Open access • Book Chapter • DOI:10.1007/978-3-642-19858-8_8

Research preview: supporting end-user requirements elicitation using product line variability models — Source link

Deepak Dhungana, Norbert Seyff, Florian Graf

Institutions: Siemens, University of Zurich, Johannes Kepler University of Linz

Published on: 28 Mar 2011 - Requirements Engineering: Foundation for Software Quality

Topics: Requirements elicitation, Requirement, Software product line, Requirement prioritization and Vision document

Research preview: Supporting end-user requirements elicitation using product line variability models

Dhungana, Deepak ; Seyff, Norbert ; Graf, Florian

Abstract: [Context and motivation] Product line variability models have been primarily used for product configuration purposes. We suggest that such models contain information that is relevant for early software engineering activities too. [Question/Problem] So far, the knowledge contained in variability models has not been used to improve requirements elicitation activities. State-of-the-art requirements elicitation approaches furthermore do not focus on the cost-effective identification of individual end-user needs, which, for example, is highly relevant for the customization of service-oriented systems. [Principal idea/results] The planned research will investigate how end-users can be empowered to document their individual needs themselves. We propose a tentative solution which facilitates end-users requirements elicitation by providing contextual information codified in software product line variability models. [Contribution] We present the idea of a "smart" tool for end-users allowing them to specify their needs and to customize, for example, a service-oriented system based on contextual information in variability models.

DOI: https://doi.org/10.1007/978-3-642-19858-8_8

Research Preview: Supporting End-user Requirements Elicitation Using Product Line Variability Models

Deepak Dhungana [1], Norbert Seyff [2], Florian Graf [3]

[1] Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems, Technical University of Vienna, Austria
deepak.dhungana@tuwien.ac.at

[2] University of Zurich, Department of Informatics
Binzmuehlestrasse 14, 8050, Zurich, Switzerland
seyff@ifi.uzh.ch

[3] Johannes Kepler University Linz, Austria
florian.graf1@gmail.com

Abstract. [Context and motivation] Product line variability models have so far been primarily used for product configuration purposes. We suggest that such models contain information that is relevant for early software engineering activities too. **[Question/Problem]** In this research preview paper, we propose using such models to support requirements elicitation. Service-oriented approaches allow providing customized software systems. However, current requirements elicitation approaches do not focus on the cost-effective identification of individual end-user needs. **[Principal idea/results]** The planned research investigates on how end-users can be empowered to document their individual needs themselves. We propose a tentative solution which facilitates end-users requirements elicitation by providing contextual information codified in software product line variability models. **[Contribution]** We present the idea of a "smart" tool for end-users allowing them to specify their needs and to customize a service-oriented system based on contextual information in variability models.

Keywords: End-user requirements elicitation; codified context knowledge; software product lines; automated customization of applications.

1 Introduction and Motivation

Novel software engineering paradigms such as service-oriented computing promote the reuse of available functionality and allow the cost- and time-effective composition of tailored software systems [1]. With such developments, we need to adapt traditional Requirements Engineering (RE) approaches to strengthen end-user involvement in software engineering activities. We foresee that end-users will be directly involved in customizing and tailoring applications to immediately get software fulfilling their needs [2]. This vision has to consider technical, social and methodological constraints and has significant implications for RE research and practice.

In traditional software engineering it is the *requirements analyst* who facilitates requirements gathering and who abstracts technical details away from end-users. However, requirements analysts are typically not involved in the daily lives of end-users and therefore they usually do not support them in specifying high quality requirements descriptions in situ. Research is needed to explore how end-users can be supported in documenting their needs themselves. Furthermore we need to explore how such end-user needs can be used to identify available software solutions and how end-users can be supported in adapting these software solutions to their needs.

The goal of our research is to *explore how contextual knowledge (in the form of product line variability models) can support end-users in specifying individual needs and self-customizing service-oriented solutions*. We present a preview of a tool-supported approach for end-user requirements elicitation using the knowledge "codified" in software product line variability models. This approach will enable end-users to specify their needs using natural language text. While documenting their needs sophisticated product line tools provide further information supporting requirements gathering. Particularly, end-users are asked questions about missing details regarding their needs. The advantage of this approach is the possibility to automatically evaluate given answers for further processing and customization of service-based applications. Thereby it will be possible to provide a service-based system fulfilling individual end-users' needs within a short time.

2 Research Preview

The aim of the planned research is to build tools and techniques that enable end-users in gathering requirements and consequently customizing a service-based application. We foresee, that product line variability models represent the "knowledge base" containing contextual information. This includes architectural aspects of service based solutions as well as information on their inherent variability.

Our research is based on the idea that end-users are able to express individual needs with natural language text descriptions [2]. We envision providing a "smart" tool, which allows end-users to enter their needs using natural language text. Analyzing these needs the tool identifies relevant context stored in product line variability models and based on this information it presents a dynamic questionnaire to the user. We envision that the questionnaire stimulates the requirements elicitation process. In ideal cases, answering the questions will enable the end-user to automatically customize the requested service-oriented application. This is possible using product generators that can process the set of reusable software artifacts, automatically identified by the product line variability modeling engine. In the long run, individual end-user needs are also used to maintain and evolve the product line variability model. As new relevant information can be mined from the users' needs, the product line models need to be continuously maintained and evolved to ensure the correctness and completeness of the codified information.

2.1 Approach Overview

An overview of the planned conceptual solution is presented in Figure 1. It depicts the key activities and the flow of information among the different participants. A service provider usually knows about the features of a service and how it can be adapted to different contextual situations. (1) In our approach this knowledge is codified in the form of product line variability models. (2) The variability model is used as input for an end-user requirements elicitation tool. The user enters natural language text and answers the questions presented by the tool. (3) The tool attempts to configure the required product based on the answers of the user. (4) The variability model is updated each time the user comes up with new contextual information. (5) The user's answers and the underlying product line model allow the generation of a service-based prototype application.

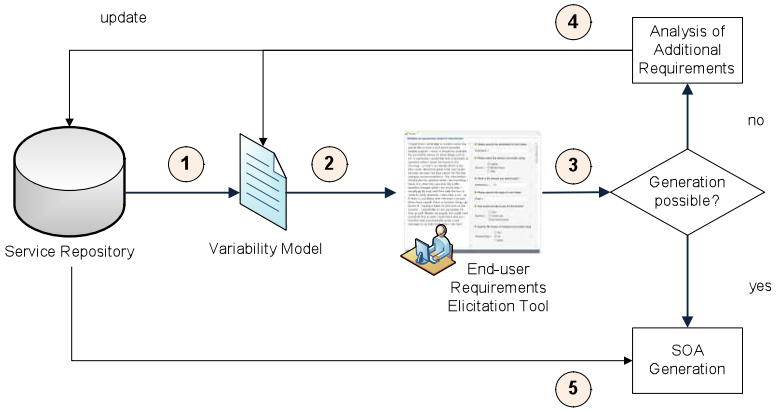

Figure 1 Overview of different activities associated with requirements elicitation using codified context knowledge models of service based applications.

2.2 Prototypic Tool Support

We have implemented an early tool prototype called EuReCuS (End-user Requirements Elicitation and Customization of Services), which enables end-users to enter requirements using natural language text and presents relevant questions, as the user enters her needs. EuReCuS is currently available as an Eclipse plug-in, utilizing the product line variability modeling capabilities of the DOPLER [3, 4] tool suite. DOPLER variability models consist of decisions, the users can take and rules that need to be considered when selecting services based on the users answers to the relevant questions. These variability models are executed using rule engines that are capable of mapping the user's decisions to available services and propagating the effects of such decisions in the configuration of the future system.

We envision that EuReCuS will enable end-users to document their needs with the help of a seemingly simple text editor. The text editor is however sensitive to what the user is typing. It is linked to the variability model execution engine to identify relevant questions and pass the end-user's answers on these questions. We are

currently adopting Apache Lucene, as a natural language analyzer and tokenizer. It is a high-performance, full-featured text search engine library written in Java. Lucene provides advanced features like stemming and synonym-based search.

Based on the entered text the tool identifies relevant decisions within the variability model and it displays corresponding questions to the end-user. Using interactive UI elements the end-user is able to answer the upcoming questions (see Figure 2). The answers are then passed to the variability modeling execution engine. The model execution engine passes information about selected services to the service composer. The output is an automatically generated customized application.

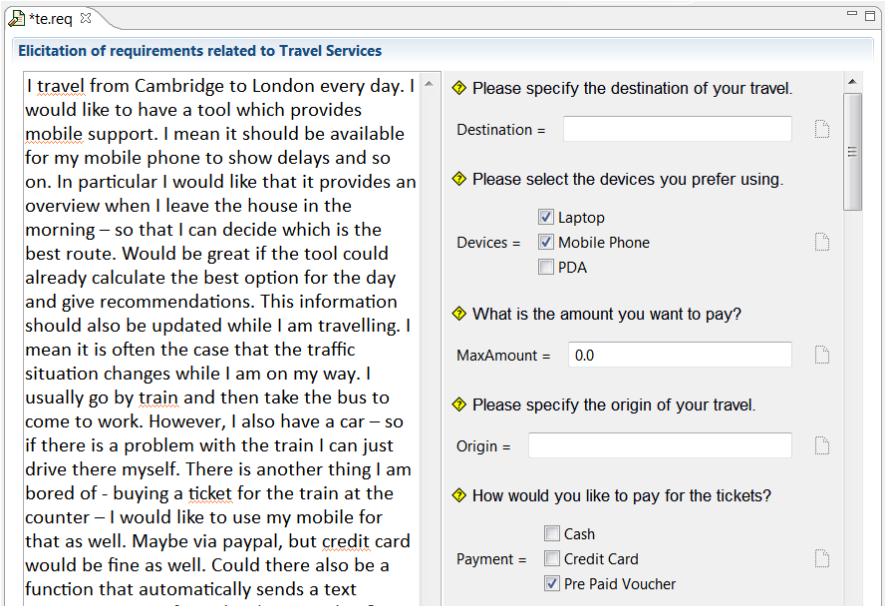

Figure 2 Screenshot of an early EuReCuS tool prototype, depicting the text editor on the left and the set of automatically identified relevant contextual questions on the right.

3 Application Example

To highlight the application of our approach we prepared an example discussing how an end-user would use the developed EuReCuS tool in order to document requirements and customize a software solution. We decided to use an example which discusses everyday needs of an end-user named Tom. His requirements describe how a future software system should support his daily commuting. Using our tool prototype Tom is able to document individual needs in the form of natural language text (see Figure 2).

Tom starts to describe individual needs with the help of natural language text using the text editor of EuReCuS. However, Tom, not being an RE expert and unfamiliar with requirements documentation, will most likely not document fully specified requirements descriptions. We expect Tom to provide a mixture of needs, rationale

descriptions, and uncertainties documented in a kind of user story. In general Tom's description is supposed to include a lot of contextual information. Tom, for example, could describe needs using statements such as: I would like to have a tool which provides mobile support and this (travel) information should also be updated while I am traveling. Using the EuReCuS text editor Tom is not forced to describe his needs following a predefined structure. Furthermore, the approach is not limiting Tom's creativity as he is allowed to document whatever comes into his mind.

While Tom is brainstorming his vision of the future system EuReCuS uses this information to identify codified context knowledge which could support Tom in refining and identifying needs. The codified context knowledge is presented to Tom in the form of domain-specific questions together with possible answers. For example, analyzing Tom's description and using keyword matching our tool comes up with more detailed questions such as questions referring to the type of mobile device Tom is envisioning to use (e.g. Please specify the devices you prefer using). The system will provide possible answers, such as Laptop, Mobile Phone, and PDA which allows Tom to think about alternative options. Although he did not mention support for his Laptop in his initial description, he now might discover that he actually wants to use the envisioned system on his laptop as well.

Stimulating Tom's brainstorming by providing recognition cues in the form of questions in only one important aspect of EuReCuS. The questions represent codified context knowledge and include information about system variability. Tom is therefore able to customize existing software solutions with the help of codified context knowledge. This does not mean that Tom needs a technical background and has to understand the system architecture. Our solution is based on the idea that Tom is able to customize existing software solutions based on answering the provided questions. Depending on his individual needs Tom is able to select the answers reflecting what he expects from a future system. After answering relevant questions the gathered information is used to automatically generate a first prototype tailored to Tom's needs.

4 Related Work

Several attempts have been made in the past to introduce feature modeling as a means to involve end-users in service customization. For example, the authors in [5] classify web services features from the users' point of view and propose to use feature diagrams for modeling flexibility of the Web Services. In [6], authors introduce feature modeling and configuring techniques in domain engineering into service-oriented computing, and correspondingly propose a business-level service model and an end-user friendly service customization mechanism. The use of feature modeling is a promising way of customizing applications, however not very convenient for the end-users as they are strictly forced to think in terms of available features. Furthermore, end-users are not used to selecting features from a complex feature tree; it does not reflect their way of thinking and does not support creativity. Hartman et al. [7] have introduced the concept of a "Context Variability model", which contains the primary drivers for variation, e.g. different geographic regions. However, the motivation behind this research was not to support end-users during requirements elicitation. The context variability model constrains the feature model, which allows modeling multiple product lines supporting several dimensions in the context space.

5 Conclusions and Next Steps

Reuse-based system development requires changes in RE. Instead of being a front-end activity in the software engineering process and focusing on defining requirements for the development of software systems, the focus shifts towards mapping user needs to already existing reusable artifacts. This implies that knowledge about already available functionality is available and can guide requirements elicitation and system analysis. Product line variability models seem to be suitable for modeling and presenting contextual information to end-users. We foresee that making existing domain knowledge explicit might stimulate end-user's creativity and trigger new requirements.

Applying the proposed approach, in ideal cases, end-users will be able to construct tailored applications themselves (by utilizing domain-specific product generators and reusable artifacts documented in the product line model). However, even if no solution can be generated automatically, our approach allows an end-user to define a prototypic configuration of a service-based system. In addition the documented textual description of individual end-user needs is captured.

After realizing more sophisticated tool prototypes, we will explore the feasibility of the discussed approach by carrying out user-studies to measure and validate the effectiveness of the proposed techniques. Based on the feedback from users, we will work towards applying our approach in real-service repositories, empowering users to customize service-oriented applications themselves. This will provide us with further feedback and will allow improving the approach and the tools.

References

1. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State-of-the-art and Research Challenges. IEEE Computer, vol. 40 (11), 2007.
2. Seyff, N., Graf, F., Maiden, N.: Using Mobile RE Tools to Give End-Users Their Own Voice. In: International Conference on Requirements Engineering (RE 2010), Sydney, Australia, pp. 37–46, 2010.
3. Dhungana, D., Rabiser, R., Grünbacher, P., Lehner, K., Federspiel, C.: DOPLER: An Adaptable Tool Suite for Product Line Engineering. In: 11th International Software Product Line Conference (SPLC 2007), Kyoto, Japan, pp. 10-14, 2007.
4. Dhungana, D., Grünbacher, P., Rabiser, R.: Domain-specific adaptations of product line variability modeling. In: IFIP WG 8.1 Working Conference on Situational Method Engineering: Fundamentals and Experiences, Geneva, Switzerland, 2007.
5. Robak, S., Franczyk, B.: Modeling Web Services Variability with Feature Diagrams. Web, Web-Services, and Database Systems, Volume 2593/2009.
6. Wang, J., Yu, J.: A Business-Level Service Model Supporting End-User Customization. In: Service-Oriented Computing - ICSOC 2007 Workshops, Vienna, Austria, pp. 295-303, 2007.
7. Hartmann, H., Trew, T.: Using Feature Diagrams with Context Variability to Model Multiple Product Lines for Software Supply Chains. In: 12th International Software Product Line Conference, Limerick, Ireland, pp. 12-21, 2008..