Automating the Detection of Complex Semantic Conflicts between Software Requirements with OntRep
An empirical study on requirements conflict analysis with semantic technology

Matthias Heindl
Siemens IT Solutions and Services
Gudrunstrasse 11, A-1100 Vienna, Austria
matthias.a.heindl@siemens.com

Thomas Moser, Dietmar Winkler, Stefan Biffl
Inst. f. Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstrasse 9-11/188, 1040 Vienna, Austria
<first_name>.<last_name>@tuwien.ac.at

ABSTRACT
Keeping requirements consistent is a main success factor for software development projects. However, manual requirements conflict analysis takes significant effort and is error-prone. Project participants use different terminologies (due to different domain backgrounds) which makes automation of conflict analysis difficult. In this paper we propose semantic technology as foundation for automating requirements conflict analysis and introduce the automated ontology-based reporting approach OntRep. 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 at Siemens Austria. Major results were that OntRep had considerably higher recall and precision of conflict detection for all given conflict types than a manual approach. Regarding complexity levels, the comparison with manual results shows that recall and precision of OntRep is slightly better for simple conflicts, but considerably higher for complex conflicts.

Categories and Subject Descriptors
D.2.9 [Software Engineering]: Management – software quality management (SQM)

General Terms
Management, Measurement, Documentation, Economics, Experimentation, Verification

Keywords
Requirements conflict analysis, consistency checking, ontology, case study, empirical evaluation

1. INTRODUCTION
Modern software and systems engineering projects are complex due to (a) the high number and complexity of 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.

Establishing consistency of requirements takes significant effort and is error-prone when performed manually. Thus, automation approaches for requirements conflict analysis are needed that (1) increase the effectiveness and quality of analyzing requirements from different stakeholders for symptoms of inconsistency, e.g., contradicting requirements, and (2) reduce the conflict analysis effort.

Unfortunately, current automation approaches for conflict analysis suffer from the following challenges and limitations:

- Some approaches need executable code to identify requirements conflicts [6], but in early stages - where conflict analysis is important - executable code may not available.
- 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” [12][13]. These approaches allow a syntactical comparison of requirements (e.g., “is the same word used in two different requirements?” and “Is there a potential conflict between these requirements”), but not semantic analysis (e.g., is there a potential conflict between two requirements”), even if different terms are used in these requirements definitions (similarity between concepts instead of syntactic keyword matching).

The use of 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 [19]. Automated reasoning can support tasks for requirements conflict analysis.

In this paper, we propose OntRep [10], 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 works if different terminologies are used for describing semantically equivalent concepts.
We empirically evaluate OntRep [10] with a real-life project at Siemens Austria, where six project managers in two teams inspected given project requirements to identify potential requirement conflicts. A requirements engineering expert provided control data for all tasks. Then, we performed similar tasks with OntRep to compare the quality of results. Comparing with the manual results, we analyzed OntRep performance regarding (a) different types of conflicts and (b) conflicts on different complexity levels. A conflict is rated more complex the more requirements elements are involved.

The remainder of the paper is organized as follows: Section 2 summarizes related work on requirements conflict analysis and semantic technologies; Section 3 introduces the OntRep approach and motivates research issues. Section 4 outlines the case study and Section 5 presents results. Finally, Section 6 discusses the results and Section 7 concludes and suggests further work.

2. RELATED WORK
This section presents related work on requirements conflict analysis and semantic technologies as foundation for automation of conflict analysis [10].

2.1 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” [6]. Several approaches address the issue of automated requirements conflict identification:

The Trace Analyzer by Egyed and Grünbacher [6] analyzes the footprint 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. [11] 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 analyze requirements to find semantic conflicts.

Information retrieval approaches [12], such as the RETH approach [13] 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 Bakkus-Naur-Form (EBNF) [21] is a syntax for requirements, which is used to improve the understandability of requirements for humans and machines. Such syntax is a prerequisite for parsing requirements in order to identify semantic conflicts as proposed in this paper.

2.2 Semantic Technologies
Semantic Integration is defined as the solving of problems originating from the intent to share data across semantically heterogeneous data sources [9]. These challenges include the matching of ontologies [17] or schemas, the detection of duplicate entries, the reconciliation of semantic or syntactic inconsistencies, and the modeling of complex relations across different data sources [14]. Over the last years, semantic integration has become increasingly crucial to a variety of information-processing applications (such as integrating heterogeneous engineering tools in the production automation system engineering domain [2] or integrating business services in the Air Traffic Management domain [16]) and has received much attention in database and AI communities. One of the most important and most actively studied problems in semantic integration is establishing semantic mappings between the vocabularies of different data sources [5]. This can also be used for requirements tracing.

Wache et al. reviewed a set of ontology-based approaches and architectures that have been proposed in the context of data integration and interoperability. They sum up the most important uses of ontologies [22], e.g. content explication, which means that the ontology enables accurate interpretation of data from multiple sources through the explicit definition of terms and relationships in the ontology.

Noy [18] identified three major use cases of ontologies for supporting semantic integration: the task of finding mappings (semi-)automatically, the explicit formal representation of these mappings, and reasoning using these mappings. There exist two generic architectures for mapping discovery between ontologies. On the one hand, a general upper ontology which is used by developers of linked applications, such as the Suggested Upper Merged Ontology (SUMO) [16] or DOLCE [8]. On the other hand, heuristics-based or machine learning approaches that use various characteristics of ontologies (e.g. structure, concepts, instances) to find mappings. These approaches are similar to techniques for mapping XML schemas or other structured data like requirements [1][4].

2.3 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 [3]. The core purpose of NLP techniques is to achieve human-like language processing for a range of tasks or applications [14].

Most important NLP models used in this research are part-of-speech (POS) tagging and sentence parsers [3]. 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 [3]. Stanford parser/tagger1 and OpenNLP2 are the core set of NLP tools used in this paper. Furthermore, we use WordNet, a large lexical da-

1 http://nlp.stanford.edu/software/lex-parser.shtml
2 http://opennlp.sourceforge.net/
In this paper, we focus on (a) functional requirements (following EBNF template) and (b) a set of requirements constraints regarding technical, requirement-specific or documentary issues like categorization, conflict analysis, and tracing.

3. 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 [10] links similar concepts, if they share 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 conflicts regarding a set of more than two requirements and constraints, which are hard to identify manually. In the following subsections, we provide an overview on the approach and motivate research issues.

3.1 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.

![Figure 1. EBNF requirements structure (sentence level) [21].](image)

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 (used to check whether the requirements are well-formed). Additionally, project specific knowledge is included as a project glossary. Figure 2 shows examples for functional requirements, requirement constraints, and project glossary knowledge. All of these elements are used as input for the automated ontology-based conflict analysis approach.

![Figure 2. Inputs for Req. Conflict Analysis.](image)

From this input we derive three types of conflicts, which can be detected with OntRep:

- Not well-formed requirements (conflicts between requirements and EBNF grammar or documentation guidelines).
- Conflicts between functional requirements and technical and/or requirement constraints.
- Conflicts between two functional requirements.

3.2 Automated Conflict Analysis

We developed a prototype tool for the OntRep approach as a plug-in to Trac3, an open source collaboration platform consisting of a Wiki, ticket management system, and subversion integration, which can be extended by Python plug-ins.

Phase A: Link 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 [10]. The following 8 steps automate the building of this project-specific knowledge base:

1) Define the 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 (e.g., in documents). For our research prototype we export these natural language texts and import them into the 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 list4.

---

3 http://trac.edgewall.org/
4 http://www.textfixer.com/resources/common-english-words.txt
4) Bring all remaining words into their root form: this process is called “stemming” based on a well-known algorithm, like the “Porter Stemmer” algorithm [20]. An example is to stem “jumping” to “jump”.

5) Get all synonyms and hyponyms of the analyzed words in the requirements by using the natural language processing library “WordNet” [15]. For example, “house” is a synonym for “building”, “dog” is a hyponym of “animal”. Further check all relevant substrings of a word like “net” as a substring of “network”.

6) Heuristic-based assignment of each requirement to the defined concepts depending on the number of hits for 1) synonyms, 2) hyponyms and 3) substring matches meet the given threshold values. So the number of met thresholds is between 0 and 3. 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 (multi-dimensional assignment is allowed).

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 automate the process of 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 the EBNF grammar.

3) Linking of requirements to semantic concepts. The “things to be processed” of the specific requirements, which were extracted in the previous step, now 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 respectively for each “thing to be processed” 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 3 depicts an example for logical conflicts between requirements and requirement constraints. Requirement RQ-11 links the 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 “Secure 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, while relying on the facts specified in GL-1 and GL-2.

Figure 4 depicts an example for numerical 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 con-
cept 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 “Notification” is linked to the semantic concept “SSL Encryption”, for which a technical constraint 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.

3.3 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 [10], 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 3.2: numeric discrepancies and logical requirements conflicts. These conflicts may occur between different elements: between requirements, between a requirement and a constraint, and 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 analysis 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. Thus, we assume that OntRep reaches a higher completeness of complex conflict identification 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 [10], 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.

In order to address the research questions we derived the following independent variables (according to [7]) for evaluation purposes:

- The number of requirements determines the effort necessary for conflict analysis.
- Total number of true requirements conflicts (seeded conflicts according to different types and levels of complexity) existing in a list of requirements, which can be identified by various approaches for conflict detection. This is a baseline measurement for the effectiveness of an approach, i.e., a perfect approach would find 100% of the true requirements conflicts.

- Conflict type: A conflict can have one of the following types: 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).
- Complexity level: we define two complexity levels: simple and complex. Conflicts related to up to 3 elements (requirements, constraints, and glossary terms) are rated as simple, conflicts with more than 3 elements involved are considered as complex conflicts.
- Approach for conflict analysis: manual vs. automated conflict analysis approaches.
- Formality of requirements specification: e.g., for automation approaches the formal structure of requirements is an important factor. As described above, we use the EBNF template for specifying requirements. Using plain text or other formats probably affects the correctness and completeness of identified conflicts.

Dependent variables and metrics that we want to study by the evaluation are:

- Number of conflicts identified. This number consists of two measures: recall (how many of the true conflicts are identified), for measuring the effectiveness of an approach, and false positives (number of wrongly identified conflicts). Thus, we also measure the precision (how many of the identified conflicts are true conflicts) of the conflict analysis approach. We collect these numbers for each of the 3 conflict types mentioned above.
- True conflicts not been identified (false negatives). This number results from subtracting the number of true conflicts identified from the total number of true requirements conflicts.
- Completeness of each identified conflict: a conflict is identified completely, i.e. all requirements that are involved are identified. A conflict is identified partially if only a subset of all involved requirements is identified. It may also be that requirements are identified which are not relevant for the conflict. This case can be ignored as long as the number of these requirements is not too high for a given conflict (i.e. less than 50% of requirements related to a conflict).

Besides these parameters we also record the effort for requirements conflict analysis. This includes preparation effort (e.g., creating the ontology that is used for conflict analysis), and conflict analysis effort. The case study is described in detail in the following section.

4. CASE STUDY DESCRIPTION

The following subsections describe the most important characteristics of the pilot study design.

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 (see Section 3.3) with focus on different types and complexity levels of conflicts.

Study Subject. As described in [10] the case study project is a software development project at Siemens Austria with the goal to design and implement a web application that serves as a plat-
The process for the automated approach is:

B2) **Ontology preparation:** A tool expert constructed one ontology class in OntRep (Protégé) for each category and then imported the given requirements from Trac as CSV into OntRep.

B3) **OntRep requirements conflict analysis:** Then, we provided the requirements as CSV-input to OntRep. Further, the tool expert had to model the constraints as facts and the formal guidelines as rules in the ontology. We captured the effort for this preparation. Afterwards, automated conflict identification was executed. The results were summarized in a final report. 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.

## 4.1 Threats to Validity

We addressed threats internal validity [7] 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. 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.

Further, the requirements in this case study were formulated using the EBNF syntax, which is a major condition for OntRep to analyze the requirements. We did not yet analyze the quality of results with a set of requirements, which is not or only partially formulated in EBNF. Further studies are needed to evaluate this.

## 5. RESULTS

The following subsections describe the results of requirements conflict analysis with OntRep regarding different conflict types and complexity and compare the results with the manual analysis.

### 5.1 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:

1. 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);
2. 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’”;
3. Conflicts between requirements (CRR), e.g. the requirements “The system shall update the index at least 30 times
We have chosen these conflict types because they were most relevant in the case study project and we think that they are typical for software development projects. Furthermore, the necessary elements (constraints, glossary terms) can be modeled in OntRep by means of facts and rules. Table 1 summarizes recall and precision values for both the manual and the automated approaches per conflict type.

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 was 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, 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, as depicted in Table 2 and explained in the following subsection.

Table 2 and explained in the following subsection.

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 was 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, 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, as depicted in Table 2 and explained in the following subsection.

<table>
<thead>
<tr>
<th>Table 1 Overall Recall and Precision per Conflict Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Indiv.</td>
</tr>
<tr>
<td>Group</td>
</tr>
<tr>
<td>Expert</td>
</tr>
<tr>
<td>OntRep</td>
</tr>
</tbody>
</table>

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 CRR 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). The reason is that in one group one true conflict identified by an individual was considered not be a conflict by the group, and on the other hand a non-conflict was considered to be a conflict.

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). The reason is that we conducted these three activities in order to enable the evaluation of the semantically-enabled requirements conflict analysis approach:

- **Proper preparation of OntRep:** all the terms used in the requirements have to be modeled as glossary terms in the ontology. Furthermore, the constraints have to be modeled as facts/rules in the ontology as well.
- **Adaption of requirements:** As described in section 3 the requirements have to be formulated in a certain structure, so that the different parts of each requirement can be provided as input for ontology-based reasoning. In the case study, the requirements existed in an EBNF-like structure. Those requirements, that were not structured well, were adapted to fit into the EBNF.
- **Adaption of conflicts:** We adapted the seeded conflicts in a way that allows the current implementation of OntRep to recognize them, e.g. we always used a numeric representation of numbers instead of a textual one, and adjusted the requirements accordingly (e.g. so that “100” instead of “one hundred” was contained). Furthermore, we only seeded types of conflicts that the implementation of OntRep is able to analyze. Other conflict types (see discussion for examples) were not seeded.

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.

### 5.2 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, i.e., if more than 3 elements (requirements, constraints, and glossary terms) are involved in a conflict, the conflict is complex. We chose this threshold subjectively, because we think that analyzing (comparing) 3 elements cognitively is still feasible, whereas it gets more complicated and difficult to analyze a conflict when 4 and more elements are involved. We seeded 12 simple and 10 complex conflicts (for simple and complex conflicts per conflict type see detailed numbers for each conflict type in Table 2).

In comparison to OntRep results, the manual conflict analysis approach resulted in a lower completeness, no matter which type of conflict was investigated: 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.

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 (2 conflicts completely found) and automated results (7 conflicts completely found) 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 automated approach.

Comparing correctness of simple and complex conflicts shows that correctness of identified complex conflicts is higher than of simple ones: For example, regarding CRR 2.6 false positives have been introduced for simple conflicts in average. On the other hand, no false positives have been identified for complex conflicts. The OntRep results for all types of conflicts are complete: all 22 conflicts of the defined conflict classes in the given data were identified, due to the 3 reasons for 100% conflict identification described in section 5.1. 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 [10].

6. DISCUSSION

The results of conflict analysis with OntRep seem convincing for the given conflict types. We focused our evaluation on these three conflict types, because these types cover the most typical conflicts occurring in practice: conflicts between requirements, conflicts between requirements and some constraints, or conflicts of requirements with some formal guidelines. Furthermore, these conflict types can be modeled in OntRep/Protégé by means of facts and rules and are therefore suitable for an initial evaluation. The automated conflict analysis depends on the following factors:

- **Requirements structure:** OntRep analyzes different sections of each requirement to map them to the according 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 (typical for large projects) instead of simple documents, because requirements have to be understandable and clear even without the context that usually exists in a requirements document.

- **As depicted in Figure 3 and Figure 4 OntRep can identify conflicts based on numeric representation of values and logical dependencies. Other types of representations and types of conflicts are not yet covered.**

- **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: e.g. in Figure 3 the glossary terms “secured resources” and “trusted persons” make clear that the requirement “(Unauthenticated) user shall be able to access his configuration page” and the constraint “no secured resources shall be processed by untrusted persons” result in a conflict, because “configuration page” is a secured resource, but “unauthenticated user” is not a “trusted person”. Usually, there is a project glossary existing in a project that can be imported into the ontology without big effort, so that all relevant glossary terms are available for conflict analysis.

### Table 2 Results (recall and precision) of conflict identification per conflict type and complexity level

<table>
<thead>
<tr>
<th>Type CRC: To find: 0 simple and 7 complex conflicts</th>
<th>Simple Conflicts (12)</th>
<th>Complex Conflicts (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully (f)</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Partially (p)</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Recall (F/p)</td>
<td>0.8 (2.0)</td>
<td>2.0 (1.4)</td>
</tr>
<tr>
<td>Precision (P)</td>
<td>0.8</td>
<td>2.0</td>
</tr>
<tr>
<td>Not Found</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

| Type CRG: To find: 10 simple and 0 complex conflicts |
|-----------------|----------------------|
| Fully           | 2.5 (1.2)            |
| Partially       | 0.7 (1.2)            |
| Recall (F/p)    | 0.7 (1.4)            |
| Precision (P)   | 0.7 (1.0)            |
| Not Found       | 0.0                  |

| Type CRR: To find: 2 simple and 3 complex conflicts |
|-----------------|----------------------|
| Fully           | 0.0 (0.0)            |
| Partially       | 0.8 (0.4)            |
| Recall (F/p)    | 42% (50%)            |
| Precision (P)   | 28% (18%)            |
| Not Found       | 1.2 (0.4)            |
• Quality of ontology: another prerequisite is correct modeling of the ontology, which requires expert knowledge. If the user does not specify them in the intended way, the tool cannot find the conflicts. If a constraint is missing in the ontology, the tool does not find any conflicts linked to that (missing) 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.

Based on these factors, we can derive the following limitations for the current implementation of OntRep:

• Numeric representation of values but no “mixed representation” supported: at the moment OntRep supports only the comparison of numeric values, i.e. numeric representation of elements to be compared is a prerequisite for OntRep: if a requirement states “response time shall be 5 seconds” and a second requirement states “response time shall be 3 seconds” a conflict would be found, whereas there would no conflict be detected if the second requirement was stated like “response time shall be three seconds” mixed representation: numeric and textual representation.

• Logical dependencies but no “similarity of concepts” supported: OntRep finds conflicts based on contradicting numeric values and logical dependencies (example: “secured resource” and “trusted person”), but is not yet able to recognize similarity of concepts, e.g. a conflict between requirements “background color shall be green” and “background color shall be blue” is currently not detected by OntRep, but will be in the future.

• Structured requirements but no “free-form requirements” supported: when informal requirements instead of structured ones should be analyzed with OntRep, the completeness of detected conflicts would be much lower than with structured requirements, because parsing each requirement would be much more difficult. Each requirement that is given without any pre-defined structure would have to be modeled manually, resulting in high effort and complexity, since the correct modeling of each requirements needs to be ensured by comparing the original requirement and the requirement’s model.

• Limited conflict types: at the moment OntRep is able to detect the 3 types of conflicts analyzed in the case study, but further types of conflicts exist and it will be further work to adapt OntRep so that it recognizes them. Examples for further conflict types are: (1) conflicts of requirements with budget constraints, e.g. if the used technology is more expensive than the budget allocated; (2) “conflicting dates”: statements like “the administration panel is part of the user interface”, “the estimated implementation time for the administration panel is 5 weeks”, and “the user interface shall be implemented in 4 weeks” result in a “conflicting date conflict”; (3) conflicts of requirements with design decisions: at the moment OntRep focuses on conflict detection within requirements, but cross-checks with other artifacts, such as design models are also possible. It will be further work to make OntRep capable of detecting these conflict types.

Despite the given prerequisites and limitations of OntRep and the fact that we arranged the case study requirements and conflicts so that these factors enable the evaluation of the OntRep prototype, 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.

The feedback from some study participants was “…hard to do the conflict analysis manually” and “it is difficult and cumbersome to find the conflicts”. These statements justify the evaluation of automation approaches. Furthermore, existing conflict analysis automation approaches [12][13] are syntax-driven and suffer from some limitations, e.g., that different stakeholders in a domain or project often use different terminologies, i.e., different words (syntactic encoding) for the same concept, or different word forms. This makes searches for relevant requirements limited and incomplete, which may cause or at least not avoid inconsistencies. In this context, the enhancement of our approach is that not only keyword matches result in an identified conflict, but also synonym matches (via WordNet). We use the benefits of ontologies, namely that they map different terms to the same concept and thus support a better comparison of differently formulated requirements (consistency checking).

7. 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 with different domain backgrounds and terminologies have to work together in large and/or 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 at Siemens Austria 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 (conflict between requirement and constraint, conflict between requirements, conflict between requirement and documentation guideline) 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. The according efforts have been reported in [10]. Regarding complexity levels the results showed that the differences between manual and OntRep results increase with the level of complexity, that means automation becomes more valuable the more complex the conflicts are (i.e., the more elements, such as requirements, constraints, glossary terms are involved in a conflict).
Further work will focus on the replication of this pilot study in a larger context, i.e., with more participants to improve the external validity of results. 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.

Further work will also deal with enhancing the implementation of OntRep so that the currently existing limitations of OntRep (see discussion) are resolved by: “mixed representation instead of numeric values only”, analysis of “similarity of concepts”, “informal requirements (not in EBNF format) instead of structured requirements”, and “detection of further conflict types”.

The results of the study are promising for OntRep and justify the further investigation of semantic technology for the application to automated requirements conflict analysis.

8. Acknowledgements

We want to thank Alexander Wagner for the prototype implementation of the OntRep concepts and his support during the pilot study. This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria. In addition, this work has been partially funded by the Vienna University of Technology, in the Complex Systems Design & Engineering Lab.

9. REFERENCES


