On Specifying Requirements Using a Semantically Controlled Representation

Imran Sarwar Bajwa[1] and M. Asif Naeem[2]

[1] School of Computer Science, University of Birmingham
B152TT Birmingham, United Kindom
[2] Department of Computer Science, University of Auckland
Auckland, New Zealand
i.s.bajwa@cs.bham.ac.uk, mnae006@aucklanduni.ac.nz

Abstract. Requirements are typically specified in natural languages (NL) such as English and then analyzed by analysts and developers to generate formal software design/model. However, English is ambiguous and the requirements specified in English can result in erroneous and absurd software designs. We propose a semantically controlled representation based on SBVR for specifying requirements. The SBVR based controlled representation can not only result in accurate and consistent software models but also machine process able because SBVR has pure mathematical foundation. We also introduce a java based implementation of the presented approach that is a proof of concept.

Keywords: Requirement Specifications, SBVR, Natural Language Processing.

1 Introduction

Since early days of software engineering, the use of a natural language (NL) to specify requirements [1] has been a challenge due to inherent semantic ambiguities of NL [2]. In this paper, we propose the use of a semantically controlled representation [3] for specifying requirements. The used representation is based on OMG's recently introduced standard, Semantic of Business Vocabulary and Rules (SBVR) [4]. A SBVR based representation can be beneficial in manifold aspects: SBVR is easy to understand for humans as it is close to NL, SBVR is easy to machine process as SBVR has a pure mathematical foundation [4].

The presented approach works as the user captures the requirements from users in simple English and inputs a piece of English specification of requirements. NL2SBVR approach [5] processes English text, extracts SBVR vocabulary (such as concepts, fact types, etc) and then uses SBVR vocabulary to construct SBVR business rules. SBVR Conceptual Formalization [4] and Semantic Formulation [4] are applied on SBVR vocabulary to generate SBVR rules. Finally, the SBVR Structured English [4] notation is applied to form a complete SBVR rules.

R. Muñoz et al. (Eds.): NLDB 2011, LNCS 6716, pp. 217–220, 2011.

2 NL to SBVR Approach

The NL to OCL is a modular NL-based approach that generates SBVR based semantically controlled representation from English text in following two steps:

2.1 Processing NL Text

Following is the detail of processing of English text:

Lexical Processing. In lexical processing, the input English text is read and the sentence margins are identified to split sentences. Each sentence is stored separately and passed to parts-of-speech tagger to identify the basic POS tags for input text. For POS tagging, the Stanford POS tagger v3.0 [6] is used that can identify 44 POS tags.

Syntactic and Semantic Interpretation. In this syntactic processing, various parts of a sentence are identified [7] and in semantic interpretation phase [8], the basic SBVR vocabulary is extracted using following mapping rules:

- All proper nouns are mapped to the individual concepts
- All common nouns are mapped object types.
- All auxiliary verbs and action verbs are mapped to verb concepts.
- The adjectives and possessive nouns are mapped to the characteristics.
- All cardinal numbers are mapped to quantification.

2.2 Generating SBVR Rule

In this phase, a SBVR business rule is constructed from the SBVR vocabulary generated in previous phase. SBVR rule is generated in two phases as following:

Applying Semantic Formulation. A set of semantic formulations such as logical formulations, modal formulations, quantifications are applied to each fact type to construct a SBVR rule.

Applying Structured English Notation. The last step in generation of a SBVR is application of the Structured English notation in SBVR 1.0 document, Annex C [4].

3 A Case Study

A small case study is discussed from the domain of robot systems that was originally presented by Callan [15] and [16]. The problem statement for the case study is as follows:

> *"An assembly unit consists of a user, a belt, a vision system, a robot with two arms, and a tray for assembly. The user puts two kinds of parts, dish and cup, onto the belt. The belt conveys the parts towards the vision system. Whenever a part enters the sensor zone, the vision system senses it and informs the belt to stop immediately. The vision system then recognizes the type of part and informs the robot, so that the robot can pick it up from the belt. The robot picks up the part, and the belt moves again. An assembly is complete when a dish and cup are placed on the tray separately by the arms of the robot."*

The problem statement of the case study was given as input (NL specification) to the tool. The tool parses and semantically interprets English text and extracts the SBVR vocabulary from the case study as shown in table 1:

Table 1. SBVR vocabulary generated from English text

Category Count Details	*#*	*Details*
Object Types /General Concepts	13	assembly_unit, user, belt, vision_system, robot, tray, dish, cup, parts, senso_zone, senses, arms, parts
Verb Concepts	10	consists, puts, conveys, enters, informs, stop, recognizes, pick, moves, placed
Individual Concepts	00	-
Characteristics	01	Type
Quantifications	10	1 Each, 7 at least *n*, 2 exactly *n*
Unary Fact Types	03	vision_system *senses*; belt *stops;* belt *moves;* assembly *is_complete*
Associative Fact Types	08	belt *conveys* parts, parts *enter* sensor_zone, vision_system *informs* belt, vision_system *recognizes* part, vision_system *informs* robot, robot *picks_from* belt, robot *picks* part, dish and cup *placed_on* tray
Partitive fact Types	01	assembly_unit *consists of* user, ….
Categ. Fact Types	01	user *puts* two kinds of parts

Here, types and arms are characteristics but wrongly classified as object types. Similarly, a verb concept senses is wrongly characterized as noun concept. Here, the fact types are further processed to generate SBVR rule types. There were one structural constraint and six behavioural constraints as shown in table 2:

Table 2. SBVR Rule representation of structural and behavioural constraints

Category	*#*	*Details*
Structural Requirements	01	It is possibility that each assembly unit *consists* of at least one user, at least one belt, at least one vision system, at least one robot with exactly two arms, and at least one tray for assembly.
Behavioural Requirements	06	It is permitted that the user *puts* exactly two kinds of parts, dish and cup, onto the belt. It is permitted that the belt *conveys* the parts towards the vision system. It is obligatory that at least one part *enters* the sensor zone, implies the vision system senses and *informs* the belt to *stop* immediately. It is obligatory that the vision system *recognizes* the *type* of part and *informs* the robot, so that it is obligatory that the robot *can pick* from the belt. It is obligatory that the robot *picks* the part, and the belt *moves* again. It is obligatory that an assembly *is* complete when at least one dish and cup *are placed* on the tray separately by the arms of the robot.

A matrix representing software constraints accuracy (recall and precision) test (%) for structural and behavioural constraints has been constructed.

Table 3. Evaluatin results of NL to OCL

Example	N_{sample}	$N_{correct}$	$N_{incorrect}$	$N_{missing}$	*Rec%*	*Prec%*
Structural	16	15	1	0	93.75	93.75
Behavioural	27	24	2	1	88.88	92.30
Total	43	39	2	2	73.3	93.02

The results of this initial performance evaluation are very encouraging and support both the approach adopted and in this paper and the potential of this technology in general.

4 Conclusion

The primary objective of the paper was to address the ambiguous nature of natural languages (such as English) and generate a controlled representation of English so that the accuracy of machine processing can be improved. To address this challenge we have presented a NL based automated approach to parse English software requirement specifications and generated a controlled representation using SBVR. Automated object oriented analysis of SBVR based software requirements specifications using NL2SBVR provides a higher accuracy as compared to other available NL-based tools.

The future work is to extract the object-oriented information from SBVR specification of software requirements such as classes, instances and their respective attributes, operations, associations, aggregations, and generalizations. Automated extraction of such information can be helpful in automated conceptual modelling of natural language software requirement specification.

References

1. Denger, C., Berry, D.M., Kamsties, E.: Higher Quality Requirements Specifications through Natural Language Patterns. In: Proceedings of IEEE International Conference on Software-Science, Technology & Engineering (SWSTE 2003), pp. 80–85 (2003)
2. Ormandjieva, O., Hussain, I., Kosseim, L.: Toward A Text Classification System for the Quality Assessment of Software Requirements written in Natural Language. In: 4th International Workshop on Software Quality Assurance (SOQUA 2007), pp. 39–45 (2007)
3. Kuhn, T., Controlled English for Knowledge Representation. Doctoral Thesis. Faculty of Economics, Business Administration and Information Technology of the University of Zürich (2010)
4. OMG. Semantics of Business vocabulary and Rules. (SBVR) Standard v.1.0. Object Management Group, (2008), http://www.omg.org/spec/SBVR/1.0/
5. Bajwa, I.S., Lee, M.G., Bordbar, B.: SBVR Business Rules Generation from Natural Language Specification. In: AAAI Spring Symposium 2011, San Francisco, USA, pp. 2–8 (2011)
6. Toutanova, K., Manning, C.D.: Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In: The Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, Hong Kong, pp. 63–70 (2000)
7. Bajwa, I.S., Samad, A., Mumtaz, S.: Object Oriented Software modeling Using NLP based Knowledge Extraction. European Journal of Scientific Research 35(01), 22–33 (2009)
8. Bajwa, I.S., Choudhary, M.A.: A Rule Based System for Speech Language Context Understanding. Journal of Donghua University (English Edition) 23(6), 39–42 (2006)