

**AN ENHANCED TEST CASE GENERATION TECHNIQUE USING ACTIVITY
DIAGRAM FOR SYSTEM TESTING**

NANSUKUSA YUDAYA

A dissertation submitted in partial fulfillment of the
requirements for the award of the degree of
Master of Science (Computer Science)

Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia

JANUARY 2013

I declare that this dissertation entitled “*An Enhanced Test case Generation Technique Using Activity Diagram for System Testing*” is the result of my own research except as cited in the references. The dissertation has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name : NANSUKUSA YUDAYA

Date : 1^{sr} JANUARY, 2013

I affably dedicate this thesis to the biggest treasures of my life, my family. Not forgetting my best friend Farooq who has always believed in me.

ACKNOWLEDGEMENT

First of all, I would like to express my utmost gratitude to Allah S.W.T for His endless blessings and guidance throughout my entire research process and stay in Malaysia, Alhamdillillah for everything.

Then, sincere appreciation goes to my supervisor **Associate Professor Dayang Norhayati Abang Jawawi** for her continued support, guidance, and patience throughout my research. Despite her tight schedule, she always tried to make herself available. I've never seen anyone as committed in nurturing their students like she is. I will always look up to her as my academic role model.

I would also like to express my gratitude to my dear sponsors, Islamic Development Bank, for always providing me with sufficient financial support, may Allah reward you abundantly.

I will forever be grateful to my family, for their undulating support, encouragement and prayers, not forgetting my best friend Farooq. To them I am truly indebted and words alone cannot describe my earnest gratitude.

Special thanks go to all my friends who have always provided aid at various occasions through their views and tips that were undeniably constructive throughout my research and stay in Malaysia. You will all forever remain at heart.

ABSTRACT

Software Testing, a process comprised of test case generation, execution and evaluation is one of the imperative phases of the development life cycle, with its cost approximated to about 50% of the overall development cost. Researchers have automated it using models with utmost focus put on Unified Modeling Language (UML) as the up to date de facto standard utilized in software modeling. Its diagrams include both behavioral and structural. This work has generated system tests (black box) early in the development lifecycle hence the use of behavioral models, activity diagrams in particular as they are one of the earliest and simplest analysis models to be created with sufficient testing information. Also, as a way of reducing test case generation effort and time, an existing technique that supposedly involved more effort and time has been focused on in this work. It has been enhanced by reducing the key steps involved through eliminating intermediate models as a way of reducing effort and time involved in the test case formation process. The enhanced technique has been applied on the same case study as in the original technique, producing four test cases in 115 milliseconds with more ease compared to the original technique that produced five test cases in 160 milliseconds with relatively more effort. It has further been compared against another already existing model based technique (based on sequence diagrams) and also one integrated with a model based tool using both general criteria and those specific to the research problem (TCG effort and time), as a way of further confirming its applicability.

ABSTRAK

Pengujian Perisian, satu proses yang terdiri daripada penjanaan kes ujian, pelaksanaan, dan penilaian adalah salah satu fasa penting untuk kitaran hayat pembangunan, dengan kos yang dianggarkan kira-kira 50% daripada keseluruhan kos pembangunan. Penyelidik telah mengautomasikan ia menggunakan model dengan tumpuan penuh diletakkan ke atas Bahasa Permodelan Bersepadu (UML) sebagai standard de facto terkini yang digunakan dalam pemodelan perisian. Rajahnya termasuk kedua-dua tingkah laku dan struktur. Kerja ini telah menjana ujian sistem (kotak hitam) di awal kitaran hayat pembangunan, maka penggunaan model tingkah laku, rajah aktiviti khususnya kerana mereka adalah salah satu model analisis yang terawal dan paling mudah untuk diwujudkan dengan maklumat ujian yang mencukupi. Sebagai satu cara untuk mengurangkan usaha dan masa untuk menjana kes ujian, satu teknik yang sedia ada yang sepatutnya melibatkan usaha dan masa yang lebih telah diberikan tumpuan dalam kerja-kerja ini. Ia telah dipertingkatkan dengan mengurangkan langkah-langkah utama yang terlibat melalui penghapusan model perantaraan sebagai satu cara untuk mengurangkan usaha dan masa yang terlibat dalam proses pembentukan ujian kes. Teknik yang dipertingkatkan ini telah digunakan pada kajian kes yang sama seperti dalam teknik asal, menghasilkan empat kes ujian dalam 115 milisaat dengan lebih mudah berbanding dengan teknik asal yang menghasilkan lima kes ujian dalam 160 milisaat dengan usaha yang agak lebih. Ia selanjutnya telah dibandingkan dengan satu lagi teknik yang sedia ada model berdasarkan (berdasarkan rajah jujukan) dan juga bersepadu dengan alat berdasarkan model menggunakan kedua-dua kriteria umum dan yang khusus kepada masalah penyelidikan (TCG usaha dan masa), sebagai satu cara untuk selanjutnya mengesahkan penggunaannya.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF TABLES	xi
	LIST OF FIGURES	xii
	LIST OF ABBREVIATIONS	xiv
	LIST OF APPENDICES	xv
1	INTRODUCTION	
1.1	Overview	1
1.2	Problem Background	4
1.3	Problem Statement	8
1.4	Research Aim	9
1.5	Research Objectives	10
1.6	Scope of the Study	10
1.7	Significance of Study	11
1.8	Dissertation Organization	12
2	LITERATURE REVIEW	
2.1	Introduction	13
2.2	Overview of Software Testing	13
2.3	Model Based Testing	15

2.3.1	The MBT Process	16
2.4	Software Testing Metrics	17
2.4.1	Software Testing Effort	18
2.4.2	Software Testing Time	21
2.5	Test Case Generation	23
2.5.1	UML Models for Test case Generation Techniques	24
2.5.1.1	Activity Diagrams	24
2.5.1.2	Sequence Diagrams	27
2.6	Model Based Test Case Generators (Tools)	34
2.5.1	Fokus! MBT	37
2.5.2	Hexawise Tool	38
2.6	Summary	40
3	RESEARCH METHODOLOGY	
3.1	Introduction	41
3.2	Research Process Flowchart	41
3.2.1	Research Process	43
3.2.2	Phase One	44
3.2.3	Phase Two	45
3.2.4	Phase Three	46
3.3	Case Study	47
3.3.1	Bank system application case studies	48
3.4	Methodology Framework	49
3.4.1	Bank system application case study	51
3.4.2	UML Behavioral Model	51
3.4.3	Automatic Test Case Generation	53
3.5	Summary	53
4	GENERATION OF TEST CASES BASED ON THE ENHANCED TEST CASE GENERATION TECHNIQUE	
4.1	Introduction	54
4.2	Overview of the Unified Modeling Language (UML) 2.0	55
4.3	Generating Test cases using the Original Technique	56
4.3.1	Module 1: Generation of ADT	57

4.3.2	Module 2: Generation of ADG	59
4.3.3	Module 3: Test Cases Generation	60
4.3.4	Module 4: Validate Generated Test Cases	63
4.3.5	Evaluation of the Original Technique	64
4.4	Overview of the Proposed (Enhanced) Technique	66
4.4.1	Module 1: Test case Generation	67
4.4.2	Module 2: Test Case Validation	73
4.4.3	Application of Proposed (Enhanced) Technique with ATM Withdrawal Activity Diagram case study	73
4.5	Comparing Original with Proposed (Enhanced) Technique	74
4.5.1	TCG Effort comparisons for original and proposed (enhanced) techniques	75
4.5.2	TCG Time comparisons for original and proposed (enhanced) techniques	76
4.6	Discussion and Summary	79

5 COMPARING THE ENHANCED TECHNIQUE AGAINST THE EXISTING (BASED ON SEQUENCE DIAGRAMS) AND MODEL BASED TOOL INTEGRATED TECHNIQUES

5.1	Introduction	80
5.2	Generating test cases using the enhanced technique with a bank system (ATM PIN Validation) case study	81
5.2.1	Deriving Input and Output Information	82
5.2.2	Using Proposed Algorithm with Identified Inputs	83
5.2.3	Validation of Generated test cases	85
5.3	Generating Test cases using UML Sequence diagram With Bank system case study	86
5.3.1	Overview of Test case Generation from a Sequence Diagram	87
5.3.1.1	Evaluation of Generated test cases	91
5.3.2	Comparing Existing with Proposed Technique	92
5.3.2.1	General Criteria	93
5.3.2.2	Criteria Related to Current Research Problem	95
5.4	Hexawise Test Design Tool	97
5.4.1	Test case Generation using Hexawise Tool	97

5.4.1.1	Pairwise (2-Way) Interactions	99
5.4.1.2	Three (3-Way) Interactions	100
5.4.2	Comparing Enhanced Technique with Hexawise Tool Technique	101
5.4.2.1	General Criteria	102
5.4.2.2	Criteria Related to Current Research Problem	103
5.5	Discussion and Summary	104
6 CONCLUSION AND FUTURE WORK		
6.1	Introduction	106
6.2	Research Conclusion	106
6.3	Research Contribution	110
6.4	Future Works	110
REFERENCES		112
APPENDIX		120