Classifying Natural Language Text as Controlled and Uncontrolled for UML Diagrams

Nakul Sharma
Research Scholar, Dept. of Computer Science and Engineering
K L University,
Vijayawada, India

Prasanth Yalla
Professor, Dept. of Compute Science and Engineering,
K L University,
Vijayawada, India

***Abstract*—Natural language text fall within the category of Controlled and Uncontrolled Natural Language. In this paper, an algorithm is presented to show that a given language text is controlled or uncontrolled. The parameters and framework is provided for UML diagram's repository. The parameter for controlled and uncontrolled languages is provided.**

Keywords—Natural Language Processing; UML Diagrams; Software Engineering

I. Introduction

Natural Language processing is using computer to process text which is readable and understood by humans. The processing in NLP requires natural language text to be given as input [1]. This input is can be classified as controlled or uncontrolled. NLP techniques are more effective in case of controlled languages instead of uncontrolled languages. The controlled language can be in form of plain text or any formal specification such BPNF. The authors had developed code for the removal of noise in the given the text [2].

In this work, Classifying Controlled Language (CCL) methodology is proposed. CCL methodology takes as input a natural language input and verifies using wordnet, framenet and StanfordNLP parser for checking if the language is controlled or not. The controlled languages has host of advantages which are missing in the uncontrolled language. This work is in continuation with the TextToUML (TTU) methdology presented earlier [3].

Our contribution is hence given as following:-

1) Providing a sample textual description for the class and activity diagrams.
2) Classifying the textual description as controlled and uncontrolled for UML Diagrams.

II. Problem Definition

The problem relates to giving the appropriate input for generating UML diagrams. There have been considerable efforts or converting text to UML diagrams [6-8]. The textual description for class and activity diagrams is not presented by any researcher.

Use of textual description is extensively done for use-case diagrams [4]. The simple text in any natural language is understood by large population of people if they are native to it. In order to achieve universal programmability, use of text [5] can be done as it can be understood by humans. This implies everyone gets a chance to code while everything else being done by the computer.

Table-1 provides list of papers which have usage of textual description in UML diagrams or in related fields.

TABLE. I. Textual Description in UML Diagram

Sr. No.	*Title*	*Topic Discussed*	*Domain*
1	Implemented domain model generation [6]	UML Diagram Generati-on using NLP Software	Software Design , Requirement Engineering and NLP
2	Requirement Specifications Using Natural Languages [7]	Software Require-ments using Textual specification	Software Analysis Specification
3	Generating Natural Language Specifications from UML Class Diagrams [8]	UML Diagram generation using NLP Software	Software Design Specification
4	Generally class models through controlled requirements [9]	UML Diagram generatin-g using NLP Software	Software Analysis Specification, Software Design Specification
5	Generating UML Diagrams from Natural Language Specification [10]	UML Diagram generati-n using NLP software	Software Design Specification
6	Natural Language Processing based automated system for UML Diagrams generation [11]	UML Diagram generation using NLP software	Software Analysis Specification, Software Design Specification
7	SVBR Business Rules Generation from Natural Language Specification[11]	Requirement gathering using NL specification	Software Analysis Specification, Software Design Specification

There are following benefits of using textual description for the UML diagrams:-

1) Natural Language Text is known to humans.
2) Natural Language Text is understood by humans.

3) Subject to less interpretation and speculation if the text is controlled.
4) Understood by large number of audience.
5) Possible to derive many applications of automation.
6) More comfort of understanding
7) Most convenient form of expressing information.

Figure-1 lists the advantages of textual description. The level of comfort is more for a human readable text. The level of comfort goes down as the specification moves away from the human readable text. The machine readable code is most difficult to interpret and understand.

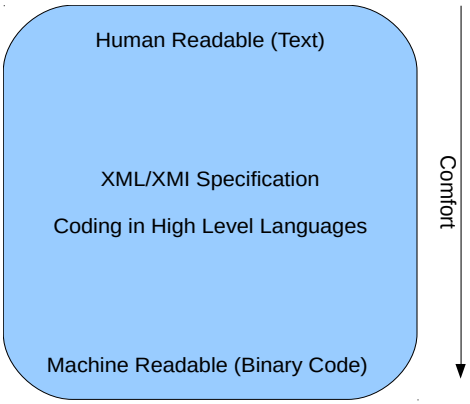

Fig. 1. Advantages of Textual Description

The following research questions were not extensively addressed in the literature review:-

RQ-1. What are the textual descriptions which describe the UML diagrams?

RQ-2. Is it possible to provide a textual specification for the entire UML diagrams?

RQ-3. How can Natural Language Text be classified as Controlled and Uncontrolled for UML Diagrams ?

The previous literature did not focus on the classifications according the domain level [13,14].

III. Problem Solution

A. Textual Description of UML Diagrams

Cockburn et.al. provides a use case template which is exhaustive example of an textual description [4]. While the use case template is specific only to use case diagrams no other diagrams have such description. UML diagram are developed when the analysis phase is about to finish and design phase gets started. Generally, the phases in SDLC are always overlapping and hence it is not feasible to tell when the analysis phase has started and when it will finish. Textual description allows automation to be made possible in early phases of SDLC. This can be made possible by generating textual descriptions of all the different artefacts, processes, methods, tools in SDLC.

B. UML Diagram Basic Usage

There exists a textual specification in form of textual use case template which is helpful in generating the use case UML Diagram [4]. Use case diagram generation is hence easy to automate. But there is no other description available. This explanation answers the research question- RQ-1. For answering the second question (RQ-2), we gave a sample textual description for the class and activity diagrams. The description was evaluated using StanfordNLP parser.

Textual Description - Class Diagram

A class diagram consists of two main features as [15]:-

1) Components of the class
   a) These are names, variables and operations.
2) Relationships with other classes.
3) These include relationships such as dependency, realization etc.

The class description should describe all the situations in which the class participates. The class diagram can be drawn from both the problem as well as solution domain. Hence, it has classes in both problems as well as solution domain must be properly described so that the problem can be automated. A sample description of the class diagram containing the above mentioned variables is available in Annexure A.

Textual Description – Activity Diagrams

1) An Activity diagram mainly focuses on the behavior of a set of objects [15]. A textual description for activity diagram should hence contain following parts:-
   a) Objects
      It includes the names of objects.
   b) Associated workflows.
2) A sample description of the activity diagram containing the above mentioned variables is available in Annexure A.

3) The textual description is first parsed using the Stanford NLP parser. The result is classified as Controlled Language and Uncontrolled language for the generation of UML diagrams.

Rules for Use Case Diagrams

For the input scanned using Stanford Parser:-

1) check the occurrence of/index of NN and NNP words these may constitute use case.
2) If it follows Subject Verb Object, the subject becomes actor.
3) If a sentence contains occurance of NNS, NNS becomes check the hierarchy in wordnet.
4) If in a sentence VB occurs get its index as it is a potentially a use case.

The above rules are applied for generating text ready to be fed into the system.

Rules for Class Diagrams

For the input scanned using Stanford Parser

1) check the occurance of/index of NN and NNP words these may constitute Class.
2) If it follows Subject Verb Object, the subject becomes class.

3) If a sentence contains occurance of NNS, NNS becomes check the hierarchy in wordnet.

4) If in a sentence VB occurs get its index as it is a potentially a use case.

Multiple sentences containing the same subjects are not considered for developing classes.

The whole process is divided into pre and post-processing of the text as shown in Figure-2.

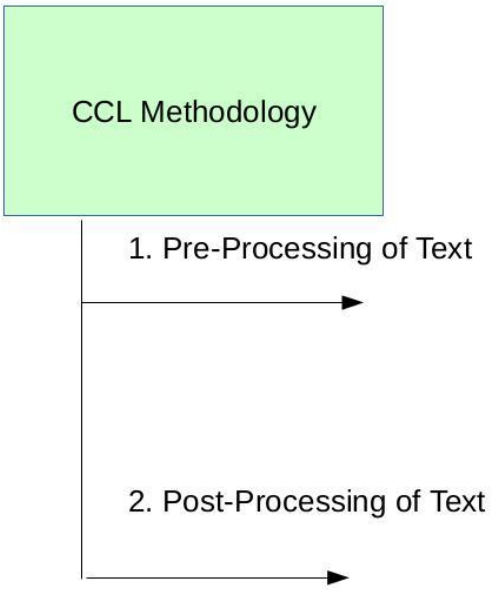

Fig. 2. CCL Methodology Steps

The CCL methodology encompasses two important phases: pre-processing and post-processing of text.

TABLE. II. SOFTWARE USED FOR PRE-PROCESSING STAGE

Sr. No.	*Operation Carried Out*	*Software's Used/Methodology used*	*Explanation*
1	Cleaning of Text	nltk	Cleaning involves removing of the special characters
2	Stemming	Algorithm Used	Stemming involves removing the stop words and reducing it to the base form.
3	Syntactic Feature Extraction	Stanford Parser, WordNet, Basic Rules	Parser parses the text be developing dependency tree and the result given to wordnet for evaluation
4	Subject/Domain Level Feature Extraction	Named Entity Recognition	For extracting person, names, nouns,

The table-II shows the software's used in the pre-processing stage of CCL methodology along with the explanation of the work done at each level.

TABLE. III. SOFTWARE USED FOR POST-PROCESSING STAGE

Sr. No.	*Operation Carried Out*	*Software's Used/Methodology used*	*Explanation*
1	Lookup Relevance	UML Repository	A Repository for activity, class and use case diagram is created along with its glossary for the purpose of checking the candidate actors, activities. (Annexure A)
2	Diagram Ready Text	UML Repository + Implementation Rules	Regenerating text to be fed into system for UML Diagram generation

IV. APPLICATION OF WORK

This work can be applied to several approaches which utilize textual information as input:-

1) Human User Textual Notation (HUTN) is a recent specification given by Object Management Group (OMG). That specification is also in textual format.[16]. The CCL methodology can be applied to it.

2) Software Requirement Specification, Requirement Document, Use Case Description, Acceptance Test Cases are the artifact on which CCL can be applied in analysis phase.

3) Software Design Specification, UML Diagrams, Test Cases are the artifact on which CCL can be applied.

4) Test cases and test manuals are the areas in which CCL algorithm can be applied.

5) Maintenance logs, user complaints, customer executive logs are also areas in which CCL can be applied.

6) UML Lookup Repository can be expanded to include other domains in Software Engineering for the purpose of generating useful information or for automation.

ACKNOWLEDGMENT

The authors are indebted to the HoD, faculties and staff of Computer Science and Engineering Department at K L University for giving the requisite amount of resources in making this research paper. The authors are also indebted to their forgiving all support and help in life.

Annexure A

Class diagrams Sample Descriptions

A person is a living entity. A person plays indoor and outdoor games. Acceptance is another operation of a person. A person goes to sleep. A passenger is special case of a person. A passenger buys a ticket from source to destination station. A person feeds his name address into the reservation form. Transaction details are also fed by the passenger.

Activity diagrams Sample Descriptions

A passenger goes to a railway counter. The passenger books the tickets by filling the source and destination details and by paying the transaction fees. The passenger boards the

train ticket for which ticket is reserved. The passenger does not board any other train and reaches the destination station.

UML Repository Table

The repository was developed using the basic information pertaining to the UML Diagram.

		Likely Related To		
Sr. No.	***Words***	***Use Case Diagrams***	***Class Diagram***	***Activity Diagram***
1	Abstract	Y	Y	Y
2	Abstract Class	-	Y	-
3	Abstract operation	-	Y	Y
4	Abstraction	Y	Y	Y
5	Action Sequence	-	-	Y
6	Action State	-	-	Y
7	Activation	-	-	-
8	Activity Diagram	-	-	Y
9	Active Class	-	Y	-
10	Active Object	-	-	Y
11	Activity	-	-	Y
12	Activity Final	-	-	Y
13	Actor	Y	Y	-
14	Aggregation	Y	Y	-
15	Artifact	Y	Y	Y
16	Association	Y	Y	-
17	Association Class	Y	Y	-
18	Attribute	Y	Y	-
19	Cardinality	Y	Y	-
20	Class	-	Y	-
21	Class Diagram	Y	Y	-
22	Classifier	-	Y	-
23	Collaboration	-	Y	-
24	Components	-	Y	-
25	Constraint	Y	Y	-
26	Dependency	Y	Y	Y
27	Encapsulation	Y	Y	Y
28	Expansion	Y	Y	Y
29	Extend	Y	Y	-
30	Final	-	Y	-
31	Flow			
32	Fork	Y	Y	-
33	Generalization	Y	Y	-
34	Generalization Tree	Y	Y	-
35	Guard	Y	-	-
36	Inheritance	-	Y	-
37	Initial node	-	Y	-
38	Interface	-	Y	-
39	Join	-	-	Y
40	Link	-	-	Y
41	Merge	-	Y	-
42	Message	Y	Y	-
43	Metadata	-	Y	
44	Metamodeling	-	Y	-
45	Modeling	-	Y	-
46	Multiplicity	-	Y	-
47	Namespace	-	Y	-
48	Navigable	Y	Y	-
49	Object Constraint Language	-	Y	-
50	Object Diagram	Y	Y	Y
51	Operation	Y	Y	
52	Package	-	-	-
53	Realization	Y	Y	-
54	Request	-	-	-
55	Role	Y	Y	-
56	Scenario	Y	Y	-
57	Sequence Diagram	-	-	-
58	State	-	-	Y
59	Static	Y	-	-
60	Sterotype	-	Y	-
61	Structure	-	Y	-
62	Superstate	-	Y	-
63	Swimlanes	Y	Y	-
64	Tagged Values	Y	Y	-
65	Use Case	Y	Y	-
66	Use Case Diagram	Y	Y	-
67	XMI	Y	Y	-
68	xUML	Y	Y	-
69	Workflow	Y	Y	-
70	Visibility	Y	Y	-

Y-Yes.

TABLE. IV. CRITERIA FOR CONTROLLED AND UNCONTROLLED LANGUAGE IN PROBLEM AS WELL AS SOLUTION DOMAIN

Sr. No.	***Parameter in POS***	***Value per sentence***	***Controlled or Uncontrolled***
1	NN	> 10	Uncontrolled
2	VB	>5	Uncontrolled
3	VBD	>7	Uncontrolled
4	VBG	>5	Uncontrolled
5	VBN	>5	Uncontrolled
6	VBP	>5	Uncontrolled
7	VBZ	>5	Uncontrolled

REFERENCES

[1] Dr. Prasanth Yalla and Nakul Sharma, Combining Natural Language Processing and Software Engineering, In. Proc. International Conference in Recent Trends and Sciences (ICRTES), March 14-15, 2014, Pages 370-373, ISBN: 978-93-5107-223-2. (Conference Proceedings)

[2] Dr. Prasanth Yalla and Nakul Sharma, Parsing Natural Language Text for Use Case description, In. Proc. International Conference of Computer and Big Data, (CSIBIG 2014), March 7-8, 2014. Page: 210-212, ISBN-978-1-4799-3063-0 (Conference Proceedings).

[3] Nakul Sharma, Dr. Prasanth Yalla, "Issues in Developing UML Diagrams from Natural Language Text", In. Proc. Recent Advances In Telecommunications, Informatics And Educational Technologies, Pages 139-145, ISBN: 978-1-61804-262-0.

[4] Alister Cockburn, 2001, Use Case Template, Addison-Wesley, (Book).

[5] Walter F. Tichy, Mathias Landhabuer and 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. (Conference Proceedings).

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

[7] Bures T., Hnetynka P., Kroha P., Simko. V., "Requirement Specifications Using Natural Languages", Charles University, Faculty of Mathematics and Physics, Dept. of Distributed and Dependable Systems, Technical Report No-D3S-TR-2012-05, December 2012.

[8] F Meziane, N. Athanasakis, S. Ananiadou, "Generating Natural Lanuage Specifications from UML Class diagrams", Requirement Engineering Journal, 13(1):1-18, Springer-Verlag, London.

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

[10] Priyanka More and 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. (Journal)

[11] Imran Sarwar Bajwa and M. Abbas Choudhary, Natural Language Processing based auto-mated system for UML diagrams generation In. Proc. 18th National Computer Conference (NCC-2006), Page No-1-6. (Conference Proceedings)

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

[13] Vidhu Bhala, "Conceptual Modeling of Natural Languag Functional Requirements, Journal of System Software , Sciencedirect, (2013), http://dx.doi.org/10.1016/j.jss.2013.08.036.

[14] Mosa Elbendak, Paul Vickers, Nick Rossiter, "Parsed use case descriptions as a basis for object-oriented class model generation", Journal of System and Software, 1209-1223.

[15] Grady Booch, James Rambaugh and Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley Longman, Inc., Second Edition, ISBN 0-201-57168-4 (Book).

[16] Helmut Vieritz, Daniel Schilberg, Sabina Jeschke, "Access to UML Diagrams with the HUTN", In. Proc. ASSETS'12, ACM, October 22–24, 2012, Boulder, Colorado, USA,978-1-4503-1321-6/12/10.