

TEST CASE GENERATOR FOR VISUAL PROGRAMMING LANGUAGE

MOHD FARID JAAFAR

**MASTER OF SCIENCE
UNIVERSITI PUTRA MALAYSIA**

2007

TEST CASE GENERATOR FOR VISUAL PROGRAMMING LANGUAGE

By

MOHD FARID JAAFAR

**Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia,
in Fulfilment of the Requirement for the Degree of Master Science**

January 2007

DEDICATION

I want to dedicate this thesis to all.

**Abstract of thesis presented to the Senate of Universiti Putra Malaysia in
fulfilment of the requirement for the degree of Master of Science**

TEST CASE GENERATOR FOR VISUAL PROGRAMMING LANGUAGE

By

MOHD FARID JAAFAR

January 2007

Chairman: Associate Professor Hj Mohd Hasan Selamat

Faculty: Computer Science and Information Technology

Designing test cases is a tedious and meticulous work. A tester needs to have a depth understanding of a program before a good test suite can be designed. Sometimes, even an expert tends to miss some test cases. The same scenario applies in designing test cases for Visual Programming Language (VPL). Most studies in designing test cases for VPL are using requirement specification, design specification, complex technique (i.e. Z) and third-party tools that requiring novice user to supply information that is not familiar to them. It is certain that the information supplied in creating test case was less accurate and thus affected the test suite produced. Therefore, this research explores the creation of automated test case generator for a VPL using Extensible Markup Language (XML) based language as data representation. The generation method is based on path coverage and boundary value analysis testing techniques. Validation and verification of the test cases created by the tools is done by comparing the generated test cases with the manually-designed test cases. Five test program are chosen and tested using the propose method. From

the result and analysis, a conclusion is drawn that the automated test case generator created are able to create test cases for VPL.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Master Sains

PENJANA KES PENGUJIAN UNTUK BAHASA PENGATURCARAAN VISUAL

Oleh

MOHD FARID JAAFAR

Januari 2007

Pengerusi: Profesor Madya Hj Mohd Hasan Selamat

Fakulti: Sains Komputer dan Teknologi Maklumat

Mereka bentuk *test case* adalah satu tugas yang remeh dan leceh. Penguji perlu mempunyai pemahaman yang mendalam mengenai sesuatu atur cara sebelum satu *test suite* yang baik dapat dihasilkan. Kadangkala, pakar juga terlupa untuk menyertakan beberapa *test case*. Senario yang sama juga berlaku di dalam menghasilkan *test case* untuk Pengaturcaraan Berasaskan Visual (PBV). Kebanyakan penyelidikan dalam mereka bentuk *test suite* untuk PBV menggunakan spesifikasi keperluan, spesifikasi reka bentuk, teknik yang kompleks (contoh. Z) dan peralatan pihak ketiga yang memerlukan pengguna amatur memberikan maklumat yang mereka sendiri kurang fahami. Apa yang pasti, maklumat yang diberikan adalah kurang tepat dan memberi impak kepada *test suite* yang dihasilkan. Oleh itu, penyelidikan ini ialah untuk menghasilkan penjana automatik *test case* yang menggunakan bahasa berasaskan Extensible Markup Language (XML) sebagai perwakilan data. Kaedah penjanaan adalah berdasarkan kepada liputan laluan dan teknik pengujian analisa nilai sempadan. Penentusahan dan pengesahan *test case*

dilakukan dengan membandingkan *test case* yang dijana dengan *test case* yang dihasilkan secara manual. Lima atur cara telah dipilih dan diuji dengan kaedah yang dicadangkan. Dari hasil kajian dan analisis, satu kesimpulan yang dapat dibuat ialah pejana automatik *test case* dapat digunakan untuk menghasilkan *test case* kepada atur cara PBV.

I certify that an Examination Committee has met on 16th January 2007 to conduct the final examination of Mohd Farid Jaafar on his Master of Science thesis entitled “Test Case Generator for Visual Programming Language” in accordance with Universiti Pertanian Malaysia (Higher Degree) Act 1980 and Universiti Pertanian Malaysia (Higher Degree) Regulations 1981. The Committee recommends that the candidate be awarded the relevant degree. Members of the Examination Committee are as follows:

Hamidah Ibrahim, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Haji Ali Mamat, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Muhamad Taufik Abdullah, PhD
Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Shamsul Sahibuddin, PhD
Associate Professor
Faculty of Computer Science and Information System
Universiti Teknologi Malaysia
(External Examiner)

HASANAH MOHD. GHAZALI, PhD
Professor/Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia.

Date: 27 APRIL 2007

This thesis submitted to Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Master of Science. The members of the Supervisory Committee are as follows:

**Hj Mohd Hasan Selamat
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)**

**Abdul Azim Abd Ghani, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)**

**AINI IDERIS, PhD
Professor/Dean
School of Graduate Studies
Universiti Putra Malaysia**

Date: 10 MAY 2007

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any degree at UPM or other institutions.

MOHD FARID JAAFAR

**Date: 28 FEBRUARY
2007**

TABLE OF CONTENTS

	Page
DEDICATION	ii
ABSTRACT	iii
ABSTRAK	v
ACKNOWLEDGEMENTS	vii
APPROVAL	viii
DECLARATION	ix
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
LIST OF ABBREVIATIONS	xv
CHAPTER	
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Research Background	1
1.3 Problem Statement	4
1.4 Research Objective	4
1.5 Scope and Limitations	4
1.6 Definition of Terms	5
1.7 Research Methodology	6
1.8 Significant of the Research	6
1.9 Thesis Organization	7
2 LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Visual Programming Language	8
2.2.1 VPL Data Representation	10
2.3 Test Case Design	15
2.3.1 Black Box Testing	16
2.3.2 White Box Testing	20
2.3.3 Gray Box Testing	20
2.4 Graph Technique	23
2.4.1 Test Case Using Graph	24
2.4.2 Test Case Using Other Techniques	25
2.5 XML Representing Graph	27
2.5.1 GraphXML	28
2.5.2 GraphML	29
2.5.3 XGMML	30
2.5.4 Other XML Representation	32
2.6 Chapter Summary	33
3 METHODOLOGY	35
3.1 Introduction	35
3.2 Research Design	35
3.3 Design of New Architecture	36

3.4	Test Case Formulation	38
3.4.1	Test Case Method	38
3.4.2	FCVPL Data Representation	39
3.4.3	XML Data Representation	39
3.5	Research Measurement	41
3.5.1	Validation and Verification Design	41
3.5.2	Test Program Criteria	42
3.5.3	Data Collection	43
3.5.4	Data Analysis	43
3.6	Chapter Summary	44
4	SYSTEM DESIGN	45
4.1	Introduction	45
4.2	Design of TCML	45
4.3	Semantic of TCML	48
4.3.1	TCML Header	48
4.3.2	TCML Graph	51
4.3.3	TCML Node	52
4.3.4	TCML Edge	54
4.4	Syntax of TCML	54
4.4.1	TCML Header	55
4.4.2	TCML Graph	56
4.4.3	TCML Node	57
4.4.4	TCML Edge	58
4.5	Designing Test Case Generator	60
4.5.1	Syntax Builder	60
4.5.2	TCML Converter	61
4.5.3	Syntax and Semantic Analyser	62
4.5.4	Test Case Generator	63
4.5.5	Error Handler	64
4.6	Chapter Summary	64
5	TOOL IMPLEMENTATION	65
5.1	Introduction	65
5.2	Development Method	65
5.3	Implementation of TSG	66
5.3.1	TCML Builder	66
5.3.2	TCML Converter	67
5.3.3	TSG Syntax and Semantic Analyzer	69
5.3.4	Test Case Generator	70
5.3.5	Error Handler	73
5.4	Chapter Summary	74
6	VALIDATION AND VERIFICATION	75
6.1	Introduction	75
6.2	Test Program	75
6.2.1	Test Program 1	75
6.2.2	Test Program 2	77
6.2.3	Test Program 3	79
6.2.4	Test Program 4	81

	6.2.5	Test Program 5	83
	6.3	Result Analysis	85
	6.4	Chapter Summary	88
7		CONCLUSIONS	109
	7.1	Introduction	90
	7.2	Conclusion	90
	7.3	Future Work	91
		REFERENCE	93
		APPENDICES	96
		BIODATA OF THE AUTHOR	151

LIST OF TABLES

Table	Page
2.1 Overview of some of the popular VPL	9
2.2 Sample of test suite	15
2.3 Comparison between testing technique	21
3.1 Common graph notation in XML	40
4.1 Proposed notation for TCML	46
6.1 Total number of path	85
6.2 Total number of path detected	86
6.3 Total number of manual test cases	86
6.4 Total number of test case generated	87
6.5 Test case comparison	87
6.6 Test case percentage	88

LIST OF FIGURES

Figure	Page
1.1 The v model for testing	2
2.1 Example for FCVPL syntax	11
2.2 Graph notation	17
3.1 Research method	36
3.2 Architecture of TSG	37
4.1 Converting FCVPL to TCML	61
4.2 Generating Test Case	63
5.1 TCML Builder	67
5.2 FCVPL file loaded into TSG	68
5.3 TCML File	69
5.4 Test Cases Generated	73
5.5 Generated Error Message	74
6.1 Test Program 1	76
6.2 Test Cases for Test Program 1	77
6.3 Test Program 2	78
6.4 Test Cases for Test Program 2	79
6.5 Test Program 3	80
6.6 Test Cases for Test Program 3	81
6.7 Test Program 4	82
6.8 Test Cases for Test Program 4	83
6.9 Test Program 5	84
6.10 Test Cases for Test Program 5	85

LIST OF ABBREVIATIONS

BNF	Backus Normal Form
BVA	Boundary Value Analysis
EBNF	Extended BNF
EP	Equivalence Partitioning
GraphML	Graph Markup Language
TCML	Test Case Markup Language
TSG	Test Suite Generator
URI	Uniform Resource Identifier
VP	Visual Programming
VPE	Visual Programming Editor
VPL	Visual Programming Language
XML	Extensible Markup Language

