Class Diagram Extraction Using NLP

Bhagat S. B. Kapadni P. R. Kapadnis N. Patil D. S. Baheti M. J.

***Abstract*— The automation of class generation from natural language requirements is highly challenging. Requirements engineers analyze requirements manually to come out with analysis artifacts such as class diagram. The time spent on the analysis and the low quality of human analysis proved the need of automated support .In this paper requirements analysis process and class diagram extraction from textual requirements using natural language processing(NLP) and domain Ontology techniques.**

***Index Terms*—About four key words or phrases in alphabetical order, separated by commas.**

I. INTRODUCTION

The common way to express requirements is with large volumes of text [1] which can be referred to as natural language (NL) requirements. NL requirements are typically coming from a pool of natural language statements which are gathered from interview excerpts, documents and notes. Due to the inherent ambiguity of natural language, it is often difficult to prove properties on NL requirements [2]. For this reason, Informal natural language requirements are better to be expressed as formal representations.

"Natural Language Processing (NLP) is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things [3]. The NLP systems use different levels of linguistic analysis: Phonetic (phonological) level, Morphological level, Lexical level, Syntactic level, Semantic level, Discourse level and Pragmatic level and analysis. NLP has several application tasks among these: Information Retrieval/Detection, Information Extraction, Question Answering Tasks,

And Text Understanding (Artificial Intelligence).User requirement analysis is an Information Extraction (IE) application of NLP. It is the identification of specific semantic elements within the user's requirements entered in textual form (i.e. entities, attributes, relationships, cardinalities and multiplicities)." The use of NLP and domain ontology techniques for the extraction of UML class diagram from informal natural language requirements by implementing a prototype tool that uses the mentioned techniques.

II. RELATED WORK

The class diagram extraction from natural language (NL) requirements. In this works that use NLP or domain ontology techniques to analyze NL requirements, and the works to extract class diagram.

A. GOOAL

It was automatically generate object models from natural language text its prototype tool GOOAL [4] produces OO static and dynamic model views of the problem.

Language pronunciation Barrier and It is not helpful for dumb people.

B. K. Li

K. Li also presented his work to solve problems related to NL that can be addressed in OOA [5].Need of a NL-based CASE tool.

C. REBUILDER UML

This tool integrates a module for translation of natural language text into an UML class diagram. This module uses an approach based on Case-Based Reasoning and Natural Language Processing [6].

This case tool needs continuous up gradation of case-base and only deals with class diagrams.

D. LOLITA

It generates an object model from NL text. LOLITA [7] only identifies objects from NL text. It cannot distinguish between classes, attributes and their respective attributes.

E. CM-BUILDER

This CASE tool was restricted to create a primary class model [8]. There was no appropriate mechanism for confining objects from NL text.

F. MOVA

This models, measures and validates the UML class diagrams and to help out the designed system to identify classes, objects and their respective methods and attributes [9]. Incapable of automatically identifying OO constituents.

III. PROPOSED APPROACH

To design a theory that can comprehensively analyze the natural language text and then implement the theory to develop a software tool UMLG (UML-Generator) proposed.

UMLG [10] can extract the required information from given piece of natural language text and then afterwards, transform this information into UML class diagram. An additional facility was also provided in the software that it can also convert the user modeling information into the blocks of programming source code. Code generation was made available in two languages; Java and VB.Net. An

Integrated Development Environment has also been provided efficient Input and output handling.

After studying many researches papers and also addressed many problems but the one's mentioned above are the major issue so the proposed system will try to solve many problems related with No one from these tools is able to extract the complete information i.e. classes, objects and their respective, attributes, methods and associations.

In order to help the Business rules Extraction, Speech Language Processing, Self-Organizing Map, Web information Extraction and to get to extract classes, objects and their respective, attributes, methods and associations.

A. *Algorithm for Class Diagram Generation*

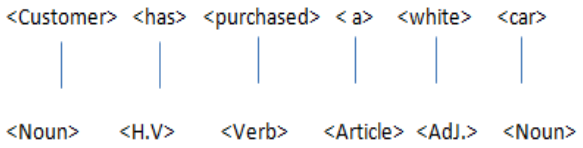
A rule based algorithm was written to analyze NL text and then extract various OO modeling elements.

Steps:

1. In first step, UMLG reads and tokenizes the text containing software requirements by the user

E.g. the output of a sentence "The Cow has four legs." is [Customer] [has] [purchased] [a] [White][Car] [.]

2. In second step, morphological analysis is performed of given text to define the structuring and Transformation of the words. POS Tagging is also performed to identify different parts of speech.



The test is lexical and syntactically analyzed and a parse tree is generated for semantic analysis.

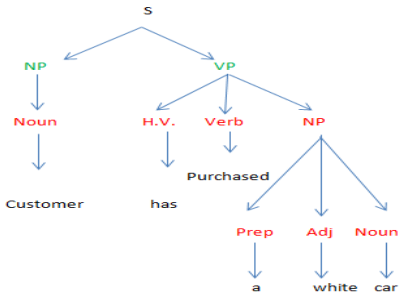
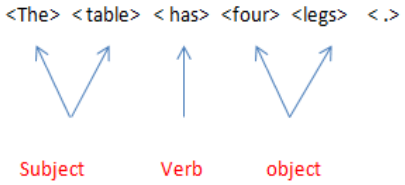


Fig 1: Parse tree generated for the example

3. Syntactic Analysis carried out to validate phrases and sentence according to grammatical rules defined by the English language. This step also helps in identifying the main parts of a sentence; Object, subject, actions, attributes, etc.



4. In this step, associations are identified by doing semantic analysis. It is determined in this specified that which actions have been performed by which object and a set of attributes belong to which object. Four Legs Cow.

5. Then a rule based module specifies subject nouns as objects, verbs as methods of the objects, and adjectives as attributes of the object. Object nouns are sometimes specified objects and Sometimes as attributes.

6. In this step associations and relationships among extracted classes and objects are performed.

Prepositions are major tool for identifying relationships and associations.

7. A logical model of the class diagrams is generates on the basis of previously extracted Information.

8. A drawing module converts the logical model into the class diagrams by connecting small pieces of images already stored in database.

9. In next step, associations among generated class diagrams will be also produces.

10. After generating class diagrams, diagrams are labeled with appropriate labels.

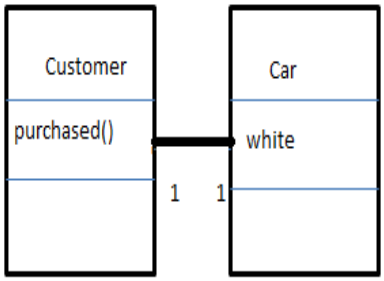


Fig2: Final Class Diagram Generated

11. The final step is conversion of logical model to VB.NET and Java Coding·

IV. MATHEMATICAL MODEL

Let fT is the rule of analysis text i/p to get the i/p information

fT (T) -> I/ØI S= Set of source code generated

= {S1, S2, S3…Sn/ Øs}

A= {T, I, C, S}

T= Set of text input information

= {T1, T2, T3…..Tn / ØT}

I =Set of information obtain

= {I1, I2,I3…….…..In/ ØI}

C= Set of classes of user diagram

= {C1, C2, C3……Cn/Ø c}

Let fI is the rule of generation of class diagram from i/p information

fI (I) -> C/ØC Let fC is the rule of source code from class diagram i/p information

fI (I) -> C/ØC.

V. OVERVIEW OF SYSTEM

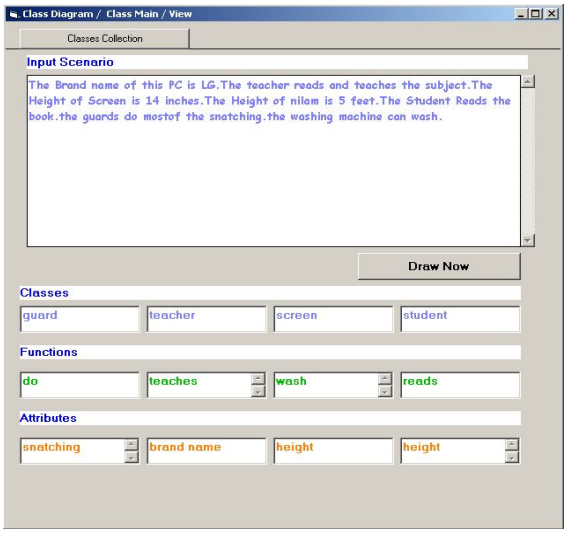


Fig 3: Extracting classes, functions and their attributes

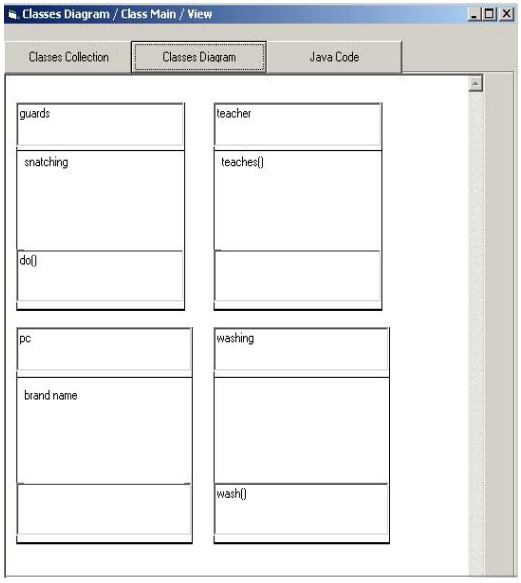


Fig 4: Generation of class diagrams

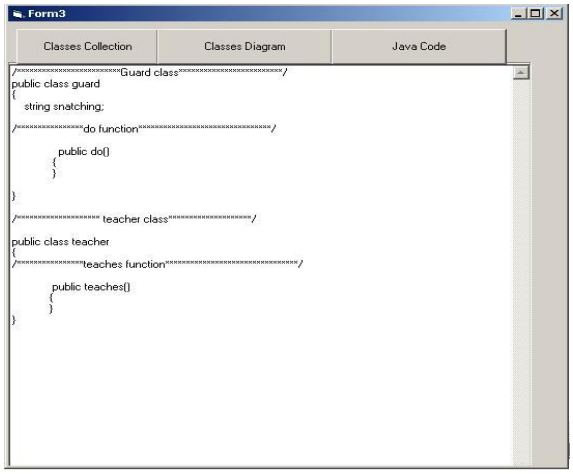


Fig 5: Generating Code in Java Language

VI. RESULT & DISCUSSION

The Class Diagram Extraction using NLP uses above Algorithm. Generation of class Diagram from Input Information. If the system doesn't have the Internet connection the system cannot be completed in polynomial time. Hence, we can say that the system belongs to NP Class. As the system acquires the internet connection the system satisfies the CNF-SAT theorem and reduces to Polynomial time. Hence, we can say that the system belongs to NP Complete.

VII. CONCLUSION

We have proposed a tool to facilitate requirement analysis process and class diagram extraction .This tool has been found to have O (n2) complexity and also the proposed algorithm has found to extract the class diagram within NP-Complete which is very less. In future, system would be implemented for generating different UML diagrams and then create corresponding source code in programming languages like VB.net.

REFERENCES

[1] Booch, G. (1994). Object-Oriented Analysis and Design with Applications.

[2] Ambriola, V. and Gervasi, V. "Processing natural languagerequirements"

[3] Elizabeth D. Liddy & Jennifer H. Liddy, 2001, "An NLP Approach for Improving Access to Statistical Information for the Masses".

[4] Hector G. Perez-Gonzalez, (2002) "Automatically Generating Object Models from Natural Language Analysis", 17th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, ACM New York, USA, pp: 86 – 87.

[5] K. Li, R.G.Dewar, R.J.Pooley, [2003] "Object-Oriented Analysis Using Natural Language Processing", www.macs.hw.ac.uk:8080/techreps/ docs/ files/HWMACS-TR-0033.pdf

[6] António Oliveira, Nuno Seco and Paulo Gomes (2004) "A CBR Approach to Text to Class Diagram Translation", TCBR Workshop at

the 8th European Conference on Case-Based Reasoning, Turkey, September 2006.

[7] L. Mich, R. Garigliano, (1996) "A linguistic approach to the development of object-oriented system using the NL system LOLITA", Object Oriented Methodologies and Systems, (ISOOMS), LNCS 858, pp. 371-386.

[8] H. M. Harmain and R. Gaizauskas, (2003) "CM-Builder: A Natural Language-based CASE Tool", Journal of Automated Software Engineering, 10, 2003, pp. 157-181.

[9] Manuel Clavel, Marina Egea, Viviane Torres da Silva, (2007) "The MOVA Tool: A Rewriting-Based UML Modeling, Measuring, and Validation Tool", in Proc. 12th Conference on Software Engineering and Databases Zaragoza (Spain), 2007.

[10] Imran Sarwar Bajwa,Ali Samad,Shahzad Mumtaz " "Object Oriented Software Modeling Using NLP Based Knowledge Extraction" *European Journal of Scientific Research* ISSN 1450-216X Vol.35 No.1 (2009), pp 22-33 © Euro Journals Publishing, Inc. 2009 Available: http://www.eurojournals.com/ejsr.htm

AUTHOR'S PROFILE

BHAGAT SUJATA is studying in B.E. in Department of Computer Engineering, SNJB's College of Engineering, Chandwad, Nashik.
Sujatabhagat28@gmail.com

KAPADNI PRIYANKA is studying in B.E. in Department of Computer Engineering, SNJB's College of Engineering, Chandwad, Nashik.
priyankakapadni@gmail.com

KAPADNIS NIKETAN is studying in B.E. in Department of Computer Engineering, SNJB's College of Engineering, Chandwad, Nashik.
kapadnisnik@gmail.com

PATIL DHANASHRI is studying in B.E. in Department of Computer Engineering, SNJB's College of Engineering, Chandwad, Nashik.
patildhanashri494@gmail.com

MAMTA BAHETI has received M.Sc. Degree in Computer Science from Sant Gadge Baba Amravati University in 2003. Since 2007, she has been a Ph.D. student of Dr. K. V. Kale. Her current research interests include pattern recognition, image analysis and document processing. She is presently working at Department of Computer Engineering, Chandwad, Nashik.