

**AN IMPROVEMENT OF AUTOMATED SOFTWARE FAULT INJECTION
TOOL FOR ROBUSTNESS**

By

**MOHD DAUD BIN ALANG HASSAN
SAIFUL FADZLI BIN SALIAN**

DECEMBER 2008

TABLE OF CONTENTS

DECLARATIONS

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

ii

LIST OF FIGURES

v

LIST OF TABLES

viii

LIST OF ABBREVIATIONS

ix

ABSTRACT

x

CHAPTER 1 INTRODUCTION

1.1	Introduction	1
1.2	Overview	2
1.3	Aim and Objectives of this Project	4
1.4	Project Outline	5

CHAPTER 2 LITERATURE REVIEW

2.1	Introduction	6
2.2	Overview Software Testing	6
2.3	A Survey of Java Based Software Testing Tools	
	2.3.1 JACA	8
	2.3.2 Javassist	9
	2.3.3 JOSIT	9
	2.3.4 JUnit	10
	2.3.5 FIONA	10
	2.3.6 Gretel	11
	2.3.7 SoftTest	11
	2.3.8 SFIT	12
2.4	Analysis of Software Testing Tools	12
2.5	Summary	14

CHAPTER 3 INTRODUCING SFIT

3.1	Introducing SFIT	15
3.2	SFIT Components	15
	3.2.1 Class Inspector	16
	3.2.2 Fault Setting	17
	3.2.3 Loader and Generator	18
	3.2.4 Fault Injector	18
	3.2.5 Log	19
3.3	SFIT Testing Process	19
3.4	Enhancement of SFIT	22
3.5	Summary	23

CHAPTER 4 EXPERIMENTS WITH SFIT

4.1	Introduction	24
4.2	Overview of Linda and Jada	25
4.3	Running Experiments	30
	4.3.1 Experiment 1: public void out (Object item)	32
	4.3.2 Experiment 2: public void out (Object objects[], int n objects)	35
	4.3.3 Experiment 3: public object in (Object match)	38
	4.3.4 Experiment 4: public object in (Object match[], int n_objects)	42
	4.3.5 Experiment 5: public object in (Object match[], long timeout)	45
	4.3.6 Experiment 6: public object in (Object match[], Int n_objects, long timeout)	48
	4.3.7 Experiment 7: public object in_nb (Object match)	51
	4.3.8 Experiment 8: public object in_nb (Object match[], int n_objects)	53
	4.3.9 Experiment 9: public object read (Object match)	57
	4.3.10 Experiment 10: public object read (Object match[], long timeout)	59

4.3.11	Experiment 11: public object read (Object match[], Int n_objects)	62
4.3.12	Experiment 12: public object read (Object match[], Int n_objects, long timeout)	66
4.3.13	Experiment 13: public object read_nb (Object match)	69
4.3.14	Experiment 14: public object read_nb (Object match[], Int n_objects)	72
4.3.15	Experiment 15: public object read (object match) with maximum allowable test cases of 2500	75
4.4	Summary	78

CHAPTER 5 ASSESSMENT AND DISCUSSION

5.1	Lesson Learned from SFIT	79
5.2	Future Work	83
5.3	Concluding Remarks	84

REFERENCES	85
-------------------	-----------

APPENDIX A Object Space Class

ABSTRACT

This project is a continuation from earlier work involving the development of a software fault injection tool, called SFIT. A number of limitation is observed in the earlier version of SIFT in terms of the support for automated testing, the integration with a graphical user interface as well as the capability to keep historical data. A new version of SFIT has been developed as a result.

In order to evaluate its suitability and applicability as a general software testing tool, a number of experiments have been devised utilizing SFIT to perform robustness testing on a public domain commercial-off-the-shelves components for distributed shared memory manipulation library, called Jada. In doing so, a number of observation have also been made in terms of the robustness of Jada (i.e. in terms of whether or not Jada can be used in a highly available and safety critical systems).