

วิธีการสร้างกรณีทดสอบสำหรับการปฏิสัมพันธ์แบบโพลีเมอร์ฟิก
จากแผนภาพซีเคอร์นซ์ของยูเอ็มแอล



นายศิริส สุภาวิตา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

138810801
13820801

AN APPROACH FOR TEST CASE GENERATION
FOR POLYMORPHIC INTERACTIONS FROM UML SEQUENCE DIAGRAMS

Mr. Siros Supavita

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

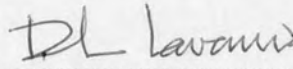
Academic Year 2006

Copyright of Chulalongkorn University

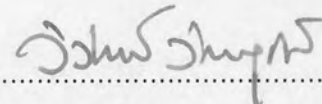
490714

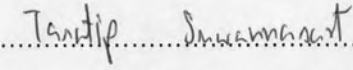
Thesis Title AN APPROACH FOR TEST CASE GENERATION
FOR POLYMORPHIC INTERACTIONS FROM UML
SEQUENCE DIAGRAMS
By Mr. Siros Supavita
Field of study Computer Engineering
Thesis Advisor Assistant Professor Taratip Suwannasart, Ph.D.

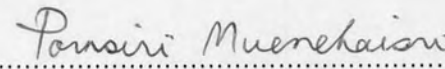
Accepted by the Faculty of Engineering, Chulalongkorn University in
Partial Fulfillment of the Requirements for the Master Degree

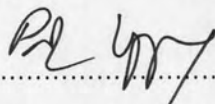

..... Dean of the Faculty of Engineering
(Professor Direk Lavansiri, Ph.D.)

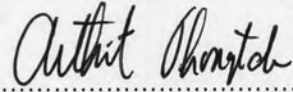
THESIS COMMITTEE


..... Chairman
(Assistant Professor Wiwat Vatanawood, Ph.D.)


..... Thesis Advisor
(Assistant Professor Taratip Suwannasart, Ph.D.)


..... Member
(Associate Professor Pornsiri Muenchaisri, Ph.D.)


..... Member
(Assistant Professor Pattarachai Lalitrojwong, Ph.D.)


..... Member
(Arthit Thongtak, D.Eng.)

ศิริส สุภาวิตา : วิธีการสร้างกรณีทดสอบสำหรับการปฏิสัมพันธ์แบบโพลีมอร์ฟิกจากแผนภาพซีเควNSSของยูเอ็มแอล (AN APPROACH FOR TEST CASE GENERATION FOR POLYMORPHIC INTERACTIONS FROM UML SEQUENCE DIAGRAMS) อ. ที่ปรึกษา: ผศ.ดร. ธราทิพย์ สุวรรณศาสตร์, 157 หน้า.

การทดสอบซอฟต์แวร์เป็นขั้นตอนที่สำคัญขั้นตอนหนึ่งของการพัฒนาซอฟต์แวร์ ได้มีการค้นพบกันมาแล้วว่าซอฟต์แวร์ที่ถูกพัฒนาโดยวิธีการเชิงวัตถุ จำเป็นต้องใช้เทคนิคในการทดสอบที่แตกต่างออกไป เนื่องจากคุณสมบัติเฉพาะของวิธีการเชิงวัตถุ คุณสมบัติโพลีมอร์ฟิกเป็นหนึ่งในคุณสมบัติที่โดดเด่นของวิธีการเชิงวัตถุ งานวิจัยนี้ นำเสนอวิธีการสำหรับทดสอบซอฟต์แวร์ที่พัฒนาโดยวิธีการเชิงวัตถุ โดยมุ่งเน้นที่ข้อผิดพลาดจากโพลีมอร์ฟิก วิธีการนี้ใช้แผนภาพซีเควNSSในยูเอ็มแอลเป็นข้อกำหนดในการทดสอบ และมุ่งเน้นที่การค้นหาข้อผิดพลาดที่เกิดจากความแตกต่างระหว่างแผนภาพซีเควNSSกับซอฟต์แวร์ที่สร้างขึ้น

วิธีการที่นำเสนอนี้ รวมถึงแบบจำลองที่ใช้สำหรับแทนลำดับการส่งเมสเซจระหว่างอ็อบเจกต์ ที่เกิดขึ้นในการปฏิสัมพันธ์หนึ่งๆ นอกจากนั้น วิธีการนี้ยังสนับสนุนการตรวจสอบลำดับการส่งเมสเซจที่เกิดขึ้นจริงในการทดสอบ เทียบกับลำดับการส่งเมสเซจที่ระบุไว้ในแผนภาพซีเควNSS วิธีการยังครอบคลุมถึงการสร้างกรณีทดสอบ และเงื่อนไขในการวัดความครอบคลุมของการทดสอบ โดยมุ่งเน้นไปที่คุณสมบัติโพลีมอร์ฟิกในซอฟต์แวร์ที่ทดสอบ ได้มีการสร้างเครื่องมือสำหรับสร้างกรณีทดสอบโดยอ้างอิงจากวิธีการทดสอบที่นำเสนอในงานวิจัย เพื่อประเมินความสามารถของวิธีการ โดยที่ผลการประเมินสะท้อนให้เห็น ได้เป็นอย่างดี ถึงความสามารถของวิธีการทดสอบ ในการสร้างกรณีทดสอบและการทดสอบในสถานการณ์ต่างๆ

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.... ลายมือชื่อนิสิต.....ศิริส สุภาวิตา
สาขาวิชา.....วิศวกรรมคอมพิวเตอร์... ลายมือชื่ออาจารย์ที่ปรึกษาอ.ดร. ธราทิพย์ สุวรรณศาสตร์
ปีการศึกษา.....2549.....

4670519621 : MAJOR COMPUTER ENGINEERING

KEY WORD: SOFTWARE TESTING / UML / SEQUENCE DIAGRAM / TEST CASE / OBJECT-ORIENTED / POLYMORPHIC INTERACTION

SIROS SUPAVITA : AN APPROACH FOR TEST CASE GENERATION FOR POLYMORPHIC INTERACTIONS FROM UML SEQUENCE DIAGRAMS. THESIS ADVISOR : ASST.PROF. TARATIP SUWANNASART, PH.D., 157 pp.

Software testing is a crucial part in software development. It has been discovered that object-oriented software needs different testing techniques due to its unique features. Polymorphism is one of the prominent features in object-oriented paradigm. This research presents an approach for testing object-oriented software focusing on polymorphism feature. The test approach uses UML sequence diagrams as test specification and aims at finding defects which are the result of unconformity between a sequence diagram and its implementation.

The approach covers the model for message sending sequence model for representing a sequence of message sent between objects in an interaction. Moreover the approach supports verification of the actual message sending sequence occurred in test execution against the expected message sending sequence from a UML sequence diagram. The approach also includes test case generation and test adequacy criteria, which focus on polymorphism. A tool is implemented based on the approach for evaluation purpose. The evaluation of the tool effectively reflects the capability of the test approach in various situations.

DepartmentComputer Engineering... Student's Signature *Siros Supavita*
 Field of study ..Computer Engineering.... Advisor's Signature *Taratip Suwannasart*
 Academic year2006.....

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my thesis advisor, Assistant Professor Dr. Taratip Suwannasart, who has been guiding me through the research. She has given me a lot of opportunities to grow as a researcher and has always encouraged me to push myself beyond my limit. The accomplishment of the research owes a great deal to her invaluable advice and support.

Also I am very appreciative of all advice, criticism, and comments by the thesis committee, Assistant Professor Dr. Wiwat Vatanawood, Associate Professor Dr. Pornsiri Muenchaisri, Dr. Arthit Thongtak, and Assistant Professor Dr. Pattarachai Lalitrojwong.

I would like to thank the faculty of Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University for their suggestion and support throughout my years at the department. My appreciation is also extended to the staff at the department for their supportive cooperation.

I also appreciate the warm and kind support from the colleagues at Software Engineering Laboratory. Without their support, my research life would have been rough and monotonous.

Finally I would like to thank my family, who has always been encouraging and supportive all through my life. My gratefulness especially goes to my parents, who are my role models in various aspects of life.

TABLE OF CONTENTS

	Page
ABSTRACT (THAI).....	iv
ABSTRACT (ENG).....	v
ACKNOWLEDGEMENTS.....	vi
TABLE OF CONTENTS.....	vii
LIST OF TABLES.....	xii
LIST OF FIGURES.....	xiii
Chapter	
I INTRODUCTION.....	1
1.1 Problem Statement.....	1
1.2 Objective.....	2
1.3 Scope.....	2
1.4 Research Methodology.....	3
1.5 Contribution.....	6
1.6 Publications.....	6
II LITERATURE REVIEWS.....	8
2.1 Concepts and Theories.....	8
2.1.1 Object-Oriented Software.....	8
2.1.2 Object-Oriented Software Testing.....	12
2.1.3 Unified Modeling Language.....	15
2.1.4 Fault Model & Verification.....	17
2.1.5 Polymorphism & Adequacy Criteria.....	17
2.2 Related Works.....	18
2.2.1 SeDiTeC.....	18

Chapter	Page
2.2.2 TACCLE and Sequence Diagram Based Testing	19
2.2.3 Using UML Collaboration Diagrams for Static Checking and Test Generation	20
2.2.4 Criteria for Testing Polymorphic Relationships	20
2.2.5 Test Adequacy Criteria for UML Design Models	20
2.2.6 Other UML Based Testing	21
III TEST APPROACH.....	22
3.1 Overview	22
3.2 Adequacy Criteria.....	23
3.3 Polymorphic Interaction Pattern	24
3.4 Test Result Verification	25
3.5 Test Case Generation.....	26
IV ADEQUACY CRITERIA.....	29
4.1 Overview	29
4.2 The Criteria	30
4.2.1 Base Class Coverage	31
4.2.2 Overriding Method Coverage.....	32
4.2.3 Inheritance Coverage	33
4.2.4 Strong Overriding Method Coverage	34
4.2.5 Strong Inheritance Coverage	35
4.3 Discussion	36
V POLYMORPHIC INTERACTION PATTERNS.....	40
5.1 Polymorphic Interaction and Testing.....	40
5.1.1 Polymorphic Interaction	40
5.1.2 Non-Polymorphic Interaction	40

Chapter	Page
5.2 The Patterns.....	42
5.2.1 Simple Polymorphic Assignment.....	42
5.2.2 Parameter-Influenced Polymorphic Assignment.....	43
5.2.3 Configuration-Influenced Polymorphic Assignment.....	45
5.3 Discussion	46
5.3.1 Variations	46
5.3.2 Combination of Patterns	47
5.4 Examples	47
VI MESSAGE SENDING SEQUENCE & VERIFICATION.....	53
6.1 Overview	53
6.2 Concept of Message Sending Sequence Model.....	54
6.2.1 Instrumentation Model Requirements	54
6.2.2 Design Model Requirements	55
6.2.3 Discussion.....	55
6.3 Message Sending Sequence Model.....	56
6.3.1 ExecutionContext.....	57
6.3.2 Instance	58
6.3.3 CallMessage	58
6.3.4 ReturnMessage.....	59
6.3.5 ExecutionSequence.....	59
6.3.6 Classifier and Operation.....	60
6.3.7 Class, Interface, and DataType.....	60
6.4 Verification Issues.....	60
6.4.1 Consideration of Defects	61

Chapter	Page
6.4.2 Polymorphism	62
6.4.3 Refinement	63
6.5 Verification Procedure	65
6.5.1 ExecutionSequence Verification	66
6.5.2 ExecutionContext Verification	66
6.5.3 CallMessage Verification	69
6.6 Example	70
VII UML SEQUENCE DIAGRAM & TEST CASE GENERATION	77
7.1 Overview	77
7.2 Extension to UML Sequence Diagram	77
7.2.1 Simple Polymorphic Assignment	82
7.2.2 Parameter-Influenced Polymorphic Assignment	83
7.2.3 Configuration-Influenced Polymorphic Assignment	86
7.3 Test Case Generation	88
7.3.1 Simple Polymorphic Assignment	89
7.3.2 Parameter-Influenced Polymorphic Assignment	91
7.3.3 Configuration-Influenced Polymorphic Assignment	93
7.4 Discussion	95
7.4.1 Mixed Pattern	95
7.4.2 Interaction with more than one polymorphic assignment	96
VIII TOOL IMPLEMENTATION & EVALUATION	97
8.1 Tool Implementation	97
8.1.1 Test Case Generator	97
8.1.2 Instrumentation Tool	102

Chapter	Page
8.1.3 Message Sending Sequence Verification Tool	103
8.2 Evaluation	104
8.2.1 Store Example.....	104
8.2.2 File Management Example	111
8.2.3 Evaluation Summary	122
IX CONCLUSION	124
9.1 Research Result.....	124
9.2 Limitations.....	126
9.3 Future Works.....	127
REFERENCES	129
Appendices	134
Appendix A.....	135
Appendix B.....	142
Appendix C.....	150
BIOGRAPHY	157

LIST OF TABLES

Table	Page
4.1 Test scenario for Base class coverage from the example	31
4.2 Test scenario for Overriding method coverage from the example.....	33
4.3 Test scenario for Inheritance coverage from the example	34
4.4 Test scenario for Strong overriding method coverage from the example.....	35
4.5 Test scenario for Strong inheritance coverage from the example	36
4.6 Minimum number of test cases for the adequacy criteria	39
5.1 Test cases for the interaction in figure 5.8.....	49
5.2 Test cases for the interaction in figure 5.9.....	51
5.3 Test cases for the interaction in figure 5.10.....	52
7.1 Tagged values proposed in the research.....	79
7.2 Tagged values used for each polymorphic assignment pattern.....	88
7.3 Test cases for the interaction in figure 7.8.....	90
7.4 Test cases for the interaction in figure 7.9.....	93
7.5 Test cases for the interaction in figure 7.11.....	95

LIST OF FIGURES

Figure	Page
1.1 Overview of the approach	4
1.2 Activities in this research.....	6
2.1 Example of inheritance.....	9
2.2 Example of overriding.....	10
2.3 Example of polymorphism in UML sequence diagram.....	11
2.4 Example of polymorphism in Java code.....	11
2.5 Adequacy criteria in test process	13
2.6 Example of class diagram.....	16
2.7 Example of sequence diagram	16
2.8 Example of sequence diagram for SeDiTeC	19
3.1 Overview of the approach.....	22
3.2 Steps in test case generation.....	27
4.1 Subsumption hierarchy of the criteria	29
4.2 Example of a UML sequence diagram	30
4.3 Example of a UML class diagram	31
5.1 Example of a UML class diagram	41
5.2 Example of an interaction without a polymorphic assignment	42
5.3 Example of Simple polymorphic assignment.....	43
5.4 Example of Parameter-influenced polymorphic assignment.....	44
5.5 Example of Configuration-influenced polymorphic assignment.....	45
5.6 Example of a variation of polymorphic assignment patterns.....	46
5.7 UML class diagram for store example	48
5.8 UML sequence diagram for purchasing product	49

Figure	Page
5.9 UML sequence diagram for checking product availability.....	50
5.10 UML sequence diagram for checking availability of the top selling product.....	51
6.1 Overview of Message Sending Sequence Verification.....	53
6.2 Class diagram of message sending sequence model.....	57
6.3 Example source code of polymorphic interaction	63
6.4 Overview of verification procedure	66
6.5 ExecutionContext verification.....	68
6.6 CallMessage verification	70
6.7 Example of a UML sequence diagram	71
6.8 UML class diagram showing the inheritance hierarchy from the example	71
6.9 Expected message sending sequence from the example.....	72
6.10 Actual message sending sequence from the example	73
6.11 Example of invalid actual message sending sequence.....	75
7.1 Example of a UML class diagram	82
7.2 Usage of extension for Simple polymorphic assignment	83
7.3 Format of condition.....	84
7.4 Usage of extension for Parameter-influenced polymorphic assignment	85
7.5 Format of environment variable declaration	86
7.6 Usage of the extension for Configuration-influenced polymorphic assignment	87
7.7 UML class diagram for store example	89
7.8 UML sequence diagram for purchasing product.....	90
7.9 UML sequence diagram for checking product availability.....	91
7.10 Factory method condition for the UML sequence diagram in figure 7.9.....	92
7.11 UML sequence diagram for checking availability of the top selling product.....	94

Figure	Page
7.12 Factory method condition for the UML sequence diagram in figure 7.11.....	94
7.13 Environment variable declaration for the sequence diagram in figure 7.11	94
8.1 Test case generation process used in the tool.....	98
8.2 UML component diagram of the test case generator.....	101
8.3 UI of the test case generator.....	101
8.4 Usage of the test harness	103
8.5 UML component diagram of the instrumentation tool and test harness framework	103
8.6 UI of the message sending sequence evaluator.....	104
8.7 Product class diagram	105
8.8 UML sequence diagram for purchasing product	105
8.9 Tagged value in ArgoUML	106
8.10 List of files generated by the test case generator	107
8.11 Test case summary for purchasing product.....	107
8.12 A test case for purchasing product	108
8.13 UML sequence diagram for checking product availability.....	109
8.14 A test case for checking product availability.....	109
8.15 UML sequence diagram for checking availability of the top selling product.....	110
8.16 A test case for checking availability of the top selling product.....	111
8.17 Class diagram for interface Command and its subtypes	112
8.18 Class diagram of interface File and its subtypes.....	113
8.19 UML sequence diagram for print file info.....	114
8.20 Test case summary for print file info using overriding method coverage	115
8.21 Test case summary for print file info using inheritance coverage.....	115
8.22 UML sequence diagram for print command usage.....	116

Figure	Page
8.23 Test case summary for print command usage using Overriding method coverage	117
8.24 Test case summary for print command usage using Inheritance coverage.....	117
8.25 UML sequence diagram for execute command	118
8.26 Test case summary for execute command using Overriding method coverage...	119
8.27 Test case summary for execute command using Inheritance coverage	119
8.28 Test case summary for execute command using Strong overriding method coverage	120
8.29 Test case summary for execute command using Strong inheritance coverage....	121
8.30 Message sending sequence evaluator – matched.....	122
8.31 Message sending sequence evaluator – mismatched.....	122