

GENERATION OF USE CASE DIAGRAMS USING NATURAL LANGUAGE PROCESSING

LIBRARY
UNIVERSITY OF MORATUWA, SRI LANKA
MORATUWA

Janani Tharmaseelan

158253L

Thesis submitted in partial fulfillment of the requirements for the degree Master of
Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

February 2019

University of Moratuwa



TH4003

TH4003

004"19"

004043)

TH4003

+

CD-ROM

DECLARATION OF THE CANDIDATE AND SUPERVISOR

“I declare that this is my own work and this thesis/dissertation² does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:

The supervisor/s should certify the thesis/dissertation with the following declaration.

The above candidate has carried out research for the Masters/MPhil/PhD thesis/dissertation under my supervision.

Signature of the supervisor:

Date

ABSTRACT

“*DesignPlus*” is an automated Software and Test Design tool which is capable of generate UML diagrams, User scenarios and System level Test Cases for the given software requirement specifications which is written in natural language. When it comes to software development life cycle Software Design and Testing phases are required considerable high amount of time and human effort which means it is cost to the Software project. In System designing there are some diagrams needs to be drawn and user scenarios need to be written in order to improve the clarity of requirement specification.

DesignPlus is capable of generating UML diagrams (Usecase Diagram) and user scenarios for the given requirement specification in a way accelerate the Software Design phase. And also designing Test cases also need be done in the Test phase of the software life cycle in order to do proper testing on different levels. The developed system is capable of generating system level test cases for the entered functional and non-functional requirement specifications. Technology has used for the Process requirement specification is Natural Language Processing (OpenNLP). Taking altogether I call this is an Integrated Software Design Accelerator since this maintains a link between Software design and Test Design and also this accelerate the process of Software development in a way to save time and reduce cost.

ACKNOWLEDGEMENT

We have taken many efforts in this research. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. First, we are grateful to our project supervisor DR. Charith Chitraranjan for his kind guidance, inspiration and constructive suggestions that were helpful for me in the preparation of this project and as well as this document.

Finally, my thanks goes to my friends and parents who helped me in many ways, and others who are not mentioned here, for the support and assistance provided in all the ways to make this a successful project.

Thank you.

Table of contents

DECLARATION OF THE CANDIDATE AND SUPERVISOR	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT	iv
List of figures.....	vi
List of Tables	vi
List of Abbreviations	vii
1. INTRODUCTION.....	8
1.1 Background Context (Literature Survey).....	8
1.2 Research Gap	9
1.3 Research Problem	12
1.4 Research Objective	14
1.4.1 Implement an algorithm to match a set of certain grammar	14
1.4.2 Develop a tool which Provide Assistance to software test designers	15
1.4.3 Designing a tool to assist software testers	16
2 METHODOLOGY	17
2.1 Methodology.....	17
2.1.1 Addressing literature.....	23
2.1.2 Project development process	25
2.1.3 Feasibility study.....	25
2.2 Testing And Implementation	26
2.2.1 Software implementation	26
2.3 Research Findings	37
3 CONCLUSIONS AND RECOMMENDATIONS.....	38
4 REFERENCES	39

List of figures

Figure 2-1 System Diagram	19
Figure 2-2 NLP Core Functionality.....	27
Figure 2-3 Grammar matching algorithm	28
Figure 2-4 getActor() method	29
Figure 2-5 getAction() method.....	29
Figure 2-6 rules	30
Figure 2-7 Interface Home	31
Figure 2-8 : Interface for Browse	31
Figure 2-9 Content Loaded Interface.....	32
Figure 2-10 Clear all Prompt	32
Figure 2-11 : Results in the Right Corner	33
Figure 2-12 XML file	33
Figure 2-13 Actor Identification	35
Figure 2-14 Actors and function drawing algorithm.....	36

List of Tables

Table 1 :NLP functions	20
------------------------------	----

List of Abbreviations

Abbreviation	Description
Xml	Extensible Markup Language
NLP	Natural Language Processing
NL	Natural Language
SDLC	Software Development Life Cycle
IT	Information Technology
UML	Unified Modeling Language
CDAP	Comprehensive Design & Analysis Project
SAX	Simple API for XML
DOM	Document Object Model
ID	Identification