Issues in Developing UML Diagrams from Natural Language Text

NAKUL SHARMA
Department of Computer Science and Engineering,
K.L. University,
Vaddesvaram, Guntur Dist. Andhra Pradesh
INDIA
nakul777@gmail.com

DR. PRASANTH YALLA
Department of Computer Science and Engineering,
K.L. University,
Vaddesvaram, Guntur Dist. Andhra Pradesh
INDIA
yallaprasanth@gmail.com

Abstract: - NLP of a given text comprises of utilizing NLP tools for better understanding and application of the text. In this paper, issues in developing UML Diagrams from Natural Language Text are discussed at SE and NLP level. A strategy taking care of these issues is hence proposed and the future scope is presented.

Key-Words: - Software Engineering (S.E.), Natural Language Processing (NLP), Unified Modeling Language (UML), Natural Language Understanding (NLU), Software Development Life Cycle (SDLC), Natural Language (NL) Text.

1 Introduction

UML diagram are at critical juncture of SDLC. They are formed when the analysis phase is about to get over and design phase is starting [9]. UML diagrams are generally developed manually although some attempts have been made to develop these diagrams from natural language text [6, 7, 8, 12]. Much of the work to generate UML diagrams is focused on Use-Case diagrams and class diagram. However the specific issues in developing UML diagrams are not quite highlighted and discussed.

In the previous works undertaken, direct text has been utilized by scanning the relevant information in generating the UML diagrams [6, 7, 12]. But if the input is better or suitable for the automation tools, then quality of the diagrams which is generated can be improved. A textual use case description already exists in form of meta-model [2], and use-case template [15]. But for other UML diagram such information is not available. Hence this research is undertaken.

2 Problem Formulation

Agt H. in his work has utilized the formal language sources to generate iterative approach for language engineers. The author develops an automated knowledge acquisition tool for supporting language engineering in the early phase of SDLC [1].

Stepane S. Some has developed a meta model for textual use case description. The author utilizes the existing Use Case specification to generate a meta model having OCL constraints [2].

Matthew Hause has discussed the role of use case diagrams outside the realm of software development. The author suggests role of use case in avionics system and system engineering. The pits falls of use cases and the solutions are also presented [3].

Stephen L. Reed et.al. describe how different ontologies can be mapped onto Cyc. The authors conversed with various subject experts and onto to map ontology onto Cyc [4].

Viliam Simko et. al. have made domain model from textual specification. The authors have utilized OpenNLP and CoreNLP technologies to complete this task [5].

Reynaldo et. al. have developed class models through controlled requirements. The author accepts input as controlled N.L. text and validates with RAVEN project [6].

Sascha et. al. have discussed how requirement engineering's error get propagated to design and coding stages. The authors hence propose the automated analysis of N.L. text in SPIDER project [7].

Priyanka More et.al. have generated UML diagrams from N.L. text. The authors have utilized RAPID steaming algorithm and OpenNLP tools to accomplish this task. [8].

Sudha et.al. has described the how natural language processing of tweeted text can help in times of crises. The authors have developed a classifier which can classify tweets for human analyses in times of crises [10].

Artis et. al. have studied that UML models are inherently static. Hence they have utilized a simulation environment called ARENA for running UML Models [16].

Mathias et. al. have developed a Requirement Engineering Feedback System (REFS) that checks for consistency with the textual requirement and models [18].

Bajwa et. al. have developed class, activity and sequence diagrams from simple English sentences. They have presented a methodology called UMLG to develop these UML diagrams. The author claim that their algorithms can future be improved by introducing learning [19].

Imran S. Bajwa et. al. discusses an approach generating SVBR rules from Natural Language Specification. The paper shows the importance automation in generation SVBR indicating that business analyst with load of documents. They have developed an algorithm for detecting the semantics of English language [21].

Imran S. Bajwa et. al. highlights the cases in which Stanford POS tagger does not identify the particular syntactic ambiguities in English specifications of software constraints. A novel approach to overcome these syntactic ambiguities is provided and better results are presented [22].

Imran S. Bajwa et. al. presents a new model for extracting necessary information from the natural language text. The authors generate Use Case, Activity, Class and Sequence diagram from the natural language text. The designed system also allows generation of system from Natural Language Text [23].

Imran S. Bajwa et. al. propose a SVBR approach to generate a unambiguous representation in English language. The input text is extracted for the relevant information of SVBR. A tool named NL2SVBR*via*SBVR is made to accomplish this task [24].

Imran S. Bajwa et. al. propose an interactive tool to draw Use-Case diagrams. The authors have utilized LESSA approach for getting useful information from the Natural Language Text [25].

The following research questions were **not studied** in-depth in the current literature, so some research questions were formulated as below:-

RQ-1.What are the issues in developing UML diagrams from natural language text?

RQ-2.What is the methodology to address the issues of developing UML diagrams from natural language text?

3 Problem Solution

While trying to answer both these research questions it is pertinent to understand how Software Engineering and Natural Language Processing are interrelated to each other.
Software Engineering deals with how the software as a product will be engineered or made [9]. Natural Language Processing (NLP) deals utilizing machine or computer to better understand and process text or speech [10].

When dealing with research from both these areas it's important to address issues governing both these areas as shown in figure-1.

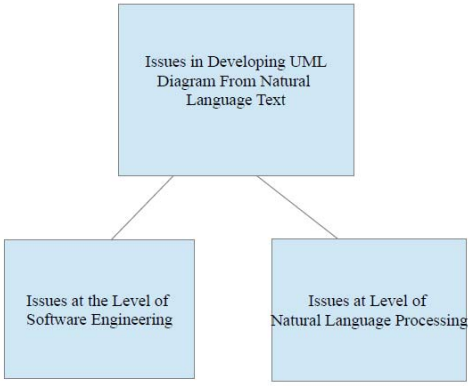

Figure-1. Issues in developing UML Diagrams

3.1 Issues at level of Software Engineering

There are following issues at the level of Software Engineering.

3.1.1 Type of UML Diagram to be generated

There are following type of specification while drawing UML diagrams:-

- Unified specification
- Booch specification
- OMT specification

These specifications only differ in the notation for various UML diagrams. For example, a Booch specification is as shown in figure-3. An OMT specification of the same diagram is as shown in figure-4.

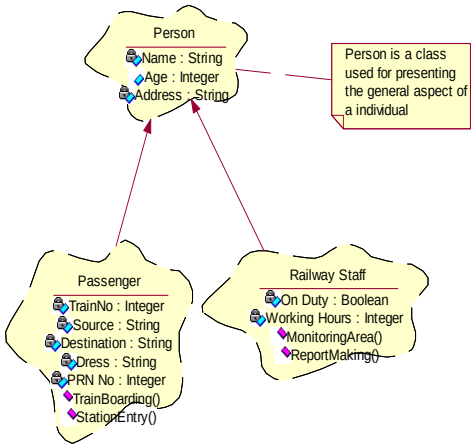

Figure-3 Booch specification

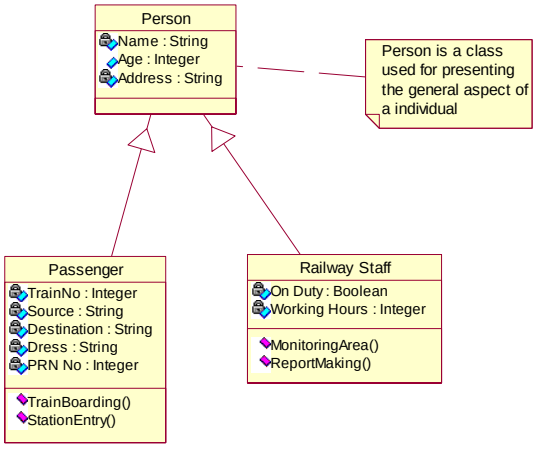

Figure- 4 OMT specification

3.1.2 Generation of Textual Information

Textual information in form of plain text or speech can prove to very critical in any stage of developing software [13]. Textual information in form of plain text needs to be generated for the entire UML diagram before any diagrams can be drawn [12]. As specification is necessary, we have developed textual description for the class, activity and component based diagrams.

3.1.3 Developing or using existing ontologies

The input text present is not alone sufficient to get the requisite amount of information for generating text [5]. The input text must be

ISBN: 978-1-61804-262-0

supported by the existing ontologies which provide relevant clarification of the input text [12].

3.1.4 Human Factors

The requirement engineer or designer who makes the UML diagrams must be well-versed with using the technical know-how of the software and the domain level information. The project can executed successfully when there is clarity in the UML diagrams developed by the humans [9]. Since the evaluator of any UML diagram is ultimately human, it is important that human factors such as understanding are also taken into account. This includes the know-how of person developing and using the system [9].

3.2 Issues at level of Natural Language Processing

3.2.1 Level of noise in a sentence

The sentence which forms input to the text of UML diagram must be free from the noise [6]. The sentence must hence be scanned with appropriate tool which gives noise free sentences.

3.2.2. Determining the complexity of sentence

An algorithm needs to design to check the complexity of a sentence. A bench-mark also needs to be created for classifying the complexity of a sentence.

3.2.3 Scanning of Textual Information for relevant information

The input text must be scanned for getting the necessary information. This done by making use of various NLP tools which are available for processing of text. It involves both semantic and syntactic processing of the text.

3.2.4 Scanning of Textual Information for ambiguity

This involves studying the input text for any ambiguous sentence and then removing those sentences.

Table 1 gives summary of these issues and also gives information if the issues are relevant to both NLP and SE domains.

Table-1 Summary of the Issues

Sr. No.	Issues	S.E. Issues	NLP Issues
1	Type of UML Diagram to generated	Yes	No
2	Generation of Textual Information	Yes	No
3	Developing or using existing ontologies	Yes	Yes
4	Human Factors	Yes	Yes
5	Level of noise in a sentence	Yes	Yes
6	Determining the complexity of sentence	No	Yes
7	Scanning of Textual Information for relevant information	Yes	Yes
8	Scanning of Textual Information for ambiguity	No	Yes

4. Textual Description for UML Class Diagrams

We have developed a textual specification for the class diagram which can be given as an input for generating class diagram. This textual description should include the following important aspects of the class diagram:-

- Name of the class
- Domain (problem/solution) being addressed by the class
- Attributes of the class
- Operations performed by the class
- Relationships which are shared by the class.

A methodology named TextToUml (*TTU*) has been been proposed keeping both the Software Engineering and Natural Language Processing issues into consideration. TTU methodology is as follows:-

1. Define a **parameter** about the quality of N.L. text.

ISBN: 978-1-61804-262-0

This involves classifying the text as in controlled language and uncontrolled language

2. Understand the issues at level of **text** such as:-
    a. Level of noise
    b. Level of complexity of a sentence in terms of controlled language and uncontrolled language.
    c. Determining the sentence in following types:-
        i. Simple
        ii. Semi-complex
        iii. Complex
3. Identifying the **type** of diagram corresponding to the description given in the text.
4. Understanding the **specification** of UML diagrams to be developed. Currently there exist three different specification as:-
    a. OMT
    b. Jacobson
    c. Booch
5. Deriving UML specification in tune with N.L. text available for all UML diagrams.
6. Developing an interface between different ontologies and application to generate UML diagram.

We have already implemented an algorithm No_REM to remove the noise of an input text [14].

5 Result Discussion and Future Scope

The UML diagrams can be generated from Natural Language Text [6, 7, 8, 12]. But the quality of the generated diagrams will depend upon how the issues in generating UML diagrams are dealt with. Based on these issues, UML diagrams can be generated from Natural Language Text with high quality.

5.1 Advantages of Current Work

The analysis of issues in generation of UML Diagrams from N.L. Text has following advantages:-

- UML diagrams generated will be good quality.
- Possibility of assessing the quality of N.L. Text.
- Possibility of assessing the quality of document generated from N.L. text.
- Help in better automation as two research areas are being addressed.
- A generic framework for addressing the interdisciplinary research can be developed.

5.2 Disadvantages of Current Work

There are following disadvantages of current work:-

- Software's for NLP are required to be downloaded.
- Quality of N.L. text makes the UML diagrams from such a text, poor.
- Issues such as human factors (level of understanding of abstraction in UML diagrams) are not studied in-depth.

6 Conclusion and Future Scope

In this paper we have discussed the important issues while trying to create UML diagrams from natural language text. The natural language text forms part of NLP and UML diagram is drawn in SE field. By such analysis, it may be possible to automate the creation of UML diagrams and also work towards achieving universal programmability [17]. The work also can be extended to check the quality of NL text, define parameter for the quality of UML diagrams generated. By addressing the issues it will be possible to generate good quality UML diagrams.

References:

[1] Agt, H, *"Supporting Software Language Engineering by Automated Domain Knowledge Acquisition"*, In.Proc. Kienzle, J. (ed.)

MODELS 2011 Workshops. LNCS, vol. 7167, pp. 4–11. Springer, Heidelberg (2012).

[2] Stepane S. Some: *"A Meta- Model for Textual Use Case Description"*, Journal of Object Technology (JOT), vol. 8, no. 7, November-December 2009, pages 87-106.

[3] Matthew Hause, *"Finding Roles for Use-Cases"*, In. Proc. IEEE Information Professional June/July 2005, Pages 34-38.

[4] Stephen L. Reed, Douglas B. Lenat, "*Mapping Ontologies into Cyc*", American Association of Artificial Intelligence, 2002.

[5] Viliam Simko, Petr Kroha, Petr Hnetynka, "*Implemented domain model generation*", Technical Report, Department of Distributed and Dependable Systems, Report No. D3S-TR-2013-03.

[6] Reynaldo Giganto, "*Generating Class Models through Controlled Requirements*", New Zealand Computer Science Research Conference (NZCSRSC) 2008, Apr 2008, Christchurch, New Zealand.

[7] Sascha Konrad, Betty H.C. Cheng, "*Automated Analysis of Natural Language Properties for UML Diagrams*".

[8] Priyanka More, Rashmi Phalnikar, "*Generating UML Diagrams from Natural Language Specifications*", International Journal of Applied Information Systems, Foundation of Computer Science, Vol-1-No-8, Apr-2012.

[9] Pressman R.S., "*Software Engineering A Practitioner's Approach*", TMH Publishing House, 7th addition.

[10] Sudha Verma, Sarah Vieweg, William J. Corvey, Leysia Palen1, James H. Martin1, Martha Palmer, Aaron Schram1 & Kenneth M. Anderson, "*Natural Language Processing to the Rescue?: Extracting "Situational Awareness" Tweets During Mass Emergency"*, In. Proc. Association for the Advancement of Artificial Intelligence, 2011.

[11] [Online] http://rationalrose.com

[12] Mathias Landhauber, Sven J. Korner, Walter F. Tichy, "*From Requirements to UML Models and Bac: how automatic processing of text can support requirements engineering"*, Software Quality General, March 2013, Vol 22, Issue 1, pp 121-149.

[13] Dr. Prasanth Yalla, Nakul Sharma, "*Combining Natural Language Processing and Software Engineering"*, In. Proc. International Conference in Recent Trends in Engineering Sciences(ICRTES-2014), Organized by Elsevier and IET, March ISBN-978-93-5107-223-2.

[14] Dr. Prasanth Yalla, Nakul Sharma, "*Parsing Natural Language Text of Use Case description"*, In.Proc. International Conference on IT in Business, Industry and Government (CSIBIG-2014), March 8-9, 2014. ISBN- 978-1-4789-3063-0.

[15] A. Cookburn, "*Writing Effective Use Cases"*, Addison Wesley, 2001.

[16] Artis Teilans, Yuri Merkuryev, Andris Grinbergs, "*Simulation of UML models using ARENA"*, In. Proc. 6th WSEAS International Conference on SYSTEM SCIENCE and SIMULATION IN ENGINEERING, Venice, Italy, Nov 21-23, 2007. Pg No. 190-195.

[17] Walter F. Tichy, Mathias Landhabuer, Sven J. Korner, *"Universal Programmability- How AI Can Help ?"*,In Proc. 2nd International Conference NFS sponsored workshop on Realizing Artificial Intelligence Synergies in Software Engineering, May 2013.

[18] Mathias Landhauber, Sven J. Korner, Walter F. Tichy, "*From Requirements to UML Models and Back How Automatic Processing of Text Can Support Requirement Engineering"*, Springer's Software Quality Journal, March 2014, Vol. 22, Issue 1, Pages 121-149.

[19] Imran Sarwar Bajwa, M. Abbas Choudhary, "*Natural Language Prcoessing Based automated system for UML Diagrams generation"*, In. Proc. 18th National Computer Conference 2006, Sandi Computer Society.

[20] Oksana Nikiforova, Olegs Konstantins Gusarovs, Dace Ahilcenoka, Andrejs Bajovs, Luddmila Kozacenko, Nadezda Skindere, Dainis Ungurs, "*Development of BRAINTOOL from the two-hemisphere model based on the two-hemisphere Model transformation itself"*, In. Proc International Conference on Applied Information and Communication Technologies (AICT 2013), 25-26 April 2013, Latvin. Page 267- 274.

[21] Imran S. Bajwa, Mark G. Lee, Behzad Bordbar, *"SVBR Business Rules Generation from Natural Language Specification"*, In. Proc. Artificial Intelligence for Business Agility-Papers from AAAI 2011 Spring Symposium(SS-11-03), Pg. No. 2-8.

[22] Imran S. Bajwa, Mark Lee, and Behzad Bordbar, "*Resolving Syntatic Ambiguities in Natural Language Specification of Constraints"*, In. Proc. CICLing 2012, Lecture Notes in Computer Science (LNCS) 7181, pp-178-187, 2012. Springer-Verlag, Heidelberg, 2012.

[23] Imran S. Bajwa, M. Imran Siddique, M. Abbas Choudhary, *"Rule Based Production Systems for Automatic Code Generation in Java"*, In. Proc. International Conference on Digital Information Management-ICDIM 2006, Banglore, India.

[24] Imran Sarwar Bajwa, M. Asif Naeem, Ahsan Ali Chaudhri, Shahzad Ali, *"A Controlled Natural Language Interface to Class Models"*, In. Proc. 13th International Conference on Enterprise Information Systems, Page No- 102-110. DOI: 10.5220/0003509801020110, Science and Technology Publications, SciTePress, Pakistan.

[25] Imran Sarwar Bajwa, Irfan Hyder, *"UCD-Generator A LESSA Application for Use Case Design"*, In. Proc. IEEE-International Conference on Information and Emerging Technologies, IEEE-ICIET, Karachi-Pakistan.