การสร้างกรณีทดสอบจากแผนภาพซีเควนซ์

นางสาวฉัฐธร ทองระอา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาการพัฒนาซอฟต์แวร์ด้านธุรกิจ ภาควิชาสถิติ
คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2550
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

TEST CASE GENERATION BASED ON SEQUENCE DIAGRAM

Miss Chuttorn Tongra-ar

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Business Software Development

Department of Statistics

Faculty of Commerce and Accountancy

Chulalongkorn University

Academic Year 2007

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การสร้างกรณีทดสอบจากแผนภาพซีเควนซ์
โดย	นางสาว ฉัฐธร ทองระอา
สาขาวิชา	การพัฒนาซอฟต์แวร์ด้านธุรกิจ
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สมจารี ปรียานนท์

คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะพาณิชยศาสตร์และการบัญชี

(รองศาสตราจารย์ ดร. อรรณพ ตันละมัย)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย ดร. อัษฎาพร ทรัพย์สมบูรณ์)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. สมจารี ปรียานนท์)

..... กรรมการ

(อาจารย์ ดร. จันทร์เจ้า มงคลนาวิน)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ฉัฐธร ทองระอา : การสร้างกรณีทดสอบจากแผนภาพซีเควนซ์. (TEST CASE GENERATION BASED ON SEQUENCE DIAGRAM) อ. ที่ปรึกษา: ผศ.ดร. สมจารี ปรียานนท์, 236 หน้า.

งานวิจัยนี้นำเสนอวิธีการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์และโอซีแอล ด้วยการประยุกต์ใช้กราฟควบคุมกระแสไออาร์ซีเอฟจีของ Rountev และคณะ เพื่อให้กราฟควบคุมกระแสสามารถนำมาใช้ในการสร้างกรณีทดสอบได้ งานวิจัยนี้จึงได้เพิ่มคอนดิชันโหนด เลเบิลที และเลเบิลมาเป็นองค์ประกอบของกราฟควบคุมกระแสไออาร์ซีเอฟจี พร้อมทั้งสร้างหลักการแปลงภาพซีเควนซ์เป็นไออาร์ซีเอฟจี และในส่วนของการสร้างข้อมูลทดสอบ ผู้วิจัยได้ใช้การวิเคราะห์ค่าขอบเขตเพื่อหาค่าของตัวแปร โดยขั้นตอนการสร้างกรณีทดสอบของงานวิจัยนี้ ประกอบด้วย การแปลงแผนภาพซีเควนซ์เป็นกราฟกระแสไออาร์ซีเอฟจี การกำหนดเงื่อนไขก่อนและหลังดำเนินการของเมท็อดด้วยโอซีแอล การสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ การสร้างข้อมูลทดสอบ และการสร้างกรณีทดสอบ

จากขั้นตอนและหลักการที่สร้างขึ้นสามารถนำมาพัฒนาเป็นเครื่องมือสร้างกรณีทดสอบจากแผนซีเควนซ์ด้วยภาษาซีชาร์ป โดยข้อมูลนำเข้าคือแผนภาพซีเควนซ์ในรูปแบบของเอกสารเอกซ์เอ็มแอลที่สร้างขึ้นด้วยโปรแกรมวิชวลพาราแดมสำหรับยูเอ็มแอลเวอร์ชัน 6.0 ผลลัพธ์ที่ได้คือชุดของกรณีทดสอบ

ผลจากการประเมินความสามารถของเครื่องมือด้วยการเปรียบเทียบกรณีทดสอบที่สร้างจากเครื่องมือกับกรณีทดสอบที่สร้างจากผู้เชี่ยวชาญแล้ว พบว่าจำนวนกรณีทดสอบที่เครื่องมือสร้างคิดเป็นร้อยละ 75 ของจำนวนกรณีทดสอบที่ผู้เชี่ยวชาญสร้าง และจำนวนข้อมูลของกรณีทดสอบที่สร้างจากเครื่องมือตรงกับข้อมูลของกรณีทดสอบที่ผู้เชี่ยวชาญสร้าง คิดเป็นร้อยละ 43.15 ของจำนวนข้อมูลของกรณีทดสอบที่ผู้เชี่ยวชาญสร้าง จากผลการประเมินสามารถมั่นใจได้ว่าเมื่อนำเครื่องมือนี้ไปใช้งาน จะช่วยลดค่าใช้จ่ายและแรงงานในขั้นตอนการออกแบบกรณีทดสอบได้เป็นอย่างมาก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา สถิติ ลายมือชื่อนิสิต .....

สาขาวิชา การพัฒนาซอฟต์แวร์ด้านธุรกิจ ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา 2550

4882169826 : MAJOR BUSINESS SOFTWARE DEVELOPMENT

KEY WORD: TEST CASE/ SEQUENCE DIAGRAM / OBJECT-ORIENTED TESTING

CHUTTORN TOGNRA-AR : TEST CASE GENERATION BASED ON SEQUENCE DIAGRAM. THESIS ADVISOR : ASST. PROF. SOMJAREE PREEYANONT, Doc.Eng., 236 pp.

This research proposes an approach for creating test case from UML sequence diagram and OCL by applying IRCFG control flow graph, which is presented by Rountev et al. In order to use the IRCFG control flow graph for generating test case, the researcher adds condition nodes and some label into the IRCFG control flow graph and also creates the method for transforming the UML sequence diagram to the IRCFG control flow graph. For generating test data, the researcher applies boundary value analysis theory to get the value of variables. The processing step for generating test case are (1) transformation UML sequence diagram to a tree representation (IRCFG) (2) defining OCL pre-post conditions of methods (3) creating the All-IRCFG path and (4) generating the test data and the test case.

According to the processing step as mention, the researcher developed the software tool for generating test case from UML sequence diagram by using C# language. The tool gets UML sequence diagram data as input in XML document form which is created by converting UML sequence diagram to XML document by Visual Paradigm for UML 6.0 then the tool generates the number of test case.

The researcher evaluated the tool by comparing tool's test case with the test case that created by expert tester. The comparison results show that the number of tool's test case is 75 percent of the number of test case which is created by expert. Furthermore, the number of tool's test case information is approximately 43.15 percent of the number of test case information which is created by expert. The results can confirm that this tool can reduce testing cost and effort on the process of test case design.

Department: Statistics Student's Signature: Chuttorn Tognra-ar

Field of Study: Business Software Development Advisor's Signature: Somjaree Preeyanont

Academic Year: 2007

กิตติกรรมประกาศ

ผู้วิจัยขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. สมจารี ปรียานนท์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้การชี้แนะแนวทางต่างๆ ให้กับผู้วิจัยจนสำเร็จเป็นวิทยานิพนธ์ฉบับนี้ และขอขอบพระคุณผู้ช่วยศาสตราจารย์ ผู้ช่วยศาสตราจารย์ ดร. อัษฎาพร ทรัพย์สมบูรณ์ ประธานกรรมการวิทยานิพนธ์ อาจารย์ ดร. จันทร์เจ้า มงคลนาวิน กรรมการวิทยานิพนธ์ ที่ช่วยชี้แนะสิ่งต่างๆ และอาจารย์ทุกท่านที่ให้ความรู้และอบรมสิ่งต่างๆ ให้กับผู้วิจัย และขอขอบคุณบริษัทผลิตซอฟต์แวร์และน้อง ๆ ที่ให้ความช่วยเหลือในการให้ข้อมูลเพื่อนำมาใช้เป็นหน่วยตัวอย่างสำหรับงานวิจัยนี้

ที่สำคัญที่สุดขอขอบพระคุณคุณพ่อ คุณแม่ คุณป้า และพี่น้องทุกคนที่ให้ความสนับสนุนและให้กำลังใจตลอดมา สุดท้ายขอขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจและให้ความช่วยเหลือรวมทั้ง นางสาวชมพูนุช เปลี่ยนวิถี ผู้ให้คำปรึกษาที่ดีตลอดมา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง..... ฎ

สารบัญภาพ..... ฒ

บทที่ 1 บทนำ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์งานวิจัย..... 4

1.3 ขอบเขตงานวิจัย..... 4

1.4 ขั้นตอนการดำเนินงานวิจัย..... 5

1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 5

1.6 นิยามคำศัพท์สำคัญ..... 6

บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 7

2.1 การทดสอบซอฟต์แวร์ (Software Testing)..... 7

2.1.1 กระบวนการทดสอบซอฟต์แวร์ (Software Testing Process)..... 8

2.1.2 ระดับของการทดสอบซอฟต์แวร์ (Level of Software Testing)..... 9

2.1.3 เทคนิคในการทดสอบซอฟต์แวร์ (Software Testing Technique)..... 14

2.1.3.1 การทดสอบแบบกล่องดำ (Black-box Testing)..... 14

2.1.3.2 การทดสอบแบบกล่องขาว (White-box Testing)..... 16

2.2 เกณฑ์ความคลอบคลุมของแผนภาพยูเอ็มแอล
(UML-Based Coverage Criteria)..... 21

2.2.1 เกณฑ์ของแผนภาพคลาส (Class Diagram Criteria)..... 21

2.2.2 เกณฑ์ของแผนภาพซีเควนซ์ (Sequence Diagram Criteria)..... 22

2.2.3 เกณฑ์ของแผนภาพคอมมิวนิเคชั่น (Communication Diagram Criteria)..... 23

หน้า

2.2.4 เกณฑ์ของแผนภาพสเตทแมชชีน (State Machine Diagram Criteria)..... 23
2.2.5 เกณฑ์ของแผนภาพแอคทิวิตี (Activity Diagram Criteria)..... 24
2.3 แผนภาพซีเควนซ์ (Sequence Diagram)..... 24
2.4 อินเตอร์โพรซิเดอรัลเรสตริกต์คอนโทรลโฟลกราฟหรือไออาร์ซีเอฟจี (Interprocedural Restricted Control-Flow Graph -IRCFG) และเกณฑ์ความ ครอบคลุม..... 31
2.5 ภาษาข้อจำกัดเชิงวัตถุหรือโอซีแอล (Object Constraint Language-OCL)..... 36
2.5.1 ข้อจำกัดของคลาสซิไฟเออร์ (Constraint on Classifiers)..... 38
2.5.2 ข้อจำกัดของการดำเนินการ (Constraints on Operations)..... 38
2.5.3 ข้อจำกัดของแอตทริบิวส์ (Constraint on Attribute)..... 39
2.6 งานวิจัยที่เกี่ยวข้อง..... 40

บทที่ 3 แนวคิดและขั้นตอนการดำเนินงานวิจัย..... 42
3.1 ศึกษาแนวทางในการแปลงแผนภาพซีเควนซ์ไปเป็นไออาร์ซีเอฟจีที่รองรับคอมบายด์แฟรกเมนต์และสร้างหลักในการแปลง..... 48
3.2 ศึกษาและหลักการในการสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ ค่าของข้อมูลทดสอบ และวิธีการสร้างกรณีทดสอบ..... 52
3.3 ออกแบบและพัฒนาเครื่องมือในการสร้างกรณีทดสอบ..... 53
3.4 ประเมินการใช้เครื่องมือและความสามารถของเครื่องมือ..... 54
3.4.1 คุณสมบัติของหน่วยตัวอย่าง..... 54
3.4.2 การเลือกหน่วยตัวอย่าง..... 54
3.4.2 ที่มาของหน่วตัวอย่าง..... 55
3.4.3 การประเมินความสามารถของเครื่องมือ..... 58

บทที่ 4 ผลการวิจัย..... 59
4.1 ขั้นตอนในการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์..... 59
4.2 การแปลงแผนภาพซีเควนซ์เป็นกราฟควบคุมกระแสไออาร์ซีเอฟจี..... 61
4.2.1 การเพิ่มองค์ประกอบในกราฟกระแสไออาร์ซีเอฟจี..... 62

หน้า

4.2.2 หลักการสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจีจากแผนภาพซีเควนซ์..... 63
4.2.2.1 การสร้างอาร์ซีเอฟจี..... 63
4.2.2.2 การสร้างโหนดภายในอาร์ซีเอฟจี..... 64
4.2.2.3 การสร้างเส้นเชื่อมอินทราโพรซิเดอรัลหรือเส้นเชื่อมอาร์ซีเอฟจี..... 66
4.2.2.4 การสร้างเส้นเชื่อมอินเตอร์โพรซิเดอรัล..... 74
4.3 การกำหนดเงื่อนไขก่อนและหลังดำเนินการเมท็อดด้วยภาษาโอซีแอล..... 76
4.4 การสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์..... 77
4.5 การสร้างข้อมูลทดสอบ..... 80
4.5.1 การเลือกคอนดิชันโหนด..... 80
4.5.2 การสร้างค่าของตัวแปรในเพรดิเคต..... 83
4.6 การสร้างกรณีทดสอบ..... 88
4.7 การออกแบบและพัฒนาเครื่องมือสร้างกรณีทดสอบ..... 91
4.7.1 ขอบเขตของเครื่องมือและผู้ใช้งาน..... 91
4.7.2 เอกสารความต้องการซอฟต์แวร์ (Software Requirement Specification)..... 92
4.7.3 แผนภาพยูสเคส (Use Case Diagram) 93
4.7.4 แผนภาพคลาส (Class Diagram) 94
4.7.5 การพัฒนาเครื่องมือสร้างกรณีทดสอบ..... 94

บทที่ 5 การทดสอบและประเมินผลเครื่องมือสร้างกรณีทดสอบ..... 98
5.1 การทดสอบเครื่องมือสร้างกรณีทดสอบ..... 98
5.1.1 ขั้นตอนการทดสอบ..... 98
5.1.2 ผลการทดสอบ..... 101
5.1.3 สรุปผลการทดสอบ..... 102
5.2 การประเมินการใช้เครื่องมือและความสามารถของเครื่องมือ..... 102
5.2.1 การทดสอบโปรแกรมด้วยกรณีทดสอบที่ได้จากเครื่องมือ..... 102
5.2.2 การเปรียบเทียบกรณีทดสอบที่สร้างจากเครื่องมือเทียบกับกรณีทดสอบที่สร้างจากผู้เชี่ยวชาญ..... 113

หน้า

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ..... 120
6.1 สรุปผลการวิจัย..... 120
6.1.1 ขั้นตอนและหลักการที่ใช้ในการสร้างกรณีทดสอบ..... 120
6.1.2 ผลการทดสอบเครื่องมือ..... 121
6.1.3 ผลการประเมินความสามารถของเครื่องมือ..... 121
6.2 การนำงานวิจัยไปประยุกต์ใช้ (Contribution)..... 122
6.2.1 การนำงานวิจัยไปใช้ในเชิงทฤษฎี (Theoretical Contribution)..... 122
6.2.2 การนำงานวิจัยไปใช้ในเชิงประยุกต์ (Practical Contribution)..... 122
6.3 ข้อจำกัดและข้อเสนอแนะ..... 123

รายการอ้างอิง..... 125
ภาคผนวก..... 129
ภาคผนวก ก เอกสารแสดงการออกแบบซอฟต์แวร์สร้างกรณีทดสอบ..... 130
ภาคผนวก ข เอกสารประกอบการสร้างกรณีทดสอบและกรณีทดสอบจากเครื่องมือ
และผู้เชี่ยวชาญ..... 164
ภาคผนวก ค ซอร์สโค้ดของยูสเคสการซื้อตั๋วที่นำมาใช้ในการทดสอบ..... 230
ประวัติผู้เขียนวิทยานิพนธ์..... 236

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 2-1 แสดงความสัมพันธ์ของระดับการทดสอบ ข้อกำหนดและเทคนิคที่ใช้ในการทดสอบ และผู้ดำเนินการทดสอบในแต่ละระดับ..... 19
ตารางที่ 2-2 แสดงความสัมพันธ์ของแผนภาพยูเอ็มแอลในแต่ละระดับของการทดสอบในกระบวนการทดสอบซอฟต์แวร์..... 20
ตารางที่ 2-3 แสดงชุดของเส้นทางที่เป็นไปตามเกณฑ์ทุกสาขาอาร์ซีเอฟจี..... 35
ตารางที่ 2-4 แสดงชุดของเส้นทางที่เป็นไปตามเกณฑ์ทุกสาขาที่ไม่ซ้ำ..... 36
ตารางที่ 3-1 แสดงการเปรียบแนวคิดการสร้างกรณีทดสอบของงานวิจัยนี้กับแนวคิดการสร้างกรณีทดสอบที่ใช้กับซอฟต์แวร์แบบสัญนิยม..... 44
ตารางที่ 3-2 แสดงกรณีทดสอบที่เป็นไปตามเส้นทางในรูปที่ 3-3 46
ตารางที่ 3-3 แสดงกรณีทดสอบในการเรียกใช้งานเมท็อดของซอร์สโค้ดในรูปที่ 3-4..... 48
ตารางที่ 3-4 สรุปประเภทคอมบายด์แฟรกเมนต์ที่งานวิจัยนี้รองรับ..... 49
ตารางที่ 3-5 ตารางแสดงรายละเอียดของแผนภาพซีเควนซ์ที่แปลงกลับจากซอร์สโค้ด..... 57
ตารางที่ 3-6 ตารางแสดงรายละเอียดคอมบายด์แฟรกเมนต์ภายในแผนภาพซีเควนซ์ที่นำมาเป็นหน่วยตัวอย่างของงานวิจัยนี้..... 57
ตารางที่ 3-7 ตารางแสดงเอกสารที่ใช้ในการสร้างกรณีทดสอบจากเครื่องมือและจากผู้เชี่ยวชาญ..... 58
ตารางที่ 4-1 ตารางแสดงรายละเอียดของวัตถุประสงค์และกิจกรรมโดยสรุปของแต่ละกระบวนการในการสร้างกรณีทดสอบ..... 60
ตารางที่ 4-2 แสดงตัวอย่างของการกำหนดเงื่อนไขก่อนและหลังดำเนินการของเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ยูสเคสการถอนเงิน..... 76
ตารางที่ 4-3 แสดงชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ภายในกราฟกระแสไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การถอนเงิน..... 79
ตารางที่ 4-4 แสดงลำดับของการเลือกคอนดิชันโหนดจากเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ในรูปที่ 4-13..... 81
ตารางที่ 4-5 แสดงความสัมพันธ์ของนิพจน์พร้อมทั้งผลลัพท์..... 84

หน้า

ตารางที่ 4-6 แสดงลักษณะของเพรดิเคต ผลลัพธ์เพรดิเคต และค่าของข้อมูลประเภทตัวแปรสตริง..... 87
ตารางที่ 4-7 แสดงลักษณะของเพรดิเคต ผลลัพธ์เพรดิเคต และค่าของข้อมูลประเภทตัวแปรบูลีน..... 88
ตารางที่ 4-8 ตารางแสดงตัวอย่างของกรณีทดสอบที่สร้างจากหลักการโดยใช้แผนภาพซีเควนซ์การถอนเงิน..... 89
ตารางที่ 5-1 แสดงเงื่อนไขก่อนและหลังดำเนินการของแต่ละเมท็อดในแผนภาพซีเควนซ์การซื้อตั๋ว..... 100
ตารางที่ 5-2 ตารางสรุปจำนวนกรณีทดสอบของแต่ละแผนภาพซีเควนซ์ที่สร้างจากเครื่องมือ..... 102
ตารางที่ 5-3 ตารางแสดงชื่อของตัวแปรที่เป็นพารามิเตอร์ของเมท็อดแรกและชื่อตัวแปรที่เครื่องมือสามารถนำมาสร้างค่าได้ของแผนภาพซีเควนซ์การซื้อตั๋ว..... 103
ตารางที่ 5-4 แสดงชุดกรณีทดสอบของการซื้อตั๋วและผลลัพธ์ที่ได้จากการทดสอบกรณีที่การทดสอบเป็นไปตามผลลัพธ์คาดหวัง..... 105
ตารางที่ 5-5 แสดงประเภทความผิดพลาดที่ใส่ลงไปในซอร์สโค้ดสำหรับแต่ละกรณีทดสอบของยูสเคสการซื้อตั๋ว..... 109
ตารางที่ 5-6 แสดงชุดกรณีทดสอบของการซื้อตั๋วและผลลัพธ์ที่ได้จากการทดสอบกรณีที่การทดสอบเป็นไม่ไปตามผลลัพธ์คาดหวัง..... 110
ตารางที่ 5-7 ตารางแสดงชุดกรณีทดสอบที่สร้างโดยผู้เชี่ยวชาญ (TC_{Exp}) ของยูสเคสการซื้อตั๋ว..... 114
ตารางที่ 5-8 แสดงจำนวนกรณีทดสอบที่ได้จากผู้เชี่ยวชาญและเครื่องมือ..... 117
ตารางที่ 5-9 ตารางสรุปจำนวนข้อมูลรวมของกรณีทดสอบที่สร้างจากเครื่องมือ ข้อมูลรวมของกรณีทดสอบที่สร้างจากผู้เชี่ยวชาญ และจำนวนรวมที่มีความหมายเหมือนกันจำแนกตามยูสเคสโดยจับคู่แต่ละกรณีทดสอบที่มีค่าข้อมูลทดสอบอยู่ในขอบเขตเดียวกัน..... 118
ตารางที่ ข-1 รายละเอียดยูสเคสการซื้อตั๋ว..... 165
ตารางที่ ข-2 ตารางแสดงกรณีทดสอบของการซื้อตั๋วที่สร้างโดยผู้เชี่ยวชาญ..... 168

หน้า

ตารางที่ ข-3 ตารางแสดงเส้นทางไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การซื้อตั๋วที่เครื่องมือสร้าง..... 169
ตารางที่ ข-4 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การซื้อตั๋วที่เครื่องมือสร้าง..... 170
ตารางที่ ข-5 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การซื้อตั๋วที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)..... 174
ตารางที่ ข-6 รายละเอียดยูสเคสการเข้าใช้งาน..... 176
ตารางที่ ข-7 ตารางแสดงกรณีทดสอบของการเข้าใช้งานที่สร้างโดยผู้เชี่ยวชาญ..... 178
ตารางที่ ข-8 ตารางแสดงเส้นทางไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การเข้าใช้งานที่เครื่องมือสร้าง..... 181
ตารางที่ ข-9 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การเข้าใช้งานที่เครื่องมือสร้าง..... 182
ตารางที่ ข-10 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การเข้าใช้งานที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์) 186
ตารางที่ ข-11 รายละเอียดยูสเคสการยอมรับอินวอยซ์..... 188
ตารางที่ ข-12 ตารางแสดงกรณีทดสอบของการยอมรับอินวอยซ์ที่สร้างโดยผู้เชี่ยวชาญ..... 190
ตารางที่ ข-13 ตารางแสดงเส้นทางไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การยอมรับอินวอยซ์ที่เครื่องมือสร้าง..... 194
ตารางที่ ข-14 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การยอมรับอินวอยซ์ที่เครื่องมือสร้าง..... 195
ตารางที่ ข-15 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การยอมรับอินวอยซ์ที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)..... 198
ตารางที่ ข-16 รายละเอียดยูสเคสการโปรโมชันร่วมกับการขาย..... 200
ตารางที่ ข-17 ตารางแสดงกรณีทดสอบของการโปรโมชันร่วมกับการขายที่สร้างโดยผู้เชี่ยวชาญ..... 202
ตารางที่ ข-18 ตารางแสดงเส้นทางไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การโปรโมชันร่วมกับการขายที่เครื่องมือสร้าง..... 203
ตารางที่ ข-19 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การโปรโมชันร่วมกับการขายที่เครื่องมือสร้าง..... 204

หน้า

ตารางที่ ข-20 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การโปรโมชันร่วมกับการขายที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)..... 207
ตารางที่ ข-21 รายละเอียดยูสเคสการลดราคา..... 208
ตารางที่ ข-22 ตารางแสดงกรณีทดสอบของการลดราคาที่สร้างโดยผู้เชี่ยวชาญ..... 210
ตารางที่ ข-23 ตารางแสดงเส้นทางไอาร์ซีเอฟจีของแผนภาพซีเควนซ์การลดราคาที่เครื่องมือสร้าง..... 212
ตารางที่ ข-24 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การลดราคาที่เครื่องมือสร้าง..... 213
ตารางที่ ข-25 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การลดราคาที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)..... 219
ตารางที่ ข-26 รายละเอียดยูสเคสการส่งสินค้า..... 222
ตารางที่ ข-27 ตารางแสดงกรณีทดสอบของการส่งสินค้าที่สร้างโดยผู้เชี่ยวชาญ..... 224
ตารางที่ ข-28 ตารางแสดงเส้นทางไอาร์ซีเอฟจีของแผนภาพซีเควนซ์การส่งสินค้าที่เครื่องมือสร้าง..... 225
ตารางที่ ข-29 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การส่งสินค้าที่เครื่องมือสร้าง..... 226
ตารางที่ ข-30 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การส่งสินค้าที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)..... 228

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 2-1 แสดง 3 ลำดับชั้นของแต่ละหน่วย..... 10
รูปที่ 2-2 แสดงวิธีการรวมแต่ละหน่วยโดยใช้กลยุทธ์การทดสอบแบบบิ๊กแบง..... 11
รูปที่ 2-3 แสดงวิธีการรวมแต่ละหน่วยโดยใช้กลยุทธ์การทดสอบแบบบนลงล่าง..... 11
รูปที่ 2-4 แสดงวิธีการรวมแต่ละหน่วยโดยใช้กลยุทธ์การทดสอบแบบล่างขึ้นบน..... 12
รูปที่ 2-5 แสดงวิธีการรวมแต่ละหน่วยย่อยโดยใช้กลยุทธ์การทดสอบแบบแซนด์วิช..... 12
รูปที่ 2-6 ตัวอย่างของการแปลงผังงานมาเป็นกราฟควบคุมกระแส..... 17
รูปที่ 2-7 แสดงรูปแบบของเงื่อนไขผสมในกราฟควบคุมกระแส..... 17
รูปที่ 2-8 แสดงความสัมพันธ์แบบ เอ็ม-เอ็ม ระหว่างคลาสในแผนภาพคลาส..... 21
รูปที่ 2-9 แสดงแผนภาพทั้งหมดของยูเอ็มแอลรุ่น 2.0..... 25
รูปที่ 2-10 สัญลักษณ์แสดงเฟรมของแผนภาพ..... 25
รูปที่ 2-11 แสดงรูปแบบการเขียนเลเบิลของแผนภาพ..... 26
รูปที่ 2-12 สัญลักษณ์รูปคนซึ่งเป็นตัวแทนของแอคเตอร์..... 26
รูปที่ 2-13 รูปแบบและเครื่องหมายในการแสดงอ็อบเจกต์..... 26
รูปที่ 2-14 แสดงไลฟ์ไลน์ของอ็อบเจกต์..... 26
รูปที่ 2-15 แสดงแอคติเวชั่น..... 27
รูปที่ 2-16 แสดงเมจเสจที่ใช้ในการสื่อสารระหว่างอ็อบเจกต์..... 27
รูปที่ 2-17 แสดงอ็อบเจกต์เดสสตรัคชั่น..... 27
รูปที่ 2-18 แสดงสัญลักษณ์ของคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟ..... 27
รูปที่ 2-19 แสดงรูปแบบการเขียนคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟ..... 29
รูปที่ 2-20 แสดงรูปแบบการเขียนคอมบายด์แฟรกเมนต์ประเภทออปชัน..... 30
รูปที่ 2-21 แสดงรูปแบบการเขียนคอมบายด์แฟรกเมนต์ประเภทลูป..... 30
รูปที่ 2-22 ตัวอย่างแผนภาพซีเควนซ์และไออาร์ซีเอฟจีของแผนภาพซีเควนซ์..... 31
รูปที่ 2-23 แสดงขอบเขตของอาร์ซีเอฟจี..... 32
รูปที่ 2-24 แสดงสัญลักษณ์ที่เป็นตัวแทนของเมท็อดที่ถูกเรียกใช้..... 32
รูปที่ 2-25 แสดงความสัมพันธ์ระหว่างโหนดภายในอาร์ซีเอฟจี..... 33
รูปที่ 2-26 แสดงความสัมพันธ์ระหว่างอาร์ซีเอฟจี..... 33

หน้า

รูปที่ 2-27 แสดงตัวอย่างของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์..... 34
รูปที่ 2-28 แสดงตัวอย่างของเงื่อนไขคงที่ เงื่อนไขก่อนดำเนินการ และเงื่อนไขหลังดำเนินการในโอซีแอลที่แนบไว้เป็นหมายเหตุในแบบจำลองยูเอ็มแอล (แผนภาพคลาส)..... 37
รูปที่ 2-29 แสดงรูปแบบในการเขียนข้อจำกัดประเภทเงื่อนไขคงที่ในรูปแบบข้อความ..... 38
รูปที่ 2-30 แสดงตัวอย่างการเขียนข้อจำกัดเงื่อนไขคงที่..... 38
รูปที่ 2-31 แสดงรูปแบบการเขียนข้อจำกัดของการดำเนินการในรูปแบบข้อความ..... 39
รูปที่ 2-32 แสดงตัวอย่างการเขียนข้อจำกัดของการดำเนินการ..... 39
รูปที่ 2-33 แสดงตัวอย่างการเขียนข้อจำกัดเงื่อนไขหลังดำเนินการโดยใช้คำสงวน result..... 39
รูปที่ 2-34 แสดงรูปแบบการเขียนข้อจำกัดของแอตทริบิวส์ในรูปแบบข้อความ..... 39
รูปที่ 2-35 แสดงตัวอย่างการเขียนข้อจำกัดของแอตทริบิวส์..... 40
รูปที่ 3-1 ภาพรวมในการออกแบบและสร้างกรณีทดสอบของงานวิจัยนี้..... 42
รูปที่ 3-2 แสดงตัวอย่างซอร์สโค้ดภาษาซี (C) ในฟังก์ชันชื่อ getNumDaysInMonth..... 45
รูปที่ 3-3 แสดงกราฟควบคุมกระแสของฟังก์ชัน getNumDaysInMonth..... 46
รูปที่ 3-4 แสดงตัวอย่างซอร์สโค้ดภาษาซีชาร์ป (C#) ในเมท็อดชื่อ checkWithdraw..... 47
รูปที่ 3-5 แสดงตัวอย่างของแผนภาพซีเควนซ์ที่ออกแบบโดยใช้คอมบายแฟรกเมนต์ประเภทอัลเทอร์เนทีฟ..... 49
รูปที่ 3-6 แสดงตัวอย่างของแผนภาพซีเควนซ์ที่ออกแบบโดยใช้คอมบายแฟรกเมนต์ประเภทออบชัน..... 50
รูปที่ 3-7 แสดงตัวอย่างของแผนภาพซีเควนซ์ที่ออกแบบโดยใช้คอมบายแฟรกเมนต์ประเภทลูป..... 50
รูปที่ 3-8 (a) แสดงรูปแบบกราฟควบคุมกระแสของ McCabeและ(b) แสดงรูปแบบไออาร์ซีเอฟจีของ Rountev และคณะ..... 51
รูปที่ 3-9 ตัวอย่าง(a) แผนภาพซีเควนซ์และ (b) ไออาร์ซีเอฟจีของแผนภาพซีเควนซ์ซึ่ง Rountev และคณะใช้เพื่อประกอบการอธิบาย..... 52
รูปที่ 3-10 แสดงการโต้ตอบ (Interaction) ของไคล์เอนท์-เซิร์ฟเวอร์อ็อบเจกต์ (Client-server Object)..... 56

หน้า

รูปที่ 4-1 แสดงภาพรวมของขั้นตอนและกระบวนการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์..... 59
รูปที่ 4-2 ตัวอย่างของการเพิ่มคอนดิชันโหนด เลเบิลที และเลเบิลเอฟลงในกราฟควบคุมกระแสไออาร์ซีเอฟจี..... 63
รูปที่ 4-3 แสดงวิธีการแมปปิ้ง (Mapping) เมจเสจจากแผนภาพซีเควนซ์ไปเป็นอาร์ซีเอฟจีของกราฟกระแสไออาร์ซีเอฟจี..... 64
รูปที่ 4-4 แสดงเงื่อนไขในการเลือกเมจเสจที่จะนำมาเป็นเมท็อดโหนดในอาร์ซีเอฟจี Msg_i... 64
รูปที่ 4-5 แสดงตัวอย่างการแมปปิ้งเมจเสจที่แสดงในแผนภาพซีเควนซ์มาเป็นเมท็อดโหนดภายในอาร์ซีเอฟจีของกราฟกระแสไออาร์ซีเอฟจี..... 65
รูปที่ 4-6 แสดงตัวอย่างการแมปปิ้งข้อจำกัดภายในแผนภาพซีเควนซ์มาเป็นคอนดิชันโหนดภายในอาร์ซีเอฟจี..... 65
รูปที่ 4-7 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลทีจากคอนดิชันโหนดไปยังเมท็อดโหนด..... 66
รูปที่ 4-8 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลทีจากคอนดิชันโหนดไปยังคอนดิชันโหนด..... 67
รูปที่ 4-9 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังเมท็อดโหนดกรณีที่ 1..... 68
รูปที่ 4-10 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังเมท็อดโหนดกรณีที่ 2..... 68
รูปที่ 4-11 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังคอนดิชันโหนดกรณีที่ 1..... 69
รูปที่ 4-12 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังคอนดิชันโหนดกรณีที่ 2..... 70
รูปที่ 4-13 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังเมท็อดโหนดกรณีที่ 1..... 71
รูปที่ 4-14 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังเมท็อดโหนดกรณีที่ 2..... 71

หน้า

รูปที่ 4-15 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังเมท็อดโหนดกรณีที่ 3..... 72
รูปที่ 4-16 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังคอนดิชันโหนดกรณีที่ 1..... 72
รูปที่ 4-17 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังคอนดิชันโหนดกรณีที่ 2..... 73
รูปที่ 4-18 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังคอนดิชันโหนดกรณีที่ 3..... 73
รูปที่ 4-19 แสดงตัวอย่างการสร้างเส้นเชื่อมอินเตอร์โพรซิเดอรัล..... 74
รูปที่ 4-20 แสดงตัวอย่างของแผนภาพซีเควนซ์ของยูสเคสการถอนเงิน..... 75
รูปที่ 4-21 แสดงกราฟกระแสไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การถอนเงิน..... 75
รูปที่ 4-22 แสดงอาร์ซีเอฟจีของ m1(a) พร้อมทั้งชุดเส้นทางอาร์ซีเอฟจีของ m1(b)..... 77
รูปที่ 4-23 แสดงลักษณะการท่องไปยัง $RCFG_{m2}$ ที่เชื่อมต่อกับเมท็อดโหนด m2..... 78
รูปที่ 4-24 แสดงชุดเส้นทางย่อยเมื่อท่องผ่านเมท็อดโหนด m2..... 78
รูปที่ 4-25 แสดงขั้นตอนในการสร้างข้อมูลทดสอบของแต่ละเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์..... 80
รูปที่ 4-26 แสดงตัวอย่าง 1 เส้นทางไออาร์ซีเอฟจีที่สมบูรณ์..... 81
รูปที่ 4-27 ตัวอย่างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์..... 82
รูปที่ 4-28 ตัวอย่างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์..... 83
รูปที่ 4-29 แสดงรูปแบบของนิพจน์ความสัมพันธ์..... 83
รูปที่ 4-30 แสดงรูปแบบของฟังก์ชันเพื่อหาค่าของตัวแปร..... 84
รูปที่ 4-31 แสดงตัวอย่างและวิธีแตกนิพจน์ย่อยและเลือกสุ่มค่าในกรณีที่ E_1 หรือ E_2 เป็นตัวเลข..... 85
รูปที่ 4-32 แสดงตัวอย่างและวิธีแตกนิพจน์ย่อยและเลือกสุ่มค่าในกรณีที่มี E_1 และ E_2 ประกอบด้วยตัวแปรตั้งแต่ 1 ตัวขึ้นไป..... 86
รูปที่ 4-33 รูปแบบของข้อมูลประเภทสตริงของงานวิจัยนี้..... 87
รูปที่ 4-34 รูปแบบของข้อมูลประเภทบูลีนของงานวิจัยนี้..... 87

หน้า

รูปที่ 4-35 แสดงการสรุปข้อมูลนำเข้าและข้อมูลนำออกของแต่ละกระบวนการในการสร้างกรณีทดทดสอบ..... 90
รูปที่ 4-36 แสดงภาพรวมของการสร้างกรณีทดสอบด้วยเครื่องมือสร้างกรณีทดสอบ..... 91
รูปที่ 4-37 แสดงแผนภาพยูสเคสของเครื่องมือที่พัฒนาขึ้น..... 93
รูปที่ 4-38 แสดงแผนภาพคลาสของเครื่องมือที่พัฒนาขึ้น..... 94
รูปที่ 4-39 หน้าจอแสดงรายละเอียดเอกสารเอกซ์เอ็มแอลของเครื่องมือสร้างกรณีทดสอบ 96
รูปที่ 4-40 หน้าจอแสดงรูปภาพของแผนภาพซีเควนซ์ของเครื่องมือสร้างกรณีทดสอบ..... 96
รูปที่ 4-41 หน้าจอแสดงคุณสมบัติของเมจเสจ โอซีแอล และตัวแปรของเครื่องมือสร้างกรณีทดสอบ..... 97
รูปที่ 4-42 หน้าจอแสดงเส้นทางไอออาร์ซีเอฟจีที่สมบูรณ์และกรณีทดสอบของเครื่องมือสร้างกรณีทดสอบ..... 97
รูปที่ 5-1 แสดงแผนภาพซีเควนซ์การซื้อตั๋ว..... 99
รูปที่ 5-2 แสดงเอกสารเอกซ์เอ็มแอลของแผนภาพซีเควนซ์การซื้อตั๋ว..... 100
รูปที่ ข-1 แผนภาพซีเควนซ์ของการซื้อตั๋ว..... 166
รูปที่ ข-2 แผนภาพซีเควนซ์ของการเข้าใช้งาน..... 177
รูปที่ ข-3 แผนภาพซีเควนซ์ของการยอมรับอินวอยซ์..... 189
รูปที่ ข-4 แผนภาพซีเควนซ์ของการโปรโมชันร่วมกับการขาย..... 201
รูปที่ ข-5 แผนภาพซีเควนซ์ของการลดราคา..... 209
รูปที่ ข-6 แผนภาพซีเควนซ์ของการส่งสินค้า..... 223

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทดสอบซอฟต์แวร์เป็นหนึ่งในขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์และถือเป็นขั้นตอนที่มีความสำคัญเป็นอย่างมาก เนื่องจากการทดสอบซอฟต์แวร์เป็นส่วนหนึ่งในกระบวนการควบคุมคุณภาพซอฟต์แวร์ (Software Quality Control) [1] ซึ่งหมายถึงการตรวจสอบกระบวนการพัฒนาซอฟต์แวร์โดยประเมินว่าซอฟต์แวร์ที่พัฒนาขึ้น สอดคล้องกับความต้องการ (Requirement Specification) ที่ได้ระบุไว้ในขั้นตอนก่อนหน้าหรือไม่ เพื่อให้ซอฟต์แวร์ที่ผลิตขึ้นมานั้นมีคุณภาพตรงตามที่กำหนดไว้ [2] วัตถุประสงค์ของการทดสอบคือการหาข้อผิดพลาดที่มีอยู่ในโปรแกรมให้ได้มากที่สุด เพื่อให้ซอฟต์แวร์มีประสิทธิภาพ น่าเชื่อถือ และถูกต้องตามความต้องการของผู้ใช้งาน Pressman [3] กล่าวไว้ว่า ขั้นตอนในการทดสอบซอฟต์แวร์มีค่าใช้จ่ายมากกว่าขั้นตอนอื่น ๆ ในกระบวนการพัฒนาซอฟต์แวร์ และ Harrold M.J. [4] ได้ระบุไว้ว่ามากกว่า 50 % ของค่าใช้จ่ายในการพัฒนาซอฟต์แวร์ได้ถูกใช้ไปในขั้นตอนการทดสอบ กระบวนการในการทดสอบซอฟต์แวร์ประกอบไปด้วยขั้นตอนดังต่อไปนี้ [3]

- การวางแผนการทดสอบ (Test Planning)
- การออกแบบกรณีทดสอบ (Test Case Design)
- การดำเนินการทดสอบ (Test Execution)
- การรวบรวมผลลัพธ์และประเมินการทดสอบ (Resultant Data Collection and Evaluation)

การทดสอบสามารถแบ่งออกเป็น 6 ระดับ [5] ได้แก่ การทดสอบระดับหน่วย (Unit Testing) การทดสอบระดับบูรณาการ (Integration Testing) การทดสอบระบบ (System Testing) การทดสอบเพื่อการยอมรับ (Acceptance Testing) การทดสอบแบบถดถอย (Regression Testing) และการทดสอบแบบอัลฟาและแบบเบต้า (Alpha and Beta Testing) ในการทดสอบแต่ละระดับจะต้องมีการสร้างกรณีทดสอบ (Test Cases) เพื่อนำไปใช้ในการดำเนินการทดสอบ โดยกรณีทดสอบประกอบด้วย ค่าของข้อมูลทดสอบ (Input Value) ผลลัพธ์คาดหวัง (Expected Result) และข้อมูลที่จำเป็นซึ่งทำให้ระบบอยู่ในสถานะที่เหมาะสมสำหรับค่าของข้อมูลทดสอบ [6]

ปัจจุบันการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ได้เข้ามามีบทบาทสำคัญและถูกนำมาใช้กันอย่างแพร่หลายในอุตสาหกรรมการพัฒนาซอฟต์แวร์ ทำให้เทคนิคและวิธีในการทดสอบซอฟต์แวร์แบบสัญนิยม (Conventional Software) ที่มีอยู่ในปัจจุบันไม่สามารถนำมาใช้ในการทดสอบซอฟต์แวร์เชิงวัตถุ (Object Oriented Software) ได้อย่างครบถ้วน เนื่องจากภาษาเชิงวัตถุมีลักษณะที่แตกต่างจากซอฟต์แวร์แบบสัญนิยม ในเรื่องของความสามารถการสืบทอด (Inheritance) ความสามารถภาวะพหุสัณฐาน (Polymorphism) [6] และการทำงานร่วมกันระหว่างอ็อบเจกต์ (Object Interaction) ของคลาส (Class) ที่ต่างกัน [7] ด้วยลักษณะดังกล่าวทำให้กลวิธี (Strategy) ที่ใช้ทดสอบกลไกการดำเนินการภายในของซอฟต์แวร์แบบสัญ-นิยมไม่เพียงพอในการทดสอบกลไกการดำเนินการภายในของซอฟต์แวร์เชิงวัตถุ ปัจจุบันมีงานวิจัยจำนวนหนึ่ง [8, 9, 10] ที่นำเสนอวิธีและเทคนิคต่าง ๆ ในการทดสอบซอฟต์แวร์เชิงวัตถุในระดับของการทดสอบคลาส (Class Level Testing) ซึ่งเทียบได้กับการทดสอบระดับหน่วย (Unit Testing) ในซอฟต์แวร์แบบสัญนิยม [9]

โดยทั่วไปแล้วคลาสที่ถูกสร้างขึ้นมักจะถูกทดสอบการดำเนินการภายในโดยผู้ที่พัฒนาคลาสนั้นเอง ดังนั้นจึงค่อนข้างจะมั่นใจได้ว่าทุกคลาสน่าจะได้รับการทดสอบมาเป็นอย่างดีแล้ว แต่เนื่องจากในกระบวนการของการสร้างซอฟต์แวร์ แต่ละคลาสหรือแต่ละแพ็กเกจ (Package) จะถูกมอบหมายให้ผู้พัฒนาแต่ละคนรับผิดชอบในการดำเนินการสร้าง เมื่อมีการนำคลาสหรือแพ็กเกจมาทำงานร่วมกันแล้ว จึงไม่สามารถรับประกันได้ว่าการทำงานระหว่างกันของอ็อบเจกต์ของคลาสหรืออ็อบเจกต์ที่อยู่ในแพ็กเกจที่แตกต่างกันจะสามารถทำงานร่วมกันได้อย่างถูกต้องและให้ผลลัพธ์ตรงตามที่ต้องการ จะเห็นได้ว่าการทดสอบโดยแยกคลาสมาทดสอบเพียงอย่างเดียวนั้น ไม่เพียงพอในการทดสอบซอฟต์แวร์เชิงวัตถุ ยังจะต้องคำนึงถึงการทำงานระหว่างกันของอ็อบเจกต์ของคลาสที่ต่างกันเป็นสำคัญด้วย [9]

ภาษายูเอ็มแอล (Unified Modeling Language - UML) เป็นภาษามาตรฐานที่ถูกกำหนดขึ้นโดยองค์กรโอเอ็มจี (Object Management Group - OMG) และเป็นที่นิยมใช้ในการเขียนแบบจำลองในการวิเคราะห์และออกแบบระบบ โดยยูเอ็มแอลรุ่นที่ 2.0 มีแผนภาพทั้งหมด 13 แผนภาพ [11] โดยสามารถแบ่งออกเป็น 2 ประเภท คือ แผนภาพเชิงโครงสร้าง (Structural Diagrams) และ แผนภาพเชิงพฤติกรรม (Behavioral Diagrams) แผนภาพเชิงโครงสร้าง ได้แก่ แผนภาพคลาส (Class Diagram) แผนภาพแพ็กเกจ (Package Diagram) แผนภาพวัตถุ (Object Diagram) แผนภาพคอมโพเนนต์ (Component Diagram) แผนภาพดีพลอยเมนต์ (Deployment Diagram) และแผนภาพคอมโพซีทสตรักเจอร์ (Composite Structure Diagram) ส่วนแผนภาพ

เชิงพฤติกรรม ได้แก่ แผนภาพยูสเคส (Use Case Diagram) แผนภาพซีเควนซ์ (Sequence Diagram) แผนภาพคอมมิวนิเคชัน (Communication Diagram) แผนภาพสเตทแมชชีน (State Machine Diagram) แผนภาพกิจกรรม (Activity diagram) แผนภาพแสดงการโต้ตอบ (Interaction Overview Diagram) และแผนภาพไทม์มิง (Timing Diagram) ปัจจุบันมีหลายงานวิจัยที่ได้นำเสนอเกณฑ์ความครอบคลุม (Coverage Criteria) ในการทดสอบ โดยอ้างอิงจากแผนภาพยูเอ็มแอล [7,12,13,14] และมีหลายงานวิจัยที่สร้างแนวคิดในการสร้างกรณีทดสอบจากแผนภาพยูเอ็มแอล [8,6,10,14]

แผนภาพยูเอ็มแอลที่แสดงการทำงานระหว่างกันของอ็อบเจกต์ได้แก่ แผนภาพคอมมิวนิ-เคชัน และแผนภาพซีเควนซ์ [11] Rountev และคณะ [7] ได้กล่าวไว้ว่า แผนภาพซีเควนซ์นั้นถูกใช้เพื่อออกแบบการทำงานระหว่างอ็อบเจกต์จึงเหมาะสำหรับการนำไปใช้ในการระบุวัตถุประสงค์หรือเป้าหมายในการทดสอบ และงานวิจัยดังกล่าวยังได้นำเสนอกลุ่มของเกณฑ์ความครอบคลุมในการทดสอบการทำงานระหว่างกันของอ็อบเจกต์โดยสร้างจากแผนภาพซีเควนซ์ และจากงานวิจัยของ Abdurazik และOffutt [6] ซึ่งเปรียบเทียบกรณีทดสอบที่ได้จากแผนภาพสเตทแมช-ชีนกับกรณีทดสอบที่ได้จากแผนภาพซีเควนซ์ ผลสรุปหนึ่งของงานวิจัยสรุปไว้ว่าแผนภาพซีเควนซ์นั้นเหมาะแก่การทดสอบในระดับบูรณาการ (Integration Testing) และเหมาะกับการใช้เทคนิคการทดสอบแบบกล่องขาว (White-box Testing)

ในความเป็นจริงแล้วซอฟต์แวร์ไม่สามารถถูกนำมาทดสอบในทุกกรณีได้ เนื่องจากการทดสอบจะต้องกระทำภายใต้ทรัพยากรที่มีอยู่อย่างจำกัด เช่น เงินทุนและเวลา เป็นต้น ดังนั้นจึงจำเป็นต้องมีการระบุเกณฑ์ความครอบคลุมไว้ในแผนการทดสอบ ซึ่งสามารถถูกกำหนดได้ตั้งแต่ขั้นตอนการออกแบบ (Design Phase) ไม่จำเป็นต้องรอให้ดำเนินการสร้างซอฟต์แวร์ให้เสร็จสมบูรณ์ก่อน (Coding Phase) การทำเช่นนี้จะทำให้สามารถกำหนดระยะเวลาและค่าใช้จ่ายในการทดสอบได้แต่เนิ่น ๆ อีกทั้งกรณีทดสอบก็สามารถถูกสร้างหลังจากการออกแบบเสร็จสิ้นเช่นกัน นอกเหนือจากนั้นแล้ว เมื่อกรณีทดสอบถูกสร้างขึ้น ผู้ทดสอบ (Tester) มักจะพบความไม่ต้องกัน (Inconsistency) และความกำกวม (Ambiguity) ในเอกสารข้อกำหนดซอฟต์แวร์ (Software Requirement Specification) หรือเอกสารการออกแบบซอฟต์แวร์ (Software Design Specification) จึงทำให้เอกสารข้อกำหนดและเอกสารการออกแบบได้รับการปรับปรุงให้ถูกต้องหรือสอดคล้องกันก่อนลงมือเขียนโปรแกรม [6] กรณีทดสอบจะถูกสร้างขึ้นโดยคำนึงถึงเกณฑ์ความครอบคลุมในการทดสอบที่ได้กำหนดไว้ ดังนั้นหากเรามีหลักการในการสร้างกรณีทดสอบ

เหล่านั้นโดยอัตโนมัติและสามารถสร้างได้ตรงต่อเกณฑ์ที่กำหนดไว้ จะสามารถลดค่าใช้จ่ายและระยะเวลาของการทดสอบซอฟต์แวร์ในส่วนของขั้นตอนการออกแบบกรณีทดสอบได้เป็นอย่างมาก

ในการออกแบบระบบด้วยแผนภาพยูเอ็มแอล ผู้ออกแบบสามารถระบุข้อจำกัดขององค์ประกอบในแผนภาพ ด้วยการใช้ภาษาเชิงข้อจำกัดหรือโอซีแอล (Object Constraint Language-OCL) ซึ่งเป็นภาษาทางการ (Formal Language) ที่ใช้เพื่อแสดงข้อจำกัด (Constraint) และตรรกะ (Logic) ในแบบจำลองยูเอ็มแอล (UML Model) [15] และเป็นส่วนหนึ่งของยูเอ็มแอลที่ถูกสร้างขึ้นโดยองค์กรโอเอ็มจี ข้อจำกัดถูกแสดงในรูปของนิพจน์บูลีน (Boolean Expression) ซึ่งมีการคืนค่ากลับเป็นจริงหรือเท็จ โดยข้อจำกัดมีอยู่ด้วยกัน 3 ประเภท [9,15] ได้แก่ เงื่อนไขคงที่ (Invariant) เงื่อนไขก่อนดำเนินการ (Precondition) และเงื่อนไขหลังดำเนินการ (Postcondition)

งานวิจัยนี้สนใจในส่วนของการออกแบบกรณีทดสอบที่นำไปใช้ในการทดสอบการทำงานร่วมกันระหว่างอ็อบเจกต์ ซึ่งเทียบได้กับการทดสอบระดับบูรณาการของซอฟต์แวร์แบบสัญนิยม โดยงานวิจัยนี้จะนำเสนอหลักการในการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์และโอซีแอล ซึ่งกรณีทดสอบที่สร้างขึ้นนั้นจะคำนึงถึงเกณฑ์ความครอบคลุมที่ได้นำเสนอไว้ในงานวิจัยของ Rountev และคณะ [7] และจะประยุกต์ใช้ไอออาร์ซีเอฟจี (Interprocedural Restricted Control Flow Graph - IRCFG) ซึ่งอยู่ในรูปแบบของกราฟควบคุมกระแสที่แสดงลำดับของเมจเสจในแผนภาพซีเควนซ์ พร้อมทั้งพัฒนาเครื่องมือสร้างกรณีทดสอบตามแนวคิดที่นำเสนอ เพื่อพิสูจน์ว่าแนวคิดที่นำเสนอนั้นสามารถนำไปใช้ได้จริงและจะช่วยลดระยะเวลาในส่วนของขั้นตอนการออกแบบกรณีทดสอบได้เป็นอย่างมาก

1.2 วัตถุประสงค์งานวิจัย

1.2.1 เพื่อสร้างหลักการในการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์และโอซีแอลตามเกณฑ์ความครอบคลุมในการทดสอบ ทุกเส้นทางไออาร์ซีเอฟจี (All-IRCFG Path Criterion) โดยประยุกต์ใช้กราฟควบคุมกระแสไออาร์ซีเอฟจี (Interprocedural Restricted Control Flow Graph -IRCFG)

1.2.2 เพื่อพัฒนาเครื่องมือสร้างกรณีทดสอบจากแผนภาพซีเควนซ์และโอซีแอล

1.3 ขอบเขตงานวิจัย

1.3.1 ผู้วิจัยใช้แนวคิดการแปลงแผนภาพซีเควนซ์ไปเป็นกราฟควบคุมกระแสไออาร์ซีเอฟจีของ Rountev และคณะ [7] เป็นหลักในการระบุเกณฑ์ความครอบคลุมในการสร้างกรณีทดสอบ

1.3.2 แผนภาพซีเควนซ์ที่งานวิจัยนี้รองรับจะต้องถูกออกแบบโดยใช้คอมบายด์แฟรกเมนต์ (Combined Fragment) ประเภท อัลเทอร์เนทีฟ (Alternative) ออปชัน (Option) และลูป (Loop) เท่านั้น

1.3.3 แผนภาพซีเควนซ์ที่งานวิจัยนี้รองรับ จะต้องมีเมจเสจเริ่มต้นเพียง 1 เมจเสจเท่านั้น

1.3.4 ใช้โอซีแอลในส่วนข้อจำกัดของการดำเนินการ (Constraint on Operation) เป็นส่วนที่กำหนดรายละเอียดข้อมูลเพื่อนำไปใช้ในการสร้างกรณีทดสอบ

1.3.5 แผนภาพซีเควนซ์ที่นำมาใช้ในงานวิจัยจะต้องถูกสร้างโดยโปรแกรมวิชวลพาราแดมสำหรับยูเอ็มแอล 6.0 (Visual Paradigm for UML 6.0) เท่านั้น

1.3.6 เครื่องมือที่ใช้ในการแปลงแผนภาพซีเควนซ์ไปเป็นเอกสารเอกซ์เอ็มแอล (XML Document) ได้แก่โปรแกรมวิชวลพาราแดมสำหรับยูเอ็มแอล 6.0

1.4 ขั้นตอนการดำเนินงานวิจัย

1.4.1 ศึกษาการใช้แผนภาพซีเควนซ์ในการออกแบบระบบ

1.4.2 ศึกษาแนวคิดการแปลงแผนภาพซีเควนซ์ให้อยู่ในรูปของกราฟควบคุมกระแสไออาร์-ซีเอฟจีและสร้างหลักการในการแปลง

1.4.3 ศึกษาเกณฑ์ความครอบคลุมในการทดสอบของแผนภาพซีเควนซ์โดยอ้างอิงจากกราฟควบคุมกระแสไออาร์ซีเอฟจี

1.4.4 ศึกษาความหมายและรูปแบบในการเขียนโอซีแอล

1.4.5 ออกแบบแนวคิดและหลักการในการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์และโอซีแอล

1.4.6 ศึกษาวิธีการแปลงแผนภาพซีเควนซ์ เป็นภาษาเอกซ์เอ็มแอลโดยใช้โปรแกรมวิชวลพาราแดมสำหรับยูเอ็มแอล 6.0

1.4.7 พัฒนาเครื่องมือสำหรับสร้างกรณีทดสอบอัตโนมัติจากแผนภาพซีเควนซ์และโอซีแอล

1.4.8 ประเมินการใช้เครื่องมือและความสามารถของเครื่องมือ

1.4.9 สรุปผลการวิจัยและข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้หลักการในการสร้างกรณีทดสอบโดยยึดตามเกณฑ์ความครอบคลุมในการทดสอบของแผนภาพซีเควนซ์

1.5.2 ช่วยลดระยะเวลาและค่าใช้จ่ายในการสร้างกรณีทดสอบที่จะนำไปใช้ในการดำเนินการทดสอบโปรแกรม

1.6 นิยามคำศัพท์สำคัญ

1.6.1 เกณฑ์ความครอบคลุม (Coverage Criteria) คือกลุ่มของกฎที่ใช้ตัดสินว่าชุดของกรณีทดสอบนั้นเพียงพอในการทดสอบโปรแกรมหรือไม่ เช่น ครอบคลุมเส้นทาง (Path Coverage) ครอบคลุมส่วนการตัดสินใจ (Branch Coverage) เป็นต้น [16]

1.6.2 การทดสอบระดับหน่วย (Unit Testing Level) เป็นการทดสอบการทำงานภายในหน่วยของซอฟต์แวร์ซึ่งเป็นองค์ประกอบที่เล็กที่สุดและไม่สามารถถูกแบ่งออกเป็นองค์ประกอบย่อยได้อีก

1.6.3 การทดสอบระดับบูรณาการ (Integration Testing Level) เป็นการทดสอบการทำงานร่วมกันของหน่วยของซอฟต์แวร์ ที่ได้ผ่านการทดสอบในระดับหน่วยมาแล้ว เพื่อประเมินว่าสามารถทำงานร่วมกันได้ถูกต้องตามหน้าที่โดยรวมที่ต้องการหรือไม่

1.6.4 การทดสอบแบบกล่องขาว (White-box Testing) เป็นเทคนิคในการทดสอบที่คำนึงถึงกลไกการทำงานภายในของระบบหรือองค์ประกอบ [17]

1.6.5 ผู้ทดสอบ (Tester) คือบุคคลที่มีบทบาทในการดำเนินการทดสอบและหรือออกแบบกรณีทดสอบด้วยวัตถุประสงค์เพื่อค้นหาข้อผิดพลาดที่มีอยู่ในซอฟต์แวร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เมื่อปี พ.ศ. 2548 Rountev และคณะ [7] ได้นำเสนองานวิจัยที่มีการกำหนดกลุ่มของเกณฑ์ความครอบคลุม (Coverage Criteria) ในการทดสอบการทำงานระหว่างกันของอ็อบเจกต์ โดยยึดจากแผนภาพซีเควนซ์ที่สร้างย้อนกลับมาจากซอร์สโค้ด (Reverse-engineered Sequence Diagram) พร้อมทั้งนำเสนอโครงสร้างข้อมูลซึ่งแสดงในรูปของกราฟควบคุมกระแส (Control Flow Graph) ที่เรียกว่าไอการ์ซีเอฟจี (Interprocedural Restricted Control-Flow Graph-IRCFG) เพื่อนำมาใช้ในการกำหนดกลุ่มของเกณฑ์ความครอบคลุมได้อย่างเป็นระบบ

ผู้วิจัยเห็นว่ากราฟกระแสไอการ์ซีเอฟจีที่ได้นำเสนอไว้ในงานวิจัยดังกล่าวน่าจะมีประโยชน์เป็นอย่างมากเมื่อนำมาใช้ร่วมกันกับการออกแบบและสร้างกรณีทดสอบ เนื่องจากกราฟควบคุมกระแสนี้ได้ถูกสร้างขึ้นเพื่อใช้ในการอธิบายกลุ่มของเกณฑ์ความครอบคลุมได้อย่างเป็นระบบ ซึ่งการออกแบบและสร้างกรณีทดสอบนั้นจะต้องคำนึงถึงเกณฑ์ความครอบคลุมในการทดสอบเหล่านี้ด้วย งานวิจัยนี้จึงสนใจที่จะนำเสนอหลักการในการสร้างกรณีทดสอบ (Test Case) จากแผนภาพซีเควนซ์ (Sequence Diagram) และโอซีแอล (Object Constraint Language - OCL) โดยประยุกต์ใช้กราฟควบคุมกระแสไอการ์ซีเอฟจีของ Rountev และคณะ [7] โดยกรณีทดสอบที่ได้จากงานวิจัยนี้ได้ถูกสร้างขึ้นเพื่อนำไปใช้ในการทดสอบการทำงานระหว่างกันของอ็อบเจกต์ในซอฟต์แวร์เชิงวัตถุ (Object-Oriented System) ในระดับการทดสอบแบบบูรณาการ (Integration Testing)

ทฤษฎีที่เกี่ยวข้องกับงานวิจัยนี้ได้แก่ การทดสอบซอฟต์แวร์ (Software Testing) เกณฑ์ความครอบคลุมของแผนภาพยูเอ็มแอล (UML-Based Coverage Criteria) แผนภาพซีเควนซ์ กราฟควบคุมกระแสไอการ์ซีเอฟจีและเกณฑ์ความครอบคลุม และภาษาข้อจำกัดเชิงวัตถุหรือโอซีแอล

2.1 การทดสอบซอฟต์แวร์ (Software Testing)

การทดสอบซอฟต์แวร์เป็นกระบวนการของการวิเคราะห์และประเมินซอฟต์แวร์เพื่อตรวจสอบความถูกต้องและระบุข้อผิดพลาดที่มีอยู่ในซอฟต์แวร์ [1] ซึ่งเป็นวิธีสำคัญที่ใช้ในการประเมินคุณภาพของซอฟต์แวร์และถือเป็นหนึ่งในกระบวนการของการตรวจทาน (Verification) และการตรวจรับ (Validation) หรือ วีแอนด์วี (V&V) [5]

การตรวจทานเป็นกระบวนการประเมินระบบหรือส่วนประกอบ (Component) เพื่อตัดสินว่าผลิตภัณฑ์ (Product) ที่ได้จากขั้น (Phase) ต่าง ๆ ในกระบวนการพัฒนาเป็นไปตามเงื่อนไขที่ถูกกำหนดไว้ในตอนเริ่มต้นของขั้นนั้น ๆ หรือไม่ [17]

การตรวจรับเป็นกระบวนการประเมินระบบหรือส่วนประกอบ ระหว่างหรือหลังจากจบกระบวนการพัฒนาเพื่อตัดสินว่าตรงตามความต้องการของลูกค้าที่ได้กำหนดไว้หรือไม่ (Requirement Specification) [17]

2.1.1 กระบวนการทดสอบซอฟต์แวร์ (Software Testing Process) ประกอบด้วย 4 ขั้นตอนหลัก [3] ได้แก่ การวางแผนการทดสอบ (Test Planning) การออกแบบกรณีทดสอบ (Test Case Design) การดำเนินการทดสอบ (Test Execution) และการรวบรวมและประเมินผลลัพธ์ (Resultant Data Collection and Evaluation) โดยรายละเอียดในแต่ละขั้นตอนมีดังต่อไปนี้

1. การวางแผนการทดสอบ (Test Planning) เป็นขั้นตอนที่ดำเนินการในช่วงแรกเริ่มของโครงการ ผู้ทดสอบ (Tester) จะเข้าไปมีส่วนร่วมในการทบทวนความต้องการของลูกค้า เพื่อให้แน่ใจว่าความต้องการมีความถูกต้อง (Correctness) และสามารถทดสอบได้ (Testable) [1] หลังจากความต้องการได้ถูกทบทวนและปรับแก้ให้ถูกต้องแล้ว ผู้มีหน้าที่ในการวางแผนการทดสอบจะกำหนดรายละเอียดต่าง ๆ ที่จำเป็นต้องใช้ในการทดสอบและบันทึกลงในเอกสารชื่อ แผนการทดสอบ (Test Plan) ซึ่งรายละเอียดในเอกสารนี้ประกอบด้วย ทรัพยากรที่ต้องใช้ในการทดสอบ สภาพแวดล้อมที่ต้องมีเพื่อใช้ในการทดสอบ เครื่องมือในการทดสอบ และกรณีทดสอบเพื่อตรวจสอบกับความต้องการ

2. การออกแบบกรณีทดสอบ (Test Case Design) เป็นขั้นตอนการออกแบบและสร้างกรณีทดสอบต่าง ๆ สำหรับการทดสอบโปรแกรม โดยการออกแบบกรณีทดสอบต้องจัดทำขึ้นเพื่อให้สามารถพบข้อผิดพลาดได้มากที่สุดเท่าที่จะสามารถพบได้ หลักสำคัญของการออกแบบคือกรณีทดสอบต่าง ๆ ต้องครอบคลุมทุกลักษณะของโปรแกรม [3] และเป็นไปตามความต้องการในการทดสอบซึ่งได้กำหนดไว้ในแผนการทดสอบ

กรณีทดสอบเป็นชุดของค่าของข้อมูลทดสอบ (Input Value) ผลลัพธ์คาดหวัง (Expected Results) และข้อมูลที่จำเป็นซึ่งทำให้ระบบอยู่ในสถานะที่เหมาะสมสำหรับค่าของข้อมูลทดสอบ [6] ซึ่งนำไปดำเนินการกับส่วนของโปรแกรมที่ต้องการทดสอบด้วยวัตถุประสงค์เพื่อที่จะทำให้เกิดความล้มเหลวและตรวจพบข้อผิดพลาด [15] โดยกรณีทดสอบถูกสร้างขึ้นตามเงื่อนไขหรือความต้องการในการทดสอบที่ได้กำหนดไว้ในแผนการทดสอบ Kaner, Falk และ Nguyen อ้างอิงจาก Pressman [3] ได้แนะนำคุณลักษณะของกรณีทดสอบที่ดีไว้ดังต่อไปนี้

- กรณีทดสอบที่ดีมีความน่าจะเป็นสูงในการหาความผิดพลาดพบ การที่ผู้ทดสอบจะบรรลุเป้าหมายในการออกแบบกรณีทดสอบที่ดีได้ ผู้ทดสอบจำเป็นต้องมีความเข้าใจในตัวซอฟต์แวร์เป็นอย่างดี และพยายามสร้างสถานการณ์ที่ทำให้ซอฟต์แวร์นั้นล้มเหลว
- กรณีทดสอบที่ดีไม่ควรซ้ำซ้อน เนื่องจากทรัพยากรและเวลาในการทดสอบมีจำกัด จึงไม่มีประโยชน์ที่จะใช้กรณีทดสอบที่มีจุดประสงค์เดียวกับกรณีทดสอบอื่น ๆ นั่นคือ ทุก ๆ กรณีทดสอบควรมีวัตถุประสงค์ที่แตกต่างกัน
- กรณีทดสอบที่ดีควรเป็นตัวที่คิดมาดีที่สุดในกลุ่มในบรรดาตัวทดสอบที่มีวัตถุประสงค์คล้าย ๆ กัน ผู้ทดสอบควรคัดสรรชุดทดสอบย่อยเพื่อประหยัดเวลาและทรัพยากร กรณีทดสอบที่ถูกคัดสรรควรมีความเป็นไปได้สูงที่จะเปิดเผยความผิดพลาดทั้งหมดในกลุ่มนี้
- กรณีทดสอบที่ดีไม่ควรเป็นทั้งธรรมดาเกินไปหรือซับซ้อนเกินไป การผสมผสานชุดของกรณีทดสอบหลาย ๆ ชุดเข้าด้วยกันเป็นหนึ่งกรณีทดสอบ อาจก่อให้เกิดผลข้างเคียงที่บดบังความผิดพลาดได้ โดยทั่วไปแล้ว กรณีทดสอบแต่ละตัวควรถูกใช้งานแยกกัน

3. **การดำเนินการทดสอบ** (Test Execution) เป็นขั้นตอนที่ผู้ทดสอบดำเนินการทดสอบซอฟต์แวร์ตามเงื่อนไขที่ได้กำหนดไว้ในแผนการทดสอบ โดยข้อมูลที่ใช้ในการทดสอบมาจากกรณีทดสอบที่ได้ถูกสร้างขึ้นจากขั้นตอนที่แล้ว เมื่อดำเนินการทดสอบในแต่ละกรณีแล้ว ผู้ทดสอบจะบันทึกผลที่ได้จากการทดสอบจริง (Actual Result) เพื่อนำมาเปรียบเทียบว่าตรงกับผลลัพธ์คาดหวังหรือไม่

4. **การรวบรวมและประเมินผลลัพธ์** (Resultant Data Collection and Evaluation) เป็นขั้นตอนการประเมินและตรวจสอบผลที่ได้จากการดำเนินการทดสอบว่าสามารถยอมรับได้หรือไม่ และส่งข้อผิดพลาดที่พบให้กับทีมพัฒนาในการแก้ไขข้อผิดพลาดนั้นให้ถูกต้องก่อนการส่งมอบซอฟต์แวร์ และนำข้อมูลที่สรุปไว้ไปใช้เป็นแนวทางในการปรับปรุงการทดสอบในครั้งถัดไปให้ดียิ่งขึ้น

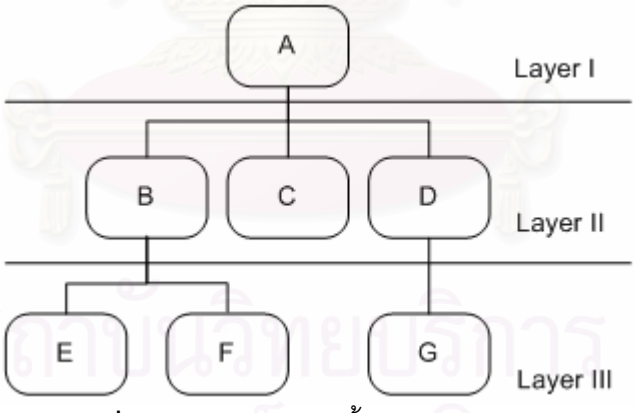
2.1.2 **ระดับของการทดสอบซอฟต์แวร์** (Level of Software Testing) การทดสอบซอฟต์แวร์สามารถแบ่งออกเป็น 6 ระดับ [5] ดังต่อไปนี้

1. **การทดสอบระดับหน่วย** (Unit Testing) เป็นการทดสอบการทำงานภายในหน่วยของซอฟต์แวร์หรือกลุ่มของหน่วยที่เกี่ยวข้องกัน [17] หน่วย (Unit) เป็นองค์ประกอบที่เล็กที่สุดของซอฟต์แวร์ซึ่งไม่สามารถถูกแบ่งออกเป็นองค์ประกอบย่อยได้อีก การทดสอบระดับหน่วยจะเจาะจงทดสอบด้านตรรกะ กระบวนการทำงานภายในและโครงสร้างข้อมูลภายในขอบเขตของ

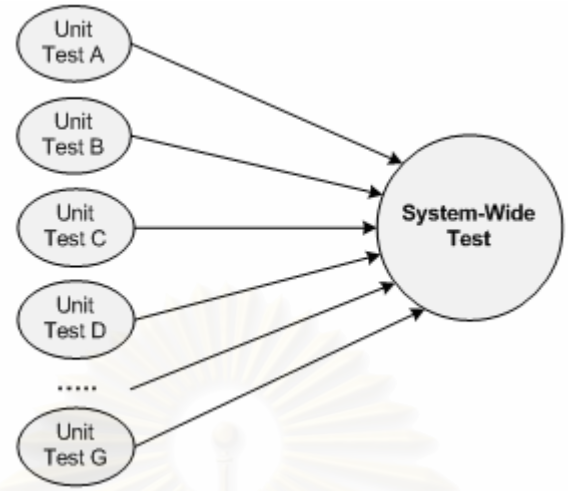
องค์ประกอบ [3] มอดูล (Module) ถือเป็นหน่วยที่เล็กที่สุดของซอฟต์แวร์แบบสัญนิยม (Conventional software) ซึ่งเป็นกลุ่มของฟังก์ชันที่ถูกแบ่งตามหน้าที่การทำงาน ในขณะที่หน่วยสำหรับซอฟต์แวร์เชิงวัตถุได้แก่ คลาส (Class) ซึ่งรวมข้อมูลและกลุ่มของฟังก์ชันเข้าไว้ด้วยกัน [2] การทดสอบระดับหน่วยของซอฟต์แวร์แบบสัญนิยมจะเน้นที่รายละเอียดการทำงานหรืออัลกอริทึมของมอดูล ขณะที่การทดสอบคลาสสำหรับซอฟต์แวร์เชิงวัตถุเป็นการทดสอบพฤติกรรมในแต่ละสถานะของคลาส

2. **การทดสอบระดับบูรณาการ** (Integration Testing) เป็นการทดสอบการทำงานร่วมกันของหน่วยที่ได้ผ่านการทดสอบในระดับหน่วยมาแล้ว เพื่อประเมินว่าสามารถทำงานร่วมกันได้ถูกต้องตามหน้าที่โดยรวมดังที่ต้องการหรือไม่ จากรูปที่ 2-1 แสดงตัวอย่างของหน่วยในแต่ละลำดับชั้น เพื่อใช้ในการยกตัวอย่างของแต่ละกลยุทธ์ที่ใช้ในการทดสอบระดับบูรณาการซึ่งมีด้วยกัน 4 กลยุทธ์ [15] ได้แก่

- **การทดสอบแบบบิ๊กแบง** (Big Bang Testing) เป็นการรวมทุกหน่วยที่ผ่านการทดสอบในระดับหน่วยมาแล้ว เข้าไว้ด้วยกันทั้งหมดและทดสอบการทำงานร่วมกันภายในครั้งเดียว การรวมหน่วยจากรูปที่ 2-1 โดยใช้กลยุทธ์นี้ถูกแสดงไว้ดังรูปที่ 2-2

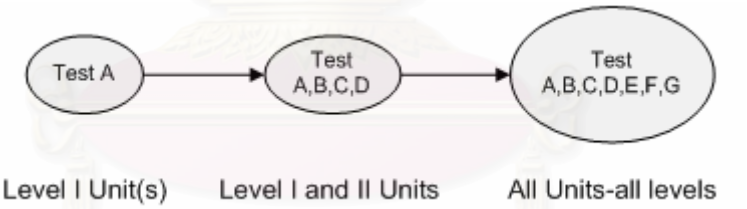


รูปที่ 2-1 แสดง 3 ลำดับชั้นของแต่ละหน่วย [15]



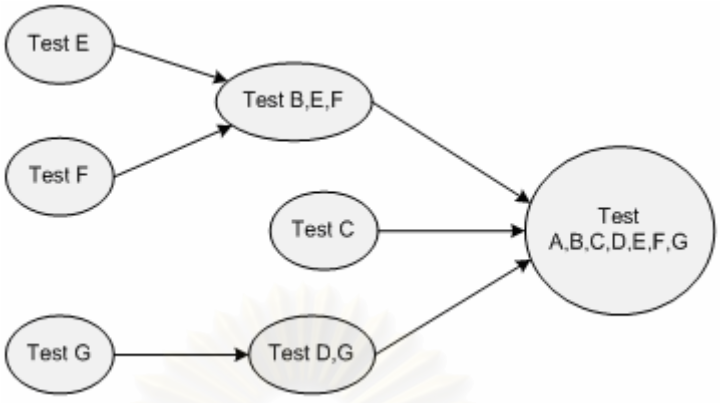
รูปที่ 2-2 แสดงวิธีการรวมแต่ละหน่วยโดยใช้กลยุทธ์การทดสอบแบบบิ๊กแบง [18]

- **การทดสอบแบบบนลงล่าง** (Top-down Testing) เป็นการรวมแต่ละหน่วยจากลำดับชั้นบนสุดแล้วทดสอบ จากนั้นจึงรวมหน่วยของลำดับชั้นถัดลงมาเข้าไว้ด้วยกันแล้วทดสอบ กลยุทธ์การทดสอบในลักษณะนี้จะทำซ้ำเช่นนี้ไปเรื่อยๆ จนกระทั่งถึงลำดับชั้นที่อยู่ล่างสุด และทุกหน่วยได้ถูกทดสอบร่วมกัน การรวมหน่วยจากรูปที่ 2-1 โดยใช้กลยุทธ์นี้ถูกแสดงไว้ดังรูปที่ 2-3



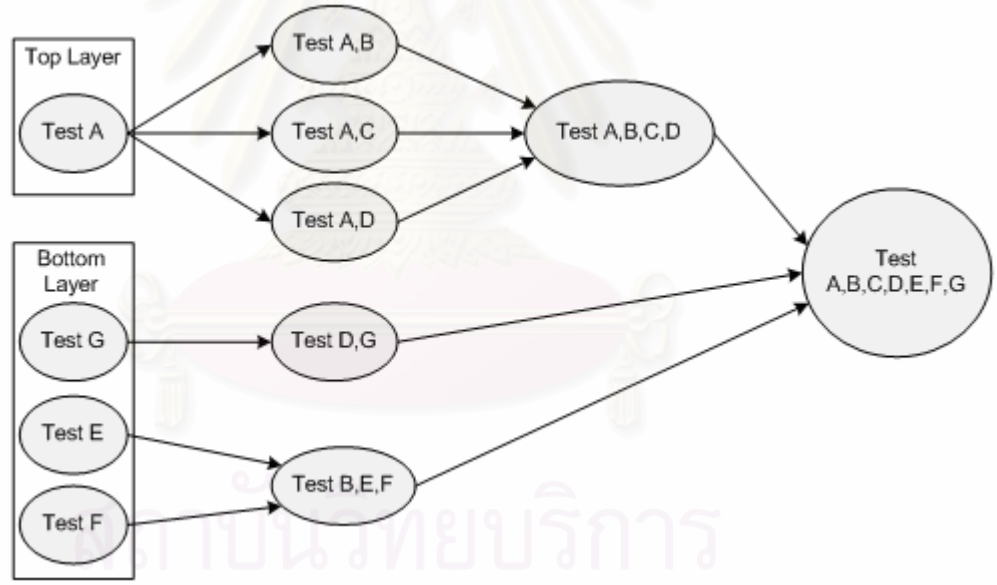
รูปที่ 2-3 แสดงวิธีการรวมแต่ละหน่วยโดยใช้กลยุทธ์การทดสอบแบบบนลงล่าง [18]

- **การทดสอบแบบล่างขึ้นบน** (Bottom-up Testing) จะทดสอบหน่วยในชั้นล่างสุดแยกกัน จากนั้นจึงรวมกับหน่วยในชั้นถัดไปแล้วทดสอบ หากหน่วยย่อยสองหน่วยถูกทดสอบร่วมกันจะเรียกว่า ดับเบิ้ลเทส (Double Test) การทดสอบหน่อยย่อยสามหน่วยด้วยกันเรียกว่า ทริเปิ้ลเทส (Triple Test) และการทดสอบหน่วยย่อยสี่หน่วยพร้อมกันเรียกว่า ควอดดรัปเปิ้ลเทส (Quadruple Test) การรวมหน่วยจากรูปที่ 2-1 โดยใช้กลยุทธ์นี้ถูกแสดงไว้ดังรูปที่ 2-4



รูปที่ 2-4 แสดงวิธีการรวมแต่ละหน่วยโดยใช้กลยุทธ์การทดสอบแบบล่างขึ้นบน [18]

- **การทดสอบแบบแซนด์วิช** (Sandwich Testing) เป็นกลยุทธ์การทดสอบที่ผสมผสานกลยุทธ์แบบบนลงล่างและล่างขึ้นบนเข้าด้วยกัน การรวมหน่วยจากรูปที่ 2-1 โดยใช้กลยุทธ์นี้ถูกแสดงไว้ดังรูปที่ 2-5



รูปที่ 2-5 แสดงวิธีการรวมแต่ละหน่วยย่อยโดยใช้กลยุทธ์การทดสอบแบบแซนด์วิช [15]

3. **การทดสอบระบบหรือการทดสอบการทำงาน** (System Testing) เป็นการตรวจสอบซอฟต์แวร์ว่าสามารถทำงานได้ถูกต้อง เป็นไปตามความต้องการของผู้ใช้งานและข้อกำหนดของซอฟต์แวร์หรือไม่ โดยประกอบไปด้วย 6 การทดสอบย่อยดังนี้

- การทดสอบหน้าที่การทำงาน (Functional Testing) เป็นการทดสอบการทำงานของซอฟต์แวร์เพื่อตรวจสอบว่าการทำงานนั้นเป็นไปตามความต้องการของผู้ใช้งานโดยใช้เทคนิคการทดสอบแบบกล่องดำ (Black-box Testing)

- การทดสอบแรงเครียด (Stress Testing) เป็นการทดสอบซอฟต์แวร์ภายใต้สภาวะการใช้งานอย่างหนัก เพื่อตรวจหาความสามารถสูงสุดของซอฟต์แวร์ภายใต้ทรัพยากรที่กำหนด ใช้กับระบบในลักษณะที่ใช้ทรัพยากรจำนวนมากผิดปกติ หรือบ่อยมาก หรือปริมาณมาก [3]
- การทดสอบประสิทธิภาพ (Performance Testing) เป็นการทดสอบประสิทธิภาพซอฟต์แวร์ตามที่ระบุในข้อกำหนดของซอฟต์แวร์ ด้วยการตรวจสอบจากเวลาตอบสนองการทำงานของซอฟต์แวร์ภายใต้สภาวะการใช้งานปกติ
- การทดสอบความปลอดภัย (Security Testing) เป็นการทดสอบระบบรักษาความปลอดภัยของซอฟต์แวร์จากผู้ไม่มีสิทธิใช้งานซอฟต์แวร์
- การทดสอบการกู้คืน (Recovery Testing) เป็นการทดสอบระบบที่ทำให้ซอฟต์แวร์ทำงานล้มเหลวในหลากหลายสถานการณ์และตรวจสอบว่าการกู้คืนได้เกิดขึ้นอย่างเหมาะสม [3]
- การทดสอบพื้นหลัง (Background Testing) เป็นการทดสอบว่าระบบสามารถรองรับการทำงานร่วมกันกับระบบอื่นได้ดีเพียงใด การทดสอบในลักษณะนี้สำคัญเมื่อระบบจะต้องใช้ทรัพยากรร่วมกัน

4. **การทดสอบเพื่อการยอมรับ (Acceptance Testing)** เป็นการทดสอบที่เป็นทางการเพื่อตัดสินว่าระบบเป็นไปตามเงื่อนไขข้อยอมรับ (เงื่อนไขของระบบที่จะต้องตอบสนองความต้องการและถูกยอมรับโดยลูกค้า) หรือไม่ และเพื่อให้ลูกค้าตัดสินว่ายอมรับระบบหรือไม่ [17]

5. **การทดสอบแบบถดถอย (Regression Testing)** เมื่อเกิดการเปลี่ยนแปลงแก้ไขซอฟต์แวร์หรือมอดูลใหม่ ๆ ถูกเพิ่มเข้ามาในซอฟต์แวร์ จะต้องดำเนินการทดสอบซอฟต์แวร์ดังกล่าวซ้ำโดยใช้ชุดของกรณีทดสอบเดิมที่เคยทำมาแล้ว เพื่อให้มั่นใจว่าการเปลี่ยนแปลงที่เกิดขึ้นไม่ส่งผลกระทบที่ไม่ดีหรือทำความเสียหายให้กับการทำงานเดิมที่มีอยู่ก่อนแล้ว

6. **การทดสอบแบบอัลฟ่าและเบต้า (Alpha and Beta Testing)** หากซอฟต์แวร์ที่พัฒนาเป็นผลิตภัณฑ์สำหรับผู้ใช้หลาย ๆ ราย จะไม่สามารถทดสอบการส่งมอบอย่างเป็นทางการกับผู้ใช้ทุกรายได้ การทดสอบในลักษณะนี้มีวัตถุประสงค์เพื่อค้นหาข้อผิดพลาดที่ผู้ใช้งานเท่านั้นที่อาจหาพบได้ โดยรายละเอียดของแต่ละการทดสอบมีดังต่อไปนี้ [3]

- การทดสอบแบบอัลฟ่า ดำเนินการ ณ สถานที่ของผู้พัฒนาระบบโดยผู้ทดสอบคือผู้ใช้ (End User) ซึ่งซอฟต์แวร์จะถูกใช้ในลักษณะที่ใช้จริง โดยมีผู้พัฒนาเฝ้า

สังเกตการณ์การใช้งานตามปกติของผู้ใช้ และบันทึกข้อผิดพลาดรวมทั้งปัญหาในการใช้งานอื่น ๆ การทดสอบแบบอัลฟ่ากระทำภายใต้สิ่งแวดล้อมที่มีการควบคุม

- การทดสอบแบบเบต้า ดำเนินการ ณ สถานที่ของผู้ใช้ ผู้พัฒนามักไม่ได้อยู่ด้วย ดังนั้น การทดสอบแบบเบต้าจึงเป็นการประยุกต์ใช้งานจริงของซอฟต์แวร์ ภายใต้สิ่งแวดล้อมที่ผู้พัฒนาควบคุมไม่ได้ ผู้ใช้บันทึกปัญหาทุกอย่างที่พบขณะทดสอบและรายงานสิ่งเหล่านี้ให้ผู้พัฒนาทราบเป็นระยะ ๆ จากรายงานการทดสอบแบบเบต้า ผู้พัฒนาซอฟต์แวร์จะปรับปรุงแก้ไขซอฟต์แวร์ และเตรียมพร้อมเพื่อวางผลิตภัณฑ์รุ่นถัดไปให้ลูกค้าทุกกลุ่ม

2.1.3 **เทคนิคในการทดสอบซอฟต์แวร์ (Software Testing Technique)** มีอยู่ด้วยกันสองเทคนิคหลัก ได้แก่ การทดสอบแบบกล่องดำ (Black-box Testing) และการทดสอบแบบกล่องขาว (White-box Testing) ซึ่งความหมายและความแตกต่างของทั้งสองเทคนิคในการทดสอบมีดังต่อไปนี้ [3,5]

2.1.3.1 **การทดสอบแบบกล่องดำ (Black-box Testing)** หรือการทดสอบฟังก์ชันนัล (Functional Testing) เป็นการทดสอบโดยไม่คำนึงถึงกลไกการทำงานภายในของระบบหรือองค์ประกอบ การทดสอบแบบกล่องดำทำขึ้นเพื่อต้องการตรวจสอบว่าแต่ละการทำงาน (Function) ของซอฟต์แวร์สามารถดำเนินการได้อย่างถูกต้อง เป็นไปตามที่กำหนดไว้ในเอกสารความต้องการ (Requirement Specification) และในขณะเดียวกันก็เป็นการค้นหาข้อผิดพลาดในแต่ละการทำงานอีกด้วย Pressman [3] ได้ระบุประเภทของความผิดพลาดที่สามารถตรวจพบด้วยการทดสอบแบบกล่องดำดังต่อไปนี้ (1) หน้าที่การทำงาน (Functionality) ผิดพลาดหรือหายไป (2) ความผิดพลาดจากตัวต่อประสาน (Interface) (3) ความผิดพลาดในโครงสร้างข้อมูลหรือการเข้าถึงฐานข้อมูลภายนอก (4) ความผิดพลาดในพฤติกรรมหรือประสิทธิภาพและ (5) ความผิดพลาดจากการเริ่มต้นหรือการจบงาน วิธีที่ใช้ในการทดสอบแบบกล่องดำมีดังต่อไปนี้

1. **การแบ่งส่วนสมมูล (Equivalence Partitioning - EP)** เป็นวิธีที่แบ่งข้อมูลนำเข้า (Input) หลักของโปรแกรมออกเป็นชั้นสมมูล (Equivalence Class) โดยกลุ่มของข้อมูลภายในแต่ละชั้นสมมูลจะต้องได้รับการดำเนินการภายในมอดูลที่ต้องการทดสอบเหมือนกันและให้ผลลัพธ์ที่เหมือนกัน ซึ่งในการแบ่งต้องคำนึงถึงสถานะภายในขอบเขต (Valid) และสถานะภายนอกขอบเขต (Invalid) ของข้อมูลนำเข้า กรณีทดสอบจะถูกออกแบบตามข้อมูลนำเข้าภายในชั้นสมมูล ขั้นตอนในการสร้างกรณีทดสอบจากวิธีการแบ่งส่วนสมมูลมีดังต่อไปนี้ [19]

ขั้นตอนที่ 1 กำหนดชั้นสมมูลและให้ชื่อกับกลุ่มนั้น ๆ โดยในการกำหนดกลุ่มสมมูลมีข้อแนะนำดังต่อไปนี้ [3]

- ถ้าเงื่อนไขข้อมูลระบุเป็นช่วง ให้นิยามกลุ่มค่าภายในขอบเขตหนึ่งกลุ่มและกลุ่มค่าภายนอกขอบเขตสองกลุ่ม
- ถ้าเงื่อนไขข้อมูลเข้าต้องเป็นค่าเฉพาะหนึ่ง ให้นิยามกลุ่มค่าภายในขอบเขตหนึ่งกลุ่มและกลุ่มค่าภายนอกขอบเขตหนึ่งกลุ่ม
- ถ้าเงื่อนไขข้อมูลเข้าระบุสมาชิกของเซต ให้นิยามกลุ่มสมมูลเป็นกลุ่มค่าภายในขอบเขตหนึ่งกลุ่มและภายนอกขอบเขตหนึ่งกลุ่ม
- ถ้าเงื่อนไขข้อมูลนำเข้าเป็นบูลีน ให้นิยามกลุ่มภายในขอบเขตและกลุ่มภายนอกขอบเขตอย่างละหนึ่งกลุ่ม

ขั้นตอนที่ 2 สำหรับชั้นสมมูลที่เป็นกลุ่มค่าภายในขอบเขต (Valid Equivalence Partition) ให้สร้างกรณีทดสอบให้คลุมทุกชั้นสมมูล

ขั้นตอนที่ 3 สำหรับชั้นสมมูลที่เป็นกลุ่มค่าภายนอกขอบเขต (Invalid Equivalence Partition) ให้สร้างกรณีทดสอบคลุมชั้นสมมูลเดียวเท่านั้น

2. **การวิเคราะห์ค่าขอบเขต** (Boundary Value Analysis - BVA) เป็นวิธีที่ใช้ในการออกแบบกรณีทดสอบโดยเสริมกับวิธีการแบ่งส่วนสมมูล คือให้เลือกค่าที่จะนำไปใช้ในการทดสอบจากปลายขอบข้อมูลของกลุ่มสมมูลแทนที่จะเลือกค่าใด ๆ ในกลุ่มสมมูล ข้อแนะนำสำหรับการวิเคราะห์ค่าขอบเขตจะคล้ายกับการแบ่งส่วนสมมูล โดยมีแนวทางดังต่อไปนี้ [3]

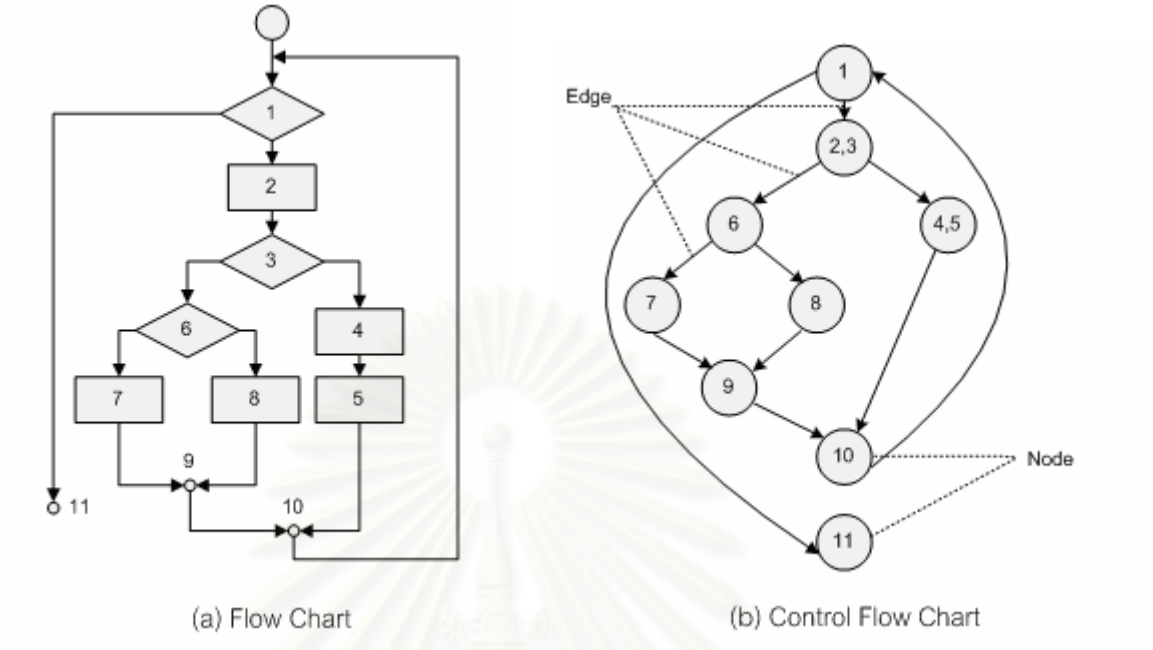
- ถ้าเงื่อนไขข้อมูลนำเข้าระบุช่วงที่ปิดล้อมด้วยค่า a และ b กรณีทดสอบควรประกอบไปด้วยค่า a และ b และค่าที่มากกว่าและน้อยกว่าค่า a และ b เล็กน้อย
- ถ้าเงื่อนไขข้อมูลนำเข้าระบุค่าหลาย ๆ ค่าแล้ว กรณีทดสอบควรจะสามารถทำงานกับค่าน้อยที่สุดและค่ามากที่สุด โดยทำการทดสอบกับค่าที่อยู่ใกล้เคียงกับค่าน้อยที่สุดและค่ามากที่สุดนั้นด้วย
- ประยุกต์ใช้ข้อแนะนำสองข้อเบื้องต้นกับเงื่อนไขข้อมูลนำออก ตัวอย่างเช่น สมมติให้ตารางความดันต่ออุณหภูมิเป็นข้อมูลออกจากโปรแกรมวิเคราะห์ทางวิศวกรรม กรณีทดสอบควรสร้างรายงานที่ทำงานกับค่าขอบเขตของโครงสร้างข้อมูลนั้น
- ถ้าโครงสร้างข้อมูลภายในมีขอบเขตที่ระบุไว้ล่วงหน้า เช่น อาเรย์ ให้ออกแบบกรณีทดสอบที่ทำงานกับค่าขอบเขตของโครงสร้างข้อมูลนั้น

3. **การวิเคราะห์เหตุและผล** (Cause-Effect Analysis) เป็นวิธีการทดสอบที่เหมาะสำหรับการทดสอบการทำงานที่มีการผสมผสานของข้อมูลนำเข้าที่จะต้องถูกทดสอบร่วมกัน วิธีการนี้ใช้ตารางการตัดสินใจ (Decision Table) และต้นไม้การตัดสินใจ (Decision Tree) มาใช้ในการออกแบบกรณีทดสอบ

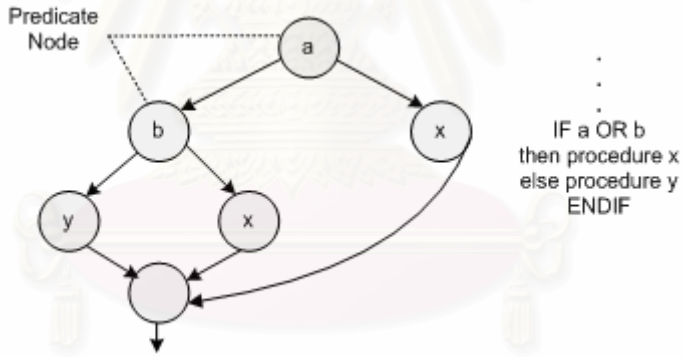
2.1.3.2 **การทดสอบแบบกล่องขาว** (White-box Testing) หรือการทดสอบแบบกล่องแก้ว (Glass-box Testing) เป็นการทดสอบโดยคำนึงถึงกลไกการทำงานภายในของระบบหรือองค์ประกอบ [17] ผู้ทดสอบได้เห็นการทำงานภายในของซอฟต์แวร์ ตรรกะและโครงสร้างของซอร์สโค้ด (Source Code) ผู้พัฒนาจะออกแบบกรณีทดสอบซึ่งสามารถ (1) รับประกันได้ว่าทุกเส้นทางอิสระ (Independent Path) ภายในมอดูลได้รับการทดสอบอย่างน้อยหนึ่งครั้ง (2) ทดสอบการตัดสินใจทางตรรกะทั้งในกรณีที่เป็นจริงและเท็จ (3) ทดสอบลูปที่ขอบเขตและที่การทำงานภายใน และ(4) ทดสอบโครงสร้างของข้อมูลภายในเพื่อตรวจสอบความถูกต้อง [3] ซึ่งวิธีที่ใช้ในการทดสอบแบบกล่องขาวมีดังต่อไปนี้

1. **เส้นทางพื้นฐาน** (Basic Path) เป็นวิธีการทดสอบแบบกล่องขาวซึ่งถูกนำเสนอเป็นครั้งแรกโดย Tom McCabe [3] วิธีในการทดสอบนี้ทำให้แน่ใจว่าทุกเส้นทางอิสระของซอร์สโค้ดภายในมอดูลได้ถูกทดสอบแล้วอย่างน้อยหนึ่งครั้ง เพื่อทดสอบความสลับซับซ้อนทางตรรกะของโปรแกรม

จากรูปที่ 2-6 แสดงตัวอย่างของการแปลงผังงาน (Flow Chart) ในรูปที่ 2-6(a) มาเป็นกราฟควบคุมกระแส (Control Flow Graph) ดังรูปที่ 2-6(b) เพื่อให้สามารถเห็นการควบคุมทางตรรกะ โดยสัญลักษณ์รูปวงกลมในกราฟควบคุมกระแสเรียกว่า โหนด (Node) ซึ่งแทนประโยคคำสั่งตั้งแต่หนึ่งประโยคคำสั่งขึ้นไป ในกรณีที่มีเงื่อนไขผสม (Compound Condition) คือมีตัวดำเนินการบูลีนอยู่ในคำสั่งเงื่อนไข (Condition Statement) จะต้องแยกแต่ละเงื่อนไขออกมาเป็นโหนดดังรูปที่ 2-7 ซึ่งโหนดเหล่านั้นจะถูกเรียกว่า เพรดิเคตโหนด (Predicate Node)



รูปที่ 2-6 ตัวอย่างของการแปลงผังงานมาเป็นกราฟควบคุมกระแส [3]



รูปที่ 2-7 แสดงรูปแบบของเงื่อนไขผสมในกราฟควบคุมกระแส [3]

เส้นทางอิสระ (Independent Path) คือเส้นทางใด ๆ ที่ผ่านโปรแกรมที่มีประโยคคำสั่ง (Statement) หรือเงื่อนไขใหม่อย่างน้อย 1 ประโยค จากรูปที่ 2-6(b) เส้นทางอิสระต้องวิ่งผ่านเส้นเชื่อม (Edges) อย่างน้อย 1 เส้นที่ยังไม่เคยผ่าน ตัวอย่างของชุดเส้นทางอิสระของกราฟควบคุมกระแสในรูปที่ 2-6(b) คือ

เส้นทางที่ 1 1-11

เส้นทางที่ 2 1-2-3-4-5-10-1-11

เส้นทางที่ 3 1-2-3-6-8-9-10-1-11

เส้นทางที่ 4 1-2-3-6-7-9-10-1-11

เส้นทางที่ 1-4 นั้นประกอบขึ้นเป็นเซตมูลฐาน (Basic Set) หากกลุ่มของกรณีทดสอบถูกออกแบบให้สามารถใช้งานเส้นทางเหล่านี้ได้หมด จะเป็นการรับประกันได้ว่าทุกประโยคในโปรแกรมได้รับการทดสอบอย่างน้อยหนึ่งครั้ง และทุกเงื่อนไขได้รับการทดสอบทั้งด้านจริงและด้านเท็จ ในการหาจำนวนของเส้นทางอิสระที่มากที่สุดนั้นสามารถคำนวณได้จากค่าความซับซ้อน-ไซโคมาตริก (Cyclomatic Complexity) [3] ซึ่งมีวิธีในการหา 3 วิธีได้แก่

1. จากจำนวนขอบเขตของกราฟควบคุมกระแส
2. จากสูตร V(G) = E-N+2

เมื่อ V(G) คือความซับซ้อนไซโคมาตริกของกราฟควบคุมกระแส G โดยที่ E คือจำนวนเส้นเชื่อม และ N คือจำนวนโหนดของกราฟควบคุมกระแส G

3. จากสูตร V(G) = P+1

โดย P คือจำนวนเพรดิเคตโหนดในกราฟควบคุมกระแส G

2. การแบ่งส่วนสมมูล/การวิเคราะห์ค่าขอบเขต (Equivalence Partitioning/Boundary Value Analysis) เป็นวิธีที่สามารถนำมาใช้ในการสร้างกรณีทดสอบสำหรับเทคนิคการทดสอบแบบกล่องขาวได้เช่นกัน โดยวิธีที่ใช้ในการสร้างข้อมูลทดสอบนั้นจะเหมือนกับที่ได้กล่าวไว้ในการทดสอบแบบกล่องดำ แต่ในกรณีของกล่องขาวจะนำมาใช้กับลูป (Loop) หรือเงื่อนไข (Condition) ภายในโปรแกรม และใช้วิธีการวิเคราะห์ค่าขอบเขตเมื่อต้องการให้แน่ใจว่าลูปหรือเงื่อนไขสามารถดำเนินการได้ตามขอบเขตที่ได้กำหนดไว้ [20]

การออกแบบกรณีทดสอบด้วยเทคนิคแบบกล่องขาวที่ดีจะต้องพิจารณากราฟควบคุมกระแส (Control Flow Graph) ของโปรแกรม ซึ่งได้ถูกนำเสนอดังรูปที่ 2-6(b) โดยนำมาใช้พิจารณาหลากหลายแง่มุม เพื่อให้แน่ใจว่าชุดของกรณีทดสอบนั้นเพียงพอสำหรับการทดสอบ โดยความเพียงพอ (Adequacy) ของกรณีทดสอบจะถูกวัดด้วยมาตรวัดที่เรียกว่าความครอบคลุม (Coverage) ซึ่งเป็นสิ่งที่ใช้ในการวัดความสมบูรณ์ของกรณีทดสอบโดยประเภทของความครอบคลุมมีดังต่อไปนี้ [20]

1. ความครอบคลุมของเส้นทาง (Path Coverage) เป็นการวัดสัดส่วนของเส้นทางที่ถูกใช้จากกรณีทดสอบ

2. ความครอบคลุมของประโยคคำสั่ง (Statement Coverage) เป็นการวัดสัดส่วนของประโยคคำสั่งที่ถูกใช้เมื่อกรณีทดสอบได้ถูกดำเนินการในการทดสอบ

3. ความครอบคลุมของส่วนตัดสินใจหรือสาขา (Decision/Branch Coverage) เป็นการวัดจำนวนของนิพจน์บูลีน (Boolean Expression) ของโปรแกรมที่ถูกประเมินทั้งในกรณีที่เป็นจริงและกรณีที่เป็นเท็จ

4. ความครอบคลุมเงื่อนไข (Condition Coverage) เป็นการรายงานถึงผลที่เป็นจริงและเป็นเท็จของแต่ละ นิพจน์บูลีนย่อย (Boolean Sub-expression) ของเงื่อนไขผสมซึ่งประกอบด้วย โหนดเพรดิเคตมากกว่าหนึ่งโหนด

จากรายละเอียดที่ได้กล่าวมาข้างต้น สามารถสรุปถึงข้อกำหนด เทคนิคและผู้ทดสอบในแต่ละระดับของการทดสอบซอฟต์แวร์ซึ่งแสดงไว้ในตารางที่ 2-1

ตารางที่ 2-1 แสดงความสัมพันธ์ของระดับการทดสอบ ข้อกำหนดและเทคนิคที่ใช้ในการทดสอบและผู้ดำเนินการทดสอบในแต่ละระดับ [5]

ระดับของการทดสอบ	ข้อกำหนดที่ใช้การทดสอบ (Specification)	เทคนิคที่ใช้ในการทดสอบ	ผู้ดำเนินการทดสอบ
Unit	Low-Level Design Actual Code Structure	White Box	Developer
Integration	Low-Level Design High-Level Design	White Box Black Box	Developer
Functional and System	High-Level Design Requirement Analysis	Black Box	Tester
Acceptance	Requirement Analysis	Black Box	User
Beta/Alpha	Ad hoc	Black Box	User
Regression	Changed Documentation High-Level Design	White Box Black Box	Developer Tester

ประเภทของข้อผิดพลาดที่พบจากการทดสอบโดยใช้เทคนิคกล่องดำและกล่องขาวมีความแตกต่างกัน แต่ความผิดพลาดในระดับหน่วยอาจถูกพบในการทดสอบแบบกล่องดำได้หากการทดสอบแบบกล่องขาวไม่ได้ถูกดำเนินการอย่างเพียงพอ [20] ดังนั้นข้อผิดพลาดควรที่จะถูกตรวจพบให้ได้มากที่สุดเท่าที่จะทำได้โดยใช้การทดสอบแบบกล่องขาว เนื่องจากผู้ดำเนินการทดสอบ

ด้วยเทคนิคนี้คือผู้พัฒนาซอฟต์แวร์เอง (Developer) จึงเป็นการง่ายที่จะตรวจหาสาเหตุที่ทำให้เกิดข้อผิดพลาดและแก้ไขให้ถูกต้องได้อย่างรวดเร็ว

แผนภาพยูเอ็มแอลเป็นภาษามาตรฐานซึ่งได้ถูกพัฒนาขึ้นเพื่อนำมาใช้ในการออกแบบสร้างซอฟต์แวร์เชิงวัตถุ อีกทั้งยังมีบทบาทสำคัญในกระบวนการของการทดสอบซอฟต์แวร์เชิงวัตถุ [11] ซึ่งได้ถูกสำรวจโดย Williams, C. E [21] และได้สรุปแนวทางในการใช้แผนภาพยูเอ็มแอลต่าง ๆ ในกระบวนการทดสอบพร้อมทั้งข้อผิดพลาดที่อาจพบจากแบบจำลอง (Fault Models) ดังตารางที่ 2-2

ตารางที่ 2-2 แสดงความสัมพันธ์ของแผนภาพยูเอ็มแอลในแต่ละระดับของการทดสอบในกระบวนการทดสอบซอฟต์แวร์ [21]

Test Type	Coverage Criteria	Fault Model	UML Diagram
Unit	Code	Correctness, error handling pre/post condition, invariants	Class and state diagram
Function	Functional	Functional and API behavior, integration issues	Interaction and class diagram
System	Operational Scenarios	Workload contention, synchronous, recovery	Use case, activity and interaction diagram
Regression	Functional	Unexpected behavior from new/changed function	Interaction and class diagram

จากตารางที่ 2-2 จะเห็นได้ว่าแผนภาพยูเอ็มแอลถูกนำมาใช้ในแต่ละระดับของการทดสอบด้วยแผนภาพที่ต่างกัน ทั้งนี้เนื่องจากวัตถุประสงค์ของระดับการทดสอบและวัตถุประสงค์ของแผนภาพในการนำเสนอแง่มุมของระบบจะต้องมีความสัมพันธ์กันนั่นเอง และเนื่องด้วยการเข้ามามีบทบาทในกระบวนการทดสอบ ทำให้ความครอบคลุมที่ได้กล่าวถึงในข้างต้นซึ่งเป็นความครอบคลุมที่ได้นิยมใช้ในการทดสอบซอฟต์แวร์สัญนิยมนั้น ไม่เพียงพอต่อการทดสอบซอฟต์แวร์เชิงวัตถุซึ่งได้ถูกสร้างตามการออกแบบด้วยแผนภาพยูเอ็มแอล จึงได้มีหลายงานวิจัยที่นำเสนอเกณฑ์ความครอบคลุมในการทดสอบโดยอ้างอิงจากแผนภาพยูเอ็มแอล [7,12,13,14,22] ซึ่งรายละเอียดของแต่ละเกณฑ์ความครอบคลุมของแต่ละแผนภาพยูเอ็มแอลนั้นจะถูกนำเสนอในหัวข้อถัดไป

2.2 เกณฑ์ความครอบคลุมของแผนภาพยูเอ็มแอล (UML-Based Coverage Criteria)

จากการทบทวนวรรณในอดีต (Literature Review) ได้พบงานวิจัยหลายงานที่นำเสนอเกณฑ์ความครอบคลุมในการทดสอบจากโดยอ้างอิงจากแผนภาพยูเอ็มแอล และจากรายงานการสำรวจของ McQuillan และ Power [14] สามารถจำแนกเกณฑ์ความคลอบคลุมตามแผนภาพยูเอ็มแอลออกเป็น 5 แผนภาพ ดังต่อไปนี้

2.2.1 เกณฑ์ของแผนภาพคลาส (Class Diagram Criteria)

งานวิจัยของ Andrews และคณะ [12] นำเสนอเกณฑ์ความครอบคลุมแผนภาพคลาสดังต่อไปนี้

1. **เกณฑ์แอสโซซิเอชันเอนด์มัลติพลิซิตี (Association End Multiplicity-AEM Criterion)** เป็นการระบุเซตของตัวแทนภาวะรากซ้ำ (Multiplicity) ซึ่งจะต้องถูกสร้างขึ้นเพื่อใช้ในการทดสอบ วิธีการนี้จะแบ่งกลุ่มของโดเมนภาวะรากซ้ำ (Multiplicity Domain) ออกเป็นชั้นสมมูล (Equivalence Class) จากนั้นจึงเลือกเพียงหนึ่งค่าของแต่ละชั้นสมมูลของที่เป็นตัวแทนของภาวะรากซ้ำ ยกตัวอย่างเช่น คลาส Customer และ คลาส Loyalty Program มีความสัมพันธ์ต่อกันแบบเอ็ม-เอ็ม (m-m) ดังรูปที่ 2-8

รูปที่ 2-8 แสดงความสัมพันธ์แบบ เอ็ม-เอ็ม ระหว่างคลาสในแผนภาพคลาส

จากรูปที่ 2-8 กลุ่มสมมูลของคลาส Customer = {0},{1,…, n-1},{n}

กลุ่มสมมูลของคลาส Loyalty Program = {0},{1,…,m-1},{m}

ค่าที่เป็นไปได้สำหรับกลุ่มของคลาส Customer คือ {0,u,n}

ค่าที่เป็นไปได้สำหรับกลุ่มของคลาส Loyalty Program คือ {0,v,m}

โดยที่ u และ v เป็นค่าที่เลือกมาจากเซต {1,…, n-1} และ {1,…,m-1} ตามลำดับ

จากนั้นจึงทำผลคูณคาร์ทีเซียน (Cartesian Product) ระหว่างคลาส Customer และ Loyalty Program ซึ่งได้เซตดังต่อไปนี้ { (0,0), (0,v), (0,m), (u,0), (u,v), (u,m), (n,0), (n,v), (n,m) }

2. **เกณฑ์เจนเนอรัลไลเซชัน (Generalization-GN Criterion)** เป็นการระบุเซตซึ่งเป็นตัวแทนของคลาสลูกซึ่งจะต้องถูกสร้างจากคลาสแม่ในระหว่างการทดสอบระบบอย่างน้อย 1 ครั้ง

3. **เกณฑ์คลาสแอตทริบิวส์** (Class Attribute-CA Criterion) เป็นการระบุเซตที่อยู่ภายในขอบเขต (Valid) และภายนอกขอบเขต (Invalid) ของแอตทริบิวส์ (Attribute) ซึ่งเป็นการผสมกันระหว่างค่าของตัวแทนแต่ละตัวแปรจากการทำผลคูณคาร์ทีเซียนโดยช่วงของค่า (Value Space) ของตัวแปรแต่ละตัวนั้นได้มาจากการระบุถึงข้อจำกัดของตัวแปรภายในคลาสโดยใช้โอซีแอล

2.2.2 เกณฑ์ของแผนภาพซีเควนซ์ (Sequence Diagram Criteria)

เกณฑ์ความครอบคลุมของแผนภาพซีเควนซ์นั้นถูกนำเสนอไว้ในหลายงานวิจัยดังต่อไปนี้

1. **ครอบคลุมทุกเส้นทาง** (All-Paths-Coverage) คือชุดของเส้นทางข้อความ P ซึ่งรองรับเกณฑ์ครอบคลุมทุกเส้นทาง โดย P ประกอบไปด้วยทุกเส้นทางเมจเสจเริ่มต้นถึงเมจ-เสจสุดท้าย (Start-to-End Message Path) ในแผนภาพซีเควนซ์ ซึ่งเกณฑ์นี้ได้ถูกนำเสนอไว้ในงานวิจัยของ Basanieri และBertolino [22]

2. **ทุกเส้นทางไออาร์ซีเอฟจี** (All-IRCFG-Paths) **ทุกเส้นทางอาร์ซีเอฟจี** (All-RCFG-Path) **ทุกสาขาอาร์ซีเอฟจี** (All-RCFG-Branches) **และทุกสาขาที่ไม่ซ้ำ** (All-Unique-Branches) เกณฑ์เหล่านี้ถูกนำเสนอโดย Rountev และคณะ [7] ด้วยการอ้างอิงจากกราฟควบคุมกระแสไออาร์ซีเอฟจี ที่แสดงให้เห็นถึงชุดของลำดับเมจเสจในแผนภาพซีเควนซ์ ซึ่งในรายงานการสำรวจของ McQuillan และ Power [14] ได้กำหนดคำนิยามของแต่ละเกณฑ์ความครอบคลุมไว้ดังต่อไปนี้

- เกณฑ์ทุกเส้นทางไออาร์ซีเอฟจี (All-IRCFG-Path Criterion) คือชุดของเส้นทางไออาร์ซีเอฟจี P ที่ตอบสนองเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจี (IRCFG-Path) โดยที่ P ประกอบไปด้วยทุกเส้นทางไออาร์ซีเอฟจี
- เกณฑ์ทุกเส้นทางอาร์ซีเอฟจี (All-RCFG-Path Criterion) คือชุดของเส้นทางไออาร์ซีเอฟจี P ที่ตอบสนองเกณฑ์ความครอบคลุมทุกเส้นทางอาร์ซีเอฟจี (RCFG Path) โดยที่ P ประกอบไปด้วยทุกเส้นทางอาร์ซีเอฟจี
- เกณฑ์ทุกสาขาอาร์ซีเอฟจี (All-RCFG-Branch Criterion) คือชุดของเส้นทางไออาร์ซีเอฟจี P ที่ตอบสนองเกณฑ์ความครอบคลุมทุกสาขาอาร์ซีเอฟจี (RCFG Branch) โดยที่ทุกเส้นเชื่อม e ในแต่ละอาร์ซีเอฟจี มีอย่างน้อยหนึ่งเส้นทาง p ในชุดเส้นทาง P ที่ประกอบไปด้วยเส้นเชื่อม e

- เกณฑ์ทุกสาขาที่ไม่ซ้ำ (All-Unique-Branch Criterion) คือชุดของเส้นทางไออาร์ซีเอฟจี P ที่ตอบสนองเกณฑ์ความครอบคลุมทุกสาขาอาร์ซีเอฟจี โดยที่ทุกเส้นเชื่อม e ในแต่ละอาร์ซีเอฟจี มีอย่างน้อยหนึ่งเส้นทาง p ใน P โดยที่ p ประกอบด้วยเส้นเชื่อม e

รายละเอียดของกราฟควบคุมกระแสไออาร์ซีเอฟจีและเกณฑ์ความครอบคลุมที่ได้นำเสนอไว้ในงานวิจัยของ Rountev และคณะ [7] ที่ได้กล่าวมาข้างต้นจะถูกแสดงไว้ในหัวข้อที่ 2.6 โดยละเอียด

2.2.3 เกณฑ์ของแผนภาพคอมมิวนิเคชั่น (Communication Diagram Criteria)

งานวิจัยของ Andrews และคณะ [12] นำเสนอเกณฑ์ความครอบคลุมที่ใช้ในการทดสอบจากแผนภาพคอมมิวนิเคชั่น ดังต่อไปนี้

1. **เกณฑ์ความครอบคลุมเงื่อนไข** (Condition Coverage-Cond Criterion) คือชุดข้อมูลทดสอบจะต้องทำให้แต่ละเงื่อนไขในส่วนของการตัดสินของแผนภาพคอมมิวนิเคชั่นได้ผลลัพท์ออกมาทั้งกรณีที่เป็นจริงและเป็นเท็จ

2. **เกณฑ์ความครอบคลุมฟูลล์เพรดิเคต** (Full Predicate Coverage-FP Criterion) คือชุดข้อมูลทดสอบจะต้องทำให้แต่ละประโยค (Clause) ในทุกเงื่อนไขในแผนภาพคอมมิวนิเคชั่นได้ผลลัพท์ออกมาทั้งกรณีที่เป็นจริงและเป็นเท็จ

3. **เกณฑ์ความครอบคลุมแต่ละเมจเสจบนเส้นเชื่อมโยง** (Each Message on Link -EML Criterion) คือชุดข้อมูลทดสอบที่ทำให้แต่ละเมจเสจที่เชื่อมระหว่างสองอ็อบเจกต์ในแผนภาพคอมมิวนิเคชั่นได้ถูกดำเนินการอย่างน้อยหนึ่งครั้ง

4. **เกณฑ์ทุกเส้นทางเมจเสจ** (All Message Path-AMP Criterion) คือชุดข้อมูลทดสอบที่ทำให้เส้นทางเมจเสจที่สามารถเป็นไปได้ (ลำดับของเมจเสจ) ในแผนภาพคอมมิวนิเคชั่นได้ถูกดำเนินการอย่างน้อยหนึ่งครั้ง

2.2.4 เกณฑ์ของแผนภาพสเตทแมชชีน (State Machine Diagram Criteria)

McQuillan และ Power [14] ได้กำหนดคำนิยามของแต่ละเกณฑ์ความครอบคลุมแผนภาพสเตทแมชชีนซึ่งได้ถูกนำเสนอไว้ในงานวิจัยของ Offutt และ Abdurazik [13] ดังต่อไปนี้

1. **ครอบคลุมทุกทรานซิชัน** (All Transitions Coverage) คือชุดข้อมูลทดสอบที่ทำให้ทุกทรานซิชันที่มีอยู่ในแผนภาพสเตทแมชชีนได้ถูกทดสอบ

2. **ครอบคลุมฟูลล์เพรดิเคต** (Full predicate Coverage) คือชุดข้อมูลทดสอบที่ทำให้แต่ละประโยค (Clause) ในทุกเงื่อนไขในแผนภาพสเตทแมชชีนได้ผลลัพท์ออกมาทั้งกรณีที่เป็นจริงและเป็นเท็จ

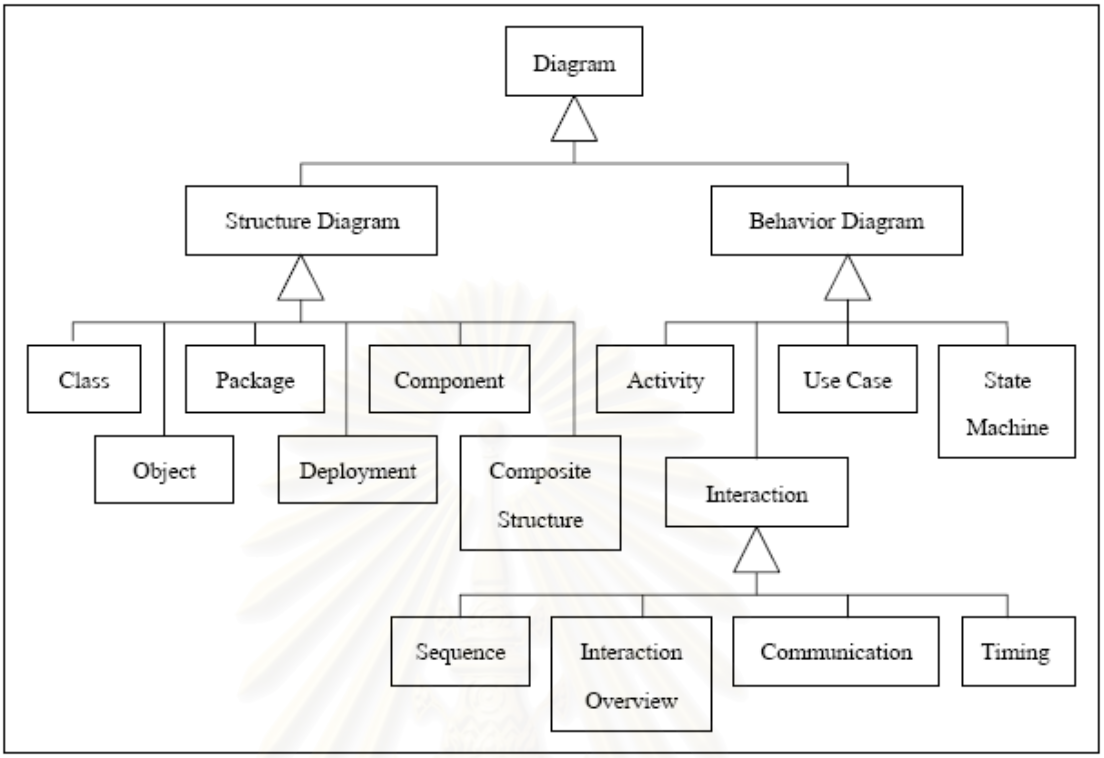
3. **ครอบคลุมคู่ของทรานซิชัน** (Transition-pair Coverage) คือชุดข้อมูลทดสอบที่ทำให้แต่ละคู่ของทรานซิชันที่เชื่อมต่อกันได้ถูกท่องผ่าน

2.2.5 เกณฑ์ของแผนภาพแอคทิวิตี (Activity Diagram Criteria)

แผนภาพแอคทิวิตีเน้นรูปแบบกระแสควบคุมจากกิจกรรมหนึ่งไปอีกกิจกรรมหนึ่งและเป็นกรณีพิเศษ (Special Case) ของแผนภาพสเตทแมชชีน ดังนั้นจึงสามารถนำเกณฑ์ความครอบคลุมของแผนภาพสเตทแมชชีนมาใช้เป็นเกณฑ์ความครอบคลุมของแผนภาพแอคทิวิตีได้ [14]

2.3 แผนภาพซีเควนซ์ (Sequence Diagram)

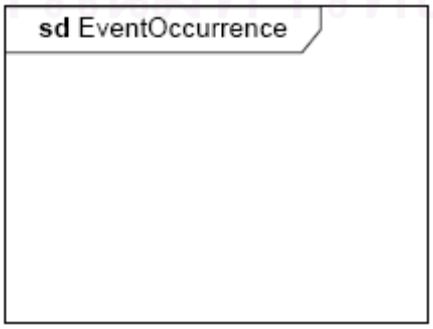
แผนภาพยูเอ็มแอลเป็นภาษาสัญลักษณ์รูปภาพมาตรฐานที่ใช้สำหรับการออกแบบจำลองเชิงวัตถุ (Object-Oriented Modeling) โดยมีพื้นฐานอยู่บนหลักการของวิชวลโมเดล-ลิง (Visual Modeling) คือการสร้างแบบจำลองโดยการใช้สัญลักษณ์รูปภาพเพื่อทำความเข้าใจกับความต้องการของลูกค้า ทำให้ระบบที่ออกแบบมีความชัดเจน และสามารถบำรุงรักษาระบบได้ง่ายขึ้น เนื่องด้วยยูเอ็มแอลเป็นภาษารูปภาพที่มีมาตรฐานและเป็นภาษาสากลที่ใช้ในการออกแบบซอฟต์แวร์ที่พัฒนาเชิงวัตถุ ดังนั้นเอกสารแสดงการออกแบบซอฟต์แวร์ที่ใช้แผนภาพยูเอ็มแอล จึงสามารถใช้เพื่อสื่อสารระหว่างผู้ร่วมงานให้เข้าใจระบบได้ตรงกันภายในระยะเวลาอันรวดเร็ว นอกจากนี้การนำเสนอด้วยแผนภาพยูเอ็มแอล ยังสนับสนุนหลักการเชิงวัตถุได้อย่างครบถ้วน [2] ในปัจจุบันแผนภาพยูเอ็มแอลที่เผยแพร่ออกสู่สาธารณะคือรุ่นที่ 2.0 ถูกพัฒนาโดยองค์กรโอเอ็มจี (OMG) ในปี ค.ศ. 2003 ซึ่งประกอบด้วยแผนภาพทั้งหมด 13 แผนภาพ [11] ดังแสดงในรูปที่ 2-9



รูปที่ 2-9 แสดงแผนภาพทั้งหมดของยูเอ็มแอลรุ่น 2.0 [2]

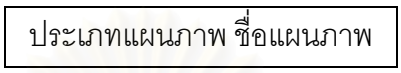
แผนภาพซีเควนซ์เป็นหนึ่งในประเภทแผนภาพเชิงพฤติกรรมของแผนภาพยูเอ็มแอล ซึ่งใช้เพื่อนำเสนอรายละเอียดในยูสเคส (Use Case) วัตถุประสงค์หลักของแผนภาพซีเควนซ์เน้นที่การลำดับเวลาของเมจเสจที่ส่งระหว่างอ็อบเจกต์ [23] สัญลักษณ์ของแผนภาพซีเควนซ์ในยูเอ็มแอลรุ่นที่ 2.0 ประกอบไปด้วย

1. **เฟรม (Frame)** เป็นสัญลักษณ์ที่ใช้กำหนดขอบเขตของแผนภาพ โดยใช้เครื่องหมายสี่เหลี่ยมและมีเลเบิลของแผนภาพ (Diagram's Label) ปรากฏในเครื่องหมายสี่เหลี่ยมพับมุม (Dog-eared Rectangle) โดยวางไว้ที่ตำแหน่งบนซ้ายสุดของเฟรม [24] ดังรูปที่ 2-10



รูปที่ 2-10 สัญลักษณ์แสดงเฟรมของแผนภาพ [25]

ในการเขียนเลเบิลของแผนภาพจะต้องเขียนให้อยู่ในรูปแบบดังแสดงในรูปที่ 2-11 ยกตัวอย่างเช่น sd Balance ซึ่ง sd คือประเภทแผนภาพซีเควนซ์และ Balance คือชื่อของแผนภาพซีเควนซ์ นอกเหนือจากใช้เพื่อกำหนดขอบเขตของแผนภาพแล้ว เฟรมยังถูกใช้เพื่อกำหนดขอบเขตของคอมบายด์แฟรกเมนต์ (Combined Fragment) ในแผนภาพซีเควนซ์อีกด้วย



รูปที่ 2-11 แสดงรูปแบบการเขียนเลเบิลของแผนภาพ [25]

2. **แอคเตอร์** (Actor) คือบุคคลหรือระบบภายนอกซึ่งโต้ตอบกับระบบโดยการส่งเมจเสจ ซึ่งแทนด้วยสัญลักษณ์รูปคน [26] ดังรูปที่ 2-12

รูปที่ 2-12 สัญลักษณ์รูปคนซึ่งเป็นตัวแทนของแอคเตอร์

3. **อ็อบเจกต์** (Object) ถูกแสดงอยู่ภายในเครื่องหมายรูปสี่เหลี่ยม ซึ่งแสดงถึงอ็อบเจกต์ที่ถูกสร้างขึ้นจากคลาสที่ประกาศอยู่หลังเครื่องหมายมหัพภาพ (:) ดังรูปที่ 2-13 สัญลักษณ์นี้จะถูกวางไว้ในส่วนบนสุดของแผนภาพ

รูปที่ 2-13 รูปแบบและเครื่องหมายในการแสดงอ็อบเจกต์

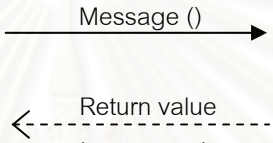
4. **ไลฟ์ไลน์** (Lifeline) เป็นเครื่องหมายที่แสดงถึงช่วงระยะเวลาการมีชีวิตอยู่ของอ็อบเจกต์ในการส่งหรือรับเมจเสจ โดยถูกแสดงด้วยเส้นประแนวตั้ง ดังรูปที่ 2-14

รูปที่ 2-14 แสดงไลฟ์ไลน์ของอ็อบเจกต์

5. **แอคติเวชั่น** (Activation) เป็นเครื่องหมายรูปสี่เหลี่ยมผืนผ้ายาวแนวตั้ง ดังรูปที่ 2-15 ถูกแสดงเมื่ออ็อบเจกต์ส่งหรือรับเมจเสจ วางอยู่เหนือไลฟ์ไลน์ [11]

รูปที่ 2-15 แสดงแอคติเวชั่น

6. **เมจเสจ** (Message) เป็นตัวที่ส่งข้อมูลจากอ็อบเจกต์หนึ่งไปยังอีกอ็อบเจกต์หนึ่ง [11] โดยแบ่งออกเป็นสองลักษณะคือ การเรียกใช้และการคืนค่า (Return value) ซึ่งถูกแสดงด้วยเครื่องหมายลูกศรเส้นทึบและลูกศรเส้นประตามลำดับ ดังรูปที่ 2-16



รูปที่ 2-16 แสดงเมจเสจที่ใช้ในการสื่อสารระหว่างอ็อบเจกต์

7. **อ็อบเจกต์เดสสตรักชั่น** (Object Destruction) ถูกแสดงด้วยเครื่องหมายกากบาท ดังรูปที่ 2-17 โดยวางไว้ที่ปลายไลฟ์ไลน์ของอ็อบเจกต์เพื่อแสดงว่าอ็อบเจกต์นั้นสิ้นสุดลงและไม่มีตัวตนแล้ว [8]

รูปที่ 2-17 แสดงอ็อบเจกต์เดสสตรัคชั่น

8. **คอมบายด์แฟรกเมนต์** (Combined Fragment) เป็นส่วนที่ใช้ในการรวมกลุ่มของเมจเสจเข้าไว้ด้วยกันเพื่อแสดงถึงกระแสอย่างมีเงื่อนไข (Conditional Flow) เพื่อให้แบบจำลองสามารถกำหนดรูปแบบของตรรกศาสตร์ที่ต้องการให้ระบบดำเนินการได้ [24] สัญลักษณ์ที่ใช้แสดงดังรูปที่ 2-18

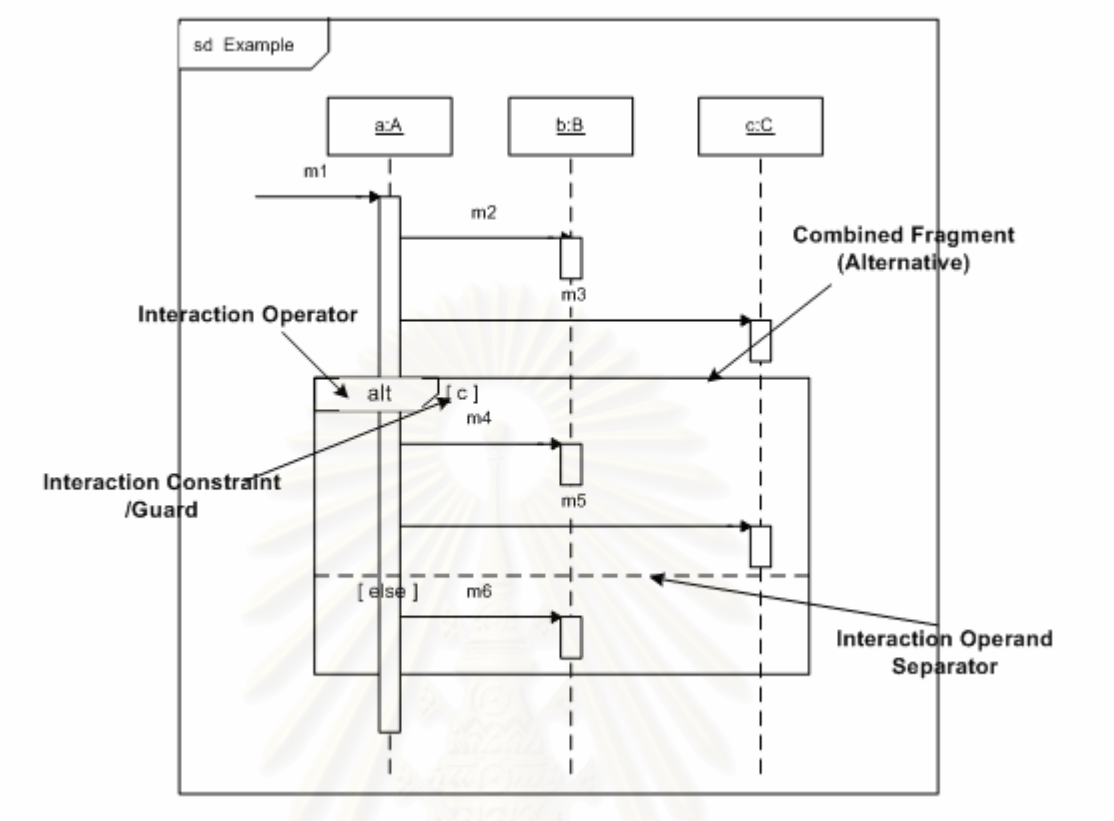
รูปที่ 2-18 แสดงสัญลักษณ์ของคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟ [25]

ในส่วนของคอมบายด์แฟรกเมนต์ได้ถูกกำหนดเพิ่มเติมไว้ในยูเอ็มแอลรุ่นที่ 2.0 ซึ่งมีอยู่ด้วยกัน 11 ประเภทได้แก่

- วีคซีเควนซิง (Weak Sequencing)
- อัลเทอร์เนทีฟ (Alternative)
- ออปชัน (Option)
- เบรค (Break)
- พาราแรล (Parallel)
- สตริกท์ซีเควนซิง (Strict Sequencing)
- ลูป (Loop)
- คริทิคอลรีเจียนท์ (Critical Region)
- เนกาทีฟ (Negative)
- แอสเสิร์ทชัน (Assertion)
- อิกนอร์/คอนสิเดอร์ (Ignore/Consider)

ในที่นี้จะอธิบายรายละเอียดของคอมบายด์แฟรกเมนต์เพียง 3 ประเภทเท่านั้นได้แก่ อัล-เทอร์เนทีฟ ออปชัน ลูป เนื่องจากเป็นคอมบายด์แฟรกเมนต์ที่นักวิเคราะห์และนักออกแบบส่วนใหญ่นิยมใช้ในการออกแบบโปรแกรมเชิงวัตถุ [24] โดยมีรายละเอียดดังต่อไปนี้

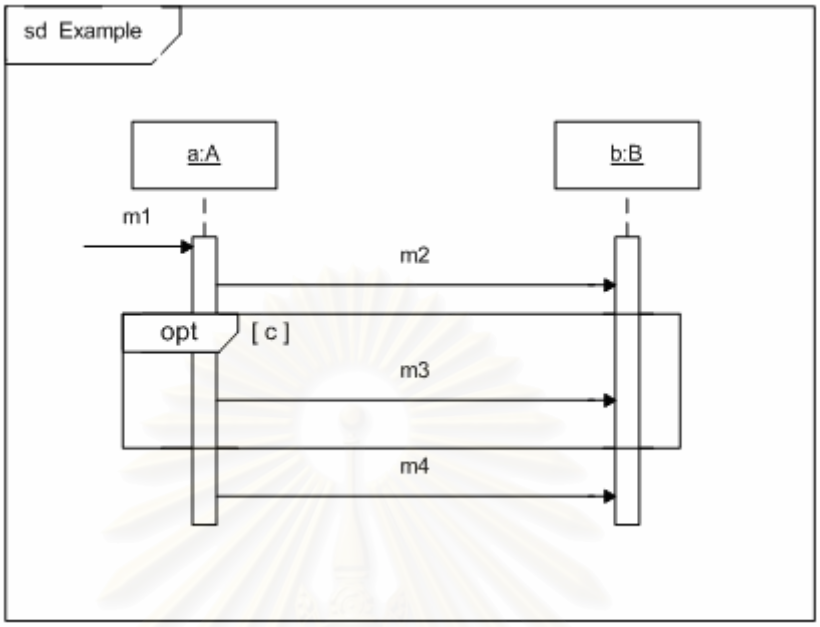
1) อัลเทอร์เนทีฟ (Alternative) เป็นคอมบายด์แฟรกเมนต์ที่ใช้เพื่อแสดงถึงทางเลือกของพฤติกรรม [25] ในรูปแบบของตรรกศาสตร์ “if then else” คือกำหนดทางเลือกของเหตุการณ์ที่ไม่เกิดร่วมกัน ในการใช้งานจะระบุอินเตอร์แรคชั่นโอเปอร์เรเตอร์ (Interaction Operator) “alt” ไว้ในชื่อของคอมบายด์แฟรกเมนต์ (ส่วนที่เป็นสี่เหลี่ยมพับมุม) ดังรูปที่ 2-19 ภายในคอมบายด์แฟรก-เมนต์ประเภทอัลเทอร์เนทีฟจะถูกแบ่งออกเป็นตัวถูกดำเนินการ (Operand) ด้วยเครื่องหมายเส้นประ และภายในตัวถูกดำเนินการจะมีอินเตอร์แรคชั่นคอนแสตรนท์หรือการ์ด (Interaction Constraint/Guard) ซึ่งเป็นสิ่งที่บ่งบอกถึงเงื่อนไข โดยวางไว้ในตำแหน่งบนซ้ายสุดของแต่ละตัวถูกดำเนินการ



รูปที่ 2-19 แสดงรูปแบบการเขียนคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟ [24,25]

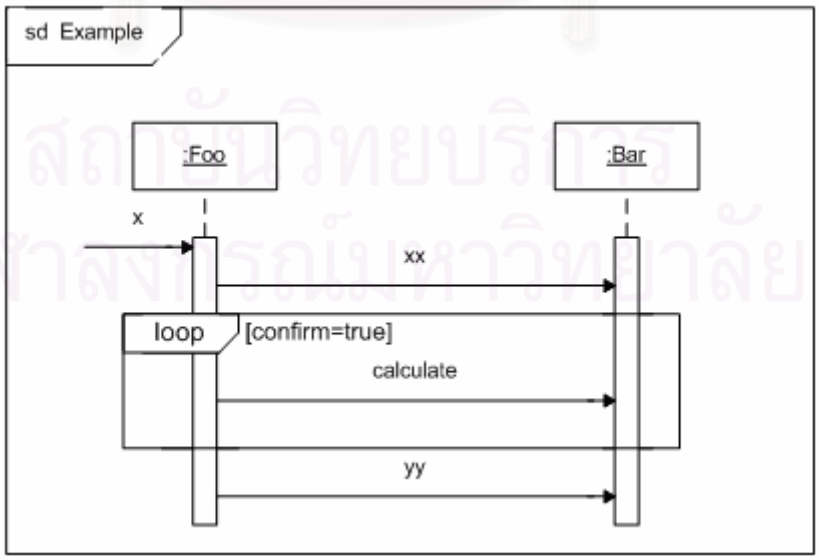
2) ออบชัน (Option) เป็นคอมบายด์แฟรกเมนต์ที่ใช้เพื่อแสดงถึงทางเลือกของพฤติกรรมที่อาจเกิดขึ้นหรือไม่เกิดขึ้นก็ได้ [25] ซึ่งคล้ายกันกับคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟต่างกันตรงที่คอมบายด์แฟรกเมนต์ประเภทออบชันจะมีเพียงหนึ่งตัวถูกดำเนินการเท่านั้น [24] อินเตอร์แรคชั่นโอเปอร์เรเตอร์ที่แสดงถึงคอมบายแฟรกเมนต์ประเภทนี้คือ “opt”

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2-20 แสดงรูปแบบการเขียนคอมบายด์แฟรกเมนต์ประเภทออปชัน [24]

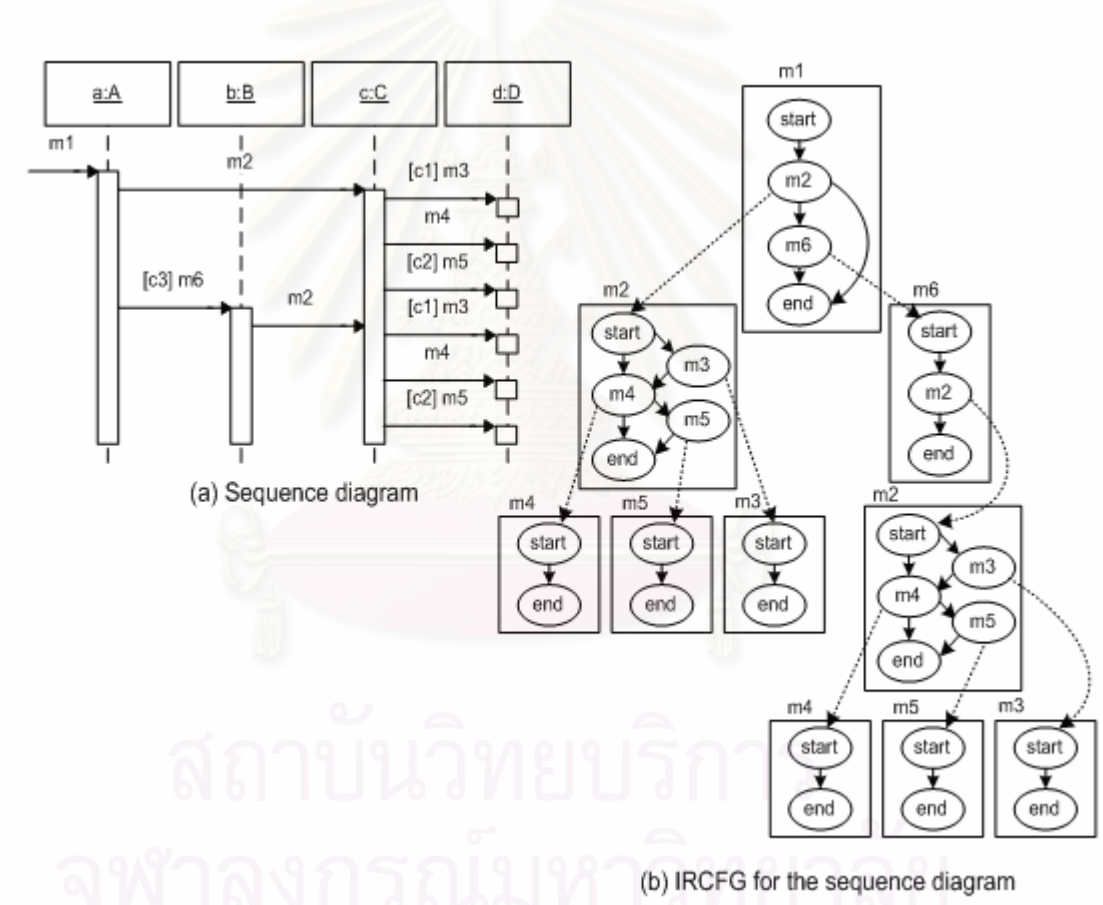
3) ลูป (Loop) เป็นคอมบายด์แฟรกเมนต์ที่ใช้เพื่อแสดงถึงการวนทำซ้ำของเมจเสจภายในตัวถูกดำเนินการ ใช้อินเตอร์แรคชั่นโอเปอร์เรเตอร์ “loop” เพื่อแสดงถึงคอมบายด์แฟรกเมนต์ประเภทนี้ การด์ภายในลูปนอกจากเป็นเงื่อนไขบูลีนตามปกติแล้วยังสามารถประกอบไปด้วยจำนวนเงื่อนไขการทำซ้ำที่น้อยที่สุดและจำนวนเงื่อนไขการทำซ้ำที่มากที่สุด (minint และ maxint ตามลำดับ)



รูปที่ 2-21 แสดงรูปแบบการเขียนคอมบายด์แฟรกเมนต์ประเภทลูป

2.4 อินเตอร์โพรซิเดอรัลเรสตริกด์คอนโทรลโฟลกราฟหรือไออาร์ซีเอฟจี (Interprocedural Restricted Control-Flow Graph -IRCFG) และเกณฑ์ความครอบคลุม

งานวิจัยของ Rountev และคณะ [7] นำเสนอกราฟควบคุมกระแสที่แสดงลำดับของเมจเสจในแผนภาพซีเควนซ์ซึ่งอยู่ในรูปของโครงสร้างของข้อมูล กราฟควบคุมกระแสนี้มีชื่อว่า อินเตอร์โพรซิเดอรัลเรสตริกด์คอนโทรลโฟลกราฟหรือไออาร์ซีเอฟจี ถูกสร้างขึ้นเพื่อ (1) ใช้ในการให้คำนิยามอย่างเป็นทางการของเกณฑ์ความครอบคลุมโดยสอดคล้องกับแผนภาพซีเควนซ์ และ (2) ใช้เป็นพื้นฐานสำหรับการวิเคราะห์ขณะใช้งาน (Run-time Analysis) ในการวัดความครอบคลุมระหว่างการทดสอบ



รูปที่ 2-22 ตัวอย่างแผนภาพซีเควนซ์และไออาร์ซีเอฟจีของแผนภาพซีเควนซ์ [7]

ในงานวิจัยของ Rountev และคณะ [7] ไม่ได้บอกถึงหลักการอย่างเป็นทางการในการแปลงแผนภาพซีเควนซ์ไปเป็นไออาร์ซีเอฟจีไว้อย่างชัดเจน แต่ใช้วิธีการยกตัวอย่างดังรูปที่ 2-22 เพื่อให้เข้าใจถึงแนวทางการแปลงแผนภาพซีเควนซ์ให้อยู่ในรูปของไออาร์ซีเอฟจีเท่านั้น โดยรายละเอียดของไออาร์ซีเอฟจีที่งานวิจัยดังกล่าวระบุไว้มีดังต่อไปนี้ [7]

ไออาร์ซีเอฟจีประกอบไปด้วยกลุ่มของเรสตริกด์ซีเอฟจี (Restricted CFGs - RCFGs) และเชื่อมต่อระหว่างกันด้วยเส้นเชื่อม (Edges) แต่ละอาร์ซีเอฟจี (RCFG) เป็นของเมท็อด (Method) เพียงเมท็อดเดียวและเสมือนเป็นกราฟควบคุมกระแสสำหรับเมท็อดนั้น ๆ ส่วนประกอบของไออาร์ซีเอฟจีมีดังต่อไปนี้

1. **เรสตริกด์คอนโทรลโฟลกราฟหรืออาร์ซีเอฟจี** (Restricted Control Flow Graph-RCFG) เป็นส่วนที่แสดงถึงการเรียกใช้เมท็อดภายในบอดี้ (Body) ของอีกเมท็อดหนึ่ง อาร์ซีเอฟจีของแต่ละเมท็อด จะถูกแสดงอยู่ภายในสัญลักษณ์รูปสี่เหลี่ยมผืนผ้าโดยมีชื่อของเมท็อด (m) ระบุไว้ ดังรูปที่ 2-23

รูปที่ 2-23 แสดงขอบเขตของอาร์ซีเอฟจี

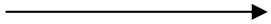
จากรูปที่ 2-22(b) m1 ที่อยู่ภายนอกกล่องสี่เหลี่ยมผืนผ้านั้นเกิดขึ้นจากการส่งเมจเสจชื่อ m1 ไปยังอ็อบเจกต์ a ที่แสดงในรูปที่ 2-22(a) เพื่อเรียกใช้งานเมท็อด m1

2. **โหนด (Node)** แต่ละโหนด ในอาร์ซีเอฟจีของเมท็อด m เป็นตัวแทนของเมท็อดที่ถูกเรียกถึงภายในบอดี้ (Body) ของเมท็อด m โดยถูกแสดงด้วยสัญลักษณ์รูปวงรี ดังรูปที่ 2-24

รูปที่ 2-24 แสดงสัญลักษณ์ที่เป็นตัวแทนของเมท็อดที่ถูกเรียกใช้

ในแต่ละอาร์ซีเอฟจีจะต้องประกอบไปด้วยโหนดอย่างน้อย 2 โหนด ได้แก่ โหนดเริ่มต้น (Start Node) และโหนดสิ้นสุด (End Node) ซึ่งเป็นตัวแทนของเวลาที่เริ่มดำเนินการในเมท็อดและเวลาที่มีการคืนค่ากลับไปยังส่วนที่เรียกใช้งานเมท็อด ตามลำดับ

3. **เส้นเชื่อมอินทราโพรซิเดอรัลหรือเส้นเชื่อมอาร์ซีเอฟจี** (Intraprocedural Edge-RCFG Edge) เป็นการแสดงถึงลำดับความสัมพันธ์ระหว่างโหนดภายในอาร์ซีเอฟจี ซึ่งความสัมพันธ์นี้ถูกแสดงด้วยเครื่องหมายลูกศรเส้นทึบ ดังรูปที่ 2-25



รูปที่ 2-25 แสดงความสัมพันธ์ระหว่างโหนดภายในอาร์ซีเอฟจี

จากรูปที่ 2-22(a) เมื่อเริ่มดำเนินการในเมท็อด m1 ของอ็อบเจกต์ a และมีการเรียกในเมท็อด m2 ของอ็อบเจกต์ b ภายในการดำเนินการของเมท็อด m1 ความสัมพันธ์นี้จะถูกเขียนในรูปของเส้นเชื่อม (start, m2) ในอาร์ซีเอฟจีของเมท็อด m1 และหลังจากการเรียกใช้งานเมท็อด m2 ได้สิ้นสุดลง เมท็อด m6 อาจจะถูกเรียกดำเนินการโดยเมท็อด m1 หรือการดำเนินการภายในเมท็อด m1 อาจจะสิ้นสุดลงโดยไม่มีการเรียกใช้เมท็อด m6 ซึ่งขึ้นอยู่กับเงื่อนไข c3 ความเป็นไปได้ของทั้ง 2 กรณีนี้ถูกนำเสนอในรูปแบบเส้นเชื่อมอาร์ซีเอฟจี (m2, m6) และ (m2, end) ตามลำดับ

4. **เส้นเชื่อมอินเตอร์โพรซิเดอรัล** (Interprocedural Edge) เป็นการแสดงถึงความสัมพันธ์ระหว่างอาร์ซีเอฟจี ซึ่งเชื่อมโหนด n ในอาร์ซีเอฟจีของเมท็อดที่มีการเรียกใช้เมท็อด n กับโหนดเริ่มต้นในอาร์ซีเอฟจีของเมท็อด n โดยความสัมพันธ์นี้ถูกแสดงด้วยเครื่องหมายลูกศรเส้นประ ดังรูปที่ 2-26

รูปที่ 2-26 แสดงความสัมพันธ์ระหว่างอาร์ซีเอฟจี

เกณฑ์ความครอบคลุมของแผนภาพซีเควนซ์ที่ได้นำเสนอโดย Rountev และคณะ [7] โดยอ้างอิงจากการแปลงแผนภาพซีเควนซ์ให้อยู่ในรูปของกราฟควบคุมกระแสไออาร์ซีเอฟจี มีด้วยกันทั้งหมด 4 เกณฑ์ความครอบคลุม ได้แก่

1. **เกณฑ์ทุกเส้นทางไออาร์ซีเอฟจี** (All-IRCFG-Path Criterion) เป็นเกณฑ์ที่ต้องการให้การทดสอบดำเนินการเป็นไปตามชุดของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ (Complete IRFCF Path) ทุกเส้นทาง ซึ่งเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ คือเส้นทางที่ท่องไปยังแต่ละโหนดเริ่มต้นถึงโหนดสิ้นสุด (Start-to-End) ของไออาร์ซีเอฟจี ตัวอย่างของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์หนึ่งเส้นทาง ซึ่งอ้างอิงจากรูปที่ 2-22(b) ได้แก่

($start_{m1}$,m2, $start_{m2}$, m4, $start_{m4}$, end_{m4}, m5, $start_{m5}$, end_{m5}, end_{m2}, end_{m1})

รูปที่ 2-27 แสดงตัวอย่างของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์

ให้ p เป็นลำดับของโหนดไออาร์ซีเอฟจีที่ต่อเนื่องกันโดยโหนดลำดับแรกคือโหนดเริ่มต้น และโหนดลำดับสุดท้ายคือโหนดสิ้นสุดภายในอาร์ซีเอฟจีที่เป็นราก (Root RCFG) พิจารณาโหนด n_i ใน p และให้ R แทนอาร์ซีเอฟจีที่มีโหนด n_i อยู่ภายใน โดยที่ลำดับถัดจากโหนด n_i ใน p คือโหนด n_j ซึ่ง p ที่ถือว่าเป็นเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์นั้นจะต้องมีคุณสมบัติดังต่อไปนี้

a) หาก n_i เป็นโหนดเริ่มต้นของ R จะต้องมีเส้นเชื่อมอินทราโพรซิเดอรัลหรือเส้นเชื่อมอาร์ซีเอฟจี (n_i,n_j)

b) หาก n_i ไม่ใช่โหนดเริ่มต้นหรือโหนดสิ้นสุดของ R แล้ว

i. จะต้องมีเส้นเชื่อมอินเตอร์โพรซิเดอรัล (n_i ,n_j) โดยที่ n_j คือโหนดเริ่มต้นของอาร์ซีเอฟจีที่เป็นลูกของ R ในต้นไม้อาร์ซีเอฟจี (RCFG Tree) หรือ

ii. ไม่มีเส้นเชื่อมอินเตอร์โพรซิเดอรัลที่เริ่มต้นที่ n_i และ (n_i ,n_j) คือ เส้นเชื่อมอินทราโพรซิเดอรัลใน R

c) หาก n_i คือโหนดสิ้นสุดของ R แล้วอาร์ซีเอฟจีที่เป็นแม่ของ R (Parent of R) ในต้นไม้อาร์ซีเอฟจี จะประกอบด้วยเส้นเชื่อมอินทราโพรซิเดอรัล (n_k,n_i) และมีเส้นเชื่อมอินเตอร์โพร-ซิเดอรัลจาก n_k ไปยังโหนดเริ่มต้นของ R

จากตัวอย่างในรูปที่ 2-22 (b) การร้องขอ (Invocation) m2 จาก m1 ทำให้เกิด 4 เส้นทางย่อย (Subpath) ที่แตกต่างกัน เช่นเดียวกันกับการร้องขอ m6 จาก m1 ก็ทำให้เกิด 4 เส้นทางย่อยเช่นกัน ดังนั้นในการที่ m1 เรียกใช้ m2 และ m6 จะมีจำนวนเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ 16 เส้นทาง และเมื่อพิจารณากรณีที่ m6 ถูกข้าม จำนวนทั้งหมดของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์จะกลายเป็น 20 เส้นทาง ในตัวอย่างนี้การทดสอบที่ครอบคลุมทั้ง 20 เส้นทางนี้เป็นไปตามเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจี

2. **เกณฑ์ทุกเส้นทางอาร์ซีเอฟจี** (All-RCFG-Path Criterion) เป็นเกณฑ์ที่ต้องการให้การทดสอบดำเนินการเป็นไปตามชุดของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ซึ่งครอบคลุมทุกเส้นทางอาร์ซีเอฟจี ซึ่งเส้นทางอาร์ซีเอฟจีเป็นลำดับของโหนดอาร์ซีเอฟจีภายในอาร์ซีเอฟจี R ซึ่งเริ่มต้นด้วยโหนดเริ่มต้นของ R และจบลงด้วยโหนดสิ้นสุดของ R แต่ละคู่ของโหนดที่อยู่ติดกันในเส้นทางจะต้องเป็นไปตามเส้นเชื่อมอินทราโพรซิเดอรัลใน R เช่น อาร์ซีเอฟจีที่เป็นรากในรูปที่

2-22(b) มี 2 เส้นทางอาร์ซีเอฟจี ได้แก่ (start, m2, end) และ (start, m2, m6, end)

เส้นทางไออาร์ซีเอฟจีที่สมบูรณ์สามารถครอบคลุมได้หลายเส้นทางอาร์ซีเอฟจี ดังตัวอย่างของเส้นทาง ไออาร์ซีเอฟจี ที่สมบูรณ์ซึ่งแสดงไว้ในรูปที่ 2-27 เส้นทางนี้ได้ครอบคลุมเส้นทาง อาร์ซีเอฟจี 4 เส้นทางดังต่อไปนี้

- ($start_{m1}$, m2, end_{m1})
- ($start_{m2}$,m4,m5,end_{m2})
- ($start_{m4}$,end_{m4})
- ($start_{m5}$,end_{m5})

เกณฑ์ความครอบคลุมทุกเส้นทางอาร์ซีเอฟจี ต้องการให้การทดสอบดำเนินการให้เป็นไปตามชุดของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ ซึ่งครอบคลุมทุกเส้นทางอาร์ซีเอฟจี ในรูปที่ 2-22(b) เส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ที่เป็นไปตามเกณฑ์ความครอบคลุมนี้มีทั้งหมดด้วยกัน 5 เส้นทาง

3. **เกณฑ์ทุกสาขาอาร์ซีเอฟจี** (All-RCFG-Branch Criterion) เป็นเกณฑ์ที่ต้องการให้การทดสอบดำเนินการเป็นไปตามชุดของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ซึ่งครอบคลุมทุกพฤติกรรมที่มีเงื่อนไข (All Conditional Behavior) หรือต้องการให้ครอบคลุมทุกเส้นเชื่อมอาร์ซี-เอฟจี จากรูปที่ 2-22(b) ชุดของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ซึ่งรองรับเกณฑ์นี้มี 3 เส้นทางด้วยกันซึ่งได้ถูกแสดงไว้ในตารางที่ 2-3

ตารางที่ 2-3 แสดงชุดของเส้นทางที่เป็นไปตามเกณฑ์ทุกสาขาอาร์ซีเอฟจี

เส้นทางที่	**เส้นทางไออาร์ซีเอฟจีที่สมบูรณ์**
1	($start_{m1}$, m2, $start_{m2}$, m3, $start_{m3}$, end_{m3}, m4, $start_{m4}$, end_{m4}, m5, $start_{m5}$, end_{m5},end_{m2}, end_{m1})
2	($start_{m1}$, m2, $start_{m2}$, m4, $start_{m4}$, end_{m4}, end_{m2}, m6, $start_{m6}$, m2, $start_{m2}$, m3, $start_{m3}$, end_{m3}, m4, $start_{m4}$, end_{m4}, end_{m2}, end_{m6}, end_{m1})
3	($start_{m1}$, m2, $start_{m2}$, m4, $start_{m4}$, end_{m4}, end_{m2}, m6, $start_{m6}$, m2, $start_{m2}$, m4, $start_{m4}$, end_{m4}, m5, $start_{m5}$, end_{m5}, end_{m2}, end_{m6}, end_{m1})

4. **เกณฑ์ทุกสาขาที่ไม่ซ้ำ** (All-Unique-Branch Criterion) เป็นเกณฑ์ที่ต้องการให้การทดสอบดำเนินการให้เป็นไปตามชุดของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ซึ่งครอบคลุมเส้นเชื่อมอาร์ซีเอฟจีที่ไม่ซ้ำ คือจะไม่สนใจว่าเส้นเชื่อมอาร์ซีเอฟจีจะเกิดขึ้นกี่ครั้งในไออาร์ซีเอฟจี เกณฑ์นี้สนใจเพียงให้การทดสอบนั้นครอบคลุมเส้นเชื่อมอาร์ซีเอฟจีอย่างน้อยเพียงหนึ่งครั้งด้วยเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ จากรูปที่ 2-22(b) ชุดของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ซึ่งรองรับเกณฑ์ความครอบคลุมนี้ประกอบไปด้วย 2 เส้นทางดังแสดงในตารางที่ 2-4

ตารางที่ 2-4 แสดงชุดของเส้นทางที่เป็นไปตามเกณฑ์ทุกสาขาที่ไม่ซ้ำ

เส้นทางที่	เส้นทางไออาร์ซีเอฟจีที่สมบูรณ์
1	($start_{m1}$, m2, $start_{m2}$, m3, $start_{m3}$, end_{m3}, m4,$start_{m4}$, end_{m4}, m5, $start_{m5}$, end_{m5}, end_{m2}, end_{m1})
2	($start_{m1}$, m2, $start_{m2,}$ m4, $start_{m4}$, end_{m4}, end_{m2}, m6, $start_{m6}$, m2, $start_{m2}$, m4, $start_{m4}$, end_{m4}, end_{m2}, end_{m6}, end_{m1})

2.5 ภาษาข้อจำกัดเชิงวัตถุหรือโอซีแอล (Object Constraint Language-OCL)

เนื่องจากแผนภาพยูเอ็มแอลถูกนำเสนอด้วยแบบจำลองกราฟฟิก ซึ่งไม่เพียงพอต่อการนำเสนอข้อกำหนดหรือข้อจำกัดที่พบในเอกสารข้อกำหนดซอฟต์แวร์ ผู้ออกแบบจึงต้องการกลไกที่สามารถระบุข้อจำกัดของส่วนประกอบในแบบจำลองแผนภาพยูเอ็มแอล[27] และโดยทั่วไปแล้วข้อจำกัดมักถูกเขียนขึ้นด้วยภาษาธรรมชาติ (Natural Language) จึงทำให้เกิดความคลุมเครือไม่ชัดเจน ดังนั้นเพื่อให้ผู้ออกแบบสามารถกำหนดข้อจำกัดของส่วนประกอบในแบบจำลองแผนภาพยูเอ็มแอลได้อย่างชัดเจน องค์กรโอเอ็มจี (Object Management Group-OMG) จึงได้พัฒนาภาษาข้อจำกัดเชิงวัตถุหรือโอซีแอล (Object Constraint Language-OCL) ขึ้น

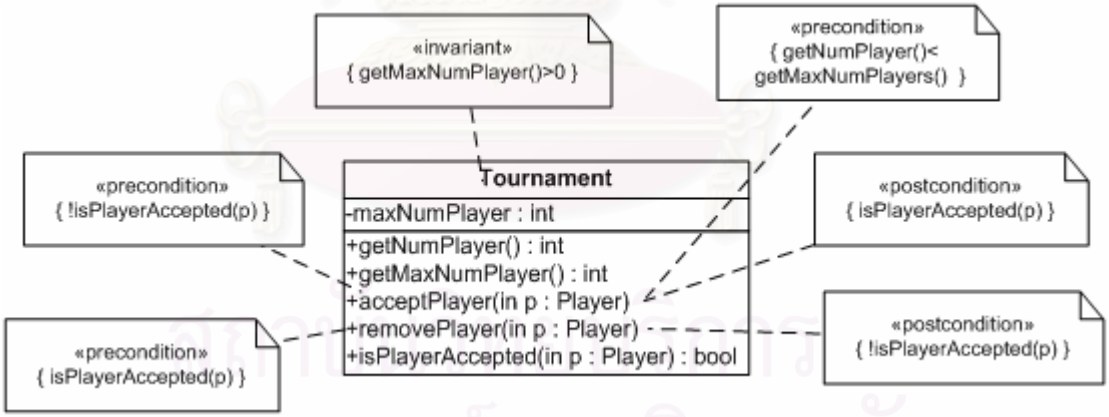
โอซีแอลเป็นภาษาทางการ (Formal Language) ที่ใช้แสดงข้อจำกัด (Constraint) และตรรกะ (Logic) ขององค์ประกอบภายในแบบจำลองยูเอ็มแอล [15] ข้อจำกัดถูกแสดงในรูปของนิพจน์บูลีน (Boolean Expression) ซึ่งมีการคืนค่ากลับเป็นจริงหรือเท็จ โดยสามารถแบ่งออกเป็น 3 ประเภท [9,15] ได้แก่

1. เงื่อนไขคงที่ (Invariant) เป็นข้อจำกัดที่บอกถึงเงื่อนไขที่จะต้องมีค่าเป็นจริงเสมอในทุกอินแสตนท์ (Instance) ของคลาส โดยเงื่อนไขคงที่นี้สามารถแบ่งเป็น 2 ลักษณะได้แก่ เงื่อนไขคงที่ของแอตทริบิวส์ (Invariant on Attribute) และ เงื่อนไขคงที่ของความสัมพันธ์ (Invariant on Association)

2. เงื่อนไขก่อนดำเนินการ (Precondition) เป็นข้อจำกัดของการดำเนินการ (Operation) ที่ต้องควบคุมให้เป็นจริงก่อนเข้าใช้การดำเนินการ โดยทั่วไปแล้วจะอยู่ในรูปของแอตทริบิวส์ของอ็อบเจกต์ที่อยู่ภายใต้การดำเนินการหรือแอตทริบิวส์ของพารามิเตอร์ (Parameter) ที่อยู่ในเมจเสจที่มีการเรียกใช้งานการดำเนินการ

3. เงื่อนไขหลังดำเนินการ (Postcondition) เป็นข้อจำกัดของการดำเนินการที่ต้องควบคุมให้เป็นจริงหลังจากการดำเนินการสิ้นสุดลง โดยทั่วไปแล้วจะอยู่ในรูปของ (1) แอตทริบิวส์ของอ็อบเจกต์ที่อยู่ภายใต้การดำเนินการ (2) แอตทริบิวส์ของพารามิเตอร์ที่อยู่ในเมจเสจที่มีการเรียกในงานการดำเนินการ (3) ในรูปของค่าคืนกลับ และหรือ (4) ในรูปของข้อยกเว้น (Exception) ที่อาจเกิดขึ้น

รูปแบบการใช้โอซีแอลสามารถแบ่งออกเป็น 2 ลักษณะคือ (1) เขียนเป็นหมายเหตุแนบไปกับองค์ประกอบของแผนภาพยูเอ็มแอล หรือ (2) เขียนด้วยรูปแบบข้อความ (Textual Form) ตัวอย่างของการเขียนโอซีแอลในรูปแบบของหมายเหตุ ถูกแสดงไว้ในรูปที่ 2-28 ข้อจำกัดของแต่ละองค์ประกอบจะปรากฏอยู่ภายในรูปสี่เหลี่ยมพับมุม (Folded Corner) และเชื่อมไปยังองค์ประกอบดังกล่าวด้วยเส้นประ



รูปที่ 2-28 แสดงตัวอย่างของเงื่อนไขคงที่ เงื่อนไขก่อนดำเนินการ และเงื่อนไขหลังดำเนินการของโอซีแอลที่แนบไว้เป็นหมายเหตุในแบบจำลองยูเอ็มแอล (แผนภาพคลาส) [15]

การแนบนิพจน์โอซีแอลลงไปในแผนภาพอาจทำให้แผนภาพดูยุ่งเหยิง ไม่เป็นระเบียบ ด้วยเหตุนี้การเขียนโอซีแอลด้วยรูปแบบข้อความ (Textual Form) จึงเป็นอีกทางเลือกหนึ่งในการแสดงนิพจน์โอซีแอลโดยมีรายละเอียดดังต่อไปนี้ [15]

- **context** เป็นคำสงวน (keyword) ที่ใช้เพื่อแสดงถึงเอนทิตี (Entity) ที่ถูกกำหนดโดยนิพจน์โอซีแอล
- **inv pre** และ**post** เป็นคำสวงนซึ่งใช้แทนสเตอริโอไทพ์ของยูเอ็มแอล (UML Stereotypes) <<invariant>> <<precondition>> และ <<postcondition>> ตามลำดับ จากนั้นจึงตามด้วยนิพจน์โอซีแอล

2.5.1 **ข้อจำกัดของคลาสซิไฟเออร์ (Constraint on Classifiers)** การนิยามบริบทของนิพจน์โอซีแอล (OCL Context) เป็นการระบุว่าส่วนใดของแบบจำลองยูเอ็มแอลที่ถูกกำหนดโดยนิพจน์นั้น ๆ โดยทั่วไปแล้วจะเป็น คลาส (Class) ตัวต่อประสาน (Interface) ประเภทของข้อมูล (Data type) หรือองค์ประกอบ (Component) ซึ่งมาตรฐานของยูเอ็มแอลจะเรียกรวมว่า คลาสซิไฟเออร์ [28] รูปแบบและตัวอย่างในการเขียนนิพจน์โอซีแอลในรูปแบบข้อความแสดงไว้ในรูปที่ 2-29 และ 2-30 ตามลำดับ

```
context <classifier name>
Inv:<ocl expression>
```

รูปที่ 2-29 แสดงรูปแบบในการเขียนข้อจำกัดประเภทเงื่อนไขคงที่ในรูปแบบข้อความ

```
context Tournament
Inv: self.getMaxNumPlayer() >0
```

รูปที่ 2-30 แสดงตัวอย่างการเขียนข้อจำกัดเงื่อนไขคงที่ [15]

จากตัวอย่างที่แสดงดังรูปที่ 2-30 เป็นการกำหนดว่าแอตทริบิวส์ maxNumPlayer ของทุกอินแสตนท์ของคลาส Tournament จะต้องมีค่าเป็นบวกเสมอ **self** เป็นคำสงวนซึ่งเป็นตัวแทนของทุกอินแสตนท์ของคลาส

2.5.2 **ข้อจำกัดของการดำเนินการ (Constraints on Operations)** ได้แก่ เงื่อนไขก่อนดำเนินการ (Preconditon) และเงื่อนไขหลังดำเนินการ (Postcondition) ซึ่งแทนด้วยคำสงวน **pre** และ **post** ตามลำดับ [26] รูปแบบและตัวอย่างในการเขียนข้อจำกัดของการดำเนินการแสดงไว้ดังรูปที่ 2-31 และ 2-32 ตามลำดับ

```
context <classifier name>::<method name(<argument name,
argument type>)>:: <return type>
pre:<ocl expression>
post:<ocl expression>
```

รูปที่ 2-31 แสดงรูปแบบการเขียนข้อจำกัดของการดำเนินการในรูปแบบข้อความ

```
context Tournament :: removePlayer (p : Player)
pre: isPlayerAccepted(p)
post: !isPlayerAccepted(p)
post: getNumPlayers() = @pre.getNumPlayers()-1
```

รูปที่ 2-32 แสดงตัวอย่างการเขียนข้อจำกัดของการดำเนินการ [15]

การเขียนเงื่อนไขหลังการดำเนินการ สามารถใช้คำสงวน **result** เพื่ออ้างถึงค่าที่จะได้กลับคืนมาจากการดำเนินการ ตัวอย่างนิพจน์ดังรูปที่ 2-33 เป็นการกำหนดว่าค่าธรรมเนียมของนักเรียนจะต้องถูกจ่ายก่อนลงทะเบียนเรียนคอร์สและการดำเนินการ registerStudent ต้องคืนค่าเป็นจริงเมื่อการดำเนินการนี้ได้สิ้นสุดลง

```
context Course :: registerStudent (s:Student) : Boolean
pre : s.tuitionPaid = true
post : result = true
```

รูปที่ 2-33 แสดงตัวอย่างการเขียนข้อจำกัดเงื่อนไขหลังดำเนินการโดยใช้คำสงวน result [26]

2.5.3 **ข้อจำกัดของแอตทริบิวส์ (Constraint on Attribute)** นิพจน์โอซีแอลสามารถใช้ในการระบุค่าเริ่มต้น (Initial Value) และค่าที่ได้ภายหลัง (Subsequent Value) สำหรับแอตทริบิวส์ของคลาสซิไฟเออร์ โดยใช้คำสวงน **init** และ **derive** ตามลำดับ [26] การใช้นิพจน์โอซี-แอลในการแสดงข้อจำกัดของแอตทริบิวส์จะต้องเขียนให้อยู่ในรูปแบบดังต่อไปนี้

```
context <classifier name>::<attribute name>:<type>
init:<ocl expression>
derive: <ocl expression>
```

รูปที่ 2-34 แสดงรูปแบบการเขียนข้อจำกัดของแอตทริบิวส์ในรูปแบบข้อความ

context School :: tuition : float
init: 2500.00

รูปที่ 2-35 แสดงตัวอย่างการเขียนข้อจำกัดของแอตทริบิวส์ [26]

2.6 งานวิจัยที่เกี่ยวข้อง

การสร้างกรณีทดสอบจากแผนภาพยูเอ็มแอลได้ถูกนำเสนอไว้ในงานวิจัยดังต่อไปนี้

เศรษฐพงศ์ ลีฬหรัตนรักษ์ [29] นำเสนอวิธีสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสในแผนภาพยูสเคส โดยนักวิเคราะห์ต้องให้ข้อมูลเพิ่มเติมแก่ยูสเคส ได้แก่ ข้อมูลนำเข้า และลำดับเหตุการณ์สำหรับประกอบการสร้างกรณีทดสอบ จากนั้นจึงนำข้อมูลรายละเอียดยูสเคสมาช่วยในการสร้างกรณีทดสอบ โดยกรณีทดสอบที่สร้างขึ้นจะครอบคลุมทุกลำดับเหตุการณ์ที่เป็นไปได้ของแต่ละยูสเคส พร้อมทั้งพัฒนาเครื่องมือสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคสตามวิธีการที่ได้นำเสนอไว้

Abdurazik และOffutt [13] ได้นำเสนอเทคนิคสำหรับสร้างกรณีทดสอบจากแผนภาพสแตทแมทชีนซึ่งใช้ในการทดสอบระดับของคลาส เทียบได้กับการทดสอบในระดับหน่วยของซอฟต์แวร์แบบสัญนิยม วิธีนี้ใช้ตารางการแปลงสถานะ (State Transition Table) เป็นข้อมูลนำเข้าในการทดสอบและสร้างกรณีทดสอบสำหรับเกณฑ์ความครอบคลุมฟูลล์เพรดิเคต (Full-Predicate Coverage)

Kansomkeat และRivepiboon [8] นำเสนอวิธีในการสร้างลำดับการทดสอบจากแผนภาพสแตทแมทชีนโดยแปลงแผนภาพให้อยู่ในรูปของกราฟควบคุมกระแสการทดสอบ (Testing Flow Graph-TFG) ซึ่งถูกใช้เพื่อสร้างลำดับของการทดสอบ

Pilskalns และคณะ [30] นำเสนอแนวทางในการทดสอบแบบจำลองการออกแบบยูเอ็ม-แอล เพื่อเปิดเผยความไม่ต้องกัน (Inconsistency) ของแผนภาพ ระหว่างแผนภาพคลาส แผนภาพซีเควนซ์ และโอซีแอล โดยสร้างแบบจำลองรวม (Aggregate Model) ซึ่งเป็นผลที่ได้จากการผสมผสานข้อมูลจากแผนภาพคลาสและแผนภาพซีเควนซ์เข้าด้วยกัน และใช้โอซีแอลเป็นส่วนที่ใช้ในการตรวจสอบความถูกต้องของการทดสอบ กรณีทดสอบที่สร้างขึ้นในงานวิจัยนี้สร้างขึ้นเพื่อทดสอบแบบจำลองรวม ไม่ได้สร้างขึ้นเพื่อนำไปทดสอบกับโปรแกรมจริง

เนื่องด้วยผลจากการทดลองของ Abdurazik และOffutt [6] ซึ่งได้เปรียบเทียบประเภทของข้อผิดพลาดในการทดสอบโดยใช้กรณีทดสอบที่ได้จากแผนภาพสถานะและกรณีทดสอบที่ได้จากแผนภาพซีเควนซ์ ผลการทดลองแสดงให้เห็นว่าแผนภาพสแตทแมทชีน สามารถเปิดเผย

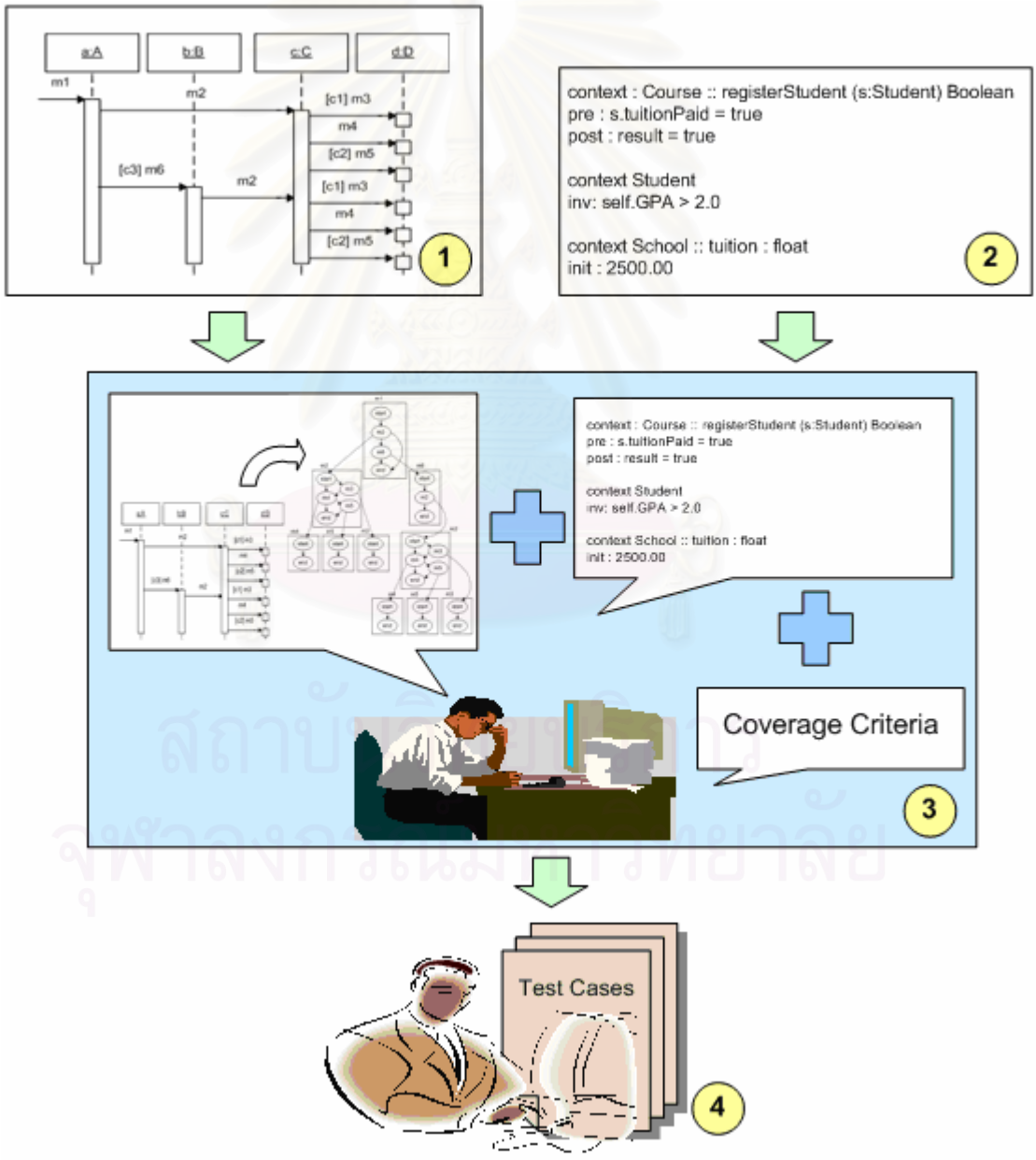
ข้อผิดพลาดในระดับหน่วยได้มากกว่าระดับบูรณาการ และในทางกลับกัน แผนภาพซีเควนซ์ก็สามารถเปิดเผยข้อผิดพลาดในระดับบูรณาการได้มากกว่าระดับหน่วยเช่นกัน เนื่องด้วยงานวิจัยนี้สนใจในส่วนของการทดสอบระดับบูรณาการจึงเลือกแผนภาพซีเควนซ์มาใช้ในการสร้างกรณีทดสอบ

งานวิจัยนี้สนใจที่จะสร้างหลักการในการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์โดยคำนึงถึงเกณฑ์ความครอบคลุมในการทดสอบทุกเส้นทางไอการ์ซีเอฟจี เนื่องจากเกณฑ์นี้ครอบคลุมทุกเส้นทางการเรียกใช้งานเมท็อด ในขณะที่ชุดเส้นทางของอีก 3 เกณฑ์ที่เหลือนั้นเป็นเส้นทางย่อยที่อยู่ภายในชุดเส้นทางที่ได้จากเกณฑ์ทุกเส้นทางไอการ์ซีเอฟจี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
แนวคิดและขั้นตอนการดำเนินงานวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อสร้างแนวคิดและหลักการในการสร้างกรณีทดสอบที่สามารถนำไปใช้ในการทดสอบซอฟต์แวร์เชิงวัตถุในระดับบูรณาการ โดยกรณีทดสอบของงานวิจัยนี้ได้มาจากข้อมูลในแผนภาพซีเควนซ์และโอซีแอล ภาพรวมของการสร้างกรณีทดสอบของงานวิจัยนี้แสดงไว้ดังรูปที่ 3-1



รูปที่ 3-1 ภาพรวมในการออกแบบและสร้างกรณีทดสอบของงานวิจัยนี้

จากรูปที่ 3-1 สามารถแบ่งออกได้เป็น 4 ส่วนหลัก โดยรายละเอียดของแต่ละส่วนมีดังต่อไปนี้

ส่วนที่ (1) คือแผนภาพซีเควนซ์ที่ได้จากขั้นตอนการออกแบบซอฟต์แวร์เชิงวัตถุ โดยองค์ประกอบและรายละเอียดของแผนภาพซีเควนซ์ที่นำมาใช้สร้างกรณีทดสอบในงานวิจัยนี้จะต้องเป็นไปตามมาตรฐานของยูเอ็มแอลรุ่นที่ 2.0 ซึ่งถูกกำหนดขึ้นโดยองค์กรโอเอ็มจี (OMG) ดังที่ได้แสดงรายละเอียดไว้ในหัวข้อที่ 2.3 งานวิจัยนี้รองรับแผนภาพซีเควนซ์ที่มีเมจเสจเริ่มต้น 1 เมจเสจและถูกออกแบบโดยใช้คอมบายด์แฟรกเมนต์เพียง 3 ประเภท ได้แก่ อัลเทอร์เนทีฟ ออบ-ชัน และลูป

ส่วนที่ (2) คือภาษาข้อจำกัดเชิงวัตถุหรือโอซีแอลซึ่งเป็นภาษาทางการที่ใช้ระบุข้อจำกัดต่าง ๆ ขององค์ประกอบในแบบจำลองยูเอ็มแอล รูปแบบของโอซีแอลที่ใช้ในงานวิจัยนี้จะต้องอยู่ในรูปแบบข้อความ (Textual Form) เท่านั้น และเป็นไปตามมาตรฐานที่กำหนดโดยโอเอ็มจี (รายละเอียดในการเขียนโอซีแอลอยู่ในหัวข้อที่ 2.5)

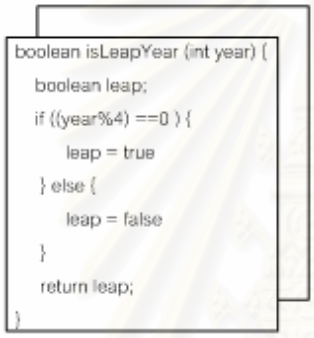
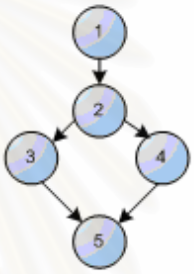
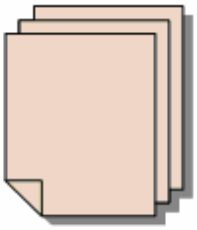
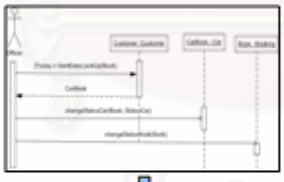
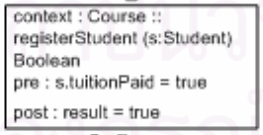
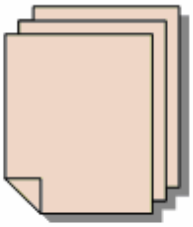
ส่วนที่ (3) เป็นส่วนที่ผู้วิจัยดำเนินการออกแบบ สร้างแนวคิดและขั้นตอนเพื่อให้สามารถสร้างกรณีทดสอบจากแผนภาพซีเควนซ์และโอซีแอลได้ โดยประยุกต์ใช้กราฟควบคุมกระแสไออาร์ซีเอฟจีของ Rountev และคณะ [7] ด้วยการเพิ่มเติมองค์ประกอบที่แสดงถึงเงื่อนไขการเรียกใช้งาน ดังที่ปรากฏอยู่ในกราฟควบคุมกระแสของ McCabe [3] พร้อมทั้งคิดค้นหลักการสำหรับแปลงแผนภาพซีเควนซ์ให้อยู่ในรูปแบบของกราฟกระแสไออาร์ซีเอฟจี และวิธีที่จะทำให้ชุดของกรณีทดสอบที่สร้างขึ้นเป็นไปตามเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจี (All-IRCFG Path Criterion) (รายละเอียดของเกณฑ์ความครอบคลุมดังกล่าวได้ถูกนำเสนอไว้ในหัวข้อที่ 2.4) จากแนวคิดและหลักการที่สร้างขึ้น ผู้วิจัยจะนำมาพัฒนาเป็นเครื่องมือสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ เพื่อพิสูจน์ว่าแนวคิดและหลักการดังกล่าวสามารถนำไปใช้งานได้จริงและเพื่อช่วยลดระยะเวลาในการสร้างกรณีทดสอบ

ส่วนที่ (4) คือกรณีทดสอบซึ่งเป็นผลลัพธ์ที่ได้จากแนวคิดที่สร้างขึ้นจากส่วนที่ (3) กรณีทดสอบนี้สามารถนำไปใช้ในการทดสอบซอฟต์แวร์เชิงวัตถุในระดับการทดสอบแบบบูรณาการ เพื่อให้มั่นใจได้ว่าการทำงานระหว่างอ็อบเจกต์ของคลาสภายในซอฟต์แวร์ที่สร้างขึ้นสามารถทำงานร่วมกันได้อย่างถูกต้องและตรงต่อเงื่อนไขและข้อจำกัดต่าง ๆ ที่ได้กำหนดไว้ในขั้นของการออกแบบ

กรณีทดสอบที่ได้จากงานวิจัยนี้ สามารถใช้ทดสอบความถูกต้องของการเรียกใช้งานเมท็อดเท่านั้น คือถูกสร้างขึ้นโดยมุ่งเน้นในส่วนของข้อมูลหรือเงื่อนไขที่จะทำให้การทดสอบ

สามารถดำเนินการไปตามกราฟควบคุมกระแสในการเรียกใช้เมท็อด โดยไม่คำนึงถึงการเรียกใช้งานในทุกประโยคคำสั่ง (Statement) ภายในเมท็อดหนึ่ง ๆ และไม่คำนึงว่าแต่ละคลาสถูกทดสอบในระดับหน่วยมาแล้วหรือไม่

ตารางที่ 3-1 แสดงการเปรียบแนวคิดการสร้างกรณีทดสอบของงานวิจัยนี้กับแนวคิดการสร้างกรณีทดสอบที่ใช้กับซอฟต์แวร์แบบสัญนิยม

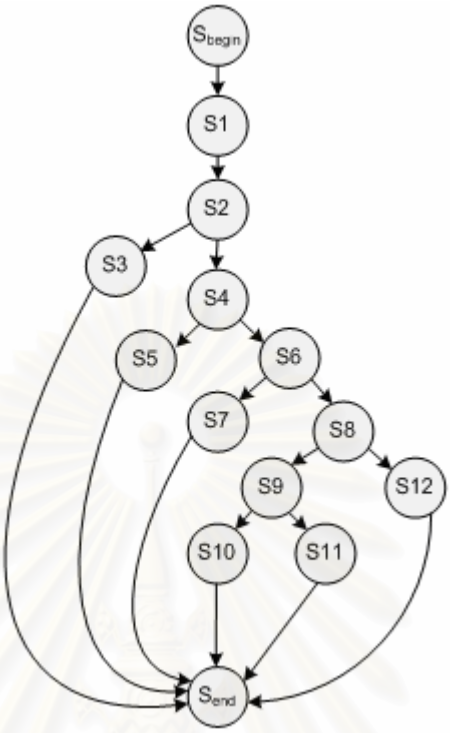
	แหล่งที่มา (Source)	**กราฟควบคุมกระแส** (Control Flow Graph)	**ผลลัพธ์** (Result)
แนวคิดของซอฟต์แวร์แบบสัญนิยม	 ซอร์สโค้ด	 กราฟควบคุมกระแสแทนทุกประโยคคำสั่งในซอร์สโค้ด	 กรณีทดสอบ
แนวคิดของงานวิจัยนี้	แผนภาพซีเควนซ์   โอซีแอล	กราฟควบคุมกระแสที่แสดงถึงลำดับของการเรียกใช้เมท็อด	 กรณีทดสอบ

ตัวอย่างของกลุ่มกรณีทดสอบที่คำนึงถึงการเรียกใช้งานในทุกประโยคคำสั่งถูกแสดงไว้ในตารางที่ 3-2 ซึ่งเป็นไปตามเส้นทางของกราฟควบคุมกระแสในรูปที่ 3-3 โดยแต่ละโหนดในเส้นทางเป็นตัวแทนของประโยคคำสั่งภายในซอร์สโค้ดฟังก์ชั่นชื่อ getNumDaysInMonth ดังแสดงไว้ในรูปที่ 3-2

```
S1 getNumDaysInMonth (int month, int year) {
S2      if (year < 1) {
S3          printf("YearOutOfBound");
        } else {
S4          If (month ==1 || month ==3 || month ==5 || month ==7 ||
                month ==10 || month ==12) {
S5              printf("number of Days = 32");
S6          } else if (month ==4 || month ==6 || month ==9 || month ==11 {
S7              printf("number of Days = 30");
S8          } else if (month ==2) {
S9              If ((year%4)==0) {
S10                 printf("number of Days = 29");
                } else {
S11                 printf("number of Days = 28");
                }
            } else {
S12             printf("MonthOutOfBounds");
            }
        }
    }
```

รูปที่ 3-2 แสดงตัวอย่างซอร์สโค้ดภาษาซี (C) ในฟังก์ชั่นชื่อ getNumDaysInMonth

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3-3 แสดงกราฟควบคุมกระแสของฟังก์ชัน getNumDaysInMonth

ตารางที่ 3-2 แสดงกรณีทดสอบที่เป็นไปตามเส้นทางในรูปที่ 3-3

Test Case	Path
(year = 0, month = 1)	{ S_{begin}, S1, S2, S3, S_{end} }
(year = 1901, month = 1)	{ S_{begin} , S1, S2, S4, S5, S_{end} }
(year = 1901, month = 2)	{ S_{begin} , S1, S2, S4, S6, S8, S9, S11, S_{end} }
(year = 1904, month = 2)	{ S_{begin} , S1, S2, S4, S6, S8, S9, S10, S_{end} }
(year = 1901, month = 4)	{ S_{begin} , S1, S2, S4, S6, S7, S_{end} }
(year = 1901, month = 0)	{ S_{begin} , S1, S2, S4, S6, S8, S12, S_{end} }

ตัวอย่างของกลุ่มกรณีทดสอบที่คำนึงถึงการเรียกใช้งานเฉพาะเมท็อดจากซอร์ดโค้ดในรูปที่ 3-4 แสดงไว้ในตารางที่ 3-3 โดยเส้นทางที่ถูกดำเนินการด้วยชุดกรณีทดสอบ คือเส้นทางของเมท็อดเพียงอย่างเดียวเท่านั้น

```
public void checkWithdraw(int checkNumber) {
        double amount;
        double balance;
        Check theCheck = new Check();
        CheckingAccount account = new CheckingAccount();

        amount = theCheck.Amount;
        balance = account.Balance;

        if (balance>=amount) {

                account.addDebitTransaction(checkNumber,amount);
                        …
                        if  (…)
                        {……}
                        …
                account.storePhotoOfCheck(theCheck);
                        …
        }
        else {
                account.addinsuffientFundFee();
                        …
                        …
                account.noteReturnedCheck(theCheck);
                this.returnCheck (theCheck);
        }
}
```

รูปที่ 3-4 แสดงตัวอย่างซอร์สโค้ดภาษาซีชาร์ป (C#) ในเมท็อดชื่อ checkWithdraw

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3-3 แสดงกรณีทดสอบในการเรียกใช้งานเมท็อดของซอร์สโค้ดในรูปที่ 3-4

Test case	Method Path Sequence	Expected Result
(balance = 1000, amount = 1000)	{ addDebitTransaction, storePhotoOfCheck }	account.Balance = 0
(balance = 2000, amount = 1000)	{ addDebitTransaction, storePhotoOfCheck }	account.Balance =1000
(balance = 1000, amount = 2000)	{ addInsuffientFundFee, noteReturnedCheck, returnCheck }	-

ขั้นตอนในการดำเนินงานวิจัยเพื่อสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ ประกอบด้วย 4 ขั้นตอนหลัก ได้แก่ (1) ศึกษาแนวทางในการแปลงแผนภาพซีเควนซ์ไปเป็นไอออาร์ซีเอฟจีที่รองรับคอมบายด์แฟรกเมนต์และสร้างหลักในการแปลง (2) ศึกษาและหาหลักการในการสร้างชุดเส้นทางไอออาร์ซีเอฟจีที่สมบูรณ์ ค่าของข้อมูลทดสอบ และวิธีการสร้างกรณีทดสอบ (3) ออกแบบและพัฒนาเครื่องมือในการสร้างกรณีทดสอบและ (4) ประเมินการใช้เครื่องมือและความสามารถของเครื่องมือ

3.1 ศึกษาแนวทางในการแปลงแผนภาพซีเควนซ์เป็นไอออาร์ซีเอฟจีที่รองรับคอมบายด์แฟรกเมนต์และสร้างหลักในการแปลง

จากกราฟควบคุมกระแสไอออาร์ซีเอฟจีที่ Rountev และคณะ [7] ได้นำเสนอไว้ ผู้วิจัยเห็นว่าจะเป็นประโยชน์อย่างมากหากนำมาใช้เป็นแนวทางในการสร้างกรณีทดสอบ โดยขั้นตอนในการดำเนินการเพื่อให้สามารถสร้างกราฟควบคุมกระแสไอออาร์ซีเอฟจีจากแผนภาพซีเควนซ์ที่งานวิจัยนี้รองรับ มีดังต่อไปนี้

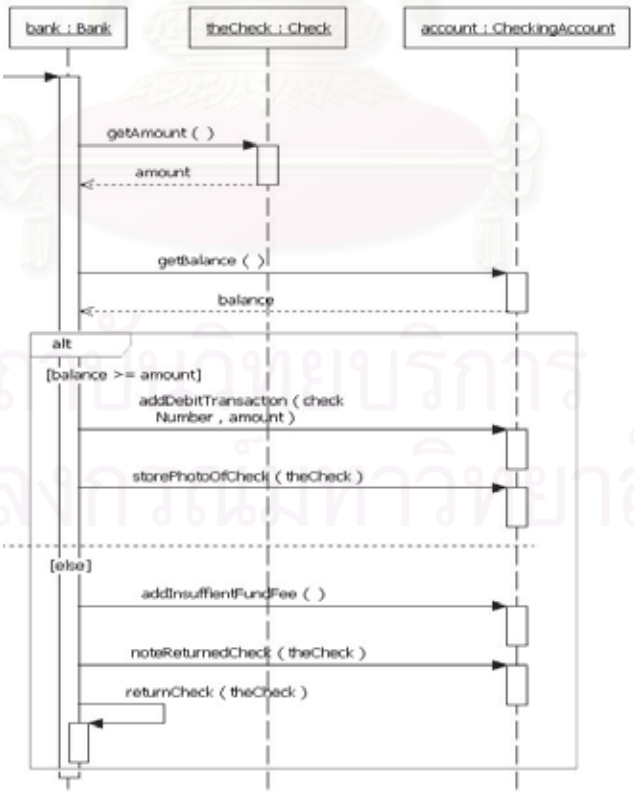
1. **ศึกษาแนวทางการแปลงและหาวิธีให้กราฟกระแสไอออาร์ซีเอฟจีสามารถรองรับแผนภาพซีเควนซ์ที่ออกแบบด้วยคอมบายด์แฟรกเมนต์**

เนื่องจากกราฟควบคุมกระแสไอออาร์ซีเอฟจี [7] ถูกสร้างขึ้นจากแผนภาพซีเควนซ์รุ่นที่ 1.x ดังนั้น คอมบายด์แฟรกเมนต์ที่แสดงให้เห็นถึงตรรกะในการเรียกใช้งานเมท็อดซึ่งได้ถูกเพิ่มเติมเข้ามาใหม่ในแผนภาพซีเควนซ์ของยูเอ็มแอลรุ่นที่ 2.0 จึงไม่สามารถแปลงให้อยู่ในรูปของกราฟควบคุมกระแสไอออาร์ซีเอฟจีดังกล่าวได้ เนื่องจากโหนดในไอออาร์ซีเอฟจีทุกโหนดเป็นตัวแทนของ

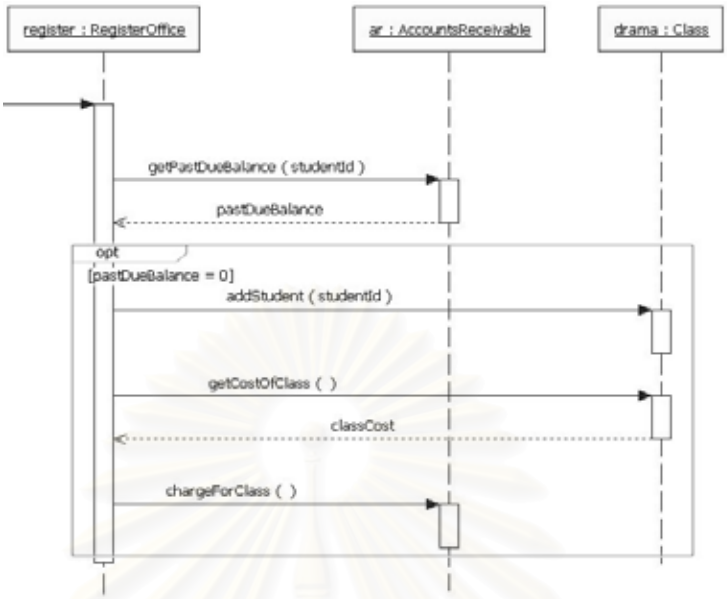
เมท็อดเพียงอย่างเดียวเท่านั้น ในส่วนนี้ผู้วิจัยต้องการให้ไออาร์ซีเอฟจีนี้สามารถรองรับแผนภาพซีเควนซ์ของยูเอ็มแอลรุ่นที่ 2.0 ได้ โดยจำกัดเพียง 3 ประเภทของคอมบายด์แฟรกเมนต์เท่านั้น ได้แก่ อัลเทอร์เนทีฟ ออบชัน และลูป โดยแสดงรายละเอียดโดยย่อของแต่ละคอมบายด์แฟรก-เมนต์ไว้ในตารางที่ 3-4 และแสดงตัวอย่างแผนภาพซีเควนซ์ที่ออกแบบด้วยคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟ ออบชัน และลูป ดังรูปที่ 3-5 3-6 และ 3-7 ตามลำดับ

ตารางที่ 3-4 สรุปประเภทคอมบายด์แฟรกเมนต์ที่งานวิจัยนี้รองรับ

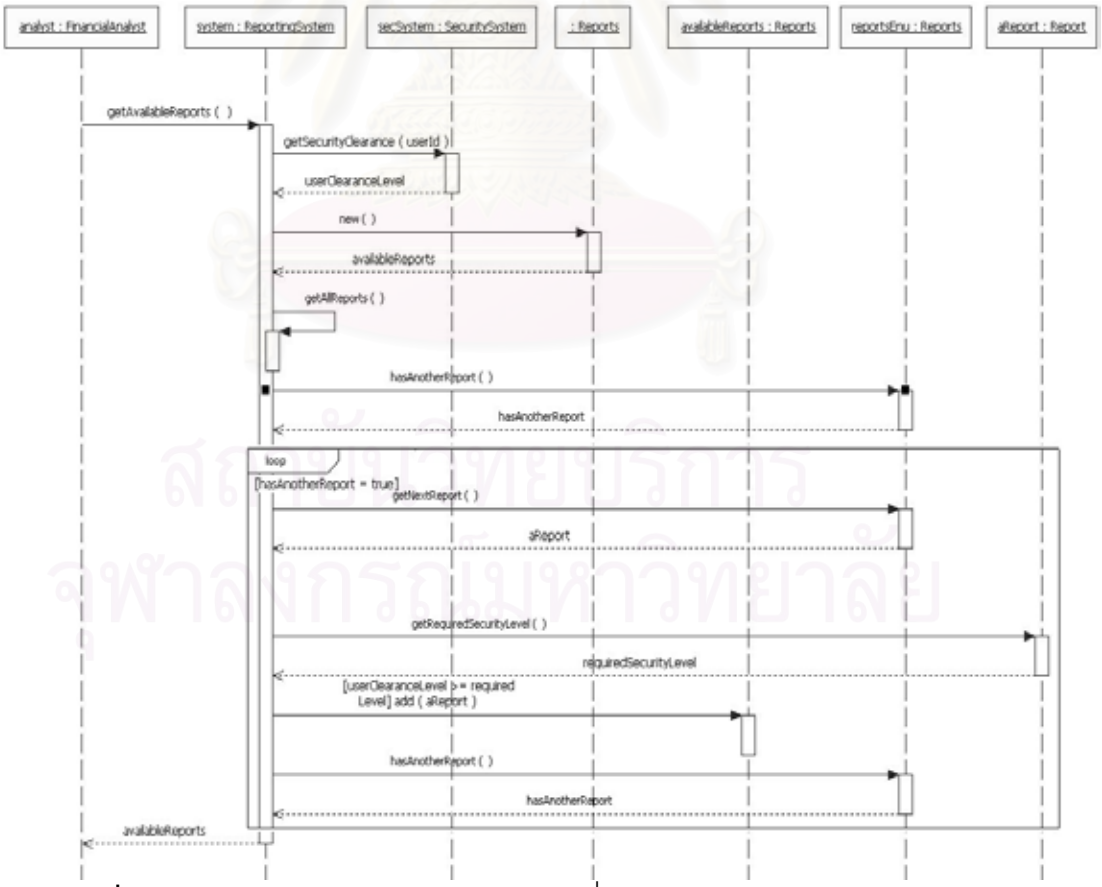
คอมบายด์แฟรกเมนต์	**อินเตอร์แรคชันโอเปอร์เรเตอร์**	**ความหมาย**
อัลเทอร์เนทีฟ	alt	แสดงถึงทางเลือกของพฤติกรรมซึ่งอยู่ในรูปแบบของตรรกศาสตร์ “if then else”
ออบชัน	opt	แสดงถึงทางเลือกของพฤติกรรมซึ่งอาจเกิดขึ้นหรือไม่เกิดขึ้นก็ได้
ลูป	loop	แสดงถึงการวนทำซ้ำของเมจเสจภายในตัวถูกดำเนินการ



รูปที่ 3-5 แสดงตัวอย่างของแผนภาพซีเควนซ์ที่ออกแบบโดยใช้คอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟ [24]

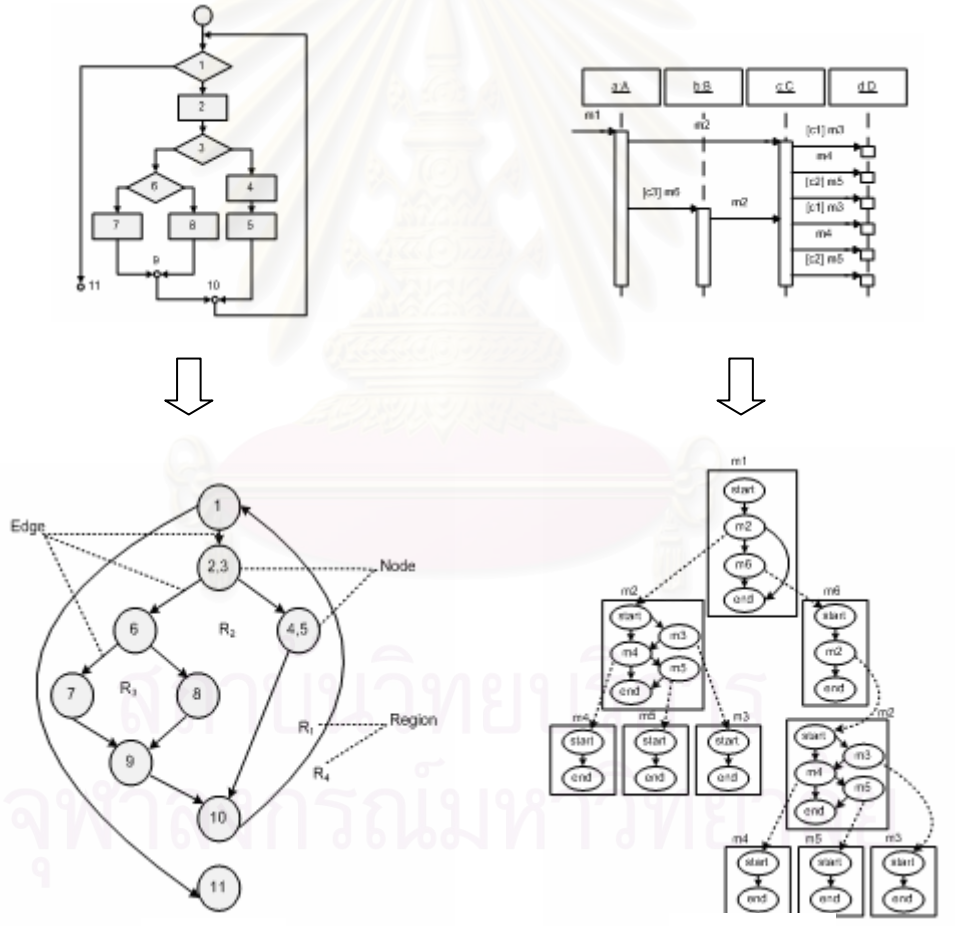


รูปที่ 3-6 แสดงตัวอย่างของแผนภาพซีเควนซ์ที่ออกแบบโดยใช้คอมบายด์แฟรกเมนต์ประเภทออปชัน [24]



รูปที่ 3-7 แสดงตัวอย่างของแผนภาพซีเควนซ์ที่ออกแบบโดยใช้คอมบายด์แฟรกเมนต์ประเภทลูป [24]

ดังนั้นเพื่อให้ไออาร์ซีเอฟจีสามารถรองรับตรรกะในการเรียกใช้งานของเมท็อดตามที่กำหนดไว้ในแผนภาพซีเควนซ์ด้วยคอมบายด์แฟรกเมนต์ 3 ประเภทตามที่ได้กล่าวไว้ดังตารางที่ 3-4 ผู้วิจัยจะนำแนวคิดของกราฟควบคุมกระแส (Control Flow Graph) ที่ถูกนำเสนอโดย McCabe มาประยุกต์ใช้กับไออาร์ซีเอฟจี เนื่องจากกราฟกระแสควบคุมของ McCabe สร้างขึ้นเพื่อรองรับการแปลงจากผังงาน (Flow Chart) หรือ ซอร์สโค้ด (Source Code) ดังนั้นกราฟนี้จึงแสดงให้เห็นถึงการควบคุมทางตรรกะที่มีอยู่ในผังงานหรือซอร์ดโค้ดได้ (เช่น การวนทำซ้ำ เงื่อนไขการทำงาน เป็นต้น) แต่ละโหนดภายในกราฟจะแทนประโยคคำสั่งตั้งแต่หนึ่งประโยคขึ้นไปหรือแสดงถึงเงื่อนไขการใช้ประโยคคำสั่ง ในขณะที่โหนดภายในไออาร์ซีเอฟจีจะเป็นตัวแทนของเมท็อดเพียงอย่างเดียวเท่านั้น ดังรูปที่ 3-8

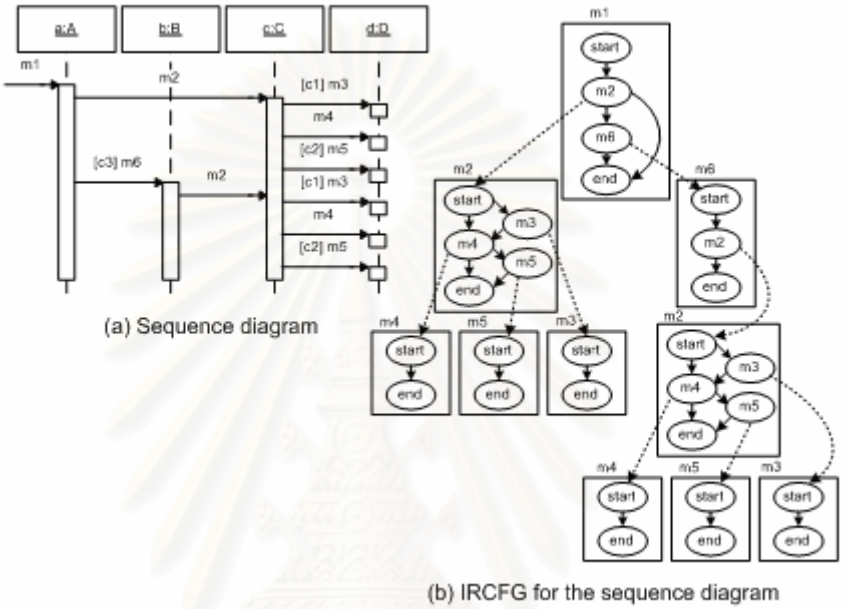


(a) CFG (b) IRCFG

รูปที่ 3-8 (a) แสดงรูปแบบกราฟควบคุมกระแสของ McCabe [3] และ(b) แสดงรูปแบบไออาร์ซีเอฟจีของ Rountev และคณะ [7]

2. **กำหนดขั้นตอนและหลักการในการสร้างกราฟกระแสไออาร์ซีเอฟจีจากแผนภาพซีเควนซ์**

เนื่องจากแนวทางการแปลงแผนภาพซีเควนซ์ให้อยู่ในรูปของไออาร์ซีเอฟจีที่แสดงไว้ในงานวิจัยของ Rountev และคณะ [7] ได้ถูกอธิบายด้วยตัวอย่างดังรูปที่ 3-9



รูปที่ 3-9 ตัวอย่าง (a) แผนภาพซีเควนซ์และ (b) ไออาร์ซีเอฟจีของแผนภาพซีเควนซ์ ซึ่ง Rountev และคณะ [7] ใช้เพื่อประกอบการอธิบาย

งานวิจัยดังกล่าวไม่มีการสร้างหลักการหรือกำหนดวิธีในการแปลงไว้ให้เห็นอย่างชัดเจน ดังนั้นหลังจากปรับกราฟควบคุมกระแสไออาร์ซีเอฟจี ให้อยู่ในรูปแบบที่สามารถรองรับตรรกะได้แล้ว ผู้วิจัยจะสร้างหลักการและขั้นตอนในการแปลงไว้ในงานวิจัยนี้ด้วย เพื่อไม่ให้เกิดความคลุมเครือในการสร้าง และเพื่อสามารถนำไปใช้ในขั้นของการออกแบบและพัฒนาเครื่องมือที่ช่วยในการสร้างกรณีทดสอบ (หัวข้อที่ 3.3) ได้ง่ายและสะดวกยิ่งขึ้น

3.2 ศึกษาและหาหลักการในการสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ ค่าของข้อมูลทดสอบ และวิธีการสร้างกรณีทดสอบ

หลังจากได้กราฟกระแสไออาร์ซีเอฟจีที่รองรับคอมบายด์แฟรกเมนต์ทั้ง 3 ประเภทแล้ว ถัดมาเป็นขั้นตอนที่ผู้วิจัยจะศึกษาและหาวิธีให้ได้มาซึ่งค่าของข้อมูลทดสอบ เงื่อนไขก่อนและหลังดำเนินการเมท็อด และกรณีทดสอบที่เป็นไปตามเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจี โดยมีรายละเอียดดังต่อไปนี้

1. หาวิธีในการสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ให้ครบตามเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจี

2. หาวิธีในการกำหนดเงื่อนไขก่อนและหลังดำเนินการของเมท็อดที่ปรากฏในแผนภาพซีเควนซ์

3. หาวิธีในการสร้างค่าของข้อมูลทดสอบ 3 ชนิด คือ ข้อมูลชุดอักขระ (String) ข้อมูลตัวเลขจำนวนเต็ม (Integer) และข้อมูลชนิดบูลีน (Boolean) ซึ่งมีวิธีในการสร้างค่าที่แตกต่างกัน

4. ในส่วนของการสร้างกรณีทดสอบ จะหาวิธีในการผนวกโอซีแอลในส่วนของเงื่อนไขก่อนและหลังดำเนินการของเมท็อดเข้ามาเป็นข้อมูลเบื้องต้นและผลลัพธ์ที่คาดหวังสำหรับกรณีทดสอบที่สร้างขึ้น โดยผู้วิจัยจะสร้างหลักและขั้นตอนในการสร้างกรณีทดสอบอย่างเป็นระบบเพื่อให้เห็นถึงแนวคิดในการสร้างกรณีทดสอบได้อย่างชัดเจน

3.3 ออกแบบและพัฒนาเครื่องมือในการสร้างกรณีทดสอบ

ขั้นตอนนี้เป็นการออกแบบและพัฒนาเครื่องมือสร้างกรณีทดสอบ โดยนำหลักการและแนวคิดที่ได้จากขั้นตอนที่ 3.1 และ 3.2 มาใช้เป็นหลักในการสร้างกรณีทดสอบเพื่อชี้ให้เห็นว่าหลักการในการสร้างกรณีทดสอบที่ได้นำเสนอในงานวิจัยนี้สามารถนำไปใช้งานได้จริง โดยสภาพแวดล้อมที่ใช้ในการพัฒนามีรายละเอียดดังต่อไปนี้

1. ฮาร์ดแวร์ (Hardware) ในการพัฒนาเครื่องมือประกอบด้วย
   - เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผลเพนเทียมเอ็ม 1.8 กิกะ-เฮิรตซ์ (Pentium M 1.8 GHz.)
   - หน่วยความจำสำรอง (RAM) 768 เมกะไบต์ (768 MB)
   - ฮาร์ดดิสก์ (Harddisk) 60 กิกะไบต์ (60 GB)

2. ซอฟต์แวร์ (Software) ในการพัฒนาเครื่องมือประกอบด้วย
   - ระบบปฏิบัติการ (Operating system) ไมโครซอฟต์วินโดวส์เอ็กซ์พี โพรเฟซ-ชันแนล (Microsoft XP Professional)
   - ภาษาที่ใช้ในการพัฒนาได้แก่ ภาษาซีชาร์ปดอทเน็ตโดยใช้โปรแกรมไมโครซอฟต์วิชวลสตูดิโอดอทเน็ต 2005 (Microsoft Visual Studio.NET 2005)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- เครื่องมือช่วยสร้างเอกสารแผนภาพซีเควนซ์และแปลงเอกสารแผนภาพซีเควนซ์เป็นเอกสารเอ็กซ์เอ็มแอล (XML Document) ได้แก่ วิชัวล์พาราแดม สำหรับ ยูเอ็มแอล เวอร์ชัน 6.0 (Visual Paradigm for UML 6.0)

3.4 ประเมินการใช้เครื่องมือและความสามารถของเครื่องมือ

ในส่วนนี้จะประเมินเครื่องมือที่พัฒนาขึ้น ด้วยการนำแผนภาพซีเควนซ์ซึ่งอยู่ในรูปแบบของเอกสารเอกซ์เอ็มแอลมาเป็นข้อมูลนำเข้าของเครื่องมือ โดยคุณสมบัติและที่มาของหน่วยตัวอย่างมีดังต่อไปนี้

3.4.1 คุณสมบัติของหน่วยตัวอย่าง

หน่วยตัวอย่างของงานวิจัยนี้ คือเอกสารแผนภาพซีเควนซ์ที่ถูกออกแบบโดยใช้คอมบายด์แฟรกเมนต์อย่างน้อย 1 ใน 3 ประเภทคือ อัลเทอร์เนทีฟ ออปชัน และลูป และเนื่องด้วยงานวิจัยนี้สนใจแค่ในส่วนของการสร้างกรณีทดสอบในระดับบูรณาการ (Integration Testing) เท่านั้น ไม่ได้คำนึงถึงการสร้างกรณีทดสอบเพื่อนำไปใช้ในการทดสอบระบบ (System Testing) กรณีทดสอบแต่ละชุดจะถูกสร้างขึ้นโดยไม่มีความเกี่ยวเนื่องกัน ดังนั้นแผนภาพซีเควนซ์ที่นำมาเป็นหน่วยตัวอย่าง จึงไม่จำเป็นต้องเป็นแผนภาพของระบบเดียวกัน กลุ่มของหน่วยตัวอย่างที่ดีควรมีการคละประเภทของคอมบายด์แฟรกเมนต์ครบทั้ง 3 ประเภท เพื่อให้การดำเนินการในส่วนของหลักการแปลงแผนภาพซีเควนซ์เป็นกราฟควบคุมกระแสไออาร์ซีเอฟที่ผู้วิจัยได้สร้างขึ้นถูกทดสอบครบทุกส่วน

3.4.2 การเลือกหน่วยตัวอย่าง

ในเบื้องต้นผู้วิจัยได้กำหนดจำนวนของหน่วยตัวอย่างที่จะนำมาใช้ทดสอบและประเมินผลเครื่องมือ โดยจำแนกตามประเภทของคอมบายด์แฟรกเมนต์ทั้ง 3 ประเภท กล่าวคือจะต้องมีจำนวนแผนภาพซีเควนซ์อย่างละ 2 แผนภาพต่อหนึ่งประเภทของคอมบายด์แฟรกเมนต์ ดังนั้นเมื่อรวมแล้วจะได้แผนภาพซีเควนซ์ที่เป็นหน่วยตัวอย่างทั้งสิ้น 6 แผนภาพ โดยแผนภาพซีเควนซ์ทั้งหมดจะต้องถูกแปลงกลับมาจากซอร์สโค้ดของ 6 ยูสเคส ดังนั้นซอร์สโค้ดของแต่ละยูส-เคสที่จะนำมาใช้ในงานวิจัยนี้ จะต้องถูกเขียนด้วยตรรกะและเงื่อนไขที่ตรงตามแต่ละประเภทของคอมบายด์แฟรกเมนต์ และเขียนขึ้นโดยใช้ภาษาเชิงวัตถุ (Object-Oriented Programming Language) เหตุที่ผู้วิจัยกำหนดให้แผนภาพซีเควนซ์ที่จะนำมาเป็นหน่วยตัวอย่างต้องมาจากการแปลงกลับของซอร์สโค้ด เนื่องมาจากในส่วนของการประเมินผลเครื่องมือ จะต้องนำกรณีทดสอบ

ที่สร้างขึ้นด้วยเอกสารแผนภาพซีเควนซ์ (สร้างโดยเครื่องมือ) มาเปรียบเทียบกับกรณีทดสอบที่สร้างขึ้นด้วยซอร์สโค้ดและรายละเอียดของยูสเคส (สร้างโดยผู้เชี่ยวชาญ)

จากข้อจำกัดและคุณสมบัติของหน่วยตัวอย่างข้างต้น ผู้วิจัยได้ประเมินซอร์สโค้ดซึ่งเป็นงานที่ได้รับมอบหมาย (Assignment) ของนิสิตปริญญามหาบัณฑิตที่ผ่านการเรียนวิชาเทคโนโลยีเชิงวัตถุ (Object-Oriented Technology) เป็นจำนวน 8 โครงงาน พบว่ามีเพียง 1 โครงงานที่สามารถนำมาใช้เป็นหน่วยตัวอย่างได้ ส่วนอีก 7 โครงงานที่เหลือนั้น ไม่ปรากฏส่วนของเงื่อนไขการเรียกใช้งานเมท็อด จึงไม่สามารถนำมาใช้เป็นหน่วยตัวอย่างของงานวิจัยนี้ได้ ผู้วิจัยคาดว่าอาจเป็นเพราะข้อกำหนดของซอฟต์แวร์ในแต่ละโครงงานไม่ได้ถูกสร้างขึ้นเพื่อรองรับการนำไปใช้งานจริงในภาคธุรกิจ ซึ่งโดยทั่วไปแล้วแต่ละธุรกิจจะมีเงื่อนไขในการใช้งานที่หลากหลายตามแต่กระบวนการของธุรกิจ (Business Process) ดังนั้นเพื่อให้ได้จำนวนหน่วยตัวอย่างครบตามที่กำหนดไว้ในเบื้องต้น ผู้วิจัยจึงขอความอนุเคราะห์ทางด้านข้อมูลของซอร์สโค้ดที่มีส่วนของเงื่อนไขการเรียกใช้งานเมท็อดจากบริษัทผลิตซอฟต์แวร์ (Software House) แห่งหนึ่ง เป็นจำนวนอีก 5 ยูสเคส ซึ่งซอร์สโค้ดเหล่านี้ถูกพัฒนาขึ้นเพื่อนำไปใช้งานในระบบธุรกิจจริง

3.4.3 ที่มาของหน่วยตัวอย่าง

หน่วยตัวอย่างของงานวิจัยนี้มีจำนวนทั้งสิ้น 6 หน่วย ซึ่งได้มาจากการแปลงกลับซอร์สโค้ดของยูสเคสจำนวน 6 ยูสเคส โดยซอร์สโค้ดทั้งหมดได้มาจาก

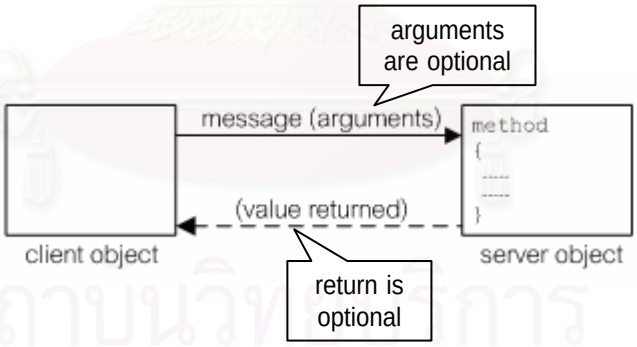
1. งานที่ได้รับมอบหมาย (Assignment) ของนิสิตปริญญามหาบัณฑิตที่ผ่านการเรียนวิชาเทคโนโลยีเชิงวัตถุ (Object-Oriented Technology) เป็นจำนวน 1 ยูสเคสที่พัฒนาด้วยภาษาซีชาร์ป (C# Language) ในรูปแบบของโปรแกรมเชิงวัตถุ ได้แก่ยูสเคสการซื้อตั๋ว (Buy Ticket)

2. ผู้วิจัยได้รับความอนุเคราะห์จากบริษัทผลิตซอฟต์แวร์ (Software House) ได้ให้ซอร์สโค้ดจำนวน 5 ยูสเคสที่พัฒนาด้วยภาษาจาวา (Java Language) ในรูปแบบของโปรแกรมเชิงวัตถุ แต่เนื่องจากข้อจำกัดทางด้านนโยบายและความปลอดภัยทางด้านข้อมูลของบริษัท จึงไม่สามารถให้ซอร์สโค้ดทั้งหมดของระบบมาได้ ซอร์สโค้ดที่ได้มาจึงเป็นเพียงบางส่วนเท่านั้น ซอร์-สโค้ดของยูสเคสที่ได้คือ ยูสเคสการเข้าใช้งาน (Login) การยอมรับอินวอยซ์(Approve Invoice) การโปรโมชันร่วมกับการขาย (Sale Promotion) การลดราคา(Calculate Discount) และการส่งสินค้า (Shipping Order)

จากนั้นผู้วิจัยนำซอร์สโค้ดทั้ง 6 ยูสเคสมาแปลงกลับเป็นเอกสารการออกแบบแผนภาพซีเควนซ์ด้วยเครื่องมือช่วยสร้างย้อนกลับ (Reverse Engineering) โปรแกรมบอร์แลนด์-ทูเก็ตเตอร์ (Borland Together) แต่เนื่องจากข้อจำกัดบางประการของโปรแกรม ผู้วิจัยจึงนำแผนภาพซีเควนซ์ที่ได้จากการแปลงกลับมาปรับแก้ดังนี้

1. เพิ่มแอคเตอร์ (Actor) ลงในแผนภาพซีเควนซ์ที่ถูกแปลงกลับ เนื่องจากโปรแกรมบอร์แลนด์ทูเก็ตเตอร์สามารถสร้างเมจเสจในแผนภาพซีเควนซ์ได้เฉพาะในส่วนการเรียกใช้งานภายในโปรแกรมเท่านั้น ผู้วิจัยจึงได้เพิ่มแอกเตอร์ที่เรียกใช้งานเมจเสจเริ่มต้นลงไปในแผนภาพซีเควนซ์ด้วย

2. แก้ไขเมจเสจจากพารามิเตอร์ (Parameter) ของเมท็อดเป็นอาร์กิวเมนต์ (Arguments) ที่ส่งไปเพื่อเรียกใช้งานเมท็อด เนื่องด้วยโปรแกรมบอร์แลนด์ทูเก็ตเตอร์สร้างเมจเสจด้วยการย้อนรอยจากพารามิเตอร์ของเมท็อด แต่ในการออกแบบแผนภาพซีเควนซ์โดยทั่วไป การร้องขอ (Invoke) เมท็อดจะเกิดขึ้นจากการส่งเมจเสจจากอ็อบเจกต์หนึ่งไปยังอีกอ็อบเจกต์หนึ่งโดยจะส่งอาร์กิวเมนต์ไปด้วยหรือไม่ก็ได้ [31, 32] ดังรูปที่ 3-10 แสดงการส่งเมจเสจของอ็อบเจกต์โดยอ็อบเจกต์ที่เป็นผู้ส่งเมจเสจเรียกว่าไคล์เอนท์อ็อบเจกต์ (Client Object) และอ็อบเจกต์ที่เป็นผู้รับเมจเสจเรียกว่า เซิร์ฟเวอร์อ็อบเจกต์ (Server Object)



รูปที่ 3-10 แสดงการโต้ตอบ (Interaction) ของไคล์เอนท์-เซิร์ฟเวอร์อ็อบเจกต์ (Client-server Object) [31]

รายละเอียดของแผนภาพซีเควนซ์ของทั้ง 6 แผนภาพที่ถูกแปลงกลับจากซอร์สโค้ดแสดงไว้ในตารางที่ 3-5 (จำนวนของไลฟ์ไลน์ไม่นับรวมแอกเตอร์) และรายละเอียดของคอมบายด์แฟรกเมนต์จำแนกตามประเภทแสดงไว้ในตารางที่ 3-6

ตารางที่ 3-5 ตารางแสดงรายละเอียดของแผนภาพซีเควนซ์ที่แปลงกลับจากซอร์สโค้ด

แผนภาพซีเควนซ์ของยูสเคส	จำนวนรายละเอียดของแผนภาพซีเควนซ์		
	เมจเสจ	ไลฟ์ไลน์	คอมบายด์แฟรกเมนต์
การซื้อตั๋ว	15	5	2
การเข้าใช้งาน	16	4	3
การยอมรับอินวอยซ์	8	4	4
การโปรโมชันร่วมกับการขาย	11	6	2
การลดราคา	10	5	4
การส่งสินค้า	14	7	2

ตารางที่ 3-6 ตารางแสดงรายละเอียดคอมบายด์แฟรกเมนต์ภายในแผนภาพซีเควนซ์ที่นำมาเป็นหน่วยตัวอย่างของงานวิจัยนี้

แผนภาพซีเควนซ์ของยูสเคส	จำนวนคอมบายด์แฟรกเมนต์ (จำแนกตามประเภท)			จำนวนคอมบายด์แฟรกเมนต์รวม
	อัลเทอร์เนทีฟ	ออปชัน	ลูป	
การซื้อตั๋ว	2	-	-	2
การเข้าใช้งาน	3	-	-	3
การยอมรับอินวอยซ์	2	1	1	4
การโปรโมชันร่วมกับการขาย	2	-	-	2
การลดราคา	4	-	-	4
การส่งสินค้า	2	-	-	2

จากตารางที่ 3-6 พบว่าจำนวนของคอมบายด์แฟรกเมนต์ประเภทออปชัน และลูป มีเพียงอย่างละ 1 ส่วนเท่านั้น ซึ่งถือว่ามีจำนวนน้อยเมื่อนำมาเทียบกับจำนวนของคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟ แต่เนื่องจากประเภทของคอมบายด์แฟรกเมนต์ไม่ได้มีผลกระทบต่อการสร้างค่าข้อมูลทดสอบของกรณีทดสอบ เพียงทำให้เกิดรูปแบบของการเชื่อมโยงระหว่างเมท็อดที่ต่างกันเมื่อนำไปสร้างเป็นกราฟควบคุมกระแสไออาร์ซีเอฟจี กลุ่มของหน่วยตัวอย่างที่ได้มามีประเภทของคอมบายด์แฟรกเมนต์ครบทั้ง 3 ประเภท เมื่อนำไปทดสอบกับเครื่องมือที่พัฒนาก็ทำให้การทดสอบนั้นครอบคลุมในส่วนของรูปแบบการแปลงแผนภาพซีเควนซ์เป็นกราฟควบคุมกระแสไออาร์ซีเอฟจีได้ครบทุกกรณีแล้ว

3.4.4 การประเมินความสามารถของเครื่องมือ

ในส่วนนี้ผู้วิจัยจะประเมินกรณีทดสอบที่ถูกสร้างขึ้นจากเครื่องมือที่พัฒนาตามหลักการที่สร้างขึ้นในงานวิจัยนี้ โดยนำมาทดสอบกับส่วนของโปรแกรมตามที่ปรากฏในแผนภาพซีเควนซ์ เพื่อประเมินว่ากรณีทดสอบนั้นสามารถนำมาใช้งานได้จริง จากนั้นจะนำกรณีทดสอบที่สร้างด้วยเครื่องมือที่พัฒนาขึ้นจากหลักการของงานวิจัยนี้มาเปรียบเทียบกับกรณีทดสอบที่ถูกสร้างขึ้นโดยผู้เชี่ยวชาญ ซึ่งผู้วิจัยได้รับความอนุเคราะห์จากผู้เชี่ยวชาญทางด้านการสร้างกรณีทดสอบระดับหน่วย (Unit) และระดับบูรณาการ (Integration) โดยผู้เชี่ยวชาญมีคุณสมบัติดังต่อไปนี้

- มีประสบการณ์ทางด้านการสร้างกรณีทดสอบระดับหน่วยและระดับบูรณาการให้กับบริษัทผลิตซอฟต์แวร์ในประเทศไทยที่ได้รับมาตรฐาน CMMI ระดับ 5 โดยอยู่ในตำแหน่งนักพัฒนา (Software Engineer) เป็นระยะเวลา 5 ปี
- ปัจจุบันอยู่ในตำแหน่งผู้นำกลุ่ม (Group Leader) ของทีมพัฒนาซอฟต์แวร์ในบริษัทดังกล่าว

จากคุณสมบัติของผู้เชี่ยวชาญข้างต้น ผู้วิจัยเห็นสมควรว่ากรณีทดสอบที่ผู้เชี่ยวชาญสร้างขึ้นนั้น มีความน่าเชื่อถือ และสามารถนำกรณีทดสอบที่สร้างจากเครื่องมือมาเปรียบเทียบกับกรณีที่ผู้เชี่ยวชาญสร้างขึ้นได้

เอกสารที่ใช้ในการสร้างกรณีทดสอบจากเครื่องมือและจากผู้เชี่ยวชาญ ถูกแสดงไว้ในตารางที่ 3-7 โดยสิ่งที่นำมาเปรียบเทียบกันได้แก่จำนวนของกรณีทดสอบและจำนวนข้อมูลภายในกรณีทดสอบ

ตารางที่ 3-7 ตารางแสดงเอกสารที่ใช้ในการสร้างกรณีทดสอบจากเครื่องมือและจากผู้เชี่ยวชาญ

	กรณีทดสอบ ที่สร้างขึ้นจากเครื่องมือ	**กรณีทดสอบ ที่สร้างขึ้นจากผู้เชี่ยวชาญ**
เอกสารที่ใช้ในการสร้างกรณีทดสอบ	1. แผนภาพซีเควนซ์ 2. ข้อจำกัดของเงื่อนไขก่อนและหลังดำเนินการด้วยภาษาโอซีแอล	1. รายละเอียดยูสเคส 2. ซอร์สโค้ด

บทที่ 4
ผลการวิจัย

ในบทนี้ผู้วิจัยนำเสนอขั้นตอนพร้อมทั้งแนวคิดที่สร้างขึ้นเพื่อนำมาใช้ในการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ โดยกรณีทดสอบของงานวิจัยนี้ประกอบด้วย 3 ส่วนหลักคือ ข้อมูลเบื้องต้น (Initial Data) ข้อมูลทดสอบ (Test Data) และผลลัพท์คาดหวัง (Expected Result) ขั้นตอนให้ได้มาซึ่งกรณีทดสอบดังกล่าวประกอบด้วย การแปลงแผนภาพซีเควนซ์เป็นกราฟควบคุมกระแสไออาร์ซีเอฟจี การกำหนดเงื่อนไขก่อนและหลังดำเนินการของเมท็อดด้วยภาษาโอซีแอล การสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ การสร้างข้อมูลทดสอบ และการสร้างกรณีทดสอบ ในส่วนสุดท้ายเป็นรายละเอียดของการพัฒนาเครื่องมือสร้างกรณีทดสอบตามแนวคิดที่ได้นำเสนอ

4.1 ขั้นตอนในการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์

รูปที่ 4-1 แสดงภาพรวมของขั้นตอนและกระบวนการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์

จากรูปที่ 4-1 แสดงให้เห็นถึงกระบวนการในการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ ซึ่งสามารถแบ่งออกได้เป็น 5 กระบวนการหลัก เริ่มต้นด้วย (1) การแปลงแผนภาพซีเควนซ์ให้อยู่ในรูปแบบของกราฟควบคุมกระแสไออาร์ซีเอฟจี ซึ่งเป็นกราฟควบคุมกระแสที่ได้กล่าวถึงไว้ในงานวิจัยของ Rountev และคณะ[7] โดยในส่วนนี้ผู้วิจัยจะปรับแก้รูปแบบของไออาร์ซีเอฟจีเดิมเพื่อให้รองรับคอมบายด์แฟรกเมนต์ทั้ง 3 ประเภท และเพื่อให้สะดวกต่อการนำไปสร้างข้อมูลทดสอบของแต่ละกรณีทดสอบ เมื่อได้กราฟควบคุมกระแสไออาร์ซีเอฟจีแล้ว ขั้นตอนต่อมาคือ (2) การกำหนดรายละเอียดเงื่อนไขก่อนและหลังดำเนินการของเมท็อดด้วยโอซีแอล (OCL) ลำดับถัดไปคือ (3) การสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ ตามเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจี [7] จากนั้นจะเป็นส่วนของ (4) การสร้างข้อมูลทดสอบ ด้วยการเลือกคอนดิชันโหนดในแต่ละเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์มาใช้ในการสร้างข้อมูลทดสอบ และขั้นตอนสุดท้าย คือ (5) การสร้างกรณีทดสอบ ซึ่งเป็นการผนวกข้อมูลที่ได้ถูกสร้างไว้ในขั้นตอนต่าง ๆ มาเป็นชุดของกรณีทดสอบ กิจกรรมโดยสรุปและวัตถุประสงค์ของแต่ละกระบวนการแสดงไว้ในตารางที่ 4-1 และรายละเอียดของทั้ง 5 กระบวนการได้ถูกกล่าวไว้ในหัวข้อที่ 4.2 4.3 4.4 4.5 และ 4.6 ตามลำดับ

ตารางที่ 4-1 ตารางแสดงรายละเอียดของวัตถุประสงค์และกิจกรรมโดยสรุปของแต่ละกระบวนการในการสร้างกรณีทดสอบ

การแปลงแผนภาพซีเควนซ์เป็นกราฟควบคุมกระแสไออาร์ซีเอฟจี	
วัตถุประสงค์	เพื่อให้เห็นเส้นทางของการเรียกใช้งานเมท็อดและเงื่อนไขการเรียกใช้งานทั้งหมดที่ปรากฏในแผนภาพซีเควนซ์
กิจกรรม	• ปรับแก้กราฟควบคุมกระแสไออาร์ซีเอฟจีให้สามารถรองรับการแปลงแผนภาพซีเควนซ์ • สร้างหลักการสำหรับแปลงแผนภาพซีเควนซ์ให้อยู่ในรูปแบบของกราฟควบคุมกระแสไออาร์ซีเอฟจี
การกำหนดเงื่อนไขก่อนและหลังดำเนินการด้วยภาษาโอซีแอล	
วัตถุประสงค์	เพื่อนำเงื่อนไขก่อนและหลังดำเนินการของเมท็อดไปใช้เป็นข้อมูลเบื้องต้นและผลลัพธ์คาดหวังของกรณีทดสอบ ตามลำดับ
กิจกรรม	ระบุเงื่อนไขก่อนและหลังดำเนินการของเมท็อดที่ปรากฏในแผนภาพซีเควนซ์

การสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์	
วัตถุประสงค์	เพื่อให้ชุดกรณีทดสอบถูกสร้างขึ้นตามชุดเส้นทางไออาร์ซีเอฟจีเพื่อให้เป็นไปตามเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจี (All-IRCFG Path Criterion) ของ Rountev และคณะ [7]
กิจกรรม	สร้างชุดเส้นทางไออาร์ซีเอฟที่สมบูรณ์ตามลำดับของโหนดภายในกราฟควบคุมกระแสไออาร์ซีเอฟจีทุกเส้นทาง
การสร้างข้อมูลทดสอบ	
วัตถุประสงค์	เพื่อให้ได้ชุดของค่าตัวแปรภายในคอนดิชันโหนดทั้งหมดที่ปรากฏในแต่ละเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ เพื่อนำไปใช้เป็นข้อมูลทดสอบสำหรับกรณีทดสอบ
กิจกรรม	• เลือกคอนดิชันโหนด • สร้างค่าของตัวแปร จำแนกตามลักษณะของข้อมูล (ข้อมูลตัวเลขจำนวนเต็ม ข้อมูลชุดอักขระ และข้อมูลบูลีน)
การสร้างกรณีทดสอบ	
วัตถุประสงค์	เพื่อสร้างชุดกรณีทดสอบตามเส้นทางไออาร์ซีเอฟจีที่ปรากฏในแผนภาพซีเควนซ์
กิจกรรม	ผนวกข้อมูลที่ถูกสร้างขึ้นจากขั้นตอนก่อนหน้าทั้งหมดเข้าไว้ด้วยกัน

4.2 การแปลงแผนภาพซีเควนซ์เป็นกราฟควบคุมกระแสไออาร์ซีเอฟจี

จากแนวคิดของ Rountev และคณะ [7] ในการสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจีจากแผนภาพซีเควนซ์ที่แสดงให้เห็นถึงเส้นทางการเรียกใช้งานระหว่างเมท็อด ผู้วิจัยเห็นว่ากราฟกระแสนี้สามารถนำมาใช้ในการสร้างข้อมูลทดสอบได้ ซึ่งถือว่าเป็นส่วนประกอบที่สำคัญในกรณีทดสอบ แต่เนื่องด้วยงานวิจัยนี้รองรับแผนภาพซีเควนซ์ที่ออกแบบโดยใช้คอมบายด์แฟรกเมนต์ (Combined Fragment) 3 ประเภทคือ อัลเทอร์เนทีฟ ออปชัน และลูป ผู้วิจัยจึงจำเป็นต้องปรับแก้กราฟควบคุมกระแสไออาร์ซีเอฟจีเพื่อให้สามารถรองรับคอมบายด์แฟรกเมนต์ดังกล่าวและเพื่อให้เหมาะสมแก่การนำไปใช้ในการสร้างข้อมูลทดสอบ อีกทั้งงานวิจัยดังกล่าวไม่ได้กล่าวถึงรูปแบบหรือหลักการใด ๆ ในการสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจี เพียงแต่ให้ผู้อ่านมองเห็นภาพของการสร้างจากรูปภาพตัวอย่าง 1 รูปเท่านั้น ดังนั้นเพื่อไม่ให้เกิดความผิดพลาดในการสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจีที่อาจเกิดขึ้นเนื่องจากผู้สร้างไม่เข้าใจไปในทิศทางเดียวกัน และ

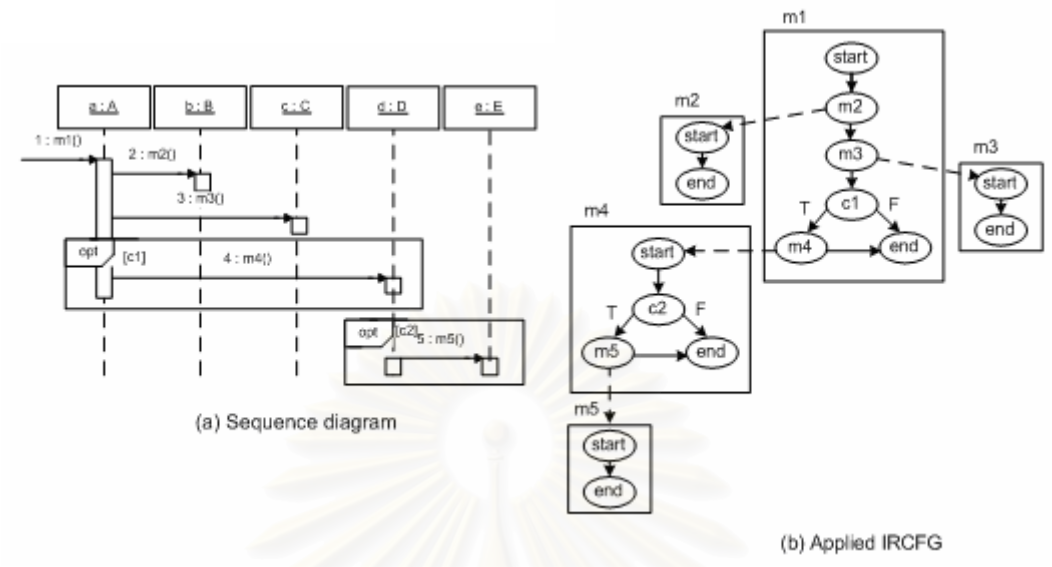
เพื่อให้สะดวกต่อการพัฒนาเครื่องมือ ผู้วิจัยจึงกำหนดหลักในการสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจีจากแผนภาพซีเควนซ์ไว้ในงานวิจัยด้วย โดยรายละเอียดต่าง ๆ มีดังต่อไปนี้

4.2.1 การเพิ่มองค์ประกอบในกราฟควบคุมกระแสไออาร์ซีเอฟจี

จากข้อจำกัดของกราฟควบคุมกระแสไออาร์ซีเอฟจีที่ได้กล่าวไว้ในเบื้องต้น ผู้วิจัยได้เพิ่มเติม 2 องค์ประกอบเข้าไปในกราฟควบคุมกระแสไออาร์ซีเอฟจี ได้แก่

1. **คอนดิชันโหนด** (Condition Node) เป็นโหนดที่แสดงข้อมูลส่วนของการตัดสินใจในการเรียกใช้งานเมท็อด ซึ่งได้มาจากข้อจำกัดของตัวถูกดำเนินการ (Operand) ที่ปรากฏอยู่ภายในคอมบายด์แฟรกเมนต์ของแผนภาพซีเควนซ์ ข้อมูลภายในคอนดิชันโหนดจะอยู่ในลักษณะคอมพาวด์เพรดิเคต (Compound Predicate) คือมีตัวดำเนินการบูลีนอยู่ในประโยคเงื่อนไข หรือเป็นซิมเพิลเพรดิเคต(Simple Predicate) คือไม่มีตัวดำเนินการบูลีนใด ๆ อยู่ในประโยคเงื่อนไขก็ได้ การที่ผู้วิจัยเพิ่มคอนดิชันโหนดไว้ในกราฟกระแสไออาร์ซีเจฟจีเพื่อที่จะนำข้อมูลเหล่านั้นมาใช้ในการสร้างข้อมูลทดสอบ (รายละเอียดของการสร้างข้อมูลทดสอบอธิบายไว้ในหัวข้อที่ 4.5) ในการสร้างคอนดิชันโหนด ผู้วิจัยได้แนวคิดมาจากเพรดิเคตโหนดซึ่งเป็นตัวแทนของการเรียกใช้งานประโยคคำสั่ง (Statement) ที่ปรากฏเป็นองค์ประกอบในกราฟควบคุมกระแสของ McCabe [3] แต่จะมีความแตกต่างกันตรงที่เพรดิเคตโหนดของ McCabe จะต้องเป็นข้อมูลในลักษณะของซิมเพิลเพรดิเคตอย่างเดียวเท่านั้น

2. **เลเบิลที** (Label T) **และเลเบิลเอฟ** (Label F) เป็นเลเบิลที่ถูกกำกับไว้บนเส้นเชื่อมที่ออกจากคอนดิชันโหนดเท่านั้น และมีข้อกำหนดว่าทุกคอนดิชันโหนดจะมีเส้นเชื่อมขาออกเป็นจำนวนสองเส้นโดยกำกับด้วยเลเบิลทีและเอฟ ตามลำดับ ส่วนนี้ถูกเพิ่มขึ้นมาเพื่อให้เห็นถึงเส้นทางการเรียกใช้เมท็อดจากคอนดิชันโหนดว่าค่าผลลัพท์ใดของเพรดิเคตภายในคอนดิชันโหนดทำให้เกิดการเรียกใช้งานเมท็อดใดบ้าง ตัวอย่างรูปแบบของกราฟควบคุมกระแสไออาร์ซีเอฟจีที่ถูกปรับแก้แล้ว แสดงไว้ดังรูปที่ 4-2



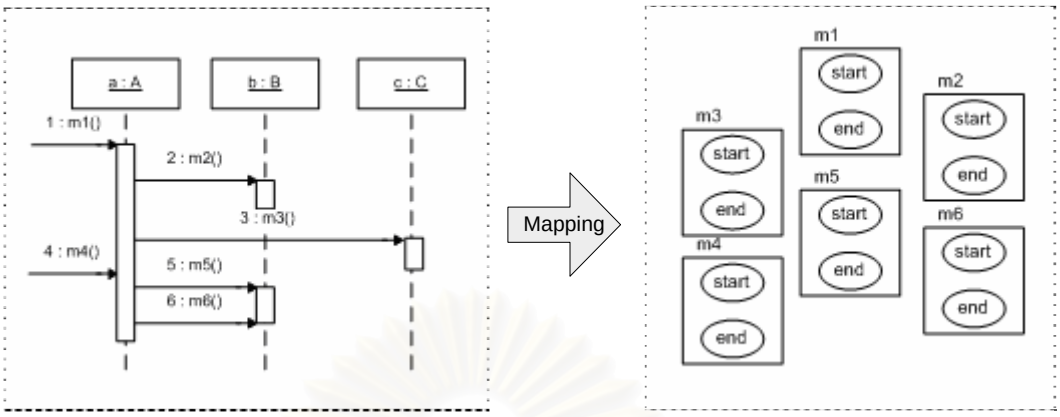
รูปที่ 4-2 ตัวอย่างของการเพิ่มคอนดิชันโหนด เลเบิลที และเลเบิลเอฟ ลงในกราฟควบคุมกระแสไออาร์ซีเอฟจี

4.2.2 **หลักการสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจีจากแผนภาพซีเควนซ์**

เมื่อปรับแก้กราฟควบคุมไออาร์ซีเอฟจีให้สามารถรองรับแบบคอมบายด์แฟรกเมนต์ 3 ประเภทได้แล้ว ในส่วนนี้ ผู้วิจัยได้สร้างหลักการและขั้นตอนสำหรับใช้ในการแปลงเพื่อให้เกิดความชัดเจนและสะดวกต่อการนำไปพัฒนาเครื่องมือ โดยหลักการแปลงประกอบไปด้วย 4 ส่วน ได้แก่ (1) การสร้างอาร์ซีเอฟจี (2) การสร้างโหนดภายในอาร์ซีเอฟจี (3) การสร้างเส้นเชื่อมอินทราโพรซิเดอรัลหรือเส้นเชื่อมอาร์ซีเอฟจี (Intraprocedural Edge-RCFG Edge) และ (4) การสร้างเส้นเชื่อมอินเตอร์โพรซิเดอรัล (Interprocedural Edge) โดยรายละเอียดในการแมปปิ้ง (Mapping) องค์ประกอบของแผนภาพซีเควนซ์มาเป็นองค์ประกอบในกราฟควบคุมกระแสไออาร์ซีเอฟจี มีดังต่อไปนี้

4.2.2.1 **การสร้างอาร์ซีเอฟจี**

อาร์ซีเอฟจีที่แสดงในกราฟกระแสเป็นตัวแทนของเมท็อดที่ปรากฏในแผนภาพซีเควซ์ ดังนั้นจำนวนของอาร์ซีเอฟจีจะเท่ากับจำนวนของเมจเสจประเภทเรียกใช้ (Message Call) ที่ปรากฏในแผนภาพซีเควนซ์ การสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจีจะเริ่มต้นด้วยการสร้างแต่ละอาร์ซีเอฟจีขึ้นมาก่อน โดยนำชื่อของเมท็อดที่ปรากฏเป็นเมจเสจของแผนภาพซีเควนซ์มากำหนดเป็นชื่อให้กับอาร์ซีเอฟจี พร้อมทั้งสร้างโหนดเริ่มต้น (Start Node) และโหนดสิ้นสุด (End Node) ไว้ภายในอาร์ซีเอฟจีนั้น ๆ ด้วย และสร้างให้ครบทุกเมจเสจ ดังแสดงในรูปที่ 4-3



(a) แผนภาพซีเควนซ์ (b) อาร์ซีเอฟจีของกระแสไออาร์ซีเอฟจี

รูปที่ 4-3 แสดงวิธีการแมปปิ้ง (Mapping) เมจเสจจากแผนภาพซีเควนซ์ ไปเป็นอาร์ซีเอฟจีของกราฟควบคุมกระแสไออาร์ซีเอฟจี

4.2.2.2 **การสร้างโหนดภายในอาร์ซีเอฟจี**

หลังจากสร้างทุกอาร์ซีเอฟจีแล้ว ลำดับถัดมาเป็นการสร้างโหนดภายในอาร์ซีเอฟจี ที่เป็นตัวแทนของเมท็อดและเงื่อนไขภายในอาร์ซีเอฟจีนั้น ๆ ซึ่งก็คือเมท็อดโหนดและคอนดิชันโหนด ตามลำดับ รายละเอียดในการสร้างโหนดทั้ง 2 ประเภทมีดังต่อไปนี้

1. **การสร้างเมท็อดโหนด (Method Node)** มีขั้นตอนดังต่อไปนี้

   1) พิจารณาไลฟ์ไลน์ L และเมจเสจ Msg_i ที่เข้ามายังไลฟ์ไลน์

   2) เลือกเมจเสจที่ออกจาก L (สมมติให้เป็น Msg_k) มาสร้างเป็นเมท็อดโหนดภายในอาร์ซีเอฟจีของเมจเสจ Msg_i โดย Msg_k จะต้องมีลำดับของเมจเสจ (Sequence Number) มากกว่ากว่าลำดับของเมจเสจ Msg_i และน้อยกว่า Msg_j ซึ่งเป็นเมจเสจที่เข้ามายัง L ถัดจาก Msg_i เงื่อนไขนี้ถูกแสดงในรูปที่ 4-4

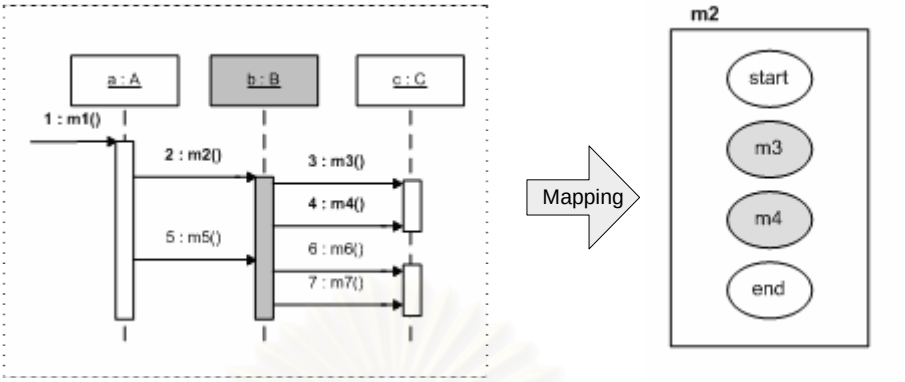
$$seqNo(Msg_j)>seqNo(Msg_k)>seqNo(Msg_i)$$

รูปที่ 4-4 แสดงเงื่อนไขในการเลือกเมจเสจที่จะนำมาเป็นเมท็อดโหนดในอาร์ซีเอฟจี Msg_i

   3) สร้างเมท็อดโหนดภายในอาร์ซีเอฟจีครบทุกเมจเสจที่เข้ามายัง L

   4) ทำซ้ำข้อที่ 1-3 จบครบทุกไลฟ์ไลน์ที่ปรากฏในแผนภาพซีเควนซ์

ตัวอย่างของการแมปปิ้งเมจเสจมาเป็นเมท็อดโหนดถูกแสดงในรูปที่ 4-5



(a) แผนภาพซีเควนซ์ (b) เมท็อดโหนดภายในอาร์ซีเอฟจี m1

รูปที่ 4-5 แสดงตัวอย่างการแมปปิ้งเมจเสจที่แสดงในแผนภาพซีเควนซ์ มาเป็นเมท็อดโหนดภายในอาร์ซีเอฟจีของกราฟควบคุมกระแสไออาร์ซีเอฟจี

2. **การสร้างคอนดิชันโหนด** (Condition Node)

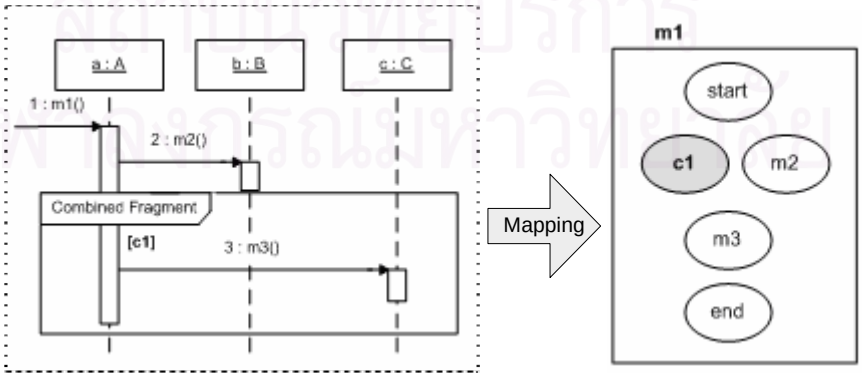
คอนดิชันโหนดภายในอาร์ซีเอฟจีเป็นตัวแทนของเงื่อนไขในการเรียกใช้เมท็อด ซึ่งก็คือข้อจำกัด (Constraint) ของแต่ละตัวถูกดำเนินการ (Operand) ภายในคอมบายด์แฟรกเมนต์ ซึ่งขั้นตอนในการสร้างคอนดิชันโหนดมีดังต่อไปนี้

1) พิจารณาตัวถูกดำเนินการ op ของคอมบายด์แฟรกเมนต์ในแผนภาพซีเควนซ์ (ยกเว้นตัวถูกดำเนินการที่มีข้อจำกัดเป็น “else”)

2) หาเมจเสจ m ที่ปรากฏเป็นเมจเสจแรกใน op

3) สร้างคอนดิชันโหนดจากข้อจำกัดของ op ไว้ในอาร์ซีเอฟจีที่มี m ปรากฏเป็นเมท็อดโหนด

4) ทำซ้ำข้อที่ 1-3 จนครบทุกตัวถูกดำเนินการที่ปรากฏในแผนภาพซีเควนซ์ ตัวอย่างของการแมปปิ้งข้อจำกัดมาเป็นคอนดิชันโหนดถูกแสดงในรูปที่ 4-6



(a) คอมบายด์แฟรกเมนต์ในแผนภาพซีเควนซ์ (b) อาร์ซีเอฟจีของเมจเสจ m1

รูปที่ 4-6 แสดงตัวอย่างการแมปปิ้งข้อจำกัดภายในแผนภาพซีเควนซ์ มาเป็นคอนดิชันโหนดภายในอาร์ซีเอฟจี

หลังจากสร้างอาร์ซีเอฟจีและโหนดภายในอาร์ซีเอฟจีครบทุกโหนดตามที่ปรากฏในแผนภาพซีเควนซ์แล้ว ลำดับถัดไปจะสร้างเส้นเชื่อมระหว่างโหนดต่าง ๆ ภายในอาร์ซีเอฟจี โดยเงื่อนไขในการสร้างเส้นเชื่อมนั้น มีดังต่อไปนี้

4.2.2.3 **การสร้างเส้นเชื่อมอินทราโพรซิเดอรัลหรือเส้นเชื่อมอาร์ซีเอฟจี** (Intraprocedural Edge-RCFG Edge)

เส้นเชื่อมอาร์ซีเอฟจี เป็นเส้นที่เชื่อมระหว่างโหนดภายในอาร์ซีเอฟจีเดียวกัน เส้นเชื่อมนี้ถูกสร้างขึ้นเพื่อให้เห็นเส้นทางการเรียกใช้งานเมท็อดภายในขอบเขตของอาร์ซีเอฟจีเดียวกัน โดยสามารถแบ่งเส้นเชื่อมออกเป็น 3 ประเภทได้แก่ (1) เส้นเชื่อมเลเบิลที (2) เส้นเชื่อมเลเบิลเอฟ และ (3) เส้นเชื่อมนอนเลเบิล โดยรูปแบบและเงื่อนไขในการสร้างเส้นเชื่อมทั้ง 3 ประเภทนั้นมีรายละเอียดดังต่อไปนี้

1. **การสร้างเส้นเชื่อมเลเบิลที** (Label T)

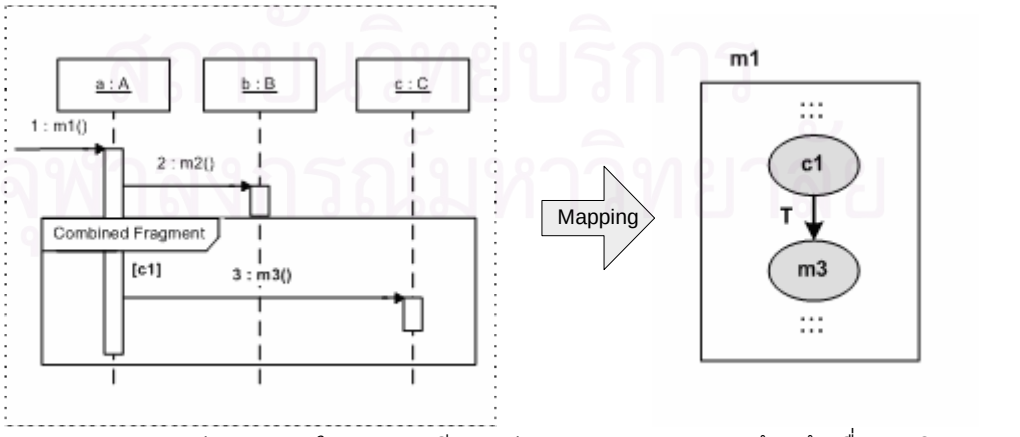
เส้นเชื่อมเลเบิลที เป็นเส้นเชื่อมที่เชื่อมจากคอนดิชันโหนดไปยังโหนดใด ๆ รูปแบบของการสร้างเส้นเชื่อมนี้แบ่งออกเป็น 2 ลักษณะคือ

a) เชื่อมระหว่างคอนดิชันโหนดกับเมท็อดโหนด

ให้ c แทนคอนดิชันโหนด และ

m แทนเมท็อดโหนด

จะสร้างเส้นเชื่อมเลเบิลทีจาก c ไปยัง m ก็ต่อเมื่อ c ถูกสร้างขึ้นจากข้อจำกัดภายในตัวถูกดำเนินการ (Operand) ของคอมบายด์แฟรกเมนต์ประเภทใด ๆ และ m ถูกสร้างจากเมจเสจที่ปรากฏเป็นเมจเสจแรกในตัวถูกดำเนินการดังกล่าว ดังรูปที่ 4-7



(a) คอมบายด์แฟรกเมนต์ในแผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมเลเบิล T จากคอนดิชันโหนดไปยังเมท็อดโหนด

รูปที่ 4-7 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลทีจากคอนดิชันโหนดไปยังเมท็อดโหนด

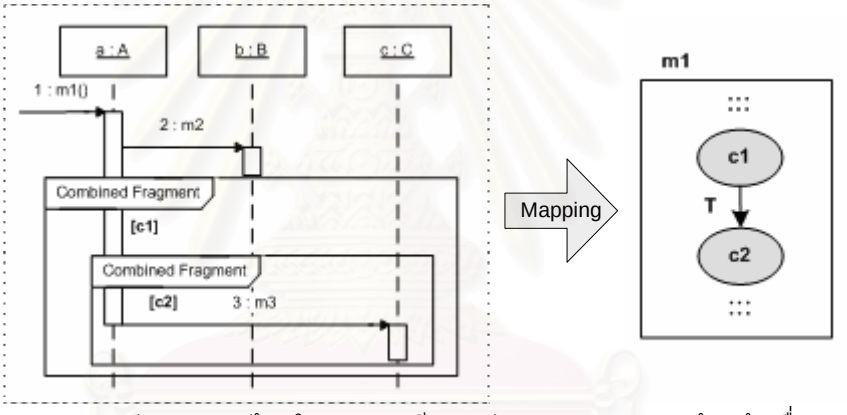
b) เชื่อมระหว่างคอนดิชันโหนดกับคอนดิชันโหนด

ให้ c_i และ c_j แทนคอนดิชันโหนด

CF_i เป็นคอมบายด์แฟรกเมนต์ประเภทใด ๆ

CF_j เป็นคอมบายด์แฟรกเมนต์ประเภทใด ๆ ที่เป็นคอมบายด์แฟรกเมนต์ลูกของ CF_i (CF_j ซ้อนอยู่ภายใน CF_i)

จะสร้างเส้นเชื่อมเลเบิลทีจาก c_i ไปยัง c_j ก็ต่อเมื่อ c_i ถูกสร้างขึ้นจากข้อจำกัดภายในตัวถูกดำเนินการของ CF_i และ c_j ถูกสร้างขึ้นจากข้อจำกัดภายในตัวถูกดำเนินการแรกของ CF_j (CF_j อาจเป็นคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟซึ่งมีได้หลายตัวถูกดำเนินการ) โดยที่ ไม่มีเมจเสจใด ๆ ปรากฏอยู่ภายในตัวถูกดำเนินการของ CF_i ก่อนหน้า CF_j เลย ดังรูปที่ 4-8



(a) ลักษณะคอมบายด์แฟรกเมนต์ซ้อนในแผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมเลเบิลทีจากคอนดิชันโหนดไปยังคอนดิชันโหนด

รูปที่ 4-8 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลทีจากคอนดิชันโหนดไปคอนดิชันโหนด

2. **การสร้างเส้นเชื่อมเลเบิลเอฟ** (Label F)

เส้นเชื่อมเลเบิลเอฟ เป็นเส้นเชื่อมที่เชื่อมจากคอนดิชันโหนดไปยังโหนดใด ๆ รูปแบบของการสร้างเส้นเชื่อมนี้แบ่งออกเป็น 2 ลักษณะคือ

a) เชื่อมระหว่างคอนดิชันโหนดกับเมท็อดโหนด

ให้ c แทนคอนดิชันโหนด และ

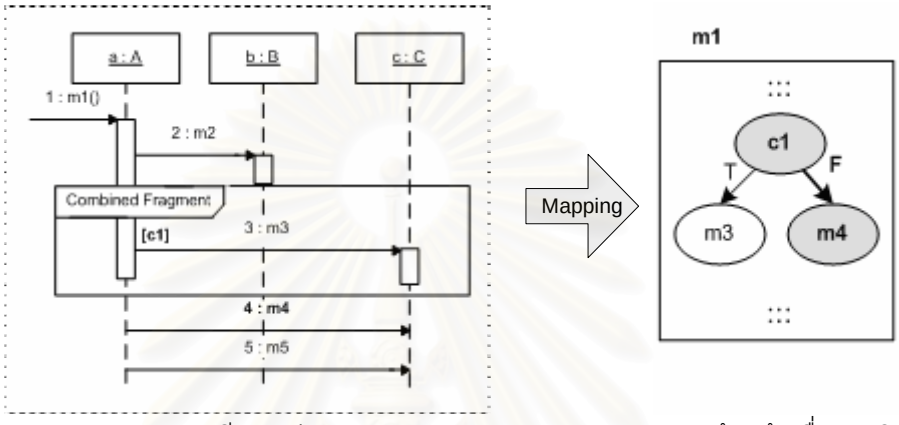
m แทนเมท็อดโหนด

CF แทนคอมบายด์แฟรกเมนต์ประเภทใด ๆ

op_i และ op_j แทนตัวถูกดำเนินการ

จะสร้างเส้นเชื่อมเลเบิลเอฟจาก c ไปยัง m ก็ต่อเมื่อ

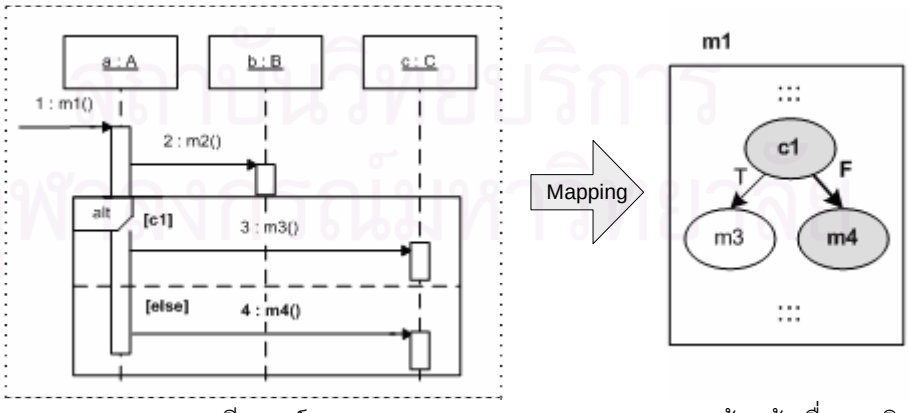
<u>กรณีที่ 1</u> c ถูกสร้างจากข้อจำกัดภายในตัวถูกดำเนินการของ CF (กรณีที่เป็นคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟ c จะต้องถูกสร้างจากข้อจำกัดที่ไม่ใช่ “else”) และ m ถูกสร้างจากเมจเสจแรกที่ถัดจาก CF ดังรูปที่ 4-9



(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังเมท็อดโหนด

รูปที่ 4-9 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังเมท็อดโหนดกรณีที่ 1

<u>กรณีที่ 2</u> c ถูกสร้างจากข้อจำกัดภายใน op_i ของ CF ประเภทอัลเทอร์เนทีฟ โดย op_j เป็นตัวถูกดำเนินการถัดจาก op_i ซึ่งมีข้อจำกัดเป็น “else” และ m ถูกสร้างจากเมจเสจแรกภายใน op_i ดังรูปที่ 4-10



(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังเมท็อดโหนด

รูปที่ 4-10 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังเมท็อดโหนดกรณีที่ 2

b) เชื่อมระหว่างคอนดิชันโหนดกับคอนดิชันโหนด

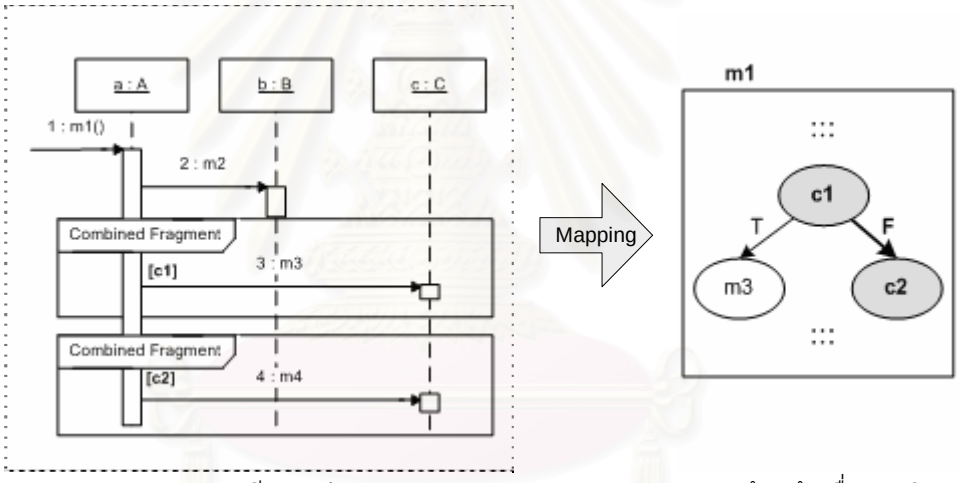
ให้ c_i และ c_j แทนคอนดิชันโหนด

CF_i แทนคอมบายด์แฟรกเมนต์

CF_j แทนคอมบายด์แฟรกเมนต์ที่อยู่ถัดจาก CF_i ซึ่งไม่ถูกกั้นระหว่างกันด้วยเมจเสจใด ๆ

จะสร้างเส้นเชื่อมเลเบิลเอฟจาก c_i ไปยัง c_j ก็ต่อเมื่อ

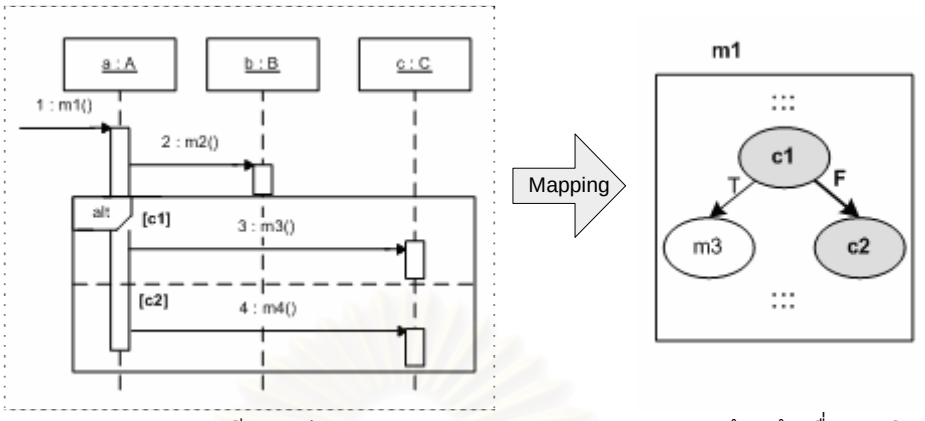
<u>กรณีที่ 1</u> c_i ถูกสร้างจากข้อจำกัดภายในตัวถูกดำเนินการของ CF_i ประเภทออบชันหรือลูป และ c_j ถูกสร้างจากข้อจำกัดภายในตัวถูกดำเนินการของ CF_j ประเภทใด ๆ (หาก CF_j เป็นคอมบายด์แฟรกเมนต์ประเภทอัลเทอร์เนทีฟ c_j จะต้องถูกสร้างขึ้นจากตัวถูกดำเนินการแรก) ดังรูปที่ 4-11



(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังคอนดิชันโหนด

รูปที่ 4-11 แสดงตัวอย่างของการแมปปิ้งเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังคอนดิชันโหนดกรณีที่ 1

<u>กรณีที่ 2</u> c_i ถูกสร้างจากข้อจำกัดภายในตัวถูกดำเนินการของ CF_i ประเภทอัลเทอร์เนทีฟ และ c_j ถูกสร้างจากข้อจำกัดภายในตัวถูกดำเนินการถัดไปของ CF_i โดยที่ข้อจำกัดของตัวถูกดำเนินการไม่ใช่ “else” ดังรูปที่ 4-12



(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังคอนดิชันโหนด

รูปที่ 4-12 แสดงตัวอย่างการแมปปิ้งเส้นเชื่อมเลเบิลเอฟจากคอนดิชันโหนดไปยังคอนดิชันโหนดกรณีที่ 2

3. การสร้างเส้นเชื่อมนอนเลเบิล

เส้นเชื่อมนอนเลเบิลคือเส้นเชื่อมที่ออกจากเมท็อดโหนดไปยังโหนดใด ๆ รูปแบบของการสร้างเส้นเชื่อมนี้แบ่งออกเป็น 2 ลักษณะคือ

a) เชื่อมระหว่างเมท็อดโหนดกับเมท็อดโหนด

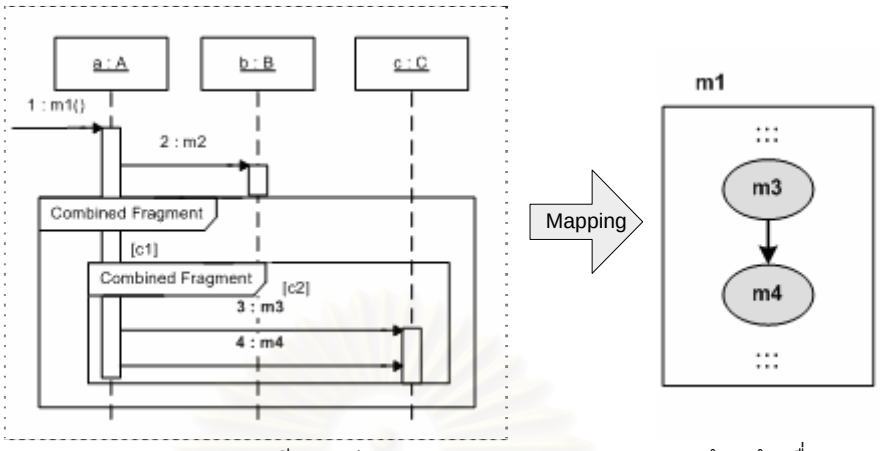
ให้ m_i แทนเมท็อดโหนดที่ถูกสร้างจากเมจเสจ M

m_j แทนเมท็อดโหนดที่ถูกสร้างจากเมจเสจที่มีลำดับถัดมาจาก M

จะสร้างเส้นเชื่อมนอนเลเบิลจาก m_i ไปยัง m_j ก็ต่อเมื่อ

<u>กรณีที่ 1</u> m_i และ m_j ถูกสร้างจากเมจเสจที่อยู่ภายใต้ตัวถูกดำเนินการเดียวกัน และไม่มีคอมบายด์แฟรกเมนต์ใดๆ คั่นระหว่าง ดังรูปที่ 4-1

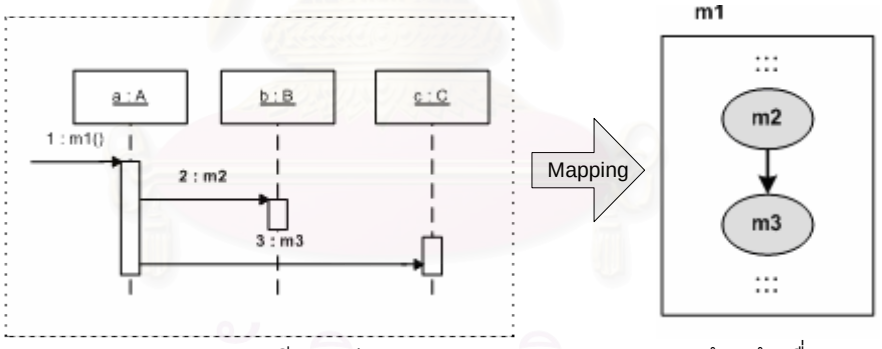
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังเมท็อดโหนด

รูปที่ 4-13 แสดงตัวอย่างของการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังเมท็อดโหนดกรณีที่ 1

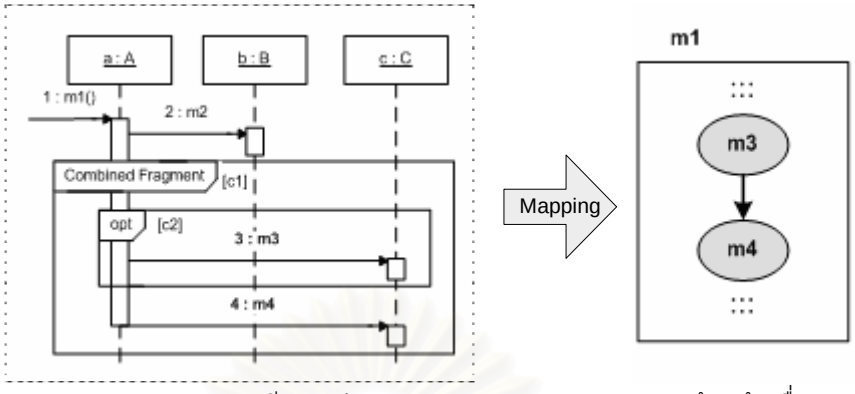
<u>กรณีที่ 2</u> m_i และ m_j ถูกสร้างจากเมจเสจที่ไม่อยู่ภายใต้ตัวดำเนินการใด ๆ และไม่มี CF ใด ๆ คั่นกลางระหว่างเมจเสจทั้งสอง ดังรูปที่ 4-14



(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังเมท็อดโหนด

รูปที่ 4-14 แสดงตัวอย่างของการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังเมท็อดโหนดกรณีที่ 2

<u>กรณีที่ 3</u> m_i ถูกสร้างจากเมจเสจลำดับสุดท้ายที่อยู่ภายใต้ตัวถูกดำเนินการของคอมบายด์แฟรกเมนต์ประเภทใด ๆ ที่ไม่ใช่ประเภทลูป และ m_j ถูกสร้างขึ้นจากเมจเสจที่ไม่ใช่เมจเสจแรกของคอมบายด์แฟรกเมนต์ประเภทใด ๆ ดังรูปที่ 4-15



(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังเมท็อดโหนด

รูปที่ 4-15 แสดงตัวอย่างของการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังเมท็อดโหนดกรณีที่ 3

b) เชื่อมระหว่างเมท็อดโหนดกับคอนดิชันโหนด

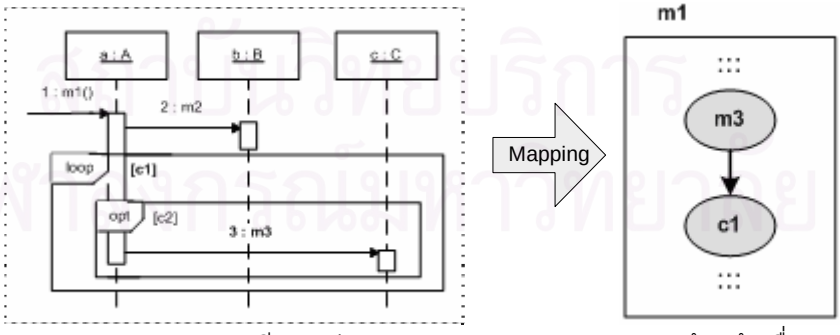
ให้ m_i แทนเมท็อดโหนดที่ถูกสร้างจากเมจเสจ M

m_j แทนเมท็อดโหนดที่ถูกสร้างจากเมจเสจที่มีลำดับถัดมาจาก M

c แทนคอนดิชันโหนด

จะสร้างเส้นเชื่อมนอนเลเบิลจาก m_i ไปยัง c ก็ต่อเมื่อ

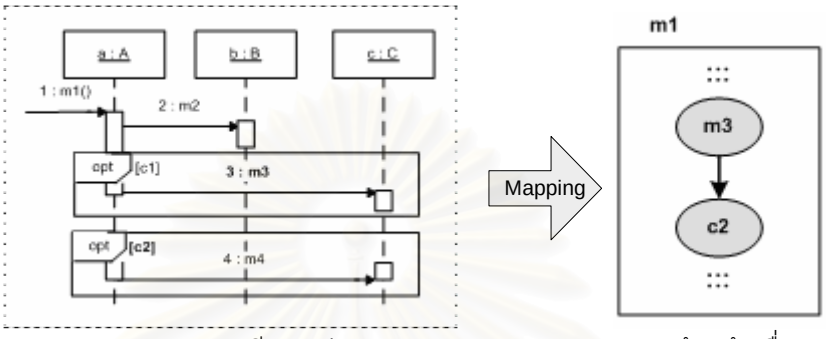
<u>กรณีที่ 1</u> m_i ถูกสร้างจากเมจเสจลำดับสุดท้ายที่อยู่ภายในตัวถูกดำเนินการของคอมบายด์แฟรกเมนต์ประเภทลูปและ c ถูกสร้างจากข้อจำกัดของคอมบายด์แฟรกเมนต์ดังกล่าว ดังรูปที่ 4-16



(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังคอนดิชันโหนด

รูปที่ 4-16 แสดงตัวอย่างของการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังคอนดิชันโหนดกรณีที่ 1

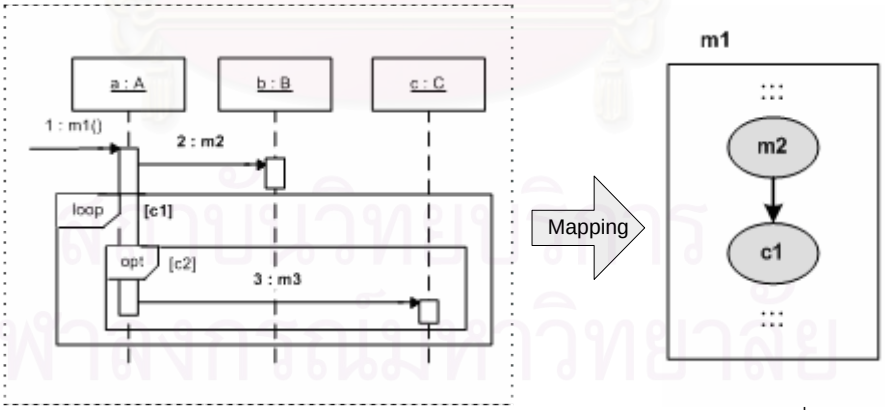
กรณีที่ 2 m_i ถูกสร้างจากเมจเสจลำดับสุดท้ายที่อยู่ภายใต้ตัวถูกดำเนินการของคอมบายด์แฟรกเมนต์ประเภทใด ๆ ที่ไม่ใช่ลูป และ c ถูกสร้างจากข้อจำกัดของคอมบายด์แฟรกเมนต์ที่อยู่ถัดไป โดยไม่มีเมจเสจใด ๆ กั้นระหว่าง ดังรูปที่ 4-17



(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังคอนดิชันโหนด

รูปที่ 4-17 แสดงตัวอย่างของการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังคอนดิชันโหนดตามกรณีที่ 2

กรณีที่ 3 m_i ถูกสร้างจากเมจเสจที่ไม่อยู่ภายใต้ตัวถูกดำเนินการของคอมบายด์แฟรกเมนต์ใด ๆ และ c ถูกสร้างจากข้อจำกัดของคอมบายด์แฟรกเมนต์ที่อยู่ถัดจากเมจเสจของ m_i ดังรูปที่ 4-18

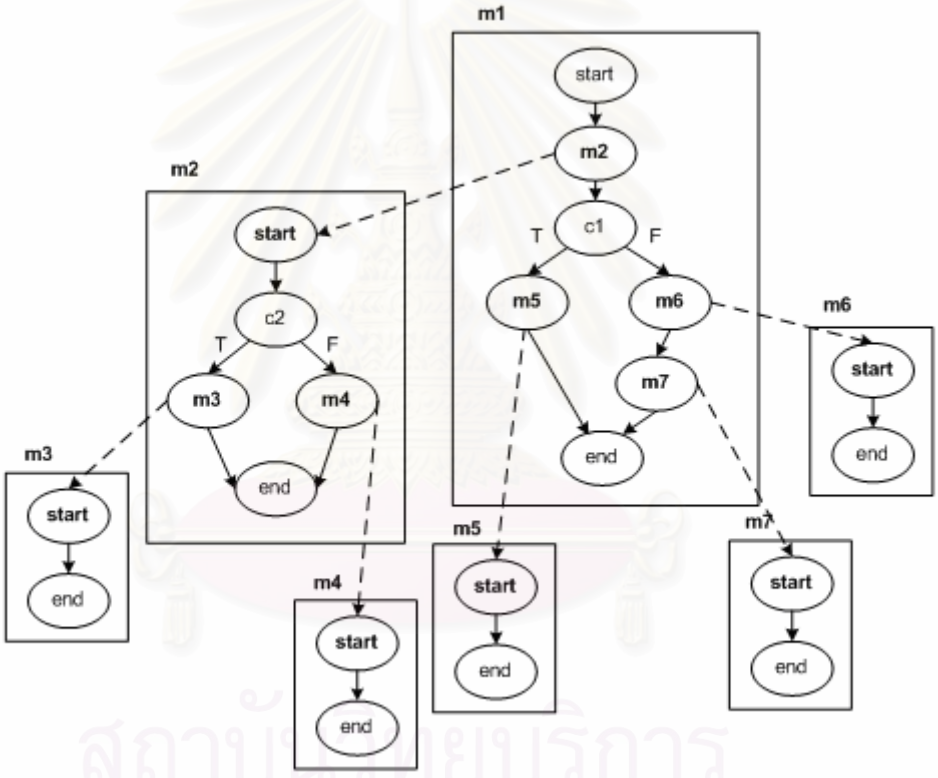


(a) แผนภาพซีเควนซ์ (b) แสดงการสร้างเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังคอนดิชันโหนด

รูปที่ 4-18 แสดงตัวอย่างของการแมปปิ้งเส้นเชื่อมนอนเลเบิลจากเมท็อดโหนดไปยังคอนดิชันโหนดตามกรณีที่ 3

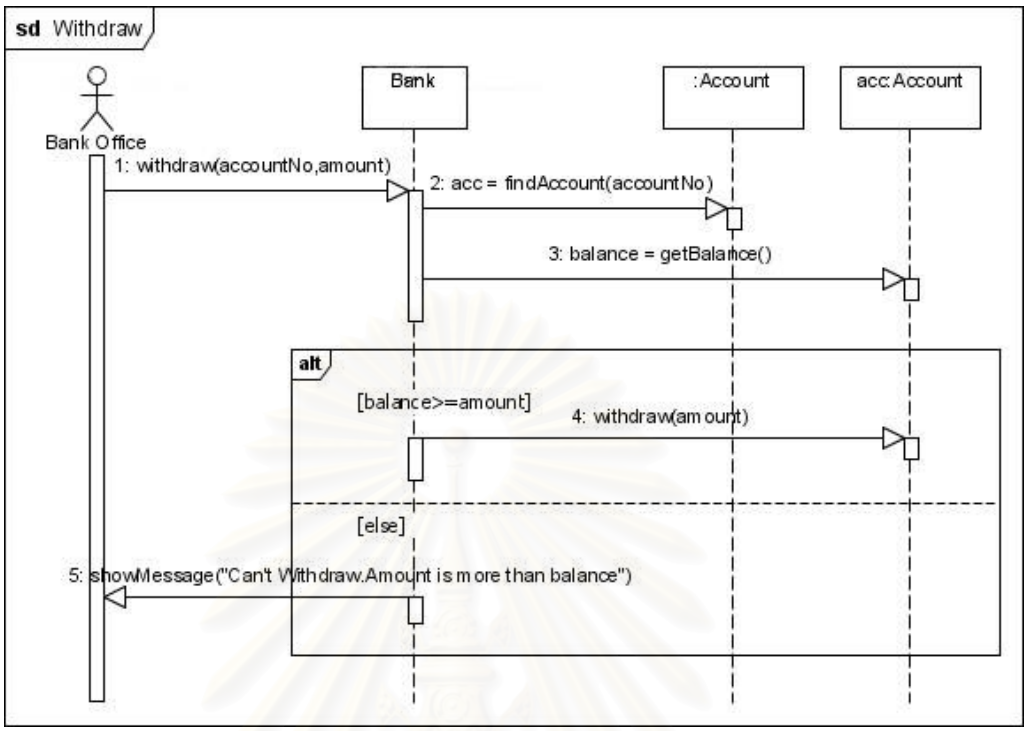
4.2.2.4 **การสร้างเส้นเชื่อมอินเตอร์โพรซิเดอรัล** (Interprocedural Edge)

หลังจากสร้างอาร์ซีเอฟจี โหนดภายในอาร์ซีเอฟจี และเส้นเชื่อมของแต่ละโหนดครบแล้ว ส่วนสุดท้ายจะเป็นการสร้างเส้นเชื่อมอินเตอร์โพรซิเดอรัล ซึ่งเป็นเส้นที่เชื่อมระหว่างอาร์ซีเอฟจี ถูกสร้างขึ้นเพื่อให้เห็นเส้นทางการเรียกใช้งานเมท็อดระหว่างอาร์ซีเอฟจี โดยสัญลักษณ์ของเส้นเชื่อมนี้ถูกแสดงด้วยเส้นประ วิธีการเชื่อมจะเชื่อมจากเมท็อดโหนด m ที่อยู่ภายในอาร์ซีเอฟจีใด ๆ ไปยังโหนดเริ่มต้นของอาร์ซีเอฟจี m ซึ่งจะทำเช่นนี้กับทุกเมท็อดโหนดที่ปรากฏอยู่ในกราฟควบคุมกระแสไออาร์ซีเอฟจี ตัวอย่างของการสร้างเส้นเชื่อมอินเตอร์โพรซิเดอรัลถูกแสดงไว้ดังรูปที่ 4-19

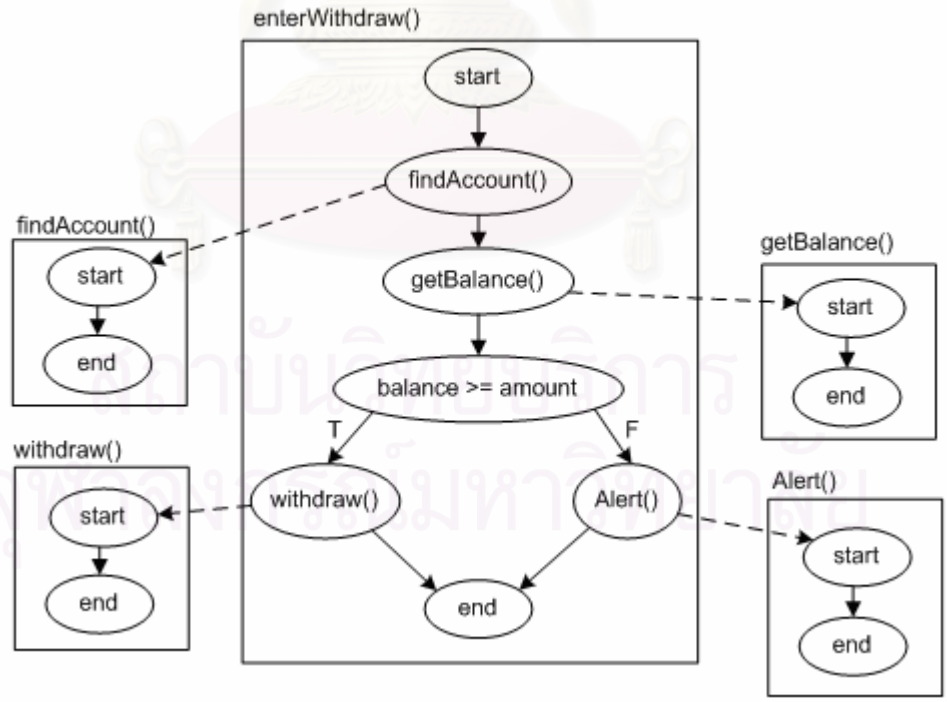


รูปที่ 4-19 แสดงตัวอย่างการสร้างเส้นเชื่อมอินเตอร์โพรซิเดอรัล

จากหลักการในการสร้างองค์ประกอบต่าง ๆ ของกราฟควบคุมกระแสไออาร์ซีเอฟจีจากแผนภาพซีเควนซ์ที่ได้กล่าวมานั้น จะสามารถสร้างกราฟกระแสอาร์ซีเอฟจีได้ดังรูปที่ 4-21 ซึ่งได้มาจากการแมปปิ้งข้อมูลจากแผนภาพซีเควนซ์การถอนเงินที่แสดงในรูปที่ 4-20



รูปที่ 4-20 แสดงตัวอย่างของแผนภาพซีเควนซ์ของยูสเคสการถอนเงิน



รูปที่ 4-21 แสดงกราฟควบคุมกระแสไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การถอนเงิน

4.3 การกำหนดเงื่อนไขก่อนและหลังดำเนินการด้วยภาษาโอซีแอล (Define OCL Pre-Post Condition)

เนื่องจากกรณีสอบที่ดีควรจะต้องประกอบไปด้วยข้อมูลอย่างน้อย 3 ส่วน [15] ได้แก่ ข้อมูลเบื้องต้น (Initial Data) ข้อมูลทดสอบ (Test Data) และผลลัพธ์คาดหวัง (Expected Result) ในส่วนของข้อมูลทดสอบสามารถสร้างขึ้นจากเงื่อนไขในการเรียกใช้งานเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ แต่ในส่วนของข้อมูลเบื้องต้นและผลลัพธ์คาดหวัง สามารถสร้างขึ้นได้เพียงเล็กน้อยจากการใช้ข้อมูลของแผนภาพซีเควนซ์ ซึ่งไม่เพียงพอต่อการนำไปใช้กับการทดสอบโปรแกรมจริง ดังนั้นเพื่อให้ข้อมูลเหล่านี้ครบถ้วน งานวิจัยนี้จึงกำหนดให้มีการระบุเงื่อนไขก่อนและหลังดำเนินการของแต่ละเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ไว้ด้วย โดยใช้ภาษามาตรฐานโอซีแอล (OCL) ในส่วนของเงื่อนไขก่อนดำเนินการ (Precondition) และเงื่อนไขหลังดำเนินการ (Postcondition) เพื่อนำไปใช้เป็นข้อมูลเบื้องต้น และผลลัพธ์คาดหวังของกรณีทดสอบตามลำดับ ตัวอย่างการของการระบุเงื่อนไขก่อนและหลังดำเนินการถูกแสดงไว้ในตารางที่ 4-2 (ข้อมูลเกี่ยวกับโอซีแอลและข้อกำหนดของโอซีแอลสามารถดูได้จากเว็บไซต์ http://www.omg.org)

ตารางที่ 4-2 แสดงตัวอย่างของการกำหนดเงื่อนไขก่อนและหลังดำเนินการของเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ยูสเคสการถอนเงิน

ชื่อเมท็อด	OCL : Precondition	OCL : Postcondition
findAccount	• Account->include(account.no =accountNo)	• result = acc
getBalance		• result = balance
withdraw	• acc.getBalance() >= amount	• acc.getBalance() = acc.getBalance()@pre-amount
showMessage	-	• result = “Can’t Withdraw. Amount is more than balance”

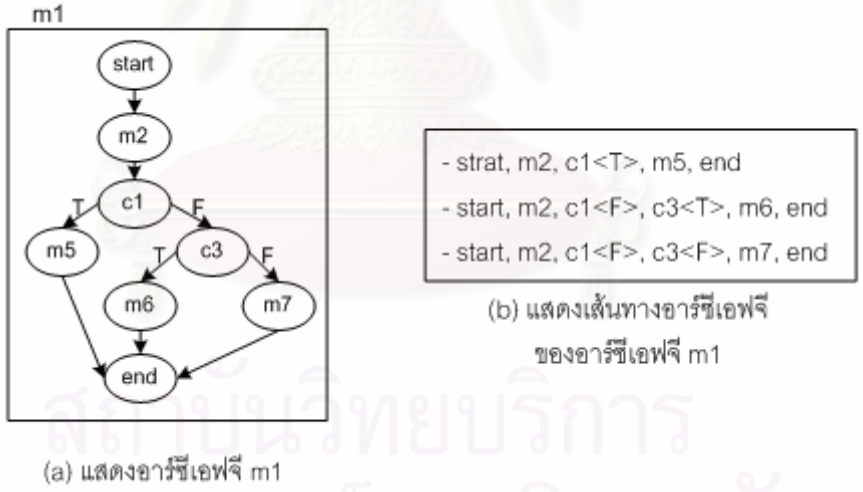
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.4 การสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์

เส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ (Complete IRCFG Path) คือเส้นทางที่เริ่มต้นด้วยโหนดเริ่มต้น (Start Node) ของอาร์ซีเอฟจีที่เป็นราก (Root RCFG) และท่องผ่านแต่ละโหนดเริ่มต้นไปยังโหนดสิ้นสุดของอาร์ซีเอฟจีที่ปรากฏเป็นเมท็อดโหนดอยู่ภายในอาร์ซีเอฟจีที่กำลังท่องผ่าน โดยที่โหนดสุดท้ายของเส้นทางจะจบลงที่โหนดสิ้นสุด (End Node) ของอาร์ซีเอฟจีที่เป็นราก (รายละเอียดและคุณสมบัติของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์กล่าวไว้ในบทที่ 2)

ในงานวิจัยนี้จะสร้างทุกเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ที่ปรากฏอยู่ในกราฟควบคุมกระแสไออาร์ซีเอฟจี ซึ่งตรงกับเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจี (All-IRCFG-Path Criterion) ที่ Rountev และคณะ [7] ได้ระบุไว้ เพื่อสร้างชุดกรณีทดสอบให้ครอบคลุมตามเกณฑ์ดังกล่าว ขั้นตอนและวิธี ที่ผู้วิจัยใช้ในการสร้างทุกเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์มีดังต่อไปนี้

1. สร้างชุดเส้นทางอาร์ซีเอฟจี (RCFG Path) สำหรับทุกอาร์ซีเอฟจีที่ปรากฏในกราฟควบคุมกระแสไออาร์ซีเอฟจี ซึ่งเส้นทางอาร์ซีเอฟจีเป็นเส้นทางย่อย (Sub Path) ของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ ตัวอย่างอาร์ซีเอฟจีและชุดของเส้นทางอาร์ซีเอฟจีแสดงไว้ในรูปที่ 4-22

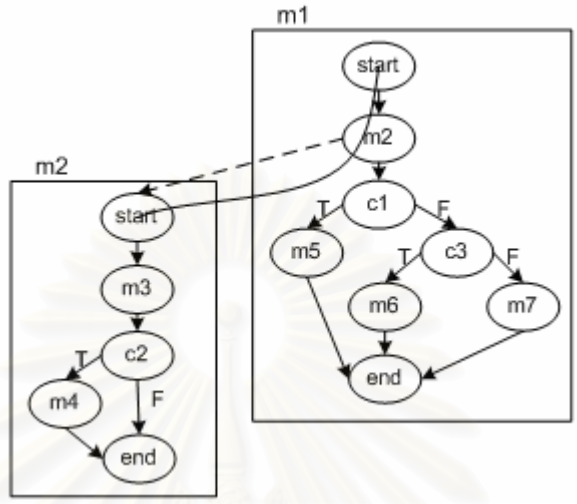


รูปที่ 4-22 แสดงอาร์ซีเอฟจีของ m1 (a) พร้อมทั้งชุดเส้นทางอาร์ซีเอฟจีของ m1 (b)

2. เลือกเส้นทางไออาร์ซีเอฟจีของอาร์ซีเอฟจีที่เป็นราก (Root RCFG) ขึ้นมา 1 เส้นทาง สมมติให้เป็น $p(RCFG_{root})$ โดยที่ p แทนเส้นทางอาร์ซีเอฟจี สมมติว่า p ที่เลือกจากรูปที่ 4-22 คือ { start, m2, c1<T>, m5, end }

3. ท่องไปยังโหนดในเส้นทางอาร์ซีเอฟจี โดยเริ่มท่องจากโหนดเริ่มต้นเป็นโหนดแรก

4. เมื่อท่องพบเมท็อดโหนด m จะท่องไปยัง $RCFG_m$ ซึ่งเชื่อมต่อกับ m ด้วยเส้นเชื่อมอินเตอร์โพรซิเดอรัล ดังรูปที่ 4-23



รูปที่ 4-23 แสดงลักษณะการท่องไปยัง $RCFG_{m2}$ ที่เชื่อมต่อกับเมท็อดโหนด m2

5. เลือกเส้นทางไอออาร์ซีเอฟจีของ $RCFG_m$ มา 1 เส้นทาง สมมติให้เป็น $p(RCFG_m)_i$ จากนั้นแทรก $p(RCFG_m)_i$ ไว้หลังเมท็อดโหนด m ดังแสดงในตัวอย่างข้างล่าง

จาก $p(RCFG_{m1})_1$ = { $start_{m1}$, m2, c1<T>, m5, end_{m1} }

และ $p(RCFG_{m2})_1$ = { $start_{m2}$, m3, c2<T>, m4, end_{m2} }

เมื่อนำ $p(RCFG_{m2})_1$ แทรกใน $p(RCFG_{m1})_1$ แล้วจะได้เป็น

{ $start_{m1}$, m2, *$start_{m2}$*, *m3*, *c2<T>*, *m4*, *end_{m2}*, c1<T>, m5, end_{m1} }

6. ทำซ้ำข้อที่ 5 จนครบทุกเส้นทางอารซีเอฟจีของ $RCFG_m$ จะได้ชุดของเส้นทางย่อยทั้งหมดดังแสดงในรูปที่ 4-24

{ $start_{m1}$, m2, *$start_{m2}$*, *m3*, *c2<T>*, *m4*, *end_{m2}*, c1<T>, m5, end_{m1} }

{ $start_{m1}$, m2, *$start_{m2}$*, *m3*, *c2<F>*, *end_{m2}*, c1<T>, m5, end_{m1} }

รูปที่ 4-24 แสดงชุดเส้นทางย่อยเมื่อท่องผ่านเมท็อดโหนด m2

7. นำแต่ละเส้นทางที่ได้จากข้อ 6 มาท่องไปยังโหนดโดยเริ่มท่องจากโหนดที่ถัดจาก m และทำซ้ำข้อที่ 4-6 ไปเรื่อย ๆ จนกระทั่งท่องครบทุกเมท็อดโหนดในเส้นทาง

8. ทำซ้ำข้อที่ 2 จนครบทุกเส้นทางอาร์ซีเอฟจีของ $RCFG_{root}$ จะได้ชุดของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ตามเกณฑ์ทุกเส้นทางไออาร์ซีเอฟจี (All-IRCFG Path Criterion) ตัวอย่างของชุดเส้นทางไออาร์ซีเอฟจีจากกราฟควบคุมกระแสไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การถอนเงินรูปที่ 4-21 แสดงไว้ในตารางที่ 4-3

จำนวนของเส้นทางย่อยของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ในแต่ละอาร์ซีเอฟจีสามารถคำนวณหาได้จากสูตร

$$\sum_{j=1}^{k} \sum_{i=1}^{n} p(m_i) \times p(m_{i+1})$$

โดยที่ i แทนลำดับของเมท็อดโหนดในเส้นทางอาร์ซีเอฟจี

j แทนลำดับของเส้นทางอาร์ซีเอฟจี

m แทนเมท็อดโหนดในเส้นทางอาร์ซีเอฟจี

p(m) แทนจำนวนเส้นทางย่อยทั้งหมดของอาร์ซีเอฟจีที่เชื่อมต่อกับเมท็อดโหนดดังกล่าวด้วยเส้นเชื่อมอินเตอร์โพรซิเดอรัล

ในการหาจำนวนเส้นทางย่อย จะเริ่มต้นหาจากอาร์ซีเอฟจีที่อยู่ลึกที่สุดในกราฟควบคุมกระแสก่อน จากนั้นจึงค่อย ๆ เลื่อนขึ้นมาเพื่อหาจำนวนเส้นทางย่อยของอาร์ซีเอฟจีที่อยู่ระดับที่สูงกว่า เมื่อทำเช่นนี้ไปเรื่อย ๆ จนกระทั่งถึงอาร์ซีเอฟจีที่เป็นราก จำนวนเส้นทางจะเท่ากับจำนวนเส้นทางอาร์ซีเอฟจีที่สมบูรณ์ทุกเส้นทาง

ตารางที่ 4-3 แสดงชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ภายในกราฟควบคุมกระแสไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การถอนเงิน

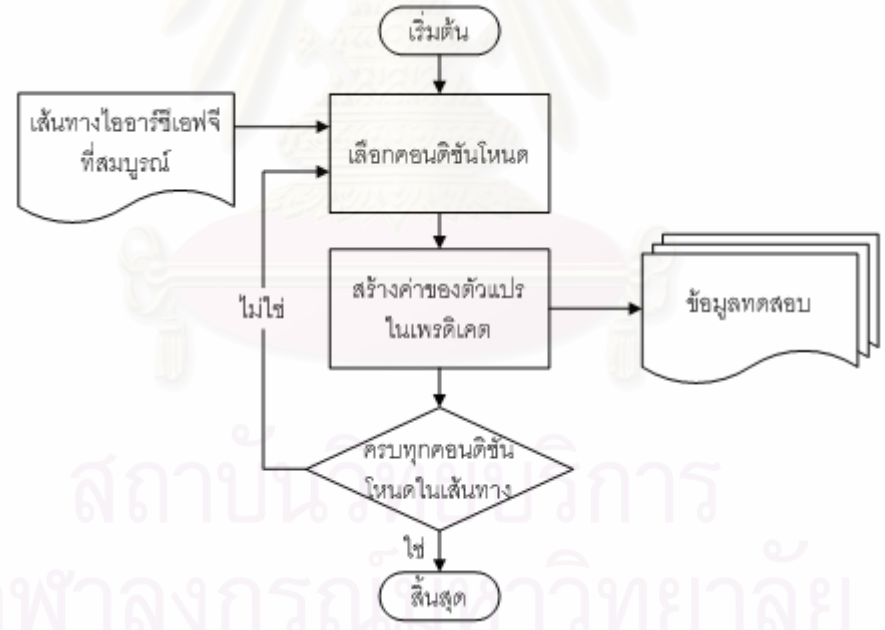
ลำดับที่	ไออาร์ซีเอฟจีที่สมบูรณ์
1	$start_{enterWithdraw}$, findAccount, $start_{findAccount}$, $end_{findAccount}$, getBalance, $start_{getBalance}$, $end_{getBalance}$, balance>=amount<T>, withdraw, $start_{withdraw}$, $end_{withdraw}$, $end_{enterWithdraw()}$
2	$start_{enterWithdraw}$, findAccount, $start_{findAccount}$, $end_{findAccount}$, getBalance, $start_{getBalance}$, $end_{getBalance}$, balance>=amount<F>, showMessage, $start_{Alert}$, end_{Alert}, $end_{enterWithdraw()}$

4.5 การสร้างข้อมูลทดสอบ

ข้อมูลทดสอบคือค่าของตัวแปรที่จะนำไปเป็นข้อมูลนำเข้าเพื่อใช้ในการทดสอบโมดูลหรือโปรแกรมที่ต้องการทดสอบ ขั้นตอนของการสร้างข้อมูลทดสอบในงานวิจัยนี้ มีดังต่อไปนี้

1. เลือกเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ขึ้นมาหนึ่งเส้นทาง สมมติให้เป็น P
2. ท่องไปยังโหนดของ P แบบโพสออร์เดอร์ (Post-Order)
3. เมื่อพบคอนดิชันโหนด นำข้อมูลภายในคอนดิชันโหนดมาสร้างค่าตัวแปรเป็นข้อมูลทดสอบ
4. ทำซ้ำข้อที่ 2-3 จนครบทุกคอนดิชันโหนดใน P
5. ทำซ้ำข้อที่ 1-4 จนครบทุกเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์

ผลลัพธ์ที่ได้คือชุดข้อมูลทดสอบของแต่ละเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ ภาพรวมของขั้นตอนการสร้างข้อมูลทดสอบสำหรับหนึ่งเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์แสดงไว้ดังรูปที่ 4-25



รูปที่ 4-25 แสดงขั้นตอนในการสร้างข้อมูลทดสอบของแต่ละเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์

4.5.1 การเลือกคอนดิชันโหนด

หลังจากสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ได้ครบทุกเส้นทางแล้ว จะนำแต่ละเส้นทางไออาร์ซีเอฟจีดังกล่าวมาเลือกข้อมูลจากคอนดิชันโหนดโดยท่องไปยังแต่ละโหนดแบบ

โพสออร์เดอร์ ตัวอย่างของเส้นทางไอออาร์ซีเอฟจีและลำดับในการเลือกถูกแสดงไว้ในรูปที่ 4-26 และตารางที่ 4-4 ตามลำดับ

$start_{enterOrder}$, rate, $start_{rate}$, end_{rate}, getx, $start_{getx}$, end_{getx}, (x-r)>=0<T>, compute, $start_{compute}$, $end_{compute}$, (r>10)and(a<=200)<T>, printgift, $start_{printgift}$, $end_{printgift}$, $end_{enterOrder}$

รูปที่ 4-26 แสดงตัวอย่าง 1 เส้นทางไอออาร์ซีเอฟจีที่สมบูรณ์

ตารางที่ 4-4 แสดงลำดับของการเลือกคอนดิชันโหนดจากเส้นทางไอออาร์ซีเอฟจีที่สมบูรณ์ในรูปที่ 4-13

ลำดับการเลือกคอนดิชันโหนด	ข้อมูลของคอนดิชันโหนด	ซิมเพิลเพรดิเคต	ผลลัพธ์ของเพรดิเคต
1	(r>10)and(a<=200)	(r>10)	T
		(a<=200)	T
2	(x-r)>=0	(x-r)>=0	T

ขั้นตอนของการนำคอนดิชันโหนดไปใช้ในการสร้างข้อมูลทดสอบมีรายละเอียดดังต่อไปนี้

1. เลือกคอนดิชันโหนด C
2. หากข้อมูลใน C อยู่ในรูปแบบของคอมพาวน์เพรดิเคต จะต้องทำให้อยู่ในรูปของซิมเพิลเพรดิเคต
3. นำข้อมูลของแต่ละเพรดิเคตไปสร้างข้อมูลทดสอบ
   a) กรณีที่พบว่ามีบางตัวแปรในเพรดิเคตถูกสร้างค่าจากเพรดิเคตก่อนหน้า ให้แทนที่ค่าของตัวแปรที่ซ้ำและคำนวณหาค่าของตัวแปรที่เหลือ
   b) กรณีที่พบว่าทุกตัวแปรในเพรดิเคตถูกสร้างค่าจากเพรดิเคตก่อนหน้า ให้แทนที่ค่าของตัวแปรที่ซ้ำและตรวจสอบความสอดคล้องของตัวแปรกับผลลัพธ์ของเพรดิเคต
   - หากสอดคล้องกัน ใช้ค่าของตัวแปรดังกล่าวมาเป็นข้อมูลทดสอบ
   - หากไม่สอดคล้องกัน จะคำนวณค่าของตัวแปรดังกล่าวใหม่ให้สอดคล้องกับผลลัพธ์ของเพรดิเคต และใช้ค่าของตัวแปรที่คำนวณค่าใหม่มาเป็นข้อมูลทดสอบ
4. ทำซ้ำข้อที่ 1-3 จนครบทุกคอนดิชันโหนดในเส้นทางไอออาร์ซีเอฟจีที่สมบูรณ์

จากข้อมูลของคอนดิชันโหนดที่เลือก หากข้อมูลนั้นอยู่ในลักษณะของคอมพาวด์เพรดิ- เคตคือมีตัวดำเนินการบูลีนเป็นตัวเชื่อม จะตัดตัวดำเนินการดังกล่าวออกและแยกคอมพาวด์เพร- ดิเคตให้อยู่ในรูปแบบของซิมเพิลเพรดิเคต ดังที่แสดงไว้ในตารางที่ 4-4 หลังจากนั้นจึงนำซิมเพิลเพรดิเคตไปสร้างข้อมูลทดสอบตามลำดับการเลือก การที่ผู้วิจัยเลือกคอนดิชันโหนดแบบโพสออร์-เดอร์นั้นเพื่อลดการสร้างข้อมูลทดสอบจากตัวแปรที่ซ้ำกับโหนดก่อนหน้า ตัวอย่างเช่น จากตารางที่ 4-4 จะเห็นได้ว่ามีเพรดิเคตที่มีตัวแปรซ้ำกัน คือ เพรดิเคต (r>10) และเพรดิเคต (x-r)>=0 ในที่นี้ (r>10) ถูกเลือกมาสร้างข้อมูลทดสอบก่อน สมมติค่าของตัวแปรที่สร้างได้คือ r =11 จากนั้นเมื่อเลือก (x-r)>=0 มาสร้างค่าของข้อมูลทดสอบ แล้วพบว่า r ได้ถูกสร้างค่ามาแล้วจากโหนดก่อนหน้า ดังนั้นในการหาค่าตัวแปร จะให้ r ของเพรดิเคตนี้มีค่าเท่ากับ r ที่ถูกสร้างขึ้นจากโหนดก่อนหน้า คือแทนที่ตัวแปร r ของเพรดิเคตดังกล่าวด้วย 11 จะได้เป็น (x-11)>=0 ค่าของ x ที่สอดคล้องกับผลลัพธ์เพรดิเคต (มีค่าเป็นจริง) ที่ระบุไว้ในเส้นทางไออาร์ซีเอฟจี คือ x=11 (วิธีการให้ได้มาซึ่งค่าของตัวแปรถูกอธิบายไว้ในหัวข้อที่ 4.5.2) ดังนั้นชุดของข้อมูลทดสอบของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์นี้คือ {r=11, x=11, a=200} เป็นต้น

การแทนที่ค่าของตัวแปรดังตัวอย่างที่กล่าวมานั้นจะเกิดขึ้นเมื่อจำนวนตัวแปรที่อยู่ในเพรดิเคตดังกล่าวมีจำนวนมากกว่าจำนวนตัวแปรที่ซ้ำกับโหนดก่อนหน้า

```
start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType="A"<T>,
        totalQty >100<F>,  totalQty >10<F>, showNetPrice(), start, end, end
```

รูปที่ 4-27 ตัวอย่างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์

หากจำนวนตัวแปรที่ซ้ำมีจำนวนเท่ากัน จะไม่สามารถนำค่าของตัวแปรที่สร้างไว้แล้วมาแทนที่ได้ทันที แต่จะต้องตรวจสอบความสอดคล้องของตัวแปรกับผลลัพธ์ของเพรดิเคตก่อน ยกตัวอย่างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ในรูปที่ 4-27 totalQty>10 ถูกเลือกมาสร้างข้อมูลทดสอบที่ทำให้ผลลัพธ์ของเพรดิเคตเป็นเท็จ ดังนั้นค่าตัวแปรของข้อมูลทดสอบเพรดิเคตนี้คือ {totalQty =10} หลังจากนั้น totalQty>100 ถูกเลือกมาสร้างข้อมูลทดสอบเป็นลำดับถัดไป แต่เนื่องจากพบว่าตัวแปรภายในเพรดิเคตเคยถูกสร้างมาแล้วจากโหนดก่อนหน้า จึงนำค่าตัวแปรดังกล่าวมาแทนที่ แล้วตรวจสอบว่าค่าของตัวแปรที่นำมาแทนที่นั้นทำให้ผลลัพธ์ของเพรดิเคตดังกล่าวเป็นเท็จตามที่ได้ระบุไว้ในเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์หรือไม่ โดยเมื่อแทนที่แล้วจะได้เป็น 10>100 พบว่าผลลัพธ์ที่ได้เป็นเท็จซึ่งสอดคล้องกับผลลัพธ์ของเพรดิเคตที่ระบุไว้ ดังนั้น {totalQty =10} จึงเป็นข้อมูลทดสอบย่อยในชุดข้อมูลทดสอบของเส้นทางไออาร์ซีเอฟจีนี้

start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType="A"<T>,
totalQty >10<F>, totalQty >100<F>, showNetPrice(), start, end, end

รูปที่ 4-28 ตัวอย่างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์

ในทางกลับกันหากเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์เป็นดังรูปที่ 4-28 totalQty>10 จะถูกเลือกมาสร้างข้อมูลทดสอบก่อน จะได้ {totalQty=100} หลังจากนั้น totalQty>10 ถูกเลือกมาสร้างข้อมูลทดสอบเป็นลำดับถัดไป เมื่อแทนที่ค่าของตัวแปรแล้วจะได้ 100>10 พบว่าผลลัพธ์ออกมาเป็นจริง ซึ่งไม่สอดคล้องกับผลลัพธ์ของเพรดิเคตที่ถูกระบุไว้ในเส้นทางไออารซีเอฟจีคือ ผลลัพธ์ของเพรดิเคต totalQty>10 จะต้องเป็นเท็จ ดังนั้นจึงต้องสร้างค่าตัวแปรของ totalQty ใหม่ให้สอดคล้องกับผลลัพธ์ของเพรดิเคตที่ระบุไว้ในเส้นทาง ดังนั้นค่าที่ได้คือ 10 ข้อมูลทดสอบย่อยของเส้นทางนี้จึงมี {totalQty=10}

4.5.2 การสร้างค่าของตัวแปรในเพรดิเคต

งานวิจัยนี้รองรับการสร้างข้อมูลทดสอบ 3 ประเภทเท่านั้นได้แก่ข้อมูลประเภทตัวเลขจำนวนเต็ม (Integer) ข้อมูลประเภทสตริง (String) และข้อมูลประเภทบูลีน (Boolean) ซึ่งวิธีการสร้างค่าของข้อมูลแต่ละประเภทนั้นมีรายละเอียดดังต่อไปนี้

1. ข้อมูลประเภทตัวเลขจำนวนเต็ม

ในการสร้างข้อมูลทดสอบของแต่ละซิมเพิลเพรดิเคตที่อยู่ในรูปแบบของข้อมูลประเภทตัวเลขจำนวนเต็ม ผู้วิจัยจะประยุกต์ใช้เกณฑ์การทดสอบขอบเขต (Boundary Testing Criterion) เพื่อนำมาหาค่าของตัวแปรที่อยู่ภายในซิมเพิลเพรดิเคต โดยข้อมูลภายในซิมเพิลเพรดิเคตที่เป็นประเภทตัวเลขจำนวนเต็มนั้นจะต้องอยู่ในรูปแบบของ นิพจน์ความสัมพันธ์ (Relational Expression) ดังแสดงในรูปที่ 4-29

$E_1 op E_2$

รูปที่ 4-29 แสดงรูปแบบของนิพจน์ความสัมพันธ์

โดย E_1 และ E_2 คือนิพจน์ตัวเลข (Arithmetic Expression) และ op คือเป็นตัวหนึ่งตัวใดในตัวดำเนินการ (Operator) ของกลุ่ม { <, <=, >, >=, ==, != } ในการหาค่าของตัวแปร แต่ละเพรดิเคต $E_1 op E_2$ จะถูกแปลงให้อยู่ในรูปแบบของบรานช์ฟังก์ชัน (Branch Function) [33] ก่อนดังรูปที่4-30 เพื่อนำไปใช้ในการสร้างค่าของตัวแปรที่ตรงกับผลลัพธ์ของเพรดิเคตที่ถูกกำหนดใน

เส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ โดยบรานช์ฟังก์ชันเป็นแนวคิดของ Bogdan [33] ที่สร้างความสัมพันธ์ระหว่างรูปแบบของนิพจน์ความสัมพันธ์กับผลลัพธ์ของเพรดิเคต

$$F = (E_2 - E_1) \text{ หรือ } F = (E_1 - E_2)$$

รูปที่ 4-30 แสดงรูปแบบของฟังก์ชันเพื่อหาค่าของตัวแปร

รูปแบบของฟังก์ชันจะขึ้นอยู่ประเภทของตัวดำเนินการ op ที่ปรากฏอยู่ในนิพจน์ความสัมพันธ์ ซึ่งแสดงไว้ในตารางที่ 4-5

ตารางที่ 4-5 แสดงความสัมพันธ์ของนิพจน์พร้อมทั้งผลลัพท์ [33]

Branch Predicate	Branch Function	Function Value Result			Predicate Value Result
		Positive	*Zero*	*Negative*	
$E_1 > E_2$	$E_2 - E_1$			✓	T
		✓	✓		F
$E_1 >= E_2$	$E_2 - E_1$		✓	✓	T
		✓			F
$E_1 < E_2$	$E_1 - E_2$			✓	T
		✓	✓		F
$E_1 <= E_2$	$E_1 - E_2$		✓	✓	T
		✓			F
$E_1 == E_2$	$abs(E_1 - E_2)$		✓		T
		✓			F
$E_1 != E_2$	$abs(E_1 - E_2)$	✓			T
			✓		F

ตารางที่ 4-5 แสดงให้เห็นถึงค่าผลลัพธ์ของบรานช์ฟังก์ชันที่มีผลต่อค่าผลลัพธ์ของเพรดิเคต ยกตัวอย่างเช่น (r>10) คือนิพจน์ภายในซิมเพิลเพรดิเคตที่ถูกเลือก และนิพจน์อยู่ในรูปแบบของ ($E_1 > E_2$) ดังนั้นบรานช์ฟังก์ชันของซิมเพิลเพรดิเคตนี้คือ 10 - r (F = 10 - r) ค่าใด ๆ ของ r ที่ทำให้ F มีค่าเป็นศูนย์ (Zero) หรือบวก (Positive) หมายความว่า ผลลัพธ์ของเพรดิเคต (r>10) มี

ค่าเป็นเท็จ (False) และค่าใด ๆ ของ r ที่ทำให้ F มีค่าเป็นลบ (Negative) หมายความว่า ผลลัพท์ของเพรดิเคต (r>10) มีค่าเป็นจริง (True)

แนวคิดที่ผู้วิจัยสร้างขึ้นเพื่อให้ได้มาซึ่งค่าของตัวแปรที่อยู่ในนิพจน์ความสัมพันธ์ภายในเพรดิเคตมีขั้นตอนดังต่อไปนี้

1. กำหนดรูปแบบของบรานช์ฟังก์ชันตามประเภทของตัวดำเนินการที่ปรากฏในเพรดิเคต โดยตรวจสอบจากตารางที่ 4-5

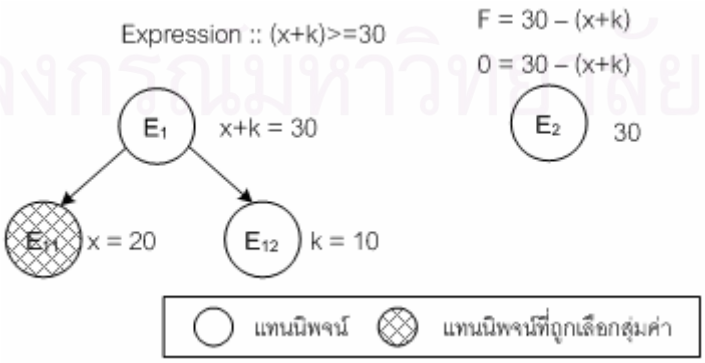
2. ตรวจสอบผลลัพธ์ของเพรดิเคตที่นำมาสร้างข้อมูลทดสอบ ซึ่งได้ถูกระบุไว้ในเส้นทางไออาร์ซีเอฟจีและนำมาเทียบกับตารางที่ 4-5 เพื่อดูว่าผลลัพธ์ใดของ F ที่ตรงกับผลลัพธ์ของเพรดิเคตดังกล่าว

3. แทนที่ค่าของ F ด้วย 1 หรือ 0 หรือ -1 กรณีที่ค่าผลลัพธ์ของ F เป็นบวก เป็นศูนย์ และเป็นลบ ตามลำดับ

4. คำนวณหาค่าของตัวแปร

ในกรณีที่ตัวแปรในนิพจน์มีจำนวนมากกว่า 1 ตัวแปร เมื่อจัดให้อยู่ในรูปของบรานช์ฟังก์ชันและแทนที่ F ด้วยค่าที่ระบุไว้ในขั้นตอนที่ 3 แล้ว จะสามารถหาค่าของตัวแปรต่าง ๆ ได้ด้วยวิธีดังต่อไปนี้

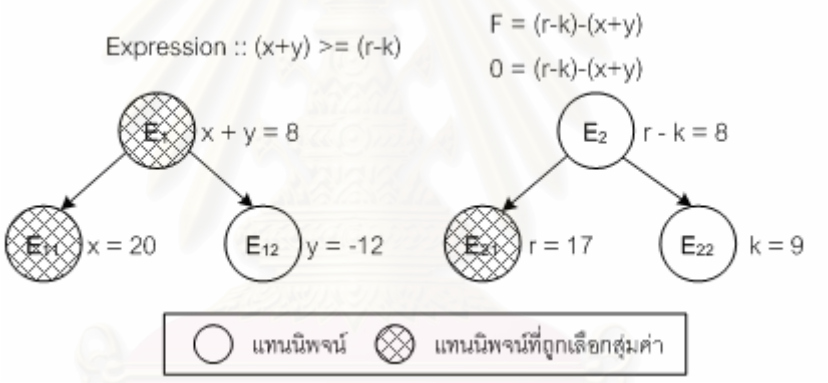
1) หากนิพจน์ E_1 หรือ E_2 เป็นตัวเลข ซึ่งไม่มีตัวดำเนินการตัวเลข (Arithmetic Operator)ใด ๆ ปรากฏอยู่ เช่น (x+k)>= 30 (T) (E_1 = x+k , E_2 = 30) ในที่นี้จะเลือกนิพจน์อีกตัวหนึ่งที่ไม่ใช่ตัวเลข (E_1) มาแตกเป็นนิพจน์ย่อย และสุ่มค่าของนิพจน์ดังกล่าว และคำนวณหาค่าตัวแปร โดยจะทำในลักษณะเช่นนี้ซ้ำไปเรื่อย ๆ จนกระทั่งไม่สามารถแตกนิพจน์ย่อยได้อีก คือไม่พบตัวดำเนินการตัวเลขในนิพจน์ย่อยอีกแล้ว ตัวอย่างของการแตกนิพจน์และสุ่มหาค่าในกรณีนี้ ถูกแสดงไว้ในรูปที่ 4-31



รูปที่ 4-31 แสดงตัวอย่างและวิธีแตกนิพจน์ย่อยและเลือกสุ่มค่าในกรณีที่ E_1 หรือ E_2 เป็นตัวเลข

จากรูปที่ 4-31 เมื่อแตกนิพจน์ย่อยเพื่อสุ่มหาค่าแล้ว จะได้ค่าของตัวแปรคือ {x =20, k=10} เมื่อเอาค่าของตัวแปรดังกล่าวมาแทนตัวแปรในนิพจน์แล้วจะได้ (20+10)>=30 ซึ่งทำให้ค่าของนิพจน์นี้เป็นจริง ดังนั้นชุดของค่าตัวแปรดังกล่าว จะถูกนำมาใช้เป็นข้อมูลทดสอบของเส้นทางไออาร์ซีเอฟจีที่มีนิพจน์ (x+k)>=30 ประกอบอยู่ในคอนดิชันโหนดและมีเส้นเชื่อมทีของคอนดิชันโหนดนั้นปรากฏอยู่ในเส้นทาง (ค่าของตัวแปรอาจแตกต่างกันไป เนื่องจากเกิดจากการสุ่มหาค่า)

2) หากนิพจน์ E_1 และ E_2 ประกอบด้วยตัวแปรตั้งแต่ 1 ตัวขึ้นไป เช่น (x+y)>=(r-k) (T) (E_1 = x+y , E_2 = r-k) ในที่นี้จะเลือกนิพจน์ E_i โดยที่ i = {1,2} เพื่อสุ่มหาค่าของนิพจน์และคำนวนหาค่าของตัวแปร และแบ่งนิพจน์ดังกล่าวให้อยู่ในรูปแบบของนิพจน์ย่อยและเลือกสุ่มค่าของนิพจน์ย่อยที่ไม่ใช่ตัวเลข และทำในลักษณะเช่นนี้ไปเรื่อย ๆ เหมือนกรณีข้างต้น ตัวอย่างของการแตกนิพจน์และสุ่มหาค่าในกรณีนี้ถูกแสดงไว้ในรูปที่ 4-32



รูปที่ 4-32 แสดงตัวอย่างและวิธีแตกนิพจน์ย่อยและเลือกสุ่มค่าในกรณีที่มี E_1 และ E_2 ประกอบด้วยตัวแปรตั้งแต่ 1 ตัวขึ้นไป

จากตัวอย่างในรูปที่ 4-32 เมื่อแตกนิพจน์ย่อยเพื่อสุ่มหาค่าแล้ว จะได้ค่าของตัวแปรคือ {x =20, y=-12, r=17, k=9} เมื่อเอาค่าของตัวแปรดังกล่าวมาแทนตัวแปรในนิพจน์แล้วจะได้เป็น (20-(-12)) >= (17-9) ซึ่งทำให้ค่าของนิพจน์นี้เป็นจริง ดังนั้นชุดของค่าตัวแปรดังกล่าว จะถูกนำมาใช้เป็นข้อมูลทดสอบของเส้นทางไออาร์ซีเอฟจีที่มีนิพจน์ (x+y)>=(r-k) ประกอบอยู่ในคอนดิชันโหนดและมีเส้นเชื่อมทีของคอนดิชันโหนดนั้นปรากฏอยู่ในเส้นทาง

2. **ข้อมูลประเภทสตริง**

รูปแบบของข้อมูลประเภทสตริงของงานวิจัยนี้จะต้องเป็นไปตามรูปที่ 4-33 โดยE_1 คือชื่อของตัวแปรสตริง E_2 คือกลุ่มของอักขระเรียงต่อกันซึ่งการใส่ข้อมูลประเภทสตริงให้กับตัวแปรนั้นจะต้องอยู่ภายใต้เครื่อง double quote (“”) และ op คือหนึ่งในตัวดำเนินการ {=, !=}

E_1 op “E_2”

รูปที่ 4-33 รูปแบบของข้อมูลประเภทสตริงของงานวิจัยนี้

ยกตัวอย่างเช่น เพรดิเคต cusType=”A” ค่าของข้อมูลที่ทำให้ผลลัพธ์ของเพรดิเคตนี้เป็นจริงคือ cusType = A และค่าของข้อมูลที่ทำให้ผลลัพธ์ของเพรดิเคตนี้เป็นเท็จคือ cusType = !A เป็นต้น ดังแสดงไว้ในตารางที่ 4-6

ตารางที่ 4-6 แสดงลักษณะของเพรดิเคต ผลลัพธ์เพรดิเคต และค่าของข้อมูลประเภทตัวแปรสตริง

Branch Predicate	ผลลัพธ์เพรดิเคต	ค่าของข้อมูล
E_1 == “E_2“	T	$E_1 = E_2$
	F	$E_1 = !E_2$
E_1 != “E_2“	T	$E_1 = !E_2$
	F	$E_1 = E_2$

3. **ข้อมูลประเภทบูลีน**

ข้อมูลประเภทบูลีน คือข้อมูลที่มีค่าเป็นจริง (True) หรือเป็นเท็จ (False) โดยรูปแบบของข้อมูลประเภทบูลีนในงานวิจัยนี้ จะต้องไม่ปรากฏตัวดำเนินการใด ๆ ภายในนิพจน์บูลีน คือจะต้องอยู่ในรูปแบบของตัวแปรเดี่ยว หรือมีเครื่องหมาย “!” วางอยู่หน้านิพจน์ ดังรูปที่ 4-34

E หรือ !E

รูปที่ 4-34 รูปแบบของข้อมูลประเภทบูลีนของงานวิจัยนี้

ยกตัวอย่างเช่น เพรดิเคต isValid อยู่ในรูปแบบของนิพจน์บูลีน ซึ่งค่าของข้อมูลที่ทำให้ผลลัพธ์ของเพรดิเคตนี้เป็นจริงคือ isValid = true และค่าของข้อมูลที่ทำให้ผลลัพธ์ของเพรดิเคตนี้เป็นเท็จคือ isValid = false เป็นต้น ดังแสดงไว้ในตารางที่ 4-7

ตารางที่ 4-7 แสดงลักษณะของเพรดิเคต ผลลัพธ์เพรดิเคต และค่าของข้อมูลประเภทตัวแปรบูลีน

Branch Predicate	**ผลลัพธ์เพรดิเคต**	**ค่าของข้อมูล**
E	T	E = true
	F	E = false
!E	T	E = false
	F	E = true

4.6 การสร้างกรณีทดสอบ

หลังจากสร้างข้อมูลทดสอบครบทุกเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์แล้ว และได้กำหนดเงื่อนไขก่อนและหลังดำเนินการของแต่ละเมท็อดเรียบร้อยแล้ว ในส่วนนี้จะเป็นการผนวกข้อมูลดังกล่าวเข้าไว้ด้วยกันให้เป็นชุดของกรณีทดสอบ โดยยึดเอาเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์เป็นหลัก โดยขั้นตอนในการกรณีทดสอบมีดังต่อไปนี้

1. เลือกเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ขึ้นมา 1 เส้นทาง
2. เลือกเมท็อดที่ปรากฏในเส้นทางโดยเลือกแบบโพสออร์เดอร์
3. นำเงื่อนไขก่อนดำเนินการของเมท็อดที่ได้เลือกในขั้นตอนที่ 2 มาใช้เป็นข้อมูลเบื้องต้นของกรณีทดสอบ
4. นำเงื่อนไขหลังดำเนินการของเมท็อดที่ได้เลือกในขั้นตอนที่ 2 มาใช้เป็นผลลัพธ์คาดหวังของกรณีทดสอบ
5. ทำขั้นตอนที่ 2-4 ซ้ำจนครบทุกเมท็อดที่ปรากฏในเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ที่ได้เลือกไว้
6. นำค่าของตัวแปรต่าง ๆ ที่ได้สร้างไว้สำหรับเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ที่ถูกสร้างไว้ในขั้นของการสร้างข้อมูลทดสอบ มาเป็นข้อมูลทดสอบของกรณีทดสอบ
7. ทำขั้นที่ 1-6 ซ้ำจนครบทุกเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์

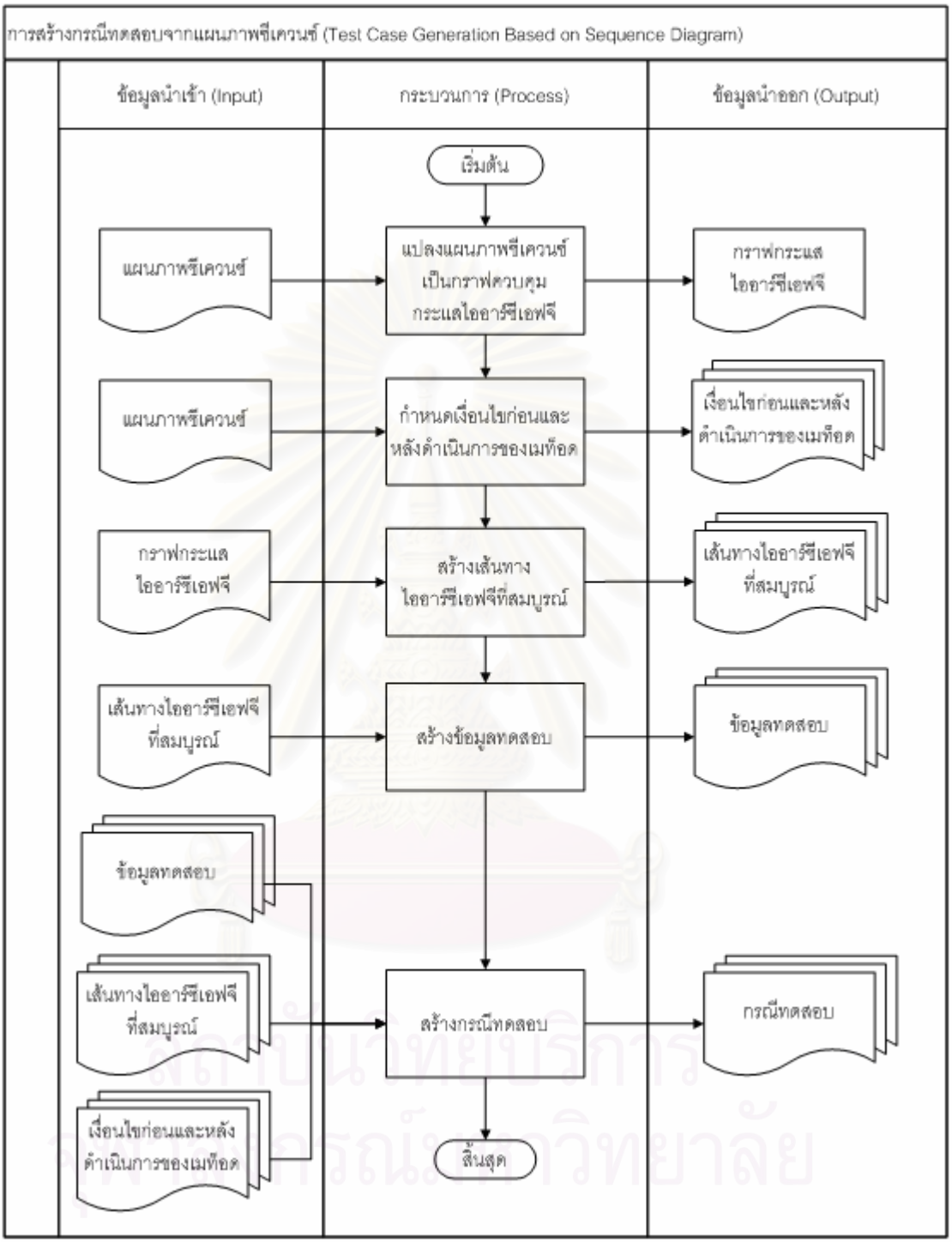
หลังจากทำครบทุกขั้นตอนแล้ว จะได้ชุดของกรณีทดสอบจากแผนภาพซีเควนซ์ ดังตัวอย่างที่แสดงในตารางที่ 4-8

ตารางที่ 4-8 ตารางแสดงตัวอย่างของกรณีทดสอบที่สร้างจากหลักการโดยใช้แผนภาพซีเควนซ์การถอนเงิน

No.	**ข้อมูลเบื้องต้น** (Initial Data)	**ข้อมูลทดสอบ** (Test Data)	**ผลลัพธ์คาดหวัง** (Expected Result)
1	• Bank->includes(acc) • acc.getBalance() >= amount	• {balance =2000, amount =2000}	• result = acc • result = balance • acc.getBalance() = acc.getBalance()@pre-amount
2	• Bank->includes(acc)	• {balance =2000, amount =1999}	• result = acc • result = balance • result = “Can’t Withdraw. Amount is more than balance”

จากขั้นตอนในการให้ได้มาซึ่งกรณีทดสอบทุกขั้นตอนที่ได้กล่าวมา สามารถสรุปข้อมูลนำเข้าและข้อมูลนำออกของแต่ละกระบวนการในการสร้างกรณีทดสอบได้ดังแสดงไว้ในรูปที่ 4-35

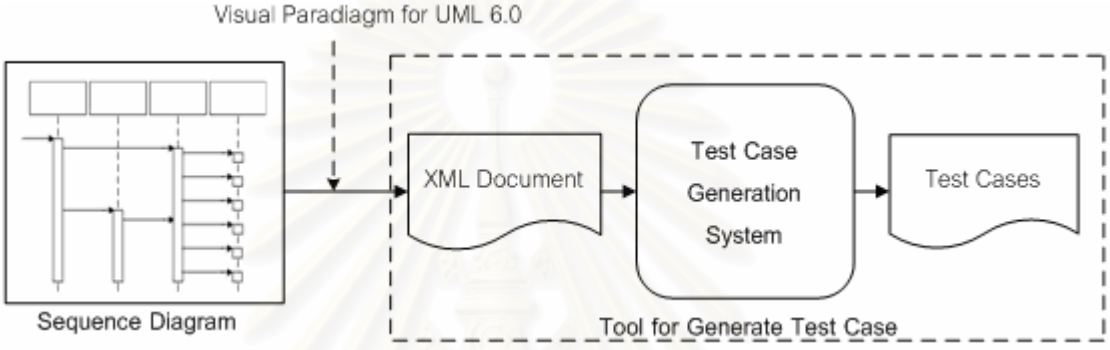
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่4-35 แสดงการสรุปข้อมูลนำเข้าและข้อมูลนำออกของแต่ละกระบวนการในการสร้างกรณีทดสอบ

4.7 การออกแบบและพัฒนาเครื่องมือสร้างกรณีทดสอบ

เครื่องมือสำหรับสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ถูกสร้างขึ้นตามแนวคิดที่ผู้วิจัยได้นำเสนอไว้ในหัวข้อต่าง ๆ ข้างต้น โดยแผนภาพซีเควนซ์ที่ต้องการนำมาสร้างกรณีทดสอบจะต้องถูกแปลงให้อยู่ในรูปแบบของเอกสารเอกซ์เอ็มแอล (XML Document) ด้วยเครื่องมือวิชวลพาราแดมสำหรับยูเอ็มแอลเวอร์ชัน 6.0 (Visual Paradigm for UML 6.0) ก่อน ดังแสดงในรูปที่ 4-36 โดยผู้วิจัยมีขั้นตอนการออกแบบและพัฒนาเครื่องมือสร้างกรณีสอบดังนี้



รูปที่ 4-36 แสดงภาพรวมของการสร้างกรณีทดสอบด้วยเครื่องมือสร้างกรณีทดสอบ

4.7.1 ขอบเขตของเครื่องมือและผู้ใช้งาน

เครื่องมือสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ เป็นเครื่องมือที่ช่วยให้ผู้ใช้งานสามารถสร้างกรณีทดสอบจากเอกสารการออกแบบแผนภาพซีเควนซ์ โดยมีลักษณะการทำงานเบื้องต้นของเครื่องมือดังต่อไปนี้

1. เครื่องมืออ่านข้อมูลแผนภาพซีเควนซ์ในรูปแบบของเอกสารเอกซ์เอ็มแอล และสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจี พร้อมทั้งสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์

2. เครื่องมือบันทึกเงื่อนไขก่อนดำเนินการและหลังดำเนินการของแต่ละเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ตามที่ผู้ใช้งานป้อน

3. เครื่องมือสร้างชุดกรณีทดสอบของแต่ละเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์และแสดงชุดกรณีทดสอบในรูปแบบตาราง

ผู้ใช้งานเครื่องมือได้แก่ ผู้ทดสอบ (Tester) ผู้วิเคราะห์การทดสอบ (Test Analyst) หรือบุคคลทั่วไปที่ต้องการสร้างกรณีทดสอบในส่วนการทดสอบระดับบูรณาการของซอฟต์แวร์เชิงวัตถุ โดยสามารถสร้างกรณีทดสอบและนำออกกรณีทดสอบที่สร้างขึ้นเป็นไฟล์เอกสารนามสกุลเอกซ์แอลเอส (xls.doc) ได้ เพื่อสะดวกต่อการนำกรณีทดสอบดังกล่าวไปใช้ในขั้นตอนอื่น ๆ ของกระบวนการพัฒนาซอฟต์แวร์

4.7.2 **เอกสารความต้องการซอฟต์แวร์** (Software Requirement Specification)

เป็นเอกสารแสดงการทำงานของซอฟต์แวร์ที่พัฒนาขึ้น สามารถแบ่งเป็นข้อกำหนดของความต้องการซอฟต์แวร์ (Software Requirement Specification) ได้ดังนี้

1. ความต้องการด้านหน้าที่ (Functional Requirement)

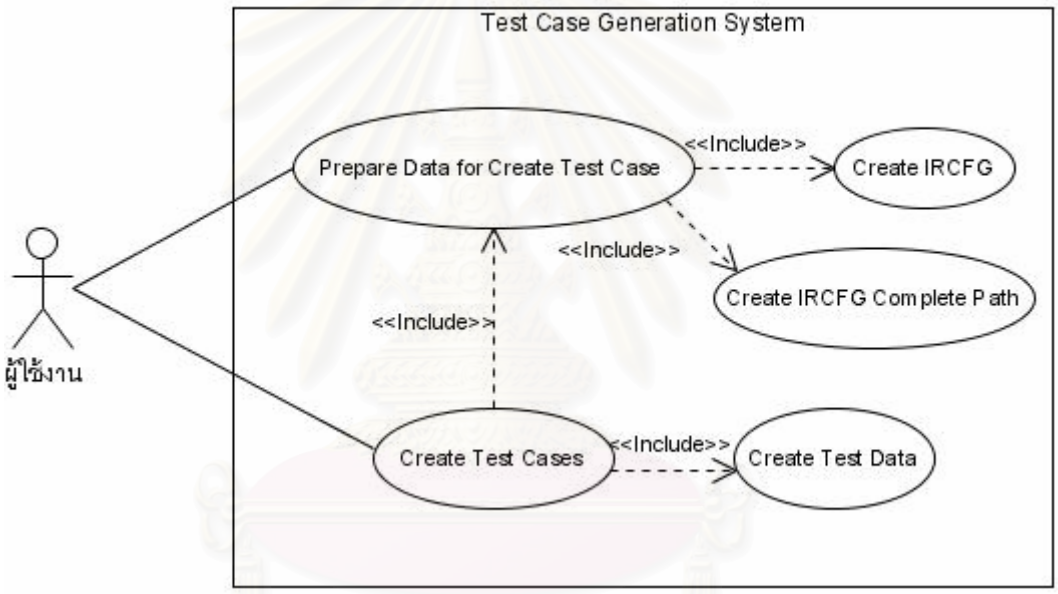
- ซอฟต์แวร์สามารถอ่านข้อมูลแผนภาพซีเควนซ์ในรูปของเอกสารเอกซ์เอ็มแอลได้ ซึ่งแผนภาพซีเควนซ์นั้นจะต้องถูกออกแบบโดยใช้คอมบายด์แฟรกเมนต์ประเภท ออบชัน (Option) อัลเทอร์เนทีฟ (Alternative) และลูป (Loop) เท่านั้น
- ซอฟต์แวร์สามารถสร้างโอซีแอล (OCL) ในส่วนของเงื่อนไขก่อนและหลังการดำเนินการแบบเบื้องต้นสำหรับเมจเสจประเภทสร้างใหม่ (Create New) และเมจเสจประเภทที่มีการคืนค่ากลับ
- ซอฟต์แวร์สามารถบันทึกข้อจำกัดโอซีแอลในส่วนของเงื่อนไขก่อนและหลังการดำเนินการ (Postcondition) ของแต่ละเมท็อด
- ซอฟต์แวร์สามารถสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจีจากข้อมูลของแผนภาพซีเควนซ์ได้ และเป็นไปตามหลักการที่ได้กำหนดไว้ในงานวิจัย
- ซอฟต์แวร์สามารถสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์จากกราฟควบคุมกระแสไออาร์ซีเอฟจี
- ซอฟต์แวร์สามารถสร้างกรณีทดสอบของแต่ละเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ได้
- ซอฟต์แวร์สามารถแสดงชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์และแสดงตารางกรณีทดสอบออกทางหน้าจอได้
- ซอฟต์แวร์สามารถบันทึกกรณีทดสอบเป็นไฟล์เอกสารนามสกุลเอกซ์แอลเอส (xls.doc) ซึ่งเปิดได้ด้วยโปรแกรมไมโครซอฟต์เอกซ์เซล์ (Microsoft Excel) ได้

2. ความต้องการที่ไม่ใช่หน้าที่ (Non-Functional Requirement)

- ขั้นตอนในการใช้งานซอฟต์แวร์สามารถเข้าใจได้โดยง่าย ไม่ทำให้ผู้ใช้งานสับสน
- ซอฟต์แวร์สามารถสร้างกรณีทดสอบได้อย่างรวดเร็ว

4.7.3 **แผนภาพยูสเคส** (Use Case Diagram)

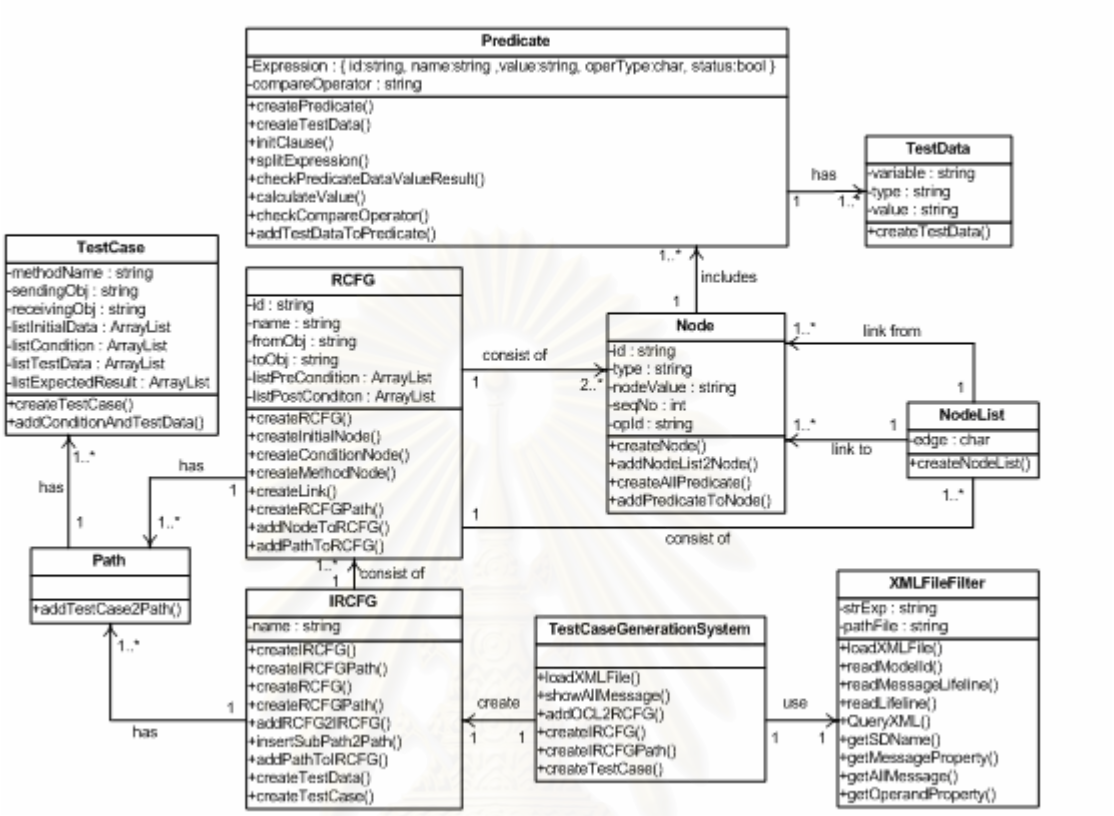
เป็นเอกสารที่แสดงการโต้ตอบระหว่างผู้ใช้กับระบบ จากเอกสารความต้องการซอฟต์แวร์ข้างต้นประกอบด้วยยูสเคสดังต่อไปนี้ (1) ยูสเคสการเตรียมข้อมูลสำหรับสร้างกรณีทดสอบ (Prepare Data for Create Test Case) (2) ยูสเคสการสร้างกราฟควบคุมกระแสไออาร์ซีเอฟจี (Create IRCFG) (3) ยูสเคสการชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์(Create IRCFG Complete Path) (4) ยูสเคสการสร้างกรณีทดสอบ (Create Test Case) และ (5) ยูสเคสการสร้างข้อมูลทดสอบ (Create Test Data) ดังแสดงไว้ในรูปที่ 4-37 และสามารถดูเอกสารคำอธิบายยูสเคสได้ที่ภาคผนวก ก



รูปที่ 4-37 แสดงแผนภาพยูสเคสของเครื่องมือที่พัฒนาขึ้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.7.4 แผนภาพคลาส (Class Diagram)



รูปที่ 4-38 แสดงแผนภาพคลาสของเครื่องมือที่พัฒนาขึ้น

4.7.5 การพัฒนาเครื่องมือสร้างกรณีทดสอบ

จากเอกสารในขั้นตอนการออกแบบ (สามารถดูเอกสารอย่างละเอียดได้ที่ภาคผนวก ก) สามารถนำมาพัฒนาเป็นซอฟต์แวร์สร้างกรณีทดสอบจากแผนภาพซีเควนซ์ ซึ่งมีลักษณะเป็นโปรแกรมประยุกต์แบบวินโดวส์ (Windows Application) โดยเครื่องมือที่ทางผู้วิจัยพัฒนาขึ้นสามารถอ่านข้อมูลของแผนภาพซีเควนซ์ในรูปแบบของเอกสารเอกซ์เอ็มแอล และสร้างข้อมูลให้อยู่ในรูปแบบของกราฟควบคุมกระแสไออาร์ซีเอฟจี สร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ และสร้างกรณีทดสอบตามเส้นทางไออาร์ซีเอฟจี โดยผู้วิจัยได้ออกแบบและสร้างตัวต่อประสาน (Interface) ของเครื่องมือสร้างกรณีทดสอบ โดยสามารถแบ่งหน้าจอการทำงานเป็น 4 ส่วน ได้แก่

1. หน้าจอแสดงรายละเอียดเอกสารเอกซ์เอ็มแอล

ในส่วนนี้เป็นการเริ่มต้นใช้งานโปรแกรม คือผู้ใช้จะต้องเลือกเปิดเอกสารเอกซ์เอ็มแอล เพื่อนำเข้าเอกสารแผนภาพซีเควนซ์ที่อยู่ในรูปแบบของเอกสารเอกซ์เอ็มแอลที่ต้องการนำมาสร้างกรณีทดสอบ จากนั้นระบบจะแสดงรายละเอียดของเอกสารเอกซ์เอ็มแอลที่เลือก ดังแสดงในรูปที่ 4-39

2. หน้าจอแสดงรูปภาพของแผนภาพซีเควนซ์

หน้าจอนี้ถูกสร้างขึ้นเพื่อให้ผู้ใช้งานสามารถเปิดรูปภาพของแผนภาพซีเควนซ์ที่ต้องการนำมาใช้สร้างกรณีทดสอบมาแสดงในโปรแกรมได้ เพื่อช่วยให้ผู้ใช้งานสามารถเห็นรูปแบบของแผนภาพซีเควนซ์และข้อมูลภายในแผนภาพซีเควนซ์ที่จะนำมาใช้สร้างกรณีทดสอบได้ ดังแสดงในรูปที่ 4-40

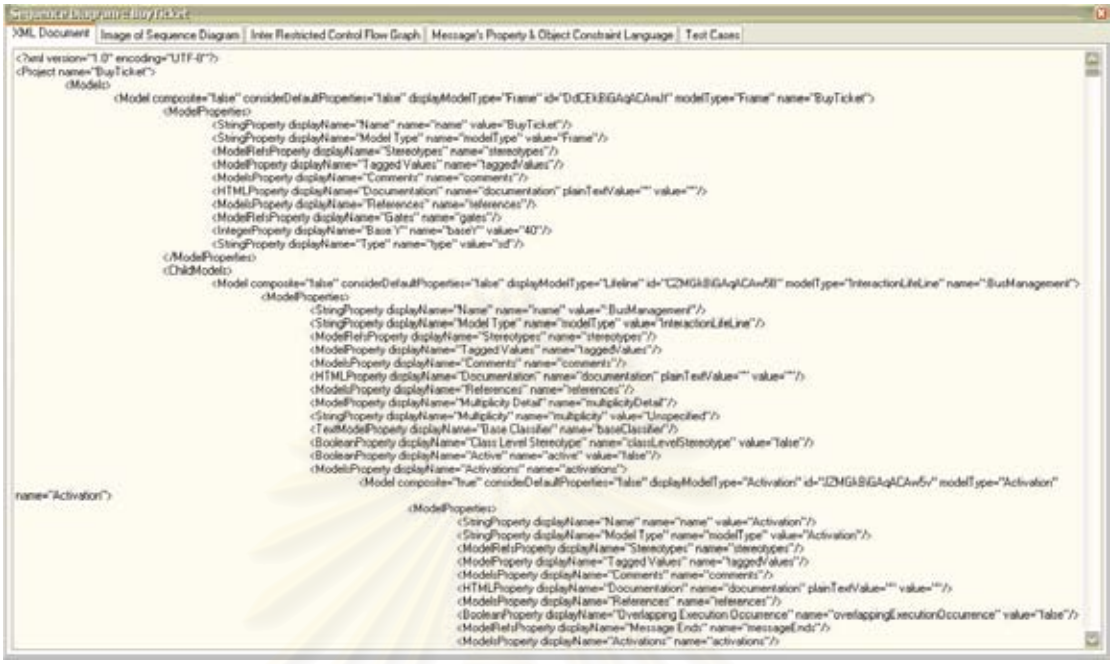
3. หน้าจอแสดงคุณสมบัติของเมจเสจ โอซีแอล และตัวแปร

หน้าจอนี้จะสามารถแบ่งส่วนของการทำงานเป็น 2 ส่วนหลัก ได้แก่

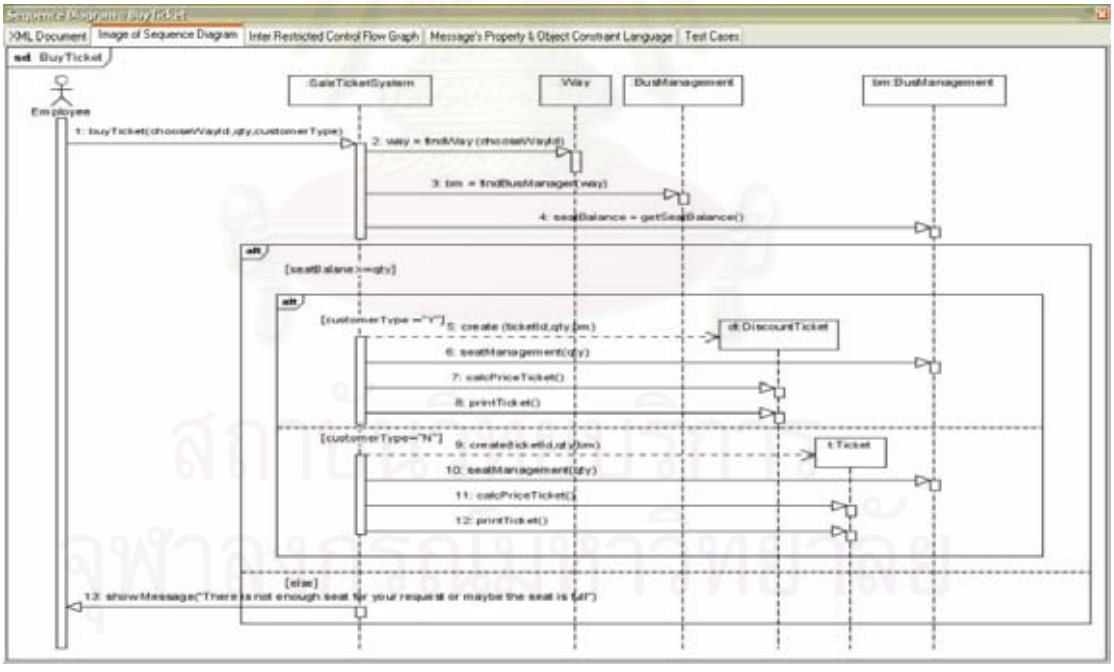
- ส่วนคุณสมบัติของเมจเสจและโอซีแอล โปรแกรมจะแสดงรายละเอียดของเมจเสจที่ปรากฏในแผนภาพซีเควนซ์ และให้ผู้ใช้งานได้กำหนดเงื่อนไขก่อนและหลังดำเนินการของเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ได้
- ส่วนของตัวแปร โปรแกรมจะแสดงรายละเอียดของตัวแปรที่พบในเงื่อนไขของการเรียกใช้งานเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ เพื่อให้ผู้ใช้งานสามารถกำหนดค่าขอบเขตสูงสุดและค่าขอบเขตต่ำสุดของตัวแปรประเภทข้อมูลตัวเลขจำนวนเต็ม รายละเอียดของหน้าจอนี้แสดงดังรูปที่ 4-41

4. หน้าจอแสดงเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์และชุดกรณีทดสอบ

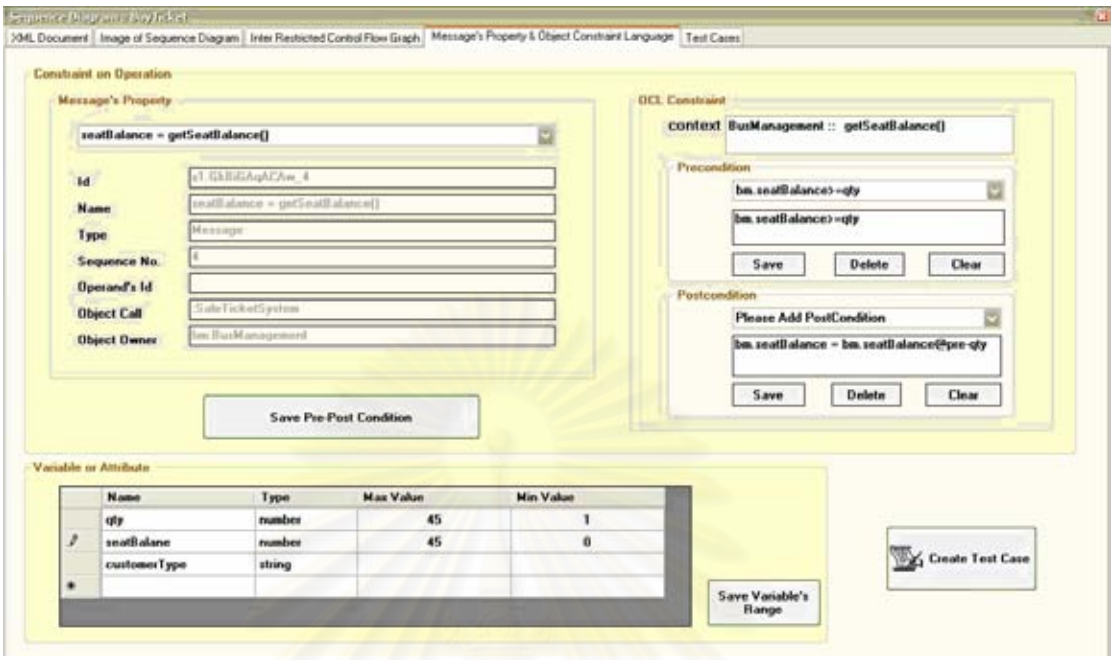
หน้าจอนี้แสดงตารางของเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ทุกเส้นทางที่ปรากฏอยู่ในแผนภาพซีเควนซ์ และแสดงตารางของชุดกรณีทดสอบซึ่งสร้างขึ้นตามข้อมูลของแผนภาพซีเควนซ์และข้อมูลที่ผู้ใช้งานได้กำหนดไว้ในหน้าที่ 3 ข้างต้น รายละเอียดของหน้าจอนี้แสดงดังรูปที่ 4-42



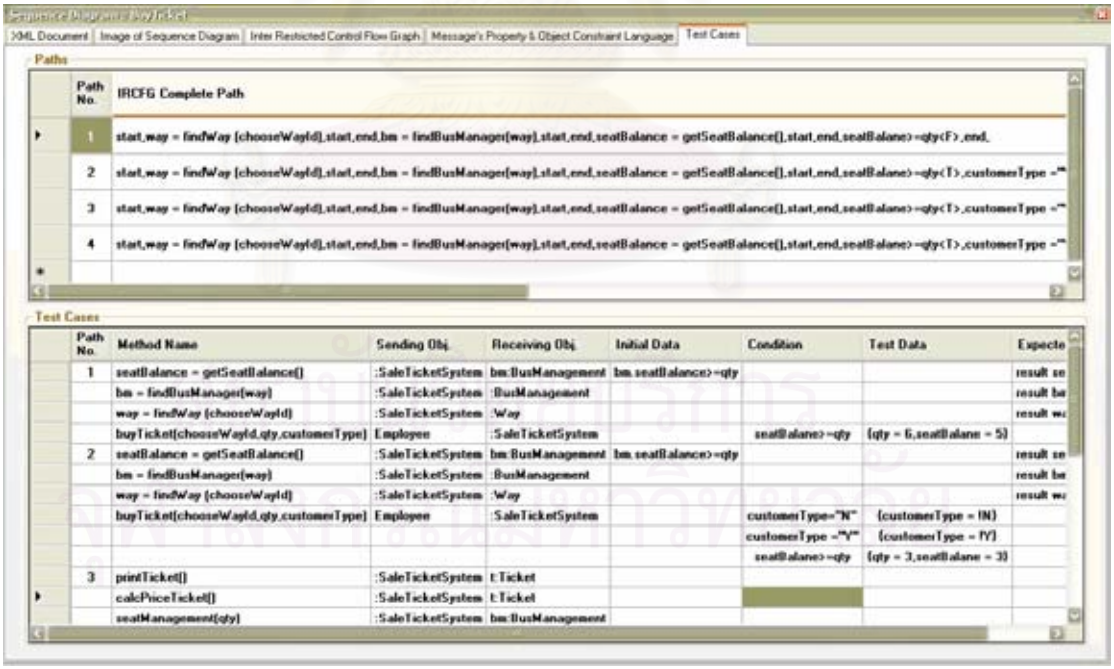
รูปที่ 4-39 หน้าจอแสดงรายละเอียดเอกสารเอกซ์เอ็มแอลของเครื่องมือสร้างกรณีทดสอบ



รูปที่ 4-40 หน้าจอแสดงรูปภาพของแผนภาพซีเควนซ์ของเครื่องมือสร้างกรณีทดสอบ



รูปที่ 4-41 หน้าจอแสดงคุณสมบัติของเมจเสจ โอซีแอล
และตัวแปรของเครื่องมือสร้างกรณีทดสอบ



รูปที่ 4-42 หน้าจอแสดงเส้นทางไอการ์ซีเอฟจีที่สมบูรณ์และกรณีทดสอบ
ของเครื่องมือสร้างกรณีทดสอบ

บทที่ 5
การทดสอบและประเมินผลเครื่องมือสร้างกรณีทดสอบ

บทนี้กล่าวถึงการทดสอบและประเมินผลเครื่องมือสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ซึ่งพัฒนาขึ้นตามแนวคิดที่ผู้วิจัยได้นำเสนอไว้ในบทที่ 4 ส่วนของการทดสอบเป็นการนำแผนภาพซีเควนซ์ของยูสเคสทั้ง 6 ยูสเคสที่ได้กล่าวไว้ในบทที่ 3 มาเป็นข้อมูลนำเข้าของเครื่องมือ และในส่วนของการประเมินผลเครื่องมือสร้างกรณีทดสอบจะเป็นการประเมินประสิทธิผลของกรณีทดสอบที่ถูกสร้างขึ้นจากเครื่องมือด้วยการนำมาเปรียบเทียบกับกรณีทดสอบที่ถูกสร้างขึ้นจากผู้เชี่ยวชาญ

5.1 การทดสอบเครื่องมือสร้างกรณีทดสอบ

จากซอร์สโค้ดของ 6 ยูสเคส ผู้วิจัยได้นำมาแปลงกลับเป็นเอกสารแผนภาพซีเควนซ์ด้วยเครื่องมือช่วยสร้างย้อนรอยบอร์แลนด์ทูเก็ตเตอร์ (Borland Together) เมื่อแปลงกลับแล้วจะได้เป็นเอกสารแผนภาพ ซีเควนซ์จำนวนรวมทั้งสิ้น 6 แผนภาพ (รายละเอียดที่มาของยูสเคสและการแปลงกลับเป็นแผนภาพซีเควนซ์กล่าวไว้ในบทที่ 3)

แผนภาพซีเควนซ์ทั้ง 6 แผนภาพที่นำมาเป็นหน่วยตัวอย่าง ได้แก่ แผนภาพของยูสเคสการซื้อตั๋ว ยูสเคสการเข้าใช้งาน ยูสเคสการยอมรับอินวอยซ์ ยูสเคสการโปรโมชันร่วมกับการขาย ยูสเคสการลดราคา และยูสเคสการส่งสินค้า ข้อมูลสรุปของแผนภาพซีเควนซ์และคอมบายด์แฟรกเมนต์ของแต่ละแผนภาพแสดงไว้ในตารางที่ 3-5 และ 3-6 ตามลำดับ (รายละเอียดของยูสเคสและแผนภาพซีเควนซ์ที่ได้จากการแปลงกลับจากซอร์สโค้ดของทั้ง 6 ยูสเคส สามารถดูได้ที่ภาคผนวก ข)

5.1.1 ขั้นตอนการทดสอบ

การทดสอบเครื่องมือที่พัฒนาขึ้นประกอบด้วยขั้นตอนต่าง ๆ ดังนี้

1. จากซอร์สโค้ดของแต่ละยูสเคส ผู้วิจัยแปลงกลับเป็นเอกสารแผนภาพซีเควนซ์ด้วยเครื่องมือบอร์แลนด์ทูเก็ตเตอร์และสร้างเป็นเอกสารแผนภาพซีเควนซ์ด้วยเครื่องมือช่วยสร้างวิชวลพาราแดม สำหรับยูเอ็มแอล (Visual Paradigm for UML)

2. นำออก (Export) เอกสารแผนภาพซีเควนซ์ในรูปแบบของเอกสารเอกซ์เอ็มแอลด้วยเครื่องมือวิชวลพาราแดม สำหรับยูเอ็มแอล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

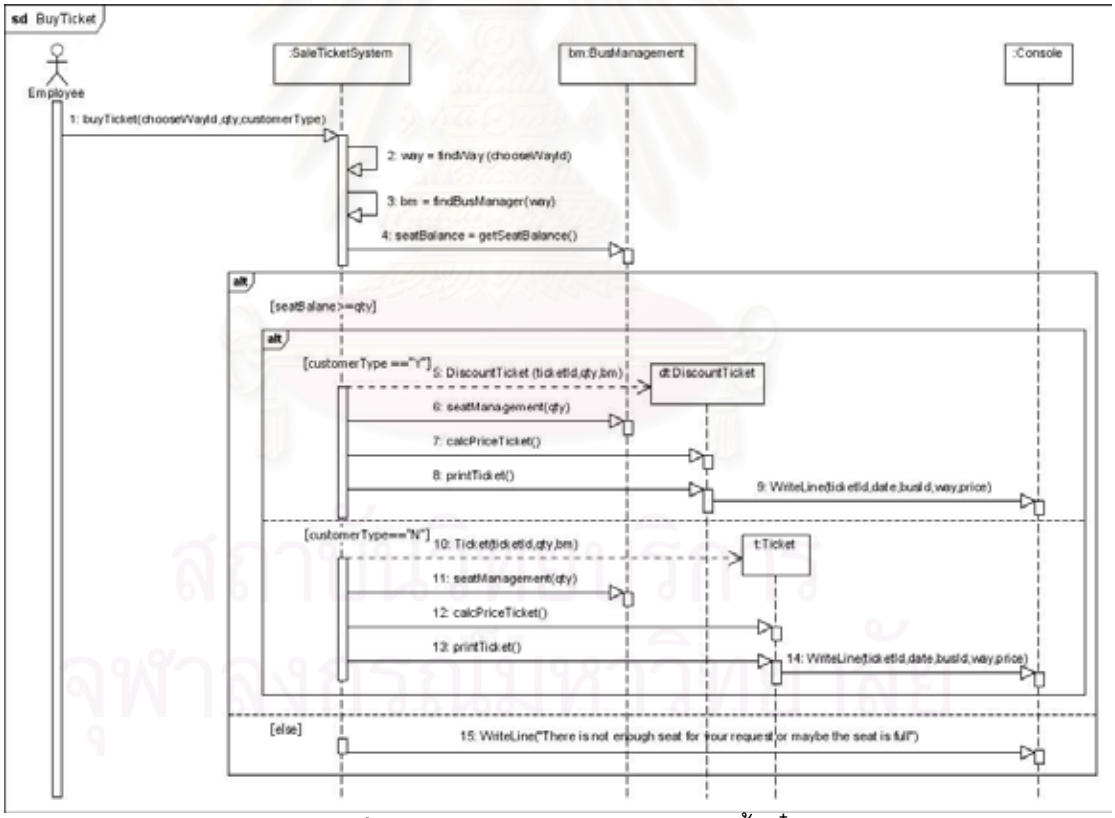
3. สร้างกรณีทดสอบจากเอกสารเอกซ์เอ็มแอลและระบุเงื่อนไขก่อนและหลังดำเนินการของเมท็อดด้วยเครื่องมือที่พัฒนาขึ้น

4. พิจารณารูปแบบข้อมูลของกราฟควบคุมกระแสไออาร์ซีเอฟจีว่าเป็นไปตามหลักการที่กำหนดไว้หรือไม่ และได้จำนวนเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ครบทุกเส้นทางหรือไม่

5. พิจารณากรณีทดสอบที่สร้างขึ้นว่ามีจำนวนเท่ากับเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ทุกเส้นทางหรือไม่

6. พิจารณาข้อมูลทดสอบว่าเป็นไปตามหลักการที่ได้กำหนดไว้ในบทที่ 4 หรือไม่

ตัวอย่างของแผนภาพซีเควนซ์ที่แปลงกลับได้จากซอร์สโค้ดและตัวอย่างของเอกสารเอกซ์เอ็มแอลของแผนภาพซีเควนซ์พร้อมทั้งเงื่อนไขก่อนดำเนินการและหลังดำเนินการเมท็อดของแผนภาพซีเควนซ์ถูกแสดงไว้ในรูปที่ 5-1 5-2 และตารางที่ 5-1 ตามลำดับ



รูปที่ 5-1 แสดงแผนภาพซีเควนซ์การซื้อตั๋ว

รูปที่ 5-2 แสดงเอกสารเอกซ์เอ็มแอลของแผนภาพซีเควนซ์การซื้อตั๋ว

ตารางที่ 5-1 แสดงเงื่อนไขก่อนและหลังดำเนินการของแต่ละเมท็อดในแผนภาพซีเควนซ์การซื้อตั๋ว

ลำดับที่	ชื่อเมท็อดที่ปรากฏในเมจเสจ	เงื่อนไขก่อนดำเนินการ (pre) และ เงื่อนไขหลังดำเนินการ(post) ด้วยภาษาโอซีแอล
1	buyTicket	-
2	findWay	**pre** way.isDefined() **pre** SaleSystem->includes(way) **post** result=way
3	findBusManager	**pre** SaleSystem->includes(bm) **pre** bm.isDefined() **pre** bm->includes(way) **post** result= bm
4	getSeatBalance	**post** seatBalance
5	DiscountTicket	**post** dt.isDefined()
6	seatManagerment	**pre** bm.seatBalance()>=qty **post** bm.seatBalance()=bm.seatBalance@pre –qty **post** bm.soldTicket = bm.soldTicket@pre+qty

ลำดับที่	ชื่อเมท็อดที่ปรากฏในเมจเสจ	เงื่อนไขก่อนดำเนินการ (pre) และ เงื่อนไขหลังดำเนินการ(post) ด้วยภาษาโอซีแอล
7	calcPriceTicket	**pre** bm.getPrice>0 **post** dt.totalPrice = (bm.getPrice*qty)- (dt.totalPrice*dt.discount)
8	printTicket	**pre** dt.qty>0
9	WriteLine	-
10	Ticket	**post** t.isDefined()
11	seatManagement	**pre** bm.getSeatBalance()>=qty **post** bm.getSeatBalance()=bm.getSeatBalance@pre – qty **post** bm.soldTicket = bm.soldTicket
12	calcPriceTicket	**pre** bm.getPrice>0 **post** t.totalPrice = (bm.getPrice*qty)
13	printTicket	**pre** t.qty>0
14	WriteLine	-
13	WriteLine	**post** result = “There is not enough seat for your request or maybe the seat is full”

5.1.2 ผลการทดสอบ

จากการนำแผนภาพซีเควนซ์ทั้ง 6 แผนภาพที่อยู่ในรูปแบบของเอกสารเอกซ์เอ็ม-แอลและเงื่อนไขก่อนและหลังดำเนินการของเมท็อดมาสร้างเป็นกรณีทดสอบด้วยเครื่องมือที่พัฒนาขึ้น สามารถสรุปจำนวนของกรณีทดสอบที่สร้างขึ้นจากแต่ละแผนภาพได้ดังแสดงในตารางที่ 5-2 (รายละเอียดของกรณีทดสอบที่สร้างขึ้นจากเครื่องมือสามารถดูได้ในภาคผนวก ข)

ตารางที่ 5-2 ตารางสรุปจำนวนกรณีทดสอบของแต่ละแผนภาพซีเควนซ์ที่สร้างจากเครื่องมือ

ชื่อแผนภาพซีเควนซ์	จำนวนกรณีทดสอบที่สร้างได้
การซื้อตั๋ว (BuyTicket)	4
การเข้าใช้งาน (Login)	5
การยอมรับอินวอยซ์ (Approve Invoice)	5
การโปรโมชันร่วมกับการขาย (Sale Promotion)	4
การลดราคา (Calculate Discount)	10
การส่งสินค้า (Shipping Order)	2

5.1.3 สรุปผลการทดสอบ

จากการทดสอบเครื่องมือสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ พบว่าเครื่องมือสามารถสร้างกรณีทดสอบได้ครอบคลุมทุกเส้นทางการเรียกใช้งานของเมจเสจที่ปรากฏในแผนภาพซีเควนซ์ คือจำนวนของกรณีทดสอบเท่ากับจำนวนของเส้นทางไอออาร์ซีเอฟจีที่สมบูรณ์ถูกต้องตามลำดับการเรียกใช้ และค่าของข้อมูลทดสอบที่สร้างได้ตรงตามหลักการที่ได้นำเสนอไว้ในบทที่ 4

5.2 การประเมินการใช้เครื่องมือและความสามารถของเครื่องมือ

หลังจากสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ทั้ง 6 แผนภาพแล้ว ผู้วิจัยได้นำกรณีทดสอบดังกล่าวมาประเมินว่าสามารถนำมาใช้งานได้จริงหรือไม่ โดยนำมาทดสอบกับโปรแกรมจริง และเพื่อประเมินว่ากรณีทดสอบที่สร้างขึ้นนั้นครบถ้วนหรือไม่ ผู้วิจัยจึงได้นำกรณีทดสอบเหล่านั้นมาเปรียบเทียบกับกรณีทดสอบที่ถูกสร้างขึ้นจากผู้เชี่ยวชาญ โดยรายละเอียดของส่วนต่าง ๆ ดังกล่าวมีดังต่อไปนี้

5.2.1 การทดสอบโปรแกรมด้วยกรณีทดสอบที่ได้จากเครื่องมือ

เพื่อเป็นการตรวจสอบว่ากรณีทดสอบที่สร้างขึ้นจากเครื่องมือ สามารถนำมาใช้งานได้จริง ในส่วนนี้ผู้วิจัยจึงนำกรณีทดสอบที่ถูกสร้างขึ้นจากเครื่องมือ มาทดสอบกับโปรแกรมตามส่วนที่ปรากฏในแผนภาพซีเควนซ์ โดยแบ่งการทดสอบออกเป็น 2 ส่วนคือ (1) กรณีผลการทดสอบโปรแกรมเป็นไปตามผลลัพธ์คาดหวัง และ (2) กรณีผลการทดสอบโปรแกรมไม่เป็นไปตามผลลัพธ์คาดหวัง เนื่องจากแผนภาพซีเควนซ์ของงานวิจัยนี้ถูกสร้างโดยการย้อนรอย (Reverse Engineering) จากโปรแกรมที่พัฒนาเสร็จเรียบร้อยแล้ว ดังนั้นเมื่อนำกรณีทดสอบที่สร้างแผนภาพ

ซีเควนซ์ด้วยเครื่องมือของงานวิจัยนี้ มาทดสอบกับโปรแกรมในส่วนเดียวกับที่ปรากฏในแผนภาพซีเควนซ์ ควรที่จะได้ผลลัพธ์ออกมาตรงกับผลลัพธ์ที่คาดหวังที่ระบุไว้ในกรณีทดสอบ และในทางกลับกันหากโปรแกรมดังกล่าวถูกแก้ไขให้เกิดความผิดพลาด (Error) เมื่อนำกรณีทดสอบที่สร้างจากเครื่องมือมาทดสอบแล้ว ผลลัพธ์ที่ได้ก็ควรที่จะไม่ตรงกับผลลัพธ์ที่คาดหวังในกรณีทดสอบเช่นกัน รายละเอียดของการทดสอบใน 2 ส่วน มีดังต่อไปนี้

1. **กรณีที่ผลการทดสอบโปรแกรมเป็นไปตามผลลัพธ์คาดหวัง**

เมื่อนำกรณีทดสอบมาใช้ในการทดสอบกับโปรแกรมแล้ว พบว่าโปรแกรมสามารถดำเนินการได้ตรงตามผลลัพธ์ที่ต้องการ คือค่าของข้อมูลทดสอบทำให้การดำเนินงานของโปรแกรมเป็นไปตามผลลัพธ์คาดหวังที่ได้ระบุไว้ในกรณีทดสอบ ในที่นี้ผู้วิจัยยกตัวอย่างของการทดสอบโปรแกรมการซื้อตั๋วด้วยชุดกรณีทดสอบของแผนภาพซีเควนซ์การซื้อตั๋วที่ถูกสร้างขึ้นจากเครื่องมือ

ในการทดสอบโปรแกรมโดยทั่วไป ข้อมูลทดสอบของกรณีทดสอบจะถูกนำมาเป็นข้อมูลนำ (Input) เข้าของโปรแกรม ซึ่งหมายความว่าข้อมูลทดสอบจะต้องตรงกับข้อมูลนำเข้าของโปรแกรม แต่เนื่องจากข้อมูลทดสอบที่ได้จากแผนภาพซีเควนซ์นั้นมีบางส่วนที่ไม่ตรงกับข้อมูลนำเข้า ดังแสดงในตารางที่ 5-3

ตารางที่ 5-3 ตารางแสดงชื่อของตัวแปรที่เป็นพารามิเตอร์ของเมท็อดแรกและชื่อตัวแปรที่เครื่องมือสามารถนำมาสร้างค่าได้ของแผนภาพซีเควนซ์การซื้อตั๋ว

ชื่อตัวแปรที่เป็นพารามิเตอร์ของเมท็อด	ชื่อตัวแปรที่เครื่องมือใช้สร้างข้อมูลทดสอบ
chooseWayId	seatBalance
qty	qty
customerType	customerType

จากตารางที่ 5-3 จะเห็นได้ว่าเครื่องมือไม่สามารถสร้างค่าของตัวแปรชื่อ chooseWayId ได้ เนื่องจากตัวแปรนี้ไม่กระทบต่อเงื่อนไขการเรียกใช้เมท็อด ในส่วนนี้ผู้วิจัยได้กำหนดค่าให้กับตัวแปรดังกล่าวโดยให้สอดคล้องกับประเภทของตัวแปร (Data Type) ทั้งนี้ต้องกำหนดให้สอดคล้องกับข้อมูลเบื้องต้นที่สร้างก่อนนำข้อมูลทดสอบไปใช้งานด้วย

ในส่วนของข้อมูลเบื้องต้นที่ต้องกำหนดให้กับระบบก่อนนำข้อมูลทดสอบไปใช้กรณีทดสอบที่สร้างจากเครื่องมือ ไม่สามารถสร้างค่าของแอตทริบิวส์สำหรับอ็อบเจกต์ที่กำหนดให้

สร้างขึ้นได้ทั้งหมด ยกเว้นแอตทริบิวส์ที่ปรากฏอยู่ในส่วนของการเรียกใช้เมท็อด ดังนั้นในส่วนนี้ ผู้ที่จะนำกรณีทดสอบนี้ไปใช้ จะต้องกำหนดค่าเหล่านี้ขึ้นเอง โดยให้สอดคล้องกับลักษณะการทำงานของข้อมูลทดสอบ เช่นเดียวกับค่าของข้อมูลทดสอบบางตัวที่เครื่องมือไม่สามารถสร้างให้ได้ก็จะต้องกำหนดค่าเองเช่นกัน

หลังจากปรับแก้ซอร์สโค้ดบางส่วนแล้ว ผู้วิจัยจึงได้นำชุดของกรณีทดสอบของการซื้อตั๋วมาทดสอบโปรแกรม โดยก่อนการทดสอบได้กำหนดค่าแอตทริบิวส์ของอ็อบเจกต์ way และ bm ตามที่ได้ระบุว่าให้สร้างขึ้นในส่วนของข้อมูลเบื้องต้น และเมื่อทดสอบโปรแกรมด้วยค่าของข้อมูลทดสอบที่สร้างขึ้นพบว่าโปรแกรมสามารถดำเนินการได้ตรงตามผลลัพธ์ที่ต้องการ คือค่าของข้อมูลทดสอบทำให้การดำเนินงานของโปรแกรมเป็นไปตามผลลัพธ์คาดหวังที่ได้ระบุไว้ในกรณีทดสอบ โดยชุดกรณีทดสอบและผลลัพธ์ที่ได้จากทดสอบถูกแสดงไว้ในตารางที่ 5-4

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5-4 แสดงชุดกรณีทดสอบของการซื้อตั๋วและผลลัพธ์ที่ได้จากการทดสอบกรณีที่การทดสอบเป็นไปตามผลลัพธ์คาดหวัง

ลำดับที่	ข้อมูลเบื้องต้น	ข้อมูลทดสอบ	ผลลัพธ์คาดหวัง	ผลการทดสอบ
1	• bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {qty=25, seatBalance=24}	• result = seatBalance • result = bm • result = way • result = “There is not enough seat for your request or maybe the seat is full.”	ตรงกับผลลัพธ์คาดหวัง
2	• bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {customerType=!Y,!N} • {qty=27, seatBalance=27}	• result = seatBalance • result = bm • result = way	ตรงกับผลลัพธ์คาดหวัง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ลำดับที่	ข้อมูลเบื้องต้น	ข้อมูลทดสอบ	ผลลัพธ์คาดหวัง	ผลการทดสอบ
3	• bm.seatBalance>=qty • bm.getPrice>0 • t.qty>0 • bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {customerType=N} • {qty=27, seatBalance=27}	• t.isDefined() • bm.seatBalance = bm.seatBalance@pre-qty • bm.soldTicket = bm.soldTicket@pre+qty • t.totalPrice = (bm.getPrice*qty) • result = seatBalance • result = bm • result = way	ตรงกับผลลัพธ์คาดหวัง
4	• dt.qty>0 • bm.getPrice>0 • bm.seatBalance>=qty • bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {customerType=Y} • {qty=27, seatBalance=27}	• dt.totalPrice = (bm.getPrice*qty)-(dt.totalPrice*dt.discount) • bm.seatBalance = bm.seatBalance@pre-qty • bm.soldTicket = bm.soldTicket@pre+qty • dt.isDefined() • result = seatBalance	ตรงกับผลลัพธ์คาดหวัง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ลำดับที่	ข้อมูลเบื้องต้น	ข้อมูลทดสอบ	ผลลัพธ์คาดหวัง	ผลการทดสอบ
			• result = bm • result = way	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2. กรณีที่ผลการทดสอบโปรแกรมไม่เป็นไปตามผลลัพธ์คาดหวัง

หลังจากทดสอบโปรแกรมข้างต้นแล้ว พบว่าผลลัพธ์ของการทดสอบเป็นไปตามผลลัพธ์คาดหวังที่ระบุในกรณีทดสอบ ดังนั้นเพื่อตรวจสอบในอีกแง่มุมหนึ่ง คือกรณีโปรแกรมที่นำมาทดสอบถูกพัฒนาขึ้นโดยไม่สอดคล้องกับข้อกำหนดความต้องการเบื้องต้นหรือกรณีที่โปรแกรมมีข้อผิดพลาด (Error) ดังนั้นผลลัพธ์ที่ได้จากการนำชุดกรณีทดสอบไปใช้ทดสอบกับส่วนของโปรแกรมดังกล่าว ควรที่จะขัดแย้งกับผลลัพธ์คาดหวังที่ระบุไว้ในแต่ละชุดกรณีทดสอบ ในส่วนนี้ผู้วิจัยจึงปรับแก้ซอร์สโค้ดส่วนที่จะนำกรณีทดสอบจากเครื่องมือมาทดสอบให้เกิดความผิดพลาด ซึ่งโดยทั่วไปแล้วความผิดพลาดสามารถแบ่งออกเป็น 3 ประเภทได้แก่

1) ความผิดพลาดขณะแปลโปรแกรม (Compile-time Error) เป็นความผิดพลาดของโปรแกรมที่เกิดจากการเขียนคำสั่งผิดไวยากรณ์ (Syntax) ของภาษา ซึ่งเป็นความผิดพลาดที่ตรวจจับได้โดยคอมไพเลอร์ (Compiler)

2) ความผิดพลาดขณะดำเนินงาน (Run-time Error) เป็นความผิดพลาดที่เกิดขึ้นขณะดำเนินงานโปรแกรม เช่น การหารด้วยศูนย์ การอ้างอิงค่าว่าง เป็นต้น ความผิดพลาดประเภทนี้จะถูกตรวจจับได้โดยระบบปฏิบัติการ (Operating System)

3) ความผิดพลาดทางตรรกะ (Logical Error) เป็นความผิดพลาดที่ผลการทำงานของโปรแกรมไม่เป็นไปตามจุดมุ่งหมายของผู้เขียน กล่าวคือผลลัพธ์ที่ได้ไม่เป็นไปตามที่กำหนดไว้

จากประเภทของความผิดพลาดข้างต้น ผู้วิจัยเลือกใส่ความผิดพลาดลงไปในซอร์สโค้ดเพียง 2 ประเภทเท่านั้น ได้แก่ ความผิดพลาดขณะดำเนินงานและความผิดพลาดทางตรรกะ เนื่องจากความผิดพลาดทั้ง 2 ประเภทนี้ไม่สามารถตรวจจับด้วยคอมไพเลอร์ ส่งผลให้โปรแกรมสามารถดำเนินงานได้ก่อนดำเนินการทดสอบโปรแกรม และเมื่อนำซอร์สโค้ดที่ใส่ความผิดพลาดไปแล้วมาทดสอบกับกรณีทดสอบพบว่าผลลัพธ์ที่ได้ไม่ตรงกับผลลัพธ์คาดหวังที่ระบุในกรณีทดสอบ

โดยวิธีการทดสอบในส่วนนี้จะเหมือนกับการทดสอบข้างต้นทุกประการ (กรณีที่ผลการทดสอบโปรแกรมเป็นไปตามผลลัพธ์คาดหวัง) ตัวอย่างประเภทความผิดพลาดที่ใส่ในโปรแกรมการซื้อตั๋วสำหรับแต่ละกรณีทดสอบและผลลัพธ์ของการทดสอบโปรแกรมดังกล่าวด้วยชุดกรณีทดสอบที่ถูกสร้างขึ้นจากเครื่องมือแสดงไว้ในตารางที่ 5-5 และ 5-6 ตามลำดับ

ตารางที่ 5-5 แสดงประเภทความผิดพลาดที่ใส่ลงไปในซอร์สโค้ดสำหรับแต่ละกรณีทดสอบของยูสเคสการซื้อตั๋ว

กรณีทดสอบลำดับที่	**ประเภทความผิดพลาด**
1	ความผิดพลาดขณะดำเนินงาน
2	ความผิดพลาดขณะดำเนินงาน
3	ความผิดพลาดทางตรรกะ
4	ความผิดพลาดทางตรรกะ

จากตารางที่ 5-5 ความผิดพลาดที่ใส่ลงไปในซอร์สโค้ดจะถูกใส่ลงไปเพียง 1 ความผิดพลาดต่อ 1 กรณีทดสอบเท่านั้น กล่าวคือ เมื่อต้องการทดสอบโปรแกรมด้วยกรณีทดสอบลำดับที่ 1 ความผิดพลาดสำหรับกรณีทดสอบลำดับที่ 2 3 และ 4 จะไม่ถูกใส่ลงไปในซอร์สโค้ดด้วย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5-6 แสดงชุดกรณีทดสอบของการซื้อตั๋วและผลลัพธ์ที่ได้จากการทดสอบกรณีที่การทดสอบเป็นไม่ไปตามผลลัพธ์คาดหวัง

ลำดับที่	ข้อมูลเบื้องต้น	ข้อมูลทดสอบ	ผลลัพธ์คาดหวัง	ผลการทดสอบ
1	• bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {qty=25, seatBalance=24}	• result = seatBalance • result = bm • result = way • result = “There is not enough seat for your request or maybe the seat is full.”	แสดงข้อความ "NullReferenceException"
2	• bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {customerType=!Y,!N} • {qty=27, seatBalance=27}	• result = seatBalance • result = bm • result = way	แสดงข้อความ "NullReferenceException"

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ลำดับที่	ข้อมูลเบื้องต้น	ข้อมูลทดสอบ	ผลลัพธ์คาดหวัง	ผลการทดสอบ
3	• bm.seatBalance>=qty • bm.getPrice>0 • t.qty>0 • bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {customerType=N} • {qty=27, seatBalance=27}	• t.isDefined() • bm.seatBalance = bm.seatBalance@pre-qty • bm.soldTicket = bm.soldTicket@pre+qty • t.totalPrice = (bm.getPrice*qty) • result = seatBalance • result = bm • result = way	ตรงกับผลลัพธ์คาดหวังยกเว้น • seatBalance มีค่าเท่ากับ 27 (หากตรงตามผลลัพธ์คาดหวัง ควรมีค่าเท่ากับ 0 ซึ่งเป็นผลของ bm.seatBalance = 27-27)
4	• dt.qty>0 • bm.getPrice>0 • bm.seatBalance>=qty • bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm)	• {customerType=Y} • {qty=27, seatBalance=27}	• dt.totalPrice = (bm.getPrice*qty)-(dt.totalPrice*dt.discount) • bm.seatBalance = bm.seatBalance@pre-qty • bm.soldTicket =	ตรงกับผลลัพธ์คาดหวังยกเว้น • totalPrice มีค่าเท่ากับ 5400 (หากตรงตามผลลัพธ์คาดหวัง ควรมีค่าเท่ากับ 4860 ซึ่งเป็นผลของ dt.totalPrice = (200*27)-(5400*0.1))

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ลำดับที่	ข้อมูลเบื้องต้น	ข้อมูลทดสอบ	ผลลัพธ์คาดหวัง	ผลการทดสอบ
	• way.isDefined() • SaleSystem->includes(way)		bm.soldTicket@pre+qty • dt.isDefined() • result = seatBalance • result = bm • result = way	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5.2.2 การเปรียบเทียบกรณีทดสอบที่สร้างจากเครื่องมือเทียบกับกรณีทดสอบที่สร้างจากผู้เชี่ยวชาญ

ในส่วนนี้ผู้วิจัยได้นำกรณีทดสอบที่สร้างจากเครื่องมือมาวิเคราะห์และเปรียบเทียบกับกรณีทดสอบที่ถูกสร้างขึ้นจากผู้เชี่ยวชาญโดยข้อมูลที่เปรียบเทียบได้แก่จำนวนของกรณีทดสอบและจำนวนของข้อมูลภายในกรณีทดสอบ ตัวอย่างของกรณีทดสอบที่สร้างจากเครื่องมือ (TC_{Tool}) และกรณีทดสอบที่สร้างจากผู้เชี่ยวชาญ (TC_{Exp}) ถูกแสดงไว้ในตารางที่ 5-4 และตารางที่ 5-7 ตามลำดับ โดยเอกสารที่ใช้ในการสร้างกรณีทดสอบ TC_{Exp} ได้แก่เอกสารคำอธิบายยูสเคสและซอร์สโค้ด และเอกสารที่ใช้ในการสร้างกรณีทดสอบ TC_{Tool} ได้แก่เอกสารเอกซ์เอ็มแอลของแผนภาพซีเควนซ์และเงื่อนไขก่อนและหลังดำเนินการด้วยภาษาโอซีแอล กรณีทดสอบ TC_{Exp} และ TC_{Tool} ของหน่วยตัวอย่างพร้อมทั้งเอกสารประกอบของการสร้างกรณีทดสอบสามารถดูได้ที่ภาคผนวก ข

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5-7 ตารางแสดงชุดกรณีทดสอบที่สร้างโดยผู้เชี่ยวชาญ (TC_{Exp}) ของยูสเคสการซื้อตั๋ว

No.	Initial Data	Test Data	Expected Result
1	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create object Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement: BusManagement bm = new BusManagement("8.00 a.m.", way1, bus1) bm.seatBalance = 3	chooseWayId = “011” qty = 3 cutomerType = “Y”	• Object DiscountTicket is created • bm.getSeatBalance =0; • bm.soldTickets = 3; • dt.totalPrice = 540 • แสดงรายละเอียดตั๋วออกทางหน้าจอ
2	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create objec Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement: BusManagement bm = new BusManagement("8.00	chooseWayId = “011” qty = 3 customerType = “N”	• Object Ticket is created • bm.getSeatBalance =0; • bm.SoldTickets =3; • dt.totalPrice = 600; • แสดงรายละเอียดตั๋วออกทางหน้าจอ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

No.	Initial Data	Test Data	Expected Result
	a.m.", way1, bus1) bm.seatBalance = 3		
3	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create objec Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement: BusManagement bm = new BusManagement("8.00 a.m.", way1, bus1) bm.seatBalance = 3	chooseWayId = “011” qty = 10 customerType = “N”	Show Message “There is not enough seat for your request or maybe the seat is full”.
4	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create objec Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement:	chooseWayId = “011” qty = 1 customerType = “N”	• object Ticket is created • bm.getSeatBalance =0; • bm.SoldTickets =3; • dt.totalPrice = 200; • แสดงรายละเอียดตั๋วออกทางหน้าจอ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

No.	Initial Data	Test Data	Expected Result
	BusManagement bm = new BusManagement("8.00 a.m.", way1, bus1) bm.seatBalance = 1;		
5	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create objec Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement: BusManagement bm = new BusManagement("8.00 a.m.", way1, bus1) bm.seatBalance = 1;	chooseWayId = “011” qty = 2 customerType = “N”	Show Message “There is not enough seat for your request or maybe the seat is full”.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1. **การเปรียบเทียบจำนวนกรณีทดสอบ** จำนวนของกรณีทดสอบที่สร้างได้จากเครื่องมือและจำนวนของกรณีทดสอบที่ผู้เชี่ยวชาญสร้างในแต่ละยูสเคสนั้นสามารถสรุปได้ดังแสดงในตารางที่ 5-8

ตารางที่ 5-8 แสดงจำนวนกรณีทดสอบที่ได้จากผู้เชี่ยวชาญและเครื่องมือ

จำนวนกรณีทดสอบ / ยูสเคส	TC_{Exp}	TC_{Tool}	$\frac{TC_{Tool}}{TC_{Exp}}$ x100
การซื้อตั๋ว (Buy Ticket)	5	4	80
การเข้าใช้งาน (Login)	9	5	55.56
การยอมรับอินวอยซ์ (Approve Invoice)	6	5	83.34
การโปรโมชันร่วมกับการขาย (Sale Promotion)	4	4	100
การลดราคา (Calculate Discount)	12	10	83.34
การส่งสินค้า (Shipping Order)	4	2	50
รวม	**40**	**30**	**75**

จากตารางที่ 5-8 เมื่อเปรียบเทียบจำนวนของ TC_{Tool} กับ TC_{Exp} ของยูสเคสทั้ง 6 ยูสเคส แล้วพบว่าจำนวนของกรณีทดสอบ TC_{Tool} น้อยกว่า TC_{Exp} โดยภาพรวมจำนวนของกรณีทดสอบที่ได้สร้างได้จากเครื่องมือนั้นสามารถคิดเป็นร้อยละ 75 ของจำนวนของกรณีทดสอบที่ผู้เชี่ยวชาญสร้าง

2. **เปรียบเทียบจำนวนข้อมูลภายในกรณีทดสอบ** ในส่วนนี้ผู้วิจัยได้นำข้อมูลของกรณีทดสอบที่สร้างจากเครื่องมือและกรณีทดสอบที่สร้างจากผู้เชี่ยวชาญมาเปรียบเทียบกันด้วยการจับคู่ระหว่างกรณีทดสอบที่สร้างจากเครื่องมือและกรณีทดสอบที่สร้างจากผู้เชี่ยวชาญโดยพิจารณาจากค่าของข้อมูลทดสอบที่มีค่าเท่ากันหรืออยู่ภายในขอบเขตเดียวกัน ข้อมูลที่นำมาเปรียบเทียบได้แก่ จำนวนข้อมูลเบื้องต้น จำนวนกรณีทดสอบ และจำนวนผลลัพธ์คาดหวัง ซึ่งแทนด้วยสัญลักษณ์ I T และ E ตามลำดับ โดยแสดงไว้ในตารางที่ 5-9

ตารางที่ 5-9 ตารางสรุปจำนวนข้อมูลรวมของกรณีทดสอบที่สร้างจากเครื่องมือ ข้อมูลรวมของกรณีทดสอบที่สร้างจากผู้เชี่ยวชาญ และจำนวนรวมที่มีความหมายเหมือนกันจำแนกตามยูสเคส โดยจับคู่แต่ละกรณีทดสอบที่มีค่าข้อมูลทดสอบอยู่ในขอบเขตเดียวกัน

ยูสเคส	จำนวน $(I+T+E)_{Exp}$	จำนวน $(I+T+E)_{Tool}$	จำนวนข้อมูลที่มีความหมายเหมือนกัน
การซื้อตั๋ว (Buy Ticket)	56	46	27
การยอมรับใบเสร็จ (Approve Invoice)	53	40	21
การโปรโมชันร่วมกับการขาย (Sale Promotion)	23	24	13
การลดราคา (Calculate Discount)	90	66	36
การส่งสินค้า (Shipping Order)	26	23	10
รวม	**248**	**199**	**107**

จากตารางที่ 5-9 สามารถสรุปได้ว่าจำนวนของข้อมูลที่เครื่องมือสามารถหาได้นั้น ตรงกับจำนวนของข้อมูลที่ผู้เชี่ยวชาญได้สร้างขึ้นโดยคิดเป็นร้อยละ 43.15

จากการเปรียบเทียบจำนวนของข้อมูลทดสอบที่มีความหมายเหมือนกันระหว่างกรณีทดสอบที่สร้างโดยผู้เชี่ยวชาญและกรณีทดสอบที่สร้างจากเครื่องมือ พบว่าบางยูสเคสนั้น ไม่มีจำนวนของข้อมูลทดสอบที่มีความหมายตรงกันเลย เช่น ยูสเคสของการเข้าใช้งาน (Login) ดังนั้นในตารางที่ 5-10 จึงไม่นำกรณีทดสอบของยูสเคสนี้มาเปรียบเทียบกัน เหตุที่เป็นเช่นนี้เนื่องมาจากข้อมูลทดสอบที่ได้จากเครื่องมือถูกสร้างขึ้นจากเงื่อนไขในการเรียกใช้งานเมท็อด ในขณะที่ข้อมูลทดสอบจากผู้เชี่ยวชาญจะเป็นค่าของพารามิเตอร์ของเมท็อดแรกเริ่ม ดังนั้นหากพารามิเตอร์ของเมท็อดทุกตัวถูกใช้ในประโยคเงื่อนไขของการเรียกใช้งานเมท็อด จะทำให้ข้อมูลทดสอบที่ถูกสร้างจากเครื่องมือตรงกันกับข้อมูลทดสอบที่ผู้เชี่ยวชาญสร้างทุกจำนวน และในทางกลับกัน หากพารามิเตอร์ของเมท็อดทุกตัวไม่ถูกใช้ในประโยคเงื่อนไขเลย ข้อมูลทดสอบที่ถูกสร้างจากเครื่องมือก็จะไม่ตรงกับข้อมูลทดสอบที่ผู้เชี่ยวชาญสร้างเลย

แต่ในกรณีหลังนั้น ข้อมูลทดสอบที่ได้จากเครื่องมือ สามารถนำไปใช้เป็นข้อมูลเบื้องต้นหรือเป็นแนวทางในการสร้างข้อมูลทดสอบจริงได้ เนื่องจากตัวแปรของที่ถูกนำมาใช้ในประโยค

เงื่อนไข อาจมาจากการนำตัวแปรที่เป็นพารามิเตอร์ไปประมวลผลในเมท็อดและคืนค่ากลับมาเป็นตัวแปรดังกล่าว หรือหากพารามิเตอร์นั้นเป็นอ็อบเจกต์ของคลาสและค่าของตัวแปรที่นำมาใช้นั้นอาจเป็นแอตทริบิวส์ของของอ็อบเจกต์ดังกล่าว เป็นต้น ดังนั้นเมื่อจะนำกรณีทดสอบเหล่านี้ไปใช้งาน ก็จะสามารถทราบได้ว่าจะต้องกำหนดค่าตัวแปรที่เป็นพารามิเตอร์นั้นให้เป็นเท่าไหร่ เพื่อให้สามารถคืนค่ากลับมาได้เท่ากับตัวแปรที่เครื่องมือสามารถสร้างได้ หรือสามารถทราบได้ว่าจะต้องกำหนดค่าของแอตทริบิวส์นั้นเป็นเท่าไหร่ ก่อนมีการใช้งานกรณีทดสอบดังกล่าว เป็นต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

บทนี้สรุปถึงขั้นตอนและหลักการที่ผู้วิจัยได้สร้างขึ้นเพื่อใช้สร้างกรณีทดสอบจากแผนภาพซีเควนซ์ ผลการประเมินเครื่องมือ การนำงานวิจัยไปประยุกต์ใช้ และส่วนสุดท้ายเป็นข้อจำกัดและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้นำเสนอวิธีการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์และโอซีแอล โดยประยุกต์ใช้กราฟควบคุมกระแสไออาร์ซีเอฟจีของ Rountev และคณะ [7] โดยกรณีทดสอบที่สร้างขึ้นจากงานวิจัย ประกอบไปด้วย ข้อมูลเบื้องต้น ข้อมูลทดสอบ และผลลัพธ์คาดหวัง ซึ่งผู้ใช้งานสามารถนำไปใช้ทดสอบโปรแกรมเชิงวัตถุในระดับการทดสอบแบบบูรณาการได้ โดยมีขั้นตอนในการสร้างกรณีทดสอบดังนี้คือ การแปลงแผนภาพซีเควนซ์เป็นกราฟควบคุมกระแสไออาร์ซีเอฟจี การกำหนดเงื่อนไขก่อนและหลังดำเนินการ การสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ การสร้างข้อมูลทดสอบ และการสร้างกรณีทดสอบ ซึ่งมีรายละเอียดโดยสรุปดังต่อไปนี้

6.1.1 ขั้นตอนและหลักการที่ใช้ในการสร้างกรณีทดสอบ

1. การแปลงแผนภาพซีเควนซ์เป็นกราฟควบคุมกระแสไออาร์ซีเอฟจี ในส่วนนี้ผู้วิจัยเพิ่มเติมคอนดิชันโหนด เลเบิลทีและเลเบิลเอฟ ลงไปเป็นองค์ประกอบของไออาร์ซีเอฟจี เพื่อให้รองรับการแปลงจากแผนภาพซีเควนซ์ที่ออกแบบด้วยคอมบายด์แฟรกเมนต์ อัลเทอร์เนทีฟ ออบ-ชันและลูปได้ พร้อมทั้งสร้างหลักการและขั้นตอนในการแมปปิ้ง (Mapping) แผนภาพซีเควนซ์เป็นกราฟควบคุมกระแสไออาร์ซีเอฟจี

2. การกำหนดเงื่อนไขก่อนและหลังดำเนินการ เพื่อให้กรณีทดสอบมีข้อมูลครบถ้วน คือมีข้อมูลเบื้องต้นและผลลัพธ์คาดหวัง ผู้วิจัยจึงกำหนดให้การสร้างกรณีทดสอบของงานวิจัยนี้ต้องมีการกำหนดเงื่อนไขก่อนและหลังดำเนินการของแต่ละเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ด้วยภาษาข้อจำกัดเชิงวัตถุ (โอซีแอล)

3. การสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ ในส่วนนี้ได้นำไออาร์ซีเอฟที่ได้จากขั้นตอนที่ 1 มาสร้างเป็นเส้นทางให้ครบทุกเส้นทางที่ปรากฏในกราฟควบคุมกระแส เหตุที่ทำเช่นนี้เพื่อให้กรณีทดสอบที่สร้างขึ้นเป็นไปตามเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจีที่

Rountev และคณะ [7] ได้นำเสนอไว้ โดยผู้วิจัยได้นำเสนอวิธีในการสร้างชุดเส้นทางไออาร์ซีเอฟจีที่ครอบคลุมตามเกณฑ์ดังกล่าวไว้ในที่นี้ด้วย

4. การสร้างข้อมูลทดสอบ ในส่วนนี้เป็นการนำข้อมูลในเส้นทางไออาร์ซีเอฟจีมาใช้เพื่อสร้างข้อมูลทดสอบ โดยเลือกจากข้อมูลภายในคอนดิชันโหนดที่ปรากฏในเส้นทางมาใช้สร้างค่าของตัวแปร โดยสามารถสร้างข้อมูลได้ 3 ชนิด ได้แก่ ข้อมูลชนิดตัวเลขจำนวนเต็ม ข้อมูลชุดอักขระ และข้อมูลชนิดตรรกะ สำหรับการหาค่าของข้อมูลชนิดตัวเลข ผู้วิจัยประยุกต์ใช้การวิเคราะห์ค่าขอบเขตและบรานช์ฟังก์ชันที่นำเสนอโดย Bogdan [33] มาใช้ในการสร้างค่าของตัวแปร โดยผู้วิจัยได้อธิบายถึงหลักการและวิธีที่ใช้สร้างค่าของข้อมูลทดสอบไว้ในส่วนนี้ด้วย

5. การสร้างกรณีทดสอบ เมื่อทำครบทั้ง 4 ขั้นตอนข้างต้นแล้ว ส่วนนี้เป็นการผนวกข้อมูลทั้งหมดเข้าไว้ด้วยกัน โดยผู้วิจัยจะยึดจากเส้นทางไออาร์ซีเอฟจีเป็นหลัก ผลลัพธ์สุดท้ายที่ได้คือชุดของกรณีทดสอบ ที่ประกอบด้วย ข้อมูลเบื้องต้น ข้อมูลทดสอบ และผลลัพธ์คาดหวัง ที่เป็นไปตามเกณฑ์ความครอบคลุมทุกเส้นทางไออาร์ซีเอฟจี

หลังจากสร้างขั้นตอนและหลักในการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์แล้ว ผู้วิจัยจึงได้พัฒนาเครื่องมือที่ช่วยในการสร้างกรณีทดสอบขึ้น โดยข้อมูลนำเข้าของเครื่องมือเป็นแผนภาพซีเควนซ์ที่อยู่ในรูปแบบของเอกสารเอกซ์เอ็มแอล และเครื่องมือนี้อนุญาตให้ผู้ใช้สามารถกำหนดขอบเขตของข้อมูลทดสอบชนิดตัวเลขจำนวนเต็มในรูปแบบของค่าสูงสุดและค่าต่ำสุดที่เป็นไปได้ เพื่อให้ค่าของข้อมูลทดสอบออกมาใกล้เคียงกับการใช้งานจริง

6.1.2 **ผลการทดสอบเครื่องมือ**

ผลจากการทดสอบเครื่องมือที่พัฒนาขึ้น โดยการใช้หน่วยตัวอย่างทั้ง 6 หน่วย พบว่าเครื่องมือสามารถสร้างกรณีทดสอบจากเอกสารแผนภาพซีเควนซ์ในรูปของเอกสารเอกซ์เอ็มแอลและโอซีแอลออกมาได้ตรงตามหลักการที่นำเสนอไว้ คือ รูปแบบของกราฟควบคุมกระแสไออาร์ซีเอฟจีถูกสร้างขึ้นตามหลักการที่กำหนด สามารถสร้างเส้นทางไออาร์ซีเอฟจีได้ครบทุกเส้นทางที่ปรากฏกราฟควบคุมกระแส จำนวนของกรณีทดสอบเท่ากับจำนวนของเส้นทางไออาร์ซี-เอฟจี และค่าของข้อมูลทดสอบตรงกับแนวคิดที่ได้กำหนดไว้

6.1.3 **ผลการประเมินความสามารถของเครื่องมือ**

จากการประเมินประสิทธิผลของกรณีทดสอบที่สร้างจากเครื่องมือ ด้วยการนำไปทดสอบกับโปรแกรมที่ถูกพัฒนาขึ้นตามเอกสารการออกแบบแผนภาพซีเควนซ์ พบว่ากรณีทดสอบทั้งหมดที่สร้างขึ้นนั้นสามารถนำไปใช้งานได้จริง และได้ผลลัพธ์ตรงตามที่ต้องการทุกกรณี และ

ในทางกลับกันผู้วิจัยได้เพิ่มข้อผิดพลาดลงไปในโปรแกรมส่วนดังกล่าวและเมื่อนำมาทดสอบผลลัพธ์ที่ได้ไม่ตรงตามผลลัพธ์คาดหวังของกรณีทดสอบเช่นกัน จากนั้นผู้วิจัยได้นำกรณีทดสอบเหล่านั้นมาเปรียบเทียบกับกรณีทดสอบที่ถูกสร้างขึ้นโดยผู้เชี่ยวชาญ พบว่า จำนวนกรณีทดสอบที่สร้างโดยเครื่องมือนั้นคิดเป็น 75 เปอร์เซ็นต์ของจำนวนกรณีทดสอบที่ถูกสร้างขึ้นจากผู้เชี่ยวชาญ และจำนวนข้อมูลของกรณีทดสอบที่สร้างขึ้นจากเครื่องมือตรงกับจำนวนข้อมูลของกรณีทดสอบจากผู้เชี่ยวชาญ ซึ่งคิดเป็น 43.15 เปอร์เซ็นต์ของจำนวนข้อมูลของกรณีทดสอบจากผู้เชี่ยวชาญ

6.2 การนำงานวิจัยไปประยุกต์ใช้ (Contribution)

6.2.1 การนำงานวิจัยไปใช้ในเชิงทฤษฎี (Theoretical Contribution)

งานวิจัยนี้เป็นการต่อยอดองค์ความรู้ด้านการทดสอบซอฟต์แวร์ (Software Testing) เนื่องจากผู้วิจัยทบทวนวรรณกรรมในอดีต (Literature Review) แล้วพบว่ายังไม่มีงานวิจัยใดที่นำเสนอวิธีการสร้างกรณีทดสอบในระดับบูรณาการ (Integration Test) ได้อย่างครบถ้วนจากเอกสารการออกแบบแผนภาพยูเอ็มแอล ดังนั้นผู้วิจัยจึงหาแนวทางในการสร้างกรณีทดสอบจากเอกสารแผนภาพซีเควนซ์ด้วยการประยุกต์ใช้กราฟควบคุมกระแสไอาร์ซีเอฟจีที่ Rountev และคณะ [7] สร้างขึ้นเพื่อวัดความครอบคลุมของการทดสอบ และนำภาษาข้อจำกัดเชิงวัตถุ (โอซีแอล) ที่ถูกสร้างขึ้นเพื่อใช้ระบุข้อจำกัดลงในแผนภาพยูเอ็มแอลเข้ามาผนวกและสร้างเป็นข้อมูลภายในกรณีทดสอบ เพื่อเป็นแนวทางให้กับผู้ที่สนใจในเรื่องการสร้างกรณีทดสอบหรือการทดสอบซอฟต์แวร์ ได้นำข้อมูลเหล่านี้ไปใช้ในการศึกษาต่อไป

6.2.2 การนำงานวิจัยไปใช้ในเชิงประยุกต์ (Practical Contribution)

1. ประโยชน์ของการนำเครื่องมือไปใช้

1) บริษัทหรือองค์กรพัฒนาซอฟต์แวร์ สามารถนำเครื่องมือที่สร้างขึ้นในงานวิจัยนี้ไปใช้ในการสร้างกรณีทดสอบได้ โดยที่ผู้สร้างไม่จำเป็นต้องมีประสบการณ์หรือความรู้ทางด้านการทดสอบมาก่อน ซึ่งผลจากการเปรียบเทียบกรณีทดสอบข้างต้น แสดงให้เห็นว่าการนำเครื่องมือนี้ไปใช้จะสามารถลดการสร้างกรณีทดสอบลงได้อย่างน้อย 43.15 เปอร์เซ็นต์ ซึ่งหมายความว่า จะช่วยลดระยะเวลา รวมทั้งค่าใช้จ่ายในขั้นตอนของการออกแบบกรณีทดสอบได้เป็นอย่างมาก

2) เนื่องจากกรณีทดสอบของงานวิจัยนี้สร้างขึ้นโดยยึดตามเกณฑ์ทุกเส้นทางไอาร์ซีเอฟจี ดังนั้นจึงมั่นใจได้ว่า เมื่อนำกรณีทดสอบไปใช้ทดสอบโปรแกรมจึงสามารถรับประกันได้ว่า จะไม่มีเมท็อดใดที่ไม่ถูกใช้งานจากชุดกรณีทดสอบที่สร้างขึ้น

3) เนื่องจากวิทยานิพนธ์นี้ออกแบบให้เครื่องมือสามารถนำออกชุดของกรณีทดสอบในรูปของเอกสารเอกซ์แอลเอส (.xls) ได้ นอกเหนือจากความสะดวกในการนำไปใช้ทดสอบแล้ว ผู้ทดสอบหรือผู้ใช้ยังสามารถนำกรณีทดสอบดังกล่าวไปใช้ในขั้นตอนอื่น ๆ ของกระบวนการพัฒนาซอฟต์แวร์ได้ เช่น ขั้นตอนการทบทวนการทดสอบ (Test Review) เป็นต้น

2. ข้อที่ควรระวังเมื่อนำเครื่องมือไปใช้

1) เนื่องด้วยกรณีทดสอบของงานวิจัยนี้ถูกสร้างขึ้นจากแผนภาพซีเควนซ์ ดังนั้นอาจจะทำให้ข้อมูลทดสอบที่สร้างจากเครื่องมืออาจไม่ครบถ้วน เช่น หากพารามิเตอร์ในเมท็อดที่ปรากฏเป็นเมจเสจแรกเริ่ม ไม่ได้นำไปใช้ในส่วนของเงื่อนไขการตัดสินใจ เครื่องมือจะไม่ได้สามารถสร้างค่าของตัวแปรที่เป็นพารามิเตอร์เหล่านั้นได้ ผู้ใช้จะต้องกำหนดค่าขึ้นเอง โดยสามารถใช้ข้อมูลทดสอบที่เครื่องมือสร้างขึ้นมาเป็นแนวทางในการสร้างได้

2) ในส่วนของข้อมูลเบื้องต้นที่ต้องกำหนดให้กับระบบก่อนนำข้อมูลทดสอบไปใช้ กรณีทดสอบที่สร้างจากเครื่องมือ ไม่สามารถสร้างค่าของแอตทริบิวส์สำหรับอ็อบเจกต์ที่กำหนดให้สร้างขึ้นได้ทั้งหมด ยกเว้นแอตทริบิวส์ที่ปรากฏอยู่ในส่วนของการเรียกใช้เมท็อด ดังนั้นในส่วนนี้ ผู้ที่จะนำกรณีทดสอบนี้ไปใช้ จะต้องกำหนดค่าเหล่านี้ขึ้นเอง โดยให้สอดคล้องกับลักษณะการทำงานของข้อมูลทดสอบ เช่นเดียวกับค่าของข้อมูลทดสอบบางตัวที่เครื่องมือไม่สามารถสร้างให้ได้ก็จะต้องกำหนดค่าเองเช่นกัน

6.3 ข้อจำกัดและข้อเสนอแนะ

งานวิจัยนี้มีข้อจำกัดบางประการ ซึ่งผู้วิจัยขอสรุปและเสนอแนะแนวทาง ดังต่อไปนี้

1. แผนภาพซีเควนซ์ที่สามารถนำมาสร้างกรณีทดสอบได้ด้วยหลักการตามที่เสนอไว้ในงานวิจัยนี้ต้องมีจำนวนเมจเสจเริ่มต้นเพียงหนึ่งเมจเสจเท่านั้น ซึ่งโดยทั่วไปแล้วการออกแบบแผนภาพซีเควนซ์สามารถมีเมจเสจเริ่มต้นได้มากกว่า 1 เมจเสจ ดังนั้นควรปรับกราฟควบคุมกระแสหรือสร้างรูปแบบใหม่เพื่อให้สามารถรองรับการสร้างกรณีทดสอบจากแผนภาพซีเควนซ์ที่มีเมจเสจเริ่มต้นมากกว่า 1 เมจเสจได้

2. ประเภทของคอมบายด์แฟรกเมนต์ที่งานวิจัยนี้รองรับมีเพียง 3 ประเภทเท่านั้นได้แก่ อัลเทอร์เนทีฟ ออปชัน และลูป ดังนั้นควรเพิ่มประเภทของคอมบายด์แฟรกเมนต์เพื่อให้สามารถรองรับเงื่อนไขในการดำเนินงานเมท็อดได้หลายกรณีมากยิ่งขึ้น

3. งานวิจัยนี้ถูกออกแบบให้สามารถรองรับการสร้างข้อมูลทดสอบได้เพียง 3 ชนิดเท่านั้น (ข้อมูลชุดอักขระ ข้อมูลตัวเลขจำนวนเต็ม และข้อมูลชนิกตรรกะ) ดังนั้นควรเพิ่มให้สามารถรองรับชนิดข้อมูลอื่น ๆ ได้เพื่อให้เหมาะสมกับการใช้งานจริงมากยิ่งขึ้น

4. งานวิจัยนี้ได้นำเงื่อนไขในการดำเนินการก่อนและหลังของเมท็อด (โอซีแอล) มาเป็นข้อมูลเบื้องต้นและผลลัพธ์คาดหวัง โดยที่ไม่ได้นำค่าของข้อมูลเหล่านั้นมาพิจารณาว่ามีความสัมพันธ์กับตัวแปรที่นำมาสร้างข้อมูลทดสอบหรือไม่ ดังนั้นเพื่อให้ค่าของข้อมูลทดสอบมีความถูกต้องมากยิ่งขึ้น ควรที่จะนำค่าเหล่านั้นมาพิจารณาด้วย

5. ในส่วนผลลัพธ์คาดหวังของกรณีทดสอบ เครื่องมือที่พัฒนาขึ้นตามแนวคิดของงานวิจัยนี้ไม่ได้ถูกสร้างขึ้นเพื่อรองรับผลลัพธ์ในกรณีของโปรแกรมที่นำมาทดสอบถูกพัฒนาขึ้นโดยไม่สอดคล้องกับข้อกำหนดความต้องการเบื้องต้นหรือกรณีที่โปรแกรมเกิดข้อผิดพลาด (Error) ดังนั้นเพื่อให้ผลลัพธ์คาดหวังของกรณีทดสอบสามารถรองรับกรณีดังกล่าวได้ ผู้ใช้งานสามารถนำแต่ละกรณีทดสอบไปสมมติค่าตรงข้ามที่ไม่เป็นไปตามผลลัพธ์คาดหวังที่ถูกต้องเอง เช่น เกิดการดำเนินการในส่วนชุดคำสั่งจัดการสิ่งผิดปรกติ (Exception Handler) ประเภทมีการหารด้วยศูนย์ (Divide By Zero Exception) หรือมีการอ้างอิงค่าว่าง (Null Reference Exception) เป็นต้น

6. ในส่วนของการทดสอบโปรแกรมกรณีที่ผลลัพธ์ของการทดสอบไม่เป็นไปตามผลลัพธ์คาดหวังของกรณีทดสอบ สำหรับงานวิจัยนี้ ความผิดพลาดที่ใส่ในซอร์สโค้ดเป็นการใส่เพื่อเจาะจงให้เกิดความผิดพลาดสำหรับแต่ละกรณีทดสอบ แต่ในความเป็นจริงของการทดสอบซอฟต์แวร์ ผู้ทดสอบจะไม่สามารถทราบล่วงหน้าได้ว่าจะเกิดข้อผิดพลาดขึ้นด้วยกรณีทดสอบใด ดังนั้นการทดสอบในการตรวจจับข้อผิดพลาดของแต่ละกรณีทดสอบในงานวิจัยนี้ จึงไม่ครอบคลุมการทดสอบซอฟต์แวร์ในเชิงปฏิบัติจริง

7. ข้อมูลที่นำมาใช้สร้างกรณีทดสอบของงานวิจัยนี้ได้มาจากแผนภาพซีเควนซ์และโอซีแอลเท่านั้น ดังนั้นข้อมูลบางส่วนอาจไม่ครบถ้วน เช่น ค่าของแอตทริบิวส์ของอ็อบเจกต์ ที่จะต้องสร้างไว้ก่อนนำข้อมูลทดสอบไปใช้ในการทดสอบ (ส่วนของข้อมูลเบื้องต้น) ดังนั้นเพื่อให้ข้อมูลเหล่านี้ครบถ้วนมากยิ่งขึ้น อาจจะต้องมีการนำแผนภาพยูเอ็มแอลอื่น ๆ เข้ามาผนวกด้วย เช่น แผนภาพคลาสซึ่งมีรายละเอียดของแอตทริบิวส์และรายละเอียดของพารามิเตอร์ของแต่ละเมท็อด เป็นต้น

รายการอ้างอิง

[1] O'Regan, G. 2002. A Practical Approach to Software Quality. New York: Springer-Verlag.

[2] อุมาพร นิลเอวะ. 2549. การเปรียบเทียบเทคนิคการอ่านซอฟต์แวร์ในการตรวจสอบเอกสารแสดงการออกแบบซอฟต์แวร์ที่ออกแบบโดยวิธีการเชิงวัตถุ. วิทยานิพนธ์ปริญญามหาบัณฑิต. สาขาการพัฒนาซอฟต์แวร์ด้านธุรกิจ ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย.

[3] Pressman, R. S. 2005. Software Enginerring A Practitioner's Approach. Sixth Edition. Singapore: McGraw-Hill.

[4] Harrold, M. J. 2000. Testing: a roadmap. In ICSE – The Future of Software Engineering Track : 61-72.

[5] Williams, L. 2004. Testing Overview and Black-Box Testing Techniques. 8 February 2007, Available from: http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf

[6] Abdurazik, A., Offut, J. and Baldini, A. 2004. A Controlled Experimental Evaluation of Test Case Generated from UML Diagrams. Available from: www.ise.gmu.edu/techrep/2004/04_03.pdf

[7] Rountev, A., Kagan, S. and Sawin, J. 2005. Coverage criteria for testing of object interaction in sequence diagram. Fundamental Approaches to Software Engineering, 289-304. Berlin: Springer.

[8] Kansomkeat, S. and Rivepiboon, W. Automated-Generating Test Case Using UML Statechart Diagrams. Proceedings of SAICSIT2003 : 296 – 300

[9] MacGregor, J. D. and Sykes, D. A. 2001. A Practical Guide to Testing Object-Oriented Software. USA: Addison-Wesley.

[10] Tse, T.H. and Xu, Z. 1996. Test Case Generation for Class-Level Object-Oriented Testing. Proceedings of the 9th International Software Quality Week : 4T4.0-4T4.12.

[11] Dennis, A., Wixom, B. H. and Tegarden, D. 2005. System Analysis and Design with UML Version 2.0: An Object-Oriented Approach. Second Edition. USA: John Wiley & Sons.

[12] Andrews, A., France, R., Ghosh, S. and Craig, G. Test Adequacy Criteria for UML Design Models. 2003. Software Testing, Verification and Reliability 13 (2003): 95-127.

[13] Abdurazik, A and Offutt, J. Generating tests from UML specification. Proceedings of the Second International Conference on the Unified Modeling Language (UML'99) : 416-429.

[14] Mcquillan, J. A. and Power, J. F. 2005. A Survey of UML-Based Coverage Criteria for Software Testing. Technical Report Series, National University of Ireland , Maynooth 8 (2005). 18 December 2006, Available from: http://www.minds.nuim.ie/~jmcq/nuim-cs-tr-2005-08.pdf

[15] Bruegge, B. and Dutoit, A. H. 2004. Object-Oriented Software Engineering Using UML, Patterns, and Java. USA: Pearson Education.

[16] Memon, A. M., Soffa, M. L. and Pollack, M. E. 2001. Coverage Criteria for GUI Testing. Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering : 256-267.

[17] IEEE Standard. 1990. IEEE Standard Glossary of Software Engineering Terminology. 20 January 2007, Available from: http://ieeexplore.ieee.org/iel1/2238/4148/00159342.pdf

[18] Akadej, U. 2006. Introduction to Software Testing: Level of Testing. 13 February 2007, Available from: http://www.akadej.com/Course/05026032/STChapter3.pdf

[19] Aumpornsitikul, A. 2006. Software Testing: Software Quality Assurance. [n.p.].

[20] Williams, L. 2004. White-Box Testing Techniques. 8 February 2007, Available from: http://agile.csc.ncsu.edu/SEMaterials/WhiteBox.pdf

[21] Williams, C. E. 1999. Software Testing and the UML. 28 December 206, Available from: www.chillarege.com/fastabstracts/issre99/99119.pdf

[22] Basanieri , F. and Bertolino, A. 2000. A practical approach to UML-based derivation of integration tests. In Proceeding of 4th International Quality Week Europe. Brussels, Belgium.

[23] ชาลี วรกุลพิพัฒน์ และ เทพฤทธิ์ บัณฑิตวัฒนาวงศ์. 2544. UML ภาษามาตรฐานเพื่อผู้พัฒนาซอฟต์แวร์. กรุงเทพมหานคร: ซีเอ็ดยูเคชั่น.

[24] Bell, D. [n.d.]. UML's Sequence Diagram. 10 February 2007, Available from: http://www-128.ibm.com/developerworks/rational/library/3103.html

[25] The Object Management Group. 2004. UML 2.0 Superstructure Specification. 10 February 2007, Available from: http://www.omg.org/docs/ptc/04-10-02.pdf

[26] Pilone, D. and Pitman, N. 2005. UML 2.0 in a Nutshell. First Edition. USA: O'Reilly Media.

[27] Sajayasree, K. K. [n.d.]. Object Constraint Language ?OCL. 9 January 2007, Available from: http://lcm.csa.iisc.ernet.in/soft_arch/OCL.htm

[28] Warmer, J. and Kleppe, A. 2005. Introduction to OCL. 6 January 2007, Available from: http/www.klasse.nl/ocl/index.html

[29] เศรษฐพงศ์ ลีพัหรัตนรักษ์. 2547. วิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.

[30] Pilskalns, O., Andrews, A., Knight, A., Ghosh, S. and France, R. 2006. Testing UML Designs. Information and Software Technology 49 (August 2007) : 892-912.

[31] Doke, E. R., Satzinger, J. W. And Williams, S. R. 2002. Object-Oriented Application Development Using Java. USA: Thomson Learning.

[32] Larman, C. 2005. Applying UML and Patterns: An introduction to object-oriented analysis and design and iterative development. Third Edition. USA: Pearson Education.

[33] Bogdan, K. 1990. Automated Software Test Data Generation. IEEE Transaction on Software Engineering 16 (August 1990): 870-879

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก
เอกสารแสดงการออกแบบซอฟต์แวร์สร้างกรณีทดสอบ

ความต้องการด้านหน้าที่ (Functional Requirement)

ระบบการสร้างกรณีทดสอบนี้จัดทำขึ้นเพื่อให้ผู้ใช้งานสามารถสร้างกรณีทดสอบจากแผนภาพซีเควนซ์และโอซีแอล โดยระบบประกอบด้วยฟังก์ชันการทำงานต่าง ๆ ดังต่อไปนี้

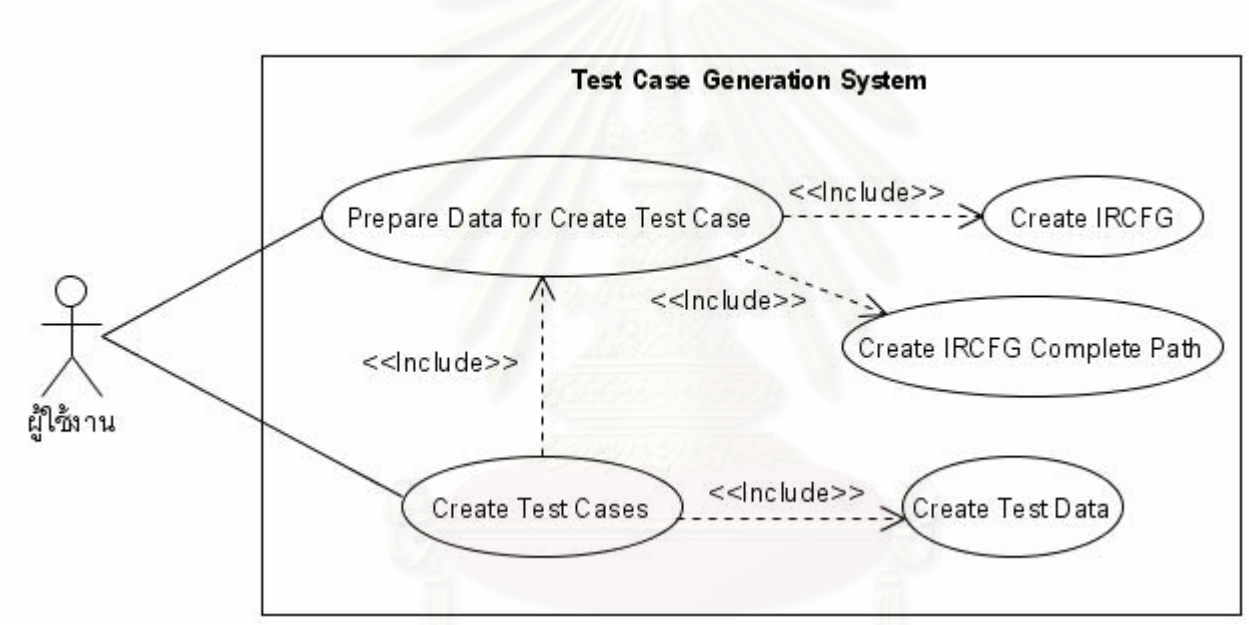
1. ระบบสามารถอ่านข้อมูลแผนภาพซีเควนซ์ในรูปของเอกสารเอกซ์เอ็มแอลและแปลงในอยู่ในรูปของข้อมูลกราฟกระแสไออาร์ซีเอฟจี
2. ระบบสามารถบันทึกข้อจำกัดโอซีแอลในส่วนของเงื่อนไขก่อนและหลังการดำเนินการของแต่ละเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ได้
3. ระบบสามารถสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ครบทุกเส้นทางที่ปรากฏในกราฟกระแสไออาร์ซีเอฟจี
4. ระบบสามารถสร้างข้อมูลทดสอบจากคอนดิชันโหนดในเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ได้
5. ระบบสามารถสร้างกรณีทดสอบและแสดงผลกรณีทดสอบได้
6. ระบบสามารถบันทึกกรณีทดสอบเป็นไฟล์เอกสารนามสกุลเอกซ์แอลเอส (xls.doc) ซึ่งเปิดได้ด้วยโปรแกรมไมโครซอฟต์เอกซ์เซล (Microsoft Excel) ได้

จากความต้องการด้านหน้าที่ในเอกสารข้อกำหนดของความต้องการ (Software Requirement Specification) ข้างต้นสามารถออกแบบซอฟต์แวร์สร้างกรณีทดสอบ ได้ดังนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1. **แผนภาพยูสเคสและคำอธิบายยูสเคส** (Use Case Diagram and Use Case Description)

แผนภาพยูสเคส (Use Case Diagram)



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เอกสารคำอธิบายยูสเคส (Use Case Description)

1. **ยูสเคสเตรียมข้อมูลสำหรับสร้างกรณีทดสอบ** (Prepare Data for Create Test Case)	
เป้าหมาย (Goal)	เพื่อจัดเตรียมข้อมูลเบื้องต้นสำหรับใช้ประกอบการสร้างกรณีทดสอบ
เอกเตอร์ (Actor)	ผู้ใช้งาน
เงื่อนไขก่อน (Precondition)	สร้างเอกสารแผนภาพซีเควนซ์และนำออก (Export) เอกสารแผนภาพซีเควนซ์ที่สร้างด้วยโปรแกรมวิชัวร์พาราแดมในรูปของเอกสารเอกซ์เอ็มแอล (XML Document)
เงื่อนไขหลัง (Postcondition)	• ระบบโหลดเอกสารเอกซ์เอ็มแอลและแสดงรายละเอียดของเอกสารเอกซ์เอ็มแอล • ระบบจัดเก็บข้อมูลเงื่อนไขก่อนและหลังดำเนินการของเมท็อดที่ปรากฏในแผนภาพซีเควนซ์ • ระบบจัดเก็บค่าขอบเขตของตัวแปรที่ปรากฏเป็นข้อจำกัด (Constraint) ภายในตัวถูกดำเนินการ (Operand) ของคอมบายด์แฟรกเมนต์ (Combined Fragment)
เมนซัคเซส ซีนาริโอ (Main Success Scenario)	1. ผู้ใช้งานเปิดเมนูโอเพนท์ไดอาล็อก (Open Dialog) 2. ผู้ใช้งานเลือกเอกสารนามสกุลเอกซ์เอ็มแอล (File.xml) ของแผนภาพซีเควนซ์ที่ต้องการนำมาสร้างกรณีทดสอบ 3. ระบบโหลดข้อมูลจากเอกสารเอกซ์เอ็มแอลและแสดงรายละเอียดของเอกสารเอกซ์เอ็มแอลบนหน้าจอ 4. ระบบสร้างกราฟกระแสไออาร์ซีเอฟจีโดยเรียกใช้ยูสเคสสร้างกราฟกระแสไออาร์ซีเอฟจีและสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์โดยเรียกใช้ยูสเคสสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ 5. ระบบอ่านรายละเอียดของทุกเมจเสจภายในแผนภาพซีเควนซ์ขึ้นมาแสดงบนหน้าจอ 6. ผู้ใช้เลือกเมจเสจที่ต้องการระบุเงื่อนไขก่อนและหลังดำเนินการของเมท็อด

1. **ยูสเคสเตรียมข้อมูลสำหรับสร้างกรณีทดสอบ** (Prepare Data for Create Test Case)	
	7. ระบบแสดงรูปแบบของโอซีแอลคอนเทกซ์ (OCL Context) ตามเมจเสจที่ผู้ใช้งานเลือก 8. ผู้ใช้งานระบุเงื่อนไขก่อนและหลังดำเนินการของเมท็อดที่ปรากฏอยู่ในเมจเสจที่เลือกและบันทึกลงสู่ระบบ 9. ระบบจัดเก็บรายการเงื่อนไขก่อนและหลังดำเนินการของแต่ละเมท็อด 10. ระบบแสดงตารางของตัวแปรที่ปรากฏเป็นข้อจำกัด (Constraint) ภายในตัวถูกดำเนินการ (Operand) ของคอมบายด์แฟรกเมนต์ (Combined Fragment) 11. ผู้ใช้งานกำหนดค่าขอบเขตสูงสุดและต่ำสุดของตัวแปรประเภทข้อมูลตัวเลขจำนวนเต็ม (Integer) 12. ระบบจัดเก็บค่าขอบเขตของตัวแปร
เอกซ์เทนชัน (Extension)	5a. กรณีที่เมจเสจเป็นประเภทสร้างใหม่ (Create New Type) หรือลักษณะของเมจเสจสื่อให้เห็นว่าเป็นเมท็อดที่มีการคืนค่ากลับ เช่น balance=getBalance() เป็นต้น ระบบจะสร้างเงื่อนไขก่อนและหลังดำเนินการของเมท็อดดังกล่าวแบบเบื้องต้นให้โดยอัตโนมัติ 5b. กรณีที่เป็นเมจเสจประเภทรีเทิร์น (Return Type) ระบบจะไม่แสดงรายละเอียดของเมจเสจขึ้นมาแสดง 12a. กรณีผู้ใช้งานไม่ระบุค่าขอบเขตสูงสุด ระบบจะกำหนดให้ค่าสูงสุดต่ำสุดเป็น 32767 และ -32678 ตามลำดับ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2. **ยูสเคสสร้างกรณีทดสอบ** (Create Test Case)	
เป้าหมาย (Goal)	เพื่อสร้างกรณีทดสอบ
เอกเตอร์ (Actor)	ผู้ใช้งาน
เงื่อนไขก่อน (Precondition)	มีการจัดเตรียมข้อมูลสำหรับสร้างกรณีทดสอบแล้ว
เงื่อนไขหลัง (Postcondition)	ระบบแสดงตารางกรณีทดสอบออกทางหน้าจอ
เมนซัคเซส ซีนาริโอ (Main Success Scenario)	1. ผู้ใช้งานเลือกสร้างกรณีทดสอบ 2. ระบบสร้างข้อมูลทดสอบของแต่ละเส้นทางไออาร์ซีเอฟจีโดยเรียกใช้งานยูสเคสสร้างข้อมูลทดสอบ 3. ระบบแสดงตารางชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ 4. ระบบแสดงตารางกรณีทดสอบของแต่ละเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ ซึ่งกรณีทดสอบประกอบไปด้วย หมายเลขเส้นทาง (Path No.) ชื่อของเมท็อด (Method's Name) ออบเจ็กต์ผู้ส่ง (Sending Obj.) ออบเจ็กต์ผู้รับ (Receiveing) ค่าเริ่มต้น (Initial Data) เงื่อนไข (Condition) ข้อมูลทดสอบ (Test Data) และผลลัพธ์คาดหวัง (Expected Result)
เอกซ์เทนชัน (Extension)	4a. กรณีที่ผู้ใช้งานต้องการบันทึกกรณีทดสอบให้กดที่เมนูบันทึก ระบบจะบันทึกข้อมูลตารางกรณีทดสอบโดยจัดเก็บเป็นเอกสารนามสกุลดอทเอกซ์แอลเอส (File.xls)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3. **ยูสเคสสร้างกราฟกระแสไออาร์ซีเอฟจี** (Create IRCFG)	
เป้าหมาย (Goal)	เพื่อสร้างกราฟกระแสควบคุมที่แสดงการเรียกใช้งานระหว่างกันของเมท็อดภายในแผนภาพซีเควนซ์ (Interprocedural Restricted Control Flow Graph -IRCFG)
เอกเตอร์ (Actor)	ผู้ใช้งาน
เงื่อนไขก่อน (Precondition)	ระบบได้โหลดเอกสารเอกซ์เอ็มแอลเข้าสู่ระบบแล้ว
เงื่อนไขหลัง (Postcondition)	ระบบสร้างข้อมูลของกราฟกระแสไออาร์ซีเอฟจี
เมนซัคเซส ซีนาริโอ (Main Success Scenario)	1. ระบบอ่านข้อมูลการเรียกใช้งานเมท็อดจากการส่งและรับเมจเสจของแต่ละไลฟไลน์ในเอกสารเอกซ์เอ็มแอลของแผนภาพซีเควนซ์เพื่อนำมาใช้งานสร้างอาร์ซีเอฟจี และเมท็อดโหนดภายในอาร์ซีเอฟจี 2. ระบบอ่านข้อจำกัดการดำเนินการในแต่ละตัวถูกดำเนินการภายในคอมบายด์แฟรกเมนต์และนำข้อจำกัดการดำเนินการมาสร้างเป็นคอนดิชันโหนดภายในอาร์ซีเอฟจี 3. ระบบสร้างเส้นเชื่อมอาร์ซีเอฟจีที่เชื่อมระหว่างเมท็อดโหนดและคอนดิชันโหนดภายในแต่ละอาร์ซีเอฟจีโดยตรวจสอบตามลำดับการเรียกใช้และเงื่อนไขการเรียกใช้ 4. ระบบสร้างเส้นเชื่อมอินเตอร์โพรซิเดอรัลระหว่างโหนดเริ่มต้นของอาร์ซีเอฟจีกับเมท็อดโหนดที่มีความสัมพันธ์กัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. **ยูสเคสสร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์** (Create IRCFG Complete Path)	
เป้าหมาย (Goal)	เพื่อสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์ของกราฟกระแสไออาร์ซีเอฟจี
เอกเตอร์ (Actor)	ผู้ใช้งาน
เงื่อนไขก่อน (Precondition)	ระบบสร้างกราฟกระแสไออาร์ซีเอฟจีแล้ว
เงื่อนไขหลัง (Postcondition)	ระบบสร้างชุดเส้นทางไออาร์ซีที่สมบูรณ์
เมนซัคเซส ซีนาริโอ (Main Success Scenario)	1. ระบบสร้างชุดเส้นทางอารซีเอฟจี (RCFG Path) ของแต่ละอาร์ซีเอฟจีในกราฟกระแสไออาร์ซีเอฟจี 2. ระบบสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์โดยการผนวกแต่ละเส้นทางอาร์ซีเอฟจีที่สัมพันธ์กันเข้าไว้ด้วยกัน

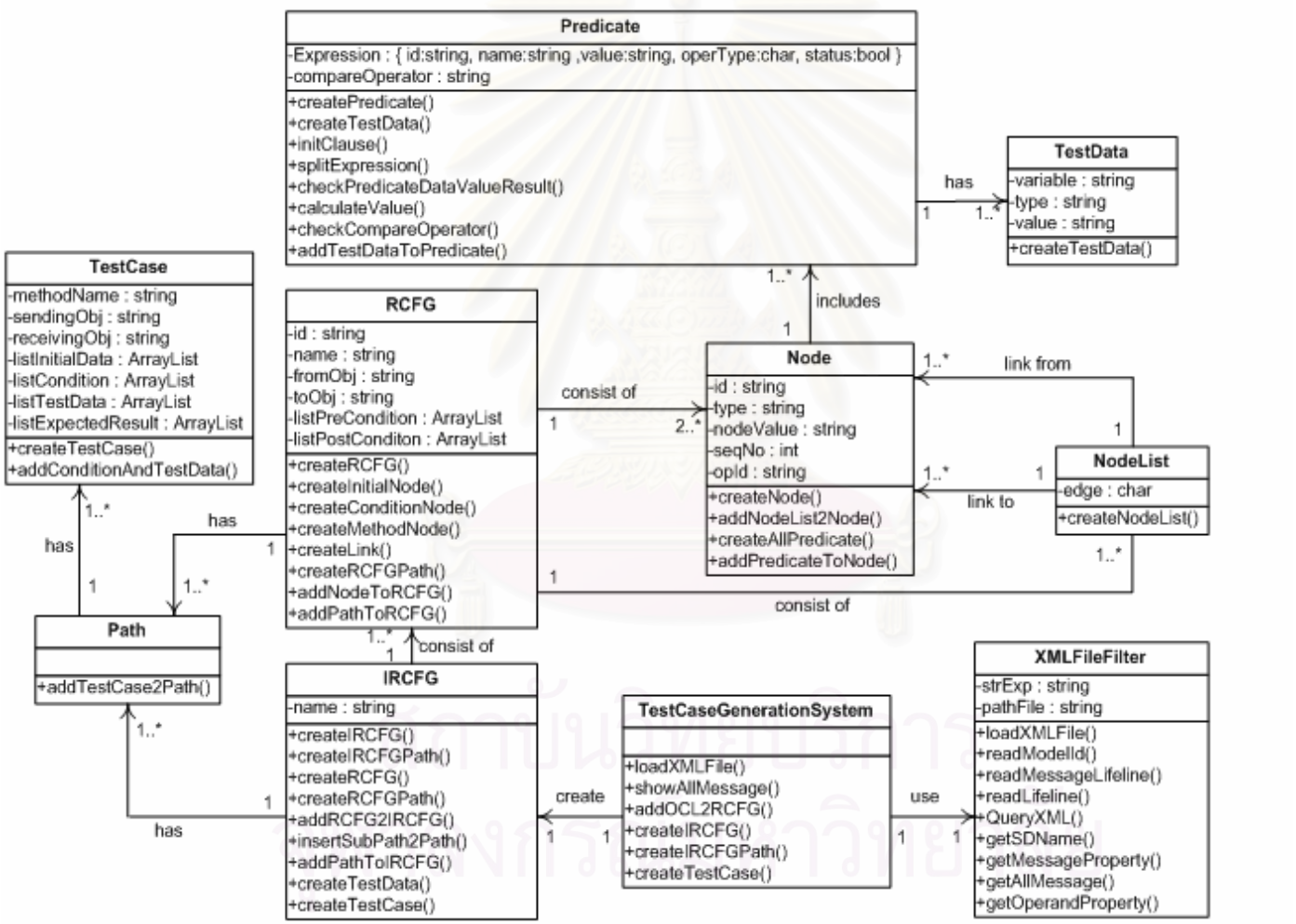
5. **ยูสเคสสร้างข้อมูลทดสอบ** (Create Test Data)	
เป้าหมาย (Goal)	เพื่อสร้างชุดข้อมูลทดสอบจากโหนดเงื่อนไขตามเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์
เอกเตอร์ (Actor)	ผู้ใช้งาน
เงื่อนไขก่อน (Precondition)	ระบบสร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์แล้ว
เงื่อนไขหลัง (Postcondition)	ระบบสร้างชุดข้อมูลทดสอบ
เมนซัคเซส ซีนาริโอ (Main Success Scenario)	1. ระบบเลือกเส้นทางไออาร์ซีเอฟจีที่ได้สร้างไว้แล้ว 2. ระบบเลือกคอนดิชันโหนดจากเส้นทางไออาร์ซีเอฟจีโดยเลือกแบบโพสออร์เดอร์ (Post-Order) 3. ระบบสร้างชุดข้อมูลทดสอบจากตัวแปรของนิพจน์ภายในคอนดิชันโหนด โดยค่าของชุดข้อมูลดังกล่าวที่สร้างจะทำให้ผลลัพธ์ของคอนดิชันโหนดเป็นไปตามลาเบลที่ปรากฏต่อท้ายโหนดเงื่อนไข (T=ผลลัพท์ของคอนดิชันโหนดเป็นจริง, F=ผลลัพท์ของคอนดิชันโหนดเป็นเท็จ) และระบบจะสร้างชุดข้อมูลทดสอบจนครบทุกเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์

5. **ยูสเคสสร้างข้อมูลทดสอบ** (Create Test Data)	
เอกซ์เทนชัน (Extension)	3a. กรณีที่จำนวนตัวแปรที่อยู่ในคอนดิชันโหนดมีจำนวนมากกว่าจำนวนตัวแปรที่ซ้ำกับคอนดิชันโหนดก่อนหน้า(มีการสร้างค่าของตัวแปรไว้ก่อนหน้าแล้ว) จะดึงค่าของตัวแปรดังกล่าวมาใช้ได้เลย 3b. กรณีที่จำนวนตัวแปรที่ซ้ำมีจำนวนเท่ากัน จะไม่สามารถนำค่าของตัวแปรที่สร้างไว้แล้วนั้นมาแทนที่ได้ทันที แต่จะต้องตรวจสอบความสอดคล้องของตัวแปรกับผลลัพธ์ของคอนดิชันโหนดก่อน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2. **แผนภาพคลาสและคำอธิบายคลาส** (Class Diagram and Class Description)

แผนภาพคลาส



เอกสารคำอธิบายคลาส

1. TestCaseGenerationSystem			
Description	เป็นคลาสที่ติดต่อกับ User Interface (Control Class)		
Methods			
1.1 loadXMLFile			
Description	โหลดเอกสารเอกซ์เอ็มแอล		
Visibility	Public		
Parameters	Name	Description	Data Type
	pathFile	ที่อยู่ของเอกสารเอกซ์เอ็มแอล	String
Return Type	String		
1.2 showAllMessage			
Description	แสดงข้อมูลของเมจเสจ		
Visibility	Public		
Return Type	ArrayList		
1.3 addOCL2RCFG			
Description	บันทึกเงื่อนไขก่อนดำเนินการและหลังดำเนินการในแต่ละ RCFG		
Visibility	Public		

Parameters	Name	Description	Data Type
	preCondList	รายการเงื่อนไขก่อนดำเนินการของเมท็อด	ArrayList
	postCondList	รายการเงื่อนไขหลังดำเนินการของเมท็อด	ArrayList
Return Type	void		
1.4 createIRCFG			
Description	สร้างกราฟกระแสไออาร์ซีเอฟจี		
Visibility	Public		
Return Type	IRCFG		
1.5 createIRCFGPath			
Description	สร้างชุดเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์		
Visibility	Public		
Return Type	Path		
1.6 createTestCase			
Description	สร้างกรณีทดสอบ		
Visibility	Public		
Return Type	TestCase		

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2. IRCFG				
Description	กราฟกระแสไออาร์ซีเอฟจี			
Attributes				
Name	Description	Data Type	Initial Value	Visibility
name	ชื่อกราฟกระแสไออาร์ซีเอฟจี (ชื่อของแผนภาพซีเควนซ์)	String	-	private
Methods				
2.1 createIRCFG				
Description	สร้างไออาร์ซีเอฟจี			
Visibility	Public			
Parameters	Name	Description	Data Type	
	xml	ที่อยู่ของเอกสารเอกซ์เอ็มแอล	XMLFileFileter	
	sdName	ชื่อแผนาภาพซีเควนซ์	String	
Return Type	void			
2.2 createIRCFGPath				
Description	สร้างเส้นทางไออาร์ซีเอฟจีที่สมบูรณ์			
Visibility	Public			
Return Type	void			

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2.3 createRCFG			
Description	สร้างอาร์ซีเอฟจี		
Visibility	Public		
Return Type	void		
2.4 createRCFGPath			
Description	สร้างเส้นทางอาร์ซีเอฟจี		
Visibility	Public		
Return Type	void		
2.5 addRCFG2IRCFG			
Description	เพิ่มอาร์ซีเอฟจีลงในกราฟกระแสไออาร์ซีเอฟจี		
Visibility	Public		
Parameters	Name	Description	Data Type
	aRCFG	อ๊อบเจกต์ของคลาส RCFG	RCFG
Return Type	void		
2.6 insertSubPath2Path			
Description	แทรกเส้นทางย่อยลงในเส้นทางอาร์ซีเอฟจี		
Visibility	Public		

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Parameters	Name	Description	Data Type
	mainPath	เส้นทางอาร์ซีเอฟจี	Path
	subPath	เส้นทางอาร์ซีเอฟจีที่จะนำมาแทรก	Path
	preNodeAddId	รหัสของเมท็อดโหนดที่อยู่ในตำแหน่งที่จะนำ subPath มาแทรก	String
Return Type	Path		
2.7 addPathToIRCFG			
Description	เพิ่มเส้นทางให้ไออาร์ซีเอฟจี		
Visibility	Public		
Parameters	Name	Description	Data Type
	aPath	อ๊อบเจกต์ของคลาส Path	Path
Return Type	void		
2.8 createTestData			
Description	สร้างข้อมูลทดสอบ		
Visibility	Public		
Return Type	void		

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.9 createTestCase	
Description	สร้างกรณีทดสอบ
Visibility	Public
Return Type	void

3. RCFG				
Description	อาร์ซีเอฟจีของกราฟกระแสไออาร์ซีเอฟจี			
Attributes				
Name	Description	Data Type	Initial Value	Visibility
id	รหัสของอาร์ซีเอฟจี	String	-	private
name	ชื่อของอาร์ซีเอฟจี	String	-	private
sendingObj	ชื่อของอ๊อบเจกต์หรือคลาสที่ส่งเมจเสจ	String	-	private
receivingObj	ชื่อของอ๊อบเจกต์หรือคลาสที่รับเมจเสจ	String	-	private
listPreCond	รายการเงื่อนไขก่อนดำเนินการ	ArrayList	-	private
listPostCond	รายการเงื่อนไขหลังดำเนินการ	ArrayList	-	private
Methods				
3.1 createRCFG				
Description	สร้างอาร์ซีเอฟจี			
Visibility	Public			

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Parameters	Name	Description	Data Type
	mId	รหัสของเมจเสจ	String
	mName	ชื่อของเมจเสจ	String
	Pre	รายการเงื่อนไขก่อนดำเนินการ	ArrayList
	Post	รายการเงื่อนไขหลังดำเนินการ	ArrayList
	objCall	อ๊อบเจกต์หรือคลาสที่เรียกใช้เมจเสจ	String
	objReceving	อ๊อบเจกต์หรือคลาสที่ส่งเมจเสจ	String
Return Type	void		
3.2 createInitialNode			
Description	สร้างโหนดเริ่มต้น (Start Node) และโหนดสิ้นสุด (End Node)		
Visibility	Public		
Return Type	void		
3.3 createMethodNode			
Description	สร้างเมท็อดโหนดในอาร์ซีเอฟจี		
Visibility	Public		
Return Type	void		

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3.4 createConditionNode	
Description	สร้างคอนดิชันโหนดในอาร์ซีเอฟจี
Visibility	Public
Return Type	void
3.5 createNodeList	
Description	สร้างเส้นเชื่อมอาร์ซีเอฟจี
Visibility	Public
Return Type	void
3.6 createRCFGPath	
Description	สร้างเส้นทางอาร์ซีเอฟจี
Visibility	Public
Return Type	void

4. Node				
Description	โหนดภายในอาร์ซีเอฟจี			
Attributes				
Name	Description	Data Type	Initial Value	Visibility
id	รหัสของโหนด	String	-	private
type	ประเภทของโหนด	String	-	private

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

nodeValue	ข้อมูลที่อยู่ภายในโหนด	String	-	private
seqNo	ลำดับของโหนดที่เป็นตัวแทนของเมท็อด	Int	-	private
opId	รหัสของตัวถูกดำเนินการ	String	-	private

Methods			
4.1 createNode			
Description	สร้างโหนด		
Visibility	Public		
Parameters	Name	Description	Data Type
	id	รหัสของโหนด	String
	type	ประเภทของโหนด (เมท็อดโหนดหรือคอนดิชันโหนด)	String
	value	ค่าของโหนด	String
	no	ลำดับของเมจเสจ	int
	opId	รหัสของตัวถูกดำเนินการที่เมจเสจปรากฏอยู่	String
	r	อาร์ซีเอฟจีของโหนด	RCFG
Return Type	void		
4.2 createAllPredicate			
Description	สร้างทุกซิมเพิลเพรดิเคตสำหรับคอนดิชันโหนด		

Visibility	Public		
Parameters	Name	Description	Data Type
	strClause	ประโยคเงื่อนไขภายในคอนดิชันโหนด	String
Return Type	void		
4.3 addNodeListToNode			
Description	เพิ่มเส้นเชื่อมลงในโหนด		
Visibility	Public		
Parameters	Name	Description	Data Type
	aNodeList	เส้นเชื่อมที่จะเพิ่มให้แก่โหนด	NodeList
Return Type	void		
4.4 addPredicateToNode			
Description	เพิ่มเพรดิเคตให้โหนด		
Visibility	Public		
Parameters	Name	Description	Data Type
	aPredicate	เพรดิเคตของโหนด	Predicate
Return Type	void		

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5. NodeList				
Description	เส้นเชื่อมระหว่างโหนดภายในอาร์ซีเอฟจี			
Attributes				
Name	Description	Data Type	Initial Value	Visibility
currentNode	โหนดหลัก	Node	-	private
edge	ประเภทของเส้นเชื่อม	Char	-	private
neighborNode	โหนดที่เชื่อมกับโหนดหลัก	Node	-	private
Methods				
5.1 createNodeList				
Description	สร้างโหนด			
Visibility	Public			
Parameters	Name	Description	Data Type	
	parent	โหนดหลัก	Node	
	child	โหนดที่เชื่อมกับ parent	Node	
	edge	ประเภทของเส้นเชื่อม	Char	
Return Type	void			

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6. Predicate				
Description	ซิมเพิลเพรดิเคตภายในคอนดิชันโหนด			
Attributes				
Name	Description	Data Type	Initial Value	Visibility
Expression	นิพจน์	Struct	-	private
id	รหัสของนิพจน์	String	-	private
name	ชื่อของนิพจน์หรือตัวแปร	String	-	private
value	ชื่อของนิพจน์หรือตัวแปร	String	-	private
operType	ตัวดำเนินการตัวแปรในนิพจน์	Char	-	private
status	สถานะว่าสามารถแตกต่อไปอีกได้หรือไม่	Boolean	-	private
compareOperator	โอเปอร์เรเตอร์ประเภทความสัมพันธ์	String	-	private
Methods				
6.1 createPredicate				
Description	สร้างซิมเพิลเพรดิเคต			
Visibility	Public			
Parameters	Name	Description		Data Type
	aCondition	ประโยคเงื่อนไขเดี่ยวในคอนดิชันโหนด		String
Return Type	void			

6.2 initClause			
Description	ทำให้ประโยคอยู่ในรูปแบบที่พร้อมจะแยกนิพจน์		
Visibility	Public		
Parameters	Name	Description	Data Type
	strACondition	ประโยคเงื่อนไขเดี่ยว	String
Return Type	String		
6.3 checkCompareOperator			
Description	ทำให้ประโยคอยู่ในรูปแบบที่พร้อมจะแยกนิพจน์		
Visibility	Public		
Parameters	Name	Description	Data Type
	strACondition	ประโยคเงื่อนไขเดี่ยว	String
Return Type	String		
6.4 createTestData			
Description	สร้างข้อมูลทดสอบ		
Visibility	Public		
Parameters	Name	Description	Data Type
	strClause	ประโยคเงื่อนไขที่จะนำมาสร้างข้อมูลทดสอบ	String
Return Type	void		

6.5 splitExpression			
Description	แยกนิพจน์		
Visibility	Public		
Parameters	Name	Description	Data Type
	id	รหัสนิพจน์ที่จะนำมาแตกเป็นนิพจน์ย่อย	String
	stringExp	นิพจน์ที่จะนำมาแตกเป็นนิพจน์ย่อย	String
Return Type	Expression[]		
6.6 checkPredicateDataValueResult			
Description	ตรวจสอบค่าผลลัพธ์ของเพรดิเคตและจัดให้อยู่ในรูปแบบของบรานช์ฟังก์ชัน		
Visibility	Public		
Parameters	Name	Description	Data Type
	exp	นิพจน์	Expression
	op	ตัวดำเนินการประเภทความสัมพันธ์	String
	result	ผลลัพธ์ของเพรดิเคต	Boolean
Return Type	Expression		
6.7 caculateValue			
Description	สร้างค่าของนิพจน์		
Visibility	Public		

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Parameters	Name	Description	Data Type
	operand	ตัวดำเนินการทางคณิตศาสตร์	Char
	x	ค่าที่จะนำมาคำนวณ	Int
	y	ค่าที่จะนำมาคำนวณ	Int
Return Type	Int		
6.8 addTestDataToPredicate			
Description	มอบหมายข้อมูลทดสอบให้กับเพรดิเคต		
Visibility	Public		
Parameters	Name	Description	Data Type
	aTestData	ข้อมูลทดสอบ	TestData
Return Type	void		

7. TestData				
Description	ข้อมูลทดสอบ			
Attributes				
Name	Description	Data Type	Initial Value	Visibility
variable	ชื่อของตัวแปร	String	-	private
type	ประเภทของตัวแปร	String	-	private
value	ค่าของตัวแปร	String	-	private

Methods			
7.1 createTestData			
Description	สร้างข้อมูลทดสอบ		
Visibility	Public		
Parameters	Name	Description	Data Type
	var	ชื่อตัวแปร	String
Return Type	void		

8. Path				
Description	เส้นทาง			
Attributes				
Name	Description	Data Type	Initial Value	Visibility
listNode	รายการของโหนดที่อยู่ในเส้นทาง	ArrayList	-	private
Methods				
8.1 addTestCaseToPath				
Description	มอบหมายกรณีทดสอบให้กับเส้นทาง			
Visibility	Public			

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Parameters	Name	Description	Data Type
	aTestCase	กรณีทดสอบ	TestCase
Return Type	void		

9. TestCase				
Description	กรณีทดสอบ			
Attributes				
Name	Description	Data Type	Initial Value	Visibility
methodName	ชื่อของเมท็อด	String	-	private
sendingObj	ชื่อของอ๊อบเจกต์หรือคลาสที่เรียกใช้งานเมท็อด	String	-	private
receivingObj	ชื่อของอ๊อบเจกต์หรือคลาสที่เป็นเจ้าของเมท็อด	String	-	private
listInitialData	รายการของเงื่อนไขก่อนดำเนินการเมท็อด	ArrayList	-	private
listCondition	รายการของข้อมูลภายในคอนดิชันโหนดที่อยู่ในเส้นทางไอาร์ซีเอฟจีที่สมบูรณ์	ArrayList	-	private
listTestData	รายการของข้อมูลทดสอบที่สร้างจากคอนดิชันโหนด	ArrayList	-	private
listExpectedResult	รายการของเงื่อนไขหลังดำเนินการเมท็อด	ArrayList	-	private

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Methods			
9.1 createTestCase			
Description	สร้างกรณีทดสอบ		
Visibility	Public		
Parameters	Name	Description	Data Type
	method	ชื่อเมท็อด	String
	initial	รายการเงื่อนไขก่อนดำเนินการเมท็อด	ArrayList
	expected	รายการเงื่อนไขหลังดำเนินการเมท็อด	ArrayList
	sendObj	อ๊อบเจกต์หรือคลาสที่เรียกใช้เมท็อด	String
	receiveObj	อ๊อบเจกต์หรือคลาสที่ถูกเรียกใช้เมท็อด	String
Return Type	void		
9.2 addConditionAndTestData			
Description	เพิ่มเงื่อนไขและข้อมูลทดสอบลงในกรณีทดสอบ		
Visibility	Public		
Parameters	Name	Description	Data Type
	aCondition	เงื่อนไขที่อยู่ภายในคอนดิชันโหนด	String
	aTestData	ข้อมูลทดสอบทั้งหมดของคอนดิชันโหนด	String
Return Type	void		

10. XMLFileFilter				
Description	เส้นทาง			
Attributes				
Name	Description	Data Type	Initial Value	Visibility
strExp	ข้อมูลที่ต้องการดึงจากเอกซ์เอ็มแอล	String	-	private
pathFile	ที่อยู่ของเอกสารเอกซ์เอ็มแอล	String	-	private
Methods				
10.1 createXMLFileFilter				
Description	สร้างตัวกรองเอกสารเอกซ์เอ็มแอล			
Visibility	Public			
Parameters	Name	Description	Data Type	
	filename	ชื่อของเอกสารเอกซ์เอ็มแอล	String	
Return Type	void			
10.2 loadXMLFile				
Description	โหลดเอกสารเอกซ์เอ็มแอล			
Visibility	Public			
Return Type	String			
10.3 readModelId				
Description	หารหัสของโมเดล			

Visibility	Public		
Parameters	Name	Description	Data Type
	modelType	ประเภทของโมเดล	String
Return Type	String		
10.4 readMessageLifeline			
Description	หาเมจเสจที่ออกจากไลฟ์ไลน์และดูว่าไปยังไลฟ์ไลน์ใด		
Visibility	Public		
Parameters	Name	Description	Data Type
	lifeline_id	รหัสของไลฟ์ไลน์	String
	min	ค่าของลำดับของเมจเสจที่น้อยที่สุดที่สามารถเลือกได้	Int
	max	ค่าของลำดับของเมจเสจที่มากที่สุดที่สามารถเลือกได้	Int
Return Type	ArrayList		
10.5 QueryXML			
Description	คิวรีข้อมูลจากเอกสารเอกซ์เอ็มแอล		
Visibility	Public		
Parameters	Name	Description	Data Type
	xNav	รหัสของไลฟ์ไลน์	XPathNavigator
	expr	ข้อมูลที่ต้องการดึงจากเอกซ์เอ็มแอล	String

Return Type	XPathNodeIterator
10.6 getSDName	
Description	อ่านชื่อของแผนภาพซีเควนซ์
Visibility	Public
Return Type	String
10.7 getMessageProperty	
Description	อ่านข้อมูลของเมจเสจ
Visibility	Public
Return Type	MessageProperty
10.8 getAllMessage	
Description	อ่านข้อมูลเมจเสจทั้งหมด
Visibility	Public
Return Type	ArrayList
10.9 getOperandProperty	
Description	อ่านข้อมูลของตัวถูกดำเนินการ
Visibility	Public
Return Type	OperandProperty

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Association	
1. create	
Description	ระบบสร้างไออาร์ซีเอฟจี
Association Type	Association
From – To (Class)	TestCaseGenerationSystem – IRCFG
Cardinality	1 : 1
2. consist of	
Description	ไออาร์ซีเอฟจีประกอบไปด้วยอาร์ซีเอฟจี
Association Type	Association
From – To (Class)	IRCFG – RCFG
Cardinality	1 : 1..*
3. consist of	
Description	อาร์ซีเอฟจีประกอบไปด้วยโหนด
Association Type	Association
From – To (Class)	RCFG – Node
Cardinality	1 : 1..*

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. consist of	
Description	อาร์ซีเอฟจีประกอบไปด้วยเส้นเชื่อมอาร์ซีเอฟจี
Association Type	Association
From – To (Class)	RCFG – NodeList
Cardinality	1 : 1..*
5. link from	
Description	เส้นเชื่อมที่เชื่อมจากโหนด
Association Type	Association
From – To (Class)	NodeList – Node
Cardinality	1 : 1..*
6. link to	
Description	เส้นเชื่อมที่เชื่อมไปยังโหนด
Association Type	Association
From – To (Class)	NodeList – Node
Cardinality	1 : 1..*

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

7. includes	
Description	โหนดประกอบด้วยเพรดิเคต
Association Type	Association
From – To (Class)	Node – Predicate
Cardinality	1 : 1..*
8. has	
Description	เพรดิเคตมีข้อมูลทดสอบ
Association Type	Association
From – To (Class)	Predicate – TestData
Cardinality	1 : 1..*
9. has	
Description	อาร์ซีเอฟจีมีเส้นทาง
Association Type	Association
From – To (Class)	RCFG – Path
Cardinality	1 : 1..*

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

10. has	
Description	ไออาร์ซีเอฟจีมีเส้นทาง
Association Type	Association
From – To (Class)	IRCFG – Path
Cardinality	1 : 1..*
11. has	
Description	เส้นทางมีกรณีทดสอบ
Association Type	Association
From – To (Class)	Path – TestCase
Cardinality	1 : 1..*
12. use	
Description	ระบบใช้ตัวกรองเอกซ์เอ็มแอล
Association Type	Association
From – To (Class)	TestCaseGenerationSystem – XMLFileFilter
Cardinality	1 : 1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข
เอกสารประกอบการสร้างกรณีทดสอบ และกรณีทดสอบจากเครื่องมือและผู้เชี่ยวชาญ

รายชื่อของยูสเคสที่ใช้ในการทดสอบและประเมินเครื่องมือ

1. การซื้อตั๋ว (Buy Ticket)
2. การเข้าใช้งาน (Login)
3. การยอมรับอินวอยซ์ (Approve Invoice)
4. การโปรโมชันร่วมกับการขาย (Sale Promotion)
5. การลดราคา (Calculate Discount)
6. การส่งสินค้า (Shipping Order)

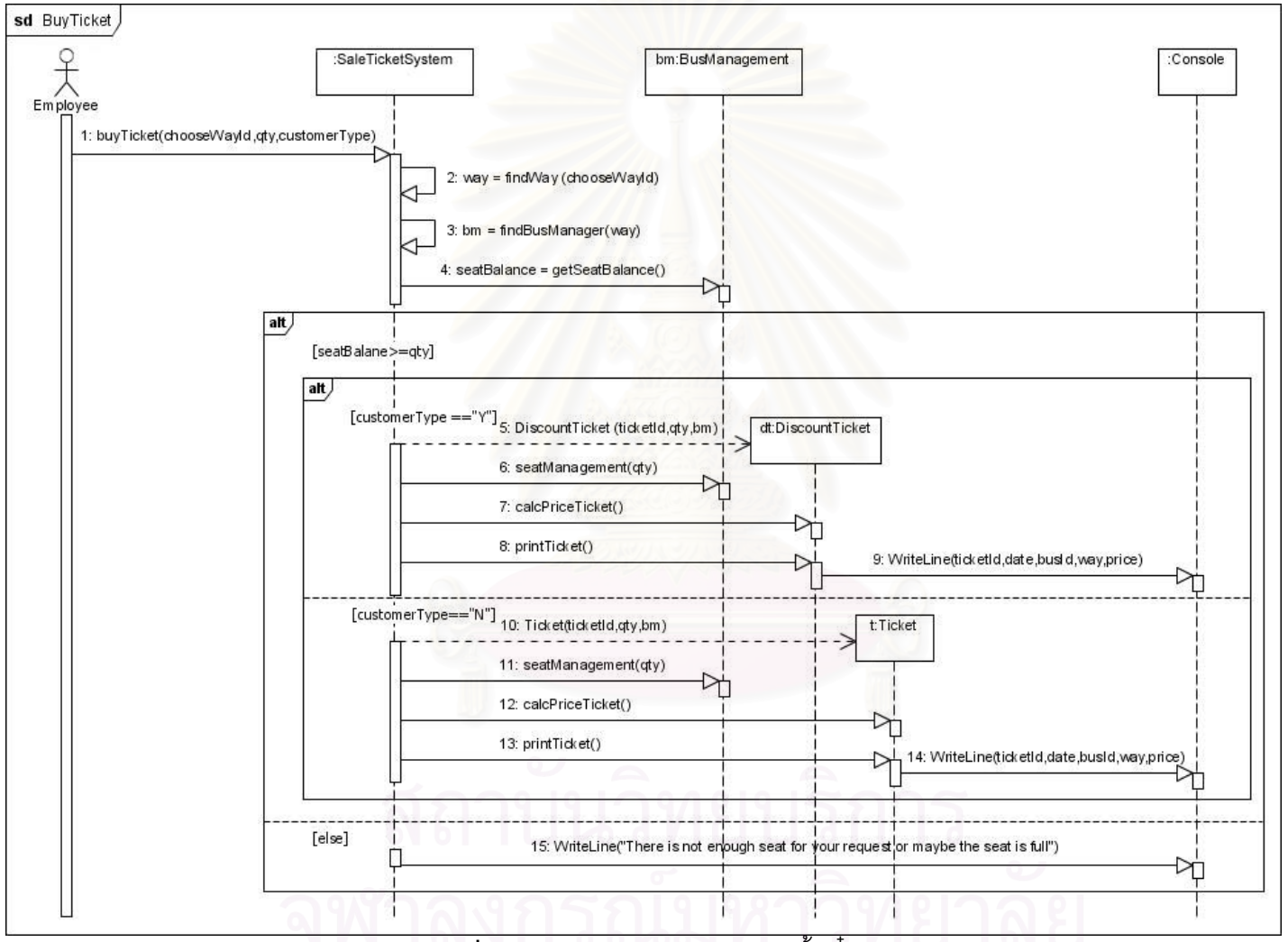
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1. การซื้อตั๋ว

ตารางที่ ข-1 รายละเอียดยูสเคสการซื้อตั๋ว

1. Use Case : BuyTicket	
Goal	บันทึกการซื้อตั๋วโดยสารของลูกค้าและคำนวณราคาพร้อมทั้งแสดงรายละเอียดตั๋ว
Actor	พนักงานขายตั๋ว
Precondition	-
Postcondition	มีการบันทึกข้อมูลการซื้อตั๋ว
Main Success Scenario	1. พนักงานระบุรหัสเส้นทางการเดินทาง จำนวนตั๋วที่ต้องการซื้อ และประเภทของลูกค้า ลงสู่ระบบ 2. กรณีที่ลูกค้าเป็นประเภท “Y” คือเป็นนักเรียนนักศึกษา ระบบจะคำนวณส่วนลดให้ 10% จากราคาตั๋ว กรณีที่ลูกค้าเป็นประเภท “N” คือไม่ใช่นักเรียนนักศึกษา ระบบจะคิดตามราคาตั๋วตามปกติ 3. ระบบแสดงราคารวมพร้อมทั้งแสดงข้อมูลของแต่ละตั๋วออกมาแสดงที่หน้าจอ
Extension	1a. กรณีที่จำนวนตั๋วที่ต้องการซื้อมีปริมาณมากกว่าจำนวนที่นั่งคงเหลือ ระบบจะแจ้งเตือนเพื่อให้ตรวจสอบจำนวนที่นั่งของเส้นทางการเดินรถอีกครั้ง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ข-1 แผนภาพซีเควนซ์ของการซื้อตั๋ว

ตารางที่ ข-2 ตารางแสดงกรณีทดสอบของการซื้อตั๋วที่สร้างโดยผู้เชี่ยวชาญ

No.	Initial Data	Test Data	Expected Result
1	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create object Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement: BusManagement bm = new BusManagement("8.00 a.m.", way1, bus1) bm.seatBalance = 3	chooseWayId = “011” qty = 3 cutomerType = “Y”	• Object DiscountTicket is created • bm.getSeatBalance =0; • bm.soldTickets = 3; • dt.totalPrice = 540 • แสดงรายละเอียดตั๋วออกทางหน้าจอ
2	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create objec Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement: BusManagement bm = new BusManagement("8.00 a.m.", way1, bus1) bm.seatBalance = 3	chooseWayId = “011” qty = 3 customerType = “N”	• Object Ticket is created • bm.getSeatBalance =0; • bm.SoldTickets =3; • dt.totalPrice = 600; • แสดงรายละเอียดตั๋วออกทางหน้าจอ
3	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create objec Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement:	chooseWayId = “011” qty = 10 customerType = “N”	Show Message “There is not enough seat for your request or maybe the seat is full”.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

No.	Initial Data	Test Data	Expected Result
	BusManagement bm = new BusManagement("8.00 a.m.", way1, bus1) bm.seatBalance = 3		
4	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create objec Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement: BusManagement bm = new BusManagement("8.00 a.m.", way1, bus1) bm.seatBalance = 1;	chooseWayId = “011” qty = 1 customerType = “N”	• object Ticket is created • bm.getSeatBalance =0; • bm.SoldTickets =3; • dt.totalPrice = 200; • แสดงรายละเอียดตั๋วออกทางหน้าจอ
5	Create object Way: Way way1 = new Way("011", "Bangkok - Phitsanulok", 200) Create objec Bus: Bus bus1 = new Bus("111", 3) Create object BusManagement: BusManagement bm = new BusManagement("8.00 a.m.", way1, bus1) bm.seatBalance = 1;	chooseWayId = “011” qty = 2 customerType = “N”	Show Message “There is not enough seat for your request or maybe the seat is full”.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-3 ตารางแสดงเส้นทางไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การซื้อตั๋วที่เครื่องมือสร้าง

Path No.	IRCFG Complete Path
1	start, way = findWay (chooseWayId), start, end, bm = findBusManager(way), start, end, seatBalance = getSeatBalance(), start, end, seatBalane>=qty<F>, WriteLine(“There is not enough seat for your request or maybe the seat is full”), start, end, end
2	start, way = findWay (chooseWayId), start, end, bm = findBusManager(way), start, end, seatBalance = getSeatBalance(), start, end, seatBalane>=qty<T>, customerType =="Y"<F>, customerType=="N"<F>, end
3	start, way = findWay (chooseWayId), start, end, bm = findBusManager(way), start, end, seatBalance = getSeatBalance(), start, end, seatBalane>=qty<T>, customerType =="Y"<F>, customerType=="N"<T>, create(ticketId,qty,bm), start, end, seatManagement(qty), start, end, calcPriceTicket(), start, end, printTicket(), start, WriteLine(ticketId, date, busId, way, price), start, end, end, end
4	start, way = findWay (chooseWayId), start, end, bm = findBusManager(way), start, end, seatBalance = getSeatBalance(), start, end, seatBalane>=qty<T>, customerType =="Y"<T>, create (ticketId,qty,bm), start, end, seatManagement(qty), start, end, calcPriceTicket(), start, end, printTicket(), start, WriteLine(ticketId, date, busId, way, price), start, end, end, end

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-4 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การซื้อตั๋วที่เครื่องมือสร้าง

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
1	WriteLine("There is not enough seat for your request or maybe the seat is full")	:SaleTicketSystem	:Console				• result = "There is not enough seat for your request or maybe the seat is full"
	seatBalance = getSeatBalance()	:SaleTicketSystem	bm:BusManagement				• result = seatBalance
	bm = findBusManager(way)	:SaleTicketSystem	:SaleTicketSystem	• bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm)			• result = bm
	way = findWay(chooseWayId)	:SaleTicketSystem	:SaleTicketSystem	• way.isDefined() • SaleSystem->includes(way)			• result = way
	buyTicket(chooseWayId,qty, customerType	Employee	:SaleTicketSystem		• seatBalance>=qty	• {qty=25, seatBalance=24}	
2	seatBalance = getSeatBalance()	:SaleTicketSystem	bm:BusManagement				• result = seatBalance

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	bm = findBusManager(way)	:SaleTicketSystem	:SaleTicketSystem	• bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm)			• result = bm
	way = findWay(chooseWayId)	:SaleTicketSystem	:SaleTicketSystem	• way.isDefined() • SaleSystem->includes(way)			• result = way
	buyTicket(chooseWayId,qty, customerType	Employee	:SaleTicketSystem		• customerType=="N" • customerType=="Y" • seatBalance>=qty	• {customerType=!Y,!N} • {qty=27, seatBalance=27}	
3	WriteLine(ticketId, date, busId, way, price)	:SaleTicketSystem	:Console				
	printTicket()	:SaleTicketSystem	t:Ticket	• t.qty>0			
	calcPriceTicket()	:SaleTicketSystem	t:Ticket	• bm.getPrice>0			• t.totalPrice = (bm.getPrice*qty)
	seatManagement(qty)	:SaleTicketSystem	bm:BusManagement	• bm.seatBalance>=qty			• bm.seatBalance = bm.seatBalance@pre-qty • bm.soldTicket = bm.soldTicket@pre+qty

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	create(ticketId,qty,bm)	:SaleTicketSystem	t:Ticket				• t.isDefined()
	seatBalance = getSeatBalance()	:SaleTicketSystem	bm:BusManagement				• result = seatBalance
	bm = findBusManager(way)	:SaleTicketSystem	:SaleTicketSystem	• bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm)			• result = bm
	way = findWay(chooseWayId)	:SaleTicketSystem	:SaleTicketSystem	• way.isDefined() • SaleSystem->includes(way)			• result = way
	buyTicket(chooseWayId,qty, customerType	Employee	:SaleTicketSystem		• customerType=="N" • customerType=="Y" • seatBalance>=qty	• {customerType=N} • {qty=27, seatBalance=27}	
4	WriteLine(ticketId, date, busId, way, price)	:SaleTicketSystem	:Console				
	printTicket()	:SaleTicketSystem	dt:DiscountTicket	• dt.qty>0			
	calcPriceTicket()	:SaleTicketSystem	dt:DiscountTicket	• bm.getPrice>0			• dt.totalPrice = (bm.getPrice*qty)-(dt.totalPrice*dt.discount)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	seatManagement(qty)	:SaleTicketSystem	bm:BusManagement	• bm.seatBalance>= qty			• bm.seatBalance = bm.seatBalance@pre-qty • bm.soldTicket = bm.soldTicket@pre +qty
	create(ticketId,qty,bm)	:SaleTicketSystem	dt:DiscountTicket				• dt.isDefined()
	seatBalance = getSeatBalance()	:SaleTicket System	bm:BusManagement				• result = seatBalance
	bm = findBusManager(way)	:SaleTicketSystem	:SaleTicketSystem	• bm.isDefined() • bm->includes(way) • SaleSystem-> includes(bm)			• result = bm
	way = findWay(chooseWayId)	:SaleTicketSystem	:SaleTicketSystem	• way.isDefined() • SaleSystem-> includes(way)			• result = way
	buyTicket(chooseWayId,qty, customerType	Employee	:SaleTicketSystem		• customerType=="Y" • seatBalance>=qty	• {customerType=Y} • {qty=27, seatBalance=27}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-5 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การซื้อตั๋วที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)

Path No.	Initial Data	Test Data	Expected Result
1	• bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {qty=25, seatBalance=24}	• result = seatBalance • result = bm • result = way • result = “There is not enough seat for your request or maybe the seat is full.”
2	• bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {customerType=!Y,!N} • {qty=27, seatBalance=27}	• result = seatBalance • result = bm • result = way
3	• bm.seatBalance>=qty • bm.getPrice>0 • t.qty>0 • bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {customerType=N} • {qty=27, seatBalance=27}	• t.isDefined() • bm.seatBalance = bm.seatBalance@pre-qty • bm.soldTicket = bm.soldTicket@pre+qty • t.totalPrice = (bm.getPrice*qty) • result = seatBalance • result = bm • result = way

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Initial Data	Test Data	Expected Result
4	• dt.qty>0 • bm.getPrice>0 • bm.seatBalance>=qty • bm.isDefined() • bm->includes(way) • SaleSystem->includes(bm) • way.isDefined() • SaleSystem->includes(way)	• {customerType=Y} • {qty=27, seatBalance=27}	• dt.totalPrice = (bm.getPrice*qty)- (dt.totalPrice*dt.discount) • bm.seatBalance = bm.seatBalance@pre-qty • bm.soldTicket = bm.soldTicket@pre+qty • dt.isDefined() • result = seatBalance • result = bm • result = way

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

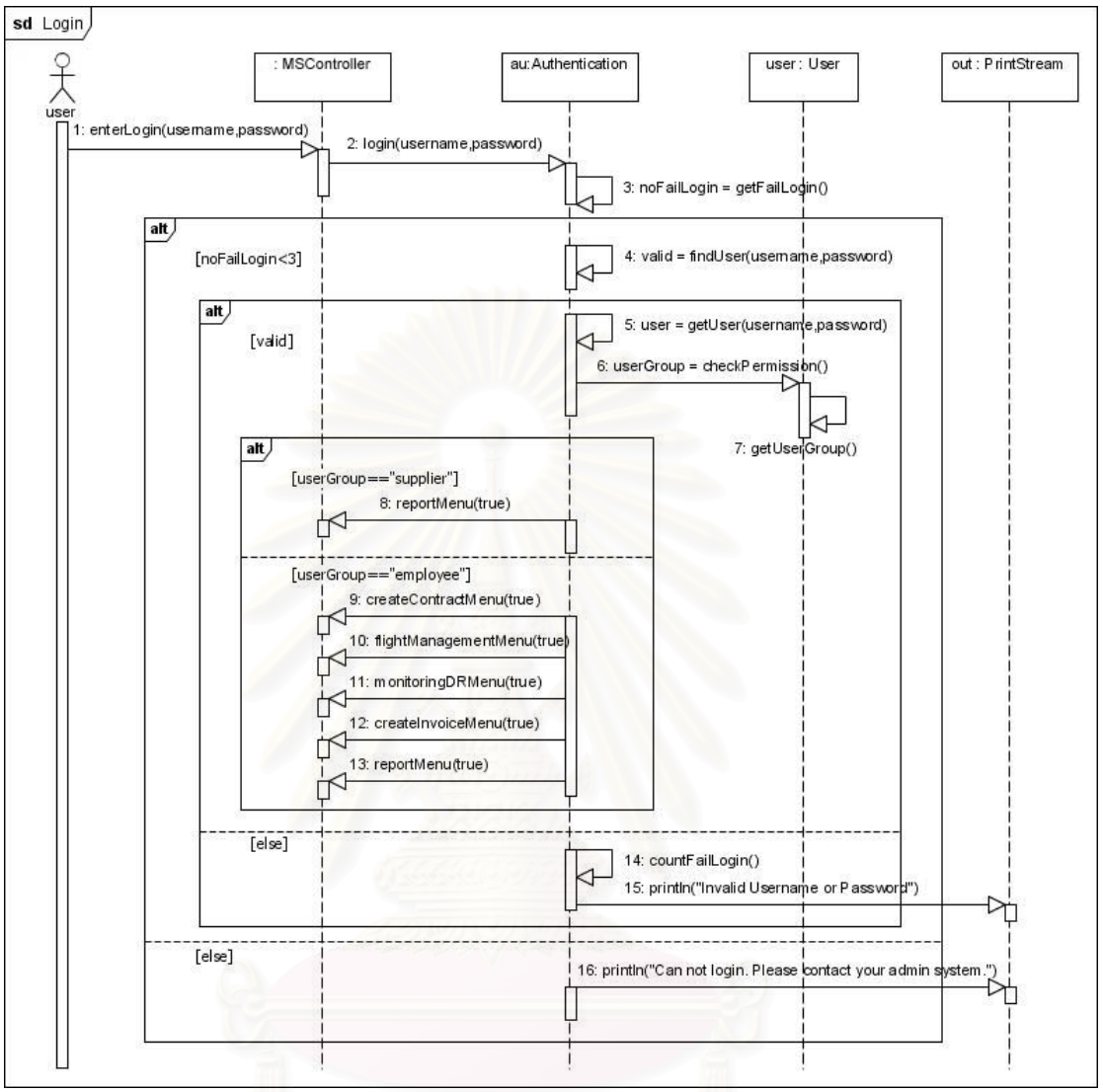
2. การเข้าใช้งาน (Login)

ตารางที่ ข-6 รายละเอียดยูสเคสการเข้าใช้งาน

2. Use Case : Login	
Goal	Login เข้าสู่ระบบและตรวจสอบสิทธิการเข้าใช้งานของ User
Actor	User
Precondition	-
Postcondition	ผู้ใช้ Login เข้าสู่ระบบและระบบแสดงหน้าเมนูตามสิทธิการใช้งาน
Main Success Scenario	1. ผู้ใช้ป้อน Username และ Password ที่ตรงกันกับฐานข้อมูลจะสามารถเข้าสู่หน้าเมนูหลักได้ • กรณีผู้ใช้งานอยู่ใน UserGroup “Supplier” ระบบจะแสดงเฉพาะเมนู Report เพื่อให้ผู้ใช้สามารถเข้าถึงการทำงานในส่วนนี้ได้ • กรณีผู้ใช้งานอยู่ใน UserGroup “Employee” ระบบจะแสดงเมนู Report, Create Contact, Flight Management, Create Invoice เพื่อให้ผู้ใช้สามารถเข้าถึงการทำงานในส่วนนี้ได้
Extension	1a. กรณีที่ Username และ/หรือ Password ไม่ตรงกับข้อมูลในฐานข้อมูล ระบบจะแสดงข้อความเตือนและยังคงอยู่ที่หน้า Login 1b. กรณีที่มีการป้อน Password ซึ่งไม่ตรงกับข้อมูลในฐานข้อมูลเป็นเวลา 3 ครั้งติดต่อกัน User จะไม่สามารถป้อน Username และ Password ได้อีก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

177



รูปที่ ข-2 แผนภาพซีเควนซ์ของการเข้าใช้งาน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-7 ตารางแสดงกรณีทดสอบของการเข้าใช้งานที่สร้างโดยผู้เชี่ยวชาญ

No.	Initial Data	Test Data	Expected Result
1	Create new obj Authentication Authentication au = new Authentication(14/01/2008,11:34) au.countFail = 0 Create new obj User User user = new User("employee1", "abc", "employee")	Username = employee1 Password = abc	• แสดงหน้าจอ Main Menu 1. Contact Management 2. Flight Management 3. Monitoring DR 4. Invoice Advice 5. Report • au.countFail=0
2	Create new obj Authentication Authentication au = new Authentication(14/01/2008,11:34) au.countFail = 0 Create new obj User User user = new User("employee1", "abc", "employee")	Username = employee1 Password = xyz	• แสดงหน้าจอ Login ใหม่ พร้อมข้อความ “Invalid Username or Password ” • au.countFail=1
3	Create new obj Authentication Authentication au = new Authentication(14/01/2008,11:34) au.countFail = 1 Create new obj User User user = new User("employee1", "abc", "employee")	Username = employee1 Password = abc	• แสดงหน้าจอ Main Menu 1. Contact Management 2. Flight Management 3. Monitoring DR 4. Invoice Advice 5. Report • au.countFail=1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

No.	Initial Data	Test Data	Expected Result
4	Create new obj Authentication Authentication au = new Authentication(14/01/2008,11:34) au.countFail = 2 Create new obj User User user = new User("employee1", "abc", "employee")	Username = employee1 Password = abc	• แสดงหน้าจอ Main Menu 1. Contact Management 2. Flight Management 3. Monitoring DR 4. Invoice Advice 5. Report • au.countFail=2
5	Create new obj Authentication Authentication au = new Authentication(14/01/2008,11:34) au.countFail = 2 Create new obj User User user = new User("employee1", "abc", "employee")	Username = employee1 Password = xyz	• แสดงหน้าจอ Login ใหม่ พร้อมข้อความ “Can not login. Please contact your admin system” • au.countFail = 3
6	Create new obj Authentication Authentication au = new Authentication(14/01/2008,11:34) au.countFail = 3 Create new obj User User user = new User("employee1", "abc", "employee")	Username = employee1 Password = xyz	• แสดงหน้าจอ Login ใหม่ พร้อมข้อความ “Can not login. Please contact your admin system” • au.countFail = 3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

No.	Initial Data	Test Data	Expected Result
7	Create new obj Authentication Authentication au = new Authentication(14/01/2008,11:34,0) au.countFail = 0 Create new obj User User user = new User("supplier1", "xyz", "supplier")	Username = supplier1 Password = xyz	แสดงหน้าจอ Main Menu • Report • au.countFail = 0
8	Create new obj Authentication Authentication au = new Authentication(14/01/2008,11:34,1) au.countFail = 1 Create new obj User User user = new User("supplier1", "xyz", "supplier")	Username = supplier1 Password = xyz	แสดงหน้าจอ Main Menu • Report • au.countFail = 1
9	Create new obj Authentication Authentication au = new Authentication(14/01/2008,11:34,2) au.countFail = 2 Create new obj User User user = new User("supplier1", "xyz", "supplier")	Username = supplier1 Password = xyz	แสดงหน้าจอ Main Menu • Report au.countFail = 2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-8 ตารางแสดงเส้นทางไอออาร์ซีเอฟจีของแผนภาพซีเควนซ์การเข้าใช้งานที่เครื่องมือสร้าง

Path No.	IRCFG Complete Path
1	start, login(username,password), start, noFailLogin = getFailLoginCount(), start, end, noFailLogin<3<F>, println("Can not login. Please contact your admin system"), start, end, end, end
2	start, login(username,password), start, noFailLogin = getFailLoginCount(), start, end, noFailLogin<3<T>, valid = findUser(username,password), start, end, valid<F>, countFailLogin(), start, end, println("Invalid Username or Password"), start, end, end, end
3	start, login(username,password), start, noFailLogin = getFailLoginCount(), start, end, noFailLogin<3<T>, valid = findUser(username,password), start, end, valid<T>, user = getUser(username,password), start, end, userGroup = checkPermission(), start, getUserGroup(), start, end, end, userGroup=="supplier"<F>, userGroup =="employee"<F>, end, end
4	start, login(username,password), start, noFailLogin = getFailLoginCount(), start, end, noFailLogin<3<T>, valid = findUser(username,password), start, end, valid<T>, user = getUser(username,password), start, end, userGroup = checkPermission(), start, getUserGroup(), start, end, end, userGroup=="supplier"<F>, userGroup =="employee"<T>, createContractMenu(true), start, end, flightManagementMenu(true), start, end, monitoringDRMenu(true), start, end, createInvoiceMenu(true), start, end, reportMenu(true), start, end, end, end
5	start, login(username,password), start, noFailLogin = getFailLoginCount(), start, end, noFailLogin<3<T>, valid = findUser(username,password), start, end, valid<T>, user = getUser(username,password), start, end, userGroup = checkPermission(), start, getUserGroup(), start, end, end, userGroup=="supplier"<T>, reportMenu(true), start, end, end, end

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-9 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การเข้าใช้งานที่เครื่องมือสร้าง

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
1	println ("Can not login. Please contact your admin system")	au:Authentication	out:PrintStream				• result = "Can not login. Please contact your admin system"
	noFailLogin = getFailLoginCount()	au:Authentication	au:Authentication	• au.isDefined()			• result = au.countFail
	login(username,password)	:MSController	au:Authentication		• noFailLogin<3	• {noFailLogin = 3}	
2	println ("Invalid Username or Password")	au:Authentication	out:PrintStream				• result = "Invalid Username or Password"
	countFailLogin()	au:Authentication	au:Authentication				• au.countFail=au.countFail@pre+1
	valid = findUser(username,password)	au:Authentication	au:Authentication				• result = valid
	noFailLogin = getFailLoginCount()	au:Authentication	au:Authentication	• au.isDefined()			• result = au.countFail
	login(username,password)	:MSController	au:Authentication		• valid • noFailLogin<3	• {valid = false} • {noFailLogin=2}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
3	getUserGroup()	user:User	user:User				
	userGroup = checkPermission()	au:Authentication	user:User				• result = userGroup
	user = getUser(username,password)	au:Authentication	au:Authentication	• user.isDefined() • au->includes(user)			• result = user
	valid = findUser(username,password)	au:Authentication	au:Authentication				• result = valid
	noFailLogin = getFailLoginCount()	au:Authentication	au:Authentication	• au.isDefined()			• result = au.countFail
	login(username,password)	:MSController	au:Authentication		• userGroup==”employee” • userGroup==”supplier” • valid • noFailLogin<3	• {userGroup=!employee,!supplier} • {valid = true} • {noFailLogin=2}	
4	reportMenu(true)	au:Authentication	:MSController	• MSController.reportVisible = false			• MSController.reportVisible = true

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	createInvoiceMenu(true)	au:Authentication	:MSController	• MSController.createInvoiceVisible = false			• MSController.createInvoiceVisible = true
	monitoringDRMenu(true)	au:Authentication	:MSController	• MSController.monitoringDRVisible = false			• MSController.monitoringDRVisible = true
	flightManagementMenu(true)	au:Authentication	:MSController	• MSController.flightManagementVisible = false			• MSController.flightManagementVisible = true
	createContractMunu(true)	au:Authentication	:MSController	• MSController.createContractVisible = false			• MSController.createContractVisible = true
	getUserGroup()	user:User	user:User				
	userGroup = checkPermission()	au:Authentication	user:User				• result = userGroup
	user = getUser(username,password)	au:Authentication	au:Authentication	• user.isDefined() • au->includes(user)			• result = user
	valid = findUser(username,password)	au:Authentication	au:Authentication				• result = valid
	noFailLogin = getFailLoginCount()	au:Authentication	au:Authentication	• au.isDefined()			• result = au.countFail
	login(username,password)	:MSController	au:Authentication		• userGroup=="employee"	• {userGroup=employee}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
					• userGroup=="supplier" • valid • noFailLogin<3	• {valid = true} {noFailLogin=2}	
5	reportMenu(true)	au:Authentication	:MSController	• UI.reportVisible = false			• :MSController.reportVisible = true
	getUserGroup()	user:User	user:User				
	userGroup = checkPermission()	au:Authentication	user:User				• result = userGroup
	user = getUser(username,password)	au:Authentication	au:Authentication	• user.isDefined() • au->includes(user)			• result = user
	valid = findUser(username,password)	au:Authentication	au:Authentication				• result = valid
	noFailLogin = getFailLoginCount()	au:Authentication	au:Authentication	• au.isDefined()			• result = au.countFail
	login(username,password)	:MSController	au:Authentication		• userGroup=="supplier" • valid • noFailLogin<3	• {userGroup=supplier} • {valid = true} {noFailLogin=2}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-10 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การเข้าใช้งานที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)

Path No.	Initial Data	Test Data	Expected Result
1	• au.isDefined()	• {noFailLogin = 3}	• result = "Can not login Please Contact your admin system" • result = au.countFail
2	• au.isDefined()	• {valid = false} • {noFailLogin=2}	• result = “Invalid Username or Password” • au.countFail=au.countFail@pre+1 • result = valid • result = au.countFail
3	• user.isDefined() • au->includes(user) • au.isDefined()	• {userGroup=!employee,!supplier} • {valid = true} • {noFailLogin=2}	• result = userGroup • result = user • result = valid • result = au.countFail

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Initial Data	Test Data	Expected Result
4	• MSController.reportVisible = false • MSController.createInvoiceVisible = false • MSController.monitoringDRVisible = false • MSController.flightManagementVisible = false • MSController.createContractVisible = false • user.isDefined() • au->includes(user) • au.isDefined()	• {userGroup=employee} • {valid = true} • {noFailLogin=2}	• MSController.report Visible = true • MSController.createInvoiceVisible = true • MSController.monitoringDRVisible = true • MSController.flightManagementVisible = true • MSController.createContractVisible = true • result = userGroup • result = user • result = valid • result = au.countFail
5	• MSController.reportVisible = false • user.isDefined() • au->includes(user) • au.isDefined()	• {userGroup=supplier} • {valid = true} • {noFailLogin=2}	• MSController.reportVisible = true • result = userGroup • result = user • result = valid • result = au.countFail

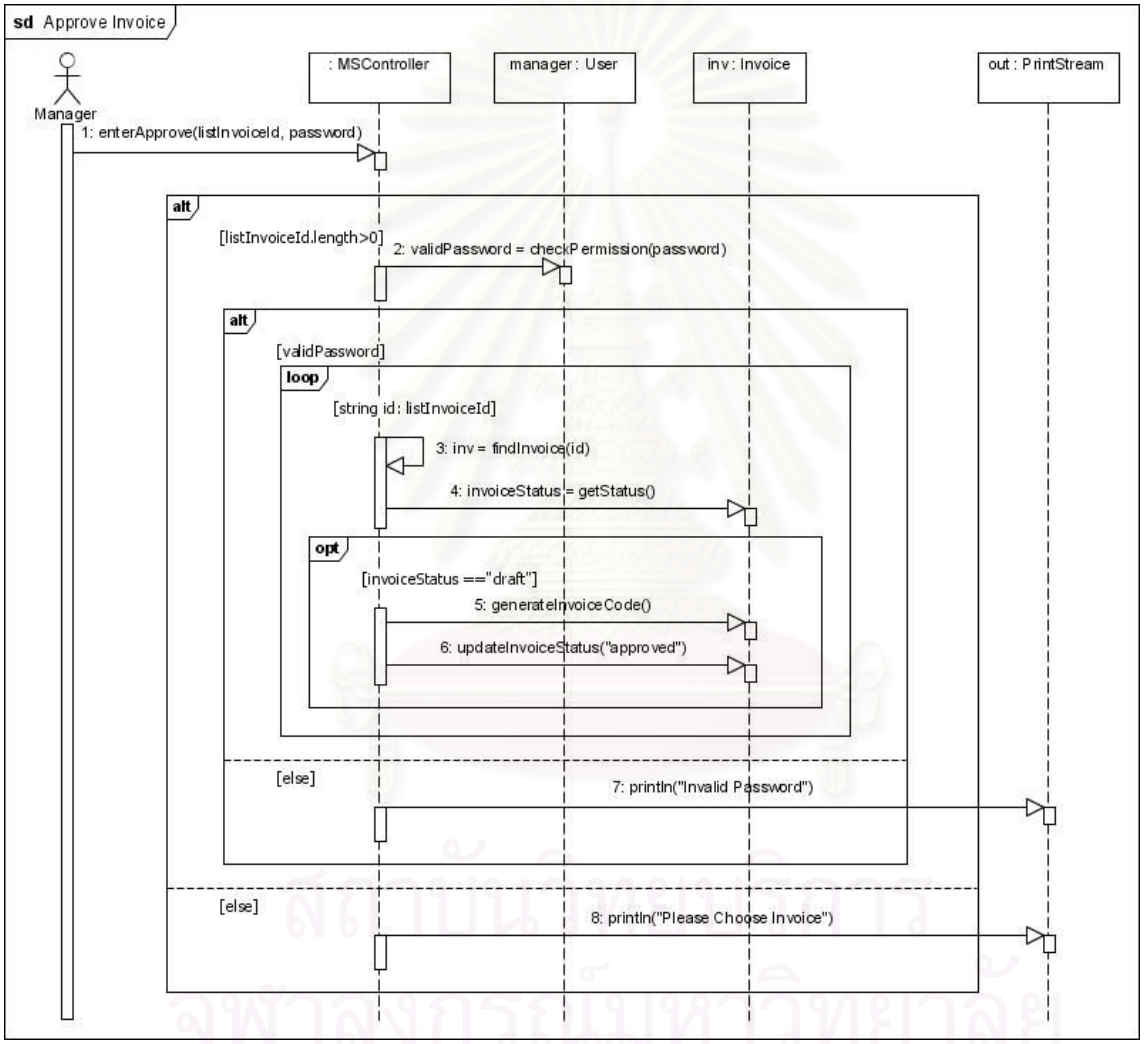
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3. **การยอมรับอินวอยซ์** (Approve Invoice)

ตารางที่ ข-11 รายละเอียดยูสเคสการยอมรับอินวอยซ์

3. Use Case : Approve Invoice	
Goal	สร้างรหัสของ Invoice ผู้ใช้เลือกและสถานะของ Invoice ถูกเปลี่ยนเป็น approved
Actor	manager
Precondition	ผู้ใช้งาน Login เข้าสู่ระบบด้วย Username และ Password ของ UserGroup =”Manager”
Postcondition	สร้างรหัสของ Invoice ผู้ใช้เลือกและสถานะของ Invoice ถูกเปลี่ยนเป็น approved
Main Success Scenario	1. ผู้จัดการเลือกใบ Invoice ที่ต้องการ Approve จากระบบและป้อนรหัส 2. ระบบตรวจสอบความถูกต้องของรหัสผ่าน หากถูกต้อง ระบบจะตรวจสอบสถานะของใบ Invoice ในรายการที่ได้เลือกไว้ หากมีสถานะเป็น “draft” ระบบจะสร้างรหัส Invoice พร้อมทั้งเปลี่ยนสถานะ Invoice ใบนั้นให้เป็น “approve”
Extension	2a. กรณีที่รหัสผ่านไม่ถูกต้องระบบจะแจ้งเตือน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ข-3 แผนภาพซีเควนซ์ของการยอมรับอินวอยซ์

ตารางที่ ข-12 ตารางแสดงกรณีทดสอบของการยอมรับอินวอยซ์ที่สร้างโดยผู้เชี่ยวชาญ

No.	Initial Data	Test Data	Expected Result
1	create user Object • User manager = new User("boss", "abc", "manager",) Login as {username =boss,password=abc}	Password = abc listInvoiceId = {}	• แสดงข้อความ ”Please Choose Invoice”
2	create user Object • User manager = new User("boss", "abc", "manager",) Login as {username =boss,password=abc} Create invoice Object • Invoice inv1 = new Invoice(1, null, "approved") • Invoice inv2 = new Invoice(2, null, "draft") • Invoice inv3 = new Invoice(3, "1/2008", "draft")	Password = xyz listInvoiceId = {1,2,3}	• แสดงข้อความ “Invalid Password”
3	create user Object • User manager = new User("boss", "abc", "manager",) Login as {username =boss,password=abc} Create invoice Object • Invoice inv1 = new Invoice(1, "1/2008", "approved") • Invoice inv2 = new Invoice(2, null, "draft")	Password = abc listInvoiceId = {1,2,3}	inv1. • Id = 1 • Code = 1/2008 • Status = “approved” inv2. • Id = 2 • Code = 2/2008

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

No.	Initial Data	Test Data	Expected Result
	• Invoice inv3 = new Invoice(3, null, "draft")		• Status = “approved” inv3. • Id = 3 • Code = 3/2008 • Status = “approved”
4	create user Object • User manager = new User("boss", "abc", "manager",) Login as {username =boss,password=abc} Create invoice Object • Invoice inv1 = new Invoice(1, "1/2008", "approved") • Invoice inv2 = new Invoice(2, "2/2008", "approved") • Invoice inv3 = new Invoice(3, "3/2008", "approved")	Password = abc listInvoiceId = {1,2,3}	inv1. • Id = 1 • Code = 1/2008 • Status = “approved” inv2. • Id = 2 • Code = 2/2008 • Status = “approved” inv3. • Id = 3 • Code = 3/2008 • Status = “approved”

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

No.	Initial Data	Test Data	Expected Result
5	create user Object • User manager = new User("boss", "abc", "manager",) Login as {username =boss,password=abc} Create invoice Object • Invoice inv1 = new Invoice(1, null, "draft") • Invoice inv2 = new Invoice(2, null, "draft") • Invoice inv3 = new Invoice(3, null, "draft")	Password = abc listInvoiceId = {1}	inv1. • Id = 1 • Code = 1/2008 • Status = “approved” inv2. • Id = 2 • Code = • Status = “draft” inv3. • Id = 3 • Code = • Status = “draft”
6	create user Object • User manager = new User("boss", "abc", "manager",) Login as {username=boss, password=abc} Create invoice Object • Invoice inv1 = new Invoice(1, "1/2008", "approved") • Invoice inv2 = new Invoice(2, "2/2008", "approved") • Invoice inv3 = new Invoice(3, "3/2008", "approved")	Password = abc listInvoiceId = {2}	inv1. • Id = 1 • Code = 1/2008 • Status = “approved” inv2. • Id = 2 • Code = 2/2008 • Status = “approved”

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

No.	Initial Data	Test Data	Expected Result
			inv3. • Id = 3 • Code = 3/2008 • Status = “approved”

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-13 ตารางแสดงเส้นทางไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การยอมรับอินวอยซ์ที่เครื่องมือสร้าง

Path No.	IRCFG Complete Path
1	start, listInvoiceId.length>0<T>, validPassword=checkPermission(password), start, end, validPassword<T>, string id: listInvoiceId <T>, inv=findInvoice(id), start, end, invoiceStatus=getStatus(), start, end, invoiceStatus==”draft”<T>, generateInvoiceCode(), start, end, updateInvoiceStatus(“approved”), start, end, string id: listInvoiceId <F>, end
2	start, listInvoiceId.length >0<T>, validPassword=checkPermission(password), start, end, validPassword<T>, string id: listInvoiceId <T>, inv=findInvoice(id), start, end, invoiceStatus=getStatus(), start, end, invoiceStatus==”draft”<F>, string id: listInvoiceId <F>, end
3	start, listInvoiceId.length>0<T>, validPassword=checkPermission(password), start, end, validPassword<T>, string id: listInvoiceId <F>, end
4	start, listInvoiceId.length>0<T>, validPassword = checkPermission(password), start, end, validPassword<F>, println("Invalid Password"), start, end, end
5	start, listInvoiceId.length>0<F>, println("Please Choose Invoice"), start, end, end

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-14 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การยอมรับอินวอยซ์ที่เครื่องมือสร้าง

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
1	updateInvoiceStatus("approved")	:MSController	inv:Invoice	• inv.Status = "draft"			• inv.Status = "approved"
	generateInvoiceCode()	:MSController	inv:Invoice				• inv.code = inv.id+"/" + DateTime.Year
	invoiceStatus=getStatus()	:MSController	inv:Invoice				• result= inv.Status
	inv = findInvoice(id)	:MSController	:MSController	• inv.isDefined() • inv.Id = id • MSController ->includes(inv)			• result=inv
	validPassword=checkPermission(password),	:MSController	manager:User	• MSController ->includes(manager)			• result=validPassword
	enterApprove(listInvoiceId,password)	Manager	:MSController	• manager.userGroup="manager" • manager.isDefined()	• listInvoiceId.length >0 • validPassword • string id: listInvoiceId • invoiceStatus=="draft"	• {listInvoiceId.Count=1} • {validPassword = true} • { string id: listInvoiceId = true} • {invoiceStatus = draft}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
2	invoiceStatus=getStatus()	:MSController	inv:Invoice				• result= inv.Status
	inv = findInvoice(id)	:MSController	:MSController	• inv.isDefined() • inv.Id = id • MSController ->includes(inv)			• result= inv
	validPassword=checkPermission(password),	:MSController	manager:User	• MSController -> includes (manager)			• result=validPassword
	enterApprove(listInvoiceId,password)	Manager	:MSController	• manager.userGroup="manager" • manager.isDefined()	• listInvoiceId.length>0 • validPassword • string id: listInvoiceId • invoiceStatus=="draft"	• {listInvoiceId.length=1} • {validPassword = true} • { string id: listInvoiceId = true} • {invoiceStatus =! draft}	
3	validPassword=checkPermission(password),	:MSController	manager:User	• MSController ->includes(manager)			• result=validPassword
	enterApprove(listInvoiceId,password)	Manager	:MSController	• manager.userGroup="manager" • manager.isDefined()	• listInvoiceId.length>0 • validPassword • string id: listInvoiceId	• {listInvoiceId.length=1} • {validPassword = true} • { string id: listInvoiceId = false}	

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
4	println ("Invalid Password")	:MSController	out:PrintStream				• result="Invalid Password"
	validPassword=checkPermission(password),	:MSController	manager:User	• MSController->includes(manager)			• result=validPassword
	enterApprove(listInvoiceId,password)	Manager	:MSController	• manager.userGroup="manager" • manager.isDefined()	• listInvoiceId.length>0 • validPassword	• {listInvoiceId.length=1} • {validPassword = false}	
5	println("PleaseChoose Invoice")	:MSController	out:PrintStream				• result="Please Choose Invoice"
	enterApprove(listInvoiceId,password)	Manager	:MSController	• manager.userGroup="manager" • manager.isDefined()	• listInvoiceId.length>0	• {listInvoiceId.length=0}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-15 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การยอมรับอินวอยซ์ที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)

Path No.	Initial Data	Test Data	Expected Result
1	• inv.Status = “draft” • inv.isDefined() • inv.Id = id • MSController->includes(inv) • MSController ->includes(manager) • manager.userGroup=”manager” • manager.isDefined()	• {listInvoiceId.length=1} • {validPassword = true} • {string id: listInvoiceId = true} • {invoiceStatus = draft}	• inv.Status = “approved” • inv.code = inv.id+”/”+DateTime.Year • result= inv.Status • result=inv • result=validPassword
2	• inv.isDefined() • inv.Id = id • MSController ->includes(inv) • MSController ->includes(manager) • manager.userGroup=”manager” • manager.isDefined()	• {listInvoiceId. length =1} • {validPassword = true} • {string id: listInvoiceId = true} • {invoiceStatus =! draft}	• result= inv.Status • result=inv • result=validPassword
3	• MSController ->includes(manager) • manager.userGroup=”manager” • manager.isDefined()	• {listInvoiceId. length =1} • {validPassword = true} • {string id: listInvoiceId = false}	• result=validPassword

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Initial Data	Test Data	Expected Result
4	• MSController ->includes(manager) • manager.userGroup=”manager” • manager.isDefined()	• {listInvoiceId. length =1} • {validPassword = false}	• result=”Invalid Password” • result=validPassword
5	• manager.userGroup=”manager” • manager.isDefined()	• {listInvoiceId. length =0}	• result=”Please Choose Invoice”

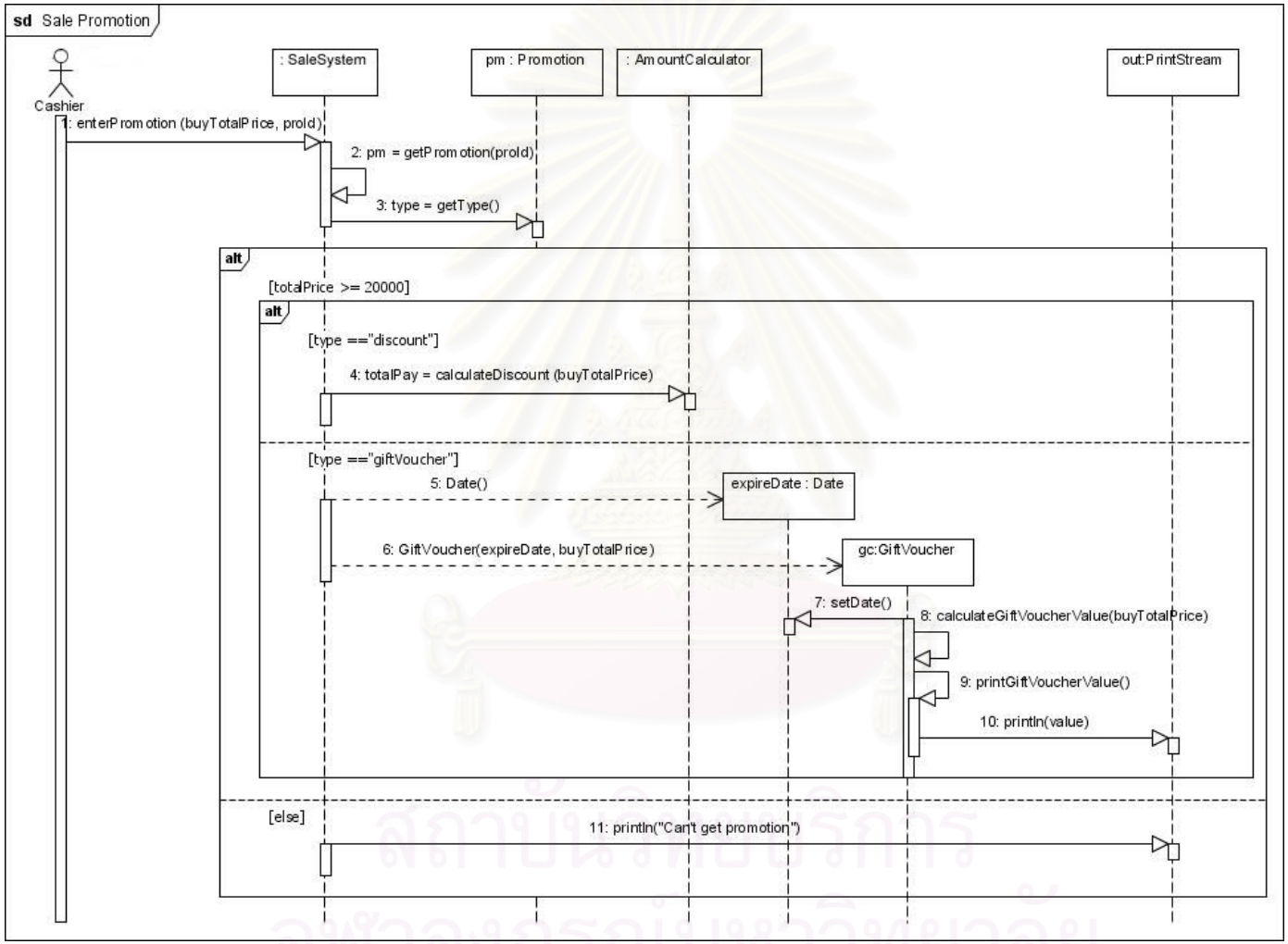
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. การโปรโมชันร่วมกับการขาย (Sale Promotion)

ตารางที่ ข-16 รายละเอียดยูสเคสการโปรโมชันร่วมกับการขาย

4. Use Case : Sale Promotion	
Goal	คำนวณส่วนลดหรือพิมพ์เช็คของขวัญ
Actor	พนักงานขาย
Precondition	-
Postcondition	ระบบคำนวนส่วนลดหรือพิมพ์เช็คของขวัญ
Main Success Scenario	1. พนักงานขายป้อนราคารวมที่ลูกค้าต้องชำระและรหัสของโปรโมชันที่ลูกค้าต้องการ 2. คำนวณส่วนลดตามประเภทโปรโมชันที่ลูกค้าเลือก a. กรณีที่ลูกค้าเลือกโปรโมชันประเภท discount ระบบจะคำนวณส่วนลดให้ 5 %จากราคารวมที่ลูกค้าต้องชำระและแสดงราคารวมที่ลดแล้วทางหน้าจอ b. กรณีที่ลูกค้าเลือกโปรโมชันประเภท gift voucher ระบบจะพิมพ์เช็คของขวัญมูลค่านับเป็น 5% จากราคารวมที่ลูกค้าต้องชำระ ซึ่งสามารถนำไปใช้แทนเงินสดในการซื้อสินค้าในครั้งถัดไปได้ตามระยะเวลาที่ระบุไว้ในเช็คของขวัญ
	1a. กรณียอดการซื้อสินค้าน้อยกว่า 20,000 บาท จะไม่สามารถเลือกใช้โปรโมชันใดได้ ระบบจะเตือนให้ทราบทางหน้าจอ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ข-4 แผนภาพซีเควนซ์ของการโปรโมชันร่วมกับการขาย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-17 ตารางแสดงกรณีทดสอบของการโปรโมชันร่วมกับการขายที่สร้างโดยผู้เชี่ยวชาญ

No.	Initial Data	Test Data	Expected Result
1	Create object Promotion: Promotion pm = new Promotion(“01”,"discount", 0.05)	buyTotalPrice = 30,000 proId = “01”	• TotalPay= 28,500
2	Create object Promotion: Promotion pm = new Promotion(“02”,"giftvoucher", 0.05)	buyTotalPrice = 30,000 proId = “02”	• object GiftVoucher is created • gc.Expire = Today()+30 • gc.Value = 1,500 // value = (.05*buyTotalPrice)
3	Create object Promotion: Promotion pm = new Promotion(“02”,"giftvoucher", 0.05)	buyTotalPrice= 10,000 proId = “02”	Show Message “Can’t get any Promotion”
4	Create object Promotion: Promotion pm = new Promotion(“01”,"discount", 0.05)	buyTotalPrice= 10,000 proId = “01”	Show Message “Can’t get any Promotion”

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-18 ตารางแสดงเส้นทางไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การโปรโมชันร่วมกับการขายที่เครื่องมือสร้าง

Path No.	IRCFG Complete Path
1	start, pm=getPromotion(proId), start, end, type=getType(), start, end, buyTotalPrice>=20000<F>, println(Can't get Promotion), start, end, end
2	start, pm=getPromotion(proId), start, end, type=getType(), start, end, buyTotalPrice>=20000<T>, type==”discount”<F>, type==”giftVoucher”<F>, end
3	start, pm=getPromotion(proId), start, end, type=getType(), start, end, buyTotalPrice>=20000<T>, type==”discount”<F>, type==”giftVoucher”<T>, Date(), start, end, GiftVoucher(expireDate, buyTotalPrice), start, setDate(), start, end, calculateGiftVoucherValue(buyTotalPrice), start, end, printGiftVoucherValue(), start, println(value), start, end, end, end, end
4	start, pm=getPromotion(proId), start, end, type=getType(), start, end, buyTotalPrice>=20000<T>, type==”discount”<T>, totalPay=calculateDiscount(buyTotalPrice), start, end, end

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-19 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การโปรโมชันร่วมกับการขายที่เครื่องมือสร้าง

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
1	println("Can't get Promotion")	:SaleSystem	out:PrintStream				• result="Can't get Promotion"
	type=getType()	:SaleSystem	pm:Promotion	• pm.isDefined()			• result=type
	pm=getPromotion(proId	:SaleSystem	:SaleSystem	• SaleSystem->include(pm)			• result=pm
	enterPromotion(buyTotalPrice,proId)	Cashier	:SaleSystem		• buyTotalPrice=>20000	• {buyTotalPrice=19999}	
2	type=getType()	:SaleSystem	pm:Promotion	• pm.isDefined()			• result=type
	pm=getPromotion(proId	:SaleSystem	:SaleSystem	• SaleSystem->include(pm)			• result=pm
	enterPromotion(buyTotalPrice,proId)	Cashier	:SaleSystem		• buyTotalPrice>=2000 • type==”discount” • type==”giftVoucher”	• {buyTotalPrice=20000} • {type=!discount,!giftVoucher}	
3	println(value)	gc:GiftVoucher	out:PrintStream				
	printGiftVoucherValue()	gc:GiftVoucher	gc:GiftVoucher				

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	calculateGiftVoucherValue(buyTotalPrice)	gc:GiftVoucher	gc:GiftVoucher	• gc.isDefined()			• gc.Value = buyTotalPrice*0.05
	setDate()	gc:GiftVoucher	expireDate:Date				• gc.Expire = dateToday+30
	GiftVoucher(expireDate, buyTotalPrice)	:SaleSystem	gc:GiftVoucher				• gc.isDefined()
	Date()	:SaleSystem	expireDate:Date				• expireDate.isDefined()
	type=getType()	:SaleSystem	pm:Promotion	• pm.isDefined()			• result=type
	pm=getPromotion(proId	:SaleSystem	:SaleSystem	• SaleSystem->include(pm)			• result=pm
	enterPromotion(buyTotalPrice,proId)	Cashier	:SaleSystem		• buyTotalPrice=>2000 • type=="discount" • type=="giftVoucher"	• {buyTotalPrice=20000} • {type=giftVoucher}	
4	totalPay = calculateDiscount(buyTotalPrice)						• totalPay = buyTotalPrice-(buyTotalPrice*.05) • result = totalPay
	type=getType()	:SaleSystem	pm:Promotion	• pm.isDefined()			• result=type

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	pm=getPromotion(proId	:SaleSystem	:SaleSystem	• SaleSystem- • >include(pm)			• result=pm
	enterPromotion(buyTotalPrice,proId)	Cashier	:SaleSystem		• buyTotalPrice=>2000 • type=="discount"	• {buyTotalPrice=20000} • {type=discount}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-20 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การโปรโมชันร่วมกับการขายที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)

Path No.	Initial Data	Test Data	Expected Result
1	• pm.isDefined() • SaleSystem->include(pm)	• {buyTotalPrice=19999}	• result="Can't get Promotion" • result = type • result = pm
2	• pm.isDefined() • SaleSystem->include(pm)	• {buyTotalPrice=2000} • {type=!discount,!giftVoucher" }	• result = type • result = pm
3	• pm.isDefined() • SaleSystem->include(pm)	• {buyTotalPrice=2000} • {type="giftVoucher"}	• gc.isDefined() • gc.Value = buyTotalPrice*0.05 • expireDate.isDefined () • gc.Expire = dateToday+30; • result = type • result = pm
4	• pm.isDefined() • SaleSystem->include(pm)	• {buyTotalPrice=2000} • {type="discount"}	• totalPay = buyTotalPrice-(buyTotalPrice*.05) • result = totalPay • result = type • result = pm

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

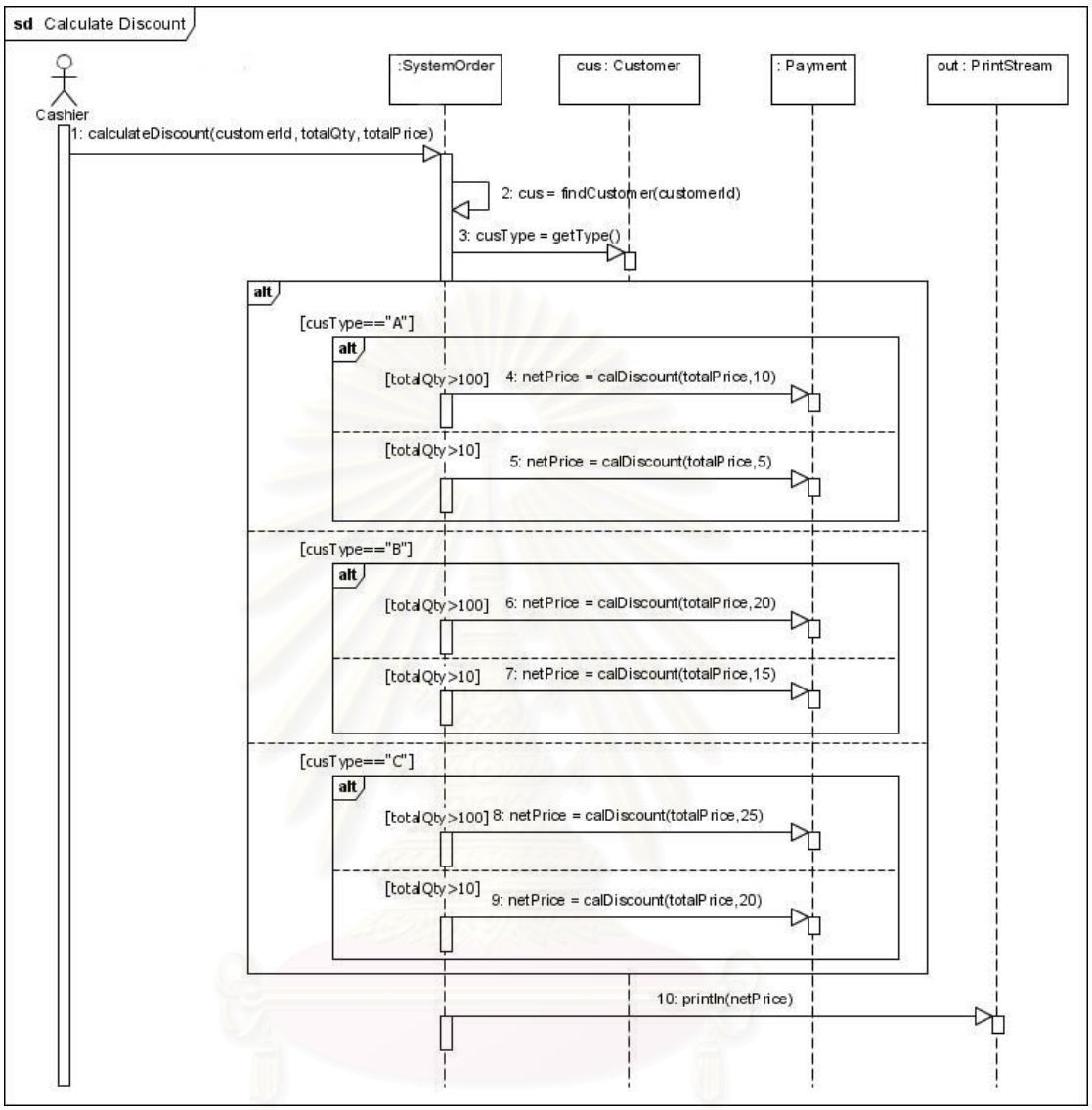
5. **การลดราคา** (Calculate Discount)

ตารางที่ ข-21 รายละเอียดยูสเคสการลดราคา

5. Use Case : Calculate Discount	
Goal	เพื่อคำนวณส่วนลดตามประเภทของลูกค้า
Actor	พนักงานขาย
Precondition	มีการบันทึกข้อมูลการสั่งซื้อและคำนวณราคารวมเรียบร้อยแล้ว
Postcondition	ระบบคำนวนส่วนลดและยอดรวม
Main Success Scenario	1. พนักงานขายป้อนรหัสลูกค้า จำนวนสินค้า และราคารวมของรายการการสั่งซื้อ 2. ระบบค้นหาประเภทลูกค้า 3. ระบบคำนวณส่วนลดและคำนวณยอดเงินที่ลูกค้าต้องชำระ • กรณีเป็นลูกค้าประเภท A และจำนวนสินค้ารวม มากกว่า 10 ชิ้นแต่ไม่เกิน 100 ชิ้น จะคำนวณส่วนลดให้ 5% จากราคาสินค้ารวม และหากจำนวนสินค้ารวมแล้วมากกว่า 100 ชิ้น ระบบจะคำนวณส่วนลดให้ 10% จากราคาสินค้ารวม • กรณีเป็นลูกค้าประเภท B และจำนวนสินค้ารวม มากกว่า 10 ชิ้นแต่ไม่เกิน 100 ชิ้น จะคำนวณส่วนลดให้ 15% จากราคาสินค้ารวม และหากจำนวนสินค้ารวมแล้วมากกว่า 100 ชิ้น ระบบจะคำนวณส่วนลดให้ 20% จากราคาสินค้ารวม • กรณีเป็นลูกค้าประเภท C และจำนวนสินค้ารวม มากกว่า 10 ชิ้นแต่ไม่เกิน 100 ชิ้น จะคำนวณส่วนลดให้ 20% จากราคาสินค้ารวม และหากจำนวนสินค้ารวมแล้วมากกว่า 100 ชิ้น ระบบจะคำนวณส่วนลดให้ 25% จากราคาสินค้ารวม 3. ระบบแสดงยอดเงินที่ลูกค้าต้องชำระ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

209



รูปที่ ข-5 แผนภาพซีเควนซ์ของการลดราคา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-22 ตารางแสดงกรณีทดสอบของการลดราคาที่สร้างโดยผู้เชี่ยวชาญ

No.	Initial Data	Test Data	Expected Result
1	Create object Customer Customer cus1 = new Customer("011", "Anukul", "Prechachan","-","02-111-1111","A")	customerId = 011 totalQty = 11 totalPrice = 4,000	• netPrice = 3,800(ลด 5%)
2	Create object Customer Customer cus1 = new Customer("011", "Anukul", "Prechachan","-","02-111-1111","A")	customerId = 011 totalQty= 200 totalPrice = 80,000	• netPrice = 72,000 (ลด 10%)
3	Create object Customer Customer cus1 = new Customer("011", "Anukul", "Prechachan","-","02-111-1111","A")	customerId = 011 totalQty = 9 totalPrice = 3,000	• netPrice = 3,000
4	Create object Customer Customer cus1 = new Customer("011", "Anukul", "Prechachan","-","02-111-1111","A")	customerId = 011 totalQty = 10 totalPrice = 3,000	• netPrice = 3,000
5	Create object Customer Customer cus1 = new Customer("012", "Wipasree", "Panakitpanit","-","02-111-1111","B")	customerId = 012 totalQty = 11 totalPrice = 4,000	• netPrice = 3,600 (ลด 10%)
6	Create object Customer Customer cus1 = new Customer("012", "Wipasree", "Panakitpanit","-","02-111-1111","B")	customerId = 012 totalQty = 200 totalPrice = 80,000	• netPrice = 68,000 (ลด 15%)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

No.	Initial Data	Test Data	Expected Result
7	Create object Customer Customer cus1 = new Customer("012", "Wipasree", "Panakitpanit","-","02-111-1111","B")	customerId = 012 qty = 9 totalPrice = 3,000	• netPrice = 3,000
8	Create object Customer Customer cus1 = new Customer("012", "Wipasree", "Panakitpanit","-","02-111-1111","B")	customerId = 012 qty = 10 totalPrice = 3,000	• netPrice = 3,000
9	Create object Customer Customer cus1 = new Customer("013", "Wararid", "Tanusin","-","02-111-1111","C")	customerId = 013 totalQty = 11 totalPrice = 4,000	• netPrice = 3,200 (ลด 20%)
10	Create object Customer Customer cus1 = new Customer("013", "Wararid", "Tanusin","-","02-111-1111","C")	customerId = 013 totalQty = 200 totalPrice = 80,000	• netPrice = 60,000 (ลด 25%)
11	Create object Customer Customer cus1 = new Customer("013", "Wararid", "Tanusin","-","02-111-1111","C")	customerId = 013 totalQty = 9 totalPrice = 3,000	• netPrice = 3,000
12	Create object Customer Customer cus1 = new Customer("013", "Wararid", "Tanusin","-","02-111-1111","C")	customerId = 013 totalQty = 10 totalPrice = 3,000	• netPrice = 3,000

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-23 ตารางแสดงเส้นทางไออาร์ซีเอฟจีของแผนภาพซีเควนซ์การลดราคาที่เครื่องมือสร้าง

Path No.	IRCFG Complete Path
1	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<T>, totalQty>100<T>, netPrice=calDiscount(totalPrice,10),start, end, println(netPrice), start, end, end
2	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<T>, totalQty >100<F>, totalQty >10<T>, netPrice=calDiscount(totalPrice,5),start, end, println(netPrice), start, end, end
3	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<T>, totalQty >100<F>, totalQty >10<F>, println(netPrice), start, end, end
4	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<F>, cusType==”B”<T>, totalQty >100<T>, netPrice=calDiscount(totalPrice,20),start, end, println(netPrice), start, end, end
5	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<F>, cusType==”B”<T>, totalQty >100<F>, totalQty >10<T>, netPrice=calDiscount(totalPrice,15),start, end, println(netPrice), start, end, end
6	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<F>, cusType==”B”<T>, totalQty >100<F>, totalQty >10<F>, println(netPrice), start, end, end
7	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<F>, cusType==”B”<F>, cusType==”C”<T>, totalQty >100<T>, netPrice=calDiscount(totalPrice,25),start, end, println(netPrice), start, end, end
8	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<F>, cusType==”B”<F>, cusType==”C”<T>, totalQty >100<F>, totalQty >10<T>, netPrice=calDiscount(totalPrice,20),start, end, println(netPrice), start, end, end
9	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<F>, cusType==”B”<F>, cusType==”C”<T>, totalQty >100<F>, totalQty >10<F>, println(netPrice), start, end, end
10	start, cus = findCustomer(), start, end, cusType=getType() ,start, end, cusType==”A”<F>, cusType==”B”<F>, cusType==”C”<F>, println(netPrice), start, end, end

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-24 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การลดราคาที่เครื่องมือสร้าง

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
1	println(netPrice)	:SaleSystem	out:PrintStream				
	netPrice=calDiscount(totalPrice,10)	:SaleSystem	:Payment				• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result = netPrice
	cusType=getType()	:SaleSystem	cus:Customer				• result= cusType
	cus = findCustomer(customerId)	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus
	calcuateDiscount(customerId,totalQty,totalPrice)	Cashier	:SaleSystem		• cusType==”A” • totalQty>100	• {cusType=”A”} • {totalQty=101}	
2	println(netPrice)	:SaleSystem	out:PrintStream				
	netPrice=calDiscount(totalPrice,5)	:SaleSystem	:Payment				• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result = netPrice
	cusType=getType()	:SaleSystem	cus:Customer				• result= cusType
	cus = findCustomer()	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	calcuateDiscount(customerId,totalQty,totalPrice)	Cashier	:SaleSystem		• cusType==”A” • totalQty>100 • totalQty>10	• {cusType=”A”} • {totalQty=11}	
3	println(netPrice)	:SaleSystem	out:PrintStream				
	cusType=getType()	:SaleSystem	cus:Customer				• result= cusType
	cus = findCustomer()	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus
	calcuateDiscount(customerId,totalQty,totalPrice)	Cashier	:SaleSystem		• cusType==”A” • totalQty>100 • totalQty>10	• {cusType=”A”} • {totalQty=10}	
4	println(netPrice)	:SaleSystem	out:PrintStream				
	netPrice=calDiscount(totalPrice,20)	:SaleSystem	:Payment				• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result = netPrice
	cusType=getType()	:SaleSystem	cus:Customer				• result= cusType
	cus = findCustomer(customerId)	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	calcuateDiscount(customerId,totalQty,totalPrice)	:SaleSystem	Cashier		• cusType==”A” • cusType==”B” • totalQty>100	• {cusType=”B”} • {totalQty=101}	
5	println(netPrice)	:SaleSystem	out:PrintStream				
	netPrice=calDiscount(totalPrice,15)	:SaleSystem	:Payment				• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result= netPrice
	cusType=getType()	:SaleSystem	cus:Customer				• result= cusType
	cus = findCustomer(customerId)	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus
	calcuateDiscount(customerId,totalQty,totalPrice)	Cashier	:SaleSystem		• cusType==”A” • cusType==”B” • totalQty>100 • totalQty>10	• {cusType=”B”} • {totalQty=11}	
6	println(netPrice)	:SaleSystem	out:PrintStream				
	cusType=getType()	:SaleSystem	cus:Customer				• result= cusType
	cus = findCustomer(customerId)	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	calcuateDiscount(customerId,totalQty,totalPrice)	Cashier	:SaleSystem		• cusType==”A” • cusType==”B” • totalQty>100 • totalQty>10	• {cusType=”B”} • {totalQty=10}	
7	println(netPrice)	:SaleSystem	out:PrintStream				
	netPrice=calDiscount(totalPrice,25)	:SaleSystem	:Payment				• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result = netPrice
	cusType=getType()	:SaleSystem	cus:Customer				• result= cusType
	cus = findCustomer(customerId)	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus
	calcuateDiscount(customerId,totalQty,totalPrice)	Cashier	:SaleSystem		• cusType==”A” • cusType==”B” • cusType==”C” • totalQty>100	• {cusType=”C”} • {totalQty=101}	
8	println(netPrice)	:SaleSystem	out:PrintStream				
	netPrice=calDiscount(totalPrice,20)	:SaleSystem	:Payment				• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result= netPrice

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
	cusType=getType()	:SaleSystem	cus:Customer				• result= cusType
	cus = findCustomer(customerId)	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus
	calcuateDiscount(customerId,totalQty,totalPrice)	Cashier	:SaleSystem		• cusType==”A” • cusType==”B” • cusType==”C” • totalQty>100 • totalQty>10	• {cusType=”C”} • {totalQty=11}	
9	println(netPrice)	:SaleSystem	out:PrintStream				
	cusType=getType()	:SaleSystem	cus:Customer				• result= cusType
	cus = findCustomer(customerId)	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus
	calcuateDiscount(customerId,totalQty,totalPrice)	Cashier	:SaleSystem		• cusType==”A” • cusType==”B” • cusType==”C” • totalQty>100 • totalQty>10	• {cusType=”C”} • {totalQty=10}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
10	println(netPrice)	:SaleSystem	out:PrintStream				
	cusType=getType()	:SaleSystem	cus:Customer				• result=cusType
	cus = findCustomer(customerId)	:SaleSystem	:SaleSystem	• cus.isDefined() • SaleSystem->includes(cus)			• result=cus
	calcuateDiscount(customerId,totalQty,totalPrice)	Cashier	:SaleSystem		• cusType=="A" • cusType=="B" • cusType=="C"	• {cusType=!A,!B,!C}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-25 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การลดราคาที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)

Path No.	Initial Data	Test Data	Expected Result
1	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType=”A”} • {totalQty=101}	• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result netPrice • result= cusType • result=cus
2	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType=”A”} • {totalQty=11}	• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result netPrice • result= cusType • result=cus
3	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType=”A”} • {totalQty=10}	• result= cusType • result=cus
4	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType=”B”} • {totalQty=101}	• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result netPrice • result= cusType • result=cus

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Initial Data	Test Data	Expected Result
5	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType="B"} • {totalQty=11}	• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result netPrice • result= cusType • result=cus
6	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType="B"} • {totalQty=10}	• result= cusType • result=cus
7	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType="C"} • {totalQty=101}	• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result netPrice • result= cusType • result=cus
8	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType="C"} • {totalQty=11}	• netPrice = totalPrice-(totalPrice*(discountPercent/100)) • result netPrice • result= cusType • result=cus

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Path No.	Initial Data	Test Data	Expected Result
9	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType=”C”} • {totalQty=10}	• result= cusType • result=cus
10	• cus.isDefined() • SaleSystem->includes(cus)	• {cusType=!A,!B,!C}	• result=cusType • result=cus

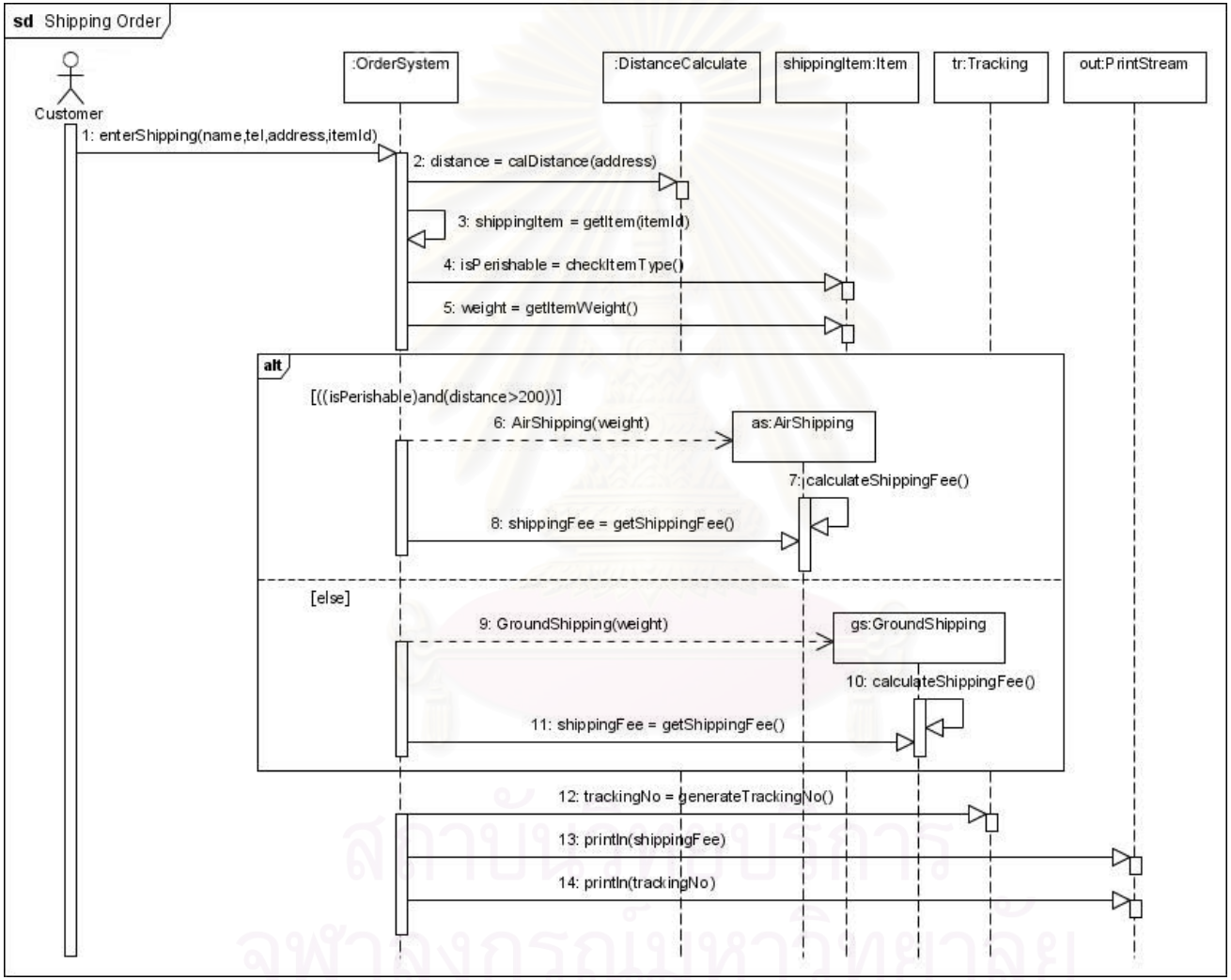
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6. **การส่งสินค้า** (Shipping Order)

ตารางที่ ข-26 รายละเอียดยูสเคสการส่งสินค้า

6. Use Case : Shipping Order	
Goal	บันทึกประเภทการจัดส่งสินค้าและคำนวณค่าใช้จ่ายในการจัดส่ง
Actor	ลูกค้า
Precondition	ลูกค้าได้สั่งสินค้าจากระบบเรียบร้อยแล้ว
Postcondition	ระบบบันทึกการจัดส่งสินค้าและแสดงค่าใช้จ่ายในการจัดส่ง
Main Success Scenario	1. ลูกค้าระบุ ชื่อผู้รับ เบอร์ติดต่อ สถานที่ในการส่งสินค้าและรหัสสินค้าที่สั่งซื้อ 2. ระบบคำนวณระยะทางในการจัดส่งระหว่างโกดังสินค้าและสถานที่จัดส่งสินค้าตามที่ลูกค้าระบุ 3. ระบบตรวจสอบประเภทของสินค้าและน้ำหนักของสินค้า 4. ระบบบันทึกการจัดส่งและคำนวณค่าใช้จ่าย a. กรณีสินค้าที่สั่งเป็นประเภท perishable และระยะทางในการจัดส่งมากกว่า 200 ไมล์ ระบบจะบันทึกการจัดส่งทางอากาศและคำนวณค่าใช้จ่ายในการส่งสินค้าและแจ้งให้ลูกค้าทราบ b. กรณีที่สินค้าไม่ใช่ประเภท perishable และ/หรือ ระยะทางในการจัดส่งไม่เกิน 200 ไมล์ ระบบจะบันทึกการจัดส่งทางบกและคำนวณค่าใช้จ่ายในการส่งสินค้าและแจ้งให้ผู้ใช้งานทราบ 5. ระบบสร้างหมายเลขติดตามของสินค้าและแสดงให้ลูกค้าได้รับทราบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ข-6 แผนภาพซีเควนซ์ของการส่งสินค้า

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-27 ตารางแสดงกรณีทดสอบของการส่งสินค้าที่สร้างโดยผู้เชี่ยวชาญ

No.	Initial Data	Test Data	Expected Result
1	• Create object Item Item shippingItem = new Item("123", "ProdA","perisable",240,3.5); • tr.getTrackNo = 011	Name = A Tel = (123)000-000 Address =New York,USA itemId = 123	• Object GroundShipping is created • gs.getShippingFee = 3.5*1.5 //(gs.rate =1.5) • tr.getTrackNo = 012
2	• Create object Item Item shippingItem = new Item("123", "ProdA","perisable",240,3.5); • tr.getTrackNo = 011	Name = A Tel = (123)000-000 Address = MexicoCity,Mexico itemId = 123	• Object AirShipping is created • as.getShippingFee = 3.5*20 //(as.rate = 20) • tr.getTrackNo = 012
3	• Create object Item Item shippingItem = new Item("345", "ProdB","nonperisable",240,3.5); • tr.getTrackNo = 011	Name = A Tel = (123)000-0000 Address =New York,USA itemId = 345	• Object GroundShipping is created • gs.getShippingFee = 3.5*1.5 //(gs.rate =1.5) tr.getTrackNo = 012
4	• Create object Item Item shippingItem = new Item("345", "ProdB","nonperisable",240,3.5); • tr.getTrackNo = 011	Name = A Tel = (123)000-0000 Address = MexicoCity,Mexico itemId = 345	• Object GroundShipping is created • gs.getShippingFee = 3.5*1.5 //(gs.rate =1.5) tr.getTrackNo = 012

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-28 ตารางแสดงเส้นทางไอออาร์ซีเอฟจีของแผนภาพซีเควนซ์การส่งสินค้าที่เครื่องมือสร้าง

Path No.	IRCFG Complete Path
1	start, distance=calDistance(address), start, end, shippingItem=getItem(itemId), start, end, isPerishable = checkItemType(), start, end, weight = getItemWeigth(), start, end, ((isPerishable)and(distance>200))<T>, AirShipping(weight), start, calculateShippingFee(), start, end, end, shippingFee = getShippingFee(), start, end, end, trackingNo = generateTrackingNo(), start, end, println(shippingFee), start, end, println(trackingNo), start, end, end
2	start, distance=calDistance(address), start, end, shippingItem=getItem(itemId), start, end, isPerishable = checkItemType(), start, end, weight = getItemWeigth(), start, end, ((isPerishable)and(distance>200))<F>, GroundShipping(weight), start, calculateShippingFee(), start, end, end, shippingFee = getShippingFee(), start, end, end, trackingNo = generateTrackingNo(), start, end, println(shippingFee), start, end, println(trackingNo), start, end, end

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-29 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การส่งสินค้าที่เครื่องมือสร้าง

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
1	println(trackingNo)	:OrderSystem	out:PrintStream				• result = trackingNo
	println(shippingFee)	:OrderSystem	out:PrintStream				• result = shippingFee
	trackingNo= generateTrackingNo( )	:OrderSystem	tr:TrackingNo	• tr.isDefined()			• tr.No = tr.No@pre+1 • result = trackingNo
	shippingFee = getShippingFee()	:OrderSystem	as:AirShipping				• result = shippingFee
	calculateShippingFee ()	as:AirShipping	as:AirShipping				• as.shippingFee=(weight* as.rate)
	AirShipping(weight)	:OrderSystem	as:AirShipping				• as.isDefined()
	weight = getItemWeigth()	:OrderSystem	as:AirShipping				• result = weight
	isPerishable = checkItemType()	:OrderSystem	shippingItem:Item				• result= isPerishable
	shippingItem=getIte m(itemId)	:OrderSystem	:OrderSystem	• OrderSystem -> includes(shippingIte m)			• result = shippingItem
	distance=calDistance (address)	:OrderSystem	:DistanceCalculate				• result = distance
	enterShipping(name,t el,address)	Customer	:OrderSystem		• ((isPerishable)and (distance>200))	• {isPerishable = true, distance=201}	

Path No.	Message's Name	Sending Obj.	Receiveing Obj.	Initial Data	Condition	Test Data	Expected Result
2	println(trackingNo)	:OrderSystem	out:PrintStream				• result = trackingNo
	println(shippingFee)	:OrderSystem	out:PrintStream				• result = shippingFee
	trackingNo= generateTrackingNo( )	:OrderSystem	tr:TrackingNo	• tr.isDefined()			• tr.No = tr.No@pre+1 • result = trackingNo
	shippingFee = getShippingFee()	:OrderSystem	as:AirShipping				• result = shippingFee
	calculateShippingFee ()	as:AirShipping	as:AirShipping				• gs.shippingFee =(weight*gs.rate)
	GroundShipping(weight)	:OrderSystem	as:AirShipping				• gs.isDefined()
	weight = getItemWeigth()	:OrderSystem	as:AirShipping				• result = weight
	isPerishable = checkItemType()	:OrderSystem	shippingItem:Item				• result= isPerishable
	shippingItem=getItem(itemId)	:OrderSystem	:OrderSystem	• OrderSystem -> includes(shippingItem)			• result = shippingItem
	distance=calDistance (address)	:OrderSystem	:DistanceCalculate				• result = distance
	enterShipping(name,tel,address)	Customer	:OrderSystem		• ((isPerishable)and (distance>200))	• {isPerishable = false, distance=200}	

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข-30 ตารางแสดงกรณีทดสอบจากแผนภาพซีเควนซ์การส่งสินค้าที่เครื่องมือสร้าง (เฉพาะ 3 คอลัมน์)

PathNo.	Initial Data	Test Data	Expected Result
1	• tr.isDefined() • OrderSystem->includes(shippingItem)	• {isPerishable = true, distance=201}	• tr.No = tr.No@pre+1 • result = trackingNo • as.shippingFee=(weight*as.rate) • result = shippingFee • as.isDefined() • result = shippingFee = calculateShippingFee() • result= isPerishable • result = shippingItem • result = distance • result = shippingFee • result = trakingNo

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

PathNo.	Initial Data	Test Data	Expected Result
2	• tr.isDefined() • OrderSystem->includes(shippingItem)	• {isPerishable = false, distance=200}	• tr.No = tr.No@pre+1 • result = trackingNo • gs.shippingFee=(weight*gs.rate) • result = shippingFee • gs.isDefined() • result = shippingFee = calculateShippingFee() • result= isPerishable • result = item • result = distance

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค
ซอร์สโค้ดของโปรแกรมการซื้อตั๋วที่ใช้ในการทดสอบ

ในส่วนนี้เป็นซอร์สโค้ดที่ใช้ในการทดสอบด้วยกรณีทดสอบที่สร้างขึ้นจากเครื่องมือ โดยนำมาแสดงเฉพาะในส่วนของการดำเนินการที่เกี่ยวข้องในเรื่องของการซื้อตั๋วเท่านั้น และส่วนที่เพิ่มเติมหรือปรับแก้เพื่อใช้ในการทดสอบโดยเฉพาะ จะมีเครื่องหมาย * วางไว้หน้าส่วนของซอร์สโค้ดดังกล่าว

```
class Program {
        static string choice;
        static string chooseWayId;
        static string customerType;
        static int qty;
        static SaleTicketSystem st;

        static void Main(string[] args)  {
                *Console.Write("Seat Balance :");
                *int seatBalance = Convert.ToInt32(Console.ReadLine());
                *st = new SaleTicketSystem(seatBalance);
                BuyTicket();
        }

        static void BuyTicket() {
                try {
                Console.Write("Enter Way ID: ");
                chooseWayId = Console.ReadLine();
                Console.Write("How many ticket you want to buy?:");
                qty = Convert.ToInt32(Console.ReadLine());

                Console.Write("Are you student?(Y/N): ");
                customerType = Console.ReadLine();
                st.buyTicket(chooseWayId, qty, customerType);
                }
                catch (System.FormatException) {
                Console.WriteLine("Format Exception");
                }
        }
}
```

```
class SaleTicketSystem {
        static Bus[] myBus = new Bus[8];
        static Driver[] myDriver = new Driver[8];
        static Way[] myWayArray = new Way[8];
        static BusManagement[] myBusManage = new BusManagement[8];
        static int ticketId;
        Ticket myTicket;

        public SaleTicketSystem(int seatBalance)  {
                *myBus[0] = new Bus("111", 3);
                *myDriver[0] = new Driver("1001", "Chaisit", "111");
                *myWayArray[0] = new Way("011", "Bangkok - Phitsanulok", 200);
                *myBusManage[0] = new BusManagement("8.00 a.m.", myWayArray[0], myBus[0]);
                *myBusManage[0].setSeatBalance(seatBalance);
        }
        public BusManagement[] MyBusManage {
                get { return myBusManage; }
        }
        public Bus[] MyBus {
           get { return myBus; }
        }
        public Driver[] MyDriver {
           get { return myDriver; }
        }
        public Way[] MyWay {
           get { return myWayArray; }
        }
        public void buyTicket(string chooseWayId,int qty,string customerType) {
           Way way = this.findWay(chooseWayId);
           BusManagement bm = this.findBusManagement(way);
           int seatBalance = bm.getSeatBalance;
           if (seatBalance >= qty) {
              if (customerType == "Y") {
                 myTicket = new DiscountTicket(ticketId, qty, bm);
                 bm.seatManagement(qty);
                 myTicket.calcPriceTicket();
                 myTicket.printTicket();
              }
              else if (customerType == "N")  {
                 myTicket = new Ticket(ticketId, qty, bm);
```

```
                myTicket.calcPriceTicket();
                bm.seatManagement(qty);
                myTicket.printTicket();
            }
            ticketId += qty;
        }
        else {
            Console.WriteLine("There is not enough seat for your request or maybe the seat is full");
        }
    }
    public Way findWay(string wayId)  {
        Way thisWay=null;
        foreach (Way w in myWayArray)
            if (wayId == w.getWayId) {
                thisWay = w;
                break;
            }
        return thisWay;
    }

    public BusManagement findBusManagement(Way w) {
        BusManagement thisBusManagement = null;
        foreach (BusManagement bm in myBusManage)
            if (w.getWayId == bm.getWayID)  {
                thisBusManagement = bm;
                break;
            }
        return thisBusManagement;
    }
}

class BusManagement
{
    private string date;
    private string time;
    private int seatBalance;
    private int soldTicket;
    private Bus bus;
    private Way way;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
public BusManagement(string t, Way thisWay, Bus thisBus) {
    date = DateTime.Today.ToShortDateString();
    time = t;
    way = thisWay;
    bus = thisBus;
    seatBalance = bus.getNumSeat;
}
public void seatManagement(int qtyBuy) {
    seatBalance -= qtyBuy;
    soldTicket += qtyBuy;
}
public string getDate {
    get { return date; }
}
public string getTime {
    get { return time; }
}
public void setSeatBalance(int sb)  {
    this.seatBalance = sb;
}
public int getSeatBalance {
    get {return seatBalance;}
}
public int getSoldTicket {
    get { return soldTicket; }
}
public double getPrice{
    get { return way.getPrice; }
}
public string getBusID {
   get { return bus.getBusId; }
}
public string getWay {
    get { return way.getDescription; }
}
public string getWayID {
    get { return way.getWayId; }
}
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        public int getNumSeat {
            get { return bus.getNumSeat; }
        }
}

class Driver {
        private string driverId;
        private string driverName;
        private string busDriver;

        public Driver(string id, string name, string bus) {
            driverId = id; driverName = name; busDriver = bus;
        }
        public string getDriverId {
            get { return driverId; }
        }
        public string getDriverName {
            get { return driverName; }
        }
        public string getDriverBus {
            get { return busDriver; }
        }
}

class Ticket {
        protected int ticketID;
        protected string dateTravel;
        protected double totalPrice;
        protected BusManagement busmanage;
        protected int qty;
        public Ticket(int id,int qtyBuy, BusManagement busM) {
            ticketID = id+qtyBuy;
            busmanage = busM;
            dateTravel = busmanage.getDate;
            qty = qtyBuy;
        }
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
        virtual public void printTicket() {
            while (qty > 0) {
                Console.WriteLine("Ticket ID : " + ticketID--);
                Console.WriteLine("Date: " + busmanage.getDate + " " + "Time: " + busmanage.getTime);
                Console.WriteLine("From - To: " + busmanage.getWay);
                Console.WriteLine("Bus No.: " + busmanage.getBusID + "  Seat No." +
Convert.ToString(((busmanage.getSeatBalance)-(qty+1))));
                Console.WriteLine("Price: " + busmanage.getPrice);
                qty--;
            }
            Console.WriteLine("           TotalPrice is " + totalPrice+" Bath");
        }
        virtual public void calcPriceTicket() {
            totalPrice = (busmanage.getPrice)*(double)qty;
        }
        public double TotalPrice {
            get { return totalPrice; }
        }
}

class Way {
        private string wayId;
        private string description;
        private double price;
        public Way(string Id, string des, double p) {
            wayId = Id; description = des; price = p;
        }
        public string getWayId {
            get { return wayId; }
        }
        public string getDescription {
            get { return description; }
        }
        public double getPrice {
            get { return price; }
        }
}
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวฉัฐธร ทองระอา เกิดวันที่ 22 พฤศจิกายน พ.ศ. 2526 สำเร็จการศึกษาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ จากภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีพ.ศ. 2548 จากนั้นได้เข้าศึกษาต่อในระดับปริญญาโท ในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาการพัฒนาซอฟต์แวร์ด้านธุรกิจ ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย