

Object Oriented Analysis using Natural Language Processing concepts: A Review

Abinash Tripathy^a, Santanu Kumar Rath^a

^aDepartment of Computer Science and Engineering, National Institute of Technology, Rourkela, Odisha, India, Contact: abi.tripathy@gmail.com, skrath@nitrkl.ac.in

The Software Development Life Cycle (SDLC) starts with eliciting requirements of the customers in the form of Software Requirement Specification (SRS). SRS document needed for software development is mostly written in Natural Language(NL) convenient for the client. From the SRS document only, the class name, its attributes and the functions incorporated in the body of the class are traced based on pre-knowledge of analyst. The paper intends to present a review on Object Oriented (OO) analysis using Natural Language Processing (NLP) techniques. This analysis can be manual where domain expert helps to generate the required diagram or automated system, where the system generates the required diagram, from the input in the form of SRS.

Keywords : Natural Language, Natural Language Processing, Object Oriented, Parts Of Speech, Software Development Life Cycle, Software Requirement Specification.

1. INTRODUCTION

Software Requirement Specification (SRS) document forms the basis of problem analysis between client and developer. SRS needs to be very specific, while serving as a basis, to proceed towards implementation of desired software. It is very often observed that SRS is expressed in any natural language as comprehensible by the client. But it may be ambiguous, possibly inconsistent, and probably unmanageably large from the software analyst's point of view.

Identifying major functionalities from the OO analysis point of view plays an important role in project success. The use of formal languages like Unified Modeling Language (UML) have been applied to avoid the inherent problems of natural language such as incompleteness and ambiguity [1]. Earlier analysis was used to help for an explanatory model called as build and fix programming style. But this style was observed to be very informal and there are no set of rules as to which one is superior. Every programmer formulates his own software development technique solely guided by his expertise

and in his own language and style [2].

In recent years, the object-oriented software development style is a preferred style over conventional style by developers as the present day software development languages are object oriented in nature. Hence, OO analysis of software helps to find out the candidate for class, function, and the attributes associated with those classes.

Natural Language Processing (NLP) combines the effect of computer science and linguistics branch which are concerned with the interaction between the computer and human languages [3]. Natural Language generation systems mostly extracts right information from statements which are in human readable form. The aim of the work is to present a review on existing literature of application of NLP in Object Oriented Analysis (OOA) based on literature available such as: Abbott [5], Saekai and Enamoto [6], Nanduri and Rugaber [7], Juristo and Moreno [8], Popescu *et al.*, [9], Ibrahim and Ahmed [10], Harmain and Gaizauskas [11], Overmyer and Rambow [12], Mich [13]. Among these literatures few authors suggested auto-

classes, the relation between the subject and object of a sentence is found out.

- For other rules like multiplicity determines are used that specify the relationship like one-one, one-many, many-one, many-many.

4. CONCLUSIONS AND FUTURE SCOPE

There are different tools that have been developed to analyze the text; but as there is no exhaustive dictionary which helps to provide POS for each words. Although few tools generate the class diagram but different authors suggest that a manual intervention is needed to improve the final result. Until and unless there is specific rules for writing the SRS document, the ambiguities continue to be present in it and that cause issue in compiling the SRS. Though many approaches have been proposed and also are used to obtain the elements of OO analysis still there is scope for research in this area. To automated understanding the SRS written in informal NL is also an issue in research.

REFERENCES

1. J Rumbaugh, I Jacobson and G Booch. Unified Modeling Language Reference Manual, *Pearson Higher Education*, 2004.
2. R S Pressman. Software Engineering: A Practitioner's Approach, *McGraw-hill New York*, 7, 2010.
3. E Kumar. Natural Language Processing, *IK International Pvt Ltd*, 2011.
4. K S Jones. Natural Language Processing: A Historical Review, in *Current Issues in Computational Linguistics: in Honour of Don Walker*, Springer, pages 3–16, 1994.
5. R J Abbott. Program Design by Informal English Descriptions, *Commun. ACM*, 26(11):882–894, Nov. 1983.
6. M Saeki, H Horai and H Enomoto. Software Development Process from Natural Language Specification, in *Proceedings of the 11th International Conference on Software Engineering*, ser. ICSE '89, New York, NY, USA: ACM, pages 64–73, 1989.
7. S Nanduri and S Rugaber. Requirements Validation *via* Automated Natural Language Parsing, in *Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences*, 3, IEEE, pages 362–368, 1995.
8. N Juristo, A M Moreno and M L ópez. How to Use Linguistic Instruments for Object-Oriented Analysis, *IEEE software*, 17(3):80–89, 2000.
9. D Popescu, S Rugaber, N Medvidovic and D M Berry. Reducing Ambiguities in Requirements Specifications *via* Automatically Created Object-Oriented Models, in *Innovations for Requirement Analysis. From Stakeholders Needs to Formal Designs*, Springer, pages 103–124, 2008.
10. M Ibrahim and R Ahmad. Class Diagram Extraction from Textual Requirements using Natural Language Processing Techniques, in *Proceedings of IEEE 2010 Second International Conference on Computer Research and Development*, pages 200–204, 2010.
11. H M Harmain and R Gaizauskas. Cm-builder: An Automated NL-based Case Tool, in *Proceedings of 15th IEEE International Conference on Automated Software Engineering*, pages 45–53, 2000.
12. S P Overmyer, B Lavoie and O Rambow. Conceptual Modeling through Linguistic Analysis using Lida, in *Proceedings of the 23rd International Conference on Software Engineering*, IEEE Computer Society, pages 401–410, 2001.
13. L Mich and R Garigliano. NL-oops: A Requirements Analysis Tool Based on Natural Language Processing, in *Proceedings of Third International Conference on Data Mining Methods and Databases for Engineering, Bologna, Italy*, 2002.
14. A D Booth. Machine Translation, *North-Holland Publishing Company*, 1967.
15. J Rumbaugh, M Blaha, W Premerlani, F Eddy, W E Lorensen *et al.*. Object-oriented Modeling and Design. *Prentice-hall Englewood Cliffs, NJ*, 199, 1991.
16. F N Paulisch and W F Tichy. Edge: An Extendible Graph Editor, *Software: Practice and Experience*, 20(1):S63–S88, 1990.
17. M Jackson. Developing Ada programs using the Vienna Development Method, *Software: Practice and Experience*, 15(3):305–318, 1985.
18. R Gaizauskas, K Humphreys, H Cunningham and Y Wilks. University of sheffield: Description of the Lasie System as used for muc-6, in *Proceedings of the 6th Conference on Message*

Understanding, Association for Computational Linguistics, pages 207–220, 1995.

19. R E Callan. Building Object-Oriented Systems: An Introduction from Concepts to Implementation in C++, *Computational Mechanics*, 1994.



Abinash Tripathy is currently pursuing his Ph.D at National Institute of Technology, Rourkela. He obtained his Master degrees, M.Sc Computer Science from Utkal University, Bhubaneswar and M.Tech Computer Science and Engg. from KIIT University, Bhubaneswar. His research interest are Software Testing,

UML, Natural Language Processing and Sentiment

Analysis.



Santanu Kumar Rath is a Professor in the Department of Computer Science and Engineering, NIT Rourkela since 1988. His research interests are in Software Engineering, System Engineering, Bioinformatics, Natural

Language Processing and Management. He is a Senior Member of the IEEE, USA and ACM, USA and Petri Net Society, Germany.