การสร้างกรณีทดสอบจียูไอจากยูเอ็มแอล

นางสาวเพชรน้ำค้าง วงษ์ศุทธิภากร

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2548

ISBN 974-53-2818-9

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

GUI TEST CASE GENERATION FROM UML

Miss Petnamkang Wongsuttipakorn

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2005

ISBN 974-53-2818-9

หัวข้อวิทยานิพนธ์	การสร้างกรณีทดสอบจียูไอจากยูเอ็มแอล
โดย	นางสาวเพชรน้ำค้าง วงษ์ศุทธิภากร
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(รองศาสตราจารย์ ดร.สาธิต วงศ์ประทีป)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์)

..... กรรมการ

(รองศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... กรรมการ

(อาจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นางสาวเพชรน้ำค้าง วงษ์ศุทธิภากร : การสร้างกรณีทดสอบจียูไอจากยูเอ็มแอล. (GUI TEST CASE GENERATION FROM UML) อ. ที่ปรึกษา : ผศ.ดร.ญาใจ ลิ่มปิยะกรณ์, 87 หน้า. ISBN 947-53-2818-9.

ปัจจุบัน ซอฟต์แวร์ทั่วไปมีการใช้ส่วนต่อประสานกราฟิกกับผู้ใช้หรือจียูไอกันมาก เนื่องจากสะดวกและใช้งานง่ายกว่าการใช้งานแบบบรรทัดคำสั่ง การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้เป็นสิ่งสำคัญ เนื่องจากหากส่วนต่อประสานกราฟิกกับผู้ใช้มีข้อผิดพลาดก็เสมือนระบบทำงานผิดพลาด แม้ว่าแอพพลิเคชันซอฟต์แวร์จะทำงานได้ถูกต้อง อย่างไรก็ตามการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ยังคงเป็นงานหัวข้อวิจัย เนื่องจากถ้าทำการทดสอบจากโครงสร้างของส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งปกติสามารถแทนได้ด้วยเครื่องสถานะจำกัด จะได้กรณีทดสอบจำนวนมหาศาล ซึ่งไม่เหมาะสมในทางปฏิบัติหรือใช้ทดสอบจริง ดังนั้นวิทยานิพนธ์นี้จึงได้นำเสนอแนวทางการสร้างกรณีทดสอบจากแผนภาพยูเอ็มแอล โดยใช้วิธีการสร้างกรณีทดสอบในระดับของงานซึ่งมีแนวคิดคล้ายคลึงกับแผนงานปัญญาประดิษฐ์ ชุดกรณีทดสอบจะครอบคลุมทุกหน้าที่การทำงานที่มีปฏิสัมพันธ์กับผู้ใช้ดังปรากฏในเอกสารการออกแบบระบบ ผลที่ได้คือจำนวนกรณีทดสอบที่เป็นไปได้ในทางปฏิบัติของการทดสอบซอฟต์แวร์จียูไอ นอกจากนี้ผู้วิจัยยังได้พัฒนาเครื่องมือกึ่งอัตโนมัติสำหรับสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากเอกสารการออกแบบยูเอ็มแอล เพื่อให้สามารถสร้างกรณีทดสอบได้ทันทีหลังจากทราบข้อกำหนดซอฟต์แวร์ วิธีการและเครื่องมือดังกล่าวจะช่วยประหยัดทรัพยากรที่ใช้ในขั้นตอนการทดสอบ รวมทั้งสนับสนุนการนำกรณีทดสอบกลับมาใช้งานได้ใหม่

ภาควิชา..... วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....เพชรน้ำค้าง วงษ์ศุทธิภากร.....
สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์.... ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา....2548....

46706530 : MAJOR SOFTWARE ENGINEERING

KEY WORD: GUI TESTING / UML / TEST CASE GENERATION

PETNAMKANG WONGSUTTIPAKORN : GUI TEST CASE GENERATION FROM UML. THESIS ADVISOR : ASST. PROF. YACHAI LIMPIYAKORN, Ph.D., 87 pp. ISBN 974-53-2818-9.

Graphical User Interfaces (GUIs) are ubiquitous as the user interfaces of most of software today. For users, GUIs provide easier access to the systems compared to using command lines. GUI testing is essential to ensure that the application software correctly functions. However, GUI testing has been an ongoing research topic. According to the GUI structure, which can be represented by Finite State Machine, the number of test cases generated is enormous and it is not practical for real-world testing. This thesis, therefore, proposes an approach to generating GUI test cases from UML diagrams. The method generates test cases at the level of task that is similar to the concept of AI planning. The test suite will cover every function that has interactions with a user as described in the system design document. As a result, the number of test cases generated from our approach is practical to serve for testing purposes. We also developed a tool to semi-automate GUI test case generation from UML design documents. This would enable the generation of test cases once the software specification had been obtained. The proposed method and the implementation of the tool in this research would save the resources during the testing phase as well as enable the reusability of GUI test cases.

Department..... Computer Engineering.....Student's signature.....

Field of study.....Computer Science.....Advisor's signature.....

Academic year....2005.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์อย่างยิ่งของผู้ช่วยศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์ อาจารย์ที่ปรึกษา ซึ่งท่านได้ให้ความรู้ แนวทางการวิจัย ตรวจสอบให้คำแนะนำ และสนับสนุนเป็นอย่างดี จนทำให้การวิจัยในครั้งนี้สำเร็จออกมาด้วยดี

ขอขอบพระคุณ รองศาสตราจารย์ ดร.สาธิต วงศ์ประทีป รองศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล และอาจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์ ซึ่งเป็นประธานและกรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอบคุณเพื่อนๆ และพี่ๆ ที่คอยช่วยเหลือ ให้คำแนะนำ และให้กำลังใจด้วยดีเสมอมา

ท้ายที่สุด ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดา มารดา และทุกคนในครอบครัว ที่ให้กำลังใจ และให้การสนับสนุนเรื่อยมาจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญภาพ ..... ฌ
สารบัญตาราง ..... ฎ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของการวิจัย 2
1.3 ขอบเขตของการวิจัย 2
1.4 ขั้นตอนและวิธีดำเนินการวิจัย 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์ 3
1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5
2.1 ทฤษฎีที่เกี่ยวข้อง 5
2.2 งานวิจัยที่เกี่ยวข้อง 15
บทที่ 3 การสร้างกรณีทดสอบจียูไอจากยูเอ็มแอล 19
3.1 ภาพรวมของวิธีการที่นำเสนอ 19
3.2 รูปแบบและวิธีการเขียนแผนภาพยูเอ็มแอลเพื่อใช้ในการสร้างกรณีทดสอบ 20
3.3 วิธีการสร้างกรณีทดสอบจากยูเอ็มแอล 28
3.4 รายละเอียดของเครื่องมือสร้างและทดสอบกรณีทดสอบ 36
บทที่ 4 การพัฒนาเครื่องมือสร้างกรณีทดสอบจียูไอจากยูเอ็มแอล 47
4.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา 47
4.2 ฐานข้อมูลของเครื่องมือ 47
บทที่ 5 การทดสอบเครื่องมือ 54
5.1 สภาพแวดล้อมที่ใช้ในการทดสอบเครื่องมือที่สร้างขึ้น 54
5.2 ขั้นตอนการทดสอบเครื่องมือ 54

หน้า

5.3 ระบบตัวอย่างที่ใช้ในการทดสอบ 55
5.4 ผลการทดสอบ 58
5.5 สรุปผลการทดสอบ 64
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ 65
6.1 สรุปผลการวิจัย 65
6.2 ข้อจำกัดและแนวทางการวิจัยต่อ 66
รายการอ้างอิง 67
ภาคผนวก 69
ภาคผนวก ก การวาดและแปลงแผนภาพยูเอ็มแอลเป็นแฟ้มเอกซ์เอ็มแอล โดยเครื่องมือเรซันเนล โรส 70
ภาคผนวก ข รายละเอียดยูสเคสและแผนภาพกิจกรรมของระบบที่ใช้ทดสอบ 77
ภาคผนวก ค คู่มือการติดตั้งและใช้งานเครื่องมือสร้างและทดสอบกรณีทดสอบจียูไอจากยูเอ็มแอล 82
ประวัติผู้เขียนวิทยานิพนธ์ 87

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 1 ผู้ใช้งานระบบ 7
รูปที่ 2 ยูสเคส 7
รูปที่ 3 ความสัมพันธ์ของยูสเคสแบบต่างๆ 8
รูปที่ 4 ตัวอย่างยูสเคสซีนาริโอของระบบ 9
รูปที่ 5 ตัวอย่างแผนภาพกิจกรรมของหน้าที่ตรวจรหัส 13
รูปที่ 6 การออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ จากความต้องการเหมือนกัน 14
รูปที่ 7 ตัวอย่างแผนภาพดับเบิลยูเอ็นดี 15
รูปที่ 8 กระบวนการทำงานหลักของระบบ 17
รูปที่ 9 ภาพรวมการทำงานของการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จาก
ยูเอ็มแอล 21
รูปที่ 10 ตัวอย่างแผนภาพยูสเคสในเครื่องมือเรชันเนล โรส 22
รูปที่ 11 ลักษณะต่างๆ ของการทำงานของยูสเคส 24
รูปที่ 12 แผนภาพดับเบิลยูเอ็นดี 26
รูปที่ 13 ตัวอย่างลักษณะการเขียนแผนภาพดับเบิลยูเอ็นดี 26
รูปที่ 14 ตัวอย่างการเขียนแผนภาพกิจกรรมในเครื่องมือเรชันเนล โรส 27
รูปที่ 15 ตัวอย่างสคริปต์ทดสอบ 33
รูปที่ 16 ตัวอย่างแผนภาพยูเอ็มแอลเมื่อแปลงเป็นแฟ้มเอกซ์เอ็มแอล 37
รูปที่ 17 แผนภาพยูสเคสของเครื่องมือสร้างและทดสอบกรณีทดสอบ 38
รูปที่ 18 แผนภาพคลาสของเครื่องมือสร้างและทดสอบกรณีทดสอบ 41
รูปที่ 19 แผนภาพลำดับของการสร้างกรณีทดสอบ 44
รูปที่ 20 แผนภาพลำดับของการทดสอบ 45
รูปที่ 21 แผนภาพอีอาร์ของฐานข้อมูล 48
รูปที่ 22 แผนภาพยูสเคสของระบบบริการในห้องสมุด 56
รูปที่ 23 แผนภาพดับเบิลยูเอ็นดีของระบบบริการในห้องสมุด 57
รูปที่ 24 แฟ้มเอกซ์เอ็มแอลของระบบบริการห้องสมุด 58
รูปที่ 25 แผนการทดสอบของภารกิจ Borrow Book 59
รูปที่ 26 ตัวดำเนินการ Borrow Book 60
รูปที่ 27 กรณีทดสอบในรูปแบบแฟ้มเอกซ์เอ็มแอล 61

หน้า

รูปที่ 28 กรณีทดสอบในรูปของสคริปต์ทดสอบ 61
รูปที่ 29 การนำสคริปต์ทดสอบมาดำเนินงานกับโปรแกรม 63
รูปที่ 30 รายงานผลการทดสอบของแผน Borrow Book 64
รูปที่ 31 หน้าจอเลือกรูปแบบการวาดแผนภาพยูเอ็มแอล 70
รูปที่ 32 หน้าจอเลือกแฟ้มคุณสมบัติ 71
รูปที่ 33 หน้าต่างของเครื่องมือเรชันเนล โรส 72
รูปที่ 34 การสร้างแผนภาพยูสเคสใหม่ 73
รูปที่ 35 การวาดเส้นความสัมพันธ์แบบต่างๆ 74
รูปที่ 36 การเปิดหน้าจอส่วนการเขียนคำอธิบายยูสเคส 74
รูปที่ 37 การสร้างแผนภาพดับเบิลยูเอ็นดี 75
รูปที่ 38 การสร้างแผนภาพกิจกรรม 76
รูปที่ 39 แผนภาพกิจกรรมของยูสเคส Update Book 77
รูปที่ 40 แผนภาพกิจกรรมของยูสเคส Add New Book 78
รูปที่ 41 แผนภาพกิจกรรมของยูสเคส Add member 79
รูปที่ 42 แผนภาพกิจกรรมของยูสเคส Borrow book 80
รูปที่ 43 แผนภาพกิจกรรมของยูสเคส Update Book Status 81
รูปที่ 44 การกำหนดพาทของแฟ้มที่ต้องการให้ระบบปฏิบัติการสามารถเรียกใช้ได้ 83
รูปที่ 45 ส่วนต่อประสานกราฟิกกับผู้ใช้ (ส่วนสร้างกรณีทดสอบ) 84
รูปที่ 46 ส่วนต่อประสานกราฟิกกับผู้ใช้ (ส่วนแก้ไขและดำเนินการทดสอบ) 84

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 1 คำอธิบายยูสเคส 10
ตารางที่ 2 ส่วนประกอบของแผนภาพกิจกรรม 11
ตารางที่ 3 ส่วนประกอบของแผนภาพดับเบิลยูเอ็นดี 14
ตารางที่ 4 คำอธิบายยูสเคสที่ใช้ในงานวิจัยนี้ 23
ตารางที่ 5 คำสงวนที่ใช้ในการเขียนคำอธิบายยูสเคส 25
ตารางที่ 6 ค่าของแอตทริบิวท์ id ในอีลีเมนต์ <seq/> 30
ตารางที่ 7 การเขียนสคริปต์ของคอมโพเนนท์แต่ละประเภท 34
ตารางที่ 8 คำอธิบายยูสเคสสร้างกรณีทดสอบ 38
ตารางที่ 9 คำอธิบายยูสเคสสร้างตัวดำเนินการและแผนการทดสอบ 38
ตารางที่ 10 คำอธิบายยูสเคสแก้ไขกรณีทดสอบและแทรกจุดตรวจสอบ 39
ตารางที่ 11 คำอธิบายยูสเคสจัดการแฟ้มเอกซ์เอ็มแอล 39
ตารางที่ 12 จัดการแฟ้ม 40
ตารางที่ 13 ทดสอบโปรแกรม 40
ตารางที่ 14 รายงานผลการทดสอบ 40
ตารางที่ 15 รายละเอียดคลาส UsecaseInfo 42
ตารางที่ 16 รายละเอียดคลาส WndInfo 42
ตารางที่ 17 รายละเอียดคลาส ActivityInfo 42
ตารางที่ 18 รายละเอียดคลาส XMLManager 43
ตารางที่ 19 รายละเอียดคลาส XML_retrieve 43
ตารางที่ 20 รายละเอียดคลาส WndInfo 43
ตารางที่ 21 รูปแบบของรายงานผลการทดสอบ 46
ตารางที่ 22 ตาราง project 49
ตารางที่ 23 ตาราง test_result 49
ตารางที่ 24 ตาราง designer 50
ตารางที่ 25 ตาราง usecase_node 50
ตารางที่ 26 ตาราง uc_association 50
ตารางที่ 27 ตาราง acticity_node 51
ตารางที่ 28 ตาราง act_connector 51

หน้า

ตารางที่ 29 ตาราง act_swimlane 52
ตารางที่ 30 ตาราง wnd_node 52
ตารางที่ 31 ตาราง wnd_connector 52
ตารางที่ 32 ตาราง action_map 53
ตารางที่ 33 ผลการสร้างกรณีทดสอบของระบบบริการในห้องสมุด 62
ตารางที่ 34 คำอธิบายยูสเคส Update Book 77
ตารางที่ 35 คำอธิบายยูสเคส Add New Book 78
ตารางที่ 36 คำอธิบายยูสเคส Add member 79
ตารางที่ 37 คำอธิบายยูสเคส Borrow book 80
ตารางที่ 38 คำอธิบายยูสเคส Update Book Status 81

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทดสอบซอฟต์แวร์เป็นกิจกรรมที่สำคัญในกระบวนการผลิตซอฟต์แวร์ เพื่อให้ได้ซอฟต์แวร์ที่ทำงานได้ถูกต้องตรงตามความต้องการของลูกค้า และการทดสอบส่วนต่อประสานแบบกราฟิกกับผู้ใช้ หรือจียูไอ (Graphical User Interface: GUI) ซึ่งเป็นส่วนที่ติดต่อกับผู้ใช้โดยตรงนั้นก็เป็นสิ่งสำคัญ เนื่องจากปัจจุบันซอฟต์แวร์ทั่วๆ ไปมีการใช้ส่วนต่อประสานกราฟิกกับผู้ใช้กันมากมาย เพราะความสะดวกและใช้งานได้ง่ายกว่าการใช้งานแบบบรรทัดคำสั่ง (Command line) ที่ใช้ในสมัยก่อน ซึ่งหากมีข้อผิดพลาดเกิดขึ้นกับส่วนต่อประสานกราฟิกกับผู้ใช้ก็เสมือนระบบทำงานผิดพลาด แม้ว่าตัวแอพพลิเคชันโปรแกรมอาจทำงานได้ถูกต้องก็ตาม แต่การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ยังไม่มีวิธีการที่เป็นมาตรฐานสากล มีหลายงานวิจัยที่พยายามจะสร้างวิธีการที่จะทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ เช่น Memon et al. [1] ได้ทำการวิจัยเรื่องของการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ และได้นำเสนอการทดสอบโดยใช้แผนงานปัญญาประดิษฐ์ (AI planning) ในการสร้างกรณีทดสอบ (Test Case) ซึ่งแผนดังกล่าวได้มาจากการมองการทำงานในระดับของภารกิจ (Task) เนื่องจากถ้าทำการทดสอบจากโครงสร้างของส่วนต่อประสานกราฟิกกับผู้ใช้ซึ่งปกติสามารถแทนได้ด้วยเครื่องสถานะจำกัด (Finite State Machine) จะได้กรณีทดสอบเป็นจำนวนมหาศาล ซึ่งบางกรณีทดสอบไม่สามารถใช้งานได้ หรือใช้งานได้แต่ไม่มีประโยชน์ในการทำงานจริง แต่ในกรณีของการสร้างกรณีทดสอบจากระดับของภารกิจจะได้กรณีทดสอบที่สามารถใช้งานได้จริง โดยครอบคลุมทุกหน้าที่การทำงานของระบบและปริมาณของกรณีทดสอบไม่มากจนเกินไปด้วย

จากความเป็นมาข้างต้น จึงมีแนวคิดว่าควรจะสร้างกรณีทดสอบจากระดับของภารกิจและมีงานวิจัยของ Phillips et al. [2] นำเสนอการใช้ แผนภาพยูสเคส (Use Case Diagram) คำอธิบายยูสเคส (Use Case Description) และแผนภาพกิจกรรม (Activity Diagram) มาประยุกต์ใช้ร่วมกันเพื่อทำการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ ผู้วิจัยจึงมีแนวความคิดว่าถ้าสามารถระบุภารกิจได้ตั้งแต่ขั้นตอนการออกแบบระบบ ก็ควรจะสร้างกรณีทดสอบของส่วนต่อประสานกราฟิกกับผู้ใช้ได้ตั้งแต่ขั้นตอนนี้เช่นกัน ดังนั้นจึงมีแนวความคิดที่จะนำแผนภาพยูสเคส คำอธิบายยูสเคส และแผนภาพกิจกรรมมาประยุกต์ให้สามารถสร้างกรณีทดสอบของส่วนต่อประสานกราฟิกกับผู้ใช้ และนำข้อกำหนดของซอฟต์แวร์ (Software Specification) เป็นตัวบ่งชี้

ความครบถ้วนถูกต้องของหน้าที่การทำงานระบบ เพื่อเป็นการประหยัดเวลาในการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้เนื่องจากการออกแบบกรณีทดสอบสามารถทำไปพร้อมกับการสร้างตัวโปรแกรม เมื่อทำการสร้างตัวโปรแกรมสำเร็จก็สามารถทำการทดสอบได้ทันที นอกจากนี้ความเป็นอัตโนมัติของการสร้างกรณีทดสอบจะช่วยประหยัดเวลาและทรัพยากรที่ใช้ในขั้นตอนการทดสอบ กอปรกับแผนการทดสอบในงานวิจัยนี้จะเป็นการนำตัวดำเนินการ (Operator) ซึ่งเป็นชุดของเหตุการณ์ (Event) ที่ใช้ทดสอบมาเรียงต่อกัน โดยตัวดำเนินการหนึ่งๆ อาจใช้ได้ในหลายภารกิจ ดังนั้นจึงเป็นการประหยัดทรัพยากรในการสร้างแผนการทดสอบ โดยสามารถนำกลับมาใช้ใหม่ได้ (Reuse)

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อสร้างแผนสำหรับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากแผนภาพยูสเคส คำอธิบายยูสเคส แผนภาพกิจกรรม และแผนภาพดับเบิลยูเอ็นดี (WND diagram) โดยเน้นการทดสอบหน้าที่การทำงาน ซึ่งครอบคลุมทุกภารกิจในระบบ
2. เพื่อประเมินผลการใช้งานได้จริงของกรณีทดสอบที่สร้างขึ้น

1.3 ขอบเขตของการวิจัย

1. สร้างกรณีทดสอบโดยใช้หลักการของความครอบคลุมหน้าที่การทำงานของระบบ
2. พัฒนาเครื่องมือในการสร้างกรณีทดสอบจียูไอในรูปแบบของเอกสารเอกซ์เอ็มแอล (XML document) หลังจากได้แผนภาพยูสเคส คำอธิบายยูสเคส แผนภาพกิจกรรม และแผนภาพดับเบิลยูเอ็นดีแล้ว
3. แผนการทดสอบในงานวิจัยนี้ เงื่อนไขก่อนหน้าและเงื่อนไขตามหลังนั้นจะให้ผู้ออกแบบทดสอบเป็นผู้แทรกลงในกรณีทดสอบเอง
4. การประเมินการใช้งานได้จริงจะทำการนำระบบมาทำการทดสอบ 1 ระบบ โดยระบบดังกล่าวจะมีภารกิจไม่ต่ำกว่า 10 ภารกิจ

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1. ศึกษาวิธีการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ หลักการ ข้อจำกัด และงานวิจัยที่เกี่ยวข้อง
2. ศึกษาแผนภาพยูเอ็มแอล ได้แก่ แผนภาพยูสเคส คำอธิบายยูสเคส แผนภาพดับเบิลยูเอ็นดี และแผนภาพกิจกรรม

3. ศึกษาเอกซ์เอ็มแอลเพื่อนำมาจัดรูปแบบของแผนภาพต่างๆ และกรณีทดสอบ
4. ออกแบบวิธีการสร้างและรูปแบบของแผนการทดสอบจากแผนภาพยูสเคส คำอธิบายยูสเคส แผนภาพกิจกรรมและแผนภาพดับเบิลยูเอ็นดี
5. สร้างเครื่องมือในการสร้างแผนสำหรับการทดสอบ และทดสอบการทำงานของเครื่องมือ
6. สร้างเครื่องมือใช้เพื่อทดสอบกรณีทดสอบที่สร้างขึ้น รวมทั้งจัดเก็บและจัดทำรายงานผลการทดสอบ
7. วิเคราะห์และสรุปผล
8. จัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้วิธีการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ตั้งแต่ขั้นตอนการออกแบบระบบ
2. ได้เครื่องมือช่วยสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากเอกสารการออกแบบระบบ
3. ได้กรณีทดสอบที่เอื้อต่อการนำกลับมาใช้ใหม่

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 5 บทดังต่อไปนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึง ความเป็นมาและความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องในงานวิจัยนี้ บทที่ 3 กล่าวถึงรายละเอียดทั้งหมดของการสร้างกรณีทดสอบจียูไอจากยูเอ็มแอล บทที่ 4 แสดงรายละเอียดของการทดสอบและผลการทดสอบ และบทที่ 5 เป็นข้อสรุปและข้อเสนอแนะจากการวิจัย

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “GUI Test Case Generation from UML” โดย Yachai Limpiyakorn and Petnamkang Wongsuttipakorn, Proceedings of The 2005 International Conference on Software Engineering Research and Pratice, เล่ม 1 หน้า 323-329 ในงานประชุมวิชาการ “The 2005

International Conference on Software Engineering Research and Practice (SERP'05)" ณ รัฐเนวาดา ประเทศสหรัฐอเมริกา ระหว่างวันที่ 27-29 กรกฎาคม 2548

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้จะแบ่งเนื้อหาออกเป็น 2 ส่วนด้วยกัน โดยเนื้อหาในส่วนแรกจะกล่าวถึงทฤษฎีและแนวคิดพื้นฐาน ซึ่งประกอบด้วยการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ ยูเอ็มแอล และเนื้อหาในส่วนที่สองจะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 การทดสอบจียูไอ (GUI Testing)

การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้เป็นกระบวนการที่ใช้ทรัพยากรในการทดสอบเป็นจำนวนมาก ซึ่งปัจจุบันในการพัฒนาซอฟต์แวร์ส่วนใหญ่ยังไม่มีขั้นตอนและกระบวนการทดสอบที่ชัดเจนและมีประสิทธิภาพ

แม้ว่าจะมีงานวิจัยเป็นจำนวนมาก [3, 4, 5, 6, 7] เกี่ยวกับการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ แต่ส่วนใหญ่ยังคงอยู่ในขั้นตอนการวิจัย และยังไม่มีวิธีการที่ได้รับการยอมรับเป็นมาตรฐานในเชิงอุตสาหกรรม ทั้งนี้เพราะการทดสอบมีความซับซ้อนและมีลักษณะเฉพาะ ซึ่งต้องใช้ทั้งความรู้ แรงงาน และระยะเวลาในการดำเนินการ

วิธีการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้แนวทางหลักๆ ในปัจจุบัน แบ่งออกเป็น

1. การทดสอบโดยมนุษย์ คือ การทดสอบใช้งานโดยมนุษย์ ซึ่งปัจจุบันการพัฒนาซอฟต์แวร์เชิงอุตสาหกรรม ส่วนใหญ่ยังคงใช้วิธีนี้อยู่ ข้อดีของวิธีนี้ คือ สามารถทดสอบได้ง่าย ไม่ต้องเตรียมการหรือใช้ข้อมูลการออกแบบมาก แต่จะใช้เวลาในการทดสอบนาน และไม่สามารถบอกได้ว่าทดสอบเท่าใดจึงจะพอเพียง แนวทางการทดสอบแบ่งออกเป็น

    - การทดสอบโดยการทดลองใช้งาน (Walkthrough) คือ การทดสอบโดยการทดลองใช้งานโปรแกรมในส่วนการทำงานหลัก เพื่อหาข้อผิดพลาด
    - การทดสอบโดยการสังเกตการทำงานของผู้ใช้ (User observation) คือ การทดสอบที่ให้ผู้ใช้ได้ทดลองใช้งานจริง แล้วสังเกตข้อผิดพลาดที่ผู้ใช้พบ

- การทดสอบโดยการแจกจ่ายชุดทดสอบ (Beta copies distribution) คือ การทดสอบที่แจกจ่ายชุดทดสอบให้กับบุคคลที่สนใจไปทดลองใช้ และรายงานข้อผิดพลาดกลับมา

2. การทดสอบโดยเลียนแบบมนุษย์ คือ การทดสอบที่พยายามลดภาระในการทดสอบของมนุษย์ลง โดยเลียนแบบพฤติกรรมการทดสอบของมนุษย์ เพื่อให้ทดสอบได้มากขึ้นและเร็วขึ้น แต่ก็ยังคงมีข้อเสียในเรื่องความเพียงพอในการทดสอบ (coverage criteria) แนวทางการทดสอบแบ่งออกเป็น

   - การทดสอบโดยเครื่องมือบันทึกและเล่นซ้ำ (Capture & Replay) ส่วนใหญ่ใช้ในการทดสอบแบบรีเกรสชัน โดยเครื่องมือจะบันทึกลำดับของเหตุการณ์และผลลัพธ์ของการทดสอบ เพื่อนำกลับมาใช้ทดสอบใหม่ เมื่อมีการเปลี่ยนแปลงแก้ไข และอาจนำชุดทดสอบไปใช้กับโปรแกรมที่มีลักษณะการทำงานคล้ายกันได้
   - การทดสอบโดยใช้อัลกอริทึมเชิงพันธุกรรม (Genetic algorithm) [3] เป็นการทดสอบโดยพยายามที่จะเพิ่มความแปรผันให้กับชุดทดสอบ เพื่อให้ได้ชุดทดสอบใหม่ๆ ที่แตกต่างไปจากชุดทดสอบเดิม
   - การทดสอบโดยการสุ่ม (Random testing) [4] เป็นการทดสอบโดยการสุ่มสร้างเหตุการณ์ต่างๆ ขึ้น เพื่อตรวจสอบผลลัพธ์ที่ได้

3. การทดสอบโดยอัตโนมัติ คือ การทดสอบที่ใช้โปรแกรมในการสร้างและทดสอบชุดทดสอบ โดยเน้นความครบถ้วนของชุดทดสอบ ให้ครอบคลุมถึงทุกกรณีที่เป็นไปได้พิจารณาตามข้อกำหนดในการออกแบบ ซึ่งข้อดีคือ สามารถบอกได้ชัดเจนว่าทดสอบเท่าใดจึงจะพอเพียง แต่ต้องใช้ข้อมูลการออกแบบที่ชัดเจน และมีกรณีทดสอบจำนวนมาก

   - การทดสอบโดยใช้เครื่องสถานะ (State machine testing) [5] เป็นการทดสอบโดยพิจารณาระบบในรูปแบบของเครื่องสถานะ เพื่อตรวจสอบว่าตรงกับการออกแบบหรือไม่ ซึ่งจะมีจำนวนกรณีทดสอบมาก
   - การทดสอบโดยการกำหนดเป้าหมาย (Goal-driven testing) [6] เป็นการทดสอบการทำงานของระบบแต่ละหน้าที่ว่าทำงานได้ถูกต้องครบถ้วนตามที่

ได้ออกแบบไว้หรือไม่ โดยใช้กราฟการทำงานเพื่อกำหนดชุดทดสอบทั้งหมดที่เป็นไปได้ ซึ่งจะลดจำนวนกรณีทดสอบลง

4. การทดสอบโดยใช้วิธีการฟอร์มอล [7] คือ การทดสอบโดยการแปลงการออกแบบให้อยู่ในรูปแบบเชิงคณิตศาสตร์ เพื่อนำไปตรวจสอบกับชิ้นงาน ซึ่งเป็นวิธีที่มีความซับซ้อน และต้องอาศัยผู้เชี่ยวชาญ มักใช้ในระบบที่มีความสำคัญสูง

2.1.2 **ยูเอ็มแอล** (Unified Modeling Language) [8]

ยูเอ็มแอลเป็นภาษาที่ใช้ในการอธิบายโครงสร้าง และการทำงานของระบบซอฟต์แวร์ ซึ่งจะสามารถอธิบายแนวคิดเชิงวัตถุได้อย่างมีประสิทธิภาพ แผนภาพของยูเอ็มแอล ประกอบด้วย 9 แผนภาพหลัก แต่จะขออธิบายเฉพาะแผนภาพที่ใช้ในงานวิจัยนี้ ได้แก่

2.1.2.1 **แผนภาพยูสเคส** [8]

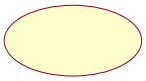
แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้กับยูสเคส โดยยูสเคสแสดงหน้าที่การทำงานของระบบ ส่วนประกอบของแผนภาพยูสเคส

1. ผู้ใช้ (Actor) แสดงบทบาทของผู้ใช้งานระบบ รูปที่ 1 แสดงสัญลักษณ์ที่ใช้แทนผู้ใช้งานระบบ

Actor

รูปที่ 1 ผู้ใช้งานระบบ

2. ยูสเคส (Use Case) แสดงหน้าที่การทำงานของระบบ โดยอาจเป็นหน้าที่การทำงานย่อย หรือคลาส ใช้วงรีเป็นสัญลักษณ์แทนยูสเคส ดังแสดงในรูปที่ 2



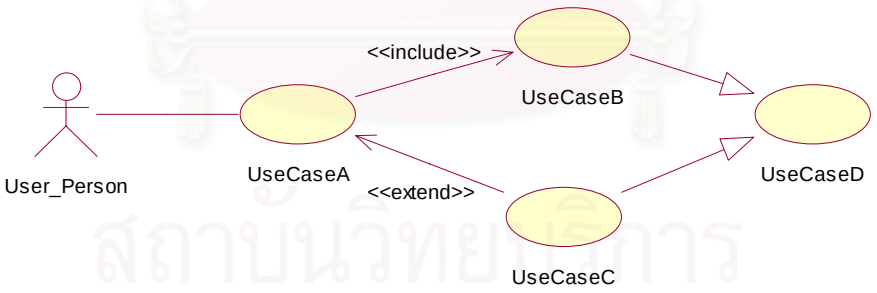
Use Case

รูปที่ 2 ยูสเคส

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

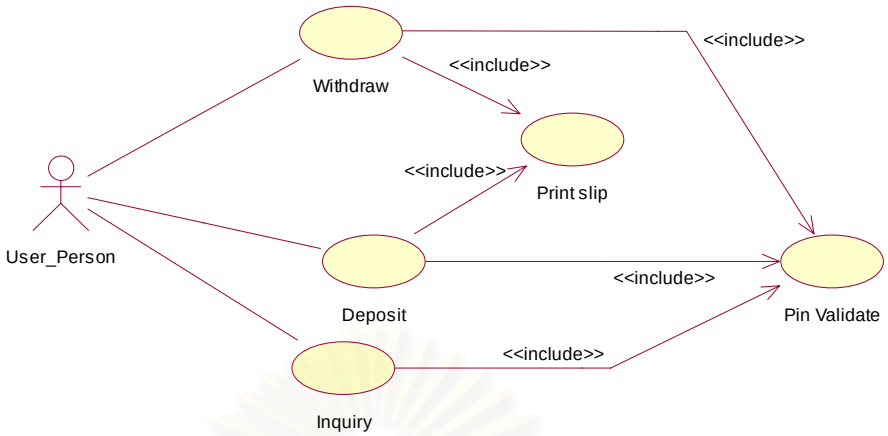
3. ความสัมพันธ์ของยูสเคส (Use Case Relationships) ความสัมพันธ์ของยูสเคสเกิดขึ้นเมื่อยูสเคสมีการเรียกหรือถูกเรียกใช้ โดยมากแล้วจะเกิดความสัมพันธ์แบบระหว่างผู้ใช้กับยูสเคสและความสัมพันธ์แบบอินคลูด รูปแบบของความสัมพันธ์แสดงในรูปที่ 3

- ความสัมพันธ์ (Association) แสดงความสัมพันธ์ระหว่างผู้ใช้กับยูสเคส แผนภาพยูสเคสใช้เส้นตรง
- อินคลูด (Include) เกิดขึ้นเมื่อยูสเคสหนึ่งเรียกใช้งานอีกยูสเคสหนึ่ง ใช้เส้นตรงที่มีลูกศรชี้ไปยังยูสเคสที่ถูกเรียกใช้ และมีคำว่า <<include>> กำกับบนเส้นลูกศร ตัวอย่างเช่นยูสเคสบี ถูกเรียกใช้โดยยูสเคสเอ
- เอกซ์เทนด์ (Extend) เกิดขึ้นเมื่อยูสเคสหนึ่งไม่สามารถทำงานได้ตามปกติจึงมียูสเคสอีกอันทำงานแทนเพื่อเป็นทางเลือก แผนภาพยูสเคสใช้เส้นตรงที่มีหัวลูกศรชี้จากยูสเคสที่ถูกเอกซ์เทนด์ และมีคำว่า <<extends>> กำกับบนเส้นลูกศร ตัวอย่างเช่นยูสเคสเอเอกซ์เทนด์ยูสเคสซี
- สืบทอด (Generalization) เกิดขึ้นเมื่อมีการสืบทอดระหว่างยูสเคส ใช้เส้นตรงที่มีหัวลูกศรเป็นรูปสามเหลี่ยมโปร่ง ชี้ไปยังยูสเคสต้นแบบ ตัวอย่างเช่นยูสเคสบีและยูสเคสซี สืบทอดจากยูสเคสดี



รูปที่ 3 ความสัมพันธ์ของยูสเคสแบบต่างๆ

ในงานวิจัยนี้จะใช้แผนภาพยูสเคสซึ่งใช้ในการอธิบายหน้าที่การทำงานของระบบ มาแสดงหน้าที่ทั้งหมดของระบบ เพื่อแสดงถึงตัวดำเนินการของระบบ และความสัมพันธ์ระหว่างหน้าที่ จะถูกนำมาใช้สร้างเป็นยูสเคสซีนาริโอของระบบ (Use Case Scenario) ดังแสดงตัวอย่างในรูปที่ 4 เป็นเค้าโครงการทำงานของการถามยอดบัญชี ซึ่งจะใช้ยูสเคสในตรวจสอบรหัส (Pin Validate) จากนั้นใช้ยูสเคสในสอบถามยอดเงิน (Inquiry)



รูปที่ 4 ตัวอย่างยูสเคสซีนาริโอของระบบ

2.1.2.2 **คำอธิบายยูสเคส** [9]

คำอธิบายยูสเคสใช้บอกถึงรายละเอียดของยูสเคสสามารถเขียนได้หลายรูปแบบไม่มีรูปแบบที่ตายตัว โดยคำอธิบายยูสเคสแบ่งออกได้เป็น 2 ระดับ ดังนี้

- ยูสเคสแบบคร่าว (Overview Use Case) บอกถึงความต้องการของระบบ ในระดับสูง ซึ่งให้รายละเอียดไม่มากนัก
- ยูสเคสจำเป็น (Essential Use Case) และยูสเคสจริง (Real Use Case) ยูสเคสจำเป็นบอกถึงข้อความน้อยที่สุดที่สามารถทำความเข้าใจหน้าที่การทำงานของระบบได้ ส่วนยูสเคสจริงบอกถึงขั้นตอนต่างๆ ในการทำงานของยูสเคส

รายละเอียดยูสเคสรูปแบบตารางหนึ่งสดมภ์ดังแสดงในตารางที่ 1 สามารถอธิบายได้ดังนี้

- Use Case Name: เป็นข้อความที่แสดงชื่อยูสเคสควรเป็นคำกริยา
- ID: หมายเลขประจำยูสเคส
- Importance level: ลำดับความสำคัญของยูสเคส
- Primary actor: เป็นข้อความแสดงชื่อบุคคล หรือระบบที่ติดต่อใช้งานยูสเคส
- Use case type: ประเภทของยูสเคส
- Stakeholders and interests: เป็นข้อความแสดงชื่อบุคคล หรือระบบที่ได้รับผลประโยชน์จากการทำงานของยูสเคส ซึ่งอาจเป็น Primary Actor ได้ และบอกประโยชน์ที่ Stakeholder ได้รับ
- Brief description: ข้อความทำความเข้าใจยูสเคส

- Trigger: เป็นข้อความแสดงกิจกรรมที่ทำให้เริ่มการทำงานยูสเคส
- Type: โดยทั่วไปเป็น external
- Relationships: แสดงความสัมพันธ์ แบ่งออกเป็น 4 รูปแบบ คือ
  1. Association: ผู้ใช้งานที่มีความสัมพันธ์
  2. Include: ยูสเคสที่มีความสัมพันธ์แบบอินคลูด
  3. Extend: ยูสเคสที่มีความสัมพันธ์แบบเอกซ์เทน
  4. Generalization: ยูสเคสที่มีความสัมพันธ์เป็นแบบสืบทอด
- Normal flow of events: เป็นข้อความแสดงลำดับเหตุการณ์สำเร็จ ซึ่งระบุเหตุการณ์ที่เกิดขึ้นเป็นข้อๆ
- Subflows: ข้อความแสดงลำดับเหตุการณ์ย่อยเป็นข้อๆ
- Alternate/Exceptional flows: เป็นข้อความแสดงเงื่อนไขที่ทำให้เกิดเหตุการณ์อื่นและลำดับเหตุการณ์ที่เกิดขึ้นเมื่อเงื่อนไขนั้นเป็นจริง ลำดับเหตุการณ์ที่เกิดขึ้นจะถูกระบุเป็นข้อๆ

ตารางที่ 1 คำอธิบายยูสเคส

Use Case Name:		ID:		Importance level:	
Primary actor:		Use case type:			
Stakeholders and interests:					
Brief description:					
Trigger:					
Type:					
Relationships:					
Association:					
Include:					
Extend:					
Generalization:					
Normal flow of events:					
Subflows:					
Alternate/Exceptional flows:					

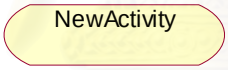
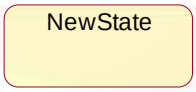
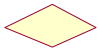
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

โดยงานวิจัยนี้จะใช้คำอธิบายยูสเคส ในระดับยูสเคสจริง เนื่องจากให้รายละเอียดการทำงานของผู้ใช้ได้ละเอียดกว่าแบบอื่นๆ และมีการเพิ่มข้อมูลบางส่วนลงไปเพื่อความสามารถปรับใช้งานให้เหมาะสมกับการสร้างแผน เช่น Precondition (เงื่อนไขก่อนหน้า) เพื่อใช้บอกเหตุการณ์ที่ต้องเกิดขึ้นก่อนจะที่เหตุการณ์ในยูสเคสจะสามารถเกิดขึ้นได้ Postcondition (เงื่อนไขตามหลัง) เพื่อใช้บอกเหตุการณ์ที่เกิดขึ้นหลังจากทำยูสเคสสำเร็จ และสามารถนำไปเขียนแผนภาพกิจกรรมได้ง่าย

2.1.2.3 **แผนภาพกิจกรรม** [8]

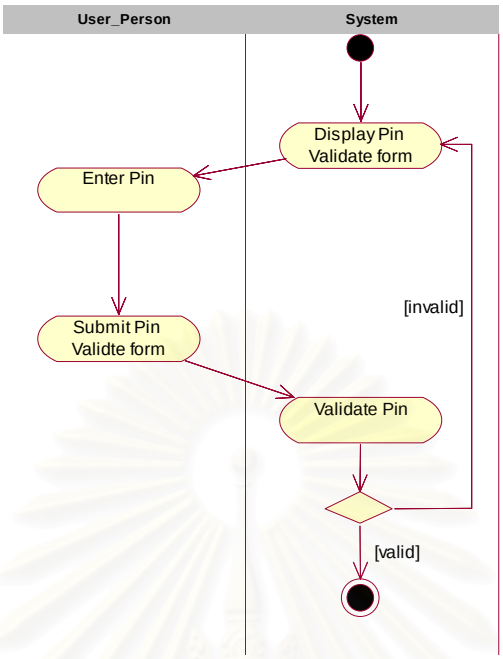
แผนภาพกิจกรรม เป็นการแสดงลำดับการทำงาน ซึ่งประกอบด้วยส่วนประกอบ ดังตารางที่ 2

ตารางที่ 2 ส่วนประกอบของแผนภาพกิจกรรม

ชื่อ	ลักษณะ	ความหมาย
วงกลมทึบ (Solid Circle)		จุดเริ่มต้น
รูปทรงแคปซูล (Capsule)	 และ 	เขียนด้วยรูปทรงสี่เหลี่ยมมุมมน และเขียนกำกับด้วยกริยาเพื่อบอกสถานะกิจกรรมหรือสถานะการกระทำ ซึ่งแบ่งออกเป็น 2 ประเภท คือ สถานะกิจกรรม (Activity State) สถานะที่สามารถแยกย่อยเป็นสถานะกิจกรรม หรือสถานะการกระทำได้ สถานะการกระทำ (Action State) สถานะที่ไม่สามารถแตกย่อยออกได้อีก
การส่งต่อ (Transition)		เครื่องหมายลูกศรระหว่างส่วนประกอบต่างๆ ของแผนภาพ ซึ่งสามารถเขียนกำกับด้วยเงื่อนไขในเครื่องหมายวงเล็บก้ามปู (brackets) ในกรณีที่เป็นเส้นออกจากไดมอนเพื่อบอกเงื่อนไข เช่น [ตรวจสอบแล้ว]
การตัดสินใจ (Decisions)		การตัดสินใจเป็นจุดที่ใช้ในการตัดสินใจหรือการทำวนซ้ำ ซึ่งถ้ามีลูกศรหลายอัน แสดงถึง

		การตัดสินใจ หรือเงื่อนไข และจำต้องทำการกำกับด้วยเงื่อนไขที่อยู่ภายในวงเล็บก้ามปู หรือใช้คำว่า else
ตาวัว (Bull's eye)		เป็นลักษณะของวงกลมทึบภายในวงกลมโปร่ง แสดงจุดสิ้นสุด
แท่งประสานเวลา (Synchronization bar)		เส้นทึบแสดงการทำงานแบบขนาน
ฟอร์ก (Fork)		ลูกศรเข้า 1 อัน และลูกศรออกหลายอัน เป็นการเริ่มการทำงานแบบขนาน
จอย (Join)		ลูกศรเข้าหลายอัน และลูกศรออก 1 อัน แสดงการสิ้นสุดการทำงานแบบขนาน โดยรอให้สิ้นสุดพร้อมกัน
สวิมเลน (Swimlanes)		เป็นการแบ่งการทำงานเป็นส่วนๆ ว่าใครทำสถานะกิจกรรมหรือสถานะการกระทำใด ซึ่งสามารถเขียนได้ทั้งแนวนอนและแนวตั้ง โดยแต่ละเลนสามารถเป็นได้ทั้ง หน่วย บทบาท คลาส คอมโพเนนท์ ยูสเคส

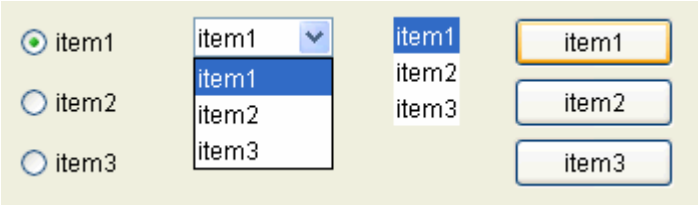
ในงานวิจัยนี้ใช้แผนภาพกิจกรรมในการแยกการทำงานในส่วนของผู้ใช้ออกจากส่วนอื่นๆ ของระบบ เพื่อใช้พิจารณาในการสร้างกรณีทดสอบเฉพาะส่วนที่มีปฏิสัมพันธ์กับผู้ใช้เท่านั้น โดยอ้างอิงจากการกระทำที่อยู่ในสวิมเลนของผู้ใช้ ดังแสดงตัวอย่างรูปที่ 5 สวิมเลนของผู้ใช้คือสวิมเลนชื่อ User_Person ดังนั้นการกระทำของผู้ใช้คือ Enter Pin และ Submit Pin Validate Form จะนำไปแปลงเป็นการกระทำในกรณีทดสอบ



รูปที่ 5 ตัวอย่างแผนภาพกิจกรรมของหน้าที่ตรวจรหัส

2.1.2.4 **แผนภาพแผนที่หน้าต่าง หรือดับเบิลยูเอ็นดี** [9]

เป็นแผนภาพที่คล้ายกับแผนภาพสเตท (Statechart Diagram) โดยเป็นแผนภาพเพิ่มเติมจากแผนภาพปกติ 9 แผนภาพในยูเอ็มแอล [10] เพื่อให้มีความเหมาะสมกับการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งแผนภาพสเตทนั้นแสดงการเปลี่ยนสถานะของวัตถุ แต่แผนภาพดับเบิลยูเอ็นดีแสดงการเปลี่ยนสถานะของส่วนต่อประสานกราฟิกกับผู้ใช้ อีกทั้งยังแสดงประเภทและชื่อของคอมโพเนนท์ภายในหน้าจอ ฟอร์ม หรือกล่องข้อความที่ปรากฏขึ้นในโปรแกรมทั้งหมดและการเปลี่ยนหน้าจอ โดยเหตุการณ์ที่เกิดขึ้นกับคอมโพเนนท์ในหน้าจอ ซึ่งข้อมูลเหล่านี้เอื้อต่อการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ให้มีความเป็นอัตโนมัติมากขึ้น อีกเหตุผลหนึ่งที่มีการใช้แผนภาพดับเบิลยูเอ็นดี คือการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ที่มีความต้องการเดียวกันสามารถออกแบบได้หลายรูปแบบ ยกตัวอย่างเช่น การแสดงข้อมูลให้เลือกจากชุดข้อมูล 1 ชุด ดังรูปที่ 6 หากไม่มีการกำหนดรูปแบบที่ชัดเจนจะเป็นการยากต่อการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ให้มีความเป็นอัตโนมัติ ดังนั้นในงานวิจัยนี้จึงใช้แผนภาพดับเบิลยูเอ็นดีเป็นเครื่องมือในการกำหนดประเภทของส่วนต่อประสานกราฟิกกับผู้ใช้ แผนภาพดับเบิลยูเอ็นดีประกอบด้วยส่วนต่างๆ ดังตารางที่ 3

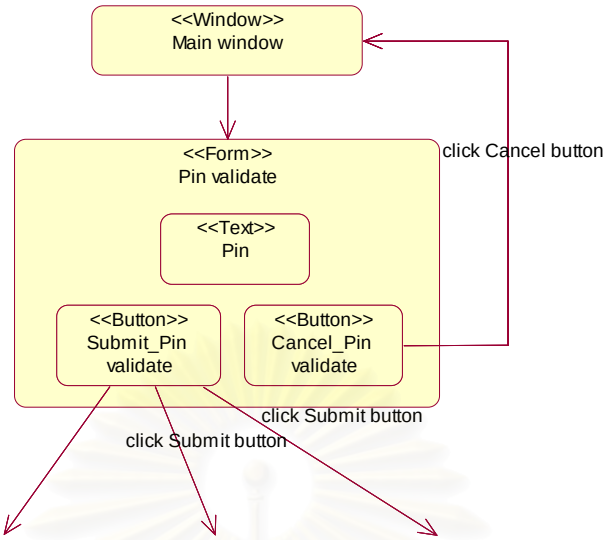


รูปที่ 6 การออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ จากความต้องการเหมือนกัน

ตารางที่ 3 ส่วนประกอบของแผนภาพดับเบิลยูเอ็นดี

ชื่อ	ลักษณะ	ความหมาย
กล่อง (Box)		คอมโพเนนท์ เช่น หน้าต่าง ปุ่ม เป็นต้น โดยสามารถมีส่วนประกอบของส่วนต่อประสานกราฟิกกับผู้ใช้ภายในได้ เพื่อใช้ในการออกแบบหน้าจอ
การส่งผ่าน (Transitions)	Click Submit Buttton →	เป็นการเปลี่ยนสถานะ โดยจะมีการตั้งชื่อของการกระทำที่ทำให้เกิดการเปลี่ยนสถานะ จากสถานะหนึ่งไปยังอีกสถานะหนึ่ง เช่น click Submit Button คือจะเกิดการเปลี่ยนสถานะก็ต่อเมื่อมีการกดปุ่ม Submit
รูปแบบ (Stereotype)	<<button>> Submit	แสดงประเภทส่วนประกอบของส่วนต่อประสานกราฟิกกับผู้ใช้ที่อยู่ในกล่อง โดยเป็นข้อความในเครื่องหมาย “<<” และ “>>” จะแสดงประเภทของส่วนประกอบของส่วนต่อประสานกราฟิกกับผู้ใช้ เช่น <<window>> แสดงถึงหน้าต่าง เป็นต้น

ในงานวิจัยนี้ใช้แผนภาพดับเบิลยูเอ็นดีในการระบุถึงส่วนประกอบต่างๆ ในหน้าจอ และการส่งต่อของหน้าจอต่างๆ จากเหตุการณ์ที่เกิดขึ้นกับคอมโพเนนท์ในหน้าจอนั้นๆ ซึ่งจะนำมาใช้ในการสร้างกรณีทดสอบในระดับของเหตุการณ์ ที่สามารถนำไปใช้ทดสอบได้จริง แสดงตัวอย่างแผนภาพดับเบิลยูเอ็นดี ดังรูปที่ 7



รูปที่ 7 ตัวอย่างแผนภาพดับเบิลยูเอ็นดี

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 “Plan Generation for GUI Testing” [1]

งานวิจัยนี้ได้นำเสนอแนวคิดในการทดสอบจียูไอ โดยการใช้แผนสำหรับการทดสอบที่เป็นเทคนิคทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) โดยจะให้ผู้ออกแบบการทดสอบทำการกำหนดภารกิจ (Task) ของผู้ใช้ ในรูปแบบของสถานะเริ่มต้น (Initial state) และสถานะสิ้นสุด (Goal state) แทนที่จะเป็นลำดับของเหตุการณ์ทั้งหมดที่จะเป็นไปได้ ทำให้ลดความซับซ้อนในการทดสอบลง โดยผู้ออกแบบการทดสอบจะเป็นผู้กำหนดตัวดำเนินการ (Operator) ที่เป็นตัวแทนของลำดับเหตุการณ์ของจียูไอ (GUI events) และแผนสำหรับการทดสอบของแต่ละภารกิจ คือ แผนการทดสอบ ซึ่งสามารถสร้างได้อย่างอัตโนมัติด้วยเครื่องมือพาธ (Planning Assisted Tester for Graphical user interface Systems: PATH) โดยข้อมูลเข้าจะเป็น ชุดของตัวดำเนินการและเหตุการณ์ของแต่ละตัวดำเนินการทั้งหมดของระบบ ซึ่งมีการกำหนดรูปแบบของตัวดำเนินการและเหตุการณ์ต่างๆ ไว้ จากนั้นเมื่อจะทำการสร้างกรณีทดสอบจากแผนสำหรับการทดสอบ จะทำการแปลงตัวดำเนินการแต่ละตัวเป็นเหตุการณ์ตามที่กำหนดไว้ในขั้นต้น ซึ่งผลลัพธ์ที่ได้จะเป็นลำดับของเหตุการณ์ทั้งหมด

เหตุการณ์ของส่วนต่อประสานกราฟิกกับผู้ใช้แบ่งออกได้เป็น 3 ประเภท คือ

- เหตุการณ์ที่ไม่ควบคุมโฟกัส (Unrestricted-focus events) คือเหตุการณ์ที่ทำการเปิดหน้าต่างที่ไม่ควบคุมโฟกัสของผู้ใช้ให้อยู่ที่หน้าต่างนั้น เช่น การเปิดเมนู
- เหตุการณ์ที่ควบคุมโฟกัส (Restricted-focus events) คือเหตุการณ์ที่ทำการเปิดหน้าต่างที่ควบคุมโฟกัสของผู้ใช้ให้อยู่ที่หน้าต่างนั้น จนกว่าจะทำการปิดหน้าต่างนั้น เพื่อสิ้นสุดการทำงาน เช่น การเปิดหน้าต่างการพิมพ์ จะต้องทำงานในหน้าต่างนั้นให้เสร็จจนกระทั้งมีการกดปุ่มตกลง (OK) หรือปุ่มยกเลิก (Cancel)
- เหตุการณ์ที่ปฏิสัมพันธ์กับระบบ (System-interaction events) คือเหตุการณ์ที่มีปฏิสัมพันธ์กับการทำงานของระบบ เช่น การบันทึกแฟ้ม

เหตุการณ์เหล่านี้ จะนำมาใช้ในการแยกประเภทของเหตุการณ์ในการกำหนด ตัวดำเนินการและชุดของเหตุการณ์สำหรับตัวดำเนินการ

รูปแบบของแผนสำหรับการทดสอบ จะประกอบด้วย ชุดของตัวดำเนินการเรียงเป็นลำดับ ข้อดีของการสร้างแผนสำหรับการทดสอบคือ เมื่อมีการเปลี่ยนแปลงลำดับของเหตุการณ์ในตัวดำเนินการ จะไม่ส่งผลกระทบต่อแผนที่ได้สร้างไว้ ทำให้แผนสำหรับการทดสอบเดิมยังสามารถใช้งานได้อยู่ เพียงแต่เมื่อทำการสร้างกรณีทดสอบใหม่ให้ใช้เหตุการณ์ชุดใหม่แทนเหตุการณ์ชุดเดิม

2.2.2 “Extending UML Use Case Modelling to Support Graphical User Interface Design” [2]

งานวิจัยนี้แสดงการขยายยูเอ็มแอลเพื่อใช้ในการพัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งใช้คำอธิบายยูสเคสมาใช้ในการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ และมีการเปรียบเทียบคำอธิบายยูสเคส 3 รูปแบบ ได้แก่ การใช้แผนภาพกิจกรรม การเขียนบรรยาย และการใช้ตาราง ซึ่งไม่มีรูปแบบใดเหมาะสมกับการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ จึงมีการขยายคำอธิบายยูสเคสโดยใช้หลักของกรอบการทำงานแบบยูดีพี (Unified Development Process: UDP) ซึ่งมีทั้งการอธิบายด้วยคำบรรยายและแผนภาพ ซึ่งจะทำให้ผู้ออกแบบสามารถเห็นภาพของหน้าจอที่ต้องออกแบบได้ง่ายขึ้น

งานวิจัยดังกล่าวได้แสดงให้เห็นถึงความสามารถในการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ จากยูเอ็มแอล ซึ่งผู้วิจัยได้ใช้เป็นแนวทางในการนำแผนภาพยูสเคส คำอธิบายยูสเคส และแผนภาพกิจกรรม มาประยุกต์ใช้ในการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จาก

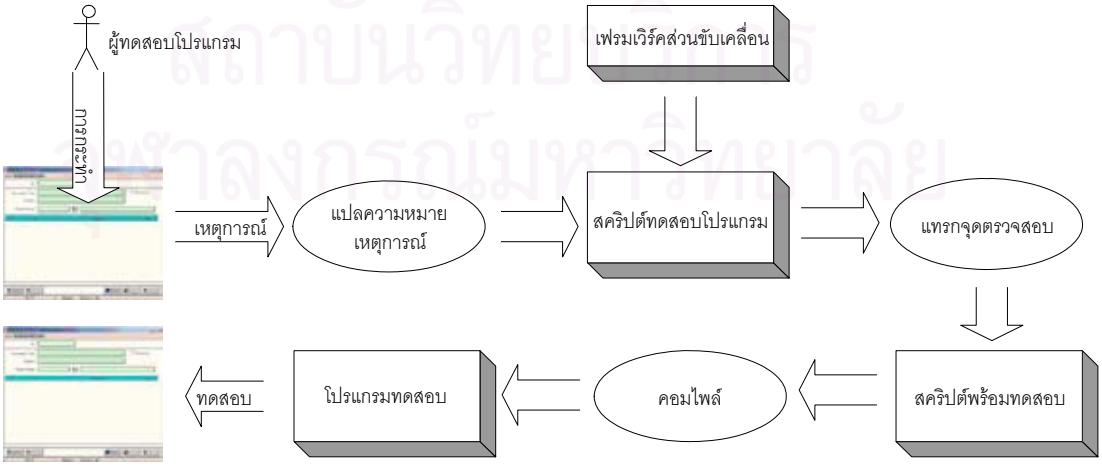
แผนภาพดังกล่าว แต่อาจต้องมีการเพิ่มเติมแผนภาพบางแผนภาพเข้าไปเพื่อช่วยเอื้อต่อการสร้างกรณีทดสอบแบบอัตโนมัติ

2.2.3 “ระบบอัตโนมัติช่วยทดสอบหน้าที่ของซอฟต์แวร์ส่วนติดต่อผู้ใช้แบบกราฟฟิกส์” [11]

งานวิจัยนี้ได้ใช้วิธีการบันทึกและเล่นซ้ำ (Capture and Replay หรือ Record and Play back) มาพัฒนาระบบอัตโนมัติช่วยทดสอบหน้าที่ของซอฟต์แวร์ส่วนต่อประสานกราฟิกกับผู้ใช้ หลักการการทำงานของวิธีบันทึกและเล่นซ้ำ ประกอบด้วยส่วนแคปเจอร์ ซึ่งจะบันทึกปฏิสัมพันธ์ของผู้ทดสอบเก็บไว้เป็นสคริปต์ และส่วนรีเพลย์ ซึ่งจะสร้างปฏิสัมพันธ์เลียนแบบการกระทำของผู้ทดสอบขึ้นตามสคริปต์

กระบวนการทำงานหลักของเครื่องมือที่พัฒนาขึ้นในงานวิจัยนี้ แสดงในรูปที่ 8 โดยเครื่องมือที่พัฒนาขึ้นใช้ทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ที่เขียนโดยภาษาจาวา และได้เพิ่มฟังก์ชันการโปรแกรมให้ผู้ทดสอบสามารถแก้ไขสคริปต์ เพิ่มจุดตรวจสอบเปรียบเทียบค่าที่คาดหวังกับค่าที่อ่านได้ ภายหลังการกระทำหนึ่งๆ ของผู้ใช้

อย่างไรก็ตาม เครื่องมือช่วยทดสอบซอฟต์แวร์ส่วนต่อประสานกราฟิกกับผู้ใช้แบบบันทึกและเล่นซ้ำ จะมีความเป็นกึ่งอัตโนมัติ ถ้าพิจารณาจากการสร้างกรณีทดสอบที่อาศัยการกดปุ่มหรือปฏิสัมพันธ์ของผู้ทดสอบ เพื่อแปลเป็นสคริปต์ทดสอบเก็บไว้ นอกจากนี้ ความครอบคลุมของกรณีทดสอบจะขึ้นอยู่กับมนุษย์ คือผู้ออกแบบการทดสอบ โดยไม่รวมอยู่ในความสามารถของเครื่องมือบันทึกและเล่นซ้ำ ซึ่งอาจทำให้ไม่ได้ความครอบคลุมการทดสอบที่เพียงพอ



รูปที่ 8 กระบวนการทำงานหลักของระบบ

2.2.4 “A Method to Automate User Interface Testing Using Variable Finite State Machines” [12]

งานวิจัยนี้ได้นำเสนอวิธีการในการทดสอบแบบอัตโนมัติ ซึ่งใช้การสร้างฟอร์มอลโมเดลของส่วนต่อประสานกับผู้ใช้ที่เรียกว่าเครื่องสถานะจำกัดแบบมีตัวแปร หรือวีเอฟเอสเอ็ม (Variable Finite State Machine: VFSM) โดยโมเดลนี้สามารถที่จะแปลงเป็นเครื่องสถานะจำกัด ซึ่งจะทำให้สามารถทำการทดสอบได้

วิธีการนี้ได้ถูกนำไปใช้กับส่วนต่อประสานกับผู้ใช้ของระบบนำทางชื่อ นาวิเกเตอร์ ทู (Navigator II) ซึ่งเป็นระบบนำทางแบบพกพาซึ่งใช้กับการซ่อมแซมเครื่องบินส่วนตัว โดยวีเอฟเอสเอ็มได้ถูกสร้างขึ้น และใช้ในการสร้างกรณีทดสอบ 5968 กรณี ซึ่งแต่ละกรณีทดสอบจะประกอบไปด้วยลำดับของข้อมูลเข้า และลำดับของข้อมูลออกที่คาดหวังซึ่งการทดสอบ ได้กระทำโดยการนำกรณีทดสอบเหล่านี้ไปทดสอบกับเครื่องมือจริง

งานวิจัยดังกล่าวยังมีข้อจำกัด คือ จะต้องทำการเขียนข้อกำหนดของซอฟต์แวร์ให้เป็นเครื่องสถานะจำกัดแบบมีตัวแปรให้ได้ซึ่งไม่สะดวกมากนัก อีกทั้งกรณีทดสอบที่สร้างขึ้นก็มีบางกรณีทดสอบที่ไม่สามารถใช้งานได้เช่นกัน เพราะเป็นการสร้างกรณีทดสอบแบบอัตโนมัติจากตัวเครื่องสถานะจำกัดที่แปลงมาจากเครื่องสถานะจำกัดแบบมีตัวแปร ทำให้บางกรณีทดสอบอาจย้อนกลับมาที่จุดเริ่มต้นในตัวเครื่องสถานะจำกัด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การสร้างกรณีทดสอบจียูไอจากยูเอ็มแอล

บทนี้กล่าวถึงการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากยูเอ็มแอล โดยแบ่งเนื้อหาออกเป็น 2 ส่วนด้วยกัน คือ ส่วนแรกอธิบายการทำงานโดยรวมของขั้นตอนวิธีการดำเนินการวิจัย และส่วนที่สองจะกล่าวถึงรายละเอียดของวิธีการที่นำเสนอ ซึ่งประกอบด้วยการทำงาน 3 ส่วน คือ 1) รูปแบบและวิธีการเขียนแผนภาพยูเอ็มแอลเพื่อใช้ในการสร้างกรณีทดสอบ 2) วิธีการสร้างกรณีทดสอบจากยูเอ็มแอล 3) รายละเอียดของเครื่องมือสร้างและทดสอบกรณีทดสอบ

3.1 ภาพรวมของวิธีการที่นำเสนอ

งานวิจัยนี้ได้สร้างวิธีการและเครื่องมือเพื่อใช้สร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากยูเอ็มแอล โดยกรณีทดสอบที่สร้างขึ้นครอบคลุมทุกภารกิจของระบบซึ่งมีปฏิสัมพันธ์กับผู้ใช้ โดยอ้างอิงจากแผนภาพยูเอ็มแอลซึ่งประกอบด้วย แผนภาพยูสเคส คำอธิบายยูสเคส แผนภาพดับเบิลยูเอ็นดี และแผนภาพกิจกรรม ข้อมูลต่างๆ ในแผนภาพเหล่านี้จะถูกแปลงและจัดเก็บในรูปของตัวดำเนินการและแผนการทดสอบ เพื่อใช้ในการสร้างกรณีทดสอบและสคริปต์ทดสอบ อีกทั้งยังสามารถนำสคริปต์ทดสอบที่สร้างได้ทดสอบจริงกับระบบที่สร้างตามแผนภาพยูเอ็มแอลโดยใช้วิธีการเขียนที่กำหนด และสร้างรายงานผลการทดสอบได้

ในงานวิจัยนี้ ภารกิจ หมายถึง หน้าที่การทำงานที่ผู้ใช้สามารถกระทำต่อระบบได้ โดยพิจารณาจากยูสเคสที่มีความสัมพันธ์โดยตรงต่อผู้ใช้ ซึ่งแต่ละภารกิจสามารถประกอบด้วยยูสเคสตั้งแต่ 1 ยูสเคสขึ้นไป โดยแต่ละยูสเคสสามารถมีการทำงานได้ตั้งแต่ 1 เส้นทางการทำงานขึ้นไป ดังนั้นแต่ละภารกิจของระบบสามารถสร้างเป็นกรณีทดสอบได้ตั้งแต่ 1 กรณีทดสอบขึ้นไป โดยพิจารณาจากยูสเคสทั้งหมดที่ทำงานร่วมกัน

ภาพรวมของวิธีการแสดงดังรูปที่ 9 ขั้นตอนแรกทำการวาดแผนภาพต่างๆ ลงในเครื่องมือเรชันเนล โรส ตามข้อกำหนดการเขียนรายละเอียดซึ่งจะอธิบายในหัวข้อถัดไป จากนั้นสร้างแผนภาพกิจกรรม จากคำอธิบายยูสเคส โดยเขียนแผนภาพกิจกรรม 1 แผนภาพต่อ 1 ยูสเคส เมื่อวาดแผนภาพทั้งหมดครบแล้วทำการแปลงแผนภาพดังกล่าวเป็นแฟ้มเอกซ์เอ็มแอล โดยใช้เครื่องมือยูนิซิส โรสเอ็กซ์เอ็มไอ ทูล (Unisys Rose XMI tool) ขั้นตอนต่อไป ทำการคัดแยกข้อมูลที่จำเป็นเก็บไว้ในฐานข้อมูล เพื่อใช้ในการสร้างตัวดำเนินการและแผนการทดสอบ หากต้องการ

แก้ไขเปลี่ยนแปลงการกระทำต่อคอมโพเนนท์ใดในตัวดำเนินการก็สามารถทำได้ จากนั้นเป็นขั้นตอนการสร้างกรณีทดสอบจากตัวดำเนินการและแผนการทดสอบซึ่งจะเก็บอยู่ในรูปแบบของแฟ้มเอกซ์เอ็มแอล

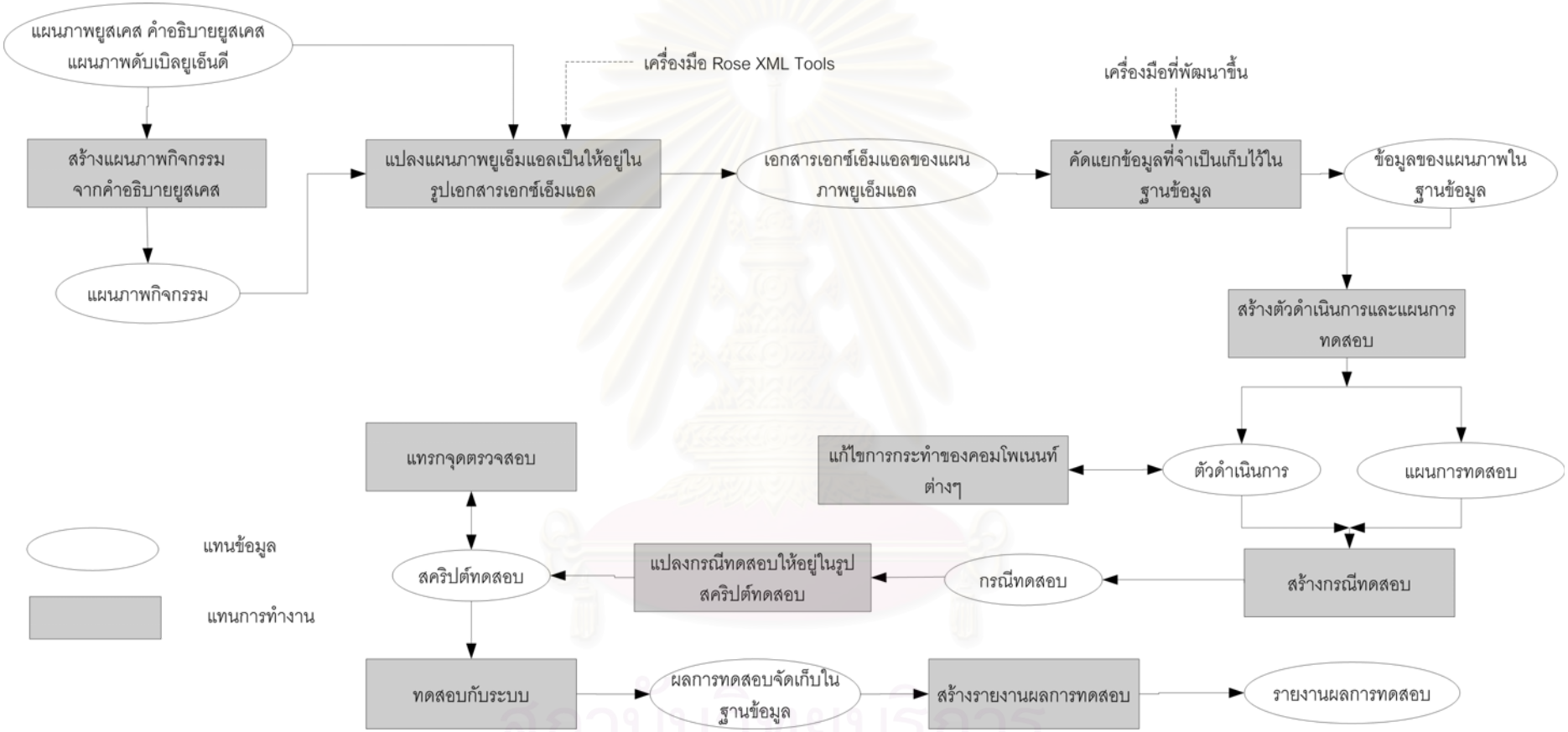
เมื่อได้กรณีทดสอบในรูปของแฟ้มเอกซ์เอ็มแอลแล้ว ทำการแปลงเป็นสคริปต์ทดสอบตามรูปแบบกรอบการทำงานเจเอฟซียูนิต (jfcUnit) จึงทำการทดสอบกับระบบจริงรวมทั้งเก็บข้อมูลผลการทดสอบลงในฐานข้อมูล เพื่อใช้ในการสร้างรายงานผลการทดสอบต่อไป

3.2 รูปแบบและวิธีการเขียนแผนภาพยูเอ็มแอลเพื่อใช้ในการสร้างกรณีทดสอบ

แผนภาพยูเอ็มแอลที่ใช้ในงานวิจัยนี้ประกอบด้วย แผนภาพยูสเคส คำอธิบายยูสเคส แผนภาพดับเบิลยูเอ็นดี และแผนภาพกิจกรรม โดยแผนภาพดับเบิลยูเอ็นดีจะเป็นแผนภาพเสริมจากแผนภาพหลักที่ใช้กันโดยทั่วไป สาเหตุที่ต้องใช้แผนภาพดับเบิลยูเอ็นดีเนื่องจากแผนภาพดังกล่าวอธิบายหน้าจอที่ปรากฏของส่วนต่อประสานกราฟิกกับผู้ใช้รวมทั้งเส้นทางของการส่งผ่านหน้าจอและรายละเอียดที่จำเป็นในการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ เช่น ประเภทของคอมโพเนนท์ ชื่อของคอมโพเนนท์ เป็นต้น ในแผนภาพแต่ละแผนภาพจะมีวิธีการเขียนรายละเอียดกำหนดไว้ เพื่อความสามารถในการนำข้อมูลดังกล่าวมาสร้างเป็นกรณีทดสอบได้

ในงานวิจัยนี้จะเขียนแผนภาพต่างๆ โดยใช้เครื่องมือเรชันเนล โรส (Rationale Rose) ซึ่งติดตั้งเครื่องมือโรส เอกซ์เอ็มแอล ทูล และแฟ้มคุณสมบัติของแผนภาพ โดยจะแสดงวิธีการเขียนแผนภาพต่างๆ โดยละเอียดในภาคผนวก ก

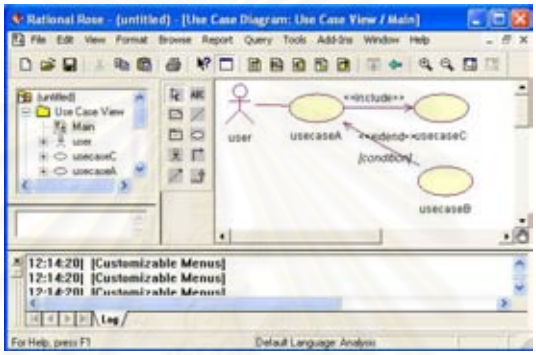
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 9 ภาพรวมการทำงานของการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากยูเอ็มแอล

3.2.1 การวาดแผนภาพยูสเคส

แผนภาพยูสเคสเขียนในระดับยูสเคสจริงซึ่งอธิบายการทำงานของระบบโดยละเอียดระดับ 1 ยูสเคสต่อ 1 หน้าที่การทำงาน โดยวาดลงในเครื่องมือเรชันเนล โรส ในส่วนมุมมองยูสเคส (Use case view) ซึ่งจะมีการวาดเหมือนกับการวาดแผนภาพยูสเคสโดยทั่วไป กล่าวคือมี ผู้ใช้ยูสเคส และความสัมพันธ์ โดยความสัมพันธ์จะแสดงประเภทของความสัมพันธ์ไว้ที่เส้นความสัมพันธ์ ดังแสดงในรูปที่ 10



รูปที่ 10 ตัวอย่างแผนภาพยูสเคสในเครื่องมือเรชันเนล โรส

ความสัมพันธ์ของยูสเคส ในงานวิจัยนี้ กำหนดการเขียนความสัมพันธ์ระหว่างยูสเคส ไว้ 3 รูปแบบ คือ ความสัมพันธ์แบบสืบทอด ความสัมพันธ์แบบอินคลูด และความสัมพันธ์แบบเอกซ์เทนด์ โดยมีรายละเอียดการเขียนแผนภาพดังนี้

1. ความสัมพันธ์แบบสืบทอด เขียนความสัมพันธ์แบบสืบทอดโดยใช้ลูกศรหัวโปร่งตามปกติ โดยหัวลูกศรชี้ไปยังยูสเคสแม่
2. ความสัมพันธ์แบบอินคลูด เขียนความสัมพันธ์โดยเลือกแบบอินคลูด จะปรากฏคำว่า “<<include>>” บนเส้นความสัมพันธ์ ซึ่งหัวลูกศรจะชี้ไปยังยูสเคสที่ถูกเรียกใช้
3. ความสัมพันธ์แบบเอกซ์เทนด์ เขียนความสัมพันธ์โดยเลือกแบบเอกซ์เทนด์ จะปรากฏคำว่า “<<extend>>” บนเส้นความสัมพันธ์ ซึ่งหัวลูกศรจะชี้ไปยังยูสเคสที่เรียกใช้

3.2.1 การเขียนคำอธิบายยูสเคส

คำอธิบายยูสเคสในงานวิจัยนี้จะเขียนโดยใช้เครื่องมือเรชันเนล โรส โดยอยู่ในส่วนคุณสมบัติของยูสเคส และได้ตัดเอาข้อมูลบางส่วนของคำอธิบายยูสเคสซึ่งซ้ำซ้อนกับข้อมูลที่

ปรากฏในแผนภาพยูสเคสออกไป แต่ได้ทำการเพิ่มข้อมูลในส่วนของเงื่อนไขก่อนหน้า (Pre-condition) และเงื่อนไขตามหลัง (Post-condition) เข้ามา ดังแสดงในตารางที่ 4

ตารางที่ 4 คำอธิบายยูสเคสที่ใช้ในงานวิจัยนี้

usecaseID	เขียนโดยเริ่มด้วยตัวอักษร “UC” ตามด้วยตัวเลข โดยเป็นตัวเลขไม่ซ้ำกัน ในแต่ละระบบ ตัวอย่าง “UC1”, “UC2”
Pre-condition	เป็นเงื่อนไขก่อนหน้า โดยใช้เป็นจุดตรวจสอบ เขียนอยู่ในรูปแบบ คอมโพเนนท์ประเภทหน้าต่างหรือไดอะลอค = true หรือ false เช่น “Result page” = true หมายถึง หน้าต่าง Result page ปรากฏขึ้นแล้ว โดยทำการตรวจสอบเมื่อเริ่มการทำงานยูสเคส
Post-condition	เช่นเดียวกับ Pre-condition คอมโพเนนท์สามารถเป็นคอมโพเนนท์ประเภทอื่นๆ ได้ เช่น คอมโพเนนท์ A = “1” หมายถึง คอมโพเนนท์ A มีการกำหนดค่าเท่ากับ 1 โดยจะทำการตรวจสอบหลังการทำงานของยูสเคสเสร็จสิ้นแล้ว
MainFlow	การทำงานหลักของยูสเคสอธิบายเป็นข้อๆ โดยใช้ตัวเลข ภายในเครื่องหมาย “[“ และ “]” ตามด้วยการกระทำต่อคอมโพเนนท์ และชื่อคอมโพเนนท์ภายในเครื่องหมาย “ และ ” ซึ่งชื่อคอมโพเนนท์จะต้องเหมือนกับที่ปรากฏอยู่ในแผนภาพดับเบิลยูเอ็นดี เช่น [1] lclick “OK” [2] display “Welcome” เป็นต้น โดยที่ “OK” คือชื่อของปุ่ม “Welcome” คือชื่อของหน้าต่างในแผนภาพดับเบิลยูเอ็นดี หรือการสิ้นสุดการทำงานจะแทนด้วย End และการข้ามไปยังการทำงานขั้นตอนอื่นๆ ใช้ เงื่อนไข Goto ตามด้วยหมายเลขประขั้นตอน ไม่ว่าจะเป็นการทำงานหลักหรือการทำงานอื่นๆ โดยเงื่อนไขจะต้องมีค่าเป็นจริงหรือเท็จเท่านั้น รูปแบบ คือ “ชื่อคอมโพเนนท์” = ค่ากำหนด
AlternateFlow	การทำงานอื่นๆ นอกเหนือจากการทำงานหลัก โดยแบ่งออกเป็น 2 กรณี คือ 1. การทำงานซึ่งแยกออกมาจากการทำงานหลัก โดยเกิดขึ้นเมื่ออยู่ในการทำงานในขั้นการทำงานหลัก แต่อาจเกิดกรณีพิเศษซึ่งทำให้ต้องทำงานแบบไม่ปกติ 2. การทำงานที่สามารถเกิดได้ทุกขณะ เช่น การยกเลิกการทำงาน ในกรณีเช่นนี้จะทดสอบโดยจะทำงานเมื่อเริ่มการทำงานในยูสเคสทันที

ลักษณะการทำงานของทั้ง 2 กรณีสามารถแสดงเป็นเส้นทางการทำงานได้ โดยแสดงในรูปที่ 11

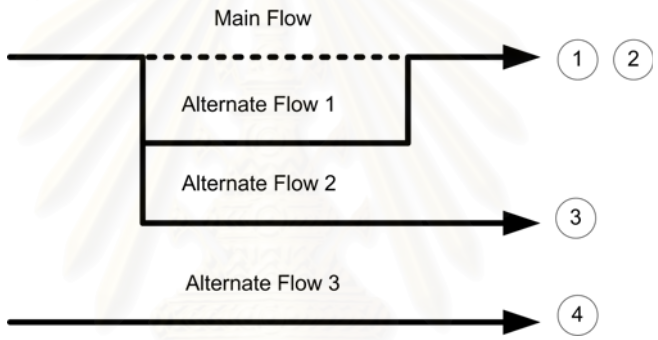
เส้นทางที่ 1 คือการทำงานหลัก

เส้นทางที่ 2 คือการทำงานอื่นๆ แบบกรณีที่ 1 แล้วกลับไปทำงานในการทำงานหลัก ซึ่งตอนท้ายจะจบการทำงานด้วยขั้นตอนของการทำงานหลัก

เส้นทางที่ 3 เป็นการทำงานอื่นๆ แบบกรณีที่ 1 แต่ไม่กลับไปทำงานในการทำงานหลัก เช่น การจัดการความผิดพลาด (Error handling)

เส้นทางที่ 4 เป็นการทำงานอื่นๆ ซึ่งสามารถเกิดในทุกเวลา ดังนั้นจึงเริ่มทำตั้งแต่เริ่มการทำงาน

โดยการทำงานใด ไม่จบการทำงานในขั้นตอนภายในการทำงานหลัก จะทำให้ยูสเคสไม่สามารถตรวจสอบด้วยเงื่อนไขตามหลังของยูสเคสนั้นได้ จะถือว่าสิ้นสุดการทดสอบในขั้นตอนนี้



รูปที่ 11 ลักษณะต่างๆ ของการทำงานของยูสเคส

ในกรณีที่ยูสเคสมีความสัมพันธ์กับยูสเคสอื่นๆ จะต้องเขียนจุดต่อความสัมพันธ์ในส่วนของการทำงานหลักและการทำงานอื่นๆ โดยรูปแบบการเขียน คือ

1. ความสัมพันธ์แบบสืบทอด ในงานวิจัยนี้หากมีความสัมพันธ์แบบสืบทอดให้ทำการปรับยูสเคสใหม่โดยตัดยูสเคสแม่ออก จากนั้นทำการสร้างความสัมพันธ์จากยูสเคสที่มีความสัมพันธ์กับยูสเคสแม่ไปยังยูสเคสลูกแบบอินคลูด [13] โดยยูสเคสลูกให้เขียนคำอธิบายยูสเคสแยกจากกัน
2. ความสัมพันธ์แบบอินคลูดจะปรากฏอยู่ในขั้นตอนการทำงานของยูสเคส โดยเขียนชื่อลักษณะความสัมพันธ์ตามด้วยเครื่องหมาย “=” และ usecaseID ภายในเครื่องหมาย “{“ และ “}” เขียนได้ดังนี้ {inc = “usecaseID”} หมายถึงยูสเคสมีการเรียกใช้ยูสเคส UC2
3. ความสัมพันธ์แบบเอกซ์เทนด์จะปรากฏอยู่ในขั้นตอนการทำงานของยูสเคส โดยเขียนชื่อความลักษณะความสัมพันธ์ตามด้วยเครื่องหมาย “=” และ usecaseID ตาม

ด้วยเครื่องหมาย “,” และเงื่อนไข โดยเงื่อนไขจะเขียนเป็นรูปแบบ “คอมโพเนนท์” = “ค่ากำหนด” โดยที่ค่าของเงื่อนไขเป็นจริงหรือเท็จเท่านั้น ภายในเครื่องหมาย “{“ และ “}” เขียนได้ดังนี้ {ext = “usecaseID”, “Check” = “123”} หมายถึง ยูสเคส UC3 ถูกเรียกใช้เมื่อ คอมโพเนนท์ Check มีค่าเท่ากับ 123

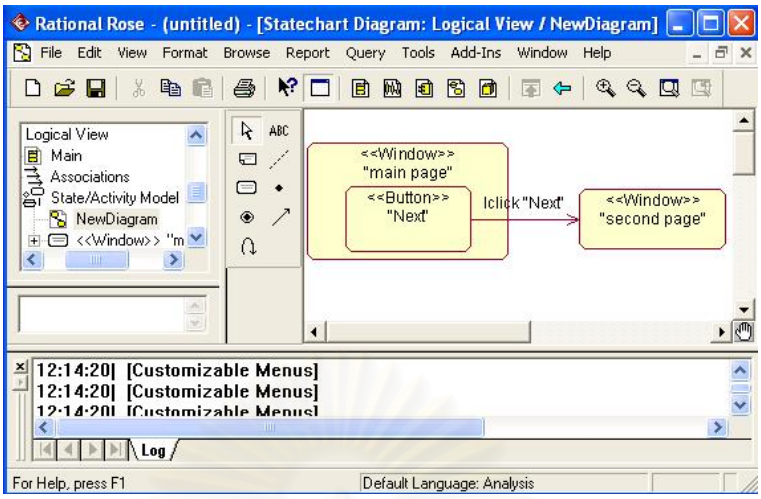
การเขียนการกระทำต่อคอมโพเนนท์ต่างๆ ในส่วนของคำอธิบายยูสเคสได้กำหนดเป็นคำสงวน ไว้ดังตารางที่ 5

ตารางที่ 5 คำสงวนที่ใช้ในการเขียนคำอธิบายยูสเคส

คำสงวน	**หมายถึง**	**ตัวอย่างคอมโพเนนท์**
lclick	การกดเม้าส์ปุ่มซ้าย 1 ครั้งแล้วปล่อย	Button, Menu, MenuItem, Tab
rclick	การกดเม้าส์ปุ่มขวา 1 ครั้งแล้วปล่อย	คอมโพเนนท์ใดๆ มักเป็นการแสดง tool tip
dbclick	การกดเม้าส์ปุ่มซ้าย 2 ครั้งติดกันแล้วปล่อย	Text, TextArea
type	การพิมพ์ข้อมูลลงในคอมโพเนนท์	Text, TextArea
choose	การเลือกข้อมูลของคอมโพเนนท์ ที่มีข้อมูลกำหนดไว้แล้ว เช่น เมนู ลิสบอกซ์ เป็นต้น	Combo, Menu, MenuItem
display	การแสดงหน้าต่างคอมโพเนนท์ประเภทหน้าต่าง หรือกล่องข้อความ	Window, Dialog
close	การปิดหน้าต่างคอมโพเนนท์ประเภทหน้าต่าง หรือกล่องข้อความ	Window, Dialog

3.2.3 การวาดแผนภาพดับเบิลยูเอ็นดี

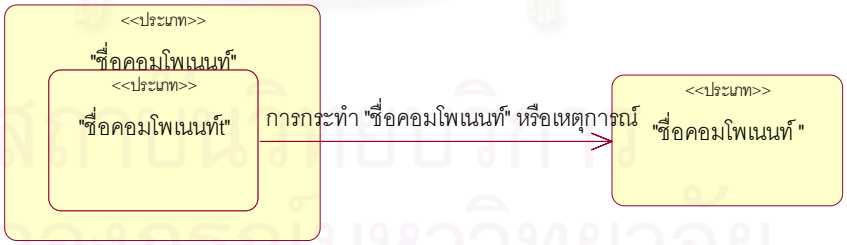
แผนภาพดับเบิลยูเอ็นดีเป็นแผนภาพแสดงลักษณะ และการส่งผ่านหน้าจอของส่วนต่อประสานกราฟิกกับผู้ใช้ของระบบที่จะถูกสร้างขึ้น รวมทั้งบอกถึงประเภทของคอมโพเนนท์และโครงสร้างลำดับของคอมโพเนนท์ว่าบรรจุอยู่ในคอมโพเนนท์ใด แสดงตัวอย่างแผนภาพดับยูเอ็นดี เมื่อวาดลงในเครื่องมือเรชันเนล โรส ดังรูปที่ 12



รูปที่ 12 แผนภาพดับเบิลยูเอ็นดี

แสดงลักษณะการเขียนแผนภาพดับเบิลยูเอ็นดี ดังรูปที่ 13 ประกอบด้วย

1. สี่เหลี่ยนมุมมน คือแทนด้วยคอมโพเนนท์ ประกอบด้วย ประเภทของคอมโพเนนท์ภายใน เครื่องหมาย “<<” และ “>>” และชื่อคอมโพเนนท์ภายในเครื่องหมาย “ และ “ โดยคอมโพเนนท์ประเภทหน้าต่าง หรือเมนูสามารถบรรจุคอมโพเนนท์ประเภทอื่นไว้ได้ ให้วาดซ้อนไว้ภายใน
2. เส้นเชื่อมระหว่างคอมโพเนนท์ แสดงเส้นทางการเชื่อมต่อระหว่างหน้าจอของส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งเชื่อมระหว่างคอมโพเนนท์ที่เกิดการกระทำหรือเหตุการณ์กับคอมโพเนนท์ประเภทหน้าต่าง หรือกล่องข้อความ

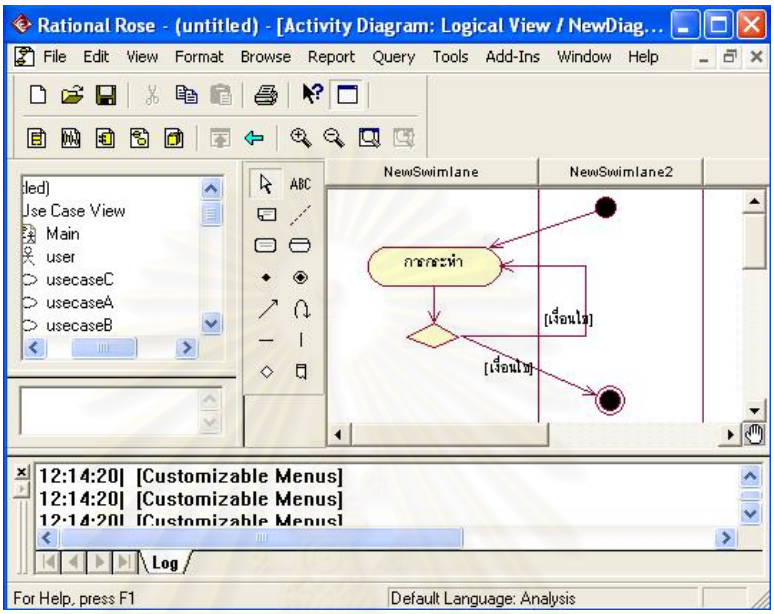


รูปที่ 13 ตัวอย่างลักษณะการเขียนแผนภาพดับเบิลยูเอ็นดี

3.2.4 การวาดแผนภาพกิจกรรม

แผนภาพกิจกรรม แสดงให้เห็นถึงการทำงานของยูสเคสโดยแบ่งการทำงานของผู้ใช้ออกจากระบบได้อย่างชัดเจน โดยจะสามารถแยกการทำงานของผู้ใช้ออกจากการทำงานของระบบได้ และสนใจการกระทำในส่วนของผู้ใช้เท่านั้น การวาดแผนภาพกิจกรรมจะมี 1 แผนภาพต่อ 1

ยูสเคส โดยอ้างอิงจากคำอธิบายยูสเคสเป็นหลัก เพื่อแสดงแต่ละขั้นตอนในการทำงานหลัก รวมทั้งการทำงานอื่นๆ ของคำอธิบายยูสเคส ด้วย 1 โนดของแผนภาพกิจกรรม แสดงตัวอย่างดังรูปที่ 14



รูปที่ 14 ตัวอย่างการเขียนแผนภาพกิจกรรมในเครื่องมือเรชันเนล โรส

การเขียนแผนภาพกิจกรรม มีลักษณะดังนี้

1. สวิมเลน แทนด้วยผู้ใช้ระบบซึ่งเป็นผู้ใช้เดียวกับที่อยู่ในแผนภาพยูสเคส และระบบซึ่งทำงานร่วมกันในแต่ละยูสเคส
2. โนดเริ่มต้น เขียนด้วยจุดทึบ บอกจุดเริ่มต้นของการทำงาน และมีเพียง 1 โนดต่อ 1 แผนภาพ และมีเส้นเชื่อมออกจากโนดเริ่มต้น 1 เส้น
3. รูปทรงแคปซูลหรือโนด เป็นการกระทำของการทำงาน ภายในคำอธิบายยูสเคส โดย 1 โนด แทนด้วย 1 ข้อของการทำงานในยูสเคส โดยการเขียนจะมีวิธีเขียนเหมือนกับใน คำอธิบายยูสเคส
4. เส้นเชื่อม เป็นส่วนเชื่อมต่อระหว่างโนดต่างๆ เพื่อบอกเส้นการทำงาน อาจมีข้อความกำกับเงื่อนไขบนเส้นได้
5. โนดตัดสินใจ บอกจุดที่แยกการทำงานออกเป็น 2 ทาง ซึ่งการเลือกเส้นทางการทำงานจะใช้เงื่อนไขระบุ โดยเงื่อนไขระบุอยู่ภายในเครื่องหมาย “[“ และ “]” รูปแบบเงื่อนไข คือ “คอมโพเนนท์” = ค่ากำหนด ซึ่งเงื่อนมีค่าเป็นจริงหรือเท็จเท่านั้น

6. เส้นรวมหรือแยกเส้นเชื่อม เป็นเส้นแยกการทำงานออกเป็นหลายทาง หรือรวมการทำงานเป็น เส้นทางเดียว ใช้ในกรณีเริ่มหรือจบการทำงานแบบขนาน
7. โนดสิ้นสุด เขียนด้วยจุดทึบภายในวงกลมโปร่ง แสดงการสิ้นสุดของการทำงาน อาจมีหลายโนดได้ในแผนภาพเดียวกัน โดยหากเป็นโนดสิ้นสุดของการทำงานหลักจะเขียนคำว่า success กำกับ และหากเป็นโนดสิ้นสุดของการทำงานอื่นๆ ที่ไม่กลับเข้ามาในการทำงานหลักให้เขียนกำกับว่า alternative

3.3 วิธีการสร้างกรณีทดสอบจากยูเอ็มแอล

เมื่อเขียนแผนภาพต่างๆ และสร้างเป็นแฟ้มเอกซ์เอ็มแอลเรียบร้อยแล้ว ขั้นตอนต่อไป จะนำข้อมูลดังกล่าวมาสร้างเป็นตัวดำเนินการและแผนการทดสอบ โดยตัวดำเนินการจะแทนการทำงานของยูสเคส 1 ยูสเคส แผนการทดสอบจะแทนลำดับของยูสเคสที่ทำงานร่วมกันในแต่ละงาน

3.3.1 ตัวดำเนินการ

ตัวดำเนินการแสดงถึงลำดับของการกระทำต่อคอมโพเนนท์ต่างๆ มาเรียงลำดับกันตามแผนภาพกิจกรรม โดย 1 โนดของแผนภาพกิจกรรมจะแทนด้วย 1 อีลีเมนต์ (Element) <seq></seq> ซึ่งจะแบ่งตามการทำงานของยูสเคสคือการทำงานหลักและการทำงานอื่นๆ โดยแสดงเป็นรูปแบบของการกระทำต่อคอมโพเนนท์ แต่ละตัวดำเนินการประกอบด้วย ชื่อตัวดำเนินการ เงื่อนไขก่อนหน้า ลำดับการกระทำ เงื่อนไขตามหลัง

```
<?xml version = '1.0' encoding = 'ISO-8859-1'?>
<Operator_root>
<Operator name="Pin validate">
  <Precondition value='"Pin validate" = true'/>
    <Flow id="1">
      <Seq id="1">
       <Window name="Pin Validate">
       <Component value="Text"/>
        <Name value ="Pin"/>
        <Action value ="Type"/>
       <Value value="UNKNOWN"/>
       </Window>
     </Seq>
     <Seq id = "2">
```

```
<Window name="Pin Validate">
<Component value=”Button”/>
<Name value=”Submit_Pin Validate”/>
<Action value=”Lclick”/>
<Value value=”Unknown”/>
</Window>
</Seq>
<Seq id = “success” ></Seq>
</Flow>
<Postcondition value='"Main Menu Window”=true’/>
</Operator>
</Operator_root>
```

รายละเอียดของแต่ละอีลีเมนต์ แสดงดังนี้

1. <Operation_root></Operation_root> มี 1 อีลีเมนต์ คือ อีลีเมนต์เริ่มต้นของแฟ้มตัวดำเนินการ
2. <Operator></Operator> มีได้หลายอีลีเมนต์ แทน 1 ตัวดำเนินการหรือ 1 ยูสเคส มีแอตทริบิวท์ name คือชื่อของตัวดำเนินการซึ่งเป็นชื่อเดียวกันกับชื่อของยูสเคส
3. <Precondition/> เงื่อนไขก่อนหน้ามาจากเงื่อนไขก่อนหน้าในคำอธิบายยูสเคส เพื่อใช้เป็นจุดตรวจสอบของการทดสอบโดยจะตรวจสอบก่อน
3. <Postcondition/> เงื่อนไขตามหลังมาจากเงื่อนไขตามหลังในคำอธิบายยูสเคส เพื่อใช้เป็นจุดตรวจสอบของการทดสอบโดยจะตรวจสอบหลังการทำการทดสอบตัวดำเนินการเสร็จสิ้น โดยจะตรวจสอบเฉพาะชุดของลำดับการทำงานที่สิ้นสุดการทำงานในขั้นตอนการทำงานหลัก
4. <Flow></Flow> แสดงชุดของลำดับการทำงานมีจำนวนเท่ากับจำนวนของการทำงานในคำอธิบายยูสเคส ซึ่งก็คือการทำงานหลักและการทำงานอื่นๆ มีแอตทริบิวท์ไอดี (id) แสดงลำดับที่ของชุด
5. <seq></seq> แทนด้วย 1 การกระทำต่อคอมโพเนนท์ หรือเทียบได้กับ 1 ขั้นตอนการทำงานที่ปรากฏในคำอธิบายยูสเคส ซึ่งมีแอตทริบิวท์ คือ id เป็นลำดับการกระทำที่เกิดขึ้นในแต่ละ อีลีเมนต์ <Flow/>

6. <Window/> แสดงหน้าต่างที่คอมโพเนนท์นั้นถูกบรรจุอยู่ มีเพื่อลดความซ้ำซ้อนของคอมโพเนนท์ที่มีชื่อเหมือนกัน แต่อยู่ต่างหน้าต่างกัน โดยอาจเป็นชื่อกล่องข้อความ
7. <Component/> แสดงประเภทของคอมโพเนนท์โดยได้จากแผนภาพดับเบิลยูเอ็นดี
8. <Name/> แสดงชื่อของคอมโพเนนท์ที่ปรากฏในแผนภาพดับเบิลยูเอ็นดี
9. <Action/> การกระทำต่อคอมโพเนนท์ เช่นการคลิก การพิมพ์
10. <Value/> สำหรับการกระทำที่ต้องการค่าที่จะกำหนดให้แก่คอมโพเนนท์ เช่นการพิมพ์จะต้องกำหนดค่าไว้ โดยจะให้ผู้ออกแบบการทดสอบเป็นผู้กำหนดค่าภายหลัง สำหรับค่าเริ่มต้นจะกำหนดไว้เป็น Unknown

ตารางที่ 6 ค่าของแอตทริบิวท์ id ในอีลีเมนต์ <seq/>

ค่าของแอตทริบิวท์	ความหมาย
ตัวเลข เช่น 1, 2, 3,...	เป็นลำดับที่ 1 2 หรือ 3 ใน ชุดการทำงานนั้นๆ
Success หรือ Alternative	เป็นการสิ้นสุดการกระทำของชุดการทำงานนั้นๆ หากเป็น success หมายถึงจบการทำงานในการทำงานหลัก alternative เป็นการจบการทำงานในการทำงานอื่นๆ
Goto 2.1	ข้ามไปทำงานยังชุดการทำงานที่ 2 การกระทำที่ 1 ภายในตัวดำเนินการเดียวกัน
Goto ชื่อตัวดำเนินการ 2.1	ข้ามไปทำงานยังชุดการทำงานที่ 2 การกระทำที่ 1 ที่ตัวดำเนินการชื่อนั้นๆ

3.3.2 แผนการทดสอบ

แผนการทดสอบแสดงลำดับของยูสเคสซึ่งทำงานตามลำดับในแต่ละภารกิจ โดยเริ่มพิจารณาจากยูสเคสที่มีความสัมพันธ์กับผู้ใช้ จากนั้นจะดูที่การทำงานหลักและการทำงานอื่นๆ หากมีความสัมพันธ์กับยูสเคสอื่น จะมีการเพิ่มตัวดำเนินการเป็นยูสเคสที่ถูกเรียกใช้เพิ่มลงในแผนการทดสอบ เช่น ยูสเคส Inquiry มีความสัมพันธ์กับยูสเคส Pin Validate แบบอินคลูด โดยในคำอธิบายยูสเคสระบุให้ทำการเรียกใช้ยูสเคสในขั้นตอนแรก ดังนั้นในแผนการทดสอบตัวดำเนินการ Pin Validate จึงมีลำดับก่อนตัวดำเนินการ Inquiry แสดงดังต่อไปนี้

```
<?xml version = '1.0' encoding = 'ISO-8859-1'?>
<Plan_root>
<Plan name="1">
```

```
<Inc_TOperator value=”Pin Validate”/>
<TOperator value=”Inquiry”/>
</Plan>
<Plan name="2">
<Inc_TOperator value=”Pin validate”/>
<TOperator value=”Withdraw”/>
<Inc_TOperator value=”Print slip”/>
</Plan>
<Plan name="3">
<Inc_TOperator value=”Pin validate”/>
<TOperator value=”Deposit”/>
<Inc_TOperator value=”Print slip”/>
</Plan>
</Plan_root>
```

รายละเอียดของแต่ละอีลีเมนต์ มีดังนี้

1. <Plan_root></Plan_root> มี 1 อีลีเมนต์ เป็นโนดเริ่มต้นของแผนการทดสอบ
2. <Plan></Plan> ชุดของตัวดำเนินการ 1 ชุด มีแอตทริบิวท์ name บอกลำดับที่ของตัวดำเนินการ
3. <TOpertor/> อีลีเมนต์แสดงชื่อของยูสเคสที่ทำงานในภารกิจนั้นๆ ถ้าเป็นยูสเคสที่มีการอินคลูดใช้อีลีเมนต์ <Inc_TOperator/> และยูสเคสที่มีการเอกซ์เทนด์ <Ext_TOperator/>

3.3.3 กรณีทดสอบและสคริปต์ทดสอบ

กรณีทดสอบ คือการนำเอาตัวดำเนินการมาแทนที่ในแผนการทดสอบ นั่นคือการนำตัวดำเนินการมาเรียงต่อกันอย่างมีลำดับเพื่อให้บรรลุภารกิจที่กำหนด เนื่องจากตัวดำเนินการสามารถมีการทำงานได้หลายเส้นทาง ดังนั้นแผนการทดสอบ 1 แผน สามารถสร้างเป็นกรณีทดสอบได้ตั้งแต่ 1 กรณีทดสอบขึ้นไป ซึ่งจำนวนของกรณีทดสอบที่ได้นั้นขึ้นอยู่กับการทำงานในคำอธิบายยูสเคส โดยที่กรณีทดสอบแรกจะเป็นกรณีทดสอบของการทำงานหลักของทุกตัวดำเนินการในแผนการทดสอบนั้นๆ ยกตัวอย่างเช่น

กำหนดให้ แผนการทดสอบ A = {a, b, c}
ตัวดำเนินการ a = {a1, a2}

ตัวดำเนินการ b = {b1, b2, b3}

ตัวดำเนินการ c = {c1}

ตัวแปร a1, b1, c1 คือ อีลีเมนต์ <Flow/> ของการทำงานหลักของตัวดำเนินการแต่ละตัว ตัวแปร a2, b2, b3 คือ อีลีเมนต์ <Flow/> ของการทำงานอื่นๆ อาจเป็นการทำงานที่สำเร็จหรือไม่สำเร็จก็ได้

ดังนั้น กรณีทดสอบที่สามารถเป็นไปได้จากแผนการทดสอบ A คือ

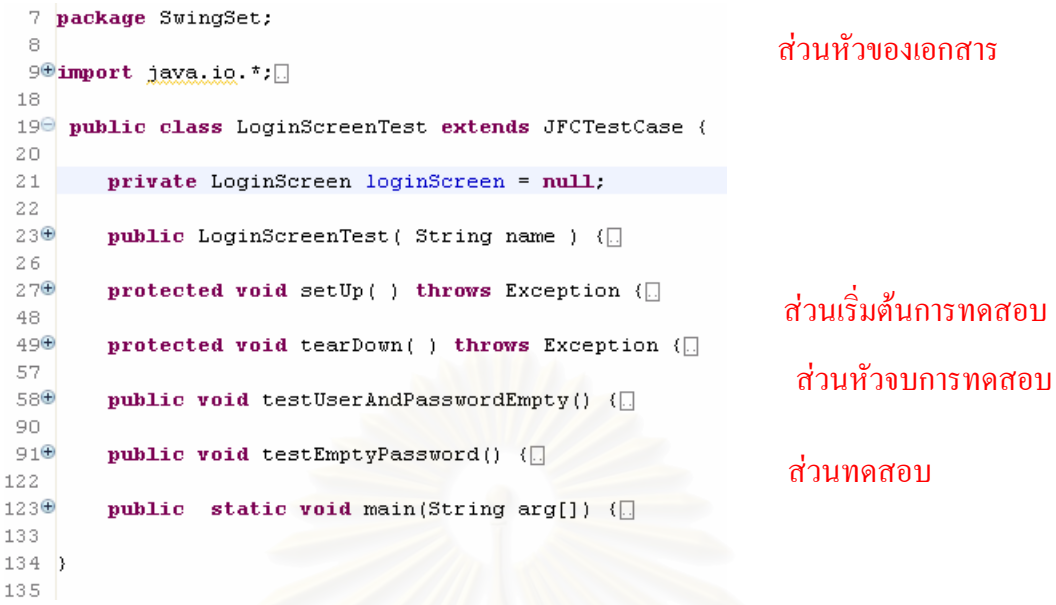
{a1, b1, c1}, {a1, b2, c1}, {a1, b3, c1}, {a1, b2}, {a1, b3},

{a2, b1, c1}, {a2, b2, c1}, {a2, b3, c1}, {a2, b2}, {a2, b3}, {a2}

จะเห็นว่ามีบางกรณีทดสอบมีเพียง 1 หรือ 2 อีลีเมนต์ สาเหตุเนื่องมาจากอีลีเมนต์ของตัวดำเนินการนั้นเป็นการทำงานอื่นๆ ของยูสเคสซึ่งไม่สามารถทำงานได้สำเร็จ ทำให้การทำงานจบลงและไม่สามารถทำงานในขั้นตอนต่อไปได้ ยกตัวอย่างเช่น กรณีทดสอบ {a2, b3} ไม่มีการทำงานที่ตัวดำเนินการ c เนื่องจากที่ขั้นตอนการทำงานในตัวดำเนินการ b เป็นการทำงานอื่นๆ ซึ่งไม่สามารถทำงานได้สำเร็จถือว่าสิ้นสุดการทำงาน ฉะนั้นจึงไม่สามารถไปทำงานในตัวดำเนินการ c ต่อได้

สคริปต์ทดสอบ คือกรณีทดสอบที่สามารถนำไปใช้ทดสอบกับระบบจริงได้ อยู่ในรูปของแฟ้มภาษาจาวาตามกรอบการทำงานเจเอฟซียูนิต ซึ่งเรียกใช้งานโดยเครื่องมือสร้างและทดสอบกรณีทดสอบที่สร้างขึ้น ในสคริปต์ทดสอบบรรจุกรณีทดสอบทุกกรณีทดสอบที่แปลงมาจากแฟ้มเอกซ์เอ็มแอลของกรณีทดสอบ และเมื่อทำการทดสอบจะทำการทดสอบเรียงลำดับจากกรณีทดสอบแรกไปจนจบทุกกรณีทดสอบ หากไม่เกิดความผิดพลาดเกิดขึ้นกับโปรแกรมที่ทดสอบนอกเหนือจากความผิดพลาดจากส่วนต่อประสานกราฟิกกับผู้ใช้ สคริปต์ทดสอบมีรูปแบบดังแสดงดังรูปที่ 15

ในงานวิจัยนี้จะเขียนสคริปต์ทดสอบตามกรอบการทำงานเจเอฟซียูนิต เพื่อให้สามารถทำการทดสอบกับระบบที่พัฒนาขึ้นด้วยภาษาจาวาและใช้คอมโพเนนท์สวิง เนื่องจากปัจจุบันนี้ระบบงานที่พัฒนาขึ้นโดยใช้ภาษาจาวามีแนวโน้มการใช้งานคอมโพเนนท์สวิงมากขึ้น เพราะหน้าจอที่ปรากฏและกระบวนการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้ประเภทนี้ ไม่ขึ้นกับระบบปฏิบัติการ ฉะนั้น ระบบจะมีหน้าจอที่ปรากฏและการทำงานเหมือนกันไม่ว่าจะทำงานบนระบบปฏิบัติการใดๆ




```
7  package SwingSet;
8
9  import java.io.*;
18
19 public class LoginScreenTest extends JFCTestCase {
20
21     private LoginScreen loginScreen = null;
22
23     public LoginScreenTest( String name ) {
26
27     protected void setUp( ) throws Exception {
48
49     protected void tearDown( ) throws Exception {
57
58     public void testUserAndPasswordEmpty() {
90
91     public void testEmptyPassword() {
122
123    public  static void main(String arg[]) {
133
134 }
135
```

รูปที่ 15 ตัวอย่างสคริปต์ทดสอบ

รูปแบบของสคริปต์ทดสอบประกอบด้วย 4 ส่วน คือ ส่วนหัวของแฟ้ม ส่วนกำหนดค่าเริ่มต้นการทดสอบ ส่วนจบการทดสอบ และส่วนทดสอบ โดยการทดสอบแต่ละกรณีทดสอบจะเริ่มต้นด้วยส่วนกำหนดค่าเริ่มต้นการทดสอบ จากนั้นจึงทำงานในส่วนทดสอบ และจบด้วยส่วนจบการทดสอบเสมอ แสดงรายละเอียด ดังนี้

1. ส่วนหัวของแฟ้ม เป็นส่วนประกาศการนำเข้า การประกาศคลาส และส่วนอื่นๆ ที่จำเป็นในการทำให้สคริปต์ทดสอบสามารถดำเนินงานได้ ซึ่งแต่ละชุดทดสอบจะเหมือนกัน จะต่างกันเพียงชื่อคลาสเท่านั้น
2. ส่วนกำหนดค่าเริ่มต้นการทดสอบ จะใช้เมธอด setUp() เป็นเมธอดเริ่มต้นเสมอ ใช้เพื่อกำหนดค่าเริ่มต้นให้กับการทดสอบให้ทุกกรณีทดสอบมีสภาพแวดล้อมเดียวกัน เช่น เริ่มเปิดโปรแกรมที่จะทดสอบใหม่สำหรับทุกกรณีทดสอบ
3. ส่วนจบการทดสอบ จะใช้เมธอด tearDown() เป็นเมธอดที่จะทำงานเมื่อจบการทดสอบแต่ละกรณีทดสอบ เพื่อคืนทรัพยากรให้แก่ระบบและพร้อมทำงานทดสอบในกรณีทดสอบถัดไป
4. ส่วนทดสอบ คือกรณีทดสอบแต่ละกรณีทดสอบ ซึ่งจะเขียนในรูปแบบของภาษาจาวาภายใต้กรอบการทำงานเจเอฟซียูนิต โดยจะมีการประกาศตัวแปรต่างๆ สำหรับแต่ละคอมโพเนนท์ การค้นหาคอมโพเนนท์ การกระทำกับคอมโพเนนท์ และจุดตรวจสอบ แสดงตัวอย่างการเขียนสคริปต์ทดสอบของคอมโพเนนท์แต่ละประเภทดังตารางที่ 7

ตารางที่ 7 ตัวอย่างการเขียนสคริปต์ของคอมโพเนนท์แต่ละประเภท

ประเภท	สคริปต์

ปุ่ม (Button)	Finder f = new AbstractButtonFinder("One"); //หาปุ่มชื่อ One
	JButton button1 = (JButton) f.find(panel, 0); //ประกาศตัวแปรปุ่มที่ได้จากการค้นหา โดย panel แทนด้วยคอมโพเนนท์ที่บรรจุปุ่ม

ปุ่มเรดิโอ (Radio box)	Finder f = new AbstractButtonFinder("One"); //ค้นหาปุ่มเรดิโอชื่อ One
	JRadioButton button1 = (JRadioButton) f.find(panel, 0); //ประกาศตัวแปรปุ่มเรดิโอที่ได้จากการค้นหา โดย panel แทนด้วยคอมโพเนนท์ที่บรรจุปุ่มเรดิโอ

ปุ่มเช็ค (Check box)	Finder f = new ComponentFinder(JCheckBox.class); //ค้นหาปุ่มเช็ค
	JCheckBox box1 = (JCheckBox) f.find(panel, 0); //ประกาศตัวแปรปุ่มเช็คที่ได้จากการค้นหาด้วยลำดับ โดย panel แทนด้วยคอมโพเนนท์ที่บรรจุปุ่มเช็ค

หน้าต่าง (Window)	FrameFinder f = new FrameFinder(getName());//ค้นหาหน้าต่าง
	JFrame frame = (JFrame) f.find();//ประกาศตัวแปรหน้าต่างที่ได้จากการค้นหา

แถบแท็ป (Tab)	JTabbedPane tabbedPane = getReferenceToTabPane();//ค้นหาแถบแท็ป
	clickAndWait(new JTabbedPaneMouseEventData(this, tabbedPane, -1, TABTITLE_BUTTONS, 1), true);//ทำการกดที่ปุ่มแท็ป

เมนู (Menu)	JMenu file = (JMenu) new JMenuItemFinder("File").find(frame, 0); //ค้นหาเมนูชื่อ File โดย frame เป็นหน้าต่างที่เมนูบรรจุอยู่
	PathData path = new PathData(new String[] {"File", "Measure", "H"}); //กำหนดทางเดินของเมนูจากต้นเมนูถึงเมนูที่ต้องการ
	JMenuBar bar = (JMenuBar) path.getRoot(file); //ประกาศเมนูที่ได้ตามทางเดินที่กำหนดไว้
	getHelper().enterClickAndLeave(new JMenuMouseEventData(this, bar, path.getIndexes(bar), 1, 0, false, 0)); //ทำการคลิกที่เมนูที่กำหนด

3.3.4 จุดตรวจสอบ และความผิดพลาดที่สามารถตรวจสอบได้

จุดตรวจสอบจะเขียนในสคริปต์ส่วนทดสอบ โดยมีรูปแบบการเขียนเหมือนกับเจยูนิต โดยจุดตรวจสอบแบ่งออกเป็น 2 ประเภท ดังนี้

1. จุดตรวจสอบแบบอัตโนมัติ โดยเครื่องมือจะสร้างขึ้นแบบอัตโนมัติจากกรณีทดสอบ ซึ่งจะใช้ตรวจสอบข้อมูลเบื้องต้นที่จำเป็น เช่น หน้าจอที่ต้องการปรากฏขึ้นหรือไม่ ปุ่มที่ต้องการปรากฏขึ้นหรือไม่ เป็นต้น
2. จุดตรวจสอบที่ผู้ออกแบบการทดสอบเป็นผู้สร้าง มีจุดประสงค์เพื่อใช้ทดสอบค่าหรือข้อความที่เป็นผลคาดหวังจากการทดสอบ เช่น กล่องข้อความแสดงค่าที่ถูกต้องหรือไม่ หรือ เมื่อกดปุ่มหน้าจอที่ต้องการปรากฏขึ้นหรือไม่ เป็นต้น

ส่วนจุดตรวจสอบนี้จะใช้คำสั่งของเจยูนิตช่วยในการตรวจสอบโปรแกรม โดยที่คำสั่งตรวจสอบนี้ประกอบไปด้วย

- คำสั่ง assertEquals() ใช้สำหรับตรวจสอบค่าสองค่าให้เท่ากัน แต่ถ้าไม่เท่าจะแสดงข้อความผิดพลาดออกมา
- คำสั่ง assertTrue() ใช้สำหรับตรวจสอบค่าให้มีค่าเป็นจริง แต่ถ้ามีค่าเป็นเท็จจะแสดงข้อความผิดพลาดออกมา
- คำสั่ง assertFalse() ใช้สำหรับตรวจสอบค่าที่ต้องมีค่าเป็นเท็จ แต่ถ้าเป็นจริงจะแสดงข้อความผิดพลาดออกมา
- คำสั่ง assertNotNull() ใช้สำหรับตรวจสอบที่ไม่เป็น Null หรือว่าต้องมีค่า แต่ถ้าไม่มีค่าหรือว่าเป็น Null จะแสดงข้อความผิดพลาดออกมา
- คำสั่ง assertNull() ใช้สำหรับตรวจสอบค่าให้ต้องไม่มีค่าหรือมีค่าเป็น Null แต่ถ้ามีค่าหรือไม่เป็น Null จะแสดงข้อความผิดพลาดออกมา
- คำสั่ง assertSame() ใช้สำหรับตรวจสอบค่าสองค่าต้องเป็นค่าเดียวกัน แต่ถ้าไม่ใช่ค่าเดียวกันจะแสดงข้อความผิดพลาดออกมา
- คำสั่ง assertNotSame() ใช้สำหรับตรวจสอบค่าสองค่าต้องไม่เหมือนกัน แต่ถ้าเป็นค่าเดียวกันจะแสดงข้อความผิดพลาดออกมา

เครื่องมือจะทำการตรวจสอบข้อผิดพลาด ในส่วนของจุดตรวจสอบที่ได้กำหนดไว้ โดยจะเกิดความผิดพลาดเมื่อ ได้ 2 กรณี คือ

1. ค่าตรรกะของจุดตรวจสอบเป็นเท็จ โดยเครื่องมือจะแสดงข้อความผิดพลาด จัดเก็บข้อความผิดพลาดไว้ในฐานข้อมูล พร้อมทั้งแสดงผลในรายงานผลการทดสอบด้วย เช่น assertNotNull("ไม่พบคอมโพเนนท์ "+"Exit button", exitButton); หากตัวแปร exitButton = null จะเกิดความผิดพลาดขึ้น พร้อมทั้งแสดงข้อความ “ไม่พบ

คอมโพเนนท์ Exit button” ผู้ออกแบบการทดสอบจึงสามารถทราบได้ว่า มีความผิดพลาดที่ปุ่ม Exit button

2. ความผิดพลาดของตัวโปรแกรมทดสอบเอง ไม่ได้เกิดจากการทำงานผิดพลาดของส่วนต่อประสานกราฟิกกับผู้ใช้ ความผิดพลาดในกรณีนี้เกิดขึ้นนอกเหนือความสามารถของเครื่องมือที่จะตรวจสอบได้ว่าเกิดความผิดพลาดกับส่วนใดของโปรแกรม เช่น ไม่สามารถติดต่อฐานข้อมูลได้ เกิดความผิดพลาดของช่วงข้อมูล เป็นต้น

3.4 รายละเอียดของเครื่องมือสร้างและทดสอบกรณีทดสอบ

เครื่องมือสร้างและทดสอบกรณีทดสอบ สร้างขึ้นด้วยภาษาจาวา และทำงานภายใต้กรอบการทำงานเจเอฟซียูนิต ดังนั้นจะสามารถทดสอบโปรแกรมที่พัฒนาด้วยคอมโพเนนท์สวิงเท่านั้น ใช้ฐานข้อมูลมายเอสคิวแอล (MySQL) ในการเก็บข้อมูล ข้อมูลนำเข้าของเครื่องมือประกอบด้วยแฟ้มเอกซ์เอ็มแอลของระบบซึ่งอธิบายแผนภาพต่างๆ ไว้ แสดงดังรูปที่ 16

3.4.1 การวิเคราะห์และออกแบบเครื่องมือสร้างกรณีทดสอบจียูไอจากยูสเคส

การสร้างกรณีทดสอบเป็นแบบกึ่งอัตโนมัติ ซึ่งจะอาศัยเครื่องมือที่สร้างขึ้น และผู้ดำเนินการทดสอบเป็นผู้แก้ไขรายละเอียดบางส่วนเพื่อความสมบูรณ์ของการทดสอบ โดยแบ่งออกเป็น 3 ขั้นตอน ดังนี้

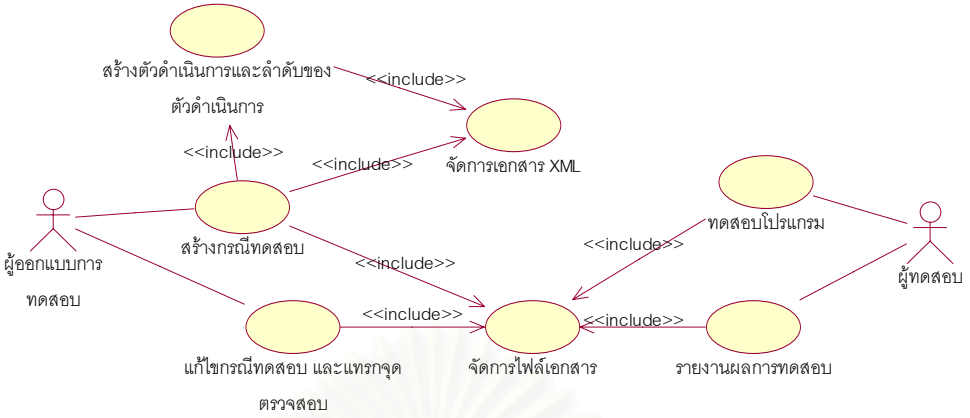
1. ในขั้นตอนแรกเป็นการใช้เครื่องมือสร้างตัวดำเนินการและแผนการทดสอบ เพื่อใช้ในการสร้างกรณีทดสอบแบบอัตโนมัติ โดยข้อมูลนำเข้าเป็นเอกสารการออกแบบระบบซึ่งสร้างจากเครื่องมือเรชันเนลโรส และอยู่ในรูปแบบแฟ้มเอกซ์เอ็มแอล โดยเป็นแฟ้มที่มีรายละเอียดของแผนภาพต่างๆ ดังแสดงในรูปที่ 16 โดยใช้หลักการเขียนแผนภาพในตอนต้นของบทนี้และในภาคผนวก ก เพื่อนำมาสร้างเป็นแฟ้มตัวดำเนินการ และแฟ้มแผนการทดสอบ แบบอัตโนมัติซึ่งผลลัพธ์ที่ได้อยู่ในรูปแบบแฟ้มเอกซ์เอ็มแอล
2. ขั้นตอนในการแก้ไขข้อมูลซึ่งจะให้ผู้ออกแบบการทดสอบเป็นผู้กรอกข้อมูลที่จำเป็นเข้าไปยังแฟ้มตัวดำเนินการ เช่น ค่าข้อมูลนำเข้า (Input value) ซึ่งค่าข้อมูลนำเข้าเหล่านี้ขึ้นกับดุลพินิจของผู้ออกแบบการทดสอบ เพื่อให้กรณีทดสอบมีความสมบูรณ์มากขึ้น

3. ขั้นตอนการแก้ไขข้อมูลจุดตรวจสอบ เพื่อสามารถตรวจสอบความถูกต้องของการทดสอบได้ โดยผู้ดำเนินการทดสอบเป็นผู้สร้างจุดตรวจสอบ ในลักษณะเงื่อนไข เช่น “Welcome window” = true หมายถึงมีการเปิดหน้าต่าง “Welcome window” เป็นต้น

รูปที่ 16 ตัวอย่างแผนภาพยูเอ็มแอลเมื่อแปลงเป็นแฟ้มเอกซ์เอ็มแอล

3.4.2 แผนภาพยูสเคสของเครื่องมือ

แผนภาพยูสเคสของเครื่องมือแสดงดังรูปที่ 17 ผู้ออกแบบการทดสอบสามารถสร้างกรณีทดสอบ และแก้ไขกรณีทดสอบ รวมทั้งแทรกจุดตรวจสอบได้ ผู้ทดสอบสามารถทำการทดสอบโปรแกรม และเรียกค้นรายงานการทดสอบได้



รูปที่ 17 แผนภาพยูสเคสของเครื่องมือสร้างและทดสอบกรณีทดสอบ

คำอธิบายยูสเคสของแต่ละยูสเคส แสดงดังตารางที่ 8-14

ตารางที่ 8 คำอธิบายยูสเคสสร้างกรณีทดสอบ

หมายเลขยูสเคส	UC1
ชื่อยูสเคส	สร้างกรณีทดสอบ
คำอธิบาย	สร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้
ผู้ใช้	ผู้ออกแบบการทดสอบ
ความสัมพันธ์	Include = UC2, UC3, UC5
ขั้นตอนการทำงานหลัก	1. include=UC5 2. include=UC2 3. เปิดแฟ้มเอกซ์เอ็มแอลของตัวดำเนินการและแผนการทดสอบ 4. สร้างกรณีทดสอบ 5. จัดเก็บกรณีทดสอบเป็นแฟ้มเอกซ์เอ็มแอล 6. แปลงกรณีทดสอบเป็นสคริปต์ทดสอบ
ขั้นตอนการทำงานอื่นๆ	-

ตารางที่ 9 คำอธิบายยูสเคสสร้างตัวดำเนินการและแผนการทดสอบ

หมายเลขยูสเคส	UC2
ชื่อยูสเคส	สร้างตัวดำเนินการและแผนการทดสอบ
คำอธิบาย	สร้างตัวดำเนินการและแผนการทดสอบ เป็นแฟ้มเอกซ์เอ็มแอล
ผู้ใช้	
ความสัมพันธ์	Include = UC3
ขั้นตอนการทำงานหลัก	1. เปิดแฟ้มเอกซ์เอ็มแอลของระบบ

	2. อ่านข้อมูลแผนภาพต่างๆ เก็บลงในฐานข้อมูล 3. สร้างตัวดำเนินการและแผนการทดสอบ 4. จัดเก็บตัวดำเนินการและแผนการทดสอบในแฟ้มเอกซ์เอ็มแอล
ขั้นตอนการทำงานอื่นๆ	-

ตารางที่ 10 คำอธิบายยูสเคสแก้ไขกรณีทดสอบและแทรกจุดตรวจสอบ

หมายเลขยูสเคส	UC4
ชื่อยูสเคส	แก้ไขกรณีทดสอบและแทรกจุดตรวจสอบ
คำอธิบาย	แก้ไขกรณีทดสอบและแทรกจุดตรวจสอบ
ผู้ใช้	ผู้ออกแบบการทดสอบ
ความสัมพันธ์	Include = UC5
ขั้นตอนการทำงานหลัก	1. include = UC5 2. อ่านข้อมูลกรณีทดสอบจากแฟ้มเอกซ์เอ็มแอล 3. แทรกจุดตรวจสอบ แก้ไขการกรณีทดสอบ 4. บันทึกกรณีทดสอบ และแปลงเป็นสคริปต์ทดสอบ
ขั้นตอนการทำงานอื่นๆ	-

ตารางที่ 11 คำอธิบายยูสเคสจัดการแฟ้มเอกซ์เอ็มแอล

หมายเลขยูสเคส	UC3
ชื่อยูสเคส	จัดการแฟ้มเอกซ์เอ็มแอล
คำอธิบาย	เปิด สร้าง แก้ไข บันทึก แฟ้มเอกซ์เอ็มแอล
ผู้ใช้	
ความสัมพันธ์	
ขั้นตอนการทำงานหลัก	1. เปิดแฟ้มเอกซ์เอ็มแอล 2. อ่านข้อมูลในแฟ้มเอกซ์เอ็มแอล 3. เก็บข้อมูลที่อ่านได้ไว้ในหน่วยความจำในรูปโครงสร้างต้นไม้
ขั้นตอนการทำงานอื่นๆ	A.1. สร้างแฟ้มเอกซ์เอ็มแอลใหม่ A.2. สร้างโนดในแฟ้มเพิ่มลงในโครงสร้างต้นไม้ A.3. บันทึกแฟ้มเอกซ์เอ็มแอล B.1. เปิดแฟ้มเอกซ์เอ็มแอล B.2. อ่านข้อมูลในแฟ้มเอกซ์เอ็มแอล B.3. แก้ไขโนดที่ต้องการ B.4. บันทึกโครงสร้างต้นไม้ที่แก้ไขแล้วเป็นแฟ้มเอกซ์เอ็มแอล

ตารางที่ 12 จัดการแฟ้ม

หมายเลขยูสเคส	UC5
ชื่อยูสเคส	จัดการแฟ้ม
คำอธิบาย	เปิด สร้าง แก้ไข และบันทึกแฟ้ม
ผู้ใช้	
ความสัมพันธ์	
ขั้นตอนการทำงานหลัก	4. เปิดแฟ้ม 5. อ่านข้อมูลในแฟ้ม 6. เก็บข้อมูลที่อ่านได้ไว้ในหน่วยความจำในรูปสตริง
ขั้นตอนการทำงานอื่นๆ	A.4. สร้างแฟ้มใหม่ A.5. เพิ่มข้อมูลลงในแฟ้ม A.6. บันทึกแฟ้ม B.5. เปิดแฟ้มเก่า B.6. อ่านข้อมูลในแฟ้ม B.7. แก้ไขข้อมูลตามต้องการ B.8. บันทึกสตริงของแฟ้มลงในแฟ้ม

ตารางที่ 13 ทดสอบโปรแกรม

หมายเลขยูสเคส	UC6
ชื่อยูสเคส	ทดสอบโปรแกรม
คำอธิบาย	นำสคริปต์ทดสอบที่สร้างไว้มาทดสอบกับโปรแกรม
ผู้ใช้	ผู้ทดสอบ
ความสัมพันธ์	Include = UC5
ขั้นตอนการทำงานหลัก	1. เปิดแฟ้มสคริปต์ทดสอบ 2. อ่านข้อมูลสคริปต์ 3. เปิดโปรแกรมทดสอบ 4. ทดสอบตามสคริปต์ 5. บันทึกผลการทดสอบ
ขั้นตอนการทำงานอื่นๆ	

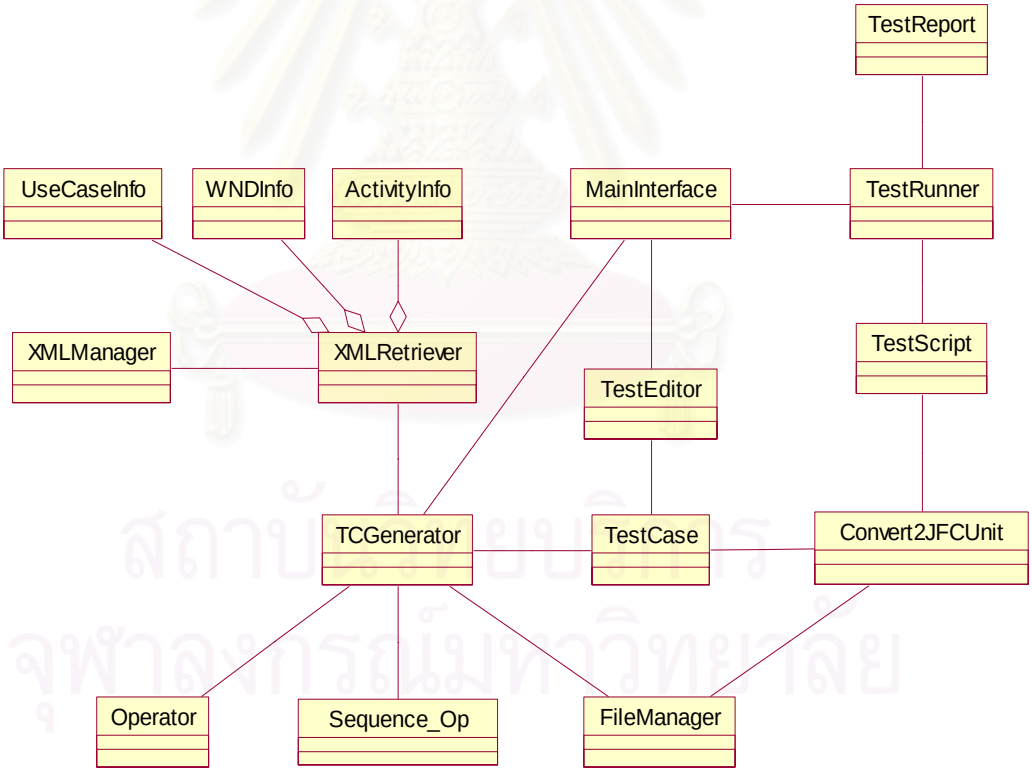
ตารางที่ 14 รายงานผลการทดสอบ

หมายเลขยูสเคส	UC7
ชื่อยูสเคส	รายงานผลการทดสอบ
คำอธิบาย	จัดทำรายงานผลการทดสอบ

ผู้ใช้	ผู้ทดสอบ
ความสัมพันธ์	Include = UC5
ขั้นตอนการทำงานหลัก	1. อ่านข้อมูลรายงานผลการทดสอบ 2. สร้างแฟ้มรายงานผลการทดสอบ 3. เพิ่มข้อมูล 4. บันทึกผลการทดสอบในรูปแฟ้มเอชทีเอ็มแอล
ขั้นตอนการทำงานอื่นๆ	

3.4.3 แผนภาพคลาสของเครื่องมือ

แผนภาพคลาสของเครื่องมือแสดงดังรูปที่ 18 ซึ่งประกอบด้วยส่วนประกอบหลัก 3 ส่วน คือ ส่วนสร้างกรณีทดสอบและสคริปต์ทดสอบ ส่วนทดสอบ และส่วนสร้างรายงานผลการทดสอบ โดยมีคลาส MainInterface เป็นคลาสหลักในการทำงาน



รูปที่ 18 แผนภาพคลาสของเครื่องมือสร้างและทดสอบกรณีทดสอบ

รายละเอียดของแต่ละคลาส แสดงดังต่อไปนี้

1. คลาส UsecaseInfo คลาสทำหน้าที่คัดแยกข้อมูลของแผนภาพยูสเคสและคำอธิบายยูสเคส จัดเก็บในฐานข้อมูล รายละเอียดแสดงดังตารางที่ 15

ตารางที่ 15 รายละเอียดคลาส UsecaseInfo

ชื่อคลาส	UsecaseInfo
ตัวแปร	UsecaseU เก็บข้อมูลของผู้ใชและยูสเคส รวมทั้งการทำงานหลักและการทำงานอื่นๆ UsecaseAs เก็บข้อมูลความสัมพันธ์ระหว่างผู้ใช้กับยูสเคสและยูสเคสกับยูสเคส proID รหัสโครงการ
เมธอด	UsecaseInfo() คัดแยกข้อมูลของแผนภาพยูสเคสและคำอธิบายยูสเคสเก็บลงในฐานข้อมูล

2. คลาส WndInfo คลาสทำหน้าที่คัดแยกข้อมูลของแผนภาพดับเบิลยูเอ็นดี จัดเก็บในฐานข้อมูล รายละเอียดแสดงดังตารางที่ 16

ตารางที่ 16 รายละเอียดคลาส WndInfo

ชื่อคลาส	WndInfo
ตัวแปร	Wnd_node เก็บข้อมูลของโนดต่างๆ ในแผนภาพดับเบิลยูเอ็นดี และคอมโพเนนท์ที่อยู่ Tran เก็บข้อมูลการเชื่อมต่อของโนดในแผนภาพดับเบิลยูเอ็นดี proID รหัสโครงการ
เมธอด	WndInfo() คัดแยกข้อมูลของแผนภาพดับเบิลยูเอ็นดีเก็บลงในฐานข้อมูล

3. คลาส ActivityInfo คลาสทำหน้าที่คัดแยกข้อมูลของแผนภาพกิจกรรม จัดเก็บในฐานข้อมูล รายละเอียดแสดงดังตารางที่ 17

ตารางที่ 17 รายละเอียดคลาส ActivityInfo

ชื่อคลาส	ActivityInfo
ตัวแปร	actSt เก็บข้อมูลโนดในแผนภาพกิจกรรม start เก็บข้อมูลจุดเริ่มต้น finish เก็บข้อมูลจุดสิ้นสุด swim เก็บข้อมูลสวิมเลน fork เก็บข้อมูลจุดแยกประเภทออกหลายทาง branch เก็บข้อมูลจุดตัดสินใจ proID รหัสโครงการ
เมธอด	ActivityInfo() คัดแยกข้อมูลของแผนภาพกิจกรรมเก็บลงในฐานข้อมูล

4. คลาส XMLManager คลาสทำหน้าที่จัดการข้อมูลของแฟ้มเอกซ์เอ็มแอล รายละเอียดแสดงดังตารางที่ 18

ตารางที่ 18 รายละเอียดคลาส XMLManager

ชื่อคลาส	XMLManager
ตัวแปร	Doc ตัวแปรโครงสร้างต้นไม้ของแฟ้มเอกซ์เอ็มแอล File ชื่อแฟ้มเอกซ์เอ็มแอล
เมธอด	XMLManager() จัดการข้อมูลของแฟ้มเอกซ์เอ็มแอล loadXML() เปิดและอ่านแฟ้มเอกซ์เอ็มแอล saveXML() บันทึกแฟ้มเอกซ์เอ็มแอล

5. คลาส XML_retrieve คลาสทำหน้าที่คัดแยกข้อมูลของแผนภาพยูเอ็มแอลในขั้นต้น เพื่อส่งให้คลาส UsecaseInfo WndInfo และ ActivityInfo จัดการต่อ รายละเอียดแสดงดังตารางที่ 19

ตารางที่ 19 รายละเอียดคลาส XML_retrieve

ชื่อคลาส	XML_retrieve
ตัวแปร	ProID รหัสโครงการ Inp แฟ้มเอกซ์เอ็มแอลของระบบ
เมธอด	Main1() คัดแยกข้อมูลของแผนภาพยูเอ็มแอลในขั้นต้น จากนั้นส่งไปยังออปเจ็คของคลาส WNDInfo UsecaseInfo และ ActivityInfo ตามลำดับ

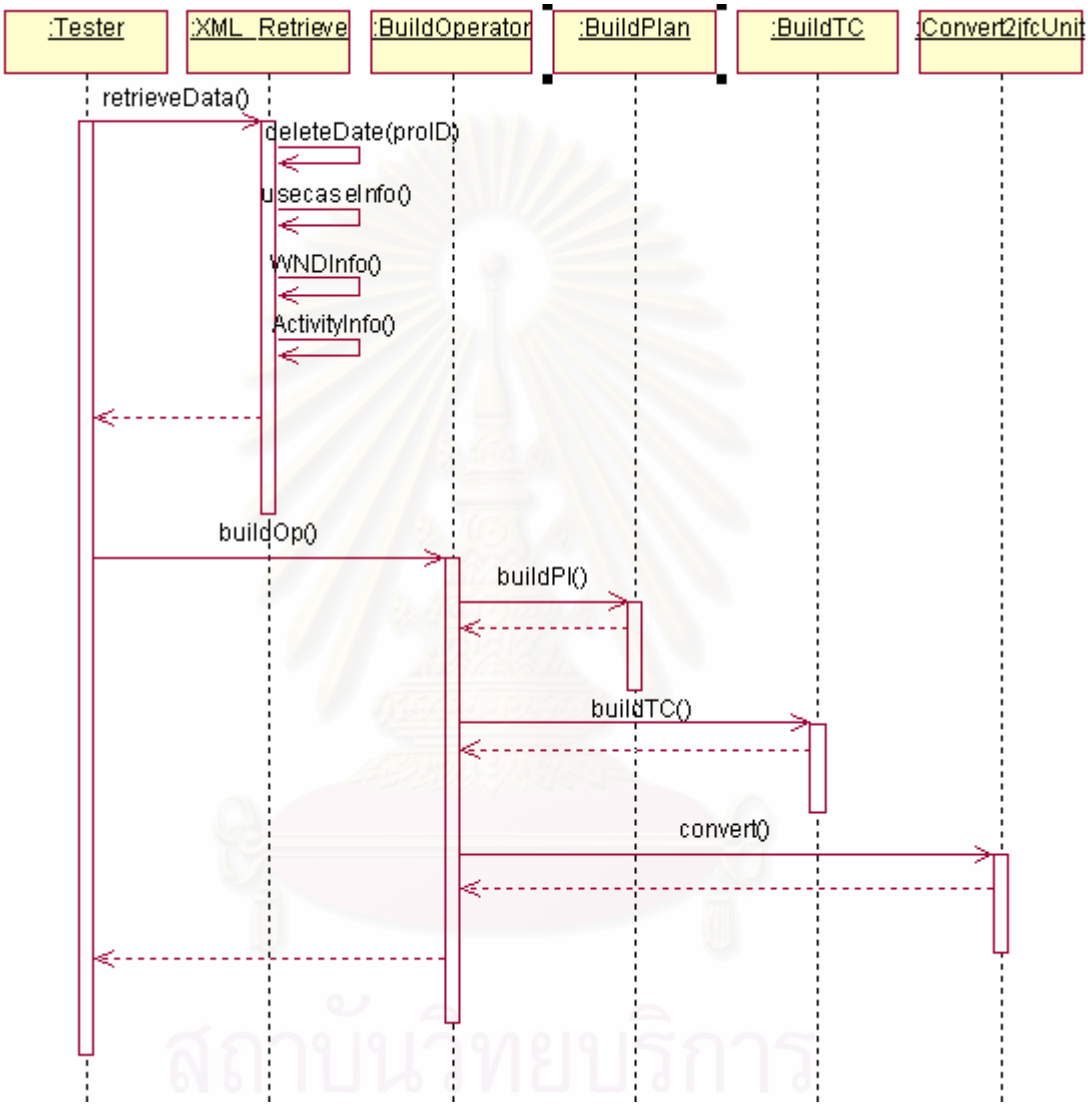
6. คลาส WndInfo คลาสทำหน้าที่คัดแยกข้อมูลของแผนภาพดับเบิลยูเอ็นดี จัดเก็บในฐานข้อมูล รายละเอียดแสดงดังตารางที่ 20

ตารางที่ 20 รายละเอียดคลาส WndInfo

ชื่อคลาส	WndInfo
ตัวแปร	Wnd_node เก็บข้อมูลโนดของแผนภาพดับเบิลยูเอ็นดี และคอมโพเนนท์ที่บรรจุ Tran เก็บข้อมูลเส้นเชื่อม proID รหัสโครงการ
เมธอด	WndInfo () คัดแยกข้อมูลของแผนภาพดับเบิลยูเอ็นดีเก็บลงในฐานข้อมูล

3.4.4 แผนภาพลำดับของเครื่องมือ

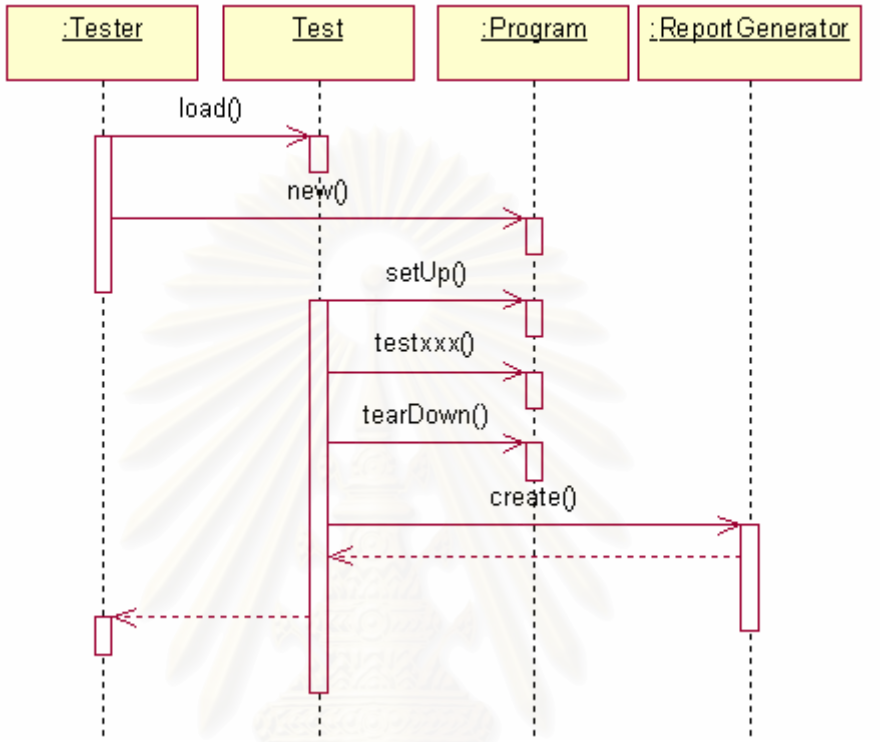
ขั้นตอนการทำงานที่สำคัญของเครื่องมือแบ่งออกเป็น 2 ส่วน คือ การสร้างกรณีทดสอบ และการทดสอบกรณีทดสอบที่สร้างขึ้น สามารถเขียนเป็นแผนภาพลำดับได้ดังนี้



รูปที่ 19 แผนภาพลำดับของการสร้างกรณีทดสอบ

จากรูปที่ 19 แสดงแผนภาพลำดับของการสร้างกรณีทดสอบ โดยเริ่มจากการลบข้อมูลในฐานข้อมูลซึ่งมีหมายเลขโครงการเดียวกันกับที่ต้องการสร้างกรณีทดสอบ จากนั้นทำการค้นหาข้อมูลของแผนภาพยูเอ็มแอล ซึ่งประกอบด้วย ข้อมูลแผนภาพยูสเคส คำอธิบายยูสเคส ข้อมูลแผนภาพดับเบิลยูเอ็นดี และข้อมูลแผนภาพกิจกรรม เมื่อทำการค้นหาข้อมูลเรียบร้อยแล้ว นำข้อมูลดังกล่าวมาสร้างเป็นตัวดำเนินการและแผนการทดสอบ เมื่อสร้างตัวดำเนินการและ

แผนการทดสอบเรียบร้อยแล้ว ทำการสร้างกรณีทดสอบ ในขั้นตอนสุดท้ายแปลงกรณีทดสอบเป็นสคริปต์ทดสอบ



รูปที่ 20 แผนภาพลำดับของการทดสอบ

จากรูปที่ 20 แสดงแผนภาพลำดับของการทดสอบโปรแกรม โดยใช้สคริปต์ทดสอบที่ได้จากกรณีทดสอบที่สร้างขึ้น เริ่มจากการเปิดสคริปต์ทดสอบ เปิดโปรแกรมที่จะทดสอบ จากนั้นทำการทดสอบตามสคริปต์ทดสอบ เมื่อทำการทดสอบเรียบร้อยแล้ว เครื่องมือจะสร้างรายงานผลการทดสอบและจัดเก็บไว้ในฐานข้อมูล

3.4.5 รายงานที่ได้จากการทดสอบกรณีทดสอบ

ผลการทดสอบที่ได้จากการเล่นซ้ำสคริปต์ทดสอบ ซึ่งเครื่องมือจะจัดเก็บข้อมูลการทดสอบไว้ในฐานข้อมูลและสามารถเรียกค้นเพื่อแสดงเป็นรายงานในรูปแบบแฟ้มเอชทีเอ็มแอล (HTML) ได้ โดยรูปแบบของรายงานผลการทดสอบมีรูปแบบ ดังตารางที่ 21

ตารางที่ 21 รูปแบบของรายงานผลการทดสอบ

โครงการ ชื่อโครงการที่ทดสอบ **รหัส** รหัสของโครงการ					**วันที่** วันที่ทำการทดสอบ
ผู้ทำการทดสอบ ชื่อผู้ทดสอบ		**ผู้ออกแบบกรณีทดสอบ** รหัสผู้ออกแบบกรณีทดสอบ			
ลำดับ	**ชื่อกรณีทดสอบ**	**ผ่าน**	**ไม่ผ่าน**	**เวลาในการทดสอบ** (millisecond)	**หมายเหตุ**
	รวม	จำนวนผ่าน	จำนวนไม่ผ่าน	เวลารวมในการทดสอบ	

รายงานจะประกอบด้วยรายละเอียดของแต่ละโครงการที่ได้ดำเนินการทดสอบ จัดเก็บไว้ในฐานข้อมูลและสามารถเรียกค้นมาแสดงผลได้โดยเลือกชื่อโครงการที่ต้องการแสดงรายงานแสดงผลและจัดเก็บเป็นแฟ้มเอชทีเอ็มแอล เพื่อความสะดวกในการอ่านและสามารถพิมพ์รายงานได้อย่างสวยงาม มีรายละเอียดดังนี้

1. โครงการ คือชื่อโครงการที่ตั้งเมื่อเริ่มการสร้างกรณีทดสอบ
2. รหัส คือรหัสของโครงการที่จัดเก็บไว้ในฐานข้อมูลเพื่อไม่ให้ซ้ำกัน
3. วันที่ คือวันที่ทำการทดสอบกรณีทดสอบ
4. ผู้ทำการทดสอบ คือชื่อผู้ทำการทดสอบ
5. ผู้ออกแบบกรณีทดสอบ คือรหัสของผู้ออกแบบการทดสอบของโครงการนี้
6. ลำดับ แสดงลำดับของกรณีทดสอบ
7. ชื่อกรณีทดสอบ แสดงชื่อของกรณีทดสอบแต่ละกรณีทดสอบ
8. ผ่าน/ไม่ผ่าน แสดงผลการทดสอบว่ากรณีทดสอบนั้นๆ ผ่านหรือไม่ผ่านการทดสอบ
9. เวลาในการทดสอบ คือเวลาตั้งแต่เริ่มทดสอบกรณีทดสอบนั้นๆ จนจบกรณีทดสอบหรือเกิดความผิดพลาดขึ้น
10. หมายเหตุ แสดงข้อความหมายเหตุของผู้ทดสอบต่อกรณีทดสอบนั้นๆ
11. รวม แสดลผลรวมของกรณีทดสอบที่ผ่าน/ไม่ผ่าน และเวลารวมที่ใช้ในการทดสอบ

บทที่ 4

การพัฒนาเครื่องมือสร้างกรณีทดสอบจียูไอจากยูเอ็มแอล

4.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา

สภาพแวดล้อมที่ใช้ในการพัฒนาระบบมีสภาพแวดล้อมทางด้านฮาร์ดแวร์และซอฟต์แวร์ดังต่อนี้

4.1.1 ฮาร์ดแวร์

1. หน่วยประมวลผล อินเทลเพนเทียม 4 3.0 กิกะเฮิร์ต (Pentium 4 3.0 GHz.)
2. หน่วยความจำ (RAM) 512 เมกกะไบต์ (512 MB)
3. ฮาร์ดดิสก์ (Harddisk) 80 กิกะไบต์ (80 GB)

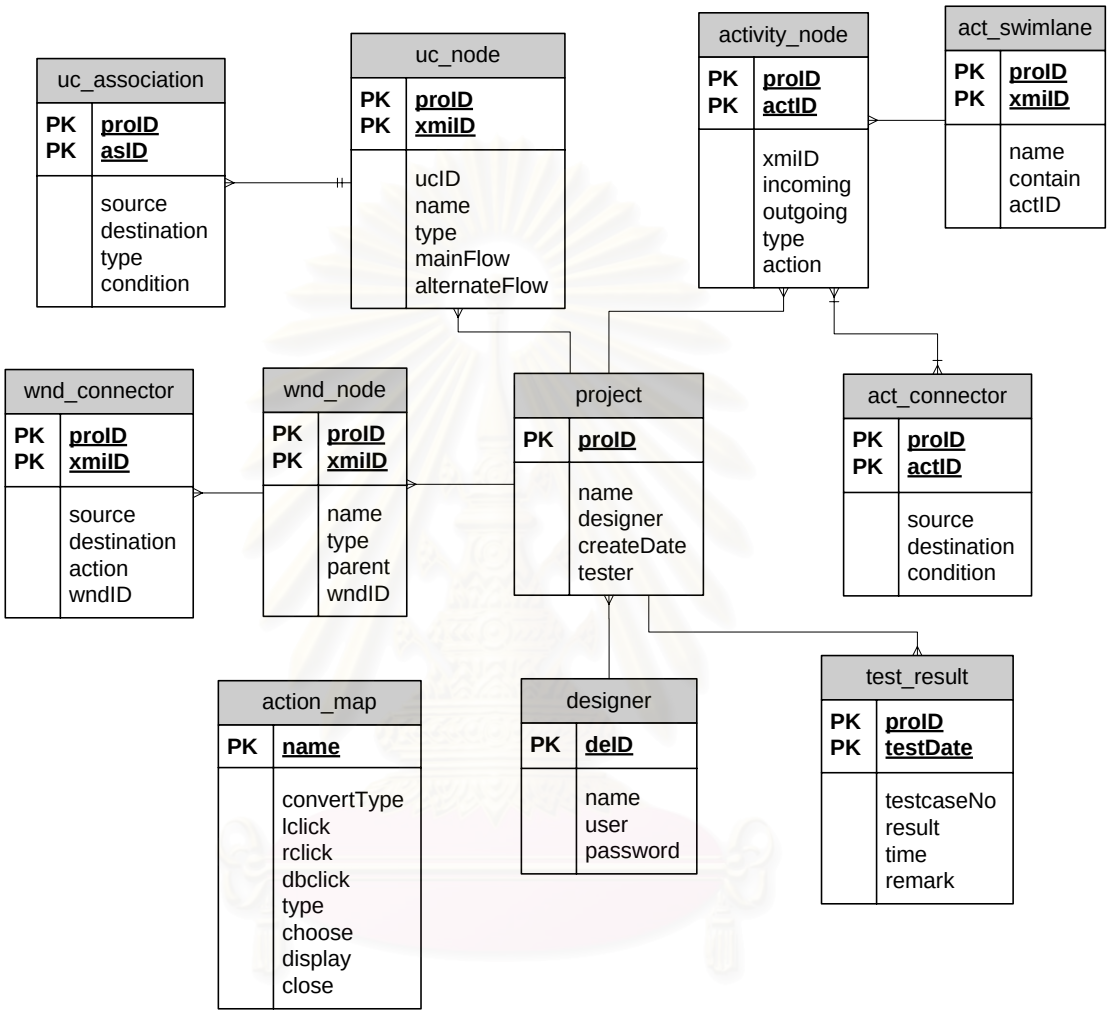
4.1.2 ซอฟต์แวร์

1. ระบบปฏิบัติการ วินโดวส์ เอกซ์พี โปรเฟสเซอร์นอล (Windows XP Professional)
2. ชุดพัฒนาโปรแกรมภาษาจาวา เวอร์ชัน 1.4.2 (J2sdk 1.4.2)
3. เจเอฟซียูนิต เวอร์ชัน 2.08
4. เครื่องมือพัฒนาโปรแกรมภาษาจาวา อีคลิป เอสดีเค เวอร์ชัน 3.1.0 (Eclipse SDK version 3.1.0)
5. ระบบจัดการฐานข้อมูลมายเอสคิวแอล เวอร์ชัน 1.4 (MySQL version 1.4) พร้อมติดตั้งเจดีบีซี สำหรับมายเอสคิวแอล เวอร์ชัน 3.0.15 (JDBC – mysql-connector-java-3.1.05-ga-bin.jar) เพื่อให้สามารถติดต่อกับโปรแกรมภาษาจาวาได้
6. เครื่องมือวาดแผนภาพยูเอ็มแอล เรชันเนล โรส เอ็นเตอร์ไพรส์ เวอร์ชัน 2002
7. เครื่องมือแปลงแผนภาพยูเอ็มแอลเป็นแฟ้มเอกซ์เอ็มแอล ยูนิซิส เอกซ์เอ็มไอ 1.3 แอด-อิน

4.2 ฐานข้อมูลของเครื่องมือ

ฐานข้อมูลประกอบด้วยข้อมูลของโครงการ ข้อมูลของแผนภาพต่างๆ ข้อมูลผู้ออกแบบการทดสอบ ข้อมูลการทดสอบ โดยมีความสัมพันธ์กัน คือ ตารางโครงการเก็บข้อมูลของโครงการ โดยตารางซึ่งเก็บข้อมูลแผนภาพทุกแผนภาพจะอ้างอิงกับหมายเลขโครงการ และมีผู้ออกแบบการ

ทดสอบ 1 คน แต่สามารถทำการทดสอบได้หลายครั้งโดยยึดวันที่ทดสอบเป็นสำคัญ โดยแสดงได้ด้วยแผนภาพอีอาร์ (E-R diagram) ดังรูปที่ 21



รูปที่ 21 แผนภาพอีอาร์ของฐานข้อมูล

จากรูปที่ 21 สามารถเขียนรายละเอียดของแผนภาพอีอาร์ของฐานข้อมูล โดยแสดงข้อมูลแต่ละตารางซึ่งประกอบด้วยข้อมูลชื่อตาราง คีย์หลักของตาราง ความสัมพันธ์กับตารางอื่นๆ ชื่อสดมภ์ ประเภทข้อมูล ค่าเป็น null ค่าช่วงข้อมูลและหมายเหตุของแต่ละตาราง โดยหมายเหตุเป็นคำอธิบายความหมายหรือเงื่อนไขของสดมภ์ แสดงได้ดังต่อไปนี้

1. ตาราง project เก็บข้อมูลรายละเอียดของโครงการ แสดงดังตารางที่ 22

ตารางที่ 22 ตาราง project

ชื่อ	usecase_node			
คีย์หลัก	proID		**ความสัมพันธ์**	designer, test_result, usecase_node, wnd_node, activity_node
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
proID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	
name	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	
createDate	DATETIME	ไม่ได้	YYYY-MM-DD HH:MM:SS	
designer	VARCHAR(45)	ไม่ได้	1 – 45 ตัวอักษร	
tester	VARCHAR(45)	ไม่ได้	1 – 45 ตัวอักษร	จากตาราง designer
xmlInput	VARCHAR(45)	ไม่ได้	1 – 45 ตัวอักษร	สถานที่เก็บแผนภาพยูเอ็มแอลในรูปของแฟ้มเอ็กซ์ของระบบที่ต้องการสร้างกรณีทดสอบ โดยเก็บเป็นพาทของแฟ้มเอกซ์เอ็มแอล

2. ตาราง test_result เก็บข้อมูลการทดสอบของแต่ละโครงการ แสดงดังตารางที่ 23

ตารางที่ 23 ตาราง test_result

ชื่อ	test_result			
คีย์หลัก	proID,testDate		**ความสัมพันธ์**	project
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
proID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	proID จากตาราง project
testDate	DATETIME	ไม่ได้	YYYY-MM-DD HH:MM:SS	วันที่ทำการทดสอบ
testcaseNo	VARCHAR(45)	ไม่ได้	1 – 45 ตัวอักษร	หมายเลขของกรณีทดสอบ
result	VARCHAR(45)	ไม่ได้	1 – 45 ตัวอักษร	ผลการทดสอบผ่าน/ไม่ผ่าน หากไม่ผ่านจะเขียนข้อความแสดงความผิดพลาดด้วย
time	INTEGER	ไม่ได้	1-4294967295	ระยะเวลาการทดสอบของแต่ละกรณีทดสอบหน่วยเป็น millisecond

remark	VARCHAR(45)	ได้	0 – 45 ตัวอักษร	หมายเหตุ
tester	VARCHAR(45)	ไม่ได้	1 – 45 ตัวอักษร	ชื่อผู้ทำการทดสอบ

3. ตาราง designer เก็บข้อมูลการทดสอบของผู้ออกแบบการทดสอบ แสดงดังตารางที่ 24

ตารางที่ 24 ตาราง designer

ชื่อ	designer			
คีย์หลัก	deID		**ความสัมพันธ์**	project
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
deID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	รหัสของผู้ออกแบบการทดสอบ
name	VARCHAR(100)	ไม่ได้	1 – 100 ตัวอักษร	ชื่อผู้ออกแบบการทดสอบ
username	VARCHAR(8)	ไม่ได้	1 – 8 ตัวอักษร	ชื่อเข้าใช้
password	VARCHAR(8)	ไม่ได้	1 – 8 ตัวอักษร	รหัสผ่าน

4. ตาราง usecase_node เก็บข้อมูลยูสเคส ผู้ใช้ และคำอธิบายยูสเคส แสดงดังตารางที่ 25

ตารางที่ 25 ตาราง usecase_node

ชื่อ	usecase_node			
คีย์หลัก	proID, xmiID		**ความสัมพันธ์**	project, uc_node
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
proID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	จากตาราง project
ucID	VARCHAR(10)	ได้	3 – 10 ตัวอักษร	เมื่อ type = actor
name	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	ชื่อยูสเคส
type	VARCHAR(20)	ไม่ได้	usecase, actor	ประเภท
mainFlow	Text	ไม่ได้	1 – 2^6 ตัวอักษร	การทำงานหลัก
alternateFlow	Text	ได้	0 – 2^6 ตัวอักษร	เป็น null เมื่อยูสเคสไม่มีการทำงานอื่นๆ

5. ตาราง uc_association เก็บข้อมูลความสัมพันธ์ของยูสเคส แสดงดังตารางที่ 26

ตารางที่ 26 ตาราง uc_association

ชื่อ	uc_association			
คีย์หลัก	proID, xmiID		**ความสัมพันธ์**	project, uc_node
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
proID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	จากตาราง project
asID	VARCHAR(10)	ได้	0 – 10 ตัวอักษร	หมายเลขของเส้นความสัมพันธ์

source	VARCHAR(20)	ได้	0 – 20 ตัวอักษร	xmiID ของ uc_node
destination	VARCHAR(20)	ได้	0 – 20 ตัวอักษร	xmiID ของ uc_node
type	VARCHAR(20)	ไม่ได้	Include, extend, normal	ประเภทของความสัมพันธ์
condition	VARCHAR(200)	ได้	0 – 200 ตัวอักษร	เมื่อ type = extend

6. ตาราง activity_node เก็บข้อมูลกิจกรรมในแผนภาพกิจกรรม แสดงดังตารางที่ 27

ตารางที่ 27 ตาราง acticity_node

ชื่อ	activity_node			
คีย์หลัก	proID, xmiID		**ความสัมพันธ์**	project, act_connector
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
proID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	จากตาราง project
actID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	หมายเลขของกิจกรรม
action	Text	ไม่ได้	1 – 2^6 ตัวอักษร	การกระทำในแต่ละกิจกรรม
type	VARCHAR(20)	ไม่ได้	start, final, activity, branch, fork	ประเภท
incoming	VARCHAR(20)	ได้	0 – 20 ตัวอักษร	เส้นที่เข้ามายังโนด จากตาราง act_connector
outgoing	VARCHAR(20)	ได้	0 – 20 ตัวอักษร	เส้นที่ออกจากโนด จากตาราง act_connector

7. ตาราง act_connector เก็บข้อมูลเส้นเชื่อมในแผนภาพกิจกรรม แสดงดังตารางที่ 28

ตารางที่ 28 ตาราง act_connector

ชื่อ	act_connector			
คีย์หลัก	proID, actID		**ความสัมพันธ์**	project, activity
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
proID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	จากตาราง project
actID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	หมายเลขเส้นเชื่อม
source	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	actID จากตาราง activity_node
destination	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	actID จากตาราง activity_node
condition	VARCHAR(20)	ได้	0 – 20 ตัวอักษร	ไม่เป็น null เมื่อเชื่อมจาก source ที่มี type=branch (การตัดสินใจ)

8. ตาราง act_swimlane เก็บข้อมูลสวิมเลนในแผนภาพกิจกรรม แสดงดังตารางที่ 29

ตารางที่ 29 ตาราง act_swimlane

ชื่อ	act_swimlane			
คีย์หลัก	proID, name		**ความสัมพันธ์**	project, activity
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
proID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	จากตาราง project
name	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	ชื่อสวิมเลน
contain	VARCHAR(20)	ได้	0 – 20 ตัวอักษร	actID จากตาราง activity_node

9. ตาราง wnd_node เก็บข้อมูลแผนภาพดับเบิลยูเอ็นดี แสดงดังตารางที่ 30

ตารางที่ 30 ตาราง wnd_node

ชื่อ	wnd_node			
คีย์หลัก	proID, xmiID		**ความสัมพันธ์**	project, wnd_connector
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
proID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	จากตาราง project
xmiID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	
name	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	ชื่อคอมโพเนนท์
type	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	ประเภทคอมโพเนนท์
wndID	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	หมายเลขโนด
parent	VARCHAR(20)	ได้	1 – 20 ตัวอักษร	คอมโพเนนท์ที่บรรจุ

10. ตาราง wnd_connector เก็บข้อมูลเส้นเชื่อมในแผนภาพดับเบิลยูเอ็นดี แสดงดังตารางที่ 31

ตารางที่ 31 ตาราง wnd_connector

ชื่อ	wnd_connector			
คีย์หลัก	proID, xmiID		**ความสัมพันธ์**	project, wnd_connector
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
proID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	จากตาราง project
xmiID	VARCHAR(10)	ไม่ได้	1 – 10 ตัวอักษร	
wndID	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	
source	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	xmiID จากตาราง wnd_node
destination	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	xmiID จากตาราง wnd_node
action	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	การกระทำที่ก่อให้เกิดเส้นเชื่อม

11. ตาราง action_map เก็บข้อมูลการแปลงประเภทของคอมโพเนนท์และการกระทำที่รองรับ แสดงดังตารางที่ 32

ตารางที่ 32 ตาราง action_map

ชื่อ	action_map			
คีย์หลัก	name		**ความสัมพันธ์**	-
ชื่อสดมภ์	**ประเภทข้อมูล**	**ค่าเป็น null**	**ช่วงข้อมูล**	**หมายเหตุ**
name	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	ชื่อของคอมโพเนนท์
convertType	VARCHAR(20)	ไม่ได้	1 – 20 ตัวอักษร	ประเภทของคอมโพเนนท์เมื่อใช้กับคอมโพเนนท์สวิง
lclick	TINYINT(1)	ไม่ได้	0,1	รองรับการคลิกซ้าย หรือไม่ 0 ไม่รองรับ 1 รองรับ
rclick	TINYINT(1)	ไม่ได้	0,1	รองรับการคลิกขวา หรือไม่ 0 ไม่รองรับ 1 รองรับ
dbclick	TINYINT(1)	ไม่ได้	0,1	รองรับการคลิกซ้าย 2 ครั้ง หรือไม่ 0 ไม่รองรับ 1 รองรับ
type	TINYINT(1)	ไม่ได้	0,1	รองรับการพิมพ์ หรือไม่ 0 ไม่รองรับ 1 รองรับ
choose	TINYINT(1)	ไม่ได้	0,1	รองรับการเลือก หรือไม่ 0 ไม่รองรับ 1 รองรับ
display	TINYINT(1)	ไม่ได้	0,1	รองรับการเปิด หรือไม่ 0 ไม่รองรับ 1 รองรับ
close	TINYINT(1)	ไม่ได้	0,1	รองรับการปิด หรือไม่ 0 ไม่รองรับ 1 รองรับ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การทดสอบเครื่องมือ

บทนี้กล่าวถึงการทดสอบการสร้างกรณีทดสอบจียูไอจากยูเอ็มแอล โดยระบบที่ใช้ทดสอบสร้างขึ้นตามแนวทางที่ได้นำเสนอในบทที่ 3 สภาพแวดล้อมที่ใช้ในการทดสอบ และผลที่ได้จากการทดสอบ ซึ่งมีรายละเอียดดังนี้

5.1 สภาพแวดล้อมที่ใช้ในการทดสอบเครื่องมือที่สร้างขึ้น

การทดสอบระบบนี้ได้ทำการติดตั้งระบบดังวิธีการติดตั้งในภาคผนวก ก. โดยใช้สภาพแวดล้อมของฮาร์ดแวร์และซอฟตแวร์ดังต่อนี้

5.1.1 ฮาร์ดแวร์

1. หน่วยประมวลผล อินเทลเพนเทียม 4 3.0 กิกะเฮิร์ต
2. หน่วยความจำ 512 เมกกะไบต์
3. ฮาร์ดดิสก์ 80 กิกะไบต์

5.1.2 ซอฟต์แวร์

1. ระบบปฏิบัติการ วินโดวส์ เอกซ์พี โปรเฟสเซอร์นอล
2. จาวารันไทม์ เวอร์ชัน 1.4.2
3. เจเอฟซียูนิต เวอร์ชัน 2.08
4. ระบบจัดการฐานข้อมูลมายเอสคิวแอล เวอร์ชัน 1.4 พร้อมติดตั้งเจดีบีซี สำหรับมายเอสคิวแอล เวอร์ชัน 3.0.15 เพื่อให้สามารถติดต่อกับโปรแกรมภาษาจาวาได้
5. เครื่องมือวาดแผนภาพยูเอ็มแอล เรชันเนล โรส เอ็นเตอร์ไพรส์ เวอร์ชัน 2002 พร้อมติดตั้งแฟ้มคุณสมบัติ myusecase.pty
6. เครื่องมือแปลงแผนภาพยูเอ็มแอลเป็นแฟ้มเอกซ์เอ็มแอล ยูนิซิส เอ็กซ์เอ็มไอ 1.3 แอด-อิน

5.2 ขั้นตอนการทดสอบเครื่องมือ

ขั้นตอนการทดสอบเครื่องมือที่พัฒนาขึ้นประกอบด้วยขั้นตอนต่างๆ ดังนี้

1. วาดแผนภาพยูเอ็มแอลของระบบที่ต้องการใช้ทดสอบ ในเครื่องมือเรชันเนล โรส โดยยึดตามวิธีการที่ได้อธิบายไว้ในในบทที่ 3 ซึ่งประกอบด้วยแผนภาพยูสเคส คำอธิบายยูสเคส แผนภาพดับเบิลยูเอ็นดี และแผนภาพกิจกรรม
2. แปลงแผนภาพยูเอ็มแอลที่วาดขึ้นเป็นแฟ้มเอกซ์เอ็มแอล โดยใช้เครื่องมือเรชันเนล โรสที่ติดตั้งยูนิซิส โรส เอกซ์เอ็มไอ ทูล
3. ทำการคัดแยกข้อมูลที่ได้จากข้อ 2 จัดเก็บไว้ในฐานข้อมูล โดยเครื่องมือที่สร้างขึ้นเพื่อใช้สร้างตัวดำเนินการและแผนการทดสอบในรูปของแฟ้มเอกซ์เอ็มแอล
4. สร้างกรณีทดสอบและสคริปต์ทดสอบ โดยสคริปต์ทดสอบจะอยู่ในรูปของภาษาจาวาตามกรอบการทำงานเจเอฟซียูนิต เพื่อสนับสนุนความสามารถในการทดสอบแบบอัตโนมัติ
5. ทดสอบเครื่องมือ โดยดำเนินงานเล่นซ้ำสคริปต์ทดสอบกับระบบที่ต้องการทดสอบ และทำการจัดเก็บข้อมูลผลการทดสอบไว้ในฐานข้อมูลด้วย
6. จัดทำรายงานผลการทดสอบ

5.3 ระบบตัวอย่างที่ใช้ในการทดสอบ

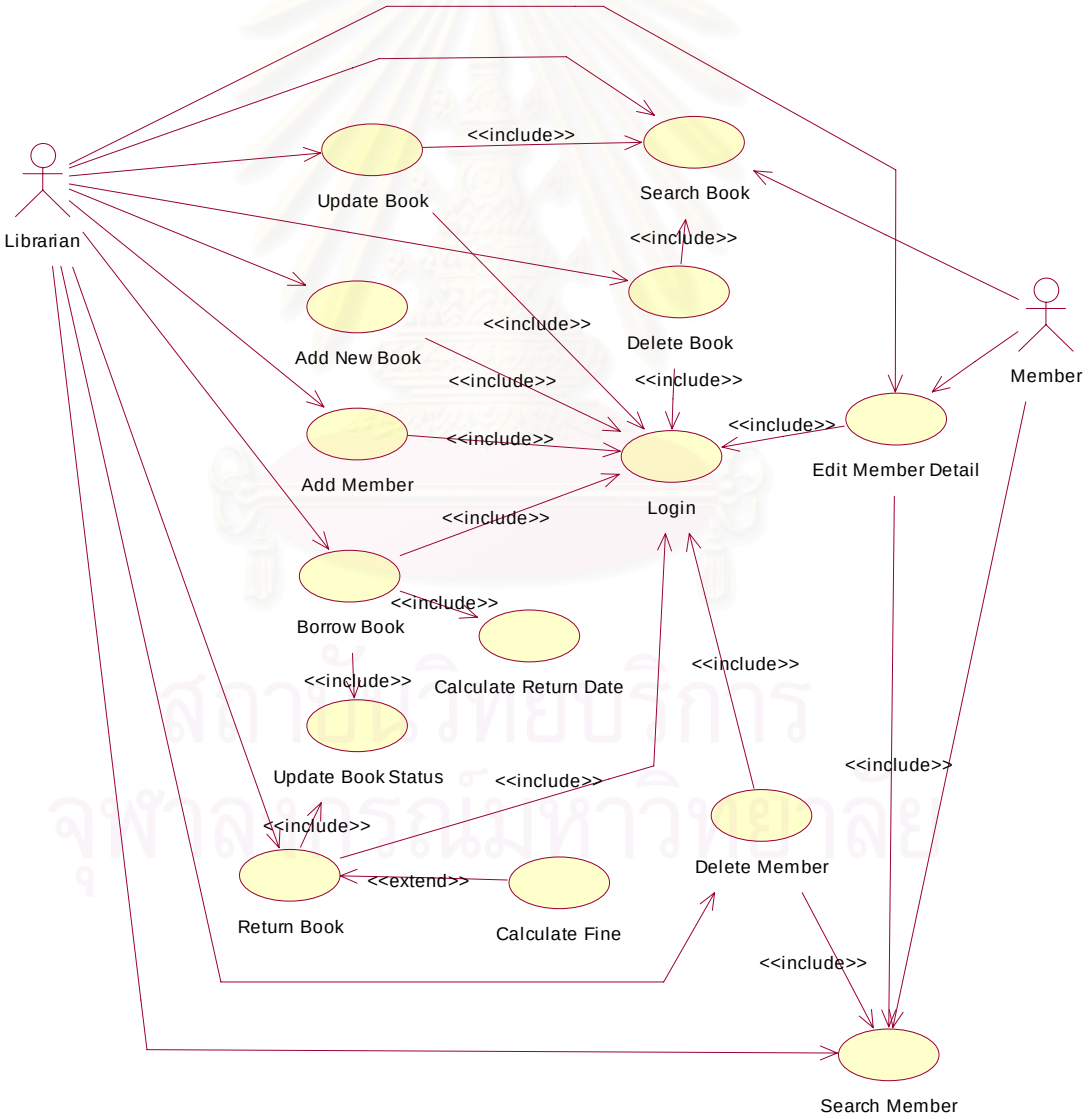
ระบบที่ใช้เป็นตัวอย่างในการทดสอบเครื่องมือ คือ ระบบบริการห้องสมุด ซึ่งการออกแบบหน้าที่การทำงานของระบบดังกล่าว สามารถแทนได้ด้วยแผนภาพยูสเคสจำนวน 14 ยูสเคส ผู้ใช้ของระบบ คือ บรรณารักษ์และสมาชิกห้องสมุด ผู้ออกแบบได้กำหนดภารกิจหลักๆ ของระบบไว้ 10 ภารกิจ ดังนี้คือ

ภารกิจ

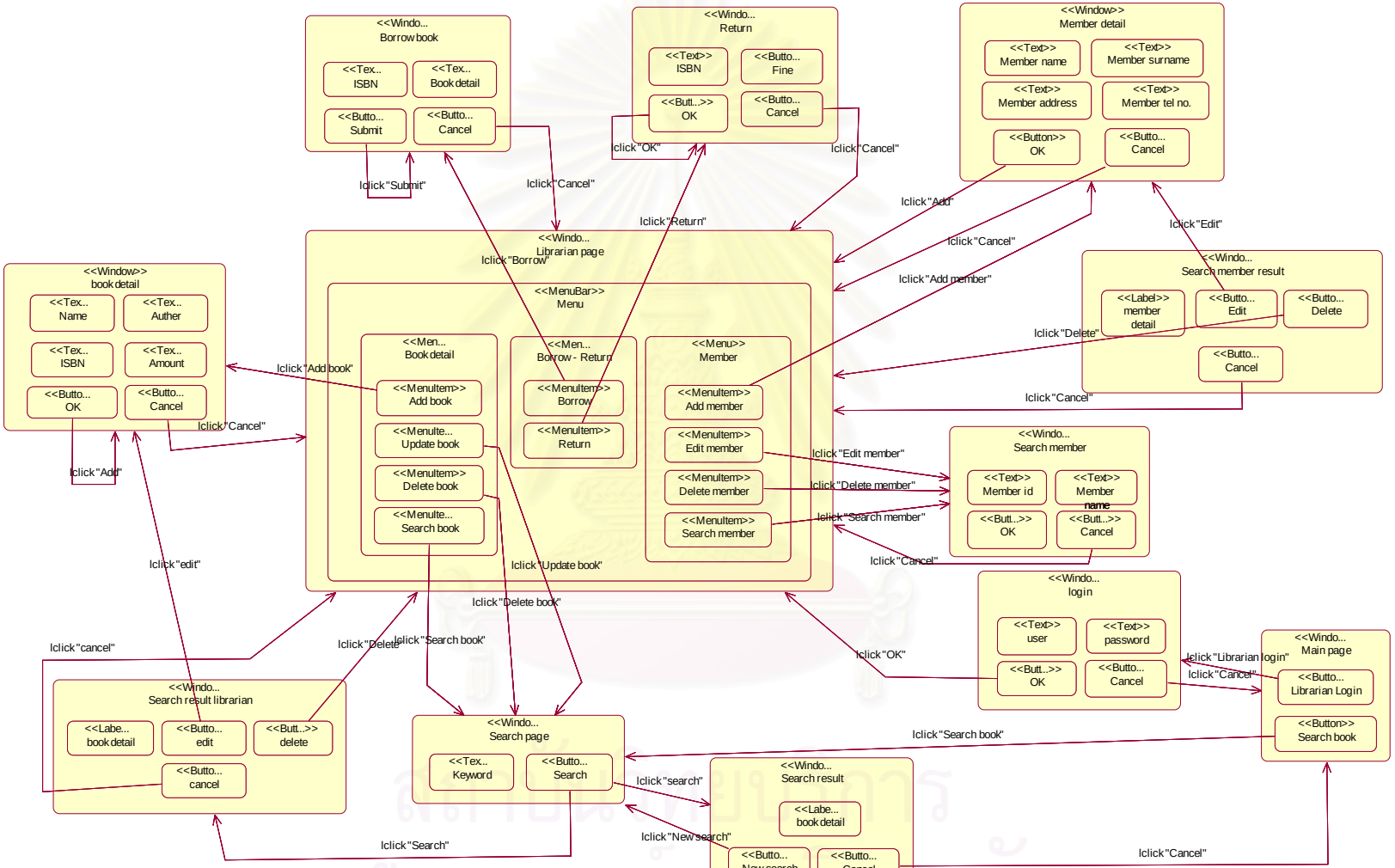
- แก้ไขรายละเอียดข้อมูลหนังสือ (Update Book) โดยบรรณารักษ์
- เพิ่มข้อมูลหนังสือ (Add New Book) โดยบรรณารักษ์
- เพิ่มข้อมูลสมาชิก (Add Member) โดยบรรณารักษ์
- ยืมหนังสือ (Borrow Book) โดยบรรณารักษ์
- คืนหนังสือ (Return Book) โดยบรรณารักษ์
- ค้นหาหนังสือ (Search Book) โดยบรรณารักษ์และสมาชิกห้องสมุด
- ลบข้อมูลหนังสือ (Delete Book) โดยบรรณารักษ์
- ลบข้อมูลสมาชิก (Delete Member) โดยบรรณารักษ์
- แก้ไขข้อมูลสมาชิก (Edit Member Detail) โดยบรรณารักษ์
- ค้นหาสมาชิก (Search Member) โดยบรรณารักษ์

หมายเหตุ: การคืนหนังสืออาจมีการเสียค่าปรับ (Calculate fine) ถ้าสมาชิกนำหนังสือมาคืนหลังกำหนดการ

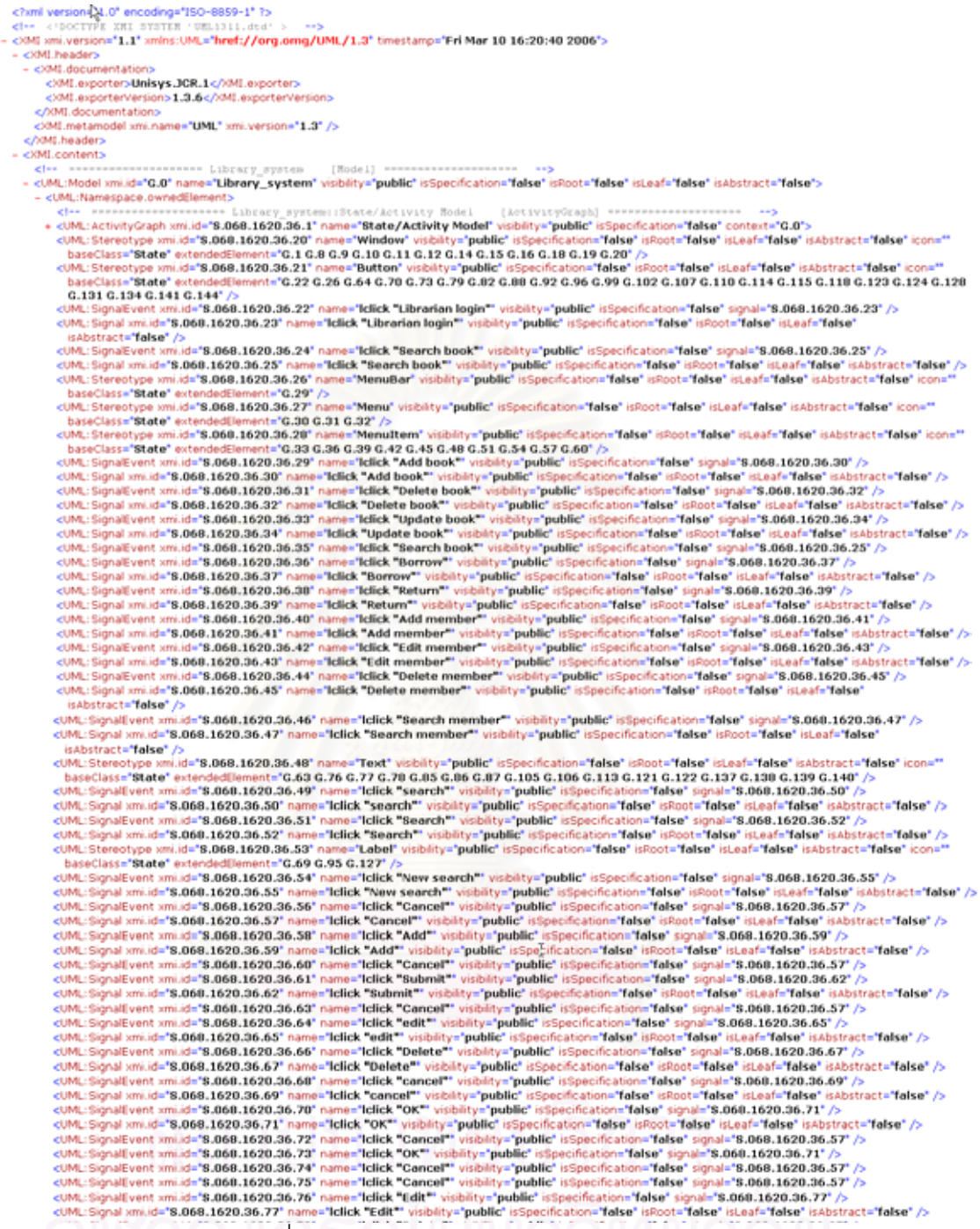
วิธีการเขียนแผนภาพยูเอ็มแอลต่างๆ ที่เกี่ยวข้อง เป็นไปตามรายละเอียดที่อธิบายไว้ในบทที่ 3 โดยแผนภาพยูสเคสของระบบบริการห้องสมุดแสดงดังรูปที่ 22 แผนภาพดับเบิลยูเอ็นดีแสดงดังรูปที่ 23 และรายละเอียดคำอธิบายยูสเคสและแผนภาพกิจกรรมของแต่ละยูสเคสแสดงในภาคผนวก ข หลังจากได้แผนภาพยูเอ็มแอลทั้งหมดที่เกี่ยวข้อง จึงใช้เครื่องมือยูนิซิส เอกซ์เอ็มไอทูล สร้างแฟ้มเอกซ์เอ็มแอลของแผนภาพยูเอ็มแอลของระบบ แสดงดังรูปที่ 24 เพื่อให้เครื่องมือสร้างและทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ที่สร้างขึ้นสามารถนำแฟ้มเอกซ์เอ็มแอลดังกล่าวไปใช้ในการสร้างกรณีทดสอบและสคริปต์ทดสอบได้



รูปที่ 22 แผนภาพยูสเคสของระบบบริการในห้องสมุด



รูปที่ 23 แผนภาพดับเบิลยูเอ็นดีของระบบบริการในห้องสมุด



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- <!DOCTYPE XMI SYSTEM 'UML1311.dtd' > -->
<XMI xmi.version="1.1" xmlns:UML="href://org.omg/UML/1.3" timestamp="Fri Mar 10 16:20:40 2006">
  <XMI.header>
    <XMI.documentation>
      <XMI.exporter>Unisys.JCR.1</XMI.exporter>
      <XMI.exporterVersion>1.3.6</XMI.exporterVersion>
    </XMI.documentation>
    <XMI.metamodel xmi.name="UML" xmi.version="1.3" />
  </XMI.header>
  <XMI.content>
    <!-- ================== Library_system    [Model] ================== -->
    <UML:Model xmi.id="G.0" name="Library_system" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false">
      <UML:Namespace.ownedElement>
        <!-- ================== Library_system::State/Activity Model    [ActivityGraph] ================== -->
        <UML:ActivityGraph xmi.id="S.068.1620.36.1" name="State/Activity Model" visibility="public" isSpecification="false" context="G.0">
          <UML:Stereotype xmi.id="S.068.1620.36.20" name="Window" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" icon=""
            baseClass="State" extendedElement="G.1 G.8 G.9 G.10 G.11 G.12 G.14 G.15 G.16 G.18 G.19 G.20" />
          <UML:Stereotype xmi.id="S.068.1620.36.21" name="Button" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" icon=""
            baseClass="State" extendedElement="G.22 G.26 G.64 G.70 G.73 G.79 G.82 G.88 G.92 G.96 G.99 G.102 G.107 G.110 G.114 G.115 G.118 G.123 G.124 G.128
            G.131 G.134 G.141 G.144" />
          <UML:SignalEvent xmi.id="S.068.1620.36.22" name="Iclick "Librarian login"" visibility="public" isSpecification="false" signal="S.068.1620.36.23" />
          <UML:Signal xmi.id="S.068.1620.36.23" name="Iclick "Librarian login"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false"
            isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.24" name="Iclick "Search book"" visibility="public" isSpecification="false" signal="S.068.1620.36.25" />
          <UML:Signal xmi.id="S.068.1620.36.25" name="Iclick "Search book"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:Stereotype xmi.id="S.068.1620.36.26" name="MenuBar" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" icon=""
            baseClass="State" extendedElement="G.29" />
          <UML:Stereotype xmi.id="S.068.1620.36.27" name="Menu" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" icon=""
            baseClass="State" extendedElement="G.30 G.31 G.32" />
          <UML:Stereotype xmi.id="S.068.1620.36.28" name="MenuItem" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" icon=""
            baseClass="State" extendedElement="G.33 G.36 G.39 G.42 G.45 G.48 G.51 G.54 G.57 G.60" />
          <UML:SignalEvent xmi.id="S.068.1620.36.29" name="Iclick "Add book"" visibility="public" isSpecification="false" signal="S.068.1620.36.30" />
          <UML:Signal xmi.id="S.068.1620.36.30" name="Iclick "Add book"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.31" name="Iclick "Delete book"" visibility="public" isSpecification="false" signal="S.068.1620.36.32" />
          <UML:Signal xmi.id="S.068.1620.36.32" name="Iclick "Delete book"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.33" name="Iclick "Update book"" visibility="public" isSpecification="false" signal="S.068.1620.36.34" />
          <UML:Signal xmi.id="S.068.1620.36.34" name="Iclick "Update book"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.35" name="Iclick "Search book"" visibility="public" isSpecification="false" signal="S.068.1620.36.25" />
          <UML:SignalEvent xmi.id="S.068.1620.36.36" name="Iclick "Borrow"" visibility="public" isSpecification="false" signal="S.068.1620.36.37" />
          <UML:Signal xmi.id="S.068.1620.36.37" name="Iclick "Borrow"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.38" name="Iclick "Return"" visibility="public" isSpecification="false" signal="S.068.1620.36.39" />
          <UML:Signal xmi.id="S.068.1620.36.39" name="Iclick "Return"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.40" name="Iclick "Add member"" visibility="public" isSpecification="false" signal="S.068.1620.36.41" />
          <UML:Signal xmi.id="S.068.1620.36.41" name="Iclick "Add member"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.42" name="Iclick "Edit member"" visibility="public" isSpecification="false" signal="S.068.1620.36.43" />
          <UML:Signal xmi.id="S.068.1620.36.43" name="Iclick "Edit member"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.44" name="Iclick "Delete member"" visibility="public" isSpecification="false" signal="S.068.1620.36.45" />
          <UML:Signal xmi.id="S.068.1620.36.45" name="Iclick "Delete member"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false"
            isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.46" name="Iclick "Search member"" visibility="public" isSpecification="false" signal="S.068.1620.36.47" />
          <UML:Signal xmi.id="S.068.1620.36.47" name="Iclick "Search member"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false"
            isAbstract="false" />
          <UML:Stereotype xmi.id="S.068.1620.36.48" name="Text" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" icon=""
            baseClass="State" extendedElement="G.63 G.76 G.77 G.78 G.85 G.86 G.87 G.105 G.106 G.113 G.121 G.122 G.137 G.138 G.139 G.140" />
          <UML:SignalEvent xmi.id="S.068.1620.36.49" name="Iclick "search"" visibility="public" isSpecification="false" signal="S.068.1620.36.50" />
          <UML:Signal xmi.id="S.068.1620.36.50" name="Iclick "search"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.51" name="Iclick "Search"" visibility="public" isSpecification="false" signal="S.068.1620.36.52" />
          <UML:Signal xmi.id="S.068.1620.36.52" name="Iclick "Search"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:Stereotype xmi.id="S.068.1620.36.53" name="Label" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" icon=""
            baseClass="State" extendedElement="G.69 G.95 G.127" />
          <UML:SignalEvent xmi.id="S.068.1620.36.54" name="Iclick "New search"" visibility="public" isSpecification="false" signal="S.068.1620.36.55" />
          <UML:Signal xmi.id="S.068.1620.36.55" name="Iclick "New search"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.56" name="Iclick "Cancel"" visibility="public" isSpecification="false" signal="S.068.1620.36.57" />
          <UML:Signal xmi.id="S.068.1620.36.57" name="Iclick "Cancel"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.58" name="Iclick "Add"" visibility="public" isSpecification="false" signal="S.068.1620.36.59" />
          <UML:Signal xmi.id="S.068.1620.36.59" name="Iclick "Add"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.60" name="Iclick "Cancel"" visibility="public" isSpecification="false" signal="S.068.1620.36.57" />
          <UML:SignalEvent xmi.id="S.068.1620.36.61" name="Iclick "Submit"" visibility="public" isSpecification="false" signal="S.068.1620.36.62" />
          <UML:Signal xmi.id="S.068.1620.36.62" name="Iclick "Submit"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.63" name="Iclick "Cancel"" visibility="public" isSpecification="false" signal="S.068.1620.36.57" />
          <UML:SignalEvent xmi.id="S.068.1620.36.64" name="Iclick "edit"" visibility="public" isSpecification="false" signal="S.068.1620.36.65" />
          <UML:Signal xmi.id="S.068.1620.36.65" name="Iclick "edit"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.66" name="Iclick "Delete"" visibility="public" isSpecification="false" signal="S.068.1620.36.67" />
          <UML:Signal xmi.id="S.068.1620.36.67" name="Iclick "Delete"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.68" name="Iclick "cancel"" visibility="public" isSpecification="false" signal="S.068.1620.36.69" />
          <UML:Signal xmi.id="S.068.1620.36.69" name="Iclick "cancel"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.70" name="Iclick "OK"" visibility="public" isSpecification="false" signal="S.068.1620.36.71" />
          <UML:Signal xmi.id="S.068.1620.36.71" name="Iclick "OK"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
          <UML:SignalEvent xmi.id="S.068.1620.36.72" name="Iclick "Cancel"" visibility="public" isSpecification="false" signal="S.068.1620.36.57" />
          <UML:SignalEvent xmi.id="S.068.1620.36.73" name="Iclick "OK"" visibility="public" isSpecification="false" signal="S.068.1620.36.71" />
          <UML:SignalEvent xmi.id="S.068.1620.36.74" name="Iclick "Cancel"" visibility="public" isSpecification="false" signal="S.068.1620.36.57" />
          <UML:SignalEvent xmi.id="S.068.1620.36.75" name="Iclick "Cancel"" visibility="public" isSpecification="false" signal="S.068.1620.36.57" />
          <UML:SignalEvent xmi.id="S.068.1620.36.76" name="Iclick "Edit"" visibility="public" isSpecification="false" signal="S.068.1620.36.77" />
          <UML:Signal xmi.id="S.068.1620.36.77" name="Iclick "Edit"" visibility="public" isSpecification="false" isRoot="false" isLeaf="false" isAbstract="false" />
```


รูปที่ 24 แฟ้มเอกซ์เอ็มแอลของระบบบริการห้องสมุด

5.4 ผลการทดสอบ

จากการนำแฟ้มเอกซ์เอ็มแอลของระบบ มาสร้างเป็นแผนการทดสอบและตัวดำเนินการเพื่อใช้ในการสร้างเป็นกรณีทดสอบ แผนการทดสอบที่สร้างได้จากแฟ้มเอกซ์เอ็มแอลของแผนภาพยูเอ็มแอล ที่ได้จากระบบบริการในห้องสมุดโดย 1 แผนแทน 1 ภารกิจของผู้ใช้ซึ่งมีปฏิสัมพันธ์กับระบบได้ จากรูปที่ 22 แผนภาพยูสเคส มียูสเคสจำนวน 10 ยูสเคสที่มีปฏิสัมพันธ์กับบรรณารักษ์

และสมาชิกห้องสมุด ส่วนยูสเคสอีก 4 ยูสเคส ได้แก่ ยูสเคส Login, Calculate Return Date, Calculate Fine และ Update Book Status ไม่มีปฏิสัมพันธ์กับผู้ใช้โดยตรงจึงไม่นับเป็นภารกิจของระบบ ผลการทดสอบในที่นี้ จะขอยกตัวอย่างแสดงผลลัพธ์การสร้างกรณีทดสอบของภารกิจ Borrow Book และการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้โดยใช้กรณีทดสอบที่สร้างขึ้น

การสร้างกรณีทดสอบจำเป็นต้องสร้างแผนการทดสอบและตัวดำเนินการของภารกิจ ตัวอย่างแผนการทดสอบและตัวดำเนินการของภารกิจ Borrow Book แสดงในรูปที่ 25 และ 26 ตามลำดับ

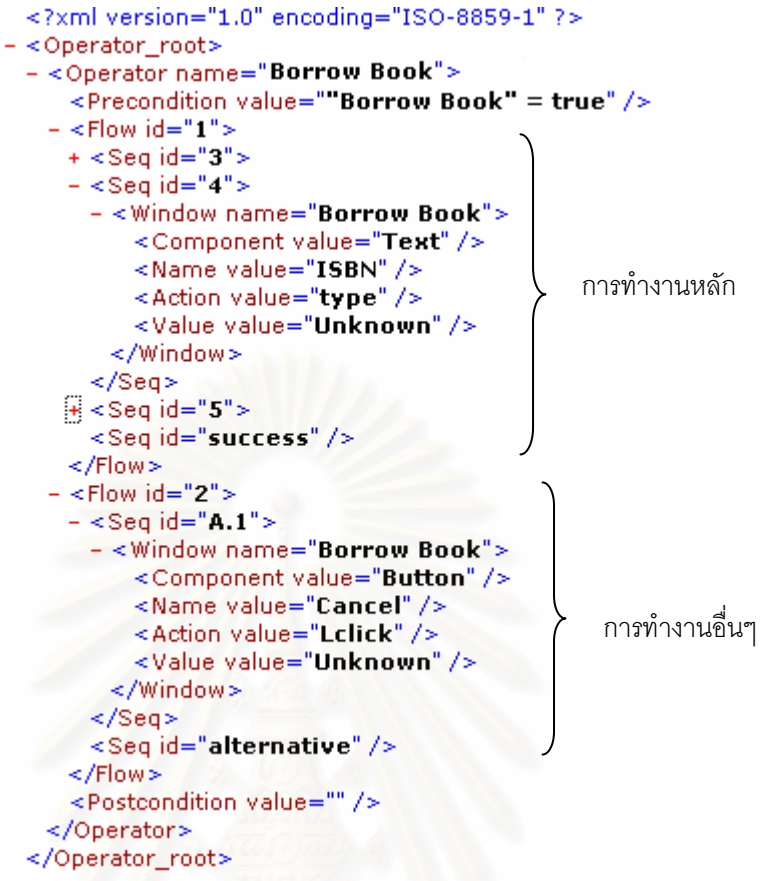
แผนที่ 9 ในรูปที่ 25 ประกอบด้วยตัวดำเนินการ Login และ Borrow Book เนื่องจากยูสเคส Update Book Status และ Calculate Return Date ไม่มีปฏิสัมพันธ์กับผู้ใช้ ดังนั้นจึงไม่สามารถสร้างตัวดำเนินการของยูสเคสดังกล่าวได้

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Plan_root>
  + <Plan name="1">
  + <Plan name="2">
  + <Plan name="3">
  + <Plan name="4">
  + <Plan name="5">
  + <Plan name="6">
  + <Plan name="7">
  + <Plan name="8">
  - <Plan name="9">
      <Inc_TOperator value="Login" />
      <TOperator value="Borrow book" />
    </Plan>
  + <Plan name="10">
  </Plan_root>
```

แผนที่ 9 ประกอบด้วยตัวดำเนินการ Login และBorrow Book

รูปที่ 25 แผนการทดสอบของภารกิจ Borrow Book

ตัวดำเนินการของระบบบริการในห้องสมุด สร้างจากแฟ้มเอกซ์เอ็มแอลของระบบโดยจะถูกสร้างขึ้นพร้อมกันกับแผนการทดสอบอย่างอัตโนมัติ แต่ละตัวดำเนินการจะถูกสร้างให้มีขั้นตอนการทำงานเหมือนกับในคำอธิบายยูสเคสที่ตรงกัน แต่เลือกเฉพาะส่วนที่มีปฏิสัมพันธ์กับผู้ใช้มาเขียนเป็นการกระทำ รูปที่ 26 แสดงตัวอย่างตัวดำเนินการ Borrow Book ประกอบด้วย 2 การทำงานคือ การทำงาหลักซึ่ง อีลีเมนต์ <Flow> มีค่าของแอตทริบิวท์ id = 1 และหากค่า id มากกว่า 1 จะเป็นการทำงานอื่นๆ




```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Operator_root>
  - <Operator name="Borrow Book">
      <Precondition value=""Borrow Book" = true" />
    - <Flow id="1">
      + <Seq id="3">
      - <Seq id="4">
        - <Window name="Borrow Book">
            <Component value="Text" />
            <Name value="ISBN" />
            <Action value="type" />
            <Value value="Unknown" />
          </Window>
        </Seq>
      + <Seq id="5">
        <Seq id="success" />
      </Flow>
    - <Flow id="2">
      - <Seq id="A.1">
        - <Window name="Borrow Book">
            <Component value="Button" />
            <Name value="Cancel" />
            <Action value="Lclick" />
            <Value value="Unknown" />
          </Window>
        </Seq>
        <Seq id="alternative" />
      </Flow>
      <Postcondition value="" />
    </Operator>
</Operator_root>
```

รูปที่ 26 ตัวดำเนินการ Borrow Book

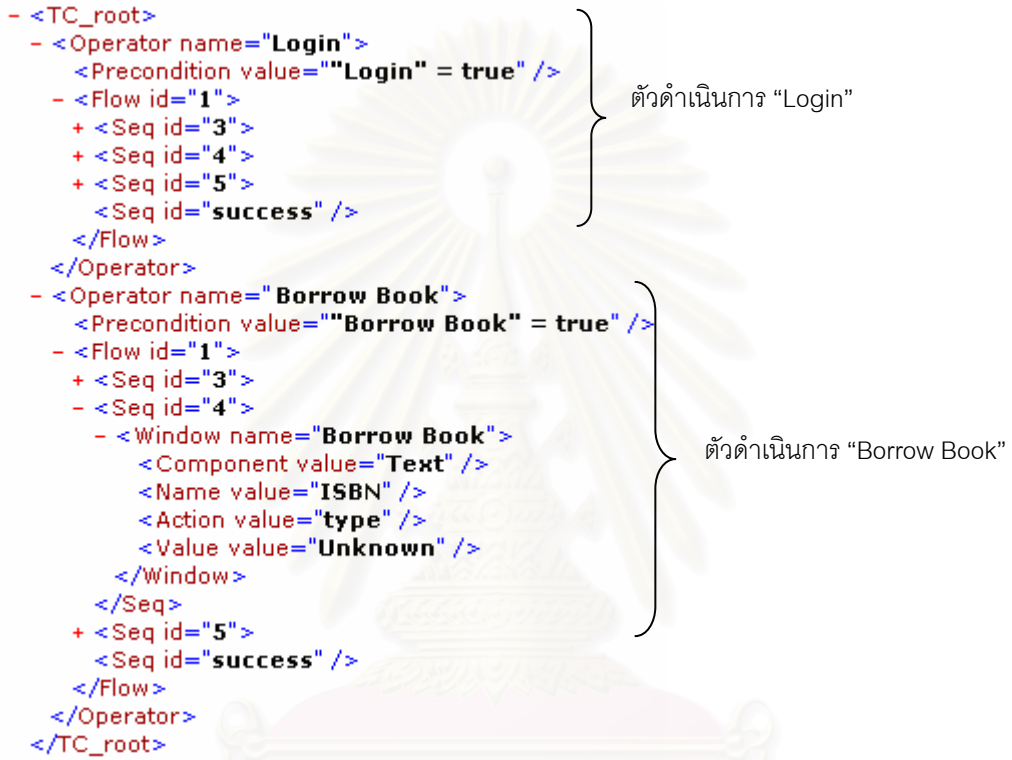
เมื่อได้แผนการทดสอบและตัวดำเนินการแล้ว ขั้นต่อไปจะทำการสร้างกรณีทดสอบ โดยนำ อีลีเมนต์ <Flow> ของตัวดำเนินการมาแทนที่ตัวดำเนินการในแผนการทดสอบ รูปแบบจะเป็นลักษณะของอีลีเมนต์ <Flow> ของตัวดำเนินการเรียงต่อกันอย่างมีลำดับ เมื่อได้กรณีทดสอบแล้วจะทำการแปลงกรณีทดสอบที่ได้เป็นสคริปต์ทดสอบ แสดงตัวอย่างกรณีทดสอบและสคริปต์ทดสอบ ดังรูปที่ 27 และ 28

รูปที่ 27 แสดงตัวดำเนินการ Login และตัวดำเนินการ Borrow Book เพียงบางส่วน โดยเป็นการทำงานหลักของทั้ง 2 ตัวดำเนินการซึ่ง id มีค่าเท่ากับ 1

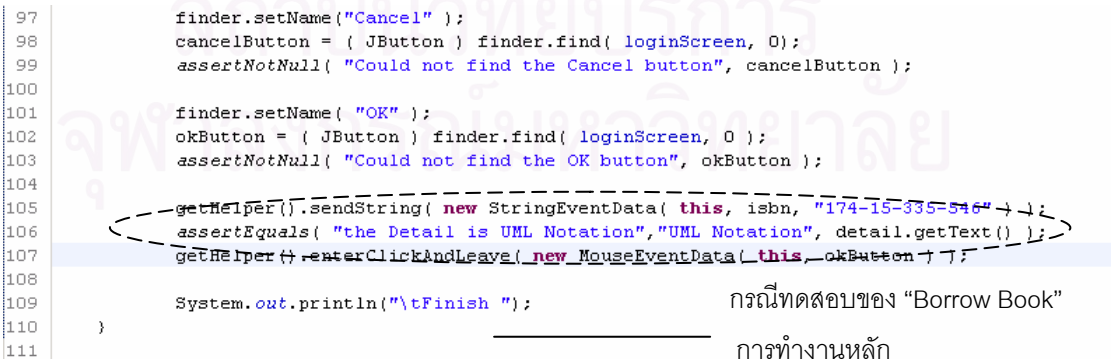
รูปที่ 28 แสดงกรณีทดสอบที่อยู่ในรูปของสคริปต์ทดสอบ ซึ่งกรณีทดสอบจะอยู่ในรูปของเมธอด เช่น public void Login1_BorrowBook1() หมายถึงกรณีทดสอบของตัวดำเนินการ Login ซึ่งเป็นการทำงานหลัก และตัวดำเนินการ Borrow Book ที่เป็นการทำงานหลักเช่นเดียวกัน

นอกจากนี้ ผู้ทดสอบสามารถใส่ค่าในจุดตรวจสอบของสคริปต์ทดสอบ เพื่อให้โปรแกรมตรวจสอบค่าที่ปรากฏบนหน้าจอเป็นค่าที่คาดหวังหรือไม่ ตัวอย่าง จุดตรวจสอบในรูปที่ 28

บรรทัดที่ 106 แสดงทำการแทรกจุดตรวจสอบโดยผู้ออกแบบการทดสอบ assertEquals("the Detail is UML Notation","UML Notation", detail.getText()); มีความหมายคือค่าของกล่องข้อความ detail จะต้องมีค่าเท่ากับ “UML Notation” หากไม่เท่ากันโปรแกรมจะจบการทดสอบในกรณีทดสอบที่กำลังดำเนินงานอยู่ แต่จะยังคงทดสอบกรณีทดสอบถัดไปจนจบการทดสอบทั้งหมดและแจ้งข้อความผิดพลาดขึ้นเมื่อการทดสอบทั้งหมดเสร็จสิ้นแล้ว



รูปที่ 27 กรณีทดสอบในรูปแบบแฟ้มเอกซ์เอ็มแอล



รูปที่ 28 กรณีทดสอบในรูปของสคริปต์ทดสอบ

เมื่อทำการสร้างกรณีทดสอบจากข้อมูลข้างต้นแล้ว ผลการสร้างกรณีทดสอบ แสดงดังตารางที่ 33 ตัวอย่างเช่น ภารกิจ Borrow Book เกิดจากการทำงานร่วมกันของยูสเคส 4 ยูสเคส ซึ่งต้องการกรณีทดสอบทั้งหมดเป็นจำนวน 5 กรณี ดังต่อไปนี้

1. Login1 BorrowBook1 เป็นการทำงานหลักของทั้ง 2 ตัวดำเนินการ

2. Login1 BorrowBook2 เป็นการทำงานหลักของตัวดำเนินการ Login และการทำงานอื่นๆ ของตัวดำเนินการ Borrow Book

3. Login2 BorrowBook1 เป็นการทำงานอื่นๆ ของตัวดำเนินการ Login ซึ่งทำงานสำเร็จ และการทำงานหลักของตัวดำเนินการ Borrow Book

4. Login2 BorrowBook1 เป็นการทำงานอื่นๆ ของตัวดำเนินการ Login และการทำงานหลักของตัวดำเนินการ Borrow Book

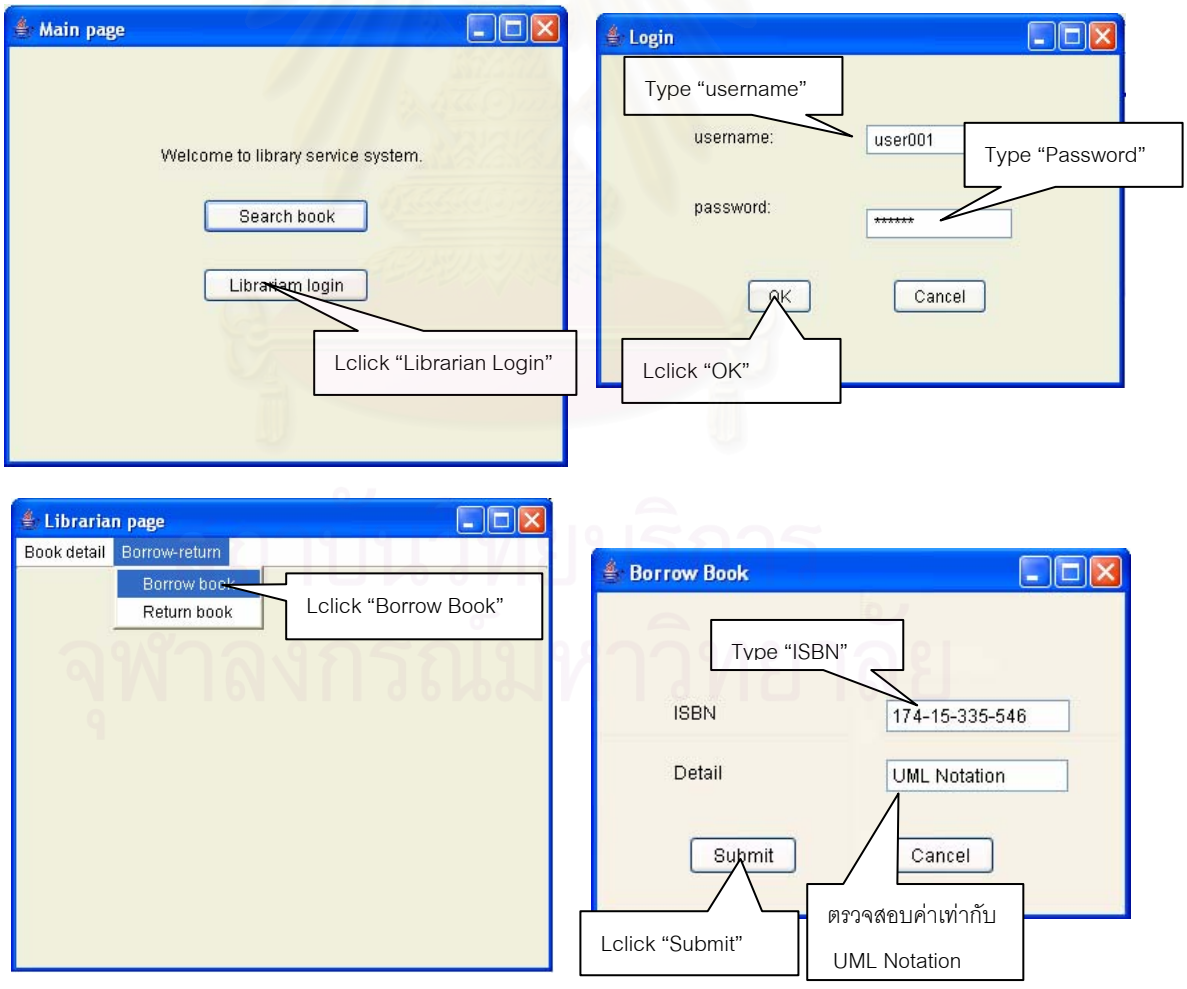
5. Login3 ไม่สามารถทำงานในตัวดำเนินการ Borrow Book ได้เนื่องจากตัวดำเนินการ Login ทำงานไม่สำเร็จจึงจบการทำงานในส่วนของตัวดำเนินการนี้

จะเห็นได้ว่าไม่มีตัวดำเนินการของยูสเคส Calculate Date Return และ Update Book Status เนื่องจากยูสเคสทั้งสองไม่มีปฏิสัมพันธ์กับผู้ใช้จึงไม่สามารถสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ของยูสเคสทั้งสองได้

ตารางที่ 33 ผลการสร้างกรณีทดสอบของระบบบริการในห้องสมุด

ลำดับ	ภารกิจ	จำนวนกรณีทดสอบ	ยูสเคสที่ใช้
1	Add New book	5	Login, Add New book
2	Delete book	9	Login, Search book, Delete book
3	Update book	9	Login, Search book, Update book
4	Search book	3	Search book
5	Add member	5	Login, Add member
6	Delete member	9	Login, Search member, Delete member
7	Edit member detail	9	Login, Search member, Edit member detail
8	Search member	3	Search member
9	Borrow book	5	Login, Borrow book, Calculate return date, Update book status
10	Return book	9	Login, Return book, Calculate fine, Update book status

เมื่อทำการสร้างสคริปต์ทดสอบเรียบร้อยแล้ว นำมาทดสอบกับโปรแกรมบริการในห้องสมุดที่ได้พัฒนาขึ้น ซึ่งเป็นโปรแกรมภาษาจาวาและใช้คอมโพเนนท์สวิงในการพัฒนา โดยการทดสอบทำการทดสอบในลักษณะเดียวกันกับส่วนเล่นซ้ำในเครื่องมือบันทึกและเล่นซ้ำ กล่าวคือ สร้างการกระทำส่งให้กับคอมโพเนนท์ในหน้าจอ จากรูปที่ 29 เป็นการทดสอบกรณีทดสอบที่ 1 คือ ทำการ Login สำเร็จและยืมหนังสือสำเร็จ Login1 Borrow Book1 การทำงานมีดังนี้ เริ่มจากการคลิกปุ่มเม้าส์ปุ่มซ้ายบนปุ่ม Librarian Login เข้าสู่หน้าจอ Login ทำการพิมพ์ username และ Password และคลิกเม้าส์ปุ่มซ้ายบนปุ่ม OK เพื่อเข้าสู่หน้าจอ Librarian Page จากนั้นกดเม้าส์ปุ่มซ้ายบนเมนู Borrow Book จะเข้าสู่หน้าจอ Borrow Book ทำการพิมพ์ ISBN และกดปุ่ม Submit จึงจบการทำสอบ 1 กรณีทดสอบ หากการทำงานไม่มีปัญหา เช่น ไม่สามารถหาคอมโพเนนท์ได้ หรือค่าที่ได้ไม่เท่ากับจุดตรวจสอบที่กำหนด ก็จะเกิดความผิดพลาดและไม่ผ่านการทดสอบในกรณีทดสอบนั้นๆ เมื่อทดสอบครบทุกกรณีทดสอบแล้วจะทำการเก็บผลการทำสอบลงในฐานข้อมูลและแสดงรายงานผลการทดสอบ ดังรูปที่ 30



รูปที่ 29 การนำสคริปต์ทดสอบมาดำเนินงานกับโปรแกรม

รายงาน โครงการ :

โครงการ Librarian_Service รหัส 12				วันที่ 09-03-2006	
ผู้ทำการทดสอบ เพชรน้ำค้าง วงษ์ศุทธิภากร		ผู้ออกแบบกรณีทดสอบ เพชรน้ำค้าง วงษ์ศุทธิภากร			
ลำดับ	ชื่อกรณีทดสอบ	ผ่าน	ไม่ผ่าน	เวลาในการทดสอบ (ms)	หมายเหตุ
1	Login1_BorrowBook1	/	-	1352	-
2	Login1_BorrowBook2	/	-	1014	-
3	Login2_BorrowBook1	/	-	2144	-
4	Login2_BorrowBook2	/	-	2030	-
5	Login3	/	-	895	-
	รวม	5	0	7435	

รูปที่ 30 รายงานผลการทดสอบของแผน Borrow Book

5.5 สรุปผลการทดสอบ

จากการทดสอบเครื่องมือสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากยูเอ็มแอล และทดสอบกับระบบบริการในห้องสมุด ผลการทดสอบพบว่า เครื่องมือสามารถสร้างกรณีทดสอบได้ครอบคลุมทุกภารกิจของระบบ ตามการทำงานที่ปรากฏในแผนภาพยูสเคส และสามารถนำมาใช้ทดสอบกับระบบที่ได้สร้างตามวิธีการที่กำหนดได้จริง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้นำเสนอวิธีการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากยูเอ็มแอล รวมทั้งพัฒนาเครื่องมือสำหรับสร้างและทดสอบกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากยูเอ็มแอล ซึ่งจะช่วยลดทรัพยากรในขั้นตอนการทดสอบลง ผลลัพธ์จากการสร้างกรณีทดสอบจากยูเอ็มแอลจะได้กรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ที่มีปริมาณลดลงจากการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้แบบเดิม เช่น วิธีการที่ใช้เครื่องสถานะ เป็นต้น แต่กรณีทดสอบที่ได้ยังคงครอบคลุมทุกหน้าที่การทำงานที่คาดหวังของระบบ ทั้งยังสามารถสร้างกรณีทดสอบได้หลังการออกแบบระบบ

นอกจากนี้ ผู้วิจัยยังได้พัฒนาเครื่องมือกึ่งอัตโนมัติเพื่อสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากแผนภาพยูเอ็มแอล โดยกรณีทดสอบจะอยู่ในรูปแบบของแฟ้มเอกซ์เอ็มแอล ซึ่งสามารถนำมาใช้ทดสอบกับซอฟต์แวร์ส่วนต่อประสานกราฟิกกับผู้ใช้ที่พัฒนาขึ้นด้วยภาษาจาวาและใช้คอมโพเนนท์ของสวิง และกรณีทดสอบดังกล่าวยังสามารถนำกลับมาใช้ทดสอบได้ใหม่ เครื่องมือที่พัฒนาขึ้นนอกจากสามารถใช้สร้างกรณีทดสอบจียูไอแล้ว ยังสามารถใช้ทดสอบโดยวิธีเล่นซ้ำกรณีทดสอบที่จัดเก็บไว้เป็นสคริปต์ทดสอบ เครื่องมือที่ใช้ทดสอบนี้จะมีความสามารถคล้ายกับส่วนเล่นซ้ำของเครื่องมือบันทึกและเล่นซ้ำ ดังต่อไปนี้

1. ค้นหาคอมโพเนนท์ตามชื่อที่ตั้งไว้
2. สร้างเหตุการณ์ให้แก่คอมโพเนนท์ทั้งจากเม้าส์และแป้นอักขระ เช่น การคลิก การพิมพ์ข้อมูล เป็นต้น
3. ตรวจสอบได้ว่าเหตุการณ์ที่เกิดขึ้น ทำให้ส่วนต่อประสานกราฟิกกับผู้ใช้ปรากฏถูกต้องหรือไม่ เช่น เมื่อกดปุ่มแล้วปรากฏหน้าจอตรงตามข้อกำหนดหรือไม่
4. ตรวจสอบค่าที่คาดหวังในคอมโพเนนท์จากเหตุการณ์ที่สร้างได้ เช่น ค่าในกล่องข้อความ มีค่าตรงกับข้อกำหนดหรือไม่

6.2 ข้อจำกัดและแนวทางการวิจัยต่อ

ในงานวิจัยนี้สร้างเครื่องมือซึ่งสนับสนุนการทดสอบระบบที่สร้างขึ้นโดยภาษาจาวา และใช้คอมโพเนนท์สวิง แต่ยังไม่สามารถทดสอบได้ครบทุกคอมโพเนนท์ของสวิง โดยยังไม่สามารถทดสอบคอมโพเนนท์สวิงที่มีความซับซ้อนซึ่งระบุค่าของข้อมูลได้ยาก ได้แก่ คอมโพเนนท์ประเภทสัญรูปหรือแสดงกราฟิก คอมโพเนนท์ประเภทแถบเลื่อน คอมโพเนนท์ตาราง คอมโพเนนท์ประเภทหน้าจอเลือกสี เป็นต้น ดังนั้นอาจจะขยายความสามารถโดยการปรับวิธีการให้สามารถใช้งานได้กับส่วนต่อประสานกราฟิกกับผู้ใช้ทุกประเภท และพัฒนาจากโปรแกรมภาษาอื่นๆ อีกทั้งกรณีทดสอบที่สร้างขึ้นโดยวิธีการนี้ ครอบคลุมทุกภารกิจที่ปรากฏในแผนภาพยูเอ็มแอลเท่านั้น ดังนั้นข้อผิดพลาดอาจเกิดขึ้นนอกเหนือจากกรณีทดสอบที่สร้างขึ้น

แนวทางการวิจัยต่อ เพื่อให้วิธีการนี้สามารถใช้งานได้อย่างกว้างขวาง และมีประสิทธิภาพมากขึ้น ได้แก่

- การสร้างกรณีทดสอบแบบอัตโนมัติจากแฟ้มการออกแบบ โดยไม่ต้องให้ผู้ออกแบบการทดสอบกรอกค่าที่ใช้ในการทดสอบลงในกรณีทดสอบ โดยอาจเป็นการใช้แฟ้มนำเข้าเป็นชุดของค่าที่ต้องการทดสอบแทน
- การสร้างแผนภาพกิจกรรมแบบอัตโนมัติจากคำอธิบายยูสเคส
- สร้างกรณีทดสอบเพิ่มเติมจากรายละเอียดที่ปรากฏในแผนภาพยูเอ็มแอล หรือกรณีทดสอบที่เป็นการทำงานร่วมกันของหลายภารกิจ
- การสร้างเครื่องมือทดสอบให้รองรับระบบงานที่ใช้ส่วนต่อประสานกราฟิกกับผู้ใช้แบบอื่น รวมทั้งระบบงานที่สร้างขึ้นโดยใช้ภาษาโปรแกรมอื่นๆ
- รูปแบบของรายงานผลการทดสอบ หากเกิดความผิดพลาด อาจมีการบันทึกหน้าจอที่เกิดความผิดพลาดเก็บไว้ เพื่อความสะดวกในการตรวจสอบและแก้ไข

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. A. Memon, M. Pollack and M. Soffa. Plan Generation for GUI Testing. Proc. 5th International Conference on Artificial Intelligence Planning and Scheduling 2000 : 226-235.
2. C. Phillips, E. Kemp and S. Mei Kek. Extending UML Use Case Modelling to Support Graphical User Interface Design. Proc. 13th Australian Conference on Software Engineering 2001 : 48-57.
3. D. J. Kasik and H. G. George. Toward automatic generation of novice user test scripts. Proceedings of the 1996 Conference on Human Factors in Computing Systems 1996 : 244-251.
4. T. Dabóczi, I. Kollár, G. Simon, and T. Megyeri. How to test graphical user interfaces. IEEE Instrumentation & Measurement Magazine 2003 : 27-33.
5. R. K. Shehady and D. P. Siewiorek. A method to automate user interface testing using variable finite state machines. Proceedings of The Twenty-Seventh Annual International Symposium on Fault-Tolerant Computing (FTCS'97) 1997 : 80-88.
6. A. M. Memon, M. E. Pollack, and M. L. Soffa. Using a goal-driven approach to generate test cases for GUIs. Proceedings of The Twenty-First International Conference on Software Engineer 1999 : 257-266.
7. T. C. Nicholas Graham. A method for the formal testing of program visualization tools. Proceedings of the Fourth Workshop on Program Comprehension 1996 : 45-54.
8. An Adopted Formal Specification of the Object Management Group, Inc. OMG Unified Modeling Language Specification. (n.p.) : Object Management Group, 2003.
9. A. Dennis, B. Wixom and D. Tagarden. System Analysis & Design An Object-Oriented Approach with UML. (n.p.) : John Wiley & Sons, 2002.
10. M. Page-Jones. Fundamentals of Object-Oriented Design in UML. (n.p.) : Addison Wesley Professional, 2000.

11. ศะรัณ เที่ยงสกุล. ระบบอัตโนมัติช่วยทดสอบหน้าที่ของซอฟต์แวร์ส่วนติดต่อผู้ใช้แบบกราฟฟิกส์. โครงงานมหาบัณฑิต วิศวกรรมซอฟต์แวร์ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2546.

12. R. Shehady and D. Siewiorek. A Method to Automate User Interface Testing Using Variable Finite State Machines. 27th Annual International Symposium, Fault-Tolerant Computing. 1997 : 80 - 88.

13. A. Cockburn. Writing Effective Use Cases. (n.p.) : Addison-Wesley, 2001

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

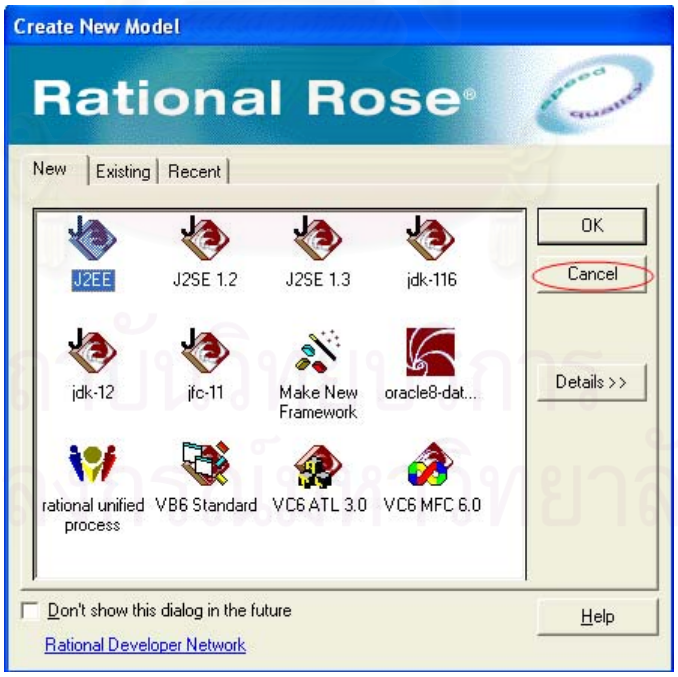
ภาคผนวก ก.

การวาดและแปลงแผนภาพยูเอ็มแอลเป็นแฟ้มเอกซ์เอ็มแอล โดยเครื่องมือเรชันเนล โรส

การวาดแผนภาพยูเอ็มแอลสำหรับใช้ในการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ มีความสำคัญอย่างมากเนื่องจากการสร้างกรณีทดสอบนั้นมีบางส่วนที่เป็นอัตโนมัติ คือ ส่วนที่สร้างข้อมูลตัวดำเนินการและแผนการทดสอบ ดังนั้นข้อมูลนำเข้าหรือแฟ้มเอกซ์เอ็มแอลของแผนภาพยูเอ็มแอลจะต้องมีความถูกต้อง ครบถ้วนสมบูรณ์ มิฉะนั้นการสร้างกรณีทดสอบอาจเกิดความผิดพลาดขึ้นได้ โดยรายละเอียดที่จำเป็นในการวาดแผนภาพมีดังนี้

ขั้นตอนเริ่มต้นก่อนการวาดแผนภาพ

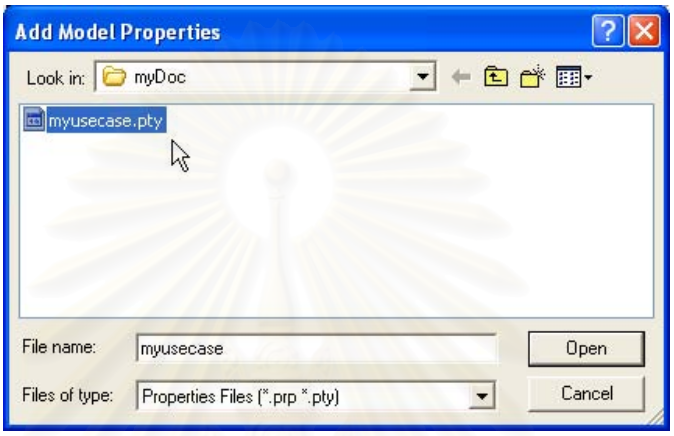
เมื่อเริ่มเปิดเครื่องมือเรชันเนล โรส จะพบกับหน้าต่างเลือกรูปแบบในการวาดแผนภาพ ซึ่งเป็นรูปแบบที่รองรับการสร้างระบบตามประเภทที่กำหนดไว้ เช่น ระบบที่สร้างด้วยภาษาจาวารูปแบบต่างๆ ระบบที่สร้างด้วยอาร์ยูพี เป็นต้น ในการเขียนแผนภาพในงานวิจัยนี้ไม่เลือกรูปแบบใดๆ กดปุ่ม Cancel ดังรูปที่ 31 เพื่อเข้าสู่หน้าจอวาดภาพแบบปกติ



รูปที่ 31 หน้าจอเลือกรูปแบบการวาดแผนภาพยูเอ็มแอล

จากนั้นนำเข้าแฟ้มคุณสมบัติ หรือพรอบเพอร์ตีแฟ้ม (Properties file or .pty) ซึ่งทำหน้าที่เติมเพิ่มคุณสมบัติของยูสเคสให้สามารถเขียนคำอธิบายยูสเคสลงในเครื่องมือได้ โดยการเลือก

เมนูตามลำดับ ดังนี้ Tools -> Model Properties -> Add จะปรากฏหน้าต่างให้เลือกแฟ้ม .pty โดยเลือกแฟ้ม .pty จากสารบบของเครื่องมือสร้างและทดสอบกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้บนเครื่องคอมพิวเตอร์นั้น ดังรูปที่ 32 เมื่อเลือกเปิดแฟ้มคุณสมบัติแล้วเครื่องมือเรชันเนล โรสจะทำการติดตั้งคุณสมบัติที่ระบุอยู่ในแฟ้มนั้นลงในเครื่องมือสำหรับการวาดแผนภาพต่อไป



รูปที่ 32 หน้าจอเลือกแฟ้มคุณสมบัติ

แฟ้มคุณสมบัติที่ใช้ในงานวิจัยนี้ สร้างขึ้นโดยทำการเพิ่มความสามารถของเครื่องมือให้สามารถเขียนอธิบายยูสลงในแผนภาพ โดยคำอธิบายยูสเคสที่ได้เพิ่มเข้าไปมีดังนี้ หมายเลขยูสเคส เงื่อนไขก่อนหน้า เงื่อนไขตามหลัง การทำงานหลัก และการทำงานอื่นๆ ซึ่งแฟ้มดังกล่าวมีรายละเอียด ดังนี้

```
(object Petal       version   45   written  "MyAddin v1.0" charSet 0)
(list Attribute_Set
(object Attribute tool   "myUseCase" name   "propertyID" value   "809135966")
#  my Use Case Property
(object Attribute   tool       "myUseCase" name          "default__UseCase"
value     (list Attribute_Set
(object Attribute tool   "myUseCase"  name  "usecaseID" value  "")
(object Attribute tool    "myUseCase" name  "Pre-condition" value "")
(object Attribute tool    "myUseCase" name  "Post-condition"  value  "")
(object Attribute tool    "myUseCase"  name  "MainFlow"  value  "")
(object Attribute tool    "myUseCase"  name  "AlternateFlow"  value  "")
)
)
```

#name "default__Association"

#name "default__Attribute"

#name "default__Generalization")

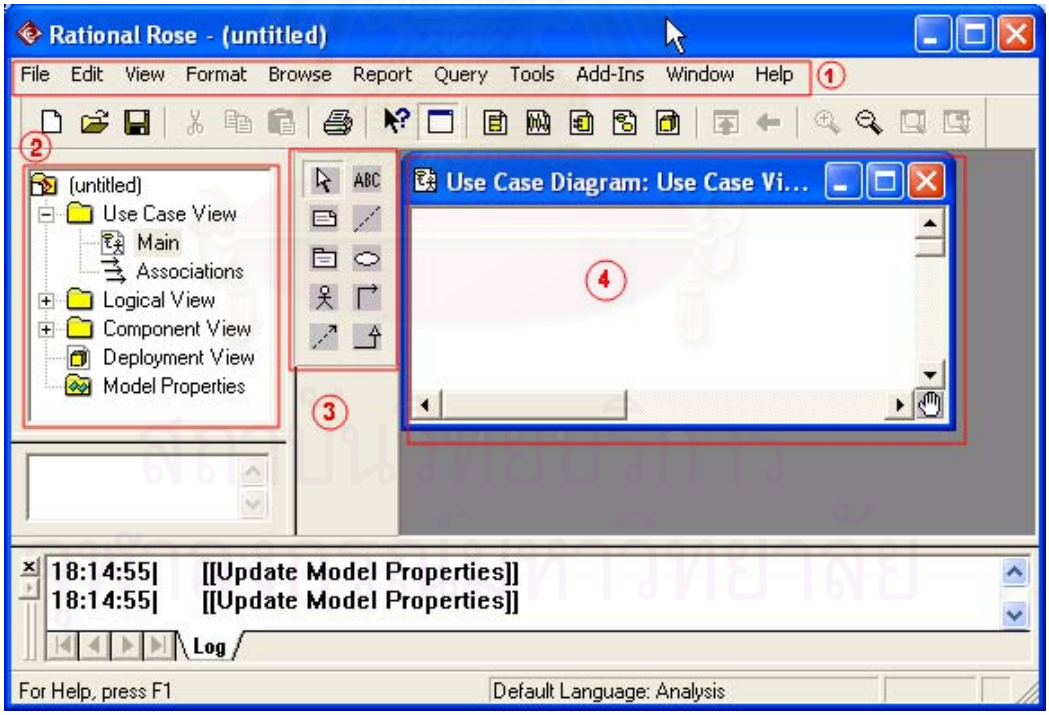
หน้าต่างของเครื่องมือมีส่วนประกอบหลัก 4 ส่วนที่ต้องใช้งาน ดังรูปที่ 33 แสดงส่วนประกอบของเครื่องมือ โดยจะอธิบายการใช้งานตามหมายเลขที่ปรากฏ คือ

หมายเลข 1 แถบเมนู ใช้เลือกเมนูต่างๆ ในเครื่องมือ เช่น สร้างใหม่ บันทึก หรือการแปลงเป็นแฟ้มเอกซ์เอ็มแอล เป็นต้น

หมายเลข 2 มุมมองต่างๆ โดยใช้ 2 มุมมอง คือ มุมมองยูสเคส (Use Case View) เพื่อใช้ในการวาดแผนภาพยูสเคส คำอธิบายยูสเคส และมุมมองโลจิคอล (Logical View) ใช้ในการวาดแผนภาพดับเบิลยูเอ็นดี แผนภาพกิจกรรม

หมายเลข 3 แถบเครื่องมือสำหรับวาดแผนภาพแต่ละแผนภาพ โดยจะเปลี่ยนเครื่องมือไปตามแผนภาพที่จะวาด

หมายเลข 4 หน้าต่างวาดแผนภาพ จะเป็นหน้าต่างว่างสำหรับวาดแผนภาพต่างๆ

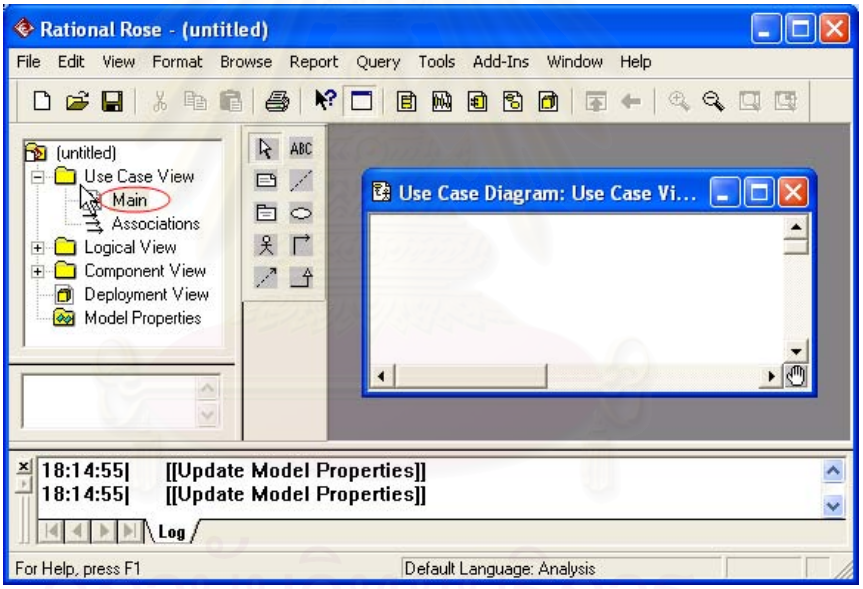


รูปที่ 33 หน้าต่างของเครื่องมือเรชันเนล โรส

การวาดแผนภาพยูสเคส

1. การสร้างแผนภาพยูสเคส

แผนภาพยูสเคสที่ใช้ในการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ จะมีเพียง 1 แผนภาพ ซึ่งวาดอยู่ในมุมมองยูสเคส เริ่มต้นโดยการคลิกที่เครื่องหมาย + หน้าสารบบมุมมองยูสเคส ภายในจะมีสัญลักษณ์แผนภาพยูสเคสชื่อเมน ดังรูปที่ 34 ให้ทำการเปลี่ยนชื่อเป็นชื่อระบบที่ต้องการ จากนั้นดับเบิลคลิกเพื่อเปิดหน้าต่างว่างขึ้น ด้านซ้ายมือของหน้าต่างจะปรากฏแถบเครื่องมือในการวาดแผนภาพยูสเคส ซึ่งเครื่องมือที่ใช้วาดแผนภาพ ประกอบด้วย 1) ผู้ใช้ 2) ยูสเคส 3) เส้นความสัมพันธ์ ความสัมพันธ์แบบธรรมดา อินคลูด และเอกซ์เทนด์ หรือ ความสัมพันธ์แบบสืบทอด ซึ่งสามารถใช้ได้ทั้งความสัมพันธ์ระหว่างผู้ใช้กับยูสเคส หรือความสัมพันธ์ระหว่าง ยูสเคส

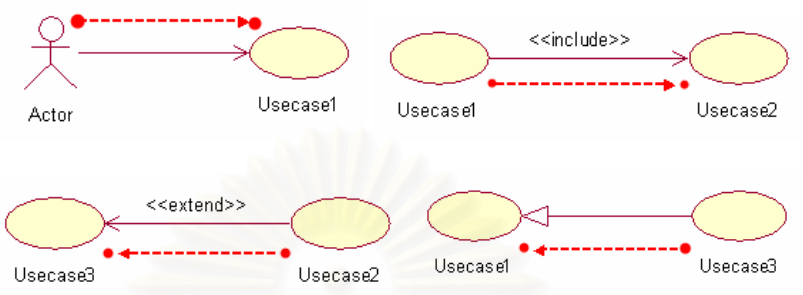


รูปที่ 34 การสร้างแผนภาพยูสเคสใหม่

2. การวาดเส้นความสัมพันธ์

การวาดเส้นความสัมพันธ์ระหว่างผู้ใช้กับยูสเคสหรือความสัมพันธ์ระหว่างยูสเคส ใช้เครื่องมือเส้นทึบหัวลูกศรเปิด ในการวาด โดยเริ่มวาดจากผู้ใช้ไปยังยูสเคสหรือถ้าเป็นความสัมพันธ์ระหว่างยูสเคสแบบอินคลูดให้วาดโดยเริ่มจากยูสเคสที่ทำการเรียกใช้ไปยังยูสเคสที่ถูกเรียกใช้ จะปรากฏหัวลูกศรชี้ไปยังยูสเคสที่ถูกเรียกใช้ ในทางกลับกันถ้าเป็นความสัมพันธ์แบบเอกซ์เทนด์ให้วาดเริ่มจากยูสเคสที่ถูกเรียกใช้ชี้กลับไปยังยูสเคสที่ทำการเรียกใช้ จะปรากฏ

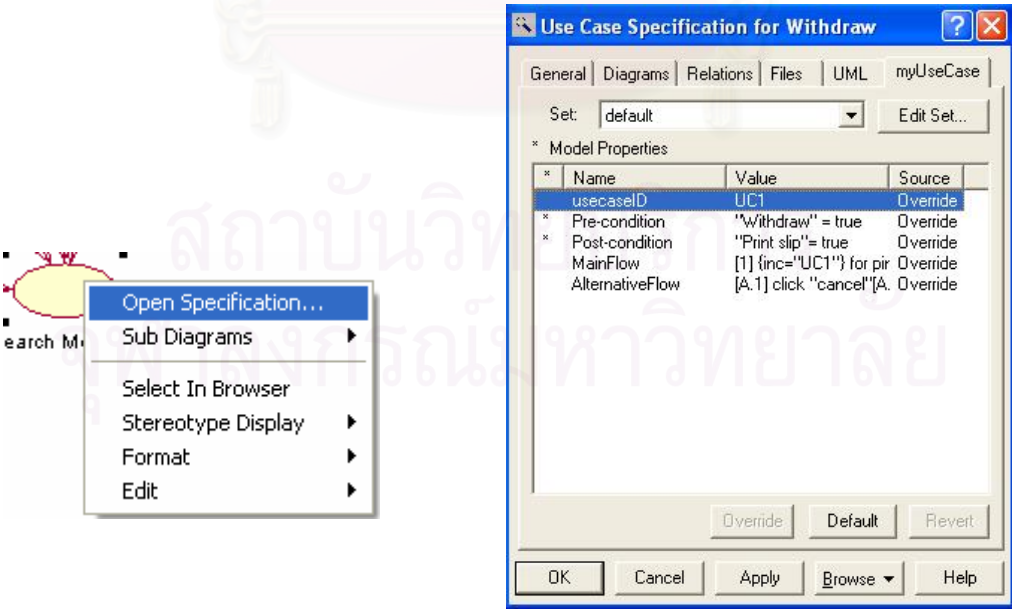
หัวลูกศรชี้ไปยังยูสเคสที่เรียกใช้ และถ้าเป็นความสัมพันธ์แบบสืบทอด ให้ทำการวาดจากยูสเคสที่สืบทอดไปยังยูสเคสที่เป็นยูสเคสแม่ เช่นเดียวกัน ถ้าเป็นความสัมพันธ์แบบสืบทอดของผู้ใช้ ก็จะมีลักษณะเดียวกันกับความสัมพันธ์แบบสืบทอดของยูสเคส ดังแสดงการวาดในรูปที่ 35



รูปที่ 35 การวาดเส้นความสัมพันธ์แบบต่างๆ

3. การเขียนคำอธิบายยูสเคส

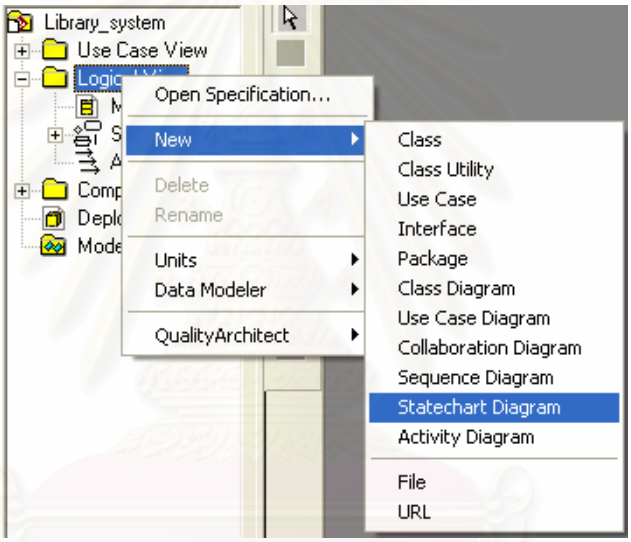
เมื่อวาดแผนภาพยูสเคสเรียบร้อยแล้ว การเขียนคำอธิบายยูสเคสสามารถเขียนได้โดยการคลิกเม้าส์ปุ่มขวาบนยูสเคส แสดงดังรูปที่ 36 จากนั้นเลือกที่แท็ปหัวข้อมายยูสเคส (MyUseCase) เพื่อเปิดหน้าจอส่วนของการเขียนคำอธิบายยูสเคส หากต้องการเขียนข้อมูลใดก็ให้คลิกเลือกที่บรรทัดนั้น แล้วกดปุ่มเขียนทับ (Override) แล้วจึงมาพิมพ์ข้อมูลที่ช่องค่า (Value) โดยการพิมพ์ข้อมูลคำอธิบายยูสเคสจะต้องเป็นไปตามข้อกำหนดซึ่งอธิบายไว้ในบทที่ 3



รูปที่ 36 การเปิดหน้าจอส่วนการเขียนคำอธิบายยูสเคส

การวาดแผนภาพดับเบิลยูเอ็นดี

1. การสร้างแผนภาพดับเบิลยูเอ็นดี แผนภาพดับเบิลยูเอ็นดีที่ใช้ในการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ จะมีเพียง 1 แผนภาพ ซึ่งวาดอยู่ในมุมมองโลจิคอล เริ่มต้นโดยการคลิกเม้าส์ปุ่มขวาเลือกเมนู New -> Statechart Diagram หน้าสารบบมุมมองโลจิคอลดังรูปที่ 36 ภายในจะปรากฏแผนภาพ NewDiagram ให้ทำการเปลี่ยนชื่อเป็นชื่อระบบที่ต้องการ จากนั้นดับเบิลคลิกเพื่อเปิดหน้าต่างว่างขึ้น ด้านซ้ายมือของหน้าต่างจะปรากฏแถบเครื่องมือในการวาดแผนภาพดับเบิลยูเอ็นดี ซึ่งเครื่องมือที่ใช้วาดแผนภาพ ประกอบด้วย 1) จุดเริ่มต้น 2) จุดสิ้นสุด 3) โนด 4) เส้นเชื่อมโนด

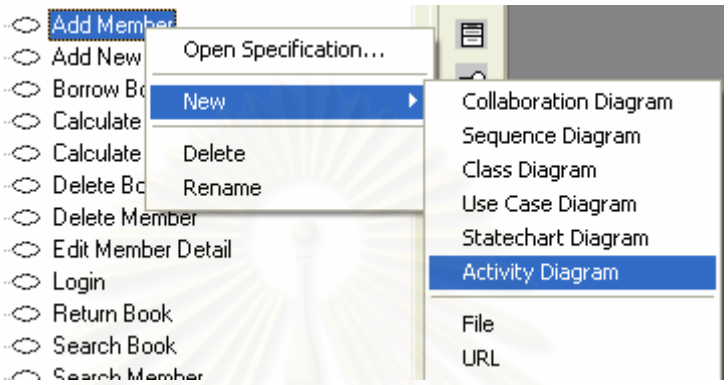


รูปที่ 37 การสร้างแผนภาพดับเบิลยูเอ็นดี

การวาดแผนภาพกิจกรรม

1. การสร้างแผนภาพกิจกรรม แผนภาพกิจกรรมที่ใช้ในการสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ จะมีได้หลายแผนภาพโดย 1 แผนภาพต่อแผนภาพยูสเคส 1 แผนภาพ ซึ่งวาดอยู่ในยูสเคส เริ่มต้นโดยการคลิกเม้าส์ปุ่มขวาที่ยูสเคสเลือกเมนู New -> Activity Diagram ดังรูปที่ 37 ภายในจะปรากฏแผนภาพ Add book ให้ทำการเปลี่ยนชื่อเป็นชื่อระบบที่ต้องการ จากนั้นดับเบิลคลิกเพื่อเปิดหน้าต่างว่างขึ้น ด้านซ้ายมือของหน้าต่างจะปรากฏแถบเครื่องมือในการวาดแผนภาพกิจกรรม ซึ่งเครื่องมือที่ใช้วาดแผนภาพ ประกอบด้วย 1) จุดเริ่มต้น 2) จุดสิ้นสุด 3) โนด 4) เส้นเชื่อมโนด 5) สวิมเลน

ข้อควรระวังในการเขียนแผนภาพกิจกรรมคือ จุดสิ้นสุดของการทำงานหลักกับจุดสิ้นสุดของการทำงานอื่นๆ ในคำอธิบายยูสเคส หากไม่ใช่จุดเดียวกันให้ตั้งชื่อให้กับจุดสิ้นสุดของการทำงานหลักให้เขียนชื่อจุดสิ้นสุดกำกับเป็น success และจุดสิ้นสุดของการทำงานอื่นๆ ให้เขียนชื่อจุดสิ้นสุดกำกับเป็น alternative



รูปที่ 38 การสร้างแผนภาพกิจกรรม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข.

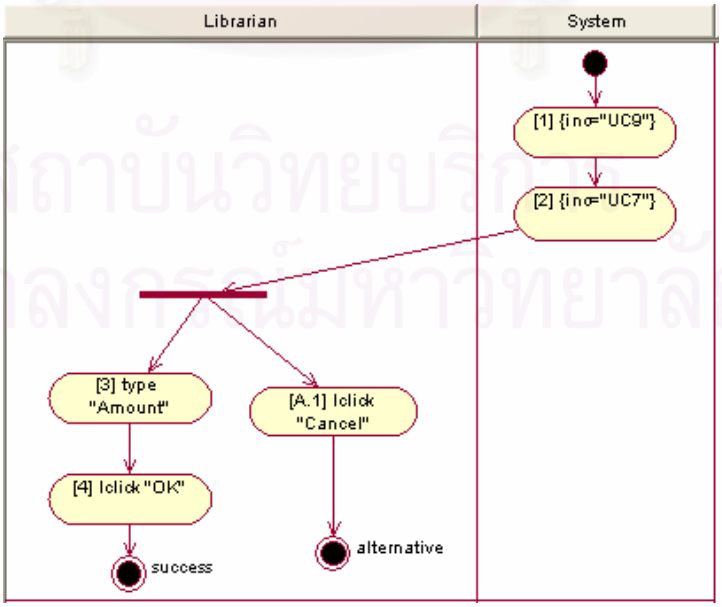
รายละเอียดยูสเคสและแผนภาพกิจกรรมของระบบที่ใช้ทดสอบ

คำอธิบายยูสเคสของระบบบริการในห้องสมุด

จากแผนภาพยูสเคสของระบบบริการในห้องสมุด ในรูปที่ 22 สามารถเขียนรายละเอียดคำอธิบายยูสเคสและแผนภาพกิจกรรมของแต่ละยูสเคส ยกตัวอย่าง 4 ยูสเคส สำหรับสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ แสดงดังต่อไปนี้

ตารางที่ 34 คำอธิบายยูสเคส Update Book

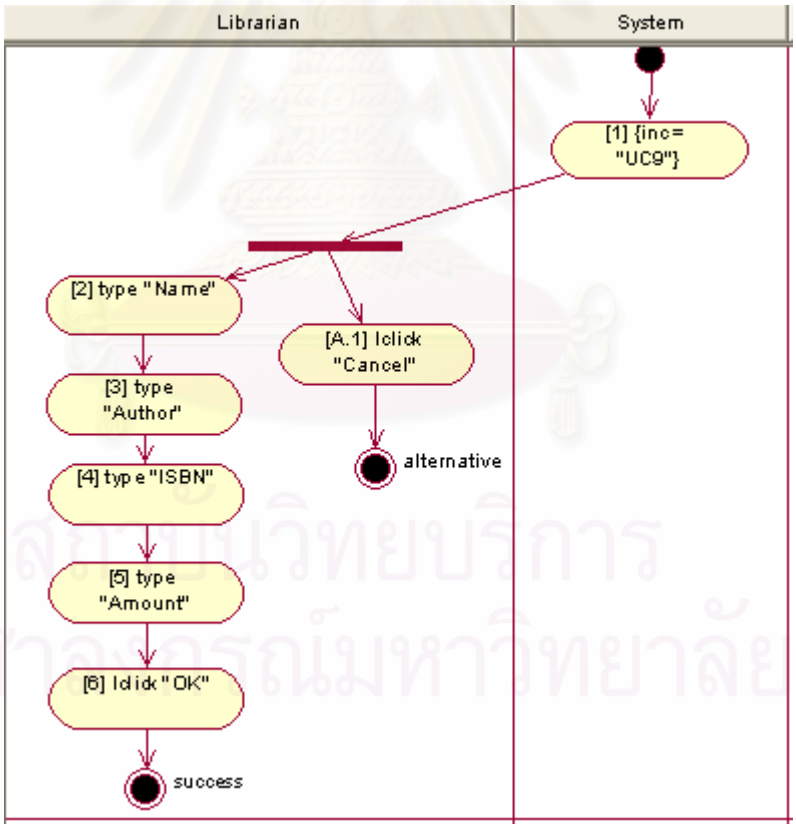
usecaseID	UC1
Precondition	“Book detail” = true
postcondition	-
mainFlow	[1] {inc = “UC9”} [2] {inc = “UC7”} [3] type “Amount” [4] lclick “OK” [5] End
alternateFlow	[A.1] lclick “Cancel” [A.2] End



รูปที่ 39 แผนภาพกิจกรรมของยูสเคส Update Book

ตารางที่ 35 คำอธิบายยูสเคส Add New Book

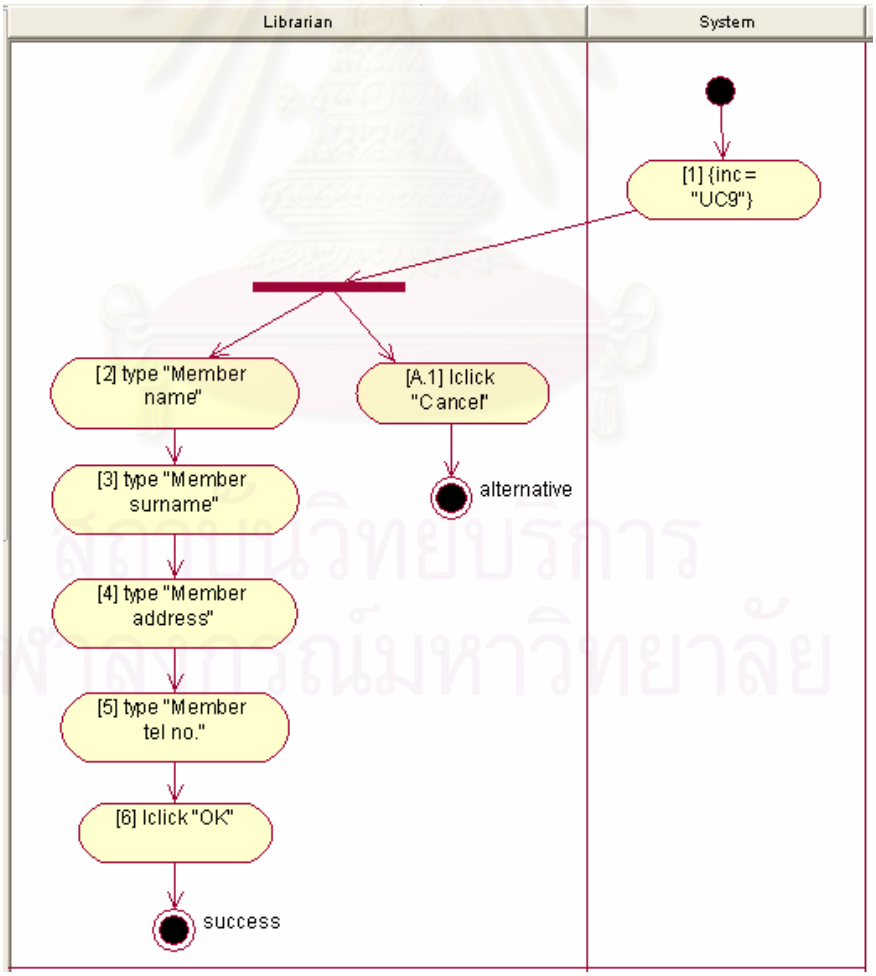
usecaseID	UC2
Precondition	“Book detail” = true
postcondition	
mainFlow	[1] {inc = “UC9”} [2] type “Name” [3] type “Author” [4] type “ISBN” [5] type “Amount” [6] lclick “OK” [7] End
alternateFlow	[A.1] lclick “Cancel” [A.2] End



รูปที่ 40 แผนภาพกิจกรรมของยูสเคส Add New Book

ตารางที่ 36 คำอธิบายยูสเคส Add member

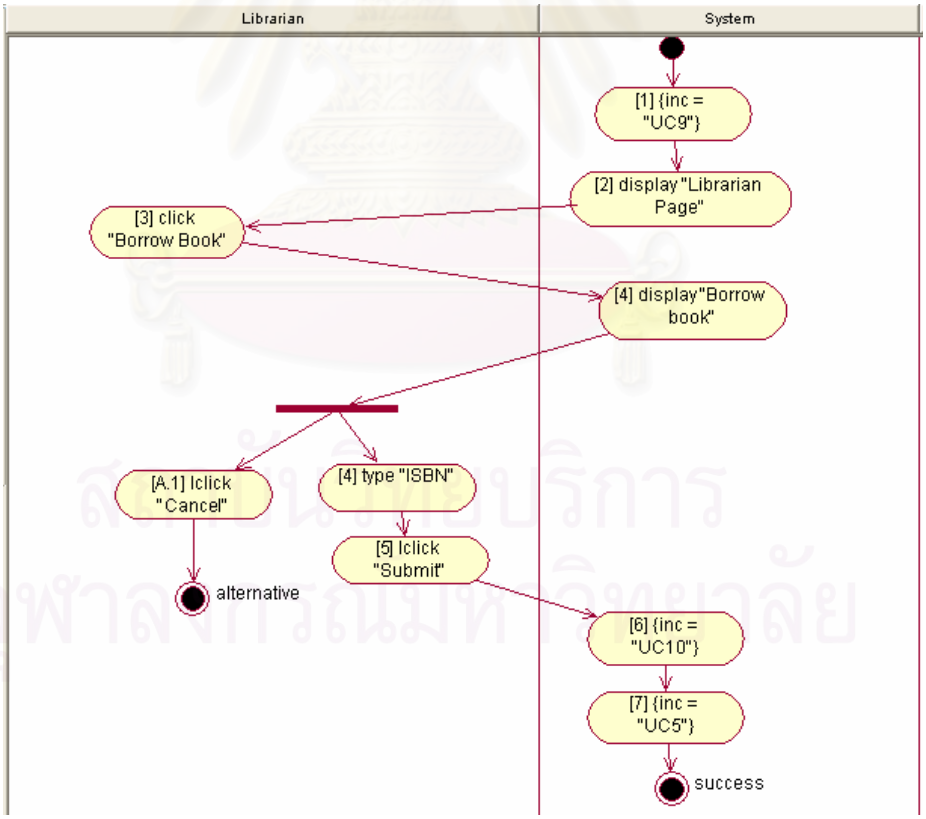
usecaseID	UC3
Precondition	“Member detail” = true
postcondition	
mainFlow	[1] {inc = “UC9”} [2] type “Name” [3] type “Surname” [4] type “Member address” [5] type “Mamber tel no.” [6] lclick “OK” [7] End
alternateFlow	[A.1] lclick “Cancel” [A.2] End



รูปที่ 41 แผนภาพกิจกรรมของยูสเคส Add member

ตารางที่ 37 คำอธิบายยูสเคส Borrow book

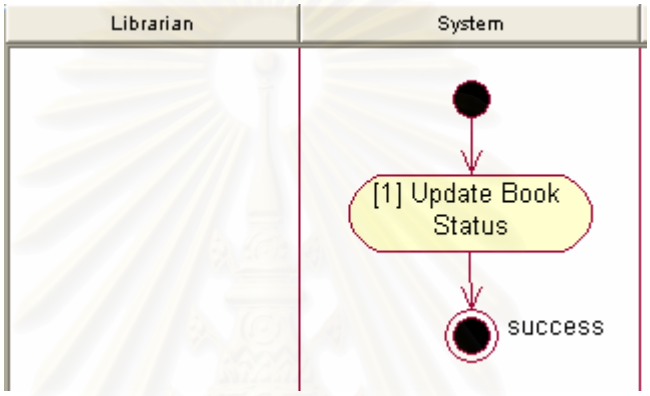
usecaseID	UC4
Precondition	“Borrow book” = true
postcondition	
mainFlow	[1] {inc = “UC9”} [2] display “Librarian Page” [3] lclick “Borrow Book” [4] type “ISBN” [5] lclick “Submit” [6] {inc = “UC10”} [7] {inc = “UC5”} [8] End
alternateFlow	[A.1] lclick “Cancel” [A.2] End



รูปที่ 42 แผนภาพกิจกรรมของยูสเคส Borrow book

ตารางที่ 38 คำอธิบายยูสเคส Update Book Status

usecaseID	UC5
Precondition	
postcondition	
mainFlow	[1] System update book status [7] End
alternateFlow	



รูปที่ 43 แผนภาพกิจกรรมของยูสเคส Update Book Status

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค.

คู่มือการติดตั้งและใช้งานเครื่องมือสร้างและทดสอบกรณีทดสอบจียูไอจากยูเอ็มแอล

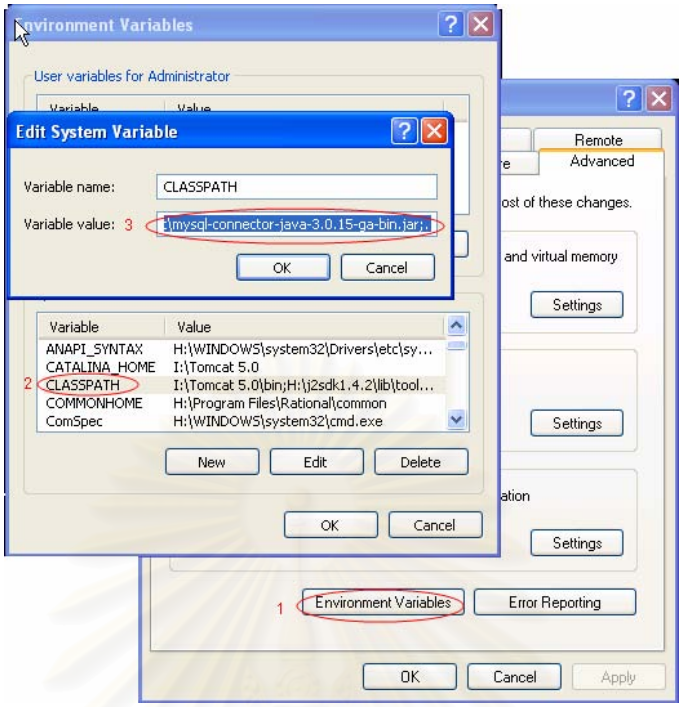
เครื่องมือสร้างกรณีทดสอบจากยูเอ็มแอลทำหน้าที่สร้างกรณีทดสอบจากเอกสารการออกแบบระบบซึ่งออกแบบโดยใช้แผนภาพยูเอ็มแอล โดยกรณีทดสอบที่สร้างได้จะอยู่ในรูปของแฟ้มเอกซ์เอ็มแอลเพื่อเอื้อต่อการนำไปใช้ทดสอบแบบอัตโนมัติ นอกจากนั้นยังมีส่วนที่ใช้ในการทดสอบกรณีทดสอบที่สร้างขึ้นโดยการนำกรณีทดสอบนั้นใช้ทดสอบกับระบบ เพื่อแสดงให้เห็นถึงความสามารถในการใช้งานได้จริงของกรณีทดสอบทีสร้างขึ้น

การติดตั้งเครื่องมือ

1. ติดตั้งตัวพัฒนาโปรแกรมภาษาจาวา เวอร์ชัน 1.4.2 พร้อมด้วยเจเอฟซียูนิต เวอร์ชั่น 2.08
2. ติดตั้งระบบจัดการฐานข้อมูลมายเอสคิวแอล เวอร์ชัน 1.4 พร้อมด้วยเจดีบีซีสำหรับมายเอสคิวแอล
3. ติดตั้งเครื่องมือเรชันเนล โรส เอ็นเตอร์ไพรส์ เวอร์ชัน 2002 พร้อมด้วยเครื่องมือยูนิซิส โรส เอกซ์เอ็มไอ ทูล 1.3 แอด-อิน
4. เมื่อทำการติดตั้งเครื่องมือเสร็จเรียบร้อยแล้ว ทำการกำหนดที่อยู่ของแฟ้มเพื่อให้ระบบปฏิบัติการสามารถเรียกใช้ได้ โดยกำหนดในส่วนคุณสมบัติของมายคอมพิวเตอร์ (My Computer) แสดงดังรูปที่ 43 ให้ทำการเลือกตามลำดับตัวเลขที่ปรากฏ โดยทำการกำหนดค่าของสารบบของตัวพัฒนาภาษาจาวา เจดีบีซีของมายเอสคิวแอล และเจเอฟซียูนิต แสดงดังรายการต่อไปนี้

   - ในตัวเลือก set path = พาทของที่ทำการติดตั้งตัวพัฒนาภาษาจาวา/bin;
   - ในตัวเลือก set classpath = พาทของเจดีบีซีมายเอสคิวแอล;
   - ในตัวเลือก set classpath = พาทของเจเอฟซียูนิต/jfcunit.jar; พาทของเจยูนิต/junit.jar; และพาทของ jakarta-regexp-1.3/jakarta-regexp-1.3.jar;

5. ทำการคัดลอกสารบบซึ่งบรรจุแฟ้มต่างๆ ของเครื่องมือสร้างและทดสอบกรณีทดสอบลงในคอมพิวเตอร์



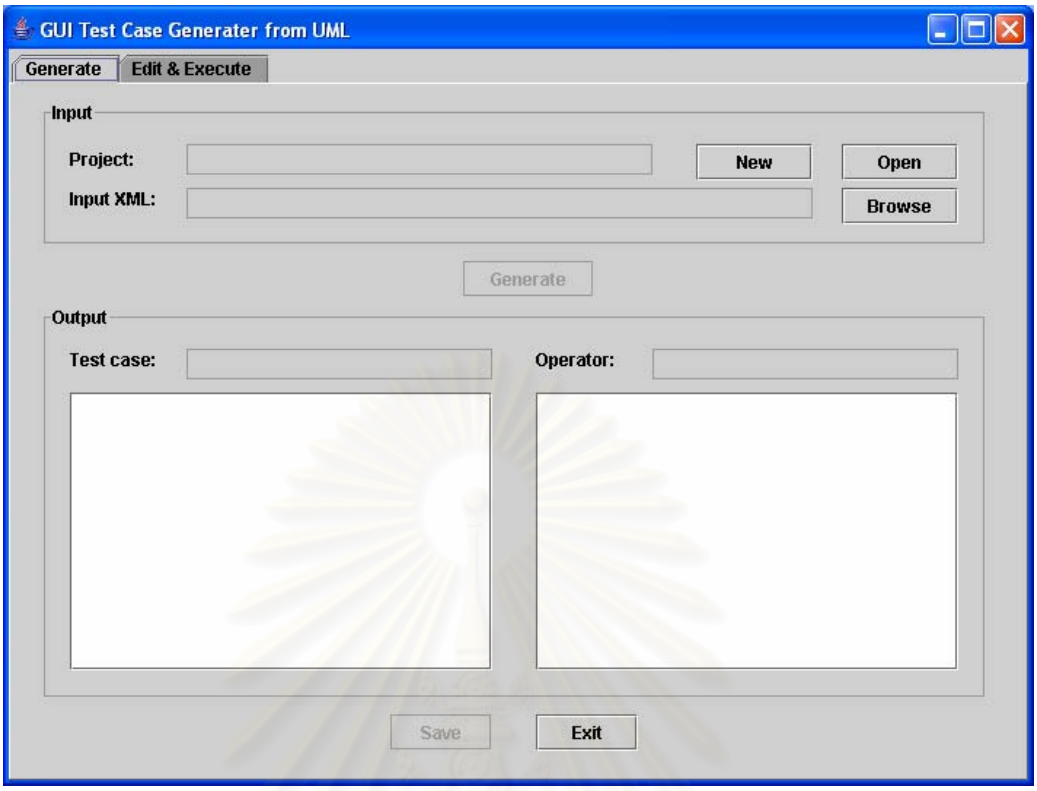
รูปที่ 44 การกำหนดพาทของแฟ้มที่ต้องการให้ระบบปฏิบัติการสามารถเรียกใช้ได้

6. เมื่อทำการติดตั้งโปรแกรมเครื่องมือทั้งหมดและคัดลอกสารบบของเครื่องมือลงยังเครื่องคอมพิวเตอร์แล้ว จากนั้นทำการติดตั้งฐานข้อมูลของเครื่องมือ โดยแฟ้มเก็บสคริปต์คำสั่งเอสคิวแอลในการสร้างตารางของเครื่องมือ

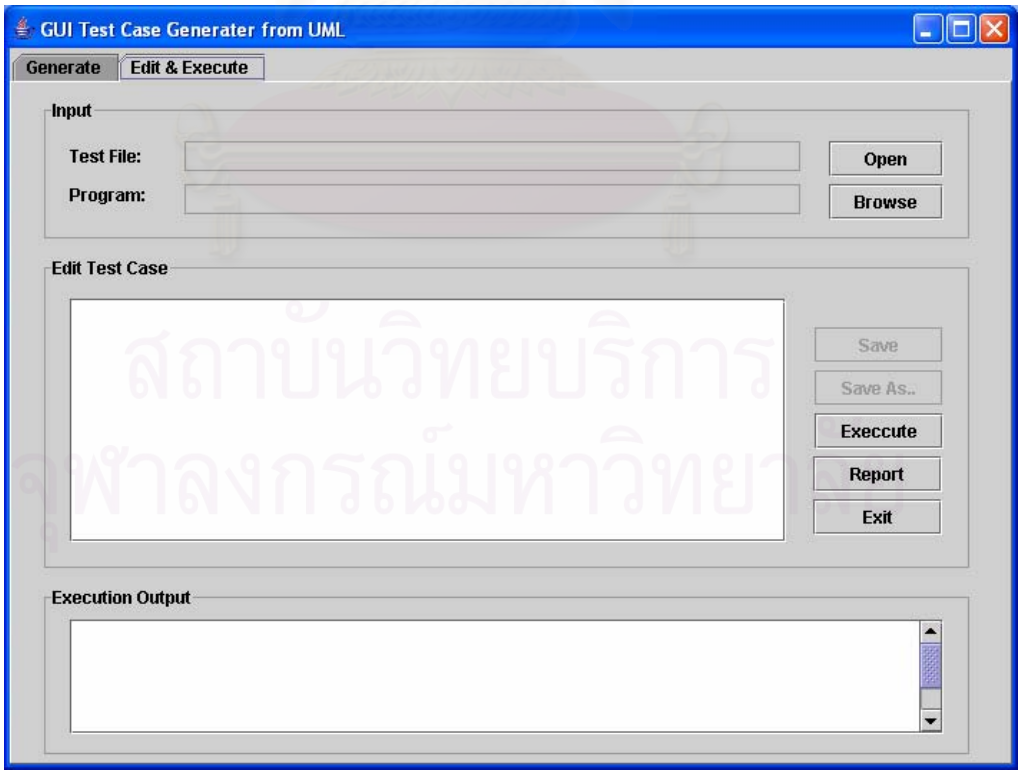
การใช้งานเครื่องมือ

เครื่องมือทำหน้าที่นำเข้าเอกสารการออกแบบระบบโดยใช้แผนภาพยูเอ็มแอล โปรแกรมจะทำงานภายใต้ระบบปฏิบัติการวินโดว์ 98 ขึ้นไป มีส่วนประกอบสำคัญอยู่ 2 ส่วน คือ

- ส่วนสร้างกรณีทดสอบ มีข้อมูลนำเข้าคือ ชื่อโครงการ แฟ้มเอกซ์เอ็มแอลการออกแบบระบบ ที่สร้างจากเครื่องมือเรชันเนลโรส โดยผลลัพธ์ที่ได้จะเป็นแฟ้มเอกซ์เอ็มแอลของตัวดำเนินการและแผนการทดสอบ ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงดังรูปที่ 45
- ส่วนแก้ไขและดำเนินการทดสอบ ข้อมูลนำเข้าคือ ชื่อโครงการ โปรแกรมของระบบที่อยู่ในรูปของแฟ้ม .class โดยเป็นแฟ้มหลักของโปรแกรม เป็นส่วนประกอบจากนั้นผู้ดำเนินการทดสอบจะแก้ไขรายละเอียดในส่วนของค่าข้อมูลที่ใช้ทดสอบ รวมทั้งสร้างจุดตรวจสอบ และดำเนินการทดสอบ ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงดังรูปที่ 46



รูปที่ 45 ส่วนต่อประสานกราฟิกกับผู้ใช้ (ส่วนสร้างกรณีทดสอบ)



รูปที่ 46 ส่วนต่อประสานกราฟิกกับผู้ใช้ (ส่วนแก้ไขและดำเนินการทดสอบ)

รายละเอียดในแต่ละส่วนของเครื่องมือ ซึ่งแยกออกด้วยการใช้แท็ป แสดงดังนี้

1. ส่วนสร้างกรณีทดสอบ อยู่ในแท็ป Generate ทำหน้าที่สร้างกรณีทดสอบจากแฟ้มเอกซ์เอ็มแอล ประกอบด้วย

    - Text field (Project) แสดงชื่อของโครงการที่ต้องการสร้างกรณีทดสอบ
    - ปุ่ม New ใช้เพื่อกรอกชื่อของโครงการที่ต้องการสร้าง
    - ปุ่ม Open ใช้เพื่อเปิดแฟ้มโครงการที่เคยสร้างแล้ว เพื่อทำการสร้างกรณีทดสอบใหม่
    - Text field (Input XML) แสดงชื่อแฟ้มเอกซ์เอ็มแอล ซึ่งเป็นข้อมูลนำเข้าของครื่องมือ
    - ปุ่ม Browse ใช้เพื่อเลือกแฟ้มเอกซ์เอ็มแอล
    - ปุ่ม Generate ใช้เพื่อดำเนินการสร้างกรณีทดสอบ
    - Text field (Test case) แสดงชื่อของแฟ้มกรณีทดสอบอยู่ในรูปของแฟ้มเอกซ์เอ็มแอล
    - Text field (Operator) แสดงชื่อของแฟ้มตัวดำเนินการอยู่ในรูปของแฟ้มเอกซ์เอ็มแอล
    - Text area ทั้งสองส่วน แสดงข้อมูลภายในแฟ้มกรณีทดสอบและ แฟ้มตัวดำเนินการ
    - ปุ่ม Save ใช้เพื่อทำการบันทึกโครงการที่ได้สร้างขึ้น
    - ปุ่ม Exit ใช้เพื่อทำการออกจากเครื่องมือ

2. ส่วนแก้ไข และดำเนินการทดสอบ อยู่ในแท็ป Edit & Execute ทำหน้าที่แก้ไขและดำเนินการทดสอบกรณีทดสอบสร้างที่สร้างขึ้น ประกอบด้วย

    - Text field (Test File) แสดงแฟ้ม Test ที่จะใช้ทดสอบ
    - ปุ่ม Open ใช้เพื่อเลือกโครงการที่ต้องการทดสอบ
    - Text field (Program) แสดงชื่อโปรแกรมที่ต้องการทดสอบ (Program Under Test)
    - ปุ่ม Browse ใช้เพื่อเลือกโปรแกรมที่ต้องการทดสอบ โดยโปรแกรมดังกล่าวต้องอยู่ในรูปแบบของ .class

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- Text Area แสดงข้อมูลของกรณีทดสอบทั้งหมด และสามารถ
- (Edit Test Case) แก้ไขข้อมูลของกรณีทดสอบ โดยจะสามารถแก้ไขข้อมูลในส่วนของ “ค่า” (Value) ที่ต้องการใช้ในการทดสอบ รวมทั้งจุดทดสอบ
- ปุ่ม Save ใช้เพื่อทำการบันทึกกรณีทดสอบที่แก้ไขแล้ว
- ปุ่ม Save As.. ใช้เพื่อทำการบันทึกกรณีทดสอบเป็นแฟ้มใหม่
- ปุ่ม Execute ทำการทดสอบ โดยจะทำการเรียกโปรแกรมที่เลือกไว้ในส่วนของ Program ขึ้นมา และดำเนินการทดสอบตามกรณีทดสอบ
- ปุ่ม Report แสดงรายงานผลการทดสอบ โดยจะแสดงผลรายงานการทดสอบเป็นแฟ้มเอชทีเอ็มแอลของโครงการปัจจุบัน
- ปุ่ม Exit ออกจากเครื่องมือ
- Text area (Output) แสดงข้อมูลต่างๆ ในการทดสอบ เช่น ข้อความผิดพลาดที่อาจเกิดขึ้น ผลการทดสอบ

เมื่อทำการทดสอบโปรแกรม จะมีการทำงานเหมือนกับเครื่องมือเล่นซ้ำ โดยจะนำค่าที่เก็บไว้ในกรณีทดสอบมาสร้างเป็นเหตุการณ์ เพื่อจำลองการทำงานของผู้ทดสอบโดยควบคุมในส่วนของอุปกรณ์ควบคุม เช่น เม้าส์ แป้นอักขระ เป็นต้น ให้สร้างเหตุการณ์แก่คอมโพเนนท์ส่วนต่อประสานกราฟิกกับผู้ใช้ ตามลำดับ และดูผลลัพธ์ที่ได้เทียบกับจุดตรวจสอบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวเพชรน้ำค้าง วงษ์ศุทธิภากร เกิดเมื่อวันที่ 6 กรกฎาคม พ.ศ. 2523 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต (วท.บ.) สาขาวิทยาการคอมพิวเตอร์ จากภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2545 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2546

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย