

Intelligent requirement analyzis and detection for product development

Akram Chamakh, Dieter Roller

Institute of Computer-aided Product Development Systems
Universität Stuttgart
Universitätsstr. 38
D-70569 Stuttgart

Akram.Chamakh@informatik.uni-stuttgart.de

Dieter.Roller@informatik.uni-stuttgart.de

Abstract: The main goal of this paper is to present a software solution to analyze and to detect textual based requirements in the first stage of product development. After detecting the orders and requirements, the algorithm is supposed to generate automatically the related codes used for the identical requirements. This code will be used later in a related software or machine.

1 Introduction

Requirements management is gaining importance especially in design of complex products and systems. The subdivisions of requirements management are integrated from documentation processes, analyzis, tracking, prioritization and coordination of the requirements.

2 The proposed approaches

The textual based requirements are semantically detected, analyzed and categorized. The goal of syntactic and semantic analysis is to decompose a given text using the methods of computational linguistics in the smallest units (morphs) and then to provide them with both syntactic and semantic annotations [Ab83] [NE08]. In the following sections two different models to translate the requirements into formal models are proposed and explained.

2.1 Analysis and Evaluation of Requirements Documentation

In this model, the specified textual requirements are semantically analyzed and categorized into appropriate collections, as shown in Figure 1. The application domain at

this point plays essential role to connect the concepts with convertible information structure. In this context, a novel system was designed to analyze and evaluate the biased requirements. Consequently, the analyzed requirements are tested by a Requirements Extractor and Classifier system (REC) ¹ whether they are classified correctly as a requirements or not. To accomplish this task, they are divided into classes using ontology [Sta]. These classes comprise weak requirements, optional requirements, candidate requirements and no requirements.

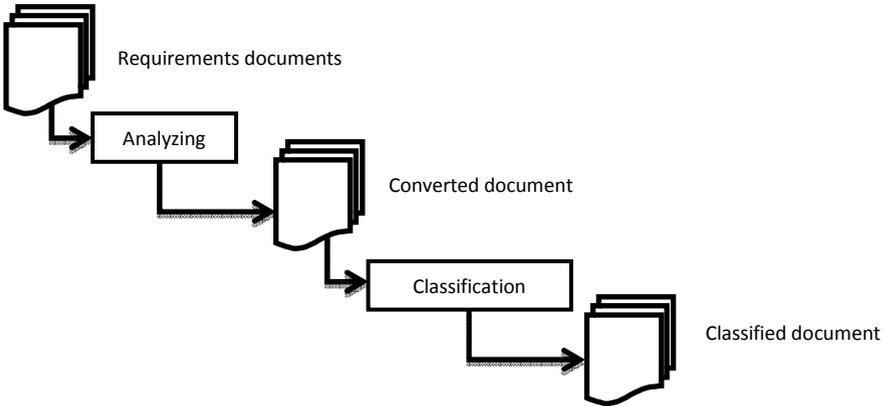


Figure 1: Workflow and technical steps of converting requirements documents into classified documents

The application domain typically provides very specific requirements of the technical system to be used for simulation or management of domain-specific tasks and problems [FIC08]. The arising requirements in this phase are incorporated particularly in the requirements analysis [Ge10]. The requirement analysis proceeds to a system development consequently to the modeling and to the generate the model for the realization. For further processing, the information obtained in the analysis phase is converted in a semantic model, which leads to an early detection of the requirements in the software-converting system [La10]. The static requirements analysis in the development time period and after that the dynamic requirements are especially time dependence. An example of a dynamic product requirement analysis is "to choose the most efficient solution". When the context dependence is rated high, the analysis will be more complex as it refers to specific needs and effective factors [RAC11].

A textual example for a context sensitive requirement is the following sentence: "the performance improvement of the previous versions". Considering the explicit and the implicit requirements, it becomes even more complex since the requirements are incorporated, in both current and other contexts. An implicit textual example of the requirements is presented by the following expression: "the system should be implemented with the latest object-oriented technology".

¹ REC uses the Stanford CoreNLP for all Natural Language Processing (NLP) analysis.

2.2 Detection and regenerating of requirements using a product configurator

Using this model, the requirements analysis is performed for some other applications as well as for the product configurator. Presently, a configurator is one of the company's main and highly reliable software systems. Within the globalization scope, it is vital for almost all companies to be able to present their own individual product design of the national as well as the international market. In this regard, a product configurator is created to optimize and to extend the classical arrangement of functionalities of the configurator [EJO13]. The aim has been so defined to understand the requirements of the user in a more efficient way. Besides reaching the targeted solution, another aim is to fulfill and to recommend the complete requirements. In addition, the market feedback and the user satisfaction have crucial importance in this approach.

3 Conclusion

The requirement analysis model should be created using intelligent procedure, like the detected structures, as a fundamental model or regenerating of requirements using a product configurator. Subsequently, this model is converted into the source code and is supplemented through adjustments of already available modules and is expanded into a completed software solution.

References

- [Ab83] R. J. Abbott: Program Design by Informal English Descriptions. In: Communications of the ACM 26 (1983), November, Nr. 11, S. 882–894. DOI 10.1145/182.358441. – ISSN 0001–0782.
- [FIC08] S. A. Fahmi, A. Ibrahim, H.-J. Choi. An Approach to Support and Partially Automate Requirements Engineering Activities. 8th International Conference on Computer and Information Technology Workshops, 2008.
- [Sta] Stanford Natural Language Processing Group. The Stanford NLP Group. URL <http://nlp.stanford.edu/>.
- [La10] M. Landhaeusser. Automatische Auszeichnung von Semantik in natuerlichsprachlichen Spezifikationen. Diplomarbeit, Karlsruher Institut fuer Technologie, 2010.
- [Ge10] T. Gelhausen, Modelleextraktion aus natuerlichen Sprachen – Eine Methode zur systematischen Erstellung von Domänenmodellen, KIT Scientific Publishing, ISBN 978-3-86644-547-5, Karlsruhe, 2010.
- [NE08] N. Niu, S. Easterbrook. Extracting and Modeling Product Line Functional Requirements. 16th IEEE International Requirements Engineering Conference, 2008.
- [RAC11] R. Rauf, M. Antkiewicz, K. Czarnecki. Logical Structure Extraction from Software Requirements Documents. 19th International Requirements Engineering, Conference, 2011.
- [EJO13] B. Erkus, D. Jakupovic, K. Oueslati. Configurator for project requirement, ProjINF, University of Stuttgart, 2013.