Analysis of Modeling Techniques used for Translating Natural Language Specification into Formal Software Requirements

Geet Sandhu
Assistant Professor
Department of Computer Science Engineering
Amity School of Engineering and Technology
Amity University Haryana

ABSTRACT

The prerequisite to any software development process is to first capture and understand the intended behaviour of the software solution based on the gathered requirements and understanding the constraints related to it. Hence requirement engineering becomes a very crucial step. Requirements are specified in Natural language by the user and it has to be translated to a formal representation for automating the requirement engineering process and also to uncover inconsistencies, ambiguities and perform conflict management in software requirements. This paper studies some existing models used for converting loosely specified natural language to a formal linguistic construct for software requirements. Observations on various existing approaches in this context are made**.**

General Terms

Requirement Engineering, Software requirements, Natural language processing

Keywords

Requirement Engineering(RE), Software Requirement Specifications(SRS), Natural Language Processing(NLP)

1. INTRODUCTION

According to IEEE Std 610.12-1990(IEEE Standard Glossary of Software Engineering Terminology)[1], Software Engineering is described in terms of an application of a methodical, closely controlled, proven approach to the development, operation, and maintenance of software or in other words the appliance of engineering to Software or Software product. Software development process translates consumer requirements into a software produce. The first segment in the development of software is to capture the anticipated behaviour of the expected software product as the desired software solution. This process is termed as requirement engineering. R.Saranya [2] suggested Requirement Engineering (RE) as an thoughtful approach to the goals to be reached by the forecasted software system. This phase is very decisive in the software development life cycle as issues like inconsistency, inaccuracy, ambiguity; incompleteness etc encountered during this time, if not handled carefully and timely in the initial stages can penetrate deep into the system hence affecting the entire process of development. As output to one stage serves as input to the next, hence there are fair chances that these issues if not resolved in the requirement engineering phase will travel to the next stages and finally lead to an undesired software product.

RE process is a four step modus operandi of elicitation of software requirements, analysis of software requirements , validation & documentation of software requirements detailed in the previous stage [3].Software requirements are uncovered, extracted and surfaced by a requirement engineer. His task is to understand and learn the stakeholder's needs and constraints and communicate effectively to the development team [4]. Since software requirements are given by the client in natural language, it becomes necessary for the Requirement Engineer to convert it first into a formal representation. For this natural language models needs to be converted to a formal model. A formal representation of language constructs helps us uncover inconsistencies and ambiguities [5]. Formal model yields an analysis of artifacts for analyzing requirements [6].

This paper aims at bringing forth a comprehensive study on existing approaches for formal representation of language constructs. Section-1 is introduction. Section-2 discusses the related work in this regard. Section-3 provides the observations on recent study. Section-4 concludes the discussion and provides useful information on the future scope in knowledge representation domain.

2. RELATED WORK

Various authors have proposed models and frameworks in the effort to represent software requirements from natural language to a more formal linguistic structure. Some of them are discussed in this section.

2.1 Predicate Logic based Approach

Ali & Khan et. al [7] combines word level & phrase level approach. The main component includes Dictionary, Sentence splitter, phrase to logical conversion & logic to phrase conversion. The author claims high degree of 80% accuracy which is primarily based on selecting predicate logic over other schemas for knowledge representation. The concept is illustrated in figure 1.

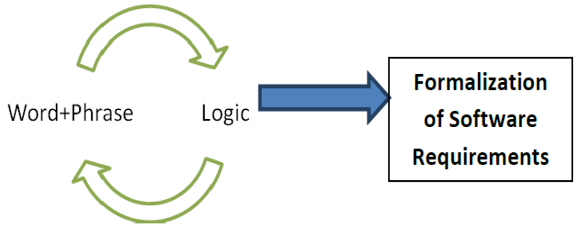

Figure1. Word-phrase logic technique

2.2 Object Oriented Analysis and UML based approach

Shinde et.al [8] simplifies the requirement specification process by representing it using UML diagrams. The approach proposed extracts knowledge from informal NL text using first POS tagging technique followed by lexical text understanding. Nouns, Verbs & Adverbs are extracted through POS tagging and subjects, objects & messages are identified through the second technique. Since the approach produces UML class diagram & UML seq. diagram, additional information are required about classes, objects, attributes & the messages exchanged between the objects. Hence respective identification rules are also suggested by the author[8]. Finally, automatic Java- code is generated. Figure 2. describes the approach discussed by the author

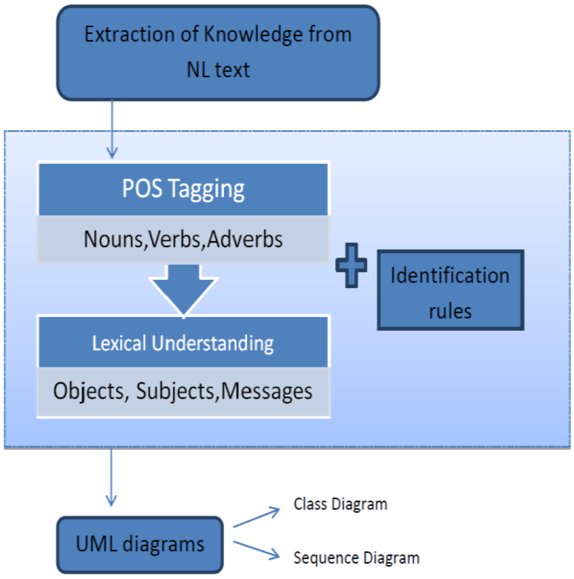

Figure2. UML based knowledge formalization technique

2.3 Ontology based approach

Sadoun et.al [9] proposes an ontology population based approach to transform NL requirements to formal specifications. This approach is based on instance property identification using extracting rules. These rules are based on 2 kinds of triples aims at identifying a property in ontology. Acquisition method of extraction rules is performed automatically. It makes a training corpus & a starting terminology by bootstrapping is used which automatically extracts rules. Then the generated extraction rule exploits syntactic dependency and semantic features. Later terminology expansion is done by making use of formal definitions of terms & their step is followed by capturing behaviour of the system through requirements using ontology population method. In this operation rules based on Antecedent & Consequent is described by the concept requirement pattern. User requirements are recognized if there exists atleast one 'Requirement Pattern' instance. Once property instances are identified, their concept instances are classified & identified. The author [9] uses OWL which applies its semantic property on individuals involved in the population to infer more knowledge & reasoning. Figure 3. discusses the requirement pattern based approach for translating natural language requirements into structured and more formal representation

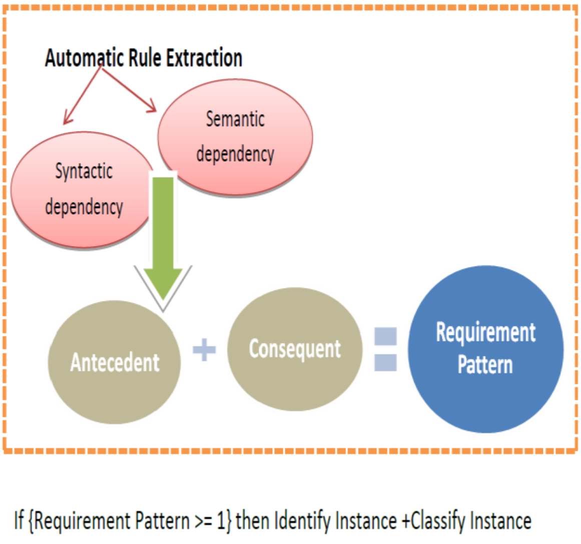
Automatic Rule Extraction
Syntactic dependency
Semantic dependency
Antecedent
+
Consequent
=
Requirement Pattern
If {Requirement Pattern >= 1} then Identify Instance +Classify Instance

Figure3. Requirement pattern approach for formal specification

Moor & Weigand [10] proposes a user driven ontology based framework in which user describes & defines the specification using a vocabulary. This approach capitalizes on user driven mapping of requirements using concept graphs. (re)Specification of requirements by the user is done if any breakdown occurs. The approach discussed by the author is shown in Figure 4.

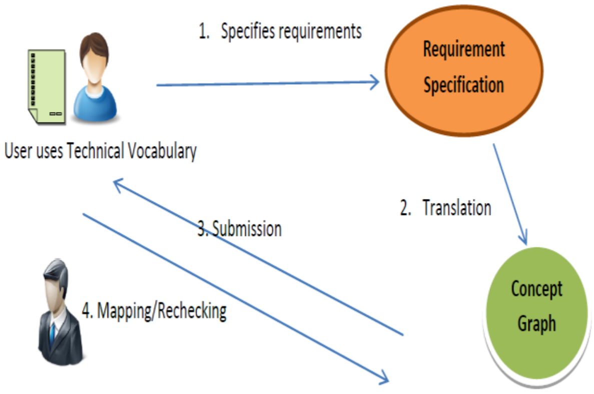
1. Specifies requirements
Requirement Specification
User uses Technical Vocabulary
2. Translation
3. Submission
Concept Graph
4. Mapping/Rechecking

Figure 4. User-driven mapping approach

2.4 Model driven engineering approach

Selway et.ol [11] proposes an iterative SBVR (Semantic of business vocabulary & business rule model that automatically transforms controlled natural language Business specifications into a formal representation. The model integrates cognitive linguistics, knowledge based configurations & model driver engineering. It's a three step model which is invoked by taking SBVR as the starting point. It is followed by semantic analysis of business specification. Finally author [11] applies MDE by making use of MDE tools which successfully transforms executable models & codes. All three steps at subsequent levels ensure to limit ambiguity, quality & correctness. The work suggested by the author is depicted in Figure5.

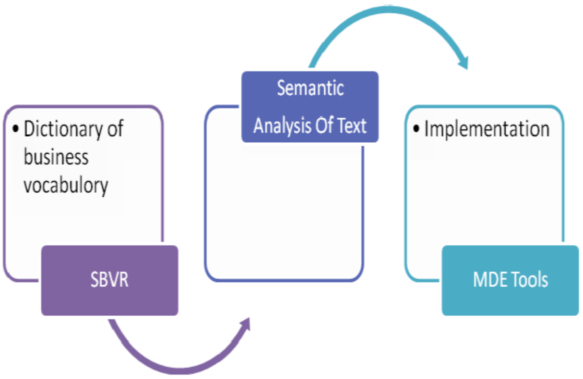

Figure 5. SBVR controlled language specification technique

2.5 NLP based approach

Although there are various NLP based techniques, One such technique is discussed by Vairalkar[12] which is parsing. The author [12] describes parsing as a program or an algorithm that analyzes natural language as input and outputs a structure in hierarchical form i.e trees which should be consistent with grammar specifications. Figure6 discusses the suggested approach by the author.

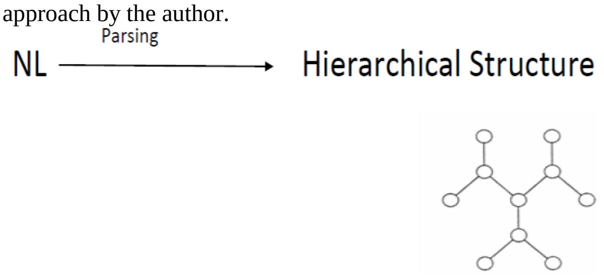

Figure6 Hierarchical structure generation from NL

Miles et. Al [13] provides some relevant inputs related to parse technique for processing NL. The author [13] develops a tool named **ANLT's** which produces all possible ambiguous sentence hence identifying ambiguity. The approach was being sampled on 750 sentences.

Fatwanto [14] in his work uses POS tagging combined with pattern recognition in order to convert NL specification into a formal representation. Besides POS tagging provides Noun analysis & Verb analysis, and uses English corpus & a reference framework. Figure 7 describes the POS tagging approach for Formal specification.

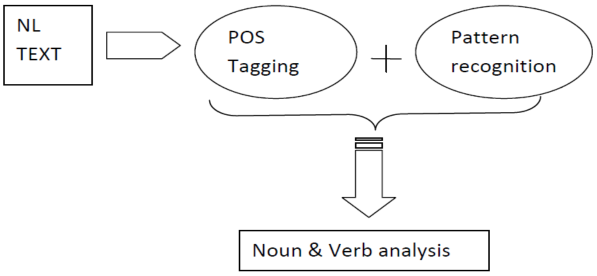

Figure7 POS tagging for formal specification

3. DISCUSSION

Observation on the word phrase level approach[8] can be extended on to complex NL linguistic structure. Object – Oriented analysis approach for processing natural language[8] reviewed in the paper is integration of various NCP techniques. It is simplistic and graphical hence provides a better understanding of the specification. Ontology population based approach [9] as discussed in the previous section is useful for various kinds of conflict Identification as it relies on both syntactic & semantic knowledge. Another advantage of using this is that it is not domain specific. A completely User driven ontology approach [10] however becomes more user centric & the desired software solution would be much to user satisfaction. SBVR model[11] for automation of business specific formalization using CL/configuration based approach aims at allowing business people maintaining business specific vocabulary & rules. As a result, the requirement engineer would have to put less effort in understanding & streamlining the requirement specifications. But this could lead to inconsistencies as it totally depends upon the degree of the understanding of the logical & lexical semantics of particular term by the client. As a result the requirement engineer's assumption of client's complete awareness of the semantics & meaning, may lead to inconsistency later on. Useful validation criteria can be derived while evaluating the results with the help of the approach suggested by Miles et.al. Parse selection on the basis of how close it is to the true parse error diagnosis resource bound, ambiguity identification etc[13]. It is observed that in the POS tagging technique [14] discussed in the earlier section addresses dynamic behaviour of Software requirements besides addressing static issues.

Table 1. Projected degree of simplicity and automation

Technique	Core concept	Degree of simplicity	Automation
Word-phrase logic Technique	Predicate logic	Medium	No
UML based knowledge formalization techniqu	Object-Oriented Analysis & Design	High	Yes
. Requirement pattern approach for formal specification	Ontology	Medium	Yes
User-driven mapping approach	Ontology	High	No
SBVR controlled language specification techniqu	Model Driven Engineering	Medium	Partial
aHierarchical structure generation from NL	Natural Language Processing	High	No
ANLT Tool based technique	Natural Language Processing	High	Yes
POS tagging	Natural Language Processing	Medium	No

Degree of Simplicity and Automation of approach are two basic parameters on the foundation of any framework suggested. Degree of simplicity defines in terms of straightforwardness, ease of use and implementation and effortlessness. Automation is referred on the basis of existence of mechanization and computerization, reducing human intervention to minimum.

The table observes the possible degree of presence of the parameters – degree of simplicity and automation on the work of authors discussed in section 2

4. CONCLUSION AND FUTURE SCOPE

Requirement engineering involves a decisive affair in the software development progression. Software requirements are slackly presented by a client as it is in natural language. Hence it is suggested that natural language specification which lacks structure should be formalized into tightly structured software requirements. Although various authors have proposed various frameworks and approaches, however it is recommended that apposite framework can me laid down that is simplistic so that it is easily comprehendible by the client. Most of the techniques require human intervention hence it is suggested that further work can be done to make approaches or frameworks automatic in nature. Graphical notations are always easier to understand both for the client as well as the developer. Hence future work can be done in this direction so that this particular feature is also incorporated every time. The frameworks should be scalable and extensible to complex linguistic structures as well. For better clarity and more fruitful result it is suggested that for a wholesome framework, rule base can be additionally incorporated that can aid in better linguistic analysis hence can provide a platform for fertile formal specification of loosely binded natural language specification. Latest trends in this subject is inclined towards using NLP techniques[5,8,9,14,15] such as POS Tagging, Parsing etc not just to model and extract UML diagrams but can be used to detect conflicts in requirements. This in combination with Supervised and unsupervised Machine learning can be used to make algorithms more accurate and prolific. Generative parsing model, probabilistic approach of machine learning can be integrated to produce better results in extracting formal specifications from more loosely specified Natural language during requirement engineering process.

5. ACKNOWLEDGMENTS

My thanks to the Dr. Sunil Sikka who have contributed towards development of the work by giving his valuable inputs.

6. REFERENCES

[1] IEEE Std 610.12-1990,1990, IEEE Standard Glossary of Software Engineering Terminology.

[2] R.Saranya, vol 90, no 17, 2014, Survey on Security Measures of Software Requirement Engineering, International journal of computer applications.

[3] Dr. SohailAsghar, Mahrukhumar, International journal of Software Engineering (IJSE), Vol 1, No. 2, August 2010, Requirement Engineering Challenges in Development of Software Applications and Selection of Customer-off-the-Shelf (COTS) Components.

[4] Didar Zowghi and Chad Coulin, Engineering and managing software requirements, Springer, 2005, Requirements Elicitation: A Survey of Techniques,Approaches, and Tools.

[5] Popescu, Rugaber, Medvidovic,Berry, ", Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs,14th Monterey Workshop 2007, Monterey, CA, USA,ISBN: 978-3-540-89777-4 (Print) 978-3-540-89778-1 (Online), September 2007 ,Reducing Ambiguities in requirement specifications via automatic created object-oriented models.

[6] Mohd Ibrahim, Rodina Ahmad , 2nd International conference on computer research and development,IEEE, 2010,Class diagram extraction from textual requirements using Natural language processing (NLP) techniques.

[7] Amjad Ali, Mohammad Abid Khan, International conference on emerging technologies, IEEE, 2009,Selecting Predicate Logic for Knowledge Representation by Comparative Study of Knowledge Representation Schemes.

[8] Shinde et.al, IJCA, 2012, NLP based Object Oriented Analysis and Design from Requirement Specification.

[9] Sadoun et.al, ICTAI, 2013, From Natural Language Requirements to Formal Specification using an Ontology.

[10] Aldo de Moor, Hans Weigand, Proceedings of the 32nd Hawaii International Conference on System Sciences – IEEE, 1999, An Ontological Framework for User-Driven System Specification.

[11] Selway et.al, EDOC, IEEE Internal conference, 2013,Formalising Natural Language Specifications using a Cognitive Linguistics/Configuration Based Approach.

[12] Vairalkar , RATMIG 2013, International Journal of Application or Innovation in Engineering & Management (IJAIEM),2013, Review on Parse Tree Generation in Natural Language Processing.

[13] Osborne et.al , 2nd international conference on requirement engineering,IEEE,1996,Processing Natural Language Software Requirement Specifications.

[14] Fatwanto, IEEE,2013, Natural Language Requirement Specification Analysis using Part-of-speech Tagging.