How to Deal with Inaccurate Service Requirements? Insights in Our Current Approach and New Ideas

Frederik S. Bäumer
University of Paderborn
frederik.baeumer@upb.de

Michaela Geierhos
University of Paderborn
michaela.geierhos@upb.de

Abstract

The main idea in On-The-Fly Computing is to automatically compose existing software services according to the wishes of end-users. However, since user requirements are often ambiguous, vague and incomplete, the selection and composition of suitable software services is a challenging task. In this report paper, we present our current approach to improve requirement descriptions before they are used for software composition. This procedure is fully automated, but also has limitations, for example, if necessary information is missing. In addition, and in response to the limitations, we provide insights into our above-mentioned current work that combines the existing optimization approach with a chatbot solution.

1 Motivation

Software requirements are challenging from the user perspective because they often allow a high degree of freedom in project implementation due to inaccuracies. In addition, the requirements are formulated in natural language (NL) and are therefore often ambiguous and partially incomplete. In project management this challenge is faced by a kind of interactive consolidation process in order to develop a common understanding of the requirements. However, such a consolidation process is not always foreseen or possible, as for instance in On-The-Fly Computing[1] (OTF Computing). Here, the individual software requirements of end-users have to be considered by the automatic composition of individual software services. The common developer role is no longer needed, while the need for a consolidation process still exists [Bäu17].

This is where our work picks up, dealing with the inaccuracies of freely formulated software requirements. To do this, we developed CORDULA (Compensation of Requirement Descriptions Using Linguistic Analysis), a system that recognizes and automatically compensates language inaccuracies (e.g., ambiguity, vagueness and incompleteness) in requirements written by end-users (cf. Figure 1). CORDULA supports the search for suitable software services that can be combined by transferring requirements descriptions into (canonical) core functionalities. In the future, end-users will be supported by CORDULA when specifying their software requirements. By using compensation strategies, the system corrects language shortcomings in requirements descriptions and transforms user wishes identified therein into corresponding software functionalities. In Section 3, we give insight into our current approach.

However, a far-reaching automation of the NL compensation also has weaknesses that we want to pick up in our future work. For example, it is very difficult to compensate missing non-standard information in case of incompleteness. Of course, it is possible to use default values or historical values here [GB16]. However, this is

[1]See `https://sfb901.upb.de` for more information about OTF Computing.

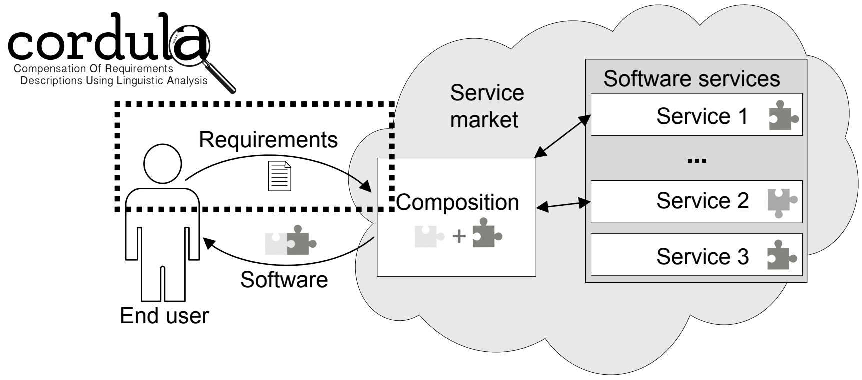

Figure 1: NL compensation as the first processing step [Bäu17]

not always expedient and, above all, not always what the user originally wanted. For this reason, we think that minimal communication between the compensation system and the user is required, even if this means slowing down the creation of software compositions. We also believe that this can be realized through chatbot technology. In this case, the challenge is to keep the conversation with the user so slim and meaningful that it will not be annoying for end-users. Furthermore, untrained users must have all the information needed for a decision (e.g., description of the problem and examples). This is part of our current and future research activity (see Section 4).

2 Project and Team Overview

The development of techniques and processes for the automatic on-the-fly configuration and provision of individual IT services is the goal of the Collaborative Research Centre 901 "On-The-Fly Computing" at the University of Paderborn, Germany. In particular, subproject B1 "Parameterized Service Specifications" deals with the efficient processing of different types of requirements specifications, which enable the successful search, composition and analysis of services. In the very first phase (until June, 2015), the focus of the project was on developing a specification language for services in order to automatically and efficiently process them. In the second phase (until June, 2019), user-friendly specifications are introduced. These are intuitively understandable specifications and are used primarily by end-users and domain experts. However, these requirements descriptions have errors that need to be corrected using Natural Language Processing (NLP). This is the task of the NLP team in subproject B1, which consists of Prof. Dr. Michaela Geierhos and Dr. Frederik Bäumer. Prof. Geierhos is a full professor for Digital Humanities at the University of Paderborn, Germany and Dr. Bäumer works as a postdoctoral researcher at the same chair.

Michaela Geierhos. Before Prof. Dr. Geierhos became a professor for Digital Humanities, she was an assistant professor for Semantic Information Processing at the University of Paderborn and worked as a postdoc at the Ludwig-Maximilians-University in Munich. After completing her studies in computational linguistics, computer science and phonetics, she worked as a research associate at the Center for Information and Speech Processing from 2006 to 2012. In 2010 she obtained a doctoral degree in computer linguistics on "BiographIE - Classification and Extraction of Career-Specific Information".

Frederik Bäumer. Dr. Frederik Bäumer has been working as a research associate at the Semantic Information Processing group in Paderborn since completing his master's degree in "Management Information Systems" with a focus on Semantic Information Processing. Since July 2013, he has been involved in various research projects at the Heinz Nixdorf Institute – initially as a master's student and then as doctoral student – and is an expert for search engine technology and information extraction. While participating in the Collaborative Research Center 901 "On-the-Fly Computing", he has been working on his doctorate since October 2014. Dr. Bäumer successfully defended his PhD thesis in July 2017 on "Indicator-based Detection and Compensation of Inaccurate and Incompletely Described Software Requirements".

3 Past Research on NLP for Requirement Engineering

At the beginning of our work, we dealt in particular with the question of the characteristics of NL requirements descriptions [GSB15]. It quickly became clear that a major challenge is off-topic information, incompleteness and ambiguity – problems that have already been subject of research in this area for a long time. For the automatic composition of software services, it is particularly crucial because in OTF Computing, it is better assumed that high-quality, formal specifications are provided as input, which are qualitatively better than NL requirements. There was, at this point, no automated procedure for checking and compensating requirements.

As a first challenge, we understood how to separate relevant information from irrelevant and how to identify the core components of requirements [BDG17, DG16]. After some attempts with rule-based approaches, we decided to go for a machine-learning approach because the text quality of user-generated requirements is very poor and we achieved only a bad recall through using rule-based approaches [DG16]. We then focused on the language deficits mentioned before. They are difficult to recognize and to compensate for end-users and sometimes cause great damage in the composition process [Bäu17]. However, each language deficit has different characteristics and must be recognized and compensated differently. In the case of incompleteness, we focused on the predicate-argument structure in requirements descriptions and tried to identify missing arguments and thus incompleteness [BG16]. For this procedure, extensive resources are necessary, which did not exist so far and were created by ourselves. Since there are hardly any real-life requirements to find, we have instead used software descriptions and feature requests from app stores (e.g., `download.com`) and developer forums (e.g., `sourceforge.com`) [BDG17, DG16].

There is a lot of work done on ambiguity in requirement descriptions. We distinguish between lexical, syntactic and referential ambiguity. All forms require different detection and compensation methods [Bäu17, GB17, GSB15]. Even more difficult to detect is vagueness. Here we have developed a rule-based approach that can recognize some forms of vagueness, but not all [GB17]. Since the compensation of vagueness is more difficult because of their different characteristics compared with ambiguities, we introduce a dialogue with the user (cf. Section 4). Based on this research and the realization that linguistic deficits are so different, we focused on a strategic approach for the optimization of NL: The goal was to develop a parameterized model (so-called strategy configuration) that automatically selects the best fitting strategy to compensate deficiencies in software specifications [Bäu17]. In order to enable this, suitable methods for the recognition and resolution of lexical, syntactic, and referential ambiguity and for the completion of NL requirements were first identified in literature and then implemented. This is the starting point of the indicator-based strategies (cf. Figure 2).

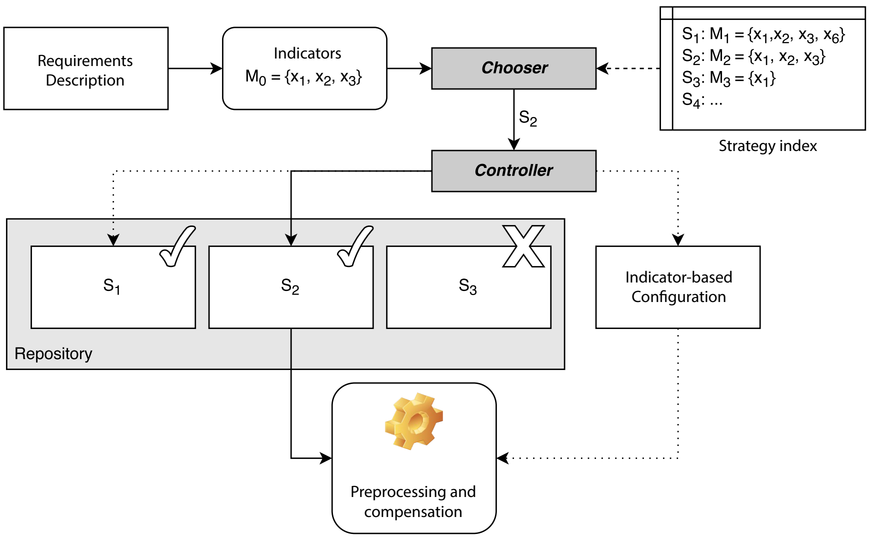

Figure 2: Basic idea of CORDULA [Bäu17]

These strategies enable CORDULA to execute and monitor individual methods, to compare results and to correct them if necessary as well as to use synergy effects. For this purpose, it was necessary to define context-sensitive indicators to determine the compensation need, which can trigger individual strategies. Employing these strategies, CORDULA is able to apply existing, highly heterogeneous processing and compensation techniques on requirements descriptions. It is only possible by means of the linguistic indicators to capture and optimize the individual text quality in a data-driven manner, as required, because CORDULA's compensation pipeline is configured on the basis of the detected shortcomings. In addition, the individual results of the components are merged and structured as a consistent compensation result.

4 Research Plan on NLP for Requirement Engineering

As mentioned before, language deficits in NL requirements can only be compensated automatically in parts. A complete compensation without any questions to the users is in many cases not possible, especially if vagueness and incompleteness are present. For this reason, we are currently working to extend CORDULA with a chat functionality. The users can submit their requirements to CORDULA via chat (left part of Figure 3) and receive an overview of the processed requirements as return (right part of Figure 3). If CORDULA is able to detect but not to compensate a problem, the system automatically asks the user to provide some helpful information or to make a decision (e.g., annotate (a part of) a sentence as an off-topic, as shown in Figure 3).

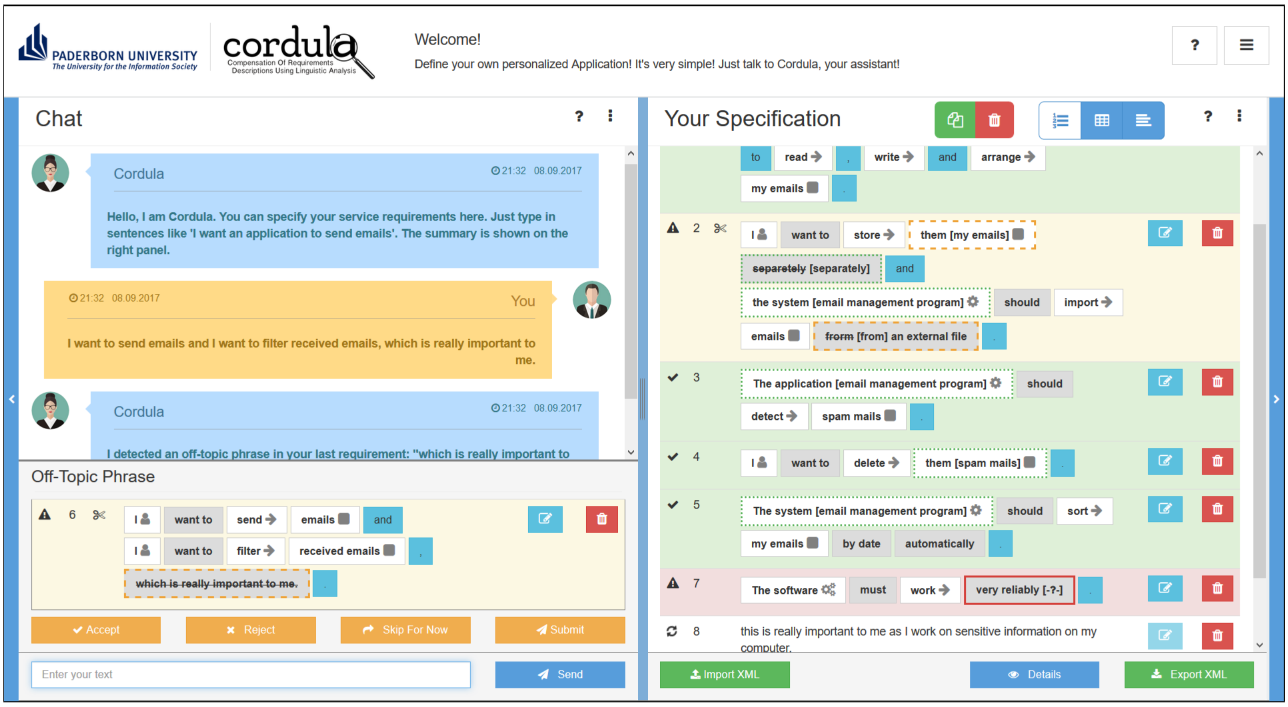

Figure 3: Expand CORDULA with a chatbot feature

First attempts with existing chatbot frameworks such as Google's api.ai or IBM Watson Conversations worked well and brought up interesting findings. They pointed out that we need more flexibility and control in the conversation and less free communication (which the framework aims for). The dialogue should not be conducted completely by the chatbot, but should address a specific problem and provide targeted information for CORDULA. For this reason, in our current vision we use our own chat solution, which can ask predefined questions to the user about predefined problems. In this way, we expect to get further relevant user input for a specific problem just-in-time. One criticism of this bidirectional chat approach remains, however: chatting slows down the entire OTF service composition process. Nevertheless, an improvement in the overall result can be expected, since the requirements will be more concrete and complete. This trade off needs to be discussed in future work.

Acknowledgements

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing" (SFB 901).

References

[Bäu17] Frederik Simon Bäumer. *Indikatorbasierte Erkennung und Kompensation von ungenauen und unvollständig beschriebenen Softwareanforderungen.* Phd thesis, University of Paderborn, Paderborn, Germany, July 2017. ISBN: 978-3-942647-91-5.

[BDG17] Frederik Simon Bäumer, Markus Dollmann, and Michaela Geierhos. Studying software descriptions in sourceforge and app stores for a better understanding of real-life requirements. In Federica Sarro, Emad Shihab, Meiyappan Nagappan, Marie Christin Platenius, and Daniel Kaimann, editors, *Proceedings of the 2nd ACM SIGSOFT International Workshop on App Market Analytics*, pages 19–25, Paderborn, Germany, September 2017. ACM.

[BG16] Frederik Simon Bäumer and Michaela Geierhos. Running out of words: How similar user stories can help to elaborate individual natural language requirement descriptions. In Giedre Dregvaite and Robertas Damasevicius, editors, *Proceedings of the 22nd Conference on Information and Software Technologies (ICIST 2016)*, volume 639 of *Communications in Computer and Information Science*, chapter Running Out of Words: How Similar User Stories Can Help to Elaborate Individual Natural Language Requirement Descriptions, pages 549–558. Springer International Publishing, Druskininkai, Lithuania, October 2016. ISBN: 978-3-319-46254-7.

[DG16] Markus Dollmann and Michaela Geierhos. On- and off-topic classification and semantic annotation of user-generated software requirements. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1807–1816, Austin, TX, USA, 1 - 5 November 2016.

[GB16] Michaela Geierhos and Frederik Simon Bäumer. How to complete customer requirements using concept expansion for requirement refinement. In *Proceedings of the 21st International Conference on Applications of Natural Language to Information Systems (NLDB 2016)*, volume 9612, pages 37–47, Salford, UK, 22 - 24 June 2016. Springer.

[GB17] Michaela Geierhos and Frederik Simon Baeumer. Guesswork? resolving vagueness in user-generated software requirements. In Henning Christiansen, M. Dolores Jimenez Lopez, Roussanka Loukanova, and Lawrence S. Moss, editors, *Partiality and Underspecification in Information, Languages, and Knowledge*, Partiality and Underspecification in Information, Languages, and Knowledge, chapter 3, pages 65–108. Cambridge Scholars Publishing, September 2017.

[GSB15] Michaela Geierhos, Sabine Schulze, and Frederik Simon Bäumer. What did you mean? facing the challenges of user-generated software requirements. In *Proceedings of the 7th International Conference on Agents and Artificial Intelligence*, pages 277–283, 10 - 12 January 2015. Lisbon, Portugal. ISBN: 978-989-758-073-4.