Automating the Detection of Complex Semantic Conflicts between Software Requirements

An empirical study on requirements conflict analysis with semantic technology

Thomas Moser, Dietmar Winkler, Matthias Heindl and Stefan Biffl
Christian Doppler Laboratory “Software Engineering Integration for Flexible Automation Systems”
Vienna University of Technology, Vienna, Austria
{firstname.lastname}@tuwien.ac.at

Abstract—Keeping requirements consistent already at early project stages is a main success factor for software development projects. However, manual requirements conflict analysis takes significant effort and is error-prone. Requirement engineers and other project participants such as technical architects use different terminologies (due to different domain backgrounds) which makes automation of conflict analysis difficult. In this paper we propose semantic approach as foundation for automating requirements conflict analysis and introduce the automated ontology-based reporting approach OntRep. We evaluate the effectiveness of OntRep referring to (a) different types of conflicts and (b) different levels of conflict complexity in a real-world industrial case study. Major results were that OntRep had considerably higher recall and precision of conflict detection for all conflict types compared to a manual approach. Regarding complexity, the comparison with manual results shows that recall and precision of OntRep is considerably higher for complex conflicts.

Keywords—requirements conflict detection, requirements consistency checking, ontology, case study, empirical evaluation.

I. INTRODUCTION

Modern software and systems engineering projects are complex due to (a) the high number and complexity of requirements (e.g., mutual dependent or contradicting requirements), and (b) geographically distributed project stakeholders with different backgrounds and domain terminologies. A major goal of requirements engineering is to achieve consistent requirements descriptions in order to create a common and agreed understanding on the set of requirements between all project stakeholders. Initial observations at our industry partners showed that establishing consistency of requirements takes significant effort and is error-prone when performed manually. Thus, automation approaches for requirements conflict analysis are needed that (a) increase the effectiveness and quality of analyzing requirements from different stakeholders for symptoms of inconsistency, e.g., contradicting requirements, and (b) reduce the conflict analysis effort. Unfortunately, current automation approaches for conflict analysis suffer from the following challenges and limitations:


- Focus on executable code. Some approaches need executable code to identify requirements conflicts [2], but in early stages - where conflict analysis is important - executable code may not available.

- Limitation to syntactical comparison. Potential conflicts are identified by using trace dependencies between requirements that have been explicitly captured before, e.g. by information retrieval approaches like “keyword matching” [6][7]. These approaches allow syntactical comparison of requirements, but do not cover semantics, even if different terms are used in these requirements definitions.

Thus, using semantic technologies seems to be a promising approach to address these challenges: ontologies provide the means for describing the concepts of a domain and the relationships between these concepts in a way that allows automated reasoning [10]. In this paper, we propose OntRep [4], an automated ontology-based reporting approach for the analysis of complex semantic conflicts between requirements based on ontologies and reasoning mechanisms. The main criteria for the evaluation of OntRep are: correctness and completeness of identified requirements conflicts and the effort to develop a project or domain ontology. OntRep aims at lowering the effort for requirements conflict analysis, while keeping requirements consistency high. Initially, OntRep automatically categorizes requirements into a given set of categories using ontology classes modeled in Protégé and mapping the terms used in the requirements to these classes. Using this foundation, OntRep analyzes the content of the requirements and identifies conflicts between requirements. Therefore, conflict analysis is not only based on traditional keyword-matching-approaches, but also considers different terminologies used for describing semantically equivalent concepts. We empirically evaluate OntRep [4] in a real-life project at our industry partner to investigate performance and quality with respect to (a) different types of conflicts and (b) conflicts on different levels of complexity.

The remainder of the paper is organized as follows: Section 2 summarizes related work on requirements conflict analysis and natural language processing; Section 3 introduces the OntRep approach and motivates research issues. Section 4 describes the performed empirical study. Section 5 presents the results which are discussed in Section 6. Finally, Section 7 concludes and suggests further work.
II. RELATED WORK

This section presents related work on requirements conflict analysis and natural language processing [4].

A. Requirements Conflict Analysis

“Requirements conflict with each other if they make contradicting statements about common software attributes […] Given that there may be up to \( n^2 \) conflicts among \( n \) requirements […], the number of potential conflicts, […], could be enormous, burdening the engineer with the time-intensive and error-prone task of identifying the true conflicts” [2].

The Trace Analyzer by Eyged and Grünbacher [2] analyzes the footprints of test cases to detect requirements conflicts. If two requirements execute overlapping lines of code, a potential conflict may exist. A prerequisite for the Trace Analyzer is to have executable code, which is often not available in early project phases, when conflict analysis is a major goal. Heitmeyer et al. [5] describe a formal analysis technique, called consistency checking, for the automated detection of syntactic errors, such as type errors, non-determinism, missing cases, and circular definitions, in requirements specifications. The approach does not find semantic conflicts.

Information retrieval approaches [6], such as the RETH approach [7] use keyword-matching techniques to identify general requirements interdependencies. These captured interdependencies can be used to identify requirements conflicts. However, these techniques do not allow identifying conflicts or other interdependencies between requirements, if they use different terms for similar concepts. Thus, these approaches are less effective in practice, because they cannot identify the full set of interdependencies between requirements. The extended Backus-Naur-Form (EBNF) [12] is a syntax for requirements, which is used to improve the understandability of requirements for humans and machines.

B. Natural Language Processing

Natural language processing (NLP) techniques are useful to extract structure and content of requirements given in natural language for transformation into the structure of an ontology. NLP generally refers to a range of computational techniques for analyzing and representing naturally occurring texts [1]. The core purpose of NLP techniques is to achieve human-like language processing for a range of tasks or applications [8].

Most important NLP models used in this research are part-of-speech (POS) tagging and sentence parsers [1]. POS tagging involves marking up the words in a text as corresponding to a particular part of speech, based on both its definition, as well as its context. In addition, sentence parsers transform text into a data structure, which provides insight into the grammatical structure and implied hierarchy of the input text [1]. Stanford parser/tagger\(^1\) and OpenNLP\(^2\) are the core set of NLP tools used in this paper. Furthermore, we use WordNet, a large lexical database in English [9]. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet is a useful building block for requirements analysis. These NLP technologies can be used for our purpose, namely to improve the effectiveness of requirements management activities like categorization, conflict analysis, and tracing.

III. ANALYSIS OF CONFLICTS WITH AN ONTOLOGY

Due to the limitations of requirements analysis approaches that address only links between requirements based on syntactic equality, we explore an approach based on semantic equality: OntRep [4] links similar concepts sharing the same meaning even if their syntactic representations are different. As ontologies are versatile for representing knowledge on requirements and for deriving new links between requirements, we introduce an ontology-based approach for reporting analysis results on a set of requirements. The goal of the ontology-based reporting approach OntRep is making requirements management tasks such as requirements conflict analysis more effective based on the automation of selected steps in these tasks. We focus on complex semantic, which we define as conflicts regarding a set of more than two requirements and constraints, which based on observations at our industry partners are hard to identify manually. In the following subsections, we provide an overview on the approach and motivate research issues.

A. Basis for Requirements Conflict Analysis

For formally specified requirement semantics, in our case following an EBNF template (see Figure 1), semantic analysis can identify inconsistencies and conflicts using a set of assertions that should hold true for all available facts. These assertions are based on available requirements, while available facts are based on the environment and properties of the target system. Typically, requirements following this EBNF template specify under which conditions (under condition) a target system should provide a certain functionality (process) regarding a specific object (thing to be processed), e.g., the system configuration, to a certain entity or role (somebody or something), e.g., to an user or to the administrator. In this paper, we focus on (a) functional requirements (following an EBNF template) and (b) a set of requirements constraints regarding technical, requirement-specific or documentary issues.

We derive three types of conflicts, which can be detected with OntRep:

\(^1\) http://nlp.stanford.edu/software/lex-parser.shtml
\(^2\) http://opennlp.sourceforge.net/
- Not well-formed requirements (conflicts between requirements and EBNF grammar).
- Conflicts between functional requirements and technical and/or requirement constraints.
- Conflicts between two functional requirements.

B. Automated Conflict Analysis

We developed a prototype tool for the OntRep approach as a plug-in to Trac\(^3\), which can be extended by Python plug-ins.

Phase A: Linking of natural language to semantic concepts.

In a first phase, natural language texts (technical constraints, requirements constraints, documentation guidelines and glossary knowledge) have to be linked to semantic concepts as preparation for further analysis and reporting [4]. The following 8 steps are used to build this project knowledge:

1) Define project-specific concepts in Protégé. Each concept is defined as an ontology class in Protégé. It is important to define project-relevant “semantic” concepts and not formal ones in order to enable the automated assignment, e.g., “Security”. Typically, these concepts can be defined based on a project glossary that contains important project-specific terms.

2) Provide input data to be categorized: Constraints, guidelines and glossary knowledge are typically represented as natural language text. For our research we export these natural language texts and import them into OntRep prototype tool.

3) Remove irrelevant stop-words, like “and”, “any”, “but”, which cannot be used for assignment. This step is performed automatically using a standard stop-word list\(^4\).

4) Bring all remaining words into their root form (e.g., “jumping” to “jump”): this process is called “stemming” based on algorithms like “Porter Stemmer” [11].

5) Get all synonyms and hyponyms of the analyzed words in the requirements by using the natural language processing library “WordNet” [9]. For example, “house” is a synonym for “building”, “dog” is a hyponym of “animal”.

6) Heuristic-based assignment of each requirement to the defined concepts depending on the number of hits for (a) synonyms, (b) hyponyms, and (c) substring matches. The heuristic checks if the hits for synonym, hyponym and substring matches meet the given threshold values. If this number is equal or higher than the number of thresholds that must be met, the word will be related to that concept, otherwise not. If several concepts reach these thresholds, the requirement will be assigned to all of these concepts.

7) Save the element as an individual of the ontology class, if it is not already in the class. This can only be checked if one or more of the elements attributes have been declared as primary keys (uniquely identifying the element). If the element has already been saved in another class as well (which could be the case), declare that the new element is the same as the already existing one with the “owl:sameAs” property.

8) Manually check the validity and correctness of the imported fact, both regarding its assignment to the right concepts as well as regarding its meaning.

Phase B: Mapping of Requirements and Semantic Concepts.

In the second phase, analysis and reporting approaches build on the mapping of requirements to semantic concepts. For formally specified requirement semantics, in our case following an EBNF template (see Figure 1), semantic analysis can identify inconsistencies and conflicts using a set of assertions that should hold true for all available facts. These assertions are based on the available requirements, while the available facts are based on the environment and properties of the target system (which were imported in the first phase). The following 4 steps describe the semantic requirements conflict analysis:

1) Import of requirements. The requirements are represented as tickets in Trac. For our research prototype we export these requirements via CSV from Trac and import them into the OntRep prototype tool.

2) Parsing of requirements. If the requirements are formally described using a specified grammar (e.g., EBNF), the information contained in the textual requirement descriptions can be semantically analyzed in order to identify possible inconsistencies and/or conflicts. Based on the specified grammar, certain parts of the requirements are extracted for further usage. In our case, this primarily affects the “thing to be processed” and the “obligation” (“shall” or “shall not”) specified in EBNF.

3) Linking of requirements to semantic concepts. The “things to be processed” of the specific requirements, which were extracted in the previous step, are linked to semantic concepts modeled in the first phase of the OntRep approach. If there is no semantic concept available for a specific “thing to be processed”, a new semantic concept is created, in order to later identify conflicts between two functional requirements. Additionally, the “obligation” for each requirement is stored to enable requirement conflict detection.

4) Requirement conflict detection. After all requirements have been successfully parsed and linked to their related semantic concepts, every semantic concept enables checking for consistency, validity, and correctness of the related requirements using ontology-based reasoning. From the point of view of semantic analysis, we focus on two different kinds of conflicts in this paper, namely on logical inconsistencies between facts, as well as numerical discrepancies of facts. In the following, we present an example for these kinds of conflicts.

![Figure 2. Logical requirement conflicts.](http://www.textfixer.com/resources/common-english-words.txt)
semantic concept “User” to the semantic concept “Configuration page”. Additionally, there exists a requirement constraint specifying that no links may exist between the semantic concepts “Secured Resource” and “Untrusted Person”. In the project glossary the semantic concept “Configuration Page” is identified as sub-concept of the “Secure Resource” concept, and the semantic concept “User” as sub-concept of the “Untrusted Person” concept since it is not a sub-concept of the “Trusted Person” concept which is defined as the negation of the “Untrusted Person” concept. Using these facts, the OntRep approach successfully identifies a logical conflict between RQ-11 and RC-1, relying on the facts specified in GL-1 and GL-2.

Figure 3 depicts an example for conflicts between requirements as well as for numerical discrepancies between requirements and constraints. There are three requirements which are linked to the semantic concept “Notification”, RQ-12, RQ-17, and RQ-21. RQ-12 and RQ-17 are additionally linked to the semantic concept of “Messages per Second”, but with a different parameter value, therefore the OntRep approach successfully identifies a conflict between these two requirements. Additionally, the semantic concept of “SSL Encryption”, which is defined, specifying a link to the semantic concept “Messages per Second” with a parameter value of 3. While this holds true for RQ-17, RQ-12 requires at least a parameter value of 4 for the “Messages per Second” semantic concept; therefore the OntRep approach detects another conflict between RQ-12, RQ-21, and TC-1.

C. Research Issues

The underlying idea of this research is to use advanced semantic technologies, like ontologies and reasoning mechanisms, to increase the effectiveness of the analysis of complex requirements conflicts. Based on our findings in [4], the main research questions of this paper are:

- **RQ1** How effective is OntRep in finding different types of conflicts compared to manual analysis by experts? We mentioned 2 kinds of conflicts in section 3.B: numeric discrepancies and logical requirements conflicts. These conflicts may occur between different elements: (a) between requirements, (b) between a requirement and a constraint, and (c) between a requirement and a formal guideline. For these 3 types of typical conflicts we analyze the correctness and completeness of conflicts found with OntRep and compare the results to a manual approach for each conflict type.

- **RQ2** How effective is OntRep to identify conflicts of different degrees of complexity compared to manual analyses by experts? A conflict may consist of two or more elements, e.g., a conflict may exist between two or three requirements. We define these conflicts as “simple conflicts” because we assume they are easy to identify. On the other hand, we define conflicts consisting of more than 3 elements to be “complex conflicts”. These complex conflicts might be more difficult to identify completely, especially when performing conflict analysis manually. We assume that OntRep reduces complexity to identify conflicts compared to the manual approach. For each complexity level we analyze the correctness and completeness of each identified conflict.

Based on our previous experience with tool support for quality assurance [4], we assume that OntRep can help increasing effectiveness and quality of requirements conflict analysis. Additionally, we also assume that OntRep reduces the effort for requirements conflict analysis.

IV. STUDY DESCRIPTION

This section summarizes study settings, i.e., goals, design, process, material, and participants, and identifies a set of threats to validity.

**Goals.** The goal of the pilot study was to analyze the effectiveness and quality of the OntRep conflict analysis approach compared to traditional manual approaches with focus on different types and complexity levels of conflicts.

**Study Subject.** As described in [4] the case study project is a software development project at our industry partner with the goal to design and implement a web application that serves as a platform for communication and networking between technology experts. This type of project seemed well suitable to apply and evaluate the OntRep prototype, because OntRep has been implemented as Trac plug-in and was easy to integrate with the project’s tool infrastructure.

**Participants** were six project managers for individual conflict detection, randomly assigned to two teams for preparing an agreed team conflict list. One requirement engineering expert, i.e., one of the authors, provided control data for all tasks.

**Material.** We used a list of requirements (i.e., a spreadsheet) containing 23 functional requirements, 11 constraints (7 technical and 4 business constraints), and 4 formal documentation rules (documentation guidelines). To focus on individual conflict classes, i.e., conflict between requirements (CRR), conflict of a requirement with a constraint (CRC), conflict of a requirement with a formal guideline, i.e., ill-formed requirement (CRG), we seeded an overall number of 22 conflicts according to different conflict types and two complexity levels. Conflicts related to up to 3 elements are rated as simple, conflicts with more than 3 elements involved are considered as complex conflicts. In addition we used a data capturing sheet, guidelines to support the participants in identifying conflicts, and questionnaires to capture the background of participants prior to the study and feedback after afterwards.

**Variables & metrics.** Independent variables were the number of seeded conflicts, the defect types, and the number of
requirements per defect type. Dependent variables are the number of identified conflicts and for conflict identification. We used recall and precision as study metrics.

Finally, we analyzed and evaluated the following results: (a) 6 spreadsheets for conflict analysis from each of the 6 individual participants, (b) 2 team spreadsheets, (c) 1 conflict analysis spreadsheet from a requirement engineering expert, and finally (d) 1 conflict analysis spreadsheet created with the OntRep approach. The results were evaluated with descriptive statistics in Excel and R and are described in the following section. We applied the Mann-Whitney-Test at a significance level of 95% (2-sided) for statistical testing.

**Threats to Validity.** Following the standard practice of empirical software engineering [13], we identified a set of internal and external threats to validity. We addressed *internal threats to validity* [3] of the study by two measures: a) intensive reviews of the study concept and materials, and b) a test run of the study conducted by a test person in order to make sure that the guidelines, explanations, and task descriptions are understandable for the participants and to estimate the required effort/time frame. Regarding *external validity*, we performed this initial case study in a professional context at a software development company. The participants had medium requirements management know-how and advanced software engineering know-how (captured by background questionnaire prior to the study). In addition, we had an expert in Requirements Engineering as control group. Nevertheless, the small number of participants might hinder the generalization of results. Therefore, we suggest replicating the study in a larger context in future work.

V. Results

The following subsections describe the results of requirements conflict analysis with OntRep.

A. Conflict analysis results by conflict types

The first research question was how effective OntRep is in finding different types of conflicts compared to manual analysis? We defined 3 types of typical conflicts: Conflicts between a requirement and a constraint (CRC), e.g., when a particular requirement requires a response time that is not feasible with the chosen technology (the chosen technology constrains response times); Conflicts between a requirement and a documentation guideline (CRG), e.g. a requirement may conflict with the documentation guideline “all requirements must use the obligation word ‘shall’”; Conflicts between requirements (CRR), e.g. the requirements “The system shall update the index at least 20 times per hour” and “The system shall update the index at least 20 times per hour” results in a conflict.

The individual recall and precision results for detecting CRC conflicts manually are low: approx. one third of existing conflicts has been identified and only one third of found conflicts were identified correctly. Group harmonization was quite effective for this type of conflict, because both recall and precision were improved, i.e., the conflicts found by each individual in the group have been merged in a discussion session, which results in a recall of approx. 60%, and some false positives were eliminated, which results in a precision of 50%. The expert performed better than the individuals regarding recall and precision, but the group results regarding recall were better. In this case the discussion of conflicts in the group was valuable.
Comparing the expert’s CRC recall value with the CRG and CRR results shows that CRC results are clearly lower than the others. This is probably due to the fact that CRC conflicts were complex only, and thus hard to find, whereas the CRG and CRR conflict bulks consisted of either simple only or both simple and complex.

Mann-Whitney Test at significance level 95% (2-side) did not show any significant differences (p-values >0.098(-)) regarding study groups, i.e., individuals, groups, experts, and OntRep and individual conflict classes, i.e., total, CRC, CRG, and CRR. The main reason is a low number of involved data sets. Note that expert (Exp) and OntRep include one data set and we had 2 groups and an overall number of 6 individuals.

The CRG results are similar to the CRC results: rather low recall and precision values of individual conflict analysis and an improvement by group harmonization. A slight difference is that the expert performed much better than both the individuals and groups. He reached pretty high recall and precision values (more than 80%). The CRG results of the expert are similar to his CRG results, but totally different regarding individual and group results: The individual results were rather good (40% recall and 56% precision), but then worsened by group harmonization (30% recall and 24% precision).

Table 1. Overall Recall and Precision per Conflict Type.

<table>
<thead>
<tr>
<th>Type</th>
<th>Total</th>
<th>CRC (f+p)</th>
<th>CRG (f+p)</th>
<th>CRR (f+p)</th>
<th>Partially</th>
<th>Fully</th>
<th>Not Found</th>
<th>False Pos.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indiv.</td>
<td>32%</td>
<td>46%</td>
<td>29%</td>
<td>31%</td>
<td>50%</td>
<td>66%</td>
<td>40%</td>
<td>56%</td>
</tr>
<tr>
<td>Group</td>
<td>48%</td>
<td>49%</td>
<td>57%</td>
<td>49%</td>
<td>50%</td>
<td>68%</td>
<td>30%</td>
<td>24%</td>
</tr>
<tr>
<td>Expert</td>
<td>68%</td>
<td>88%</td>
<td>43%</td>
<td>100%</td>
<td>80%</td>
<td>89%</td>
<td>80%</td>
<td>80%</td>
</tr>
<tr>
<td>OntRep</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 1 shows that OntRep has 100% values for both recall and precision for every conflict type. This means that OntRep found all of conflicts (7 CRC, 10 CRG, 5 CRR conflicts and no false positives). Although these results look like rigged for making OntRep look good, we think that the defined conflict types and the seeded conflicts are typical and realistic. Thus, the study is well-balanced and the comparison of OntRep with manual analysis results is valuable and meaningful.

B. Conflict analysis results by complexity

After analyzing the results regarding the different conflict types, we studied how effective OntRep is to identify conflicts of different degrees of complexity compared to a manual analysis. We defined two levels of complexity based on the number of elements involved in a conflict: simple and complex. The threshold number of elements to make a conflict complex is 3. We chose this threshold subjectively, because we think that analyzing 3 elements cognitively is still feasible, but gets more complicated and difficult to analyze conflicts with 4 or more involved elements. We seeded 12 simple and 10 complex conflicts (see Table 2).

In comparison to OntRep, the manual conflict analysis approach resulted in a lower completeness, no matter which type of conflict: the only conflict types for which the manual results come close to the OntRep results are the simple CRG and complex CRR conflicts; and there it is only the expert result that is similar to the OntRep result, the average results of individuals and groups are far away and false positives are introduced. They have been reduced slightly during group harmonization.

Table 2. Conflict identification per conflict type and complexity level.

<table>
<thead>
<tr>
<th>Type</th>
<th>Simple Conflicts (12)</th>
<th>Complex Conflicts (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ind. / Group / Expert / OntRep</td>
<td></td>
<td>Ind. / Group / Expert / OntRep</td>
</tr>
<tr>
<td>(Rec / Prec)</td>
<td>(Prec)</td>
<td>(Rec / Prec)</td>
</tr>
<tr>
<td>Fully (f)</td>
<td>0.8 (2.0)</td>
<td>2.0 (1.4)</td>
</tr>
<tr>
<td>Partially (p)</td>
<td>1.2 (1.6)</td>
<td>2.0 (0.0)</td>
</tr>
<tr>
<td>Recall (f+p)</td>
<td>29%</td>
<td>57%</td>
</tr>
<tr>
<td>False Pos.</td>
<td>3.2 (3.8)</td>
<td>0.5 (0.7)</td>
</tr>
<tr>
<td>Precision</td>
<td>63%</td>
<td>53%</td>
</tr>
<tr>
<td>Not Found</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Complex CRC conflicts seem to be very hard to identify manually: from 7 existing conflicts only 0.9 conflicts on average have been identified completely, 2 conflicts have been identified partially; i.e. less than 45% of conflicts have been identified manually. On the other hand, false positives have been introduced. We expected that the number of false positives is reduced during group harmonization, but the opposite was the case: the amount of false positives increased from average individual results to the avg. group results. This is due to the complexity of the conflicts and the discussion that was caused by it during group harmonization.

Comparing completeness of simple and complex conflicts shows that the differences between manual and OntRep results increase with the level of complexity: e.g. regarding complex CRC conflicts the difference between manual and automated results is higher than regarding simple CRG conflicts where 8 conflicts have been found manually (as optimum) and 10 conflicts have been found completely with the OntRep.
the given data were identified, due to the 3 reasons for 100% conflict identification described in section 5. We did not find any significant differences regarding the Mann-Whitney Test at a significance level of 95% (2-side). The necessary efforts for both the manual and automated approach are reported in [4].

VI. DISCUSSION

The results of conflict analysis with OntRep seem convincing for the conflict types. The automated conflict analysis depends on the following factors:

- **Requirements structure**: OntRep analyzes different sections of each requirement to map them to the underlying concepts and to identify conflicts. Thus, a certain structure (EBNF) is necessary. When documents are used, this syntax is not used so frequently, but the importance of such a grammar increases when requirements databases are used instead of documents, because requirements have to be understandable and clear even without the context that usually exists in a requirements document.

- **Completeness of glossary**: the detection of logical conflicts depends on the terms captured as glossary terms in the ontology. OntRep recognizes conflicts between requirements that use different terms only if these terms are appropriately defined and their dependencies are clear. Usually, there is a project glossary existing in a project that can be imported into the ontology without big effort, so that all relevant terms are available for analysis.

- **Quality of ontology**: another prerequisite is correct modeling of the ontology, which requires expert knowledge. If the user does not specify it correctly, the tool cannot find the conflicts. If a constraint is missing in the ontology, the tool does not find any conflicts linked to that constraint. If a constraint has been specified wrongly, the tool might find conflicts with that constraint it was not intended to be found by the user. So if all facts have been modeled correctly, the tool will find all corresponding conflicts.

Despite the given prerequisites and limitations of OntRep, we think that this study is relevant, because its focus was on (a) evaluating the general technical feasibility of applying semantic technology to automated requirements conflict analysis, and (b) compare the results with the results of a manual approach for a practical-relevant set of conflict types.

VII. CONCLUSION AND FURTHER WORK

Keeping requirements consistent is a main success factor for software development projects. That is why conflict analysis activities are important for requirements managers and project managers. However, the manual conduct of these activities takes significant effort and is error-prone, especially with an increasing number of requirements. Another issue is that participants from different domain backgrounds and terminologies have to work together in large distributed projects.

In this paper we proposed semantic technology as foundation for automating requirements conflict analysis and introduced the automated ontology-based reporting approach OntRep based on a project ontology and a reasoning mechanism. We used requirements formulated in EBNF as input to the proposed OntRep approach, which supports automated requirements conflict analysis. We evaluated the effectiveness of the OntRep conflict analysis approach referring to (a) different types of conflicts and (b) different levels of conflict complexity in a real-world industrial case study with 6 project managers in 2 teams. In addition a requirements expert and an OntRep user performed the same tasks to enable comparing the quality of results. Regarding the given conflict types the results of the evaluation are similar: OntRep found all conflicts in the requirements during the empirical study, while manual conflict analysis identified 30 to 80% of the conflicts for each conflict type and produced more false positives.

Further work will focus on the replication of this pilot study in a larger context. In addition, we want to increase the number of requirements and conflicts to be analyzed in order to get (a) more accurate numbers regarding recall and precision of the automated conflict analysis approach (b) more meaningful data regarding efforts of OntRep and manual conflict analysis approach for a higher number of requirements and conflicts.

ACKNOWLEDGMENT

We want to thank Alexander Wagner for the implementation of OntRep. This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria.

REFERENCES


