Addressing Semantic Ambiguities in Natural Language Constraints

Imran Sarwar Bajwa[1], Mark Lee[1], Behzad Bordbar[1], Ahsan Ali[2]

[1]School of Computer Science, University of Birmingham, B15 2TT, Birmingham, UK

[2]Queens Academic Group, Auckland, New Zealand

{i.s.bajwa, m.g.lee, b.bordbar}@cs.bham.ac.uk

Abstract

In NL2OCL project, we aim to translate English specification of constraints to formal constraints such as OCL (Object Constraint Language). In English to OCL translation, our contribution is a semantic analyzer that uses the output of the Stanford parser for shallow and deep semantic parsing. Our analysis of the output of shallow semantic parsing showed that semantic roles were mis-identified for a few English constraints due to semantic ambiguity. Similarly, in deep semantic parsing, it is difficult to resolve scope of quantifier operators due to scope ambiguity that is another sub-type of semantic ambiguity. In this paper, we highlight the identified cases of semantic ambiguities in English constraints. We also present a novel approach to automatically resolve the identified cases of the semantic ambiguities. The presented approach is also evaluated to show that by addressing the identified cases of semantic ambiguities, we can generate more accurate and complete formal (OCL) specifications.

1. Introduction

Resolution of ambiguity in natural language specifications of software requirements and constraints is the key challenge in automated generation of UML (OMG, 2007) (Unified Modelling Language) models and OCL (OMG, 2010) constraints, respectively. Mich (2004) showed that 71.8% of a sample of NL software specification is ambiguous. Due to the very high number of ambiguous NL software specifications, the available tools for translation of NL to UML are limited to 65%-70% levels of accuracy (Harmain, 2002) in real time software development. Similar to NL to UML translation, the NL to OCL translation is also highly affected due to inherent ambiguities (Umber, 2011) of NL especially the semantic ambiguities. The ambiguous NL specifications result in inaccurate OCL constraints.

In NL2OCL project (Bajwa, 2010), for automatic translation of English specification of constraints to OCL, input English was syntactically analyzed using the Stanford parser (Marneffe, 2006). The Stanford parser generates a parse tree and dependencies for an input English constraint. There are a few cases where the Stanford parser generates wrong parse tree or generates wrong dependencies. We have resolved the identified types of syntactic dependencies in (Bajwa, 2012).

After handling syntactic ambiguities, the output of the Stanford parser is used as input to our semantic analyzer developed for shallow and deep semantic parsing of English constraints. The output of the semantic analyzer was mapped to OCL syntax to generate complete OCL constraints. We tested a large number of sample English constraints to test the accuracy of the translation from English to OCL. In a few cases, the wrong OCL was generated from English constraints. We identified that the key reason of the wrong translation was wrong classification of semantic roles. We also figured out that the semantic roles go wrong because of semantic ambiguity as various tokens have multiple meanings and can be assigned more than one semantic roles. Due to very high number of cases with semantic ambiguity, it was critical to address semantic ambiguity for correct translation of English to OCL constraints (Bajwa, 2011a). In rest of the paper, we present a novel approach to resolve identified cases of semantic ambiguities.

SBVR Business Vocabulary

A business vocabulary (OMG, 2008; section 8.1) consists of all the specific terms and definitions of concepts used by an organization or community in course of business. In SBVR, a concept can be a noun concept or fact type (Bajwa, 2011b). Figure 1 shows an overview of SBVR metamodel.

There are various types of SBVR vocabulary. However we have used following five types of vocabulary in our approach (Bajwa, 2011c) are explained below:

- *Object Type*: A general concept that exhibits a set of characteristics to distinguished that object type from all other object types" e.g. library, student, etc.

- *Individual Noun*: A qualified noun that corresponds to only one object [12] (section: 8.1) e.g. 'London' is a famous city in UK.
- *Verb Concept*: A verb concept [12] (section: 8.1) specifies the relationships among noun concepts e.g. library *has* books.
- *Characteristic*: An abstraction of a property of an object [12] (section: 8.1) e.g. name of student is Matt, here name is characteristic.
- *Fact Type*: A fact type can be binary fact type e.g. "student *borrows* book". Other possible forms of fact types are associative fact type, partitive fact types, categorization fact types, etc.

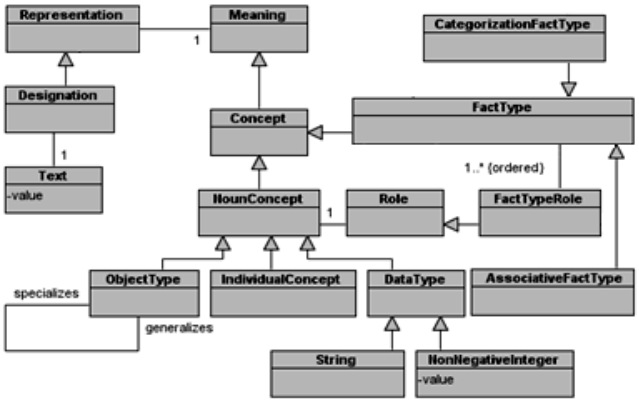

Figure 1: SBVR Vocabulary – SBVR Metamodel subset

We recommend the reader (OMG, 2008; section 8.4) for the detailed reading of SBVR business vocabulary.

The remaining paper is structured as follows: Section 2 provides the detailed description of the problem and the solution of the problem is given in Section 3. Section 4 presents the evaluation of the presented approach. Section 5 states the related work and the paper is concluded to discuss future work finally.

2. Problem Description

In this section, we discuss the various types of semantic ambiguities we came across in automatic translation of English constraints to OCL constraints. The semantic analysis of English constraints is performed into two halves: shallow semantic parsing and deep semantic parsing. Ambiguities in both these types are discussed one by one.

Shallow Semantic Analysis

In shallow semantic parsing, the Semantic Role Labeling (SRL) relies on the typed dependencies generated by the Stanford parser. The semantic or thematic roles are typically assigned to each syntactic structure in each English sentence. We use SBVR (OMG, 2008) (Semantic Vocabulary and Rules) vocabulary as the target semantic roles as mapping from SBVR to OCL is easy due to support of higher order logic for both SBVR and OCL. We have identified following mappings (see Table 1) of English text elements to SBVR vocabulary.

English Text elements	SBVR Vocabulary
Common Nouns	Object Type
Proper Nouns	Individual Concept
Generative Noun, Adjective	Characteristic
Verbs	Verb Concepts/ Fact Type

Table 1: Mapping class model to English

In semantic role labelling, we face two different cases of semantic ambiguities. First case relates to the shallow semantic parsing. In English, a verb can be in the form of a simple verb, a phrasal verb or a verbal collocation. However, with respect to Table 1, an action verb can be mapped to a *Verb Concept* or a *Fact Type*. This case is very important to resolve as if a verb is labelled as a *Verb Concept* then it will be mapped to navigation expression in OCL else it is ignored. For example, in Figure 2, the token 'place' can be a *Verb Concept* or part of a *Fact Type*.

English: A customer cannot place more than two orders.

Figure 2. Input English Constraint

Another case of ambiguity is order of the predicate arguments. For example, in Figure 2, the predicate should be `place(customer, order)`. However, if the sentence in 2 is passive voice like "More than one order can not be placed by a customer" then our rule based approach generates predicate `place(order, customer)` and it is wrong. Solution of the both cases of semantic ambiguities in shallow semantic parsing is presented in later part of the paper.

Quantification in English Constraints

The deep semantic analysis involves generation of a fine-grained semantic representation from the input text. To generate a logical representation from English constraints, we need to identify quantifications. However, there is difficult to identify the scope of a few quantifier operators as such quantifier operators are ambiguously used in an English constraints. For example, in Figure 2, article 'a' with customer can be a universal quantification ($\forall X$) or may be an existential quantification ($\exists X$) or may be Uniqueness quantification ($\exists =1X$). The identification of exact quantifications is really important to generate a correct logical representation.

3. Addressing Semantic Ambiguities

To address the known semantic ambiguities, discussed in previous section, we present a novel approach. We have identified that the ambiguities in shallow and deep semantic parsing are due to the absence of the context. However, these semantic ambiguities can be resolved by using the context of the English text. In NL2OCL project, to translate English specification of constraints to OCL constraints, two inputs are required: English specification of a constraint and a UML class model. We propose the use of the information (such as classes, methods, associations, multiplicity, etc) available in the input UML class model to handle semantic ambiguities.

The used approach for addressing the both types of semantic ambiguities is explained in remaining part of the section.

Addressing Semantic Ambiguities in SRL

It is a fact that the semantic ambiguities in English constraints are due to absence of the context of the constraint. As, a UML model is a typical context of the OCL constraints, we use the UML class model shown in Figure 3 to address the both semantic ambiguities identified in shallow semantic parsing.

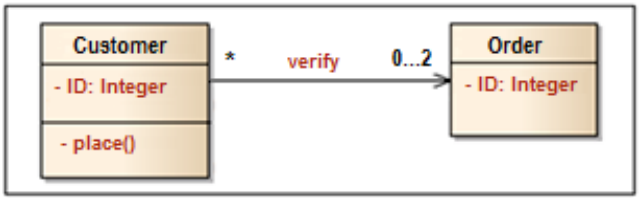

Figure 3. A UML class model

To identify correct semantic roles, we worked out a mapping in English constraints, UML class model and SBVR based semantic roles.

English Elements	UML Elements	SBVR based Semantic Roles
Common Nouns	Classes	Object Type
Proper Nouns	Objects	Individual Concept
Generative Noun, Adjective	Attributes	Characteristic
Verbs	Methods	Verb Concepts
	Associations	Fact Type

Table 2: Identifying Semantic Roles

The first case of semantic ambiguity was related to assignment of semantic roles to a verb in English constraint. We have shown in Table 1, a verb in English constraint can be labeled as a 'Verb Concept' or a 'Fact Type'. However, if we map information of Figure 2 to Figure 3, we found that 'Customer' and 'Order' are two classes while 'place' is name of a method. Due to the fact that methods in a UML class model are action performed by the class, we classify verb 'place' as a Verb Concept (see Table 2). If the verb 'place' is an association among classes 'Customer' and 'Order', the will be classified as a Fact Type.

We can identify correct semantic role by mapping information to the UML class model by checking that verb is an operation or an association. If a verb is operation it is mapped to 'Verb Concept' else it is mapped to a 'Fact Type'. Moreover, for the sake of confirmation we also map the common nouns such as 'Customer' and 'Order' to the classes in the UML class model. After identifying the correct semantic roles, following output was generated (see Figure 4) for example "A customer cannot place more than two orders."

English Elements	Assigned Semantic Roles
A	-
customer	Object Type
cannot	-
place	Verb Concept
more than two	-
orders	Object Type

Figure 4. Semantic roles assigned to input English sentence

The second case of semantic ambiguity was related to the order of predicate arguments extracted for a predicate. To resolve this type of ambiguities the information of English constraint given in Figure 2 was again mapped to the information of the UML class model shown in Figure 3. After mapping we found that 'Customer' and 'Order' are two classes and there is a directed association between these two classes. The directed association shows that the 'Customer' is an agent or an actor and 'Order' is a patient or a thematic object. In the light of this information it is simple to identify that the predicate arguments should be like `place(customer, order)`. Another benefit of such mapping is that if English sentence in passive voice the same predicate will be generated e.g. `place(customer, order)`.

Addressing Semantic Ambiguity in Quantification

To address the semantic ambiguity, first we identified the candidate quantifier operators in English constraints. Then the identified quantifiers are mapped to the multiplicities of classes in a UML class model to confirm the

quantifications. We have figured out following four types of the quantifications in English constraints.

i. Universal Quantification (∀ X): In English, the quantification structures such as 'each', 'all', and 'every' are mapped to universal quantification. However, some times the determiners 'a' and 'an' used with the subject part of the sentence can be classified as a universal quantification. For example, in Figure 5, the determiner 'a' used with the Object-Type 'Customer' due to the fact that we are processing constraints and generally constraints are mentioned for all the possible X in a universe.

ii. Existential quantification (∃ X): The keywords like many, little, bit, a bit, few, a few, several, lot, many, much, some, etc are mapped to existential quantification.

iii. Uniqueness Quantification (∃ =1X): To identify uniqueness quantification, the determiners 'a' and 'an' in English constraint are mapped to multiplicity used with a class. Uniqueness quantification is mapped to exactly-one quantification in SBVR.

iv. Solution Quantification (§X): If the keywords like more/greater than or less/smaller than are used with a cardinal number *n* then solution quantifier is mapped to at-least *n* Quantification (see Figure 5) and at-most *n* Quantification respectively.

Output of quantification handling for the example discussed in the Figure 4 is shown in the Figure 5.

English Elements	Assigned Semantic Roles
A	Universal Quantification
customer	Object Type
cannot	-
place	Verb Concept
more than two	At-least *n* Quantification
orders	Object Type

Figure 5. Semantic roles assigned to input English sentence

Semantic Interpretation: After shallow and deep semantic parsing, a final semantic interpretation is generated. A simple interpreter was written that uses the extracted semantic information and assigns an interpretation to a piece of text by placing its contents in a pattern known independently of the text. Figure 6 shows an example of the semantic interpretation we have used in the NL2OCL approach:

English: A customer cannot place more than two orders.

Semantic Interpretation:

(place
(object_type = (∀ X ~ (customer ? X)))
(object_type = §Y ~ (order ? Y))))

Figure 6. Semantic Interpretation of English constraint

We do not provide details of translation of SBVR based logical representation due to two reasons: one it is out of the scope of the paper secondly, an approach for translating SBVR to OCL has already been published in (Bajwa, 2011)

4. Evaluation

In this section, we present a case study on the "Royal & Loyal" model. The Royal & Loyal model was originally presented for introducing *OCL By Example* in (Warmer and Kleppe, 2003). Afterwards, the Royal & Loyal model is used in various publications, e.g., (Tedjasukmana, 2006, Dzidek et al., 2005, Wahler, 2008). The same model is also shipped with several tools as an example model, e.g., (Dresden Technical University, 2007).

The Royal & Loyal Model Constraints

The Royal & Loyal case study has also been solved by Wahler (2008) in his PhD thesis. We aim to compare the results of our approach to Pattern based approach as Wahler's approach is the only work that can generate OCL constraints from a natural language. There are 26 English constraints in the Royal & Loyal case study. Wahler solved 18 English constraints into OCL out of 26 using his (pattern-based) approach. In comparison to Wahler's pattern based approach, our NL-based approach has successfully translated 25 constraints to OCL.

There was only one constraint that was not fully translated by our NL base approach due to the limitation that the vocabulary used in English constraint should also be part of the input UML class model. In the following English constraint, concepts 'credits' and 'debits' are not part of the Royal & Loyal model (Warmer and Kleppe, 2003: pp.22).

In the following section, we present on example of solved constrains:

English: Male customers must be approached using the title 'Mr.'.

OCL:

```
package: royal_and_loyal
context: Customer
inv: self.isMale implies self.title=Mr.
```

Figure 7. Output of NL-Based Approach

English: The maximum age of participants in loyalty programs is 70.

OCL:
```
package: royal_and_loyal
context LoyaltyProgram
inv self.participants->
        forAll(age <= 70).
```

Figure 8. Output of NL-Based Approach

In comparison of both approaches (see Table 3), NL-based approach produced for better results than the pattern based approach:

Approach Type	Total Constraints	Solved Constraints	%
Pattern based Approach	26	18	69.23%
NL-Based Approach without addressing semantic Ambiguities	26	21	80.76%
NL Based Approach after addressing semantic Ambiguities	26	25	96.13%

Table 3. Pattern based Approach vs NL Based Approach

Another advantage over Wahler's approach is that our NL-based approach is fully automatic, while in Wahler's pattern based approach, user has to do detailed manual analysis of the English constraints to choose the right pattern and then Wahler's tool Copacabana (Wahler, 2008) translates the pattern instances to OCL code.

5. Related Work

Natural languages are inherently ambiguous and resolution of all types of ambiguities such as lexical, syntactic, semantic ambiguities is a long standing challenge. Much work has been done in the field of natural language ambiguity identification and resolution. Some of the researchers (Mich, 2004), (Uejima, 2003), (Kiyavitskaya, 2008) have presented approaches to identify the various types of ambiguities in a natural language text especially the natural language software requirements. Mich (2004) showed that 90% of the software requirements are captured in a natural language [3] such as English. Hence, the resolution of semantic ambiguities in natural language specifications of software requirements and software constraints become more critical. However, translation of natural language such English to OCL is relatively a new area of research. We aim to contribute this area of research to improve the automated software modeling from natural language software requirements that also contains constraints.

6. Conclusion and Future Work

The primary objective of the paper was to address the challenge of resolving various cases of semantic ambiguity for accurate translation of English constraints to OCL. The results (see Table 3) show that after addressing semantic ambiguities by using the presented approach there was significant improvement in the results. To address this challenge we have presented a NL based automated approach that uses a UML class model as a context of the input English (constraints) and by using the available information in the UML class model (such as classes, methods, associations, etc) we can resolve attachment ambiguity and homonymy.

We have observed that some software constraints involve implicatures and pre-suppositions. To generalize the presented approach and further improve the accuracy of English to OCL translation we need to work on pragmatic ambiguities such as implicatures and presuppositions.

References

Bajwa, I.S., Bordbar, B., Lee, M.G. 2010. OCL Constraints Generation from Natural Language Specification, 14th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2010), pp: 204-213, Vitoria Brazil.

Bajwa, I.S., Lee, M.G., 2011a. Transformation Rules for Translating Business Rules to OCL Constraints, 7th European Conference on Modelling Foundations and Applications (ECMFA 2011). Birmingham, UK. Jun 2011. pp:132-143, Birmingham, UK

Bajwa, I.S., Lee, M.G., Bordbar, B. 2011b. SBVR vs OCL: A Comparative Analysis of Standards, 14th IEEE International Multitopic Conference (INMIC 2011). pp:261-266, Karachi, Pakistan.

Bajwa, I.S. Bordbar, B. Lee, M.G. 2011d. SBVR Business Rules Generation from Natural Language Specification, AAAI Spring Symposium 2011 – AI4BA, pp:541-545. , San Francisco, USA

Bajwa, I.S., Lee, M., Bordbar, B. 2012. Resolving Syntactic Ambiguities in Natural Language Specification of Constraints. accepted in 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2012) Dehli, India

Chen, B., Su J., and Tan, C.L. 2010. Addressing Event Noun Phrases to Their Verbal Mentions, in Empirical Methods in Natural Language Processing, Pages 872-881, Cambridge, MA, October, 2010

Cer, D., Marneffe, M.C., Jurafsky, D. and Manning, C.D. (2010). Parsing to Stanford Dependencies: Trade-offs between speed and accuracy." InProceedings of LREC-10.

Harmain, H. M., Gaizauskas R. 2003. CM-Builder: A Natural Language-Based CASE Tool for Object- Oriented Analysis. Automated Software Engineering. 10(2):157-181.

Warmer, J.B. and Kleppe, A.G. 2003.The object constraint language: getting your models ready for MDA. Second Editotion, Addison Wesley

Kiyavitskaya, N., Zeni, N., Mich, L., Berry, D. (2008). Requirements for tools for ambiguity identification and measurement in natural language requirements specifications, Requirements Engineering, Vol. 13, No. 3. (2008), pp. 207-239.

Manning, C.D. (2011). Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In proceedings of CICLing (1) 2011. pp.171~189

Marneffe, M.C., MacCartney Bill and Manning, C.D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006.

Mich, L., Franch, M., Inverardi, P.N.: Market research for requirements analysis using linguistic tools. Requir. Eng.(2004) pp.40-56

OMG. 2007. Unified Modeling Language (UML), OMG Standard, v. 2.3.

OMG. 2008. Semantics of Business Vocabulary and Rules (SBVR), OMG Standard, v. 1.0.

OMG. 2010. Object Constraint Language (OCL), OMG Standard, v. 2.2.

Toutanova K., Klein D., Manning C., and Singer Y. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.

Uejima, H. , Miura, T., Shioya, I. (2003). Improving text categorization by addressing semantic ambiguity Communications, Computers and signal Processing, 2003 pp. 796-799

Umber, Bajwa, I.S. 2011. NL-Based Automated Software Requirements Elicitation and Specification, 1st International Conference on Advances in Computing and Communications (ACC-2011), pp:78-83, Kerala, India

Wahler M. 2008. Using Patterns to Develop Consistent Design Constraints. PhD Thesis, ETH Zurich, Switzerland, (2008)