassel.de/venus>.

REFERENCES

[1] M. Sawhney, "Going beyond the Product: Defining, Designing and Delivering Customer Solutions," in *The Service-dominant Logic of Marketing*, R. F. Lusch and S. L. Vargo, Eds. New York: M. E. Sharpe, 2006, pp. 365-380.

[2] J. M. Leimeister and C. Glauner, "Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik," *Wirtschaftsinformatik*, vol. 50, pp. 248-251, 2008.

[3] R. Tuli, A. Kohli, and S. Bharadwaj, "Rethinking Customer Solutions: From product Bundles to Relational Processes," *Journal of Marketing*, vol. 71, pp. 1-17, 2007.

[4] T. Böhmman, P. Langer, and M. Schermann, "Systematische Überführung von kundenspezifischen IT-Lösungen in integrierte Produkt-Dienstleistungsbausteine mit der SCORE-Methode," *Wirtschaftsinformatik*, vol. 50, pp. 196-207, 2008.

[5] A. R. Tan, T. C. McAloone, and C. Gall, "Product/Service-System Development - An explorative Case Study in a manufacturing Company," in *international conference on engineering design, ICED 07*, Paris, France, 2007.

[6] H. Krcmar, *Informationsmanagement*, 5 ed. Berlin: Springer, 2010.

[7] M. Berkovich, J. M. Leimeister, and H. Krcmar, "Suitability of Product Development Methods for Hybrid Products as Bundles of Classic Products, Software and Service Elements," in *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*, San Diego, 2009.

[8] A. Davies, "Moving base into high-value integrated solutions: a value stream approach," *Industrial & Corporate Change*, vol. 13 pp. 727-756, 2004.

[9] J. R. Galbraith, "Organizing to Deliver Solutions," *Organizational Dynamics*, vol. 31, pp. 194-207, 2002.

[10] T. Böhmman and H. Krcmar, "Hybride Produkte: Merkmale und Herausforderungen," in *Wertschöpfungsprozesse bei Dienstleistungen: Forum Dienstleistungsmanagement*, M. Bruhn and B. Stauss, Eds.: Gabler, 2007, pp. 240-255.

[11] B. A. Nuseibeh and S. M. Easterbrook, "Requirements Engineering: A Roadmap," in *22nd International Conference on Software Engineering, ICSE'00*, 2000.

[12] M. Berkovich, J. M. Leimeister, and H. Krcmar, "An empirical exploration of requirements engineering for hybrid products," in *XVIIIth European Conference on Information Systems (ECIS)*, Verona, 2009.

[13] M. Berkovich, "Requirements Engineering for IT-Enabled Product Service Systems," in *The 18th IEEE International Requirements Engineering Conference, Doctoral Consortium* Sydney, Australia, 2010.

[14] M. Berkovich, S. Esch, J. M. Leimeister, and H. Krcmar, "Requirements engineering for hybrid products as bundles of hardware, software and service elements – a literature review," in *9. Internationale Tagung Wirtschaftsinformatik (WI)*, Wien, Österreich, 2009.

[15] B. A. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - a systematic literature review," *Information and Software Technology*, vol. 51, pp. 7-15, 2009.

[16] D. Spath and L. Demuß, "Entwicklung hybrider Produkte – Gestaltung materieller und immaterieller Leistungsbündel," in *Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen*, H.-J. Bullinger and A.-W. Scheer, Eds. Berlin, Heidelberg, New York: Springer, 2003, pp. 463-502.

[17] T. A. Byrd, K. L. Cossick, and R. W. Zmud, "A synthesis of research on requirements analysis and knowledge acquisition techniques," *MIS Quarterly*, vol. 16, pp. 117-138, 1992.

[18] A. M. Hickey and A. Davis, "A Unified Model of Requirements Elicitation," *Journal of Management Information Systems*, vol. 20, pp. 65-84, 16.02.2008 2004.

[19] K. Pohl, *Requirements Engineering. Grundlagen, Prinzipien, Techniken* 1ed.: Dpunkt Verlag, 2007.

[20] G. J. Browne and M. B. Rogich, "An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques," *J. o. M. I. Systems*, Ed., 2001, p. 28.

[21] E. Hull, K. Jackson, and J. Dick, *Requirements Engineering* vol. 2. Berlin: Springer, 2004.

[22] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khahil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 4, pp. 571-583, 2007.

- [23] S. Brinkkemper, "Method engineering: engineering of information systems development methods and tools," *Information and Software Technology*, vol. 38, pp. 275-280, 1996.
- [24] S. Greiffenberg, *Methodenentwicklung in Wirtschaft und Verwaltung* vol. 1: Kovac, J., 2004.
- [25] "WI-Orientierungslisten," in *Wirtschaftsinformatik*. vol. 2, 2008.
- [26] C. v. Husen, "Anforderungsanalyse für produktbegleitende Dienstleistungen," in *Fakultät Maschinenbau Stuttgart: Universität Stuttgart*, 2007, p. 187.
- [27] I. Sommerville and P. Sawyer, "Viewpoints: principles, problems and a practical approach to requirements engineering," *Annals of Software Engineering*, vol. 3, pp. 101-130, januar 1997 1997.
- [28] A. Aurum and C. Wohlin, *Engineering and Managing Software Requirements* Ied. Berlin: Springer, 2005.
- [29] A. Lamsweerde, "Requirements engineering in the year 00: a research perspective," in *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, 2000, pp. 5 - 19.
- [30] IEEE Std 1233, "IEEE guide for developing system requirements specifications," *IEEE Std 1233, 1998 Edition*, 1998.
- [31] B. Edvardsson and J. Olsson, "Key Concepts for New Service Development," *Service Industries Journal*, vol. 16, pp. 140-164, 1996.
- [32] M. Jarke, "Requirements tracing," *Commun. ACM*, vol. 41, pp. 32-36, 1998.
- [33] O. C. Z. Gotel and C. W. Finkelstein, "An analysis of the requirements traceability problem," in *Requirements Engineering, 1994., Proceedings of the First International Conference on Colorado Springs, CO*, 1994.
- [34] A. v. Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications* vol. 1: Wiley & Sons, 2009.
- [35] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques* 1 ed.: Wiley & Sons, 1998.
- [36] A. M. Hickey and A. M. Davis, "Elicitation technique Selection: how do experts do it?," in *11th IEEE international Requirements Engineering Conference*, Monterey Bay, 2003.
- [37] D. Zowghi and C. Coulin, "Requirements Elicitation: A Survey of Techniques, Approaches," in *Engineering and Managing Software Requirements* Ied, A. Aurum and C. Wohlin, Eds. Berlin: Springer, 2005, pp. 17-47.
- [38] N. A. M. Maiden and G. Rugg, "ACRE: selecting methods for requirements acquisition," *Software Engineering Journal*, vol. 11, pp. 183-192, 1996.
- [39] J. Coughlan and R. D. Macredie, "Effective Communication in Requirements Elicitation: A Comparison of Methodologies," *Requirements Engineering Journal*, vol. 7, pp. 47-60, 2002.
- [40] M. d. Hengst, E. v. d. Kar, and J. Appelman, "Designing Mobile Information Services: User Requirements Elicitation with GSS Design and Application of a Repeatable Process," in *Proceedings of the 37th Hawaii International Conference on System Sciences Hawaii* 2004.
- [41] A. I. Antón, J. Dempster, and D. Siege, "Deriving Goals from a Use Case Based Requirements Specification for an Electronic Commerce System," in *6th Int. Workshop on Requirements Engineering: Foundation for Software Quality REFSQ'00*, 2000.
- [42] T. J. Larsen, F. Niederman, M. Limayem, and J. Chan, "The role of modelling in achieving information systems success: UML to the rescue?," *Information Systems Journal*, vol. OnlineEarly Articles, 2007.
- [43] Y. Akao, *Quality Function Deployment: Quality Function Deployment - Integrating Customer Requirements into Product Design*: Productivity Press, 1990.
- [44] Y. An, S. Lee, and Y. Park, "Development of an integrated product-service roadmap with QFD: A case study on mobile communications," *International Journal of Service Industry Management*, vol. 19, pp. 621 - 638, 2008.
- [45] N. Niu and S. Easterbrook, "So, You Think You Know Others' Goals? A Repertory Grid Study," *Software, IEEE*, vol. 24, pp. 53-61, 2007.
- [46] J. H. Hausmann, R. Heckel, and G. Taentzer, "Detection of conflicting functional requirements in a use case-driven approach," in *Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on*, 2002, pp. 105-115.
- [47] A. P. Felty and K. S. Namjoshi, "Feature specification and automated conflict detection," *ACM Trans. Softw. Eng. Methodol.*, vol. 12, pp. 3-27, 2003.
- [48] P. Gruenbacher, "Collaborative requirements negotiation with EasyWinWin," in *Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on*, 2000, pp. 954-958.
- [49] A. Herrmann, B. Paech, and D. Plaza, "ICRAD: An integrated process for the solution of requirements conflicts and architectural design," *International Journal of Software Engineering and Knowledge Engineering*, vol. 16, 2006.
- [50] R. Riehle, "Playscript for use cases," *SIGSOFT Softw. Eng. Notes*, vol. 33, pp. 1-2, 2008.
- [51] K. S. Wasson, "A Case Study in Systematic Improvement of Language for Requirements," in *Requirements Engineering, 14th IEEE International Conference*, 2006, pp. 9-18.
- [52] L. Jiang, A. Eberlein, B. H. Far, and M. Mousavi, "A methodology for the selection of requirements engineering techniques," *Software and Systems Modeling*, vol. 7, pp. 303-328, 2008.
- [53] M. J. Bickerton and J. Siddiqi, "The classification of requirements engineering methods," in *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, 1993, pp. 182-186.
- [54] L. Macaulay, "Requirements for requirements engineering techniques," in *The Second International Conference on Requirements Engineering*, Colorado Springs, CO, 1996, pp. 157 - 164.
- [55] G. J. Browne and V. Ramesh, "Improving information requirements determination: A cognitive perspective," *Information and Management*, vol. 39, pp. 625-645, 2002.