การตรวจจับข้อบกพร่องในขั้นตอนการออกแบบเพื่อปรับปรุงความสามารถในการเปลี่ยนแปลงซอฟต์แวร์เชิงวัตถุ

นางสาวธันยวัต จันทร์เปีย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DEFECTS DETECTION AT DESIGN PHASE FOR IMPROVING OBJECT-ORIENTED SOFTWARE MODIFIABILITY

Miss Tanyawat Chanpia

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2006

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การตรวจจับข้อบกพร่องในขั้นตอนการออกแบบเพื่อปรับปรุงความสามารถในการเปลี่ยนแปลงซอฟต์แวร์เชิงวัตถุ
โดย	นางสาวธันยวัต จันทร์เปีย
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี)

..... กรรมการ
(อาจารย์นครทิพย์ พร้อมพูล)

..... กรรมการ
(อาจารย์ เชษฐ พัฒโนทัย)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นางสาวธันยวัต จันทร์เปีย : การตรวจจับข้อบกพร่องในขั้นตอนการออกแบบเพื่อปรับปรุงความสามารถในการเปลี่ยนแปลงซอฟต์แวร์เชิงวัตถุ. (DEFECTS DETECTION AT DESIGN PHASE FOR IMPROVING OBJECT-ORIENTED SOFTWARE MODIFIABILITY) อ. ที่ปรึกษา: รศ.ดร.พรศิริ หมื่นไชยศรี, 240 หน้า.

วิทยานิพนธ์นี้นำเสนอวิธีการตรวจจับข้อบกพร่องของโมเดลการออกแบบ 6 ประเภท ได้แก่ Data Class, Feature Envy, Message Chains, Middle Man, God Class, และ Switch Statements โดยพิจารณาว่ามีข้อบกพร่องประเภทใดบ้างที่ส่งผลกระทบต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์เชิงวัตถุ ในงานวิจัยมีการออกแบบกลยุทธ์การตรวจจับข้อบกพร่องด้วยมาตรวัดเชิงวัตถุสำหรับโมเดลการออกแบบที่แผนภาพคลาสและแผนภาพซีเควนซ์ นำเสนอวิธีการรีแฟคทอริงเพื่อแก้ไขข้อบกพร่องแต่ละประเภท พร้อมทั้งนำเสนอวิธีการหาช่วงของมาตรวัดที่บอกถึงข้อบกพร่องโดยประยุกต์ใช้อัลกอริทึมซีมูเลทเต็ดแอนนิลลิงเพื่อหาค่าช่วงที่เหมาะสมที่สุดโดยใช้กลุ่มตัวอย่าง 35 ตัวอย่างสำหรับข้อบกพร่องแต่ละประเภท รวมถึงได้ออกแบบและพัฒนาเครื่องมือสำหรับตรวจจับข้อบกพร่องของโมเดลการออกแบบและเครื่องมือที่ใช้ในการหาค่าช่วงที่เหมาะสม

ในงานวิจัยได้ทดสอบกลยุทธ์การตรวจจับข้อบกพร่องและประเมินผลกระทบของการเกิดข้อบกพร่องต่าง ๆ ต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ด้วยกลุ่มตัวอย่างสำหรับทดสอบ 5 ตัวอย่างต่อข้อบกพร่องของโมเดลการออกแบบหนึ่งประเภท โดยเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการตรวจจับข้อบกพร่องของโมเดลการออกแบบ และประยุกต์ใช้วิธีรีแฟคทอริงสำหรับแก้ไขข้อบกพร่องของโมเดลการออกแบบแต่ละประเภท ผลของการทดสอบระบุว่าการแก้ไขระบบที่มีข้อบกพร่องของโมเดลการออกแบบประเภท Data Class, Middle Man, และ God Class สามารถทำให้ความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ดีขึ้น การแก้ไขระบบที่มีข้อบกพร่องประเภท Feature Envy และ Message Chains ไม่มีผลทำให้ความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ดีขึ้น ส่วนการแก้ไขระบบที่มีข้อบกพร่องประเภท Switch Statements มีผลทำให้ความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ลดลง

ภาควิชา.... วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....ธันยวัต จันทร์เปีย.....
สาขาวิชา....วิทยาศาสตร์คอมพิวเตอร์.....ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา .....2549.....

4771425521 : MAJOR COMPUTER SCIENCE

KEY WORD: OBJECT-ORIENTED SOFTWARE DESIGN / MODIFIABILITY / REFACTORING / DESIGN FLAWS/ BAD-SMELL / OBJECT-ORIENTED METRICS

TANYAWAT CHANPIA: DEFECTS DETECTION AT DESIGN PHASE FOR IMPROVING OBJECT-ORIENTED SOFTWARE MODIFIABILITY. THESIS ADVISOR: ASSOCIATE PROFESSOR PORNSIRI MUENCHAISRI, PH.D.,

240 pp.

This thesis proposes detection strategies for six design defects including Data Class Feature Envy, Message Chains, Middle Man, God Class, and Switch Statement to verify whether the particular design defects affect object-oriented software modifiability. The strategies use object-oriented software design metrics for determining a fraction of class and sequence diagram which is affected by particular design defects. The approach also suggests refactoring techniques for modifying the class and the sequence diagram. In addition, an approach for finding the optimized threshold values for detecting particular design defects is also provided by applying simulated annealing algorithm to 35 design models for each design defect. An automated tool for design defects detection and for calculating the optimized threshold values is also implemented.

The thesis approach is evaluated by comparing modifiability metrics before and after applying the refactoring techniques to 5 design models for each design defect. The result shows that modifiability of software is enhanced after applying the refactoring to Data Class, Middle Man, and God Class design defects and not changed for Feature Envy and Middle Man design defects. For Switch Statements design defect, the modifiability metric values are dropped.

Department.... Computer Engineering...Student's signature ธันยวัต จันทไชย
Field of study.... Computer Science.....Advisor's signature P. Muenchaisri
Academic year ...2006.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือจาก รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี อาจารย์ที่ปรึกษาวิทยานิพนธ์ของข้าพเจ้า ขอกราบขอบพระคุณอาจารย์ที่ให้คำแนะนำ และข้อเสนอแนะต่างๆ ตลอดระยะเวลาของการทำวิทยานิพนธ์ของข้าพเจ้า จนสำเร็จลุล่วงได้ด้วยดี

ขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา เป็นประธานกรรมการ อาจารย์นครทิพย์ พร้อมพูล และอาจารย์เชษฐ พัฒโนทัย เป็นกรรมการสอบวิทยานิพนธ์ ซึ่งได้สละเวลาและให้คำแนะนำต่างๆ ในการสอบวิทยานิพนธ์ของข้าพเจ้าอย่างดียิ่ง

ขอขอบคุณ คุณอติโรจน์ สืบวงศ์คล้าย และคุณเมทินี เขียวกันยะสำหรับคำแนะนำ คำปรึกษาและความช่วยเหลือในทุกๆ ด้าน

ท้ายสุดนี้ขอกราบขอบพระคุณบิดา มารดา น้องสาว และเพื่อนๆ ทุกคนที่คอยเป็นกำลังใจและให้ความสนับสนุนมาโดยตลอด

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 วัตถุประสงค์.....2
1.3 ขอบเขตงานวิจัย.....3
1.4 ขั้นตอนการวิจัย.....3
1.5 ประโยชน์ที่จะได้รับ.....4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....5
2.1 ทฤษฎีที่เกี่ยวข้อง.....5
2.1.1 โมเดลการออกแบบที่แสดงด้วยแผนภาพยูเอ็มแอล.....5
2.1.2 การวัดซอฟต์แวร์.....9
2.1.3 ข้อบกพร่องของโมเดลการออกแบบ.....10
2.1.4 ร่องรอยที่ไม่ดี.....13
2.1.5 รีแฟคทอริง.....15
2.1.6 ความสามารถในการเปลี่ยนแปลง.....16
2.1.7 วิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่ (Combinatorial Optimization) [10].....18
2.2 งานวิจัยที่เกี่ยวข้อง.....22
2.2.1 งานวิจัย “Bad-Smell Detection for Refactoring using Object-Oriented Software Metrics” [13].....22

หน้า

2.2.2 งานวิจัย “Detection Strategies: Metrics-Based Rules for Detecting Design Flaws” [14] 23
2.2.3 งานวิจัย “Towards the Optimization of Automatic Detection of Design Flaws in Object-Oriented Software System” [15] 24
บทที่ 3 วิธีการตรวจจับข้อบกพร่องของโมเดลการออกแบบหรือร่องรอยที่ไม่ดี 27
3.1 ขั้นตอนการวิจัย 28
3.1.1 การเลือกข้อบกพร่องและร่องรอยที่ไม่ดีของโมเดลการออกแบบ ที่เกี่ยวข้องกับความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ 28
3.1.2 การกำหนดกลยุทธ์สำหรับตรวจจับข้อบกพร่องของโมเดลการออกแบบ 28
3.1.3 การกำหนดช่วงของมาตรวัด 28
3.1.4 การคำนวณค่ามาตรวัดความสามารถของการเปลี่ยนแปลง 29
3.1.5 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ 29
3.1.6 การประยุกต์ใช้การรีแฟคทอริง 29
3.1.7 การคำนวณค่ามาตรวัดความสามารถของการเปลี่ยนแปลง 30
3.1.8 การตรวจสอบมาตรวัดเชิงวัตถุสำหรับการเปลี่ยนแปลงซอฟต์แวร์ 30
3.2 การออกแบบกลยุทธ์สำหรับตรวจจับข้อบกพร่องของโมเดลการออกแบบ 30
3.2.1 ขั้นตอนการออกแบบกลยุทธ์ 30
3.2.2 กลยุทธ์การตรวจจับข้อบกพร่อง 31
บทที่ 4 การกำหนดช่วงของมาตรวัด 58
4.1 วิธีการกำหนดช่วงของมาตรวัด 58
4.1.1 การเก็บกลุ่มตัวอย่าง 59
4.1.2 การจำแนกประเภทกลุ่มตัวอย่าง 60
4.1.3 การกำหนดช่วงของค่ามาตรวัดเริ่มต้นที่บอกถึงข้อบกพร่อง 61
4.1.4 การกำหนดฟังก์ชันการประเมินค่า 61

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หน้า

4.1.5 การคำนวณค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟเริ่มต้น 61
4.1.6 การคำนวณหาค่าช่วงของมาตรวัดที่เหมาะสมที่สุด 61
4.1.7 การประเมินผลค่าช่วงของมาตรวัด 62
4.2 ผลการคำนวณหาช่วงของมาตรวัดที่เหมาะสมที่สุด 62
4.2.1 ผลการคำนวณค่าช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่อง ประเภท Feature Envy 62
4.2.2 ผลการคำนวณค่าช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่อง ประเภท Message Chains 66
4.2.3 ผลการคำนวณค่าช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่อง ประเภท Middle Man 68
4.2.4 ผลการคำนวณค่าช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่อง ประเภท God Class 71
บทที่ 5 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบ 74
5.1 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Data Class 74
5.1.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ 74
5.1.2 การประยุกต์ใช้วิธีรีแฟคทอริง 79
5.1.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง 83
5.2 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Feature Envy 84
5.2.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ 84
5.2.2 การประยุกต์ใช้วิธีรีแฟคทอริง 89
5.2.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง 93

หน้า

5.3 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Message Chains 95
5.3.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ 95
5.3.2 การประยุกต์ใช้วิธีรีแฟคทอริง 98
5.3.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง 100
5.4 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Middle Man 101
5.4.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ 101
5.4.2 การประยุกต์ใช้วิธีรีแฟคทอริง 107
5.4.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง 110
5.5 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท God Class 112
5.5.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ 112
5.5.2 การประยุกต์ใช้วิธีรีแฟคทอริง 116
5.5.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง 122
5.6 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Switch Statements 123
5.6.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ 123
5.6.2 การประยุกต์ใช้วิธีรีแฟคทอริง 127
5.6.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง 129
บทที่ 6 การออกแบบและพัฒนาเครื่องมือคำนวณหาช่วงของมาตรวัด และตรวจจับข้อบกพร่อง 131

หน้า

6.1 การออกแบบและพัฒนาเครื่องมือคำนวณหาช่วงของมาตรวัด และตรวจจับข้อบกพร่อง 131
6.1.1 ยูสเคสของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ 131
6.1.2 สถาปัตยกรรมของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ 133
6.1.3 กิจกรรมภายในของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ 135
บทที่ 7 บทสรุปและข้อเสนอแนะ 137
7.1 บทสรุป 137
7.2 ข้อจำกัดและข้อเสนอแนะ 138
รายการอ้างอิง 140
ภาคผนวก 142
ประวัติผู้เขียนวิทยานิพนธ์ 240

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class 34
ตารางที่ 3.2 แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class 35
ตารางที่ 3.3 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Feature Envy..... 37
ตารางที่ 3.4 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Feature Envy..... 39
ตารางที่ 3.5 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Message Chains..... 42
ตารางที่ 3.6 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Message Chains..... 44
ตารางที่ 3.7 แสดงผลการคำนวณมาตรวัดของระบบที่มีคลาส B เป็น Middle Man 47
ตารางที่ 3.8 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Message Chains..... 48
ตารางที่ 3.9 แสดงผลการคำนวณมาตรวัดของระบบที่มีคลาส B เป็น God Class 52
ตารางที่ 3.10 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น God Class..... 54
ตารางที่ 4.1 แสดงผลการคำนวณค่าช่วงของมาตรวัดที่ใช้ในการตรวจจับข้อบกพร่องประเภท Feature Envy..... 65
ตารางที่ 4.2 แสดงผลการคำนวณค่าช่วงของมาตรวัดที่ใช้ในการตรวจจับข้อบกพร่องประเภท Message Chains..... 67
ตารางที่ 4.3 แสดงผลการคำนวณค่าช่วงของมาตรวัดที่ใช้ในการตรวจจับข้อบกพร่องประเภท Middle Man..... 70
ตารางที่ 4.4 แสดงผลการคำนวณค่าช่วงของมาตรวัดที่ใช้ในการตรวจจับข้อบกพร่องประเภท God Class..... 72
ตาราง 5.1 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 8 84
ตาราง 5.2 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37 84
ตาราง 5.3 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37 94
ตาราง 5.4 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 38 94
ตาราง 5.5 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 39 95

หน้า

ตาราง 5.6 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37 100
ตาราง 5.7 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 38 101
ตาราง 5.8 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 33 111
ตาราง 5.9 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37 111
ตาราง 5.10 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 40 112
ตาราง 5.11 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37 122
ตาราง 5.12 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 40 123
ตาราง 5.13 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 41 130
ตาราง 5.14 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 42 130
ตารางที่ ก.1 การแบ่งกลุ่มตัวอย่างเป็นเซ็ตทูนนิงและเซ็ตแวลิเดชันสำหรับข้อบกพร่องแต่ละประเภท 180
ตารางที่ ข.1 แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง 183
ตารางที่ ข.3 แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy ก่อนทำรีแฟคทอริง 194
ตารางที่ ข.4 แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง 202
ตารางที่ ข.5 แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains ก่อนทำรีแฟคทอริง 212
ตารางที่ ข.6 แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains หลังทำรีแฟคทอริง 217
ตารางที่ ข.7 แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man ก่อนทำรีแฟคทอริง 223
ตารางที่ ข.8 แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man หลังทำรีแฟคทอริง 227
ตารางที่ ข.9 แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class ก่อนทำรีแฟคทอริง 231

หน้า

ตารางที่ ข.10 แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class หลังทำรีแฟคทอริง 235
ตารางที่ ข.11 แสดงผลการคำนวณมาตรวัดของระบบที่มี Switch Statements ก่อนทำรีแฟคทอริง 239
ตารางที่ ข.12 แสดงผลการคำนวณมาตรวัดของระบบที่มี Switch Statements หลังทำรีแฟคทอริง 239

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 2.1 ตัวอย่างแผนภาพคลาสสำหรับ Appointment System 8
รูปที่ 2.2 ตัวอย่างแผนภาพซีเควนซ์สำหรับยูสเคส Make Appointment 9
รูปที่ 2.3 ตัวอย่างกราฟ (Graph) ของฟังก์ชันการประเมินค่าใดๆ 19
รูปที่ 2.4 รหัสเทียมของอัลกอริทึมซีมูเลทเต็ดแอนนิลลิง 20
รูปที่ 3.1 ขั้นตอนการตรวจจับข้อบกพร่องของโมเดลการออกแบบหรือร่องรอยที่ไม่ดี 27
รูปที่ 3.2 แผนภาพคลาสของระบบที่มีData Class 33
รูปที่ 3.3 แผนภาพซีเควนซ์ของระบบที่มี Data Class 33
รูปที่ 3.4 แผนภาพคลาสของระบบหลังการรีแฟคทอริง 34
รูปที่ 3.5 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง 34
รูปที่ 3.6 แผนภาพคลาสของระบบที่มีเมทธอดที่เป็น Feature Envy 37
รูปที่ 3.7 แผนภาพซีเควนซ์ของระบบที่มีเมทธอดที่เป็น Feature Envy 37
รูปที่ 3.8 แผนภาพคลาสของระบบหลังการรีแฟคทอริง 38
รูปที่ 3.9 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง 38
รูปที่ 3.10 แผนภาพคลาสของระบบที่มีเมทธอดที่เป็น Message Chains 41
รูปที่ 3.11 แผนภาพซีเควนซ์ของระบบที่มีเมทธอดที่เป็น Message Chains 41
รูปที่ 3.12 แผนภาพคลาสของระบบหลังการรีแฟคทอริง 43
รูปที่ 3.13 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง 43
รูปที่ 3.14 แผนภาพคลาสของระบบที่มีคลาสที่เป็น Middle Man 46
รูปที่ 3.15 แผนภาพซีเควนซ์ของระบบที่มีคลาสที่เป็น Middle Man 47
รูปที่ 3.16 แผนภาพคลาสของระบบหลังการรีแฟคทอริง 48
รูปที่ 3.17 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง 48
รูปที่ 3.18 แผนภาพคลาสของระบบที่มีคลาสที่เป็น God Class 52
รูปที่ 3.19 แผนภาพซีเควนซ์ของระบบที่มีคลาสที่เป็น God Class 52
รูปที่ 3.20 แผนภาพคลาสของระบบหลังการรีแฟคทอริง 53
รูปที่ 3.21 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง 53
รูปที่ 3.22 แผนภาพคลาสของระบบที่มี Switch Statements 56
รูปที่ 3.23 แผนภาพซีเควนซ์ของระบบที่มี Switch Statements 56
รูปที่ 3.24 แผนภาพคลาสของระบบหลังการรีแฟคทอริง 57

หน้า

รูปที่ 3.25 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง 57
รูปที่ 4.1 ขั้นตอนการหาค่าช่วงของมาตรวัด 58
รูปที่ 4.2 ตัวอย่างเอกสารเอ็กซ์เอ็มแอลสำหรับเก็บข้อมูลกลุ่มตัวอย่างของข้อบกพร่องประเภท God Class 59
รูปที่ 4.3 ตัวอย่างเอกสารเอ็กซ์เอ็มแอลที่เก็บผลการระบุข้อบกพร่องของเอนทิตี้ตัวอย่าง 60
รูปที่ 4.4 ตัวอย่างเอกสารเอ็กซ์เอ็มแอลที่เก็บค่าช่วงของมาตรวัดเริ่มต้นของข้อบกพร่องประเภท Message Chains 61
รูปที่ 5.1 แผนภาพคลาสของระบบตัวอย่างที่มี Class FoodInfo เป็น Data Class 75
รูปที่ 5.2 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มี Class FoodInfo เป็น Data Class 75
รูปที่ 5.3 แผนภาพคลาสของระบบตัวอย่างที่มี Class CreditVendor เป็น Data Clas 76
รูปที่ 5.4 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มี Class CreditVendor เป็น Data Class 77
รูปที่ 5.5 แผนภาพคลาสของระบบตัวอย่างที่มี Class ShippingPlace เป็น Data Class 78
รูปที่ 5.6 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มี Class ShippingPlace เป็น Data Class 78
รูปที่ 5.7 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส FoodInfo 79
รูปที่ 5.8 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส FoodInfo 80
รูปที่ 5.9 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส CreditVendor 81
รูปที่ 5.10 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส CreditVendor 81
รูปที่ 5.11 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส ShippingPlace 82
รูปที่ 5.12 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส ShippingPlace 83
รูปที่ 5.13 แผนภาพคลาสของระบบตัวอย่างที่มีเมทธอด calculateAmount ของคลาส POLineItem เป็น Feature Envy 85
รูปที่ 5.14 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีเมทธอด calculateAmount ของคลาส POLineItem เป็น Feature Envy 86
รูปที่ 5.15 แผนภาพคลาสของระบบตัวอย่างที่มีเมทธอด viewRentalReport ของคลาส Employee เป็น Feature Envy 87
รูปที่ 5.16 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีเมทธอด viewRentalReport ของคลาส Employee เป็น Feature Envy 87
รูปที่ 5.17 แผนภาพคลาสของระบบตัวอย่างที่มีเมทธอด postReturnMaterial ของคลาส ReturnedMaterial เป็น Feature Envy 88

หน้า

รูปที่ 5.18 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีเมทธอด postReturnMaterial ของคลาส ReturnedMaterial เป็น Feature Envy 89
รูปที่ 5.19 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด calculateAmount 90
รูปที่ 5.20 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด calculateAmount 90
รูปที่ 5.21 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด viewRentalReport 91
รูปที่ 5.22 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด viewRentalReport 92
รูปที่ 5.23 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด postReturnMaterial 92
รูปที่ 5.24 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด postReturnMaterial 93
รูปที่ 5.25 แผนภาพคลาสของระบบตัวอย่างที่เมทธอด viewDocumentFlow ของคลาส PurchaseOrder มีข้อบกพร่องประเภท Message Chains 96
รูปที่ 5.26 แผนภาพซีเควนซ์ของระบบตัวอย่างที่เมทธอด viewDocumentFlow ของคลาส PurchaseOrder มีข้อบกพร่องประเภท Message Chains 96
รูปที่ 5.27 แผนภาพคลาสของระบบตัวอย่างที่เมทธอด getTransactions ของคลาส Member มีข้อบกพร่องประเภท Message Chains 97
รูปที่ 5.28 แผนภาพซีเควนซ์ของระบบตัวอย่างที่เมทธอด getTransactions ของคลาส Member มีข้อบกพร่องประเภท Message Chains 98
รูปที่ 5.29 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด viewDocumentFlow ของคลาส PurchaseOrder 98
รูปที่ 5.30 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด viewDocumentFlow ของคลาส PurchaseOrder 99
รูปที่ 5.31 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด getTransactions ของคลาส Member 99
รูปที่ 5.32 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด getTransactions ของคลาส Member 100
รูปที่ 5.33 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส LineItem เป็น Middle Man 102
รูปที่ 5.34 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส LineItem เป็น Middle Man 102
รูปที่ 5.35 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส POScheduler เป็น Middle Man 103
รูปที่ 5.36 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส POScheduler เป็น Middle Man 104
รูปที่ 5.37 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส DigitHandler 104
รูปที่ 5.38 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส DigitHandler 105

หน้า

รูปที่ 5.39 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส BinaryOperation เป็น Middle Man ... 106
รูปที่ 5.40 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส BinaryOperation เป็น Middle Man 106
รูปที่ 5.41 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส Calculator เป็น Middle Man 107
รูปที่ 5.42 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส Calculator เป็น Middle Man 107
รูปที่ 5.43 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส LineItem 108
รูปที่ 5.44 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส LineItem 108
รูปที่ 5.45 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส POScheduler..... 109
รูปที่ 5.46 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส POScheduler 109
รูปที่ 5.47 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส BinaryOperation 110
รูปที่ 5.48 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส BinaryOperation..... 110
รูปที่ 5.49 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส PurchaseOrder เป็น God Class 113
รูปที่ 5.50 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส PurchaseOrder เป็น God Class.... 113
รูปที่ 5.51 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส PurchaseOrder เป็น God Class.... 114
รูปที่ 5.52 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส PurchaseOrder เป็น God Class.... 114
รูปที่ 5.53 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส Display เป็น God Class 115
รูปที่ 5.54 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส Display เป็น God Class 116
รูปที่ 5.55 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder และ CreditVendor 117
รูปที่ 5.56 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder และ CreditVendor 117
รูปที่ 5.57 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder, POStatusController, Material, และคลาส POScheduler..... 118
รูปที่ 5.58 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder, POStatusController, Material, และคลาส POScheduler..... 119
รูปที่ 5.59 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder และ POReportGenerator 120
รูปที่ 5.60 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder และ POReportGenerator 120
รูปที่ 5.61 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Display 121
รูปที่ 5.62 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Display..... 121

หน้า

รูปที่ 5.63 แผนภาพคลาสของระบบตัวอย่างที่ 41 ที่มี Switch Statements 124
รูปที่ 5.64 แผนภาพซีเควนซ์ของระบบตัวอย่างที่ 41 ที่มี Switch Statements 125
รูปที่ 5.65 แผนภาพคลาสของระบบตัวอย่างที่ 42 ที่มี Switch Statements 126
รูปที่ 5.66 แผนภาพซีเควนซ์ของระบบตัวอย่างที่ 42 ที่มี Switch Statements 127
รูปที่ 5.68 แผนภาพซีเควนซ์ของระบบตัวอย่างหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Customer 128
รูปที่ 5.69 แผนภาพคลาสของระบบตัวอย่างหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Search .. 129
รูปที่ 5.70 แผนภาพซีเควนซ์ของระบบตัวอย่างหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Search 129
รูปที่ 6.1 แผนยูสเคส ของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ 131
รูปที่ 6.2 แผนภาพคลาสแสดงสถาปัตยกรรม (Architecture) ของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ 133
รูปที่ 6.3 แผนภาพแอคทิวิตี้แสดงกิจกรรมภายในของเครื่องมือตรวจจับข้อบกพร่อง 135
รูปที่ ก.1 แผนภาพคลาสระบบขายสินค้าผ่านเว็บ 143
รูปที่ ก.2 แผนภาพคลาสระบบลิฟต์ 144
รูปที่ ก.3 แผนภาพคลาสระบบการผลิตเสื้อผ้าส่งออก 145
รูปที่ ก.4 แผนภาพคลาสระบบการลงทะเบียนเรียนผ่านเว็บ 146
รูปที่ ก.5 แผนภาพคลาสระบบสารสนเทศคลังยา 147
รูปที่ ก.6 แผนภาพคลาสระบบการฝากถอนคำนวณเงินดำรงฐานะของธนาคารพาณิชย์ 148
รูปที่ ก.7 แผนภาพคลาสระบบซื้อขายหุ้นผ่านโทรศัพท์เคลื่อนที่ 149
รูปที่ ก.8 แผนภาพคลาสระบบจัดการร้านอาหาร 150
รูปที่ ก.9 แผนภาพคลาสระบบงานโฆษณา 151
รูปที่ ก.10 แผนภาพคลาสระบบจองท่องเที่ยว 152
รูปที่ ก.11 แผนภาพคลาสระบบโรงภาพยนตร์แบบ Multiplex 153
รูปที่ ก.12 แผนภาพคลาสระบบขนส่งทางอากาศ 154
รูปที่ ก.13 แผนภาพคลาสระบบร้านหนังสือออนไลน์ 155
รูปที่ ก.14 แผนภาพคลาสระบบเช่าวีดีโอ 156
รูปที่ ก.15 แผนภาพคลาสระบบโรงพยาบาล 157
รูปที่ ก.16 แผนภาพคลาสระบบชำระค่าบริการล่วงหน้า 158

หน้า

รูปที่ ก.17 แผนภาพคลาสระบบห้องสมุดใหญ่ 158
รูปที่ ก.18 แผนภาพคลาสระบบสถานีบริการหลวง 159
รูปที่ ก.19 แผนภาพคลาสระบบขายซีดีออนไลน์ 160
รูปที่ ก.20 แผนภาพคลาสระบบ ATM 161
รูปที่ ก.21 แผนภาพคลาสระบบจ่ายเงิน 162
รูปที่ ก.22 แผนภาพคลาสระบบลงทะเบียนสัมมนา 162
รูปที่ ก.23 แผนภาพคลาสระบบส่วนลดของการขาย 163
รูปที่ ก.24 แผนภาพคลาสระบบจัดซื้อวัตถุดิบ 164
รูปที่ ก.25 แผนภาพคลาสระบบขายสินค้า 164
รูปที่ ก.26 แผนภาพคลาสระบบตัวแทนจำหน่ายรูปวาด 165
รูปที่ ก.27 แผนภาพคลาสระบบสั่งอุปกรณ์สุนัขออนไลน์ 166
รูปที่ ก.28 แผนภาพคลาสระบบหุ้น 167
รูปที่ ก.29 แผนภาพคลาสระบบขายตั๋วหนังออนไลน์ 168
รูปที่ ก.30 แผนภาพคลาสระบบเช่ารถ 169
รูปที่ ก.31 แผนภาพคลาสระบบ Library 170
รูปที่ ก.32 แผนภาพคลาสระบบคงคลังสินค้า 170
รูปที่ ก.33 แผนภาพคลาสระบบสินค้าบนเว็บ 171
รูปที่ ก.34 แผนภาพคลาสระบบกลุ่มงานสินค้าคงคลัง 172
รูปที่ ก.35 แผนภาพคลาส ระบบงานเงินเดือน 173
รูปที่ ก.36 แผนภาพคลาสระบบการส่งออก 174
รูปที่ ก.37 แผนภาพคลาสระบบกลุ่มงานสั่งซื้อ 175
รูปที่ ก.38 แผนภาพคลาสระบบร้านเช่าวีดีโอ 175
รูปที่ ก.39 แผนภาพคลาสระบบงานต้นทุนการผลิต 176
รูปที่ ก.40 แผนภาพคลาสระบบ Calculator 177
รูปที่ ก.41 แผนภาพคลาสระบบซื้อขายสินค้าผ่านเว็บ 178
รูปที่ ก.42 แผนภาพคลาสระบบห้องสมุด 179

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เป็นที่ทราบกันว่าขั้นตอนการบำรุงรักษา (Maintenance Phase) คือขั้นตอนที่ใช้ทรัพยากรขององค์กรมากที่สุดในกระบวนการพัฒนาซอฟต์แวร์ ไม่ว่าจะเป็นธุรกิจประเภทใด กิจกรรมในขั้นตอนการบำรุงรักษาจะมีวัตถุประสงค์เพื่อการแก้ไขข้อผิดพลาด (Corrective) การปรับซอฟต์แวร์เพื่อให้เข้ากับสภาพแวดล้อม (Adaptive) การปรับซอฟต์แวร์เมื่อมีความต้องการ (Requirements) ของผู้ใช้งานใหม่ (Perfective) หรือปรับซอฟต์แวร์เพื่อป้องกันปัญหาที่จะเกิดขึ้นในอนาคต (Preventive) [1]

การแก้ไขข้อผิดพลาด เป็นกิจกรรมที่เกิดขึ้นเมื่อผู้ใช้งานพบข้อผิดพลาด ซึ่งผู้พัฒนาระบบ (Developer) มีหน้าที่แก้ไขข้อผิดพลาดที่เกิดขึ้น หรือในบางครั้งองค์กรมีแผนการเปลี่ยนสภาพแวดล้อมใหม่สำหรับระบบเดิม เพื่อให้มีประสิทธิภาพในการทำงานดีขึ้น ผู้พัฒนาระบบจะต้องปรับซอฟต์แวร์เพื่อให้เข้ากับสภาพแวดล้อมใหม่ และเมื่อผู้ใช้งานนำซอฟต์แวร์ไปใช้งานแล้ว อาจจะมองเห็นแนวทางการเพิ่มฟังก์ชันการทำงานหรือมีความต้องการเพิ่มเติม ที่จะใช้ความสามารถของซอฟต์แวร์เข้ามาช่วยอำนวยความสะดวกต่อระบบธุรกิจขององค์กร เพื่อให้การดำเนินงานเป็นไปอย่างราบรื่นมากขึ้น ผู้พัฒนาระบบจึงมีหน้าที่ปรับซอฟต์แวร์เพื่อเพิ่มการทำงานในส่วนที่จะรองรับความต้องการใหม่ นอกจากนี้ ซอฟต์แวร์จะต้องได้รับการปรับปรุงเพื่อป้องกันปัญหาที่จะเกิดขึ้นในอนาคตอีกด้วย จากวัตถุประสงค์ของกิจกรรมในขั้นตอนบำรุงรักษาที่กล่าวมา จะเห็นได้ว่า ความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ (Software Modifiability) ซึ่งเป็นหนึ่งในคุณลักษณะระดับกลาง (Intermediate-level Characteristic) ของคุณภาพทางด้านการบำรุงรักษา (Maintainability) ของซอฟต์แวร์ [2] จึงเป็นสิ่งสำคัญสำหรับซอฟต์แวร์ที่ควรจะมีเพื่อรองรับการพัฒนาที่จะเกิดขึ้นต่อไป

การประเมินความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ในขั้นตอนต้นของกระบวนการพัฒนาซอฟต์แวร์เป็นสิ่งที่มีประโยชน์ ซึ่งถ้าสามารถกระทำในขั้นตอนการออกแบบซอฟต์แวร์ จะช่วยให้ผู้ออกแบบระบบ (Software Designer) ตระหนักถึงข้อบกพร่องของโมเดลการออกแบบ (Design Model) ที่จะส่งผลให้ความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ลดลง และผู้ออกแบบระบบจะสามารถแก้ไขโมเดลการออกแบบเพื่อให้มีความสามารถในการ

เปลี่ยนแปลงซอฟต์แวร์ดีขึ้น ก่อนที่จะส่งมอบต่อไปให้แก่ผู้พัฒนาระบบในขั้นตอนต่อไป ซึ่งภาษาที่ใช้ในการออกแบบซอฟต์แวร์เชิงวัตถุที่เป็นที่นิยมอย่างมาก ได้แก่ ภาษายูเอ็มแอล (Unified Modeling Language: UML) [3] โดยแผนภาพคลาส (Class Diagram) และแผนภาพซีเควนซ์ (Sequence Diagram) เป็นแผนภาพในภาษายูเอ็มแอลที่ผู้ออกแบบระบบใช้ในการส่งมอบต่อให้ผู้พัฒนาระบบ โดยข้อบกพร่องของโมเดลการออกแบบ (Design Flaw) [4] คือการที่คุณลักษณะทางโครงสร้างของเอนทิตีของโมเดลการออกแบบหรือชิ้นส่วนใดของโมเดลการออกแบบแสดงถึงการเบี่ยงเบนไปจากปัจจัยที่ทำให้โมเดลการออกแบบมีคุณภาพสูง และร่องรอยที่ไม่ดี(Bad-Smell) [5] หมายถึงลักษณะของการออกแบบหรือการพัฒนาโปรแกรมที่ไม่ดี ซึ่งจะส่งผลให้ซอฟต์แวร์ทำงานอย่างไม่มีประสิทธิภาพ โดยรีแฟคทอริง (Refactoring) [5] เป็นเทคนิคที่ใช้ในการเปลี่ยนโครงสร้างภายในของซอฟต์แวร์ ที่ไม่ทำให้พฤติกรรมของซอฟต์แวร์เปลี่ยนแปลงไป ซึ่งเป็นเทคนิคสำหรับแก้ไขข้อบกพร่องของโมเดลการออกแบบและร่องรอยที่ไม่ดี

จากที่กล่าวมาข้างต้น งานวิจัยนี้จึงนำเสนอวิธีการตรวจจับข้อบกพร่องของโมเดลการออกแบบ หรือร่องรอยที่ไม่ดี ที่เกิดขึ้นในขั้นตอนการออกแบบ 6 ประเภทคือ Data Class, Feature Envy, God Class, Message Chains, Middle Man และ Switch Statements โดยพิจารณาที่แผนภาพคลาสและแผนภาพซีเควนซ์ว่ามีข้อบกพร่องหรือร่องรอยที่ไม่ดีใดบ้าง ที่ส่งผลกระทบต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ โดยออกแบบวิธีการตรวจจับข้อบกพร่องหรือร่องรอยที่ไม่ดีด้วยมาตรวัดซอฟต์แวร์สำหรับขั้นตอนการออกแบบ รวมถึงพัฒนาเครื่องมือสำหรับตรวจจับข้อบกพร่องหรือร่องรอยที่ไม่ดี และเสนอวิธีรีแฟคทอริง สำหรับข้อบกพร่องหรือร่องรอยที่ไม่ดีแต่ละประเภท ผู้วิจัยได้นำวิธีรีแฟคทอริงมาประยุกต์ใช้ในการเปลี่ยนรูปแบบ (Transformation) แผนภาพคลาส

1.2 วัตถุประสงค์

1.2.1 เพื่อระบุข้อบกพร่องและร่องรอยที่ไม่ดีของโมเดลการออกแบบที่เกี่ยวข้องกับความสามารถในการเปลี่ยนแปลงซอฟต์แวร์

1.2.2 เพื่อเสนอมาตรวัดซอฟต์แวร์เชิงวัตถุสำหรับตรวจจับข้อบกพร่องและร่องรอยที่ไม่ดีของโมเดลการออกแบบ ที่เกี่ยวข้องกับความสามารถในการเปลี่ยนแปลงซอฟต์แวร์แต่ละประเภท

1.2.3 เพื่อระบุเทคนิคการรีแฟคทอริงสำหรับข้อบกพร่องและร่องรอยที่ไม่ดีของโมเดลการออกแบบแต่ละประเภท ที่เกี่ยวข้องกับความสามารถในการเปลี่ยนแปลงซอฟต์แวร์

1.3 ขอบเขตงานวิจัย

1.3.1 งานวิจัยนี้ระบุข้อบกพร่องของโมเดลการออกแบบที่เกี่ยวข้องกับความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ และเสนอมาตรวัดสำหรับตรวจจับข้อบกพร่องแต่ละประเภท

1.3.2 งานวิจัยนี้จะตรวจจับข้อบกพร่องของโมเดลการออกแบบที่แผนภาพคลาสและแผนภาพซีเควนซ์ โดยแผนภาพทั้งสองจะต้องเขียนอย่างถูกต้องตามหลักการเขียนภาษายูเอ็มแอลเวอร์ชัน 2.0 และจะต้องมีข้อมูลที่แสดงถึงการเรียกใช้คุณลักษณะระหว่างคลาส และงานวิจัยนี้จะแนะนำวิธีการรีแฟคทอริงที่เหมาะสมโดยประยุกต์ใช้กับแผนภาพคลาสเท่านั้น

1.3.3 งานวิจัยนี้จะประเมินข้อบกพร่องของโมเดลการออกแบบรวมถึงมาตรวัดสำหรับตรวจจับ โดยเปรียบเทียบค่าของมาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการรีแฟคทอริง

1.3.4 แผนภาพตัวอย่างที่จะใช้ในการหาช่วงของข้อบกพร่องของโมเดลการออกแบบจะมีอย่างน้อย 35 ระบบ แต่ละระบบประกอบด้วย แผนภาพคลาสที่มีตั้งแต่10 คลาสขึ้นไป

1.3.5 แผนภาพตัวอย่างที่จะใช้ทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบจะมีอย่างน้อย 5 ระบบ แต่ละระบบประกอบด้วย แผนภาพคลาสที่มีตั้งแต่10 คลาสขึ้นไป

1.4 ขั้นตอนการวิจัย

1.4.1 ศึกษาแบบซอฟต์แวร์ และข้อบกพร่องของโมเดลการออกแบบ รวมถึงร่องรอยที่ไม่ดีของโมเดลการออกแบบประเภทต่าง ๆ

1.4.2 ศึกษามาตรวัดซอฟต์แวร์เชิงวัตถุสำหรับแผนภาพคลาสและแผนภาพซีเควนซ์

1.4.3 ศึกษาเทคนิครีแฟคทอริงสำหรับร่องรอยไม่ดีประเภทต่าง ๆ

1.4.4 เลือกข้อบกพร่องของโมเดลการออกแบบและร่องร่อยที่ไม่ดีของโมเดลการออกแบบที่เกี่ยวข้องกับความสามารถในการเปลี่ยนแปลงซอฟต์แวร์

1.4.5 เสนอมาตรวัดสำหรับข้อบกพร่องของโมเดลการออกแบบและร่องรอยที่ไม่ดีของโมเดลการออกแบบแต่ละประเภท

1.4.6 กำหนดช่วงของมาตรวัดข้อบกพร่องของโมเดลการออกแบบและร่องรอยที่ไม่ดีของโมเดลการออกแบบแต่ละประเภท

1.4.7 แนะนำวิธีการรีแฟคทอริงสำหรับข้อบกพร่องของโมเดลการออกแบบและร่องรอยที่ไม่ดีของโมเดลการออกแบบแต่ละประเภท

1.4.8 ประเมินข้อบกพร่องของโมเดลการออกแบบและ ร่องรอยที่ไม่ดีของโมเดลการออกแบบรวมถึงมาตรวัดสำหรับตรวจจับ

1.4.9 สรุปผลการวิจัย

1.4.10 จัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่จะได้รับ

1.5.1 สามารถนำข้อบกพร่องของโมเดลการออกแบบและร่องรอยที่ไม่ดีของโมเดลการออกแบบ มาใช้ในการรีแฟคทอริงเพื่อปรับปรุงความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ได้

1.5.2 สามารถนำมาตรวัดมาใช้ในการตรวจจับข้อบกพร่องของโมเดลการออกแบบ และร่องรอยที่ไม่ดีของโมเดลการออกแบบที่ส่งผลต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์

1.5.3 สามารถนำเทคนิครีแฟคทอริงมาช่วยปรับปรุงความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ในขั้นตอนการออกแบบได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องในงานวิจัย ได้แก่ โมเดลการออกแบบที่แสดงด้วยแผนภาพยูเอ็มแอล, การวัดซอฟต์แวร์, ข้อบกพร่องของโมเดลการออกแบบ, ร่องรอยที่ไม่ดี, รีแฟคทอริง, ความสามารถในการเปลี่ยนแปลง, และวิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่ ซึ่งมีรายละเอียดดังนี้

2.1.1 โมเดลการออกแบบที่แสดงด้วยแผนภาพยูเอ็มแอล

ในกระบวนการพัฒนาซอฟต์แวร์เชิงวัตถุ ผลที่ได้จากขั้นตอนการออกแบบนั้นได้แก่ โมเดลการออกแบบ ซึ่งแผนภาพยูเอ็มแอล [3] เป็นแผนภาพที่นิยมใช้แสดงโมเดลการออกแบบ เพื่อส่งต่อให้ผู้พัฒนาระบบต่อไป

ภาษายูเอ็มแอล [3] เป็นเทคนิคที่ใช้แผนภาพพื้นฐาน วัตถุประสงค์ของภาษายูเอ็มแอลนั้นเพื่อช่วยกำหนดศัพท์ทั่วไปในเชิงวัตถุและเทคนิคการแผนภาพ ที่สามารถใช้แสดงแบบจำลองของระบบตั้งแต่ขั้นตอนการออกแบบจนถึงการพัฒนาระบบ แผนภาพยูเอ็มแอล สำหรับภาษายูเอ็มแอลเวอร์ชัน 2.0 (UML version 2.0) ได้แบ่งประเภทของแผนภาพยูเอ็มแอลออกเป็น 2 กลุ่มใหญ่ ๆ ดังนี้

2.1.1.1 แผนภาพโครงสร้าง (Structure Diagrams)

แผนภาพนี้ใช้เพื่อแสดงแบบจำลองโครงสร้างของระบบ แผนภาพโครงสร้างประกอบด้วย

- แผนภาพคลาส ใช้แสดงความสัมพันธ์ระหว่างคลาสที่อยู่ในแบบจำลองของระบบ
- แผนภาพออบเจ็กต์ (Object Diagram) ใช้แสดงความสัมพันธ์ระหว่างออบเจ็กต์ที่อยู่ในแบบจำลองของระบบ ซึ่งมักจะใช้เมื่ออินสแตนท์ของคลาสสามารถสื่อให้เข้าใจแบบจำลองได้ดีกว่า
- แผนภาพแพคเกจ (Package Diagram) ใช้จัดกลุ่มยูเอ็มแอลส่วนย่อย (UML Elements) ไว้ด้วยกัน

- แผนภาพดีพลอยเมนท์ (Deployment Diagram) ใช้แสดงสถาปัตยกรรมของระบบทางกายภาพ หรือส่วนประกอบต่าง ๆ ของซอฟต์แวร์ (Software Component) ที่จะติดตั้งบนสถาปัตยกรรม
- แผนภาพคอมโพเนนท์ (Component Diagram) ใช้แสดงความสัมพันธ์ทางกายภาพระหว่างส่วนประกอบต่าง ๆ ของซอฟต์แวร์
- แผนภาพคอมโพสิทสตรัคเชอร์ (Composite Structure) ใช้แสดงโครงสร้างภายในของคลาส ได้แก่ ความสัมพันธ์ระหว่างส่วนต่าง ๆ ของคลาส

2.1.1.2 **แผนภาพพฤติกรรม** (Behavioral Diagrams)

แผนภาพนี้ใช้แสดงแบบจำลองพฤติกรรมของระบบ แผนภาพพฤติกรรมประกอบด้วย

- แผนภาพแอคทิวิตี (Activity Diagram) ใช้แสดงกระแสงานทางธุรกิจ (Business Workflow) ที่ไม่ขึ้นกับคลาส แสดงการดำเนินงานตามแอคทิวิตีของยูสเคส (Use Case) หรือโมเดลการออกแบบรายละเอียด (Detailed Design) ของเมทธอด
- แผนภาพซีเควนซ์ (Sequence Diagram) แสดงแบบจำลองพฤติกรรมของออบเจ็กต์ที่อยู่ในยูสเคส โดยเน้นที่ลำดับการเกิดแอคทิวิตีตามเวลา
- แผนภาพคอมมิวนิเคชัน (Communication Diagram) แสดงแบบจำลองพฤติกรรมของออบเจ็กต์ที่อยู่ในยูสเคส โดยเน้นที่การสื่อสารกันระหว่างกลุ่มของออบเจ็กต์ที่ทำงานร่วมกันสำหรับแอคทิวิตีใด ๆ
- แผนภาพอินเทอแรคชันโอเวอร์วิว (Interaction Overview Timing) แสดงการอธิบายโดยสรุปของการไหลของการควบคุมของโพรเซส (Flow of control of a process)
- แผนภาพไทม์มิง (Timing Diagram) แสดงการปฏิสัมพันธ์ (Interaction) ที่เกิดขึ้นระหว่างกลุ่มของออบเจ็กต์และการเปลี่ยนแปลงสถานะ (State) ตลอดช่วงเวลา
- แผนภาพบิเฮวิออรอลสเตทแมชชีน (Behavioral State Machine Diagram) แสดงพฤติกรรมของคลาสหนึ่ง ๆ
- แผนภาพโปรโตคอลสเตทแมชชีน (Protocol State Machine Diagram) แสดงการขึ้นต่อกันระหว่างอินเตอร์เฟซ (Interfaces) ของคลาส

- แผนภาพยูสเคส (Use Case Diagram) แสดงความต้องการทางธุรกิจสำหรับระบบ และแสดงการปฏิสัมพันธ์ระหว่างระบบและสิ่งแวดล้อม

เนื่องจากงานวิจัยนี้เน้นพิจารณาแผนภาพคลาส และแผนภาพซีเควนซ์ จึงอธิบายแผนภาพทั้งสองโดยมีรายละเอียดดังนี้

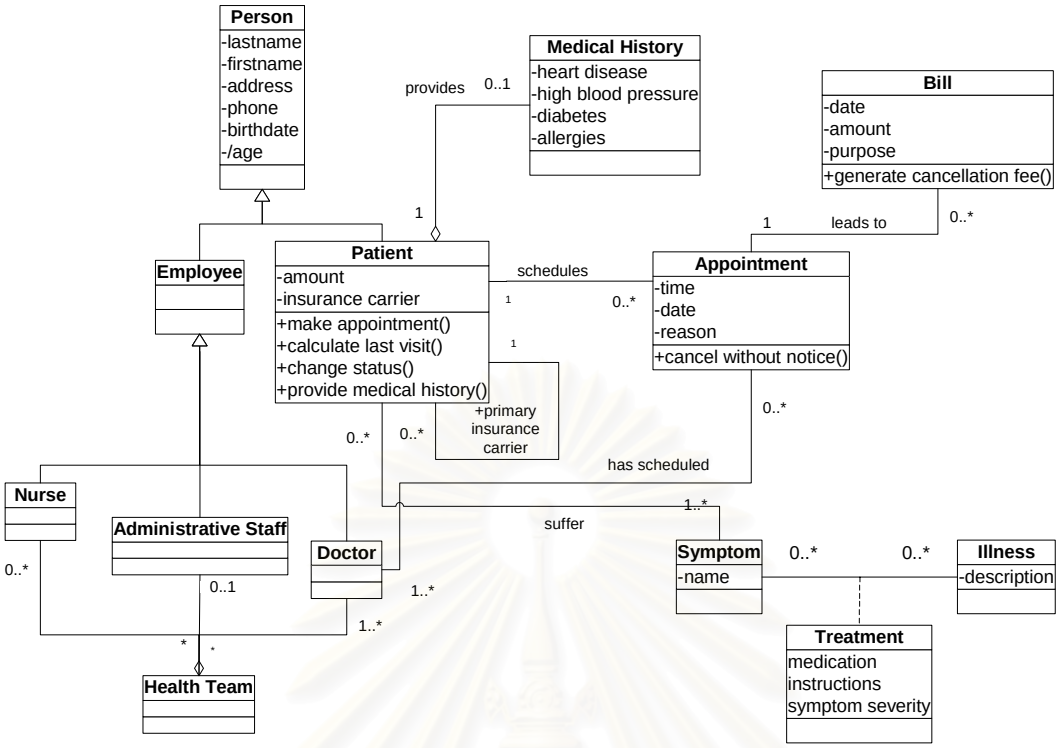
แผนภาพคลาส

แผนภาพคลาสใช้เพื่อแทนสิ่งของ หรือแนวความคิด (Concept) ที่มีอยู่ในระบบ และแสดงความสัมพันธ์ระหว่างคลาสต่าง ๆ ในระบบ วัตถุประสงค์ของแผนภาพคลาสคือเพื่อใช้ในการกำหนดศัพท์ที่ใช้ร่วมกันระหว่างผู้ออกแบบระบบและผู้ใช้งานระบบ แผนภาพคลาสเป็นแบบจำลองที่เป็นแบบสถิตย์ ซึ่งตัวอย่างแผนภาพคลาส [3] แสดงในรูปที่ 2.1 แผนภาพคลาสมีส่วนประกอบต่าง ๆ ดังนี้

- คลาส (Class) ใช้เก็บข้อมูลของระบบ โดยที่ภายในคลาสจะประกอบด้วยชื่อคลาส คุณลักษณะ และโอเปอร์เรชัน
- ความสัมพันธ์ (Relationships) บอกถึงความสัมพันธ์หรือการเชื่อมโยงกัน (Association) ของคลาสต่าง ๆ ความสัมพันธ์สามารถบอกถึงมัลติพลิซิตี (Multiplicity) ที่จะระบุว่าอินสแตนท์มีการเชื่อมโยงกันกับอินสแตนท์อื่นอย่างไร จะแสดงโดยใช้ตัวเลขเพื่อบอกจำนวนมากที่สุดหรือน้อยที่สุดของอินสแตนท์ที่จะมีการเชื่อมโยงกันได้สำหรับเส้นความสัมพันธ์นั้น ๆ

  การเชื่อมโยงกันแบบพิเศษมี 2 แบบได้แก่

1. เจเนอรัลไรเซชัน (Generalization) จะแสดงให้เห็นว่าคลาสลูก (Subclass) สืบทอดคุณสมบัติจากคลาสบรรพบุรุษ (Superclass) ทำให้คลาสลูกสามารถใช้คุณลักษณะและโอเปอร์เรชันแบบเดียวกับคลาสบรรพบุรุษได้ วิธีการสังเกตเจเนอรัลไรเซชันคือเมื่อคลาสใดมีความสัมพันธ์ในลักษณะที่เป็น “เป็นชนิดหนึ่งของ (is-a-kind-of)” กับคลาสอีกคลาสหนึ่ง
2. แอกกริเกชัน (Aggregation) ใช้เมื่อคลาสหนึ่งประกอบด้วยอีกคลาสหนึ่ง วิธีการสังเกตแอกกริเกชัน คือเมื่อคลาสใดมีความสัมพันธ์ในลักษณะที่เป็น “เป็นส่วนประกอบของ (is-a-part-of)” หรือ “(เป็นส่วนหนึ่งของ)is-made-up-of” กับคลาสอีกคลาสหนึ่ง



รูปที่ 2.1 ตัวอย่างแผนภาพคลาสสำหรับ Appointment System

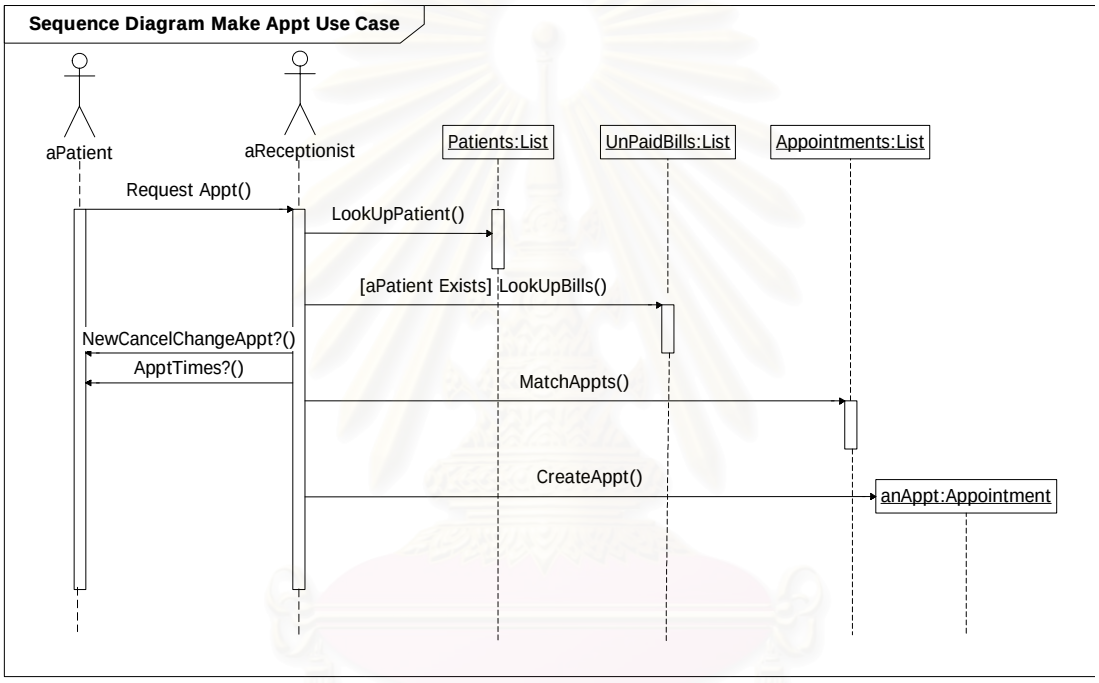
แผนภาพซีเควนซ์

แผนภาพซีเควนซ์ใช้แสดงการปฏิสัมพันธ์ระหว่างออบเจ็กต์ในระบบ แผนภาพซีเควนซ์เป็นแผนภาพที่แสดงการปฏิสัมพันธ์ที่เป็นที่นิยมที่สุดในการแสดงแบบจำลองเชิงวัตถุของระบบ โดยเน้นที่ลำดับการเกิดแอคทิวิตีตามเวลาที่เกิดขึ้น แผนภาพซีเควนซ์แสดงให้เห็นถึงการส่งข้อความ (Message) ระหว่างออบเจ็กต์สำหรับหนึ่งยูสเคส โดยแสดงเป็นแบบจำลองแบบพลวัตที่บอกถึงลำดับการส่งข้อความระหว่างออบเจ็กต์ในการปฏิสัมพันธ์ แผนภาพซีเควนซ์ประกอบด้วยส่วนต่าง ๆ ดังนี้

- แอคเตอร์ (Actor) หมายถึงคนหรือระบบที่อยู่นอกระบบที่พิจารณา และเป็นผู้มาเรียกใช้งานหรือมีการติดต่อกับระบบที่พิจารณา
- ออบเจ็กต์ คืออินสแตนท์ของคลาสที่เป็นผู้ส่งหรือรับข้อความ
- ไลฟ์ไลน์ (Lifeline) บอกช่วงชีวิตของออบเจ็กต์
- เอกซิคิวชัน ออคเคอเรนซ์ (Execution Occurrence) บอกให้รู้ว่าเวลาใดที่ออบเจ็กต์มีการส่งและรับข้อความ
- ข้อความ คือข้อมูลที่ออบเจ็กต์ส่งไปเพื่อสื่อสารกับออบเจ็กต์อื่น

- ออบเจ็กต์เดสตรัคชัน (Object Destruction) ใช้เพื่อบอกการสิ้นสุดของช่วงชีวิตของออบเจ็กต์
- เฟรม (Frame) บอกบริบท (Context) ของแผนภาพซีเควนซ์

การเขียนแผนภาพซีเควนซ์สามารถเขียนได้ทั้งแบบที่เป็นเจเนอริก (Generic Sequence Diagram) ที่ครอบคลุมทุกซีนาริโอ (Scenario) หรือเขียนแสดงสำหรับแต่ละอินสแตนท์ (Instance Sequence Diagram) เพื่อแยกแผนภาพสำหรับแต่ละซีนาริโอก็ได้ ตัวอย่างแผนภาพซีเควนซ์ [3] แสดงในรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างแผนภาพซีเควนซ์สำหรับยูสเคส Make Appointment

2.1.2 การวัดซอฟต์แวร์

การวัด (Measurement) [6] เป็นสิ่งจำเป็นสำหรับทุกศาสตร์ทางวิศวกรรม โดยที่วัตถุประสงค์ของการวัดอาจแตกต่างกันออกไปแล้วแต่มุมมองของผู้ที่ทำการวัด

การวัดคือกระบวนการกำหนดตัวเลขหรือสัญลักษณ์ (Symbol) ให้กับคุณลักษณะ (Attributes) ของเอนทิตี (Entities) ในโลกแห่งความเป็นจริง เพื่ออธิบายเอนทิตีให้สอดคล้องกับกฎที่ได้มีการกำหนดไว้ชัดเจนอยู่แล้ว โดยที่เอนทิตีคือวัตถุที่มีอยู่จริงบนโลก และคุณลักษณะคือคุณสมบัติของเอนทิตี

การวัดในทางวิศวกรรมซอฟต์แวร์ (Software Engineering) สามารถแบ่งได้เป็น 2 ประเภท ได้แก่

2.1.2.1 การวัดทางตรง (Direct measurement) เป็นการวัดเฉพาะคุณลักษณะของสิ่งที่เราสนใจโดยไม่นำคุณลักษณะ หรือ เอนทิตีอื่นมาเกี่ยวข้อง การวัดทางตรงทำให้ทราบลักษณะในด้านโครงสร้างของซอฟต์แวร์ เช่น การวัดความยาวของรหัสต้นทาง (Source Code) สามารถวัดได้จากการนับจำนวนบรรทัดทั้งหมดของโปรแกรม การวัดจำนวนข้อบกพร่องที่ค้นพบในขั้นตอนการทดสอบระบบ จะวัดได้จากการนับข้อบกพร่อง หรือการวัดเวลาที่ผู้พัฒนาระบบใช้ในโครงการ วัดได้จากจำนวนเดือนที่ใช้ในการพัฒนาระบบเป็นต้น

2.1.2.2 การวัดทางอ้อม (Indirect measurement) เป็นการวัดโดยการนำการวัดทางตรงตั้งแต่หนึ่งประเภทขึ้นไปมาพิจารณาประกอบกัน เนื่องจากคุณลักษณะบางอย่างไม่สามารถวัดได้โดยตรง ซึ่งสิ่งที่ได้จากการวัดทางอ้อมก็คือคุณภาพของซอฟต์แวร์นั่นเอง ตัวอย่างการวัดทางอ้อม เช่น การวัดความสามารถในการทดสอบ (Testability) การวัดความสามารถในการเข้าใจ (Understandability) เป็นต้น

มาตรวัดเป็นสิ่งจำเป็นเพื่อใช้ในการวัดทั้งสองประเภทโดยจะวัดคุณลักษณะของสิ่งที่เราสนใจ มาตรวัดเชิงวัตถุที่เป็นที่นิยมแพร่หลายได้แก่ ระดับความลึกของการสืบทอดคุณสมบัติของคลาส (Depth of Inheritance Tree – DIT) จำนวนคลาสลูก (Number Of Children – NOC) และการขึ้นต่อกันระหว่างออบเจ็กต์ (Coupling Between Object – CBO) และการขาดการเกาะกลุ่มกันของเมทธอด (Lack of Cohesion Method: LCOM) เป็นต้น

นอกจากนี้ยังมีมาตรวัดเชิงวัตถุสำหรับโมเดลการออกแบบโดยใช้วัดที่แผนภาพคลาส เช่น จำนวนโครงสร้างลำดับชั้นของความสัมพันธ์แบบแอกกริเกชัน (Number of Aggreagation Hierarchies: NaggH) จำนวนโครงสร้างลำดับชั้นของความสัมพันธ์แบบเจนเนอรัลไลเซชัน (Number of Generalization Hierarchies: NGenH) และมาตรวัดสำหรับแผนภาพซีเควนซ์ เช่น จำนวนซีนาริโอ (Number of Scenarios: NOS) จำนวนข้อความระหว่างออบเจ็กต์ (Weighted Messages between Objects: WMBO) เป็นต้น หลังจากหาค่ามาตรวัดต่างๆ ได้แล้ว จะต้องนำมาตรวัดเหล่านี้ มาทำการตรวจสอบ (Validation) ว่าค่ามาตรวัดที่ได้มีความน่าเชื่อถือ (Reliability) สามารถนำไปใช้วัดคุณภาพซอฟต์แวร์ต่อไปได้

2.1.3 ข้อบกพร่องของโมเดลการออกแบบ

ในงานวิจัย [7] ได้ให้คำนิยามไว้ว่า ข้อบกพร่องของโมเดลการออกแบบคือการที่คุณลักษณะทางโครงสร้างของเอนทิตีของโมเดลการออกแบบหรือชิ้นส่วนใดของโมเดลการออกแบบแสดงถึงการเบี่ยงเบนไปจากปัจจัยที่ทำให้โมเดลการออกแบบมีคุณภาพสูง โดยขยายความส่วนต่าง ๆ ของคำนิยามไว้ดังนี้

คุณลักษณะทางโครงสร้าง

เนื่องจากการเกิดข้อบกพร่องโดยทั่วไปที่กระทบต่อคุณภาพนั้นเกิดจากหลายสาเหตุเช่น การพัฒนาที่ผิดพลาด (Erroneous Implementation) การพัฒนาที่อาจก่อให้เกิดการผิดพลาด (Error-prone Implementation) การจัดทำเอกสารที่ไม่ครบถ้วน (Missing Documentation) หรือการจัดทำเอกสารไม่สัมพันธ์กัน (Incoherent Documentation) แต่ข้อบกพร่องตามคำนิยามในงานวิจัยนี้ จะพิจารณาเฉพาะข้อบกพร่องที่สอดคล้องกับโครงสร้างของโมเดลการออกแบบเท่านั้น ไม่รวมถึงข้อบกพร่องในการพัฒนาหรือการจัดทำเอกสาร

คุณลักษณะของเอนทิตี

ในที่นี้หมายถึงคุณสมบัติของเอนทิตี โดยที่ข้อบกพร่องของโมเดลการออกแบบหมายถึงคุณสมบัติที่เป็นผลเสียต่อเอนทิตี และสำหรับเอนทิตีใดๆ อาจมีข้อบกพร่องได้มากกว่าหนึ่งประเภท

เอนทิตีของโมเดลการออกแบบ และชิ้นส่วนของโมเดลการออกแบบ

เนื่องจากเอนทิตีของโมเดลการออกแบบได้แก่เมทธอด คลาส หรือแพคเกจ (Package) แต่ข้อบกพร่องของโมเดลการออกแบบอาจจะไม่ได้เกิดขึ้นที่เอนทิตีใดเอนทิตีหนึ่ง แต่จะกระทบกับหลาย ๆ เอนทิตีที่เกี่ยวข้องกัน ดังนั้น ชิ้นส่วนของโมเดลการออกแบบหมายถึงกลุ่มของเอนทิตีที่มีการทำงานร่วมกัน

ปัจจัยที่ทำให้โมเดลการออกแบบมีคุณภาพสูง

มาริเนสคูกล่าวว่า โมเดลการออกแบบที่ดีจะอธิบายได้ด้วยชุดของปัจจัยต่างๆ ตัวอย่างของปัจจัยในการออกแบบที่ดีเช่น การขึ้นต่อกันน้อย (Low Coupling) การเข้าคู่กันมาก (High Cohesion) เป็นต้น

การเบี่ยงเบนจากปัจจัยที่บอกถึงโมเดลการออกแบบที่ดี

การระบุถึงความเบี่ยงเบนจากปัจจัยนั้นจำเป็นต้องอาศัยการอธิบายถึงปัจจัยให้เป็นเชิงคุณภาพ โดยมาริเนสคูได้นำเสนอวิธีการกำหนดปัจจัยต่าง ๆ ด้วยการใช้ดีเทคชันสเตรททีจีส์ (Detection Strategies) ซึ่งจะเสนอปัจจัยที่บอกถึงโมเดลการออกแบบที่ดีจากคำแนะนำหรือกฎของการออกแบบ

ข้อบกพร่องของโมเดลการออกแบบแบ่งออกเป็น 4 ประเภท [7] ได้แก่

2.1.3.1 ข้อบกพร่องที่ระดับเมทธอด ตัวอย่างข้อบกพร่องประเภทนี้ได้แก่

- Feature Envy [5] ซึ่งได้แก่เมทธอดที่มีการเรียกใช้คุณลักษณะของคลาสอื่นมากกว่าคลาสตัวเอง
- God Method[5] ซึ่งได้แก่เมทธอดที่มีหน้าที่การทำงานมาก

2.1.3.2 ข้อบกพร่องที่ระดับคลาส ตัวอย่างข้อบกพร่องประเภทนี้ได้แก่

- Data Classes[5] [4] ได้แก่คลาสที่มีหน้าที่เก็บข้อมูลอย่างเดียวเท่านั้น ไม่มีหน้าที่อื่นอีก
- God Classes[5] ได้แก่คลาสที่มีหน้าที่ทำงานเป็นจำนวนมาก มีความซับซ้อน(Complexity) สูงมาก และความเข้าคู่กัน(Cohesion) ภายในคลาสต่ำ มีการขึ้นต่อกัน (Coupling) กับคลาสอื่นสูง
- Shotgun Surgery[5] คลาสที่เมื่อมีการเพิ่มหน้าที่การทำงานหนึ่งต้องทำการแก้ไขในคลาสอื่นหลายๆ คลาส
- Refused Bequest[5] ลักษณะที่คลาสลูกได้รับการถ่ายทอดคุณสมบัติจากคลาสแม่มาทั้งหมด ซึ่งจริง ๆ แล้วคลาสลูกไม่ต้องการใช้ ลักษณะเช่นนี้บ่งบอกถึงความผิดพลาดของลำดับชั้นของคลาส

2.1.3.3 ข้อบกพร่องที่ระดับแพคเกจ

- God Packages คือการที่แพคเกจมีขนาดใหญ่ และคลาสที่อยู่ภายในแพคเกจมีความเข้าคู่กันต่ำ
- Inflation of Atomic Packagesคือการที่แพคเกจมีขนาดเล็กมากประกอบด้วยคลาสจำนวนน้อยมากทำให้ต้องมีการเรียกใช้งานหรือมีความขึ้นต่อกันกับคลาสที่อยู่นอกแพคเกจ
- Misplaced Classes คือการที่คลาสต้องการใช้งานคลาสที่อยู่นอกแพคเกจมากกว่าคลาสที่อยู่แพคเกจเดียวกัน

2.1.3.4 ข้อบกพร่องที่คลัสเตอร์ของคลาส (Cluster of Classes) ตัวอย่างข้อบกพร่องประเภทนี้เกิดขึ้นในกรณีที่กลุ่มของโมเดลการออกแบบ (Design Fragment) ไม่ได้มีการประยุกต์ใช้ดีไซน์แพทเทิร์น (Design Pattern) ทั้งที่ควรจะมีการประยุกต์ใช้

งานวิจัยนี้จะตรวจจับข้อบกพร่องของโมเดลการออกแบบโดยพิจารณาข้อบกพร่องของโมเดลการออกแบบประเภท God Class

2.1.4 ร่องรอยที่ไม่ดี

ร่องรอยที่ไม่ดี [5] หมายถึงลักษณะของการออกแบบหรือการพัฒนาโปรแกรมที่ไม่ดี ซึ่งจะส่งผลให้ซอฟต์แวร์ทำงานอย่างไม่มีประสิทธิภาพ ทำให้ต้องแก้ไขโดยการรีแฟคทอริง เพื่อให้ซอฟต์แวร์มีคุณภาพเพิ่มมากขึ้น ซึ่งการใช้มาตรวัดเชิงวัตถุเข้ามาช่วยในการตรวจจับร่องรอยที่ไม่ดีนั้น เป็นวิธีที่นิยมวิธีหนึ่ง เนื่องจากมาตรวัดเชิงวัตถุจะสามารถระบุได้ว่า ซอฟต์แวร์มีคุณลักษณะที่บ่งบอกถึงร่องรอยที่ไม่ดีประเภทใด

ร่องรอยที่ไม่ดีสำหรับโปรแกรมเชิงวัตถุแต่ละประเภทจะมีวิธีการรีแฟคทอริงที่เหมาะสมที่จะใช้ในการกำจัดร่องรอยที่ไม่ดี ร่องรอยที่ไม่ดีสำหรับโปรแกรมเชิงวัตถุมีอยู่ 22 ประเภทได้แก่

2.1.4.1 Duplicate Code คือ การที่รหัสต้นทางมีโครงสร้างเหมือนกันมากกว่า 1 ที่

2.1.4.2 Long Method คือ ลักษณะที่เมทธอดมีขนาดใหญ่

2.1.4.3 Large Class คือ คลาสที่มีหน้าที่การทำงาน และจำนวนออบเจ็กต์มาก

2.1.4.4 Long Parameter List คือ ในการเรียกใช้งานออบเจ็กต์อื่น มีการส่งผ่าน2.1.4.1 พารามิเตอร์ (Parameter) จำนวนมาก ซึ่งตามหลักการของการออกแบบเชิงวัตถุนั้น ออบเจ็กต์ควรจะส่งผ่านพารามิเตอร์เท่าที่จำเป็นเท่านั้น

2.1.4.5 Divergent Change คือ ลักษณะที่คลาสหนึ่งมักจะต้องมีการเปลี่ยนแปลงหลายๆ อย่างเกิดขึ้น

2.1.4.6 Shotgun Surgery มีลักษณะที่ตรงข้ามกับ Divergent Change คือ เมื่อมีการเพิ่มหน้าที่การทำงานหนึ่งต้องทำการแก้ไขในคลาสอื่นหลายๆ คลาส

2.1.4.7 Feature Envy คือ ลักษณะที่เมทธอดมีการเรียกใช้คุณสมบัติในคลาสอื่นมากกว่าคลาสเจ้าของเมทธอดเอง

2.1.4.8 Data Clumps คือลักษณะที่มีการใช้กลุ่มข้อมูลชุดเดียวกันในหลายๆ ที่ภายในรหัสต้นทาง ซึ่งกลุ่มของข้อมูลในที่นี้ประกอบด้วย คุณลักษณะภายในคลาส หรือพารามิเตอร์ในเมทธอด

2.1.4.9 Primitive Obsession คือ การใช้ข้อมูลชนิดพื้นฐานแทนคลาสขนาดเล็ก ทำให้คลาสข้อมูลพื้นฐานมีขนาดใหญ่

2.1.4.10 Switch Statement คือ ลักษณะที่มีสวิทช์สเตทเมนท์ (Switch statement) เหมือนกันในรหัสต้นทางหลายๆ ที่ ทำให้เกิดรหัสต้นทางที่ซ้ำกันได้

2.1.4.11 Parallel Inheritance Hierarchy คือ เมื่อมีการสร้างคลาสลูกของคลาสหนึ่ง จะทำให้ต้องสร้างคลาสลูกของอีกคลาสหนึ่งด้วย ซึ่งจริง ๆ แล้ว ร่องรอยที่ไม่ดีนี้คือกรณีพิเศษของ Shotgun Surgery

2.1.4.12 Lazy Class คือ คลาสที่ไม่ได้ทำหน้าที่หลักๆ มาก และควรกำจัดออกไป

2.1.4.13 Speculative Generality คือ ลักษณะที่มีการสร้างคลาสหรือเมทธอดขึ้นมาเป็นกรณีทดสอบ ซึ่งเป็นกรณีที่รหัสต้นทางที่ไม่จำเป็นถูกสร้างขึ้นจากการคาดการณ์ถึงการเปลี่ยนแปลงล่วงหน้าของซอฟต์แวร์ ทำให้เพิ่มความซับซ้อนให้กับซอฟต์แวร์

2.1.4.14 Temporary Field คือ การที่อ็อบเจกต์ที่มีตัวแปรที่ใช้สำหรับเก็บค่าผลลัพธ์ชั่วคราวในรหัสต้นทาง

2.1.4.15 Message Chains คือ การที่มีการเรียกใช้ข้อมูลของอ็อบเจกต์ หนึ่งจากอ็อบเจกต์อื่น ซึ่งไม่ได้เรียกใช้จากอ็อบเจกต์นั้นโดยตรง และมีการเรียกใช้งานออบเจ็กต์ซึ่งไม่ได้เรียกใช้งานจากอ็อบเจกต์โดยตรงต่อกันเป็นทอด ๆ

2.1.4.16 Middle Man คือ คลาสที่ทำหน้าที่เป็นตัวกลางการติดต่อระหว่างออบเจ็กต์

2.1.4.17 Inappropriate Intimacy คือ ลักษณะที่มีการพยายามเรียกใช้ส่วนไพรเวท (Private) ระหว่างคลาส

2.1.4.18 Alternative Classes with Difference Interfaces คือ การที่มีเมทธอดที่ทำหน้าที่เดียวกันแต่มีส่วนซิกเนเจอร์ (Signature) ต่างกัน

2.1.4.19 Incomplete Library Class คือ การที่มีไลบรารีคลาส (Library Class) ที่ไม่สมบูรณ์ เมื่อซอฟต์แวร์นำไลบรารีคลาสมาใช้ ทำให้การทำความเข้าใจคลาสเพื่อนำกลับมาใช้ใหม่ทำได้ยากขึ้น

2.1.4.20 Data Class คือ คลาสที่มีหน้าที่การทำงานหลักเป็นการกำหนดค่าให้กับข้อมูลและภายในคลาสมีเพียงเอคเซสเซอร์เมทธอด (Accessor method) เท่านั้น

2.1.4.21 Refused Bequest คือ การที่คลาสลูกได้รับการถ่ายทอดคุณสมบัติจากคลาสแม่มาทั้งหมด ซึ่งจริง ๆ แล้วคลาสลูกไม่ต้องการใช้

2.1.4.22 Comment การเขียนคำอธิบายนั้น ไม่ได้เป็นร่องรอยที่ไม่ดี แต่มักมีการเขียนอธิบายอยู่บริเวณที่มีร่องรอยไม่ดีของรหัสต้นทาง เมื่อมีการแก้ไขร่องรอยที่ไม่ดีข้างต้นด้วยวิธีการ รีแฟคทอริงแล้ว การเขียนอธิบายนั้นก็ไม่มีความจำเป็นอีกแล้ว

ตัวอย่างการกำจัดร่องรอยที่ไม่ดีด้วยวิธีการรีแฟคทอริงเช่น ร่องรอยที่ไม่ดีประเภท Feature Envy คือ การที่เมทธอดไปเรียกใช้หรือถูกใช้โดยคลาสอื่นมากกว่าคลาสเจ้าของเมทธอดเอง ดังนั้นเราสามารถกำจัด Feature Envyได้โดยการประยุกต์ใช้ วิธีการ Move Method ซึ่งเป็นการย้ายเมทธอดจากคลาสหนึ่ง ไปยังอีกคลาสหนึ่ง แล้วลบเมทธอดในคลาสเดิมทิ้งไปหรือให้ทำงาน อย่างอื่นแทน

งานวิจัยนี้จะตรวจจับข้อบกพร่องของโมเดลการออกแบบโดยพิจารณาร่องรอยที่ไม่ดี 5 ประเภทได้แก่ Data Class, Feature Envy, Message Chains, Middle Man, และ Switch Statements

2.1.5 **รีแฟคทอริง**

รีแฟคทอริง [5] หมายถึง การเปลี่ยนแปลงโครงสร้างภายในของซอฟต์แวร์ โดยไม่ทำให้พฤติกรรมภายนอกของซอฟต์แวร์เปลี่ยนแปลงไป โดยมีวัตถุประสงค์เพื่อทำให้ซอฟต์แวร์นั้นง่ายต่อการทำความเข้าใจ ซึ่งจะเพิ่มความสามารถในการบำรุงรักษาซอฟต์แวร์

การรีแฟคทอริงอย่างเหมาะสม จะสามารถช่วยให้การออกแบบซอฟต์แวร์ ทำความเข้าใจได้ง่ายขึ้น ค้นพบข้อผิดพลาดที่ซ่อนอยู่ภายในโปรแกรม (Bugs) และช่วยให้โปรแกรมทำงานได้เร็วขึ้น

งานวิจัย [8] ได้อธิบายถึงขั้นตอนในการทำรีแฟคทอริงดังนี้

2.1.5.1 การวางแผนการปรับปรุงซอฟต์แวร์ (Improvement Planning) ประกอบด้วย 3 ขั้นตอน คือ

1. การตรวจจับร่องรอยที่ไม่ดี (Bad-smell detection)
2. การวิเคราะห์ร่องรอยที่ไม่ดี (Bad-smell analysis)
3. การวางแผนการทำรีแฟคทอริง (Refactoring planning)

2.1.5.2 การตรวจสอบความเหมาะสมของการปรับปรุงซอฟต์แวร์ (Improvement Validation) แบ่งเป็น 3 ขั้นตอน คือ

1. การประเมินแผนการทำรีแฟคทอริง (Plan evaluation)
2. การตรวจสอบความเหมาะสมของการทำรีแฟคทอริง (Refactoring validation)

3. การตรวจสอบความเท่าเทียมกันของฟังก์ชัน (Functional evaluation validation)

2.1.5.3 การกระทำการปรับปรุงซอฟต์แวร์ (Improvement Execution) แบ่งเป็น 2 ขั้นตอน คือ

1. การเตรียมการรีแฟคทอริง (Refactoring deployment)
2. การประยุกต์ใช้รีแฟคทอริง (Refactoring Application)

วิธีการรีแฟคทอริงมีทั้งหมด 72 วิธี ซึ่งการเลือกว่าจะทำการรีแฟคทอริงที่ใด การเลือกประเภทวิธีการ และระบุโครงสร้างภายในที่ควรทำรีแฟคทอริงสามารถทำได้จากการพิจารณาโดยใช้วิจารณญาณส่วนตัว (Subjective perception) และพิจารณาจากร่องรอยที่ไม่ดี

2.1.6 ความสามารถในการเปลี่ยนแปลง

ความสามารถในการเปลี่ยนแปลง [2] หมายถึง ดีกรี (Degree) ที่ระบบหรือส่วนประกอบของระบบเอื้อต่อการเปลี่ยนแปลงเมื่อมีความต้องการเปลี่ยนแปลงเกิดขึ้น

ในโมเดลคุณภาพ (Quality Model) จาก [2] ความสามารถในการเปลี่ยนแปลงเป็นหนึ่งในคุณลักษณะระดับกลาง ของคุณภาพทางด้านการบำรุงรักษา ของซอฟต์แวร์

งานวิจัยนี้มีการทดสอบผลของข้อบกพร่องหรือร่องรอยที่ไม่ดีที่กระทบต่อความสามารถในการเปลี่ยนแปลง โดยมีการวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการตรวจจับข้อบกพร่องหรือร่องรอยที่ไม่ดี ก่อนและหลังการทำรีแฟคทอริงด้วยโมเดลการวัดความสามารถในการเปลี่ยนแปลงจากงานวิจัย [9] ซึ่งโมเดลการวัดความสามารถในการเปลี่ยนแปลงที่ใช้ในงานวิจัยนี้แบ่งออกเป็นโมเดลการวัดความถูกต้องของการเปลี่ยนแปลง และโมเดลการวัดความสมบูรณ์ของการเปลี่ยนแปลงซึ่งมีรายละเอียดดังนี้

2.1.6.1 โมเดลการวัดความถูกต้องของการเปลี่ยนแปลง (Modifiability Correctness Models)

ความถูกต้องของการเปลี่ยนแปลง(Modifiability Correctness) หมายถึงอัตราส่วนระหว่างจำนวนการกระทำการเปลี่ยนแปลงในโมเดลการออกแบบที่ถูกต้อง (Number of Correct Modifications) ต่อ จำนวนการกระทำการเปลี่ยนแปลงในโมเดลการออกแบบทั้งหมด (Number of Modifications Applied) โดยการเปลี่ยนแปลงจะกระทำที่แผนภาพคลาส ความถูกต้องของการเปลี่ยนแปลงสามารถอธิบายด้วยสมการได้ดังนี้

$$\text{ความถูกต้องของการเปลี่ยนแปลง} = \frac{NumberofCorrectModifications}{NumberofModificationApplied}$$

โมเดลการวัดความถูกต้องของการเปลี่ยนแปลงมีดังนี้

1. โมเดลการวัดความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 * NAggH
   โดยโมเดลนี้มีระดับความเชื่อมั่นเป็น 0.999 (p-value มีค่าเท่ากับ 0.001) และมีค่าสัมประสิทธิ์การตัดสินใจ (Coefficient of Determination) เท่ากับ 0.408
2. โมเดลการวัดความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 * NGenH
   โมเดลนี้มีระดับความเชื่อมั่นเป็น 0.98 (p-value มีค่าเท่ากับ 0.02) และมีค่าสัมประสิทธิ์การตัดสินใจ เท่ากับ 0.205

2.1.6.2 โมเดลการวัดความสมบูรณ์ของการเปลี่ยนแปลง (Modifiability Completeness Models)

ความสมบูรณ์ของการเปลี่ยนแปลง(Modification Completeness) หมายถึงอัตราส่วนระหว่างจำนวนการกระทำการเปลี่ยนแปลงในโมเดลการออกแบบที่ถูกต้อง (Number of Correct Modifications) ต่อ จำนวนการกระทำการเปลี่ยนแปลงในโมเดลการออกแบบที่ต้องกระทำ (Number of Modifications Required) โดยการเปลี่ยนแปลงจะกระทำที่แผนภาพคลาส ความสมบูรณ์ของการเปลี่ยนแปลงสามารถอธิบายด้วยสมการได้ดังนี้

$$\text{ความสมบูรณ์ของการเปลี่ยนแปลง} = \frac{NumberofCorrectModifications}{NumberofModification\,\mathrm{Re}\,quired}$$

โมเดลการวัดความสมบูรณ์ของการเปลี่ยนแปลงมีดังนี้

1. โมเดลการวัดความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH
   โดยโมเดลนี้มีระดับความเชื่อมั่นเป็น 0.999 (p-value มีค่าเท่ากับ 0.001) และมีค่าสัมประสิทธิ์การตัดสินใจ เท่ากับ 0.401
2. โมเดลการวัดความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA
   โมเดลนี้มีระดับความเชื่อมั่นเป็น 1.00 (p-value มีค่าเท่ากับ 0.0) และมีค่าสัมประสิทธิ์การตัดสินใจเท่ากับ 0.485

โดยที่ **NAggH** (Number on Aggregation Hierarchies) คือ จำนวนโครงสร้างลำดับชั้นของความสัมพันธ์แบบแอกกริเกชัน ทั้งหมดในแผนภาพคลาส

NGenH (Number of Generalization Hierarchies) คือ จำนวนโครงสร้างลำดับชั้นของความสัมพันธ์แบบเจนเนอรัลไลเซชันทั้งหมดในแผนภาพคลาส

NA (Number of Attributes) คือจำนวนคุณลักษณะทั้งหมดในแผนภาพคลาส

2.1.7 **วิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่** (Combinatorial Optimization) [10]

วิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่เป็นวิธีการหารูปแบบของโซลูชัน (Solution) ที่ดีที่สุดของปัญหาหนึ่งๆ โดย วิธีการนี้จะมีส่วนประกอบที่สำคัญอยู่ 2 อย่างด้วยกันคือ

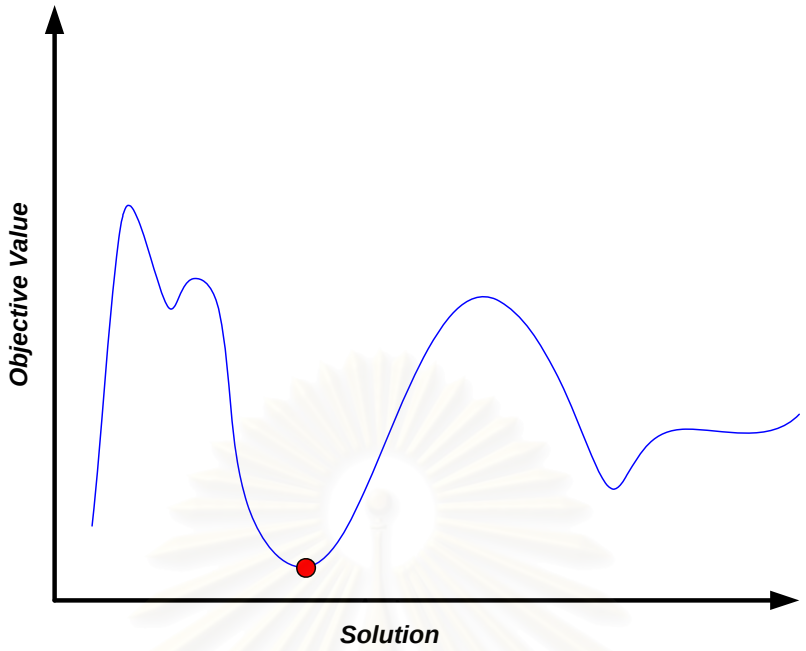
- ปัญหาที่ต้องการหาโซลูชันที่ดีที่สุด ซึ่งถูกเรียกว่า ฟังก์ชันเป้าหมาย (Objective Function)
- โซลูชันของปัญหา ซึ่งถูกเรียกว่ากลุ่มพารามิเตอร์ (Parameters) ที่เป็นอินพุท (Input) ของฟังก์ชันเป้าหมาย

โดยกลุ่มพารามิเตอร์ที่ทำให้ฟังก์ชันเป้าหมายมีค่าดีที่สุดจะถือเป็นโซลูชันที่ดีที่สุดของปัญหาหรือฟังก์ชันเป้าหมายนั้นๆ

ลักษณะสำคัญบางประการที่เกี่ยวข้องกับวิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่นี้คือ การค้นพบโซลูชันที่คิดว่าดีที่สุดสำหรับปัญหาที่กำลังสนใจ แต่ในความเป็นจริงแล้วโซลูชันนั้นเป็นโซลูชันดีกว่าโซลูชันข้างเคียงที่อยู่ติดหรือใกล้กันในบริเวณใดบริเวณหนึ่งของพื้นผิวของฟังก์ชันเป้าหมาย (ปัญหา) เท่านั้น ลักษณะแบบนี้จะถูกเรียกว่าการติดอยู่ใน ภาวะที่ดีที่สุดเฉพาะแห่ง (Local Optima) ส่วนถ้าโซลูชันที่ได้มาเป็นโซลูชันที่ดีที่สุดเมื่อเทียบกับทุกๆ โซลูชันที่มีในพื้นผิวของฟังก์ชันเป้าหมายอย่างแท้จริงแล้วโซลูชันนั้นจะสามารถเรียกได้ว่าเป็นภาวะที่ดีที่สุดของทั้งหมด (Global Optima)

จากรูปที่ 2.3 จุดแสดงถึงภาวะที่ดีที่สุดเฉพาะแห่งที่เป็นภาวะที่ดีที่สุดของทั้งหมด เมื่อพิจารณาจากพื้นผิว ซึ่งในที่นี้เป็นกราฟในสองมิติของฟังก์ชันการประเมินใดๆ โดยแกนตั้งแสดงค่าที่ได้จากฟังก์ชันเป้าหมายของโซลูชันใดๆ แกนนอนแสดงถึงค่าโซลูชันนั้นๆ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.3 ตัวอย่างกราฟ (Graph) ของฟังก์ชันการประเมินค่าใดๆ

2.1.7.1 อัลกอริทึมซีมูเลทเต็ดแอนนิลลิง (Simulated Annealing Algorithm: SA) [11]

อัลกอริทึมซีมูเลทเต็ดแอนนิลลิงเป็นวิธีที่ใช้การสุ่มในการแก้ปัญหาค่าเหมาะสมเชิงวิธีจัดหมู่โดยเลียนแบบวิธีมาจากกระบวนการทางฟิสิกส์ ในการเพิ่มความร้อนให้กับวัตถุและทำให้เย็นลงอย่างช้าๆเพื่อทำให้รูปแบบผลึกในวัตถุนั้นเรียงตัวกันอย่างมีระเบียบเป็นการเพิ่มความแข็งแรงให้กับวัตถุนั้นๆ โดยรูปแบบการจัดหมู่กันของพารามิเตอร์ที่ได้จากการใช้อัลกอริทึมซีมูเลทเต็ดแอนนิลลิงอาจไม่ใช่รูปแบบที่ดีที่สุดหรือเรียกว่าภาวะที่ดีที่สุดของทั้งหมด (Global Optima) ก็ได้ เนื่องจากวิธีการนี้จะทำการสุ่มหารูปแบบการจัดหมู่กันของค่าพารามิเตอร์หรือเรียกว่าสถานะ (State) ต่างๆมาแล้วทำการทดลองใส่เป็นอินพุทของฟังก์ชันเป้าหมายหรือเรียกว่าฟังก์ชันการประเมินค่า (Cost Function) ค่าที่เป็นผลลัพธ์จากฟังก์ชันนี้จะถูกเก็บไว้เพื่อนำไปเปรียบเทียบกับค่าผลลัพธ์จากสถานะแบบอื่นที่ทำการสุ่มมาอีก เพื่อหาสถานะที่ทำให้ได้ผลลัพธ์จากฟังก์ชันการประเมินค่าที่ดีที่สุด (ต่ำที่สุด) ที่เป็นไปได้ตามข้อจำกัดทางด้านเวลาที่ถูกกำหนด อัลกอริทึมซีมูเลทเต็ดแอนนิลลิงสามารถอธิบายได้เป็นรหัสเทียม (Pseudo Code) ดังรูปที่ 2.4

จากรหัสเทียมจะเห็นได้ว่าจะต้องมีการกำหนดอุณหภูมิและสถานะเริ่มต้นอย่างสุ่มให้กับอัลกอริทึมซีมูเลทเต็ดแอนนิลลิงเสียก่อนจากนั้นในการทำซ้ำด้านนอก (Outer Iteration) จะเป็นการเปลี่ยนแปลงอุณหภูมิโดยจะทำการลดอุณหภูมิลงทีละนิด ส่วนการทำงานในการทำซ้ำด้านใน (Inner Iteration) จะเป็นการทำงานที่อุณหภูมิคงที่ค่าหนึ่ง โดยการทำงานในการทำซ้ำด้านในนี้จะเริ่มด้วยการสุ่มเลือกสถานะใหม่ จากนั้นทำการหาค่าประเมินสถานะของสถานะเดิมและสถานะ

ใหม่ โดยใช้ฟังก์ชันการประเมินเป็นตัวคำนวณ จากนั้นทำการหาผลต่างระหว่างค่าประเมินของสถานะเดิมกับค่าประเมินของสถานะใหม่ที่ได้มา ถ้าผลต่างนี้แสดงว่าสถานะใหม่เป็นอินพุทของฟังก์ชันการประเมินที่ดีว่าสถานะเดิมก็จะทำการยอมรับสถานะใหม่ทันทีโดยปราศจากเงื่อนไข แต่ถ้าสถานะใหม่ไม่ได้เป็นโซลูชันที่ดีกว่าสถานะเดิม วิธีการซีมูเลทเต็ดแอนนิลลิงก็ยังสามารถยอมรับสถานะใหม่นี้ได้ด้วยความน่าจะเป็น $e^{(-Delta_E\ /\ T)}$ ซึ่งคุณสมบัตินี้เองทำให้อัลกอริทึมซีมูเลทเต็ดแอนนิลลิงสามารถหลุดออกจากภาวะที่ดีที่สุดเฉพาะแห่ง (Local Optima) ได้

```
T = initial temperature
state = random initial state
repeat (until done)
{
      T = reduce temperature(T)
      repeat (until inner-loop criterion)
      {
        newstate = random neighbor(state)
        Delta_E = E(newstate) - E(state)
        If Delta_E < or = 0,
              state = newstate.
        Else
              state = newstate with probability e (-Delta_E / T)
       }
}
```

รูปที่ 2.4 รหัสเทียมของอัลกอริทึมซีมูเลทเต็ดแอนนิลลิง

เงื่อนไขในการจบการทำซ้ำทั้งสองระดับสามารถแปรเปลี่ยนได้ตามความเหมาะสมเฉพาะงาน เงื่อนไขการหยุดการทำซ้ำที่ดีที่สุดอย่างเห็นได้ชัดคือไม่สามารถที่จะหาสถานะที่ดีไปกว่าสถานะเดิมได้อีกแล้ว แต่ในการนำไปใช้จริงเงื่อนไขนี้อาจทำให้การอัลกอริทึมซีมูเลทเต็ดแอนนิลลิงทำงานเป็นเวลาที่นานมาก ดังนั้นสำหรับปัญหาที่ไม่ซับซ้อนและต้องการโซลูชันที่ดีพอใช้โดยไม่จำเป็นต้องดีที่สุด เราจึงเห็นการกำหนดให้มีการทำซ้ำทั้งสองระดับเป็นจำนวนรอบที่ตายตัว

2.1.7.2 อัลกอริทึมฟาสต์ซีมูเลทเต็ดแอนนิลลิง (Fast Simulated Annealing Algorithm: FSA) [12]

เป็นอีกรูปแบบหนึ่งของอัลกอริทึมซีมูเลทเต็ดแอนนิลลิงซึ่งมีอัตราการลดอุณหภูมิที่เป็นไปอย่างรวดเร็ว กล่าวคือ

$$T = \frac{T_0}{k}$$

โดย

- T คือ อุณหภูมิในขณะใดขณะหนึ่ง
- T_0 คือ อุณหภูมิเริ่มต้น
- k คือ จำนวนรอบของการลดอุณหภูมิ

โดยอัตราการลดอุณหภูมินี้จะต้องใช้ควบคู่ไปกับการสุ่มหาสถานะใหม่ที่มีความน่าจะเป็นที่จะสุ่มเจอเป็น

$$\frac{T}{(\Delta x^2 + T^2)^{(n+1)/2}}$$

โดย

- T คือ อุณหภูมิในขณะใดขณะหนึ่ง
- Δx คือ ผลต่างระหว่างสถานะใหม่กับสถานะเก่า
- n คือ จำนวนมิติ (Dimension) ของสถานะ

เนื่องมาจากความยุ่งยากของการหาฟังก์ชันในการสุ่มหาสถานะใหม่ให้สอดคล้องกับความน่าจะเป็นดังกล่าวการเปลี่ยนมาใช้การสุ่มหาสถานะใหม่แบบสุ่มทีละมิติของสถานะ (Dimension-wise) จึงเป็นที่นิยมมากกว่าโดยจะใช้อัตราการลดอุณหภูมิ เป็น

$$T = \frac{T_0}{k^{1/n}}$$

และ ความน่าจะเป็นของการสุ่มแต่ละมิติของสถานะใหม่เป็น

$$\frac{T}{\Delta x_i^2 + T^2}$$

โดย

- Δx_i^2 คือ ผลต่างระหว่างค่าในมิติที่ i ของสถานะใหม่กับสถานะเก่า

และฟังก์ชันในการสุ่มเลือกค่าใหม่ในแต่ละมิติของสถานะจะเป็น

$$x_{i0} + T\tan[\pi(U - 0.5)]$$

โดย

- x_{i0} คือ ค่าของสถานะเดิมในมิติที่ i
- U คือ ค่าตัวเลขทศนิยมสุ่มที่มีค่าเป็นได้ตั้งแต่ 0.0 ถึง 1.0

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัย "Bad-Smell Detection for Refactoring using Object-Oriented Software Metrics" [13]

งานวิจัยนี้นำเสนอวิธีการตรวจจับร่องรอยที่ไม่ดีสำหรับรหัสต้นทางโดยใช้มาตรวัดเชิงวัตถุ ผู้วิจัยนำเสนอวิธีการตรวจจับร่องรอยที่ไม่ดี 6 ประเภท ได้แก่ Feature Envy, Large Class, Lazy Class, Long Method, Long Parameter Lists และ Switch Statement โดยผู้วิจัยแบ่งวิธีการออกแบบเป็น 6 ขั้นตอนได้แก่การนิยามร่องรอยที่ไม่ดีเพื่ออธิบายลักษณะของร่อยรอยที่ไม่ดีแต่ละประเภท การกำหนดแรงจูงใจเพื่อบอกผลกระทบของร่องรอยที่ไม่ดีและการค้นพบร่องรอยที่ไม่ดีว่าทำได้อย่างไร การกำหนดวิธีการวัดเพื่อบอกถึงวิธีการหาร่องรอยที่ไม่ดี การกำหนดมาตรวัดรวมถึงข้อกำหนดของการพิจารณามาตรวัดเพื่อพิจารณาว่าแต่ละมาตรวัดนั้น ช่วงของมาตรวัดที่จะบอกว่าเป็นร่องรอยที่ไม่ดีนั้นมีค่าอย่างไรบ้าง และเสนอการประยุกต์ใช้วิธีรีแฟคทอริงสำหรับร่องรอยที่ไม่ดีแต่ละประเภท นอกจากนี้ผู้วิจัยได้พัฒนาเครื่องมือเพื่อใช้ในการตรวจจับร่องรอยที่ไม่ดีตามวิธีการที่เสนอ สำหรับตรวจจับร่องรอยที่ไม่ดีของโปรแกรมภาษาจาวา (Java) งานวิจัยนี้ประเมินความสามารถของมาตรวัดร่องรอยที่ไม่ดีและวิธีการรีแฟคทอริงด้วยการเปรียบเทียบค่ามาตรวัดร่องรอยที่ไม่ดีก่อนและหลังการประยุกต์ใช้วิธีรีแฟคทอริง ซึ่งผลของวิจัยนี้จะเห็นว่า มาตรวัดร่องรอยที่ไม่ดี มีค่าที่ดีขึ้น นั่นคือมาตรวัดร่องรอยที่ไม่ดี ที่เสนอมานั้น สามารถใช้ตรวจจับร่องรอยที่ไม่ดีทั้ง 6 ประเภทที่รหัสต้นทางได้

2.2.2 **งานวิจัย** “Detection Strategies: Metrics-Based Rules for Detecting Design Flaws” [14]

งานวิจัยนี้นำเสนอกลยุทธ์ในการตรวจจับข้อบกพร่องในรหัสต้นทางที่ก่อให้เกิดความเบี่ยงเบนไปจากกฎการออกแบบที่ดี กลยุทธ์จะสร้างจากการนำมาตรวัดซอฟต์แวร์มาประกอบเป็นลักษณะที่บอกถึงความเบี่ยงเบนไปจากกฎการออกแบบที่ดี โดยในงานวิจัยกล่าวว่า การใช้กลยุทธ์นี้จะสามารถบ่งบอกถึงข้อบกพร่องของรหัสต้นทางได้ดีกว่าใช้วิธีตรวจจับด้วยค่ามาตรวัดใด ๆ เพียงบางมาตรวัด และหากใช้กลยุทธ์นี้ในการตรวจจับข้อบกพร่องแล้ว จะสามารถแก้ไขข้อบกพร่องได้ตรงตามลักษณะของการเบี่ยงเบนไปจากกฎการออกแบบ ซึ่งต่างจากการใช้มาตรวัดใด ๆ เพียงบางมาตรวัด ที่จะต้องมีการตีความว่าผลของการวัดที่เกิดขึ้นนั้น จะต้องมีการแก้ไขข้อบกพร่องด้วยวิธีใด

งานวิจัยนี้ได้ให้คำนิยามของกลยุทธ์ในการตรวจจับว่า คือการอธิบายกฎในเชิงปริมาณเพื่อใช้ตรวจจับส่วนของรหัสต้นทางที่สอดคล้องกับกฎนั้น ซึ่งการอธิบายในเชิงปริมาณนี้จะอธิบายโดยใช้มาตรวัดเชิงวัตถุ

วิธีการกำหนดกลยุทธ์ในการตรวจจับด้วยมาตรวัดเชิงวัตถุจะมีส่วนประกอบ 2 ส่วนคือ กลไกในส่วนของการกรองข้อมูลและกลไกในส่วนของการประกอบ

กลไกในส่วนของการกรองข้อมูล(The Filtering Mechanism) ซึ่งมีได้ 2 แบบคือการกำหนดขอบของข้อมูล(Marginal Filters) และการกำหนดช่วงของการกรองข้อมูล(Interval Filter) กลไกทั้ง 2 แบบนี้ก็คือการกำหนดขอบหรือช่วงของค่ามาตรวัดนั่นเอง

> การกำหนดขอบของมาตรวัดสามารถใช้ได้ทั้งการกำหนดเป็นแบบเฉพาะเจาะจง (Absolute Semantical Filter) เช่น มาตรวัดมีค่ามากกว่า หรือน้อยกว่าค่าใด ๆ หรือการกรองข้อมูลแบบสัมพัทธ์ (Relative Semantical Filter) เช่นกรองข้อมูลที่มีค่าสูงที่สุดหรือต่ำที่สุดของมาตรวัด นอกจากนี้ยังสามารถใช้วิธีการทางสถิติเช่นใช้วิธีบอกซ์พลอท (Box-plots) ที่ใช้ในตรวจจับค่าที่เบี่ยงเบนอย่างผิดปกติจากค่าอื่นได้อีกด้วย
>
> ส่วนการกำหนดช่วงนั้น คือการกำหนดว่าค่ามาตรวัดอยู่ระหว่าง 2 ค่าใด ๆ จึงจะสอดคล้องกับกลยุทธ์ในการตรวจจับ

กลไกในส่วนของการประกอบ(The Composition Mechanism) หมายถึงโอเปอร์เรเตอร์ที่ใช้ในการประกอบมาตรวัดต่าง ๆ เข้าด้วยกัน ได้แก่ แอนด์(and), ออร์(or), และบัทนอท(butnot)

ขั้นตอนของการกำหนดกลยุทธ์ในการตรวจจับข้อบกพร่องประกอบด้วย

- การพิจารณากฎหรือลักษณะของข้อบกพร่องให้แตกออกเป็นลักษณะย่อย ๆ ที่สามารถอธิบายได้ด้วยมาตรวัดใด ๆ เพียงมาตรวัดหนึ่ง
- การกำหนดมาตรวัดสำหรับใช้วัดกฎย่อย ๆ
- การกำหนดวิธีการกรองข้อมูลมาตรวัด
- การกำหนดวิธีการประกอบมาตรวัดต่าง ๆ เข้าด้วยกันเป็นกลยุทธ์

ในงานวิจัยนี้ได้เสนอตัวอย่างกลยุทธ์ในการตรวจจับ God Class โดยเริ่มจากการพิจารณาลักษณะของ God Class ว่าเป็นคลาสที่มีความซับซ้อนสูง, มีการเข้าคู่กันของเมทธอดภายในคลาสต่ำ, และคลาสมีการเรียกใช้งานคุณสมบัติของคลาสอื่น พร้อมกับเสนอมาตรวัดที่ใช้ในกลยุทธ์การตรวจจับได้แก่มาตรวัด WMC (Weighted Method Count) สำหรับวัดความซับซ้อนของเมทธอดทั้งหมดภายในคลาส, มาตรวัด TCC (Tight Class Cohesion) สำหรับวัดการเข้าคู่กันของเมธทอด, และมาตรวัด ATFD(Access to Foreign Data) สำหรับวัดจำนวนการเรียกใช้งานคุณสมบัติของคลาสอื่น จากนั้นได้สร้างกลยุทธ์การตรวจจับ God Class ออกมาได้ดังต่อไปนี้

$$GodClass(S) = S' \left| \begin{array}{l} S' \subseteq S, \forall C \in S' \\ (WMC(C), TopValues(25\%)) \wedge (ATFD(C), \\ HigherThan(1)) \wedge (TCC(C), BottomValues(25\%)) \end{array} \right.$$

งานวิจัยนี้ยังได้พัฒนาเครื่องมือ PRODETECTION ที่ใช้ตรวจจับข้อบกพร่องที่รหัสต้นทางโดยใช้กลยุทธ์ในการตรวจจับที่นำเสนอในงานวิจัย ซึ่งผลของการตรวจจับนั้นสามารถบ่งบอกถึงส่วนของรหัสต้นทางที่มีข้อบกพร่องได้ และนอกจากนี้ยังสามารถนำกลยุทธ์การตรวจจับข้อบกพร่องนี้ไปใช้กับข้อบกพร่องในลักษณะอื่น ๆ ได้

ผู้จัดทำวิทยานิพนธ์ได้นำวิธีการกำหนดกลยุทธ์ในการตรวจจับข้อบกพร่องที่รหัสต้นทางนี้มาประยุกต์ใช้กับการตรวจจับข้อบกพร่องหรือร่องรอยที่ไม่ดีของโมเดลการออกแบบที่แผนภาพคลาสและแผนภาพซีเควนซ์

2.2.3 งานวิจัย “Towards the Optimization of Automatic Detection of Design Flaws in Object-Oriented Software System” [15]

งานวิจัยนี้นำเสนอวิธีการหาช่วงของมาตรวัดที่เหมาะสมที่จะใช้ในการตรวจจับข้อบกพร่องของซอฟต์แวร์เชิงวัตถุ วิธีการนี้เรียกว่าทูนนิงแมชชีน (Tuning Machine) ซึ่งจะสรุปช่วงของมาตรวัดโดยอ้างอิงจากการจำแนกส่วนของเอนทิตีของโมเดลการออกแบบว่าเอนทิตีใดมีลักษณะของการเกิดข้อบกพร่อง เอนทิตีใดไม่เกิดข้อบกพร่อง สำหรับข้อบกพร่องที่ต้องการ

พิจารณาใด ๆ เช่น God Class, Data Class ซึ่งเอนทิตีในที่นี้ได้แก่ คลาส หรือเมทธอด เป็นต้น การจำแนกการเกิดข้อบกพร่องของเอนทิตีจะทำโดยผู้เชี่ยวชาญ โดยจะมีการเก็บกลุ่มตัวอย่างของเอนทิตี และผู้เชี่ยวชาญจะจำแนกว่าในกลุ่มตัวอย่างมีเอนทิตีใดที่มีข้อบกพร่องบ้าง

หลังจากได้กลุ่มตัวอย่างและการจำแนกว่าเอนทิตีในกลุ่มตัวอย่างใดมีลักษณะของข้อบกพร่องแล้ว จะแบ่งกลุ่มตัวอย่างออกเป็น 2 ประเภทได้แก่

กลุ่มตัวอย่างโพสิทีฟ (Positive Example) หมายถึงกลุ่มตัวอย่างมีลักษณะของข้อบกพร่องที่ต้องการพิจารณาตามการจำแนกของผู้เชี่ยวชาญ

กลุ่มตัวอย่างเนกาทีฟ (Negative Example) หมายถึงกลุ่มตัวอย่างที่ไม่มีลักษณะของข้อบกพร่องที่ต้องการพิจารณาตามการจำแนกของผู้เชี่ยวชาญ

หลังจากนั้นงานวิจัยนี้จะนำวิธีการกำหนดกลยุทธ์ในการตรวจจับข้อบกพร่องของ [14] มาใช้ในการตรวจจับข้อบกพร่องของเอนทิตี้ของกลุ่มตัวอย่าง จากประเภทของกลุ่มตัวอย่างทั้งสองนี้ หากเอนทิตี้ของตัวอย่างใดที่มีผู้เชี่ยวชาญจำแนกให้เป็นประเภทกลุ่มตัวอย่างโพสิทีฟแต่จากกลยุทธ์การตรวจจับระบุว่าเอนทิตี้นั้นไม่มีข้อบกพร่อง จะเรียกเอนทิตี้นั้นว่าเป็นกลุ่มตัวอย่างฟอลส์เนกาทีฟ (False Negative Example) และหากเอนทิตี้ของตัวอย่างใดที่มีผู้เชี่ยวชาญจำแนกให้เป็นประเภทกลุ่มตัวอย่างเนกาทีฟแต่จากกลยุทธ์การตรวจจับระบุว่าเอนทิตี้นั้นมีข้อบกพร่อง จะเรียกเอนทิตี้นั้นว่าเป็นกลุ่มตัวอย่างฟอลส์โพสิทีฟ (False Positive Example)

งานวิจัยนี้ระบุว่า ช่วงของมาตรวัดที่เหมาะสมที่สุดในกลยุทธ์การตรวจจับข้อบกพร่องได้แก่ค่าช่วงของมาตรวัดที่จะทำให้จำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟมีจำนวนน้อยที่สุด งานวิจัยนี้จึงได้นำเจเนติกอัลกอริธึม (Genetic Algorithm) มาใช้ในการคำนวณหาค่าช่วงที่เหมาะสมที่สุดที่จะทำให้จำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟมีค่าน้อยที่สุด โดยเจเนติกอัลกอริธึมจะทำการหาค่าพารามิเตอร์ที่จะทำให้ฟังก์ชันการประเมินค่าต่อไปนี้มีค่าน้อยที่สุด

$$f(X) = A * b * Fn_No(X) + A * (1-b) * Fp_No(X) + 1$$

โดยที่

- X คือ โซลูชัน(Solution)ที่เป็นไปได้
- Fn_No(X) คือจำนวนฟอลส์เนกาทีฟที่ได้จากกลยุทธ์การตรวจจับข้อบกพร่อง
- Fp_No(X) คือจำนวนฟอลส์โพสิทีฟที่ได้จากกลยุทธ์การตรวจจับข้อบกพร่อง
- A คือ เพนัลตีแอมพลิจูด(Penalty Amplitude) เป็นค่าคงที่ที่เป็นจำนวนเต็มที่เราสามารถกำหนดได้เพื่อใช้ถ่วงน้ำหนักหากมี X ที่ไม่เหมาะสม

- b หมายถึง ค่าความสมดุลย์ (balance) ซึ่งมีค่าคงที่ตั้งแต่ 0 ถึง 1 เป็นค่าที่ใช้ปรับระดับการให้น้ำหนักระหว่างฟอลส์เนกาทีฟและฟอลส์โพสิทีฟ ว่าต้องการให้น้ำหนักกับค่าใดมากกว่ากัน เช่นหากกำหนด b มีค่าเป็น 0.5 หมายความว่า ต้องการให้น้ำหนักกับทั้งสองค่าเท่ากัน

งานวิจัยนี้จะได้มีการทดลองใช้วิธีการโดยเลือกข้อบกพร่องประเภท God Class และ Data Class มาหาค่าช่วงของมาตรวัด โดยกลยุทธ์การตรวจจับข้อบกพร่องจะประยุกต์กับรหัสต้นทางที่พัฒนาด้วยภาษาจาวา (Java) และ ซีพลัสพลัส (C++)

ในการทดลองจะแบ่งกลุ่มตัวอย่างไว้สำหรับคำนวณหาช่วงของมาตรวัดออกเป็น 2 กลุ่ม คือกลุ่มที่เรียกว่าเซ็ตทูนนิง (Tuning Set) ที่ใช้สำหรับหาช่วงของมาตรวัด และกลุ่มที่เรียกว่าเซ็ตแวลิเดชัน (Validation Set) ที่เป็นกลุ่มซึ่งใช้สำหรับทดสอบผลของช่วงของมาตรวัดว่าหลังจากได้ค่าช่วงของมาตรวัดแล้ว เมื่อนำไปทดสอบกับเซ็ตแวลิเดชันทำให้จำนวนของฟอลส์เนกาทีฟและฟอลส์โพสิทีฟเป็นอย่างไร ในการเริ่มต้นการทดลองหรือที่เรียกว่าการทูน (Tune) หาค่าช่วงของมาตรวัดจะต้องมีการกำหนดช่วงของมาตรวัดเริ่มต้นก่อน หลังจากนั้นจะใช้เจเนติกอัลกอริธึม หาค่าช่วงของมาตรวัดที่เหมาะสมต่อไป

ผลจากการทดลองในงานวิจัยสรุปได้ว่า ค่าช่วงของมาตรวัดที่ได้จากทูนนิงแมชชีนสามารถใช้ในการตรวจจับข้อบกพร่องทั้งในเซ็ตทูนนิงและเซ็ตแวลิเดชันแล้วให้ค่าฟอลส์เนกาทีฟและฟอลส์โพสิทีฟน้อยลงจากการใช้ค่าช่วงของมาตรวัดเริ่มต้น

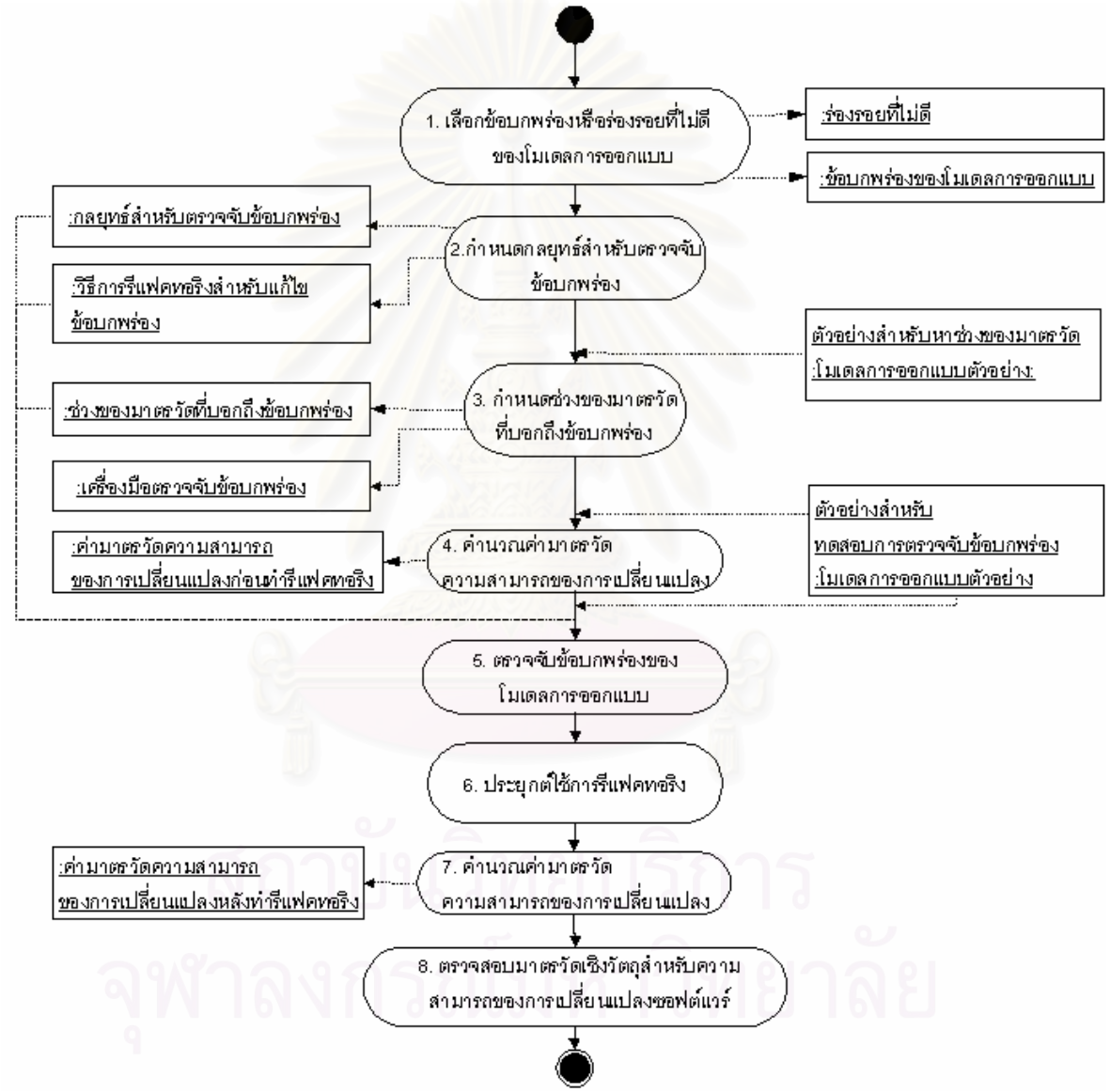
ในวิทยานิพนธ์ฉบับนี้จะนำวิธีการของทูนนิงแมชชีนมาประยุกต์ใช้ โดยจะประยุกต์ใช้กับกลยุทธ์การตรวจจับข้อบกพร่องที่ทำกับโมเดลการออกแบบ และจะใช้ซีมูเลทเต็ดแอนนิลลิงอัลกอริธึม (Simulated Annealing Algorithm) ในการหาช่วงที่เหมาะสมที่สุด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

วิธีการตรวจจับข้อบกพร่องของโมเดลการออกแบบหรือร่องรอยที่ไม่ดี

วิธีการตรวจจับข้อบกพร่องของโมเดลการออกแบบในงานวิจัยนี้แบ่งออกเป็น 8 ขั้นตอนดังแสดงเป็นแผนภาพแอคทิวิตี้ในรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการตรวจจับข้อบกพร่องของโมเดลการออกแบบหรือร่องรอยที่ไม่ดี

3.1 ขั้นตอนการวิจัย

จากรูปที่ 3.1 วิธีการตรวจจับข้อบกพร่องมีทั้งหมด 8 ขั้นตอนประกอบด้วย

3.1.1 การเลือกข้อบกพร่องและร่องรอยที่ไม่ดีของโมเดลการออกแบบ ที่เกี่ยวข้องกับความสามารถในการเปลี่ยนแปลงซอฟต์แวร์

ขั้นตอนนี้จะเป็นการเลือกข้อบกพร่องของโมเดลการออกแบบที่มีการเสนอใน [7] และร่องรอยที่ไม่ดีของซอฟต์แวร์ที่เสนอใน [5] มาพิจารณา ซึ่งจะเรียกรวมกันว่า ข้อบกพร่องของโมเดลการออกแบบ (Design Defect) โดยเลือกเฉพาะข้อบกพร่องของโมเดลการออกแบบที่กระทบต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ด้วยการพิจารณาจากลักษณะของร่องรอยที่ไม่ดีและข้อบกพร่องของโมเดลการออกแบบว่าลักษณะดังกล่าวทำให้เกิดการเปลี่ยนแปลงในอนาคตทำได้ยากหรือไม่ โดยข้อบกพร่องที่เลือกมาจะต้องสามารถตรวจจับได้ด้วยมาตรวัดเชิงวัตถุสำหรับแผนภาพคลาสและแผนภาพซีเควนซ์

ข้อบกพร่องของโมเดลการออกแบบที่เลือกมาสำหรับใช้ในงานวิจัยนี้ประกอบด้วย Data Class, Feature Envy, God Class, Message Chains, Middle Man, Switch Statements

3.1.2 การกำหนดกลยุทธ์สำหรับตรวจจับข้อบกพร่องของโมเดลการออกแบบ

ขั้นตอนนี้จะบอกถึงวิธีการพิจารณาคุณลักษณะของโมเดลการออกแบบเพื่อหาข้อบกพร่องของโมเดลการออกแบบแต่ละประเภทที่เลือกในขั้นตอน 3.1.1 โดยจะพิจารณาจากลักษณะข้อบกพร่องของโมเดลการออกแบบ เช่นหากข้อบกพร่องนั้นเป็นข้อบกพร่องที่เกิดขึ้นจากการเรียกใช้งานคุณลักษณะระหว่างคลาส ก็จะกำหนดวิธีการวัดที่เป็นการวัดลักษณะของคลาสและความสัมพันธ์ระหว่างคลาส หรือหากข้อบกพร่องมีลักษณะที่เกี่ยวข้องกับเมทธอด ก็จะใช้วิธีการวัดที่สอดคล้องกับลักษณะของเมทธอดในโมเดลการออกแบบ

ในขั้นตอนนี้จะมีการกำหนดมาตรวัดเชิงวัตถุสำหรับโมเดลการออกแบบที่จะใช้ในกลยุทธ์สำหรับตรวจจับข้อบกพร่อง รวมถึงจะมีการเสนอวิธีการรีแฟคทอริงที่เหมาะสมเพื่อแก้ไขข้อบกพร่องของโมเดลการออกแบบในแต่ละกลยุทธ์ด้วย ซึ่งวิธีรีแฟคทอริงอาจมีได้มากกว่าหนึ่งวิธีสำหรับกลยุทธ์ใด ๆ

3.1.3 การกำหนดช่วงของมาตรวัด

หลังจากได้กลยุทธ์สำหรับตรวจจับข้อบกพร่องแล้ว จะเป็นขั้นตอนการกำหนดช่วงของมาตรวัดที่ใช้ในกลยุทธ์ว่า ค่าของมาตรวัดมีค่าอย่างไรถึงจะบอกถึงลักษณะของข้อบกพร่องของโมเดลการออกแบบ

ในขั้นตอนนี้ผู้วิจัยได้พัฒนาเครื่องมือสำหรับคำนวณช่วงของมาตรวัดที่บ่งบอกถึงข้อบกพร่องของโมเดลการออกแบบ โดยใช้โมเดลการออกแบบตัวอย่างสำหรับการหาช่วงของมาตรวัดข้อบกพร่อง (Sample Design Model: Tuning Set) ซึ่งจะมี 35 ตัวอย่างสำหรับข้อบกพร่องแต่ละประเภท เครื่องมือนี้จะสามารถนำไปใช้คำนวณค่ามาตรวัดความสามารถของการเปลี่ยนแปลงซอฟต์แวร์ และตรวจจับข้อบกพร่องของโมเดลการออกแบบในขั้นตอนที่ 3.1.4 และ 3.1.5 ได้อีกด้วย บทที่ 4 จะกล่าวถึงวิธีการคำนวณช่วงของมาตรวัดที่บ่งบอกถึงข้อบกพร่องของโมเดลการออกแบบ บทที่ 5 จะกล่าวถึงผลการคำนวณช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่องแต่ละประเภทโดยใช้เครื่องมือที่พัฒนาขึ้น และในบทที่ 6 จะกล่าวถึงรายละเอียดของการพัฒนาเครื่องมือ

3.1.4 การคำนวณค่ามาตรวัดความสามารถของการเปลี่ยนแปลง

ขั้นตอนนี้จะนำโมเดลการออกแบบตัวอย่างที่เป็นตัวอย่างสำหรับทดสอบการตรวจจับข้อบกพร่อง (Sample Design Model: Validation Set) ซึ่งจะมี 5 ตัวอย่างสำหรับข้อบกพร่องแต่ละประเภท มาคำนวณค่ามาตรวัดความสามารถในการเปลี่ยนแปลงของโมเดลการออกแบบก่อนการตรวจจับข้อบกพร่องของโมเดลการออกแบบและการประยุกต์ใช้วิธีรีแฟคทอริง เพื่อเก็บค่ามาตรวัดไว้ใช้ในการประเมินผลการรีแฟคทอริง รายละเอียดในการคำนวณค่ามาตรวัดความสามารถของการเปลี่ยนแปลงจะอยู่ในบทที่ 5

3.1.5 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ

ขั้นตอนนี้จะคำนวณค่ามาตรวัดสำหรับข้อบกพร่องของโมเดลการออกแบบแต่ละประเภทเพื่อพิจารณาว่าโมเดลการออกแบบตัวอย่าง มีค่ามาตรวัดอยู่ในช่วงที่บอกถึงข้อบกพร่องของโมเดลการออกแบบที่กำหนดในขั้นตอน 3.1.3 ประเภทใด โดยจะใช้โมเดลการออกแบบตัวอย่างในเซ็ตแวลิเดชันเดียวกับขั้นตอน 3.1.4 รายละเอียดการตรวจจับข้อบกพร่องของโมเดลการออกแบบจะแสดงในบทที่ 5

3.1.6 การประยุกต์ใช้การรีแฟคทอริง

ขั้นตอนนี้จะนำวิธีรีแฟคทอริงที่เสนอในขั้นตอน 3.1.2 มาใช้กำจัดข้อบกพร่องของโมเดลการออกแบบที่ตรวจพบในขั้นตอน 3.1.5 รายละเอียดการรีแฟคทอริงโมเดลการออกแบบตัวอย่างจะแสดงในบทที่ 5

3.1.7 การคำนวณค่ามาตรวัดความสามารถของการเปลี่ยนแปลง

ขั้นตอนนี้จะนำโมเดลการออกแบบตัวอย่างในเซ็ตแวลิเดชันมาคำนวณค่ามาตรวัดความสามารถในการเปลี่ยนแปลงของโมเดลการออกแบบหลังการตรวจจับข้อบกพร่องของโมเดลการออกแบบและการประยุกต์ใช้วิธีรีแฟคทอริง โดยใช้มาตรวัดความสามารถในการเปลี่ยนแปลงมาตรวัดเดียวกับที่ใช้ในขั้นตอน 3.1.4 เพื่อเก็บค่ามาตรวัดไว้ใช้ในการประเมินผลการรีแฟคทอริง รายละเอียดในการคำนวณค่ามาตรวัดความสามารถของการเปลี่ยนแปลงจะอยู่ในบทที่ 5

3.1.8 การตรวจสอบมาตรวัดเชิงวัตถุสำหรับการเปลี่ยนแปลงซอฟต์แวร์

ขั้นตอนนี้เป็นการประเมินข้อบกพร่องของโมเดลการออกแบบประเภทต่าง ๆ โดยเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงของโมเดลการออกแบบก่อนและหลังการประยุกต์ใช้วิธีรีแฟคทอริง (ค่ามาตรวัดที่ได้จากขั้นตอน 3.1.4 และ 3.1.7) ว่าค่ามาตรวัดแสดงถึงคุณภาพทางด้านความสามารถในการเปลี่ยนแปลงของโมเดลการออกแบบที่ดีขึ้นหรือไม่ รายละเอียดในการเปรียบเทียบค่ามาตรวัดความสามารถของการเปลี่ยนแปลงจะอยู่ในบทที่ 5

3.2 การออกแบบกลยุทธ์สำหรับตรวจจับข้อบกพร่องของโมเดลการออกแบบ

3.2.1 ขั้นตอนการออกแบบกลยุทธ์

งานวิจัยนี้ได้นำวิธีกำหนดกลยุทธ์จาก [14] มาประยุกต์ใช้ โดยในงานวิจัยนี้จะมีขั้นตอนการออกแบบกลยุทธ์เป็นดังต่อไปนี้

3.2.1.1 การกำหนดคำนิยาม คือ การอธิบายลักษณะของข้อบกพร่องแต่ละประเภท

3.2.1.2 การกำหนดวิธีการวัด โดยพิจารณาลักษณะของข้อบกพร่องให้แตกออกเป็นลักษณะย่อย ๆ

3.2.1.3 การกำหนดมาตรวัดสำหรับแต่ละลักษณะย่อยของข้อบกพร่อง

โดยงานวิจัยนี้ได้มีการกำหนดมาตรวัดโดยเลือกมาตรวัดเชิงวัตถุสำหรับโมเดลการออกแบบที่มีผู้เสนออยู่แล้วมาประยุกต์ใช้ เช่นมาตรวัด RFC, มาตรวัด COH, มาตรวัด NCPA และมีการเสนอมาตรวัดใหม่ได้แก่ NNorm, NSMsg, NMsg, SOMsgCh, และมาตรวัด WOD

3.2.1.4 การกำหนดช่วงของมาตรวัดที่บอกถึงข้อบกพร่อง รวมถึงการกำหนดขอบหรือช่วงของมาตรวัด และกำหนดวิธีการประกอบเป็นกลยุทธ์ในการตรวจจับข้อบกพร่อง

การกำหนดช่วงของมาตรวัดคือการหาค่าช่วงที่บอกว่า หากตรวจจับแล้วค่ามาตรวัดอยู่ในช่วงนี้ จะบอกถึงข้อบกพร่องของโมเดลการออกแบบ

งานวิจัยนี้ได้มีการคำนวณหาช่วงของค่ามาตรวัดสำหรับแต่ละกลยุทธ์ โดยประยุกต์ใช้วิธีการหาช่วงของมาตรวัดจากงานวิจัย [15] มาใช้สำหรับกลยุทธ์สำหรับการตรวจจับข้อบกพร่องประเภท Feature Envy, Message Chains, Middle Man, และ God Class และใช้ช่วงของมาตรวัดจากงานวิจัย [13] สำหรับกลยุทธ์ประเภท Switch Statements ส่วนมาตรวัดสำหรับกลยุทธ์ Data Class นั้นได้มีการกำหนดตามนิยมของข้อบกพร่องจาก [5] โดยใช้ค่าช่วงของมาตรวัดเป็น 0

3.2.1.5 การเสนอวิธีรีแฟคทอริงสำหรับข้อบกพร่อง

เป็นการเสนอแนะวิธีการปรับปรุงข้อบกพร่องของโมเดลการออกแบบ โดยผู้วิจัยใช้วิธีรีแฟคทอริงจาก [5] มาเสนอเพื่อปรับปรุงข้อบกพร่องของโมเดลการออกแบบแต่ละประเภท

3.2.2 กลยุทธ์การตรวจจับข้อบกพร่อง

กลยุทธ์ในการตรวจจับข้อบกพร่องในงานวิจัยนี้มีอยู่ 6 ประเภท ดังมีรายละเอียดดังต่อไปนี้

3.2.2.1 Data Class

1) คำนิยาม คือ ลักษณะที่คลาสมีเพียงหน้าที่ในการกำหนดค่าให้กับคุณลักษณะภายในคลาส ซึ่งในคลาสมีเพียงแอคเซสเซอร์เมทธอดเท่านั้น ไม่มีเมทธอดที่ทำหน้าอื่นอีก คลาสลักษณะนี้มีผลทำให้เสียค่าใช้จ่ายในการบำรุงรักษา

2) วิธีการวัด

- พิจารณาที่ระดับคลาส ว่ามีการเพียงแอคเซสเซอร์เมทธอดเท่านั้น ไม่มีเมทธอดที่ทำหน้าอย่างอื่น

3) การกำหนดมาตรวัด

- มาตรวัด Number of Normal Method (NNorm)

คือจำนวนเมทธอดธรรมดาที่ไม่ใช่แอคเซสเซอร์เมทธอด คอนสตรัคเตอร์เมทธอด หรือเดสตรัคเตอร์เมทธอดของคลาสที่พิจารณา

4) กลยุทธ์การตรวจจับ

$$DataClass(S) = S' \mid S' \subseteq S, \forall C \in S' \\ NNorm(C) == 0$$

กำหนดให้ *C* คือ คลาสที่สงสัยว่ามีข้อบกพร่อง

- ข้อกำหนดของค่ามาตรวัดที่บอกถึงข้อบกพร่องคือเมทธอดที่มีค่า NNorm = 0 นั่นคือคลาสที่พิจารณาไม่มีเมทธอดที่ทำหน้าอย่างอื่นเลยนอกจากแอคเซสเซอร์เมทธอด
- เนื่องจากนิยามของ Data Class จาก [5] ระบุไว้ชัดเจนแล้วว่า Data Class คือคลาสที่มีเพียงแอคเซสเซอร์เมทธอดเท่านั้น ไม่มีเมทธอดที่ทำหน้าที่อย่างอื่น ดังนั้นงานวิจัยนี้บทที่ 5 จึงไม่ได้มีการคำนวณหาช่วงของข้อกำหนดของมาตรวัดสำหรับตรวจจับที่ระบุถึงข้อบกพร่องประเภทนี้ โดยจะใช้ช่วงของมาตรวัด NNorm = 0 สำหรับตรวจจับข้อบกพร่อง

5) วิธีการรีแฟคทอริง

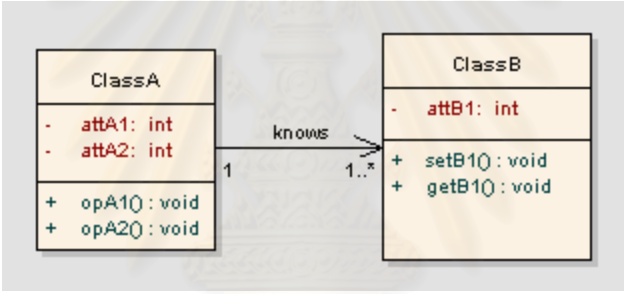
การประยุกต์วิธีรีแฟคทอริงสำหรับ Data Class มี 6 วิธีได้แก่การ Encapsulate Field, Encapsulate Collection, Remove Setting Method, Move Method, Extract Method, Hide Method

- วิธี Encapsulate Field ใช้เพื่อป้องกันการถูกกำหนดค่าให้คุณลักษณะโดยตรง ใช้เมื่อคลาสมีการประกาศคุณลักษณะเป็นแบบพับบลิค
- วิธี Encapsulate Collection ใช้เพื่อป้องกันการถูกกำหนดค่าให้คุณลักษณะโดยตรง ใช้เมื่อคลาสมีการประกาศคุณลักษณะเป็นแบบพับบลิค และคุณลักษณะมีลักษณะเป็นคอลเลคชัน
- วิธี Remove Setting Method ใช้เมื่อคุณลักษณะของคลาสไม่ควรถูกกำหนดค่า ซึ่งหากแก้ไขแล้ว คลาสอื่นจะไม่สามารถกำหนดคุณลักษณะของคลาสได้อีก
- วิธี Move Method ใช้เมื่อแอคเซสเซอร์ของคลาสถูกเรียกใช้โดยคลาสอื่น ซึ่งวิธี Move Method จะทำการย้ายเมทธอดจากคลาสที่มีการเรียกใช้แอคเซส

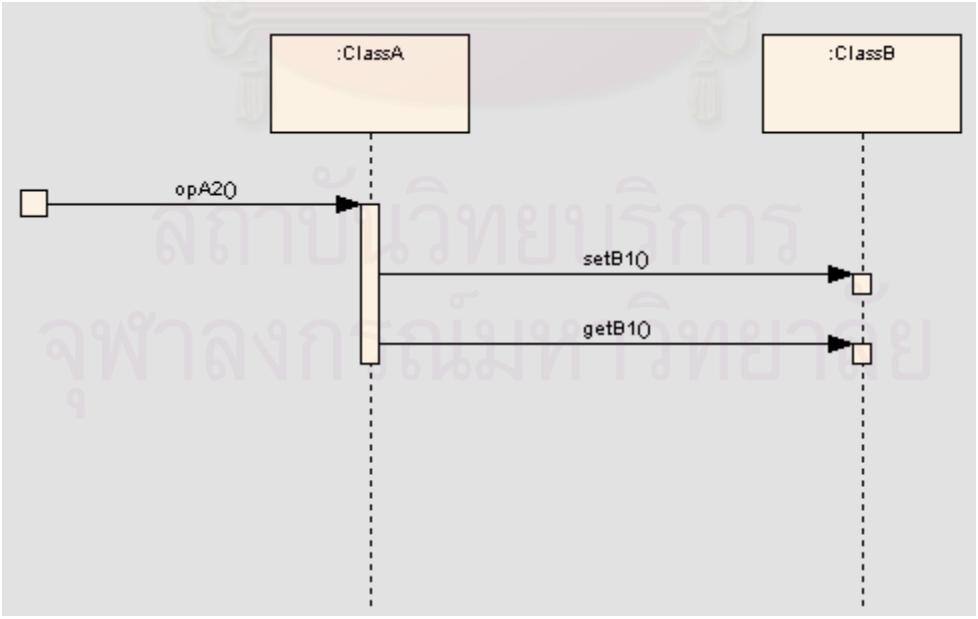
เซอร์ของ Data Class มาอยู่ที่ Data Class แทน ซึ่งจะทำให้คลาสมีเมทธอดที่ทำหน้าที่อย่างอื่นนอกจากแอคเซสเซอร์เมทธอด

- วิธี Extract Method ใช้เมื่อมีเมทธอดจากคลาสอื่นเรียกใช้แอคเซสเซอร์ของ Data Class แต่ไม่สามารถใช้วิธีรีแฟคทอริงแบบ Move Method ได้ จึงใช้วิธี Extract Method เพื่อกระจายส่วนของเมทธอดมาสร้างเป็นเมทธอดใหม่ แล้วย้ายเมทธอดเหล่านั้นมายัง Data Class ซึ่งจะทำให้คลาสมีเมทธอดที่ทำหน้าที่อย่างอื่นนอกจากแอคเซสเซอร์เมทธอด
- วิธี Hide Method ใช้เพื่อเปลี่ยนลักษณะของแอคเซสเซอร์เมทธอดให้เป็นไพรเวท เมื่อคุณลักษณะของคลาสไม่จำเป็นต้องถูกกำหนดค่าแล้ว

ตัวอย่างของลักษณะข้อบกพร่องในโมเดลการออกแบบมีดังต่อไปนี้ โดยรูปที่ 3.2 แผนภาพคลาสของระบบที่มี Data Class รูปที่ 3.3 แผนภาพซีเควนซ์ของระบบที่มี Data Class



รูปที่ 3.2 แผนภาพคลาสของระบบที่มีData Class



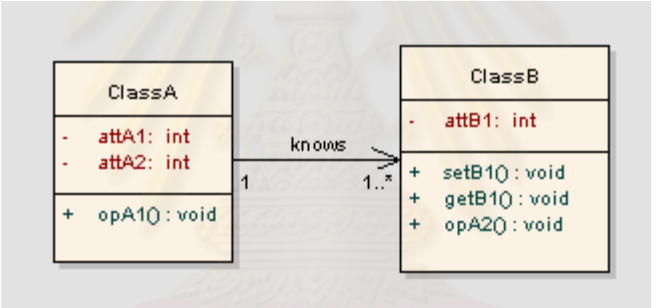
รูปที่ 3.3 แผนภาพซีเควนซ์ของระบบที่มี Data Class

จากรูปที่ 3.2, 3.3 นำมาคำนวณมาตรวัดได้ผลดังตารางที่ 3.1

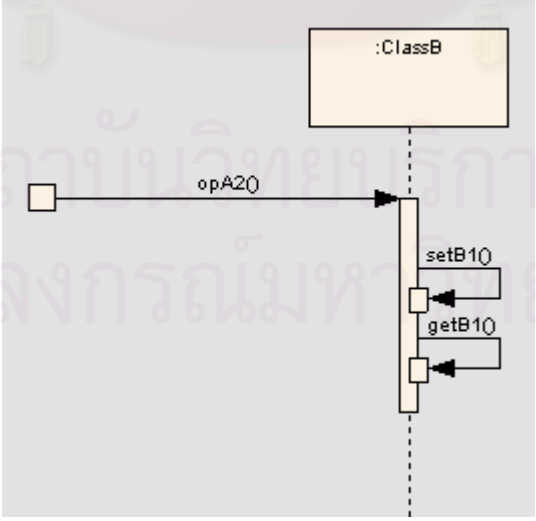
ตารางที่ 3.1 แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class

Class	ค่ามาตรวัด NNorm
A	2
B	0

จากผลที่ได้ในตารางจะเห็นว่า NNorm = 0 สำหรับคลาส Bดังนั้นควรใช้วิธีการรีแฟคทอริงด้วยการ Move Method เพื่อย้ายเมทธอด opA2 จากคลาส A ที่มีการเรียกใช้แอคเซสเซอร์เมทธอดของคลาส B ไปยังคลาส B ซึ่งได้ผลดังรูปที่ 3.4, 3.5 และได้ผลการคำนวณมาตรวัดของระบบหลังจากทำรีแฟคทอริงดังตารางที่ 3.2 ซึ่งจะเห็นว่า $NNorm_B = 1$ ซึ่งไม่อยู่ในช่วงของค่ามาตรวัดที่บอกถึงข้อบกพร่องลักษณะ Data Class



รูปที่ 3.4 แผนภาพคลาสของระบบหลังการรีแฟคทอริง



รูปที่ 3.5 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง

ตารางที่ 3.2 แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class

Class	ค่ามาตรวัด NNorm
A	1
B	1

3.2.2.2 Feature Envy

1) คำนิยาม คือ ลักษณะที่เมทธอดหรือคุณลักษณะของคลาสใด ๆ มีการเรียกใช้ หรือถูกเรียกใช้ โดยเมทธอดหรือคุณลักษณะอื่นของคลาสอื่นมากกว่าคลาสของตัวเอง

เนื่องจากงานวิจัยนี้พิจารณาถึงข้อบกพร่องที่เกิดในขั้นตอนการออกแบบ ซึ่งต้องตรวจจับข้อบกพร่องที่แผนภาพคลาสและแผนภาพซีเควนซ์ จึงมีข้อจำกัดทำให้สามารถพิจารณาลักษณะของ Feature Envy ได้เพียงลักษณะ Feature Envy ประเภทเมทธอดจากแผนภาพซีเควนซ์เท่านั้น เพราะการเขียนแผนภาพจะไม่นิยมเขียนการเรียกใช้คุณลักษณะของออบเจกต์ไลฟ์ไลน์ของคลาสตัวเองทำให้เกิดข้อจำกัดในการพิจารณาลักษณะ Feature Envy ประเภทคุณลักษณะที่โมเดลการออกแบบ

2) วิธีการวัด

- พิจารณาที่ระดับเมทธอดของคลาสใด ๆ ว่ามีการเรียกใช้เมทธอดหรือคุณลักษณะของคลาสอื่นมากกว่าการเรียกใช้เมทธอดของคลาสตัวเอง

3) การกำหนดมาตรวัด

- มาตรวัด Number of Self Message (NSMsg)

คือจำนวนเมทธอดของคลาสตัวเองที่ถูกเรียกโดยเมทธอดที่พิจารณา โดยจะวัดจากเมสเสจ (Message) ในแผนภาพซีเควนซ์ โดยถ้าเป็นเมสเสจของโอเปอร์เรชันเดียวกันจะนับเพียงหนึ่งครั้ง

- มาตรวัด Number of Message (NMsg)

NMsg คือจำนวนเมทธอดที่มากที่สุดของคลาสอื่นที่ถูกเรียกโดยเมทธอดที่พิจารณา โดยจะวัดจากเมสเสจ (Message) ในแผนภาพซีเควนซ์ โดยถ้าเป็นเมสเสจของโอเปอร์เรชันเดียวกันจะนับเพียงหนึ่งครั้ง

4) กลยุทธ์การตรวจจับ

$$FeatureEnvy(S) = S' \mid S' \subseteq S, \forall M \in S'$$
$$(NSMsg(M) / NMsg(M) < 0.31$$

กำหนดให้ *M* คือ เมทธอดที่สงสัยว่ามีข้อบกพร่อง

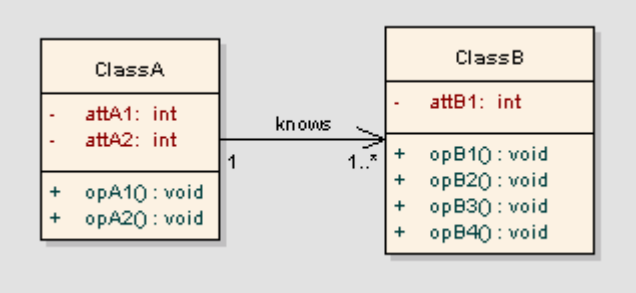
- ข้อกำหนดของค่ามาตรวัดที่บอกถึงข้อบกพร่องคือเมทธอดที่มีค่าวัดอยู่ในช่วงที่กำหนดในกลยุทธ์ นั่นคือเมทธอดมีอัตราส่วนการเรียกใช้เมทธอดของคลาสตัวเองต่อการเรียกใช้ เมทธอดของคลาสอื่นน้อย โดยหากอัตราส่วนมีค่าน้อยกว่า 0.31 แสดงว่าเมทธอดที่พิจารณานั้นมีข้อบกพร่องประเภท Feature Envy
- วิธีการหาช่วงของข้อบกพร่องแสดงรายละเอียดในบทที่ 4

5) วิธีการรีแฟคทอริง

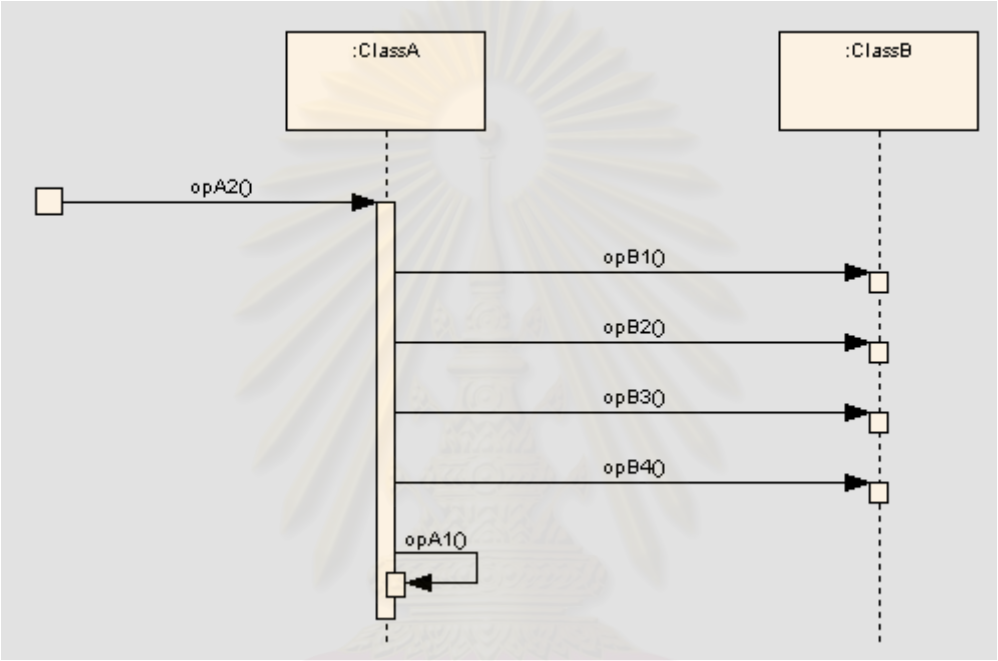
การประยุกต์วิธีรีแฟคทอริงสำหรับ Feature Envy มี 2 วิธีได้แก่การ Move Method, และ Extract Method

- วิธี Move Method ใช้เมื่อเมทธอดที่พิจารณามีอัตราการเรียกใช้เมทธอดของคลาสตัวเองต่อคลาสอื่นน้อย ดังนั้นควรย้ายเมทธอดที่พิจารณาไปยังคลาสอื่นที่เมทธอดที่พิจารณาไปเรียกใช้คุณสมบัติ ซึ่งจะทำให้อัตราการเรียกใช้เมทธอดของคลาสตัวเองต่อคลาสอื่นเพิ่มมากขึ้น
- วิธี Extract Method ใช้เมื่อเมทธอดที่พิจารณามีอัตราการเรียกใช้เมทธอดของคลาสตัวเองต่อคลาสอื่นน้อย โดยมีการเรียกใช้เมทธอดของคลาสอื่นมากกว่าหนึ่งคลาส ดังนั้นควรใช้วิธี Extract Method เพื่อกระจายส่วนของเมทธอดมาสร้างเป็นเมทธอดใหม่ แล้วย้ายเมทธอดเหล่านั้นไปยังคลาสต่าง ๆ ที่ต้องการเรียกใช้ ซึ่งจะทำให้อัตราการเรียกใช้เมทธอดของคลาสตัวเองต่อคลาสอื่นเพิ่มมากขึ้น

ตัวอย่างของลักษณะข้อบกพร่องในโมเดลการออกแบบมีดังต่อไปนี้ โดยรูปที่ 3.6 แผนภาพคลาสของระบบที่มีเมทธอดที่เป็น Feature Envy รูปที่ 3.7 แผนภาพซีเควนซ์ของระบบที่มีเมทธอดที่เป็น Feature Envy



รูปที่ 3.6 แผนภาพคลาสของระบบที่มีเมทธอดที่เป็น Feature Envy



รูปที่ 3.7 แผนภาพซีเควนซ์ของระบบที่มีเมทธอดที่เป็น Feature Envy

จากรูปที่ 3.6, 3.7 นำมาคำนวณมาตรวัดได้ผลดังตารางที่ 3.3

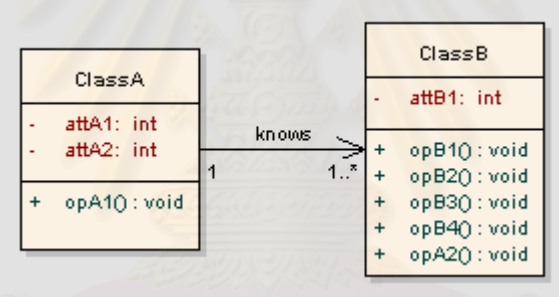
ตารางที่ 3.3 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Feature Envy

Class	Method	ค่ามาตรวัด NSMsg / NMsg
A	opA1	-
A	opA2	0.25
B	opB1	-

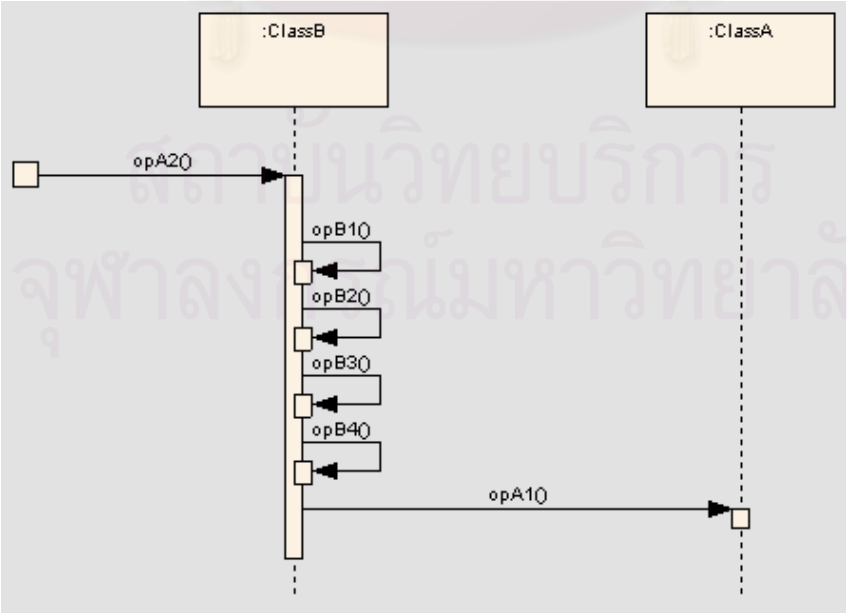
ตารางที่ 3.3(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Feature Envy

Class	Method	ค่ามาตรวัด NSMsg / NMsg
B	opB2	-
B	opB3	-
B	opB4	-

จากผลที่ได้ในตารางจะเห็นว่า $NSMsg_{opA2}$/ $NMsg_{opA2}$ = 0.25 ซึ่งมีค่าน้อยกว่า 0.31 ดังนั้นควรใช้วิธีการรีแฟคทอริงด้วยการ Move Method เพื่อย้ายเมทธอด opA2 จากคลาส A ไปยังคลาส B ซึ่งได้ผลดังรูปที่ 3.8, 3.9 และได้ผลการคำนวณมาตรวัดของระบบหลังจากทำรีแฟคทอริงดังตารางที่ 3.4 โดยค่า $NSMsg_{opA2}$/ $NMsg_{opA2}$ = 4 ซึ่งไม่อยู่ในช่วงของค่ามาตรวัดที่บอกถึงข้อบกพร่องลักษณะ Feature Envy



รูปที่ 3.8 แผนภาพคลาสของระบบหลังการรีแฟคทอริง



รูปที่ 3.9 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง

ตารางที่ 3.4 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Feature Envy

Class	Method	ค่ามาตรวัด NSMsg / NMsg
A	opA1	-
A	opA2	4
B	opB1	-
B	opB2	-
B	opB3	-
B	opB4	-

3.2.2.3 Message Chains

1) คำนิยาม คือ ลักษณะที่เมทธอดของคลาส มีการเรียกใช้อินแสตนท์ ของคลาสใดๆ ผ่านอินสแตนท์ของคลาสอื่น โดยมีการเรียกใช้อินสแตนท์ของคลาสอื่นต่อไปเรื่อย ๆ จะทำให้คลาสมีความขึ้นต่อกัน ในชุดของการเรียกใช้งานสูง

จากการเรียกใช้งานต่อเนื่องกันในรูปแบบนี้ หากมีความจำเป็นต้องเปลี่ยนแปลงคุณสมบัติของคลาสที่อยู่ระหว่างกลาง จะทำให้ต้องเปลี่ยนแปลงคลาสที่เป็นผู้เรียกใช้งานด้วย

2) วิธีการวัด

- พิจารณาที่ระดับเมทธอดของคลาสว่ามีการเรียกใช้อินแสตนท์ ของคลาสใดๆ ผ่านอินสแตนท์ของคลาสอื่น ในลักษณะที่ต่อเนื่องกันไปเรื่อย ๆ

3) การกำหนดมาตรวัด

- มาตรวัด Size of Message Chains (SOMsgCh)

  คือจำนวนช่วงความต่อเนื่องที่เมสเสจของเมทธอดใด ๆ เรียกใช้งานอินสแตนท์ของคลาสใด ๆ ผ่านอินสแตนท์ของคลาสอื่น โดยจะนับจำนวนช่วงเฉพาะเมสเสจที่เป็นจุดกำเนิดของ Chain เท่านั้น หากเมสเสจใดก่อให้เกิด Chain แต่เมสเสจนั้นไม่ใช่จุดกำเนิดของ Chain หรือเป็นเมสเสจที่อยู่ระหว่างกลางภายใน Chain จะถือว่าค่ามาตรวัดเป็น 0

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4) กลยุทธ์การตรวจจับ

$$MessageChains(S) = S' \mid S' \subseteq S, \forall M \in S'$$
$$SOMsgCh(M) \geq 2$$

กำหนดให้ *M* คือ เมทธอดที่สงสัยว่ามีข้อบกพร่อง

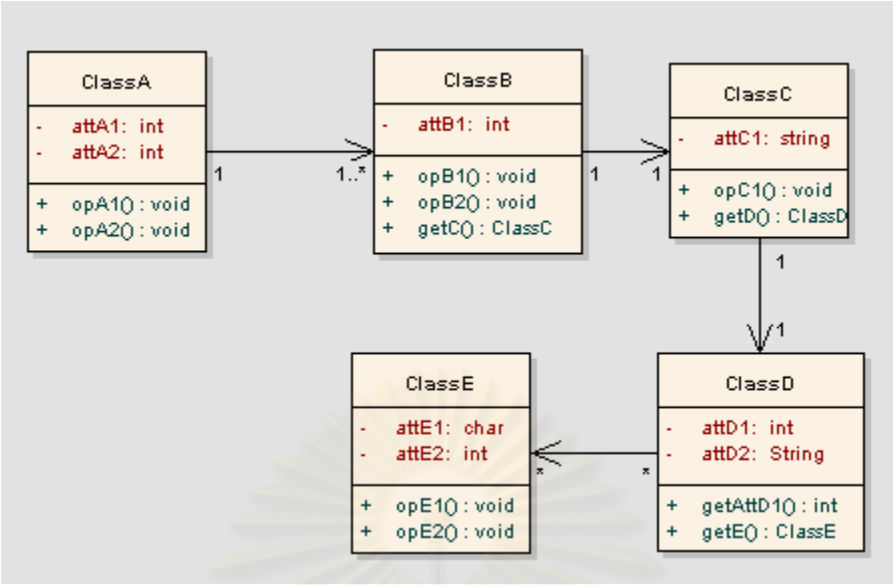
- ข้อกำหนดของค่ามาตรวัดที่บอกถึงข้อบกพร่องคือเมทธอดที่มีค่าวัดอยู่ในช่วงที่กำหนดในกลยุทธ์ นั่นคือเมสเสจของเมทธอดก่อให้เกิด Chain ของการเรียกใช้งานอินสแตนท์ โดยหากมีค่ามากกว่าหรือเท่ากับ 2 แสดงว่าเมทธอดที่พิจารณานั้นมีข้อบกพร่องประเภท Message Chains
- วิธีการหาช่วงของข้อบกพร่องแสดงรายละเอียดในบทที่ 4

5) วิธีการรีแฟคทอริง

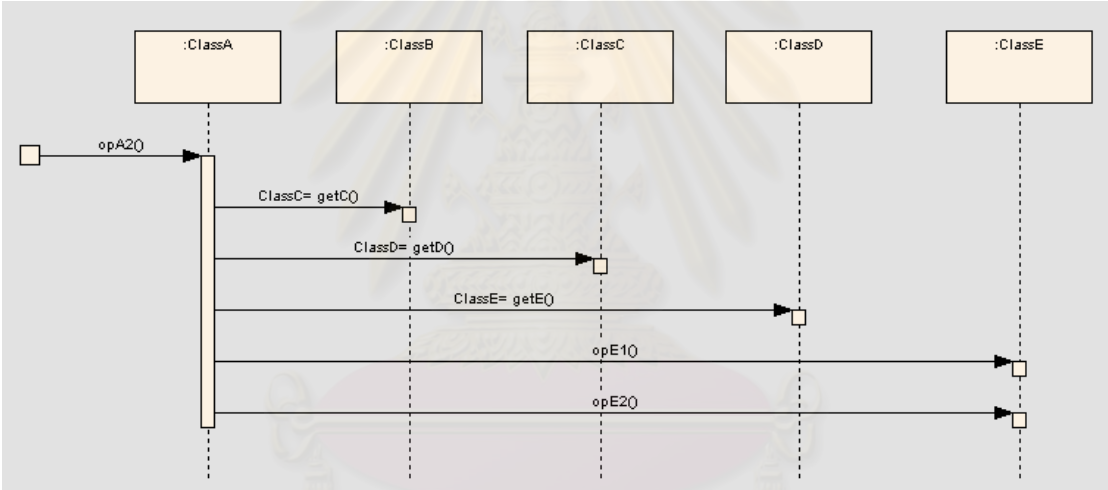
การประยุกต์วิธีรีแฟคทอริงสำหรับ Message Chains มี 3 วิธีได้แก่การ Hide Delegate, Extract Method, และ Move Method

- วิธี Hide Delegate สามารถใช้ได้ในช่วงใดก็ได้ใน Chain โดยจะเป็นการซ่อนอินสแตนท์ที่ถูกเรียกใช้งาน หรือเดลิเกท แล้วให้คลาสที่เคยทำหน้าที่เป็นตัวกลางส่งอินสแตนท์ที่ถูกเรียกให้แก่เมทธอดที่เรียกใช้งาน มาทำหน้าที่เป็นตัวกลางในการทำงานแทน หลังจากการทำรีแฟคทอริงแล้วจะทำให้จำนวนช่วงความต่อเนื่องของการเรียกใช้งานอินสแตนท์ลดลง
- วิธี Extract Method ใช้เพื่อกระจายส่วนของเมทธอดที่เรียกใช้งานอินสแตนท์ออกเป็นเมทธอดใหม่ แล้วใช้วิธี Move Method ย้ายเมทธอดที่กระจายออกมาให้มาอยู่ในส่วนล่างของ Chain เพื่อลดจำนวนช่วงของ Chain

ตัวอย่างของลักษณะข้อบกพร่องในโมเดลการออกแบบมีดังต่อไปนี้ โดยรูปที่ 3.10 แผนภาพคลาสของระบบที่มีเมทธอดที่เป็น Message Chains รูปที่ 3.11 แผนภาพซีเควนซ์ ของระบบที่มีเมทธอดที่เป็น Message Chains



รูปที่ 3.10 แผนภาพคลาสของระบบที่มีเมทธอดที่เป็น Message Chains



รูปที่ 3.11 แผนภาพซีเควนซ์ของระบบที่มีเมทธอดที่เป็น Message Chains

จากรูปที่ 3.10, 3.11 นำมาคำนวณมาตรวัดได้ผลดังตารางที่ 3.5

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.5 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Message Chains

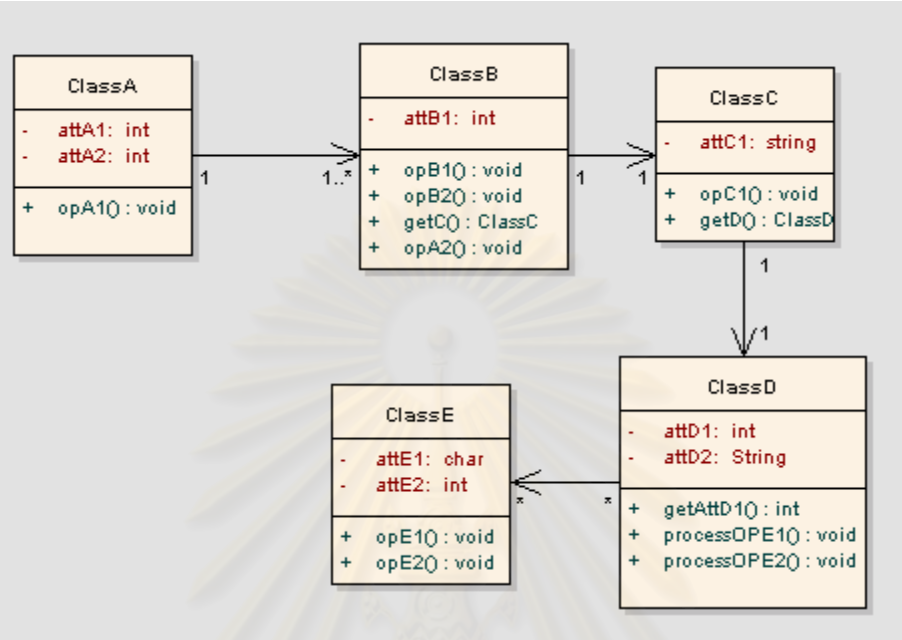
Class	Method	ค่ามาตรวัด SOMsgCh
A	opA1	0
A	opA2	0
B	opB1	0
B	opB2	0
B	getC	2
C	opC1	0
C	getD	0
D	getAttD1	0
D	getE	0
E	opE1	0
E	opE2	0

จากผลที่ได้ในตารางจะเห็นว่า $SOMsgCh_{getC}$ = 2 ซึ่งมีค่าเท่ากับ 2 ตามข้อกำหนดในกลยุทธ์ ดังนั้นควรใช้วิธีการรีแฟคทอริงด้วยการ Hide Delegate โดยเพิ่มเมทธอด processOPE1 และ processOPE2 ที่คลาส D เพื่อให้คลาส D ซ่อนอินสแตนท์ของคลาส E จากคลาส A และให้คลาส A เรียกใช้งานเมทธอด processOPE1 และเมทธอด processOPE2 ที่คลาส D แทน

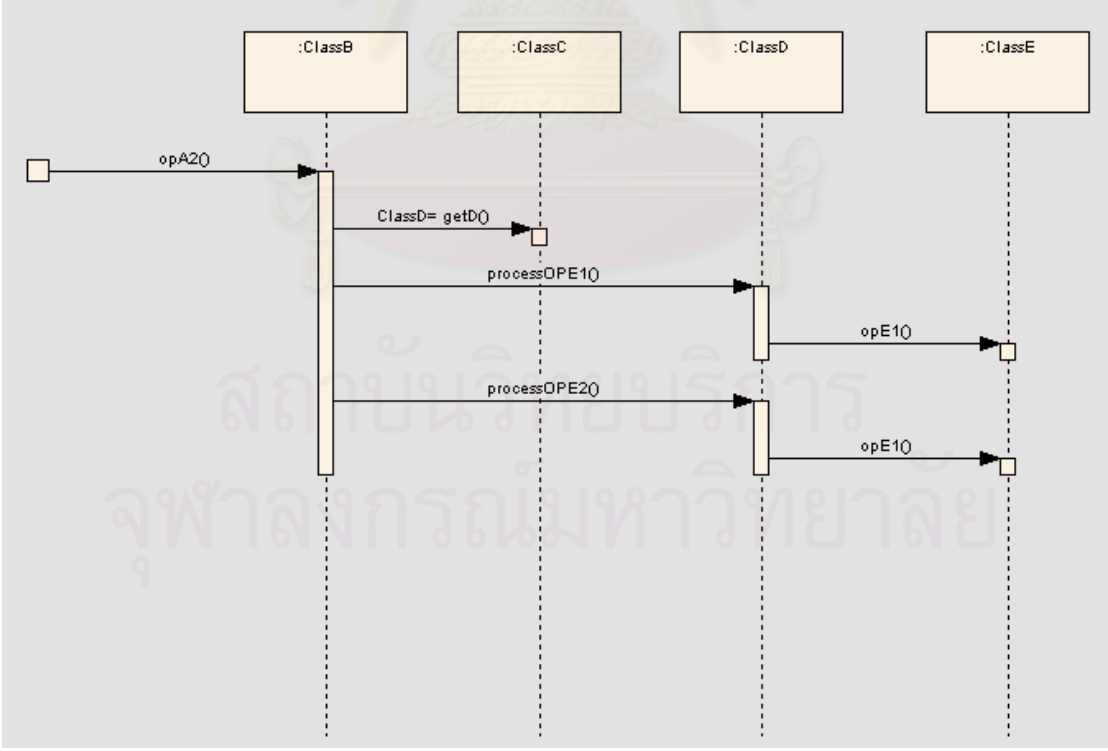
นอกจากนี้ยังใช้วิธี Extract Method เพื่อกระจายส่วนของเมทธอด opA2 ที่เรียกใช้งานอินสแตนท์ของคลาสอื่นใน Chain ออก แล้วจะใช้วิธี Move Method เพื่อย้ายเมทธอดใหม่นี้ไปไว้ที่คลาสใน Chain ส่วนล่าง แต่เนื่องจากพิจารณาแล้วจะเห็นว่าเมทธอด opA2 ไม่ถูกเรียกใช้จากภายในคลาสตัวเองหรือคลาสอื่นเลย จึงไม่ต้อง Extract Method opA2 ออกเป็นเมทธอดใหม่ แต่ใช้วิธีการ Move Method เพื่อย้าย Method opA2 ไปไว้ที่คลาส B ซึ่งจะทำให้จำนวนช่วงของ Chain สั้นลง

จากการประยุกต์ใช้วิธีรีแฟคทอริงได้ผลดังรูปที่ 3.12, 3.13 และค่ามาตรวัดของระบบหลังจากทำรีแฟคทอริงดังตารางที่ 3.6 ซึ่งจะเห็นว่า $SOMsgCh_{getC}$ กลายเป็น 0 ซึ่งไม่อยู่ในช่วง

ของค่ามาตรวัดที่บอกถึงข้อบกพร่องลักษณะ Message Chains นอกจากนี้การย้ายเมทธอด opA2 ไปยังคลาส B ไม่ทำให้เมทธอด getD ของคลาส C เกิด Message Chains เนื่องจากได้ค่า $SOMsgCh_{getD} = 0$ ตามผลการคำนวณมาตรวัดในตารางที่ 3.6



รูปที่ 3.12 แผนภาพคลาสของระบบหลังการรีแฟคทอริง



รูปที่ 3.13 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง

ตารางที่ 3.6 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Message Chains

Class	Method	ค่ามาตรวัด SOMsgCh
A	opA1	0
B	opA2	0
B	opB1	0
B	opB2	0
B	getC	0
C	opC1	0
C	getD	0
D	getAttD1	0
D	processOPE1	0
D	processOPE2	0
E	opE1	0
E	opE2	0

3.2.2.4 Middle Man

1) คำนิยาม คือ ลักษณะที่คลาสใด ๆ มีเมทธอดที่ทำหน้าที่มอบหมายงานให้กับคลาสอื่นเป็นจำนวนมากกว่าครึ่งหนึ่งของเมทธอดทั้งหมดของคลาส โดยคลาสมีเมทธอดที่ทำงานด้วยตัวเองเป็นจำนวนน้อย

จากลักษณะการมอบหมายงานจะเห็นว่า หากคลาสที่เป็นเดลิเกทที่ได้รับการมอบหมายงานต้องการเพิ่มเมทธอดใด ๆ จะทำให้คลาส Middle Man ต้องเพิ่มเมทธอดที่มีหน้าที่มอบหมายงานให้กับเมทธอดใหม่ที่เพิ่มขึ้นของเดลิเกทตามไปด้วย เพื่อรองรับการใช้งานจากคลาสอื่นในอนาคต

2) วิธีการวัด

- พิจารณาที่ระดับคลาสว่า คลาสมีอัตราส่วนระหว่างจำนวนเมทธอดที่ทำหน้าที่มอบหมายงานให้กับคลาสที่เป็นเดลิเกท ต่อจำนวนเมทธอดทั้งหมดของคลาสน้อย

3) การกำหนดมาตรวัด

- มาตรวัด Weight of Delegation (WOD)

คืออัตราส่วนระหว่างจำนวนเมทธอดของคลาสที่พิจารณาที่ทำหน้าที่มอบหมายงานให้กับคลาสที่เป็นอื่น ต่อจำนวนเมทธอดทั้งหมดของคลาสที่ไม่รวมคอนสตรัคเตอร์ ซึ่งการนับจำนวนเมทธอดของคลาสว่าเป็นเมทธอดที่ทำหน้าที่มอบหมายงานจะพิจารณาจากเส้นของเมสเสจในแผนภาพซีเควนซ์ว่าเมื่อเมสเสจของเมทธอดของออบเจกต์ไลฟ์ไลน์ถูกเรียกแล้ว ได้มีการเรียกใช้งานเมทธอดของคลาสอื่นต่อไปอีกหนึ่งเมสเสจเท่านั้นใช่หรือไม่ ถ้าใช่ แสดงว่าเมสเสจนั้นไม่มีหน้าที่อื่นอีกนอกจากการมอบหมายงานให้กับคลาสอื่น

4) กลยุทธ์การตรวจจับ

$$MiddleMan(S) = S' \mid S' \subseteq S, \forall C \in S'$$
$$WOD(C) > 0.43$$

กำหนดให้ *C* คือ คลาสที่สงสัยว่ามีข้อบกพร่อง

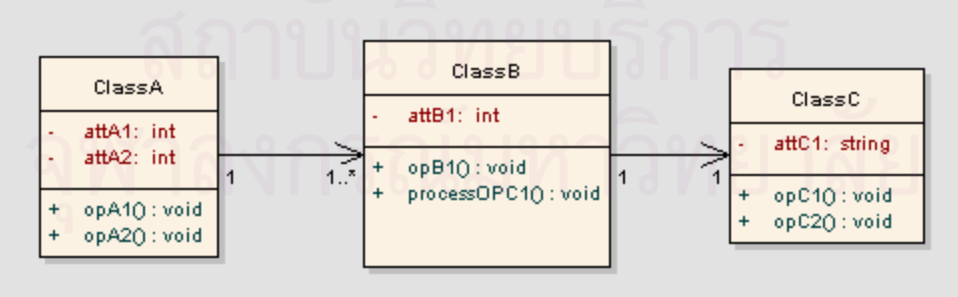
- ข้อกำหนดของค่ามาตรวัดที่บอกถึงข้อบกพร่องคือคลาสที่มีค่าวัดอยู่ในช่วงที่กำหนดในกลยุทธ์ นั่นคือหากอัตราส่วนระหว่างจำนวนเมทธอดของคลาสC ที่ทำหน้าที่มอบหมายงานให้กับคลาสที่เป็นอื่น ต่อจำนวนเมทธอดทั้งหมดของคลาส C ที่ไม่รวมคอนสตรัคเตอร์ มีค่ามากกว่า 0.43 แสดงว่าคลาสที่พิจารณานั้นมีข้อบกพร่องประเภท Middle Man
- วิธีการหาช่วงของข้อบกพร่องแสดงรายละเอียดในบทที่ 4

5) วิธีการรีแฟคทอริง

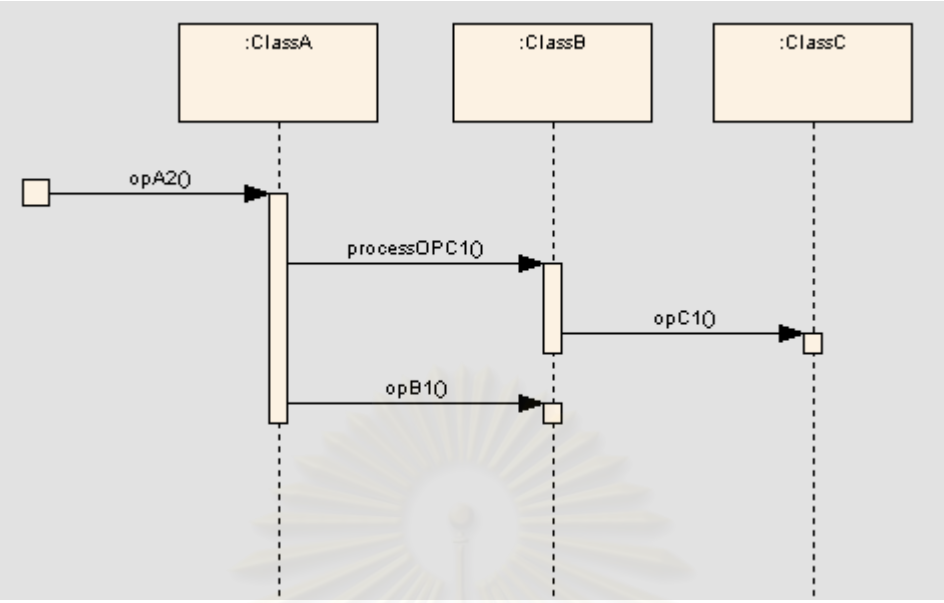
การประยุกต์วิธีรีแฟคทอริงสำหรับ Middle Man มี 3 วิธีได้แก่การ Remove Middle Man, Inline Method, และ Replace Delegation with Inheritance

- วิธี Remove Middle Man โดยการสร้างแอคเซสเซอร์ของคลาสที่เป็นเดลิเกทในคลาส Middle Man และเปลี่ยนให้คลาสผู้เรียกใช้งานเดลิเกทเมทธอด ไปเรียกใช้งานเดลิเกทเมทธอดโดยตรงไม่ต้องผ่าน Middle Man
- วิธี Inline Method ใช้เมื่อชื่อเมทธอดที่ทำหน้าที่มอบหมายงานของ Middle Man มีลักษณะเหมือนเนื้อหาภายในเมทธอด หมายความว่าเมทธอดนี้เป็นเพียงเมทธอดเล็ก ๆ ให้ทำการลบเมทธอดนี้ทิ้งแล้วย้ายเนื้อหาของเมทธอดไปไว้ยังคลาสผู้เรียก ซึ่งก็หมายถึงการลบเมทธอดที่ทำหน้าที่มอบหมายงานออก แล้วให้เมทธอดผู้เรียกไปเรียกใช้งานจากคลาสเดลิเกทโดยตรง ไม่ต้องผ่าน Middle Man
- วิธี Replace Delegation with Inheritance ใช้เมื่อคลาสที่เป็น Middle Man มีการมอบหมายงานให้เดลิเกทเมทธอด และคลาสที่เป็น Middle Man นี้มีการเรียกใช้งานเมทธอดอื่นของคลาสที่เป็นเดลิเกทด้วย จากลักษณะเช่นนี้สามารถแก้ไขโดยการเปลี่ยนให้คลาส Middle Man เป็นคลาสลูก (Subclass) ของคลาสที่เป็นเดลิเกทเลย ซึ่งจะทำให้ Middle Man ไม่ต้องมอบหมายงานให้กับเมทธอดของคลาสเดลิเกทอีก เนื่องจากได้รับการถ่ายทอดคุณสมบัติมาเรียบร้อยแล้ว

ตัวอย่างของลักษณะข้อบกพร่องประเภท Middle Man ในโมเดลการออกแบบมีดังต่อไปนี้ โดยรูปที่ 3.14 แผนภาพคลาสของระบบที่มีคลาส B ที่เป็น Middle Man รูปที่ 3.15 แผนภาพซีเควนซ์ ของระบบที่มีคลาส B ที่เป็น Middle Man



รูปที่ 3.14 แผนภาพคลาสของระบบที่มีคลาสที่เป็น Middle Man



รูปที่ 3.15 แผนภาพซีเควนซ์ของระบบที่มีคลาสที่เป็น Middle Man

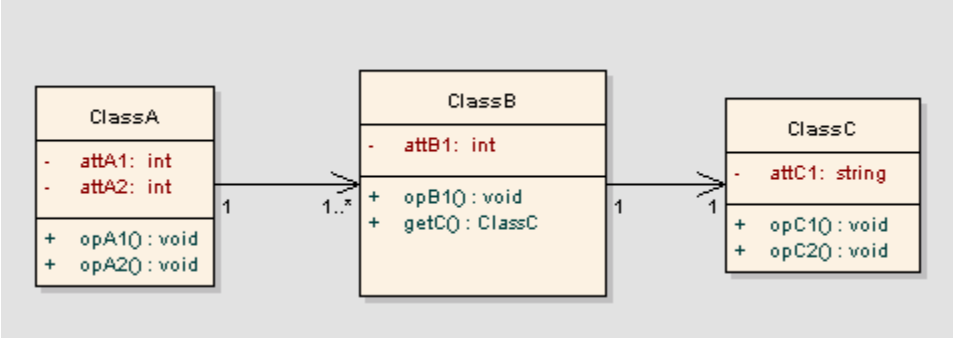
จากรูปที่ 3.14, 3.15 นำมาคำนวณมาตรวัดได้ผลดังตารางที่ 3.7

ตารางที่ 3.7 แสดงผลการคำนวณมาตรวัดของระบบที่มีคลาส B เป็น Middle Man

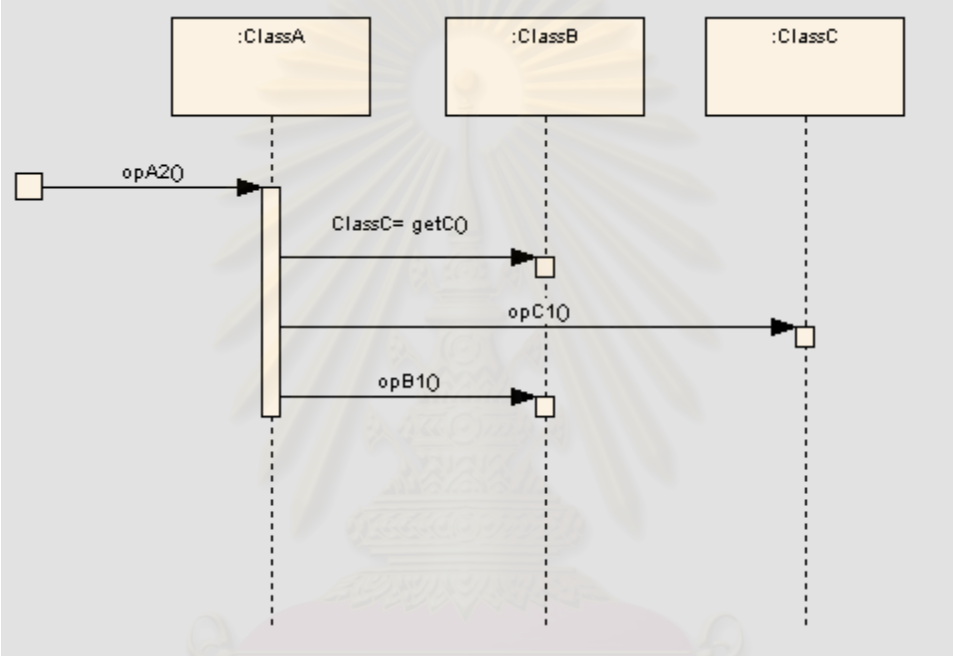
Class	ค่ามาตรวัด WOD
A	0
B	0.5
C	0

จากผลที่ได้ในตารางจะเห็นว่า WOD_B = 0.5 ซึ่งมีค่ามากกว่า 0.43 ตามข้อกำหนดในกลยุทธ์ ดังนั้นควรใช้วิธีการรีแฟคทอริงด้วยการ Remove Middle Man โดยเพิ่มเมทธอด getC ที่คลาส B เพื่อให้เมทธอด opA2 จากคลาส A เรียกใช้งานเมทธอด opC1 ได้โดยตรง และ processOPE2 ที่คลาส D เพื่อให้คลาส D ซ่อนอินสแตนท์ของคลาส E จากคลาส A และให้คลาส A เรียกใช้งานเมทธอด processOPE1 และเมทธอด processOPE2 ที่คลาส D แทน

จากการประยุกต์ใช้วิธีรีแฟคทอริงได้ผลดังรูปที่ 3.16, 3.17 และค่ามาตรวัดของระบบหลังจากทำรีแฟคทอริงดังตารางที่ 3.8 ซึ่งจะเห็นว่า WOD_B กลายเป็น 0 ซึ่งไม่อยู่ในช่วงของค่ามาตรวัดที่บอกถึงข้อบกพร่องลักษณะ Middle Man



รูปที่ 3.16 แผนภาพคลาสของระบบหลังการรีแฟคทอริง



รูปที่ 3.17 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง

ตารางที่ 3.8 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น Message Chains

Class	ค่ามาตรวัด WOD
A	0
B	0
C	0

3.2.2.5 God Class

1) คำนิยาม คือ ลักษณะที่คลาสใด ๆ ประกอบด้วยหน้าที่หลัก ๆ จำนวนมาก เมทธอดในคลาสเรียกใช้คุณลักษณะของคลาสอื่น เพื่อมาประมวลผลที่คลาสตัวเอง หมายถึงระบบไม่มีการ

กระจายหน้าที่ไปให้คลาสอื่น ๆ ในระบบ คลาสนี้อาจเป็นเพียงคลาสเดียวในระบบที่รวบรวมงานหลัก ๆ ที่เป็นของคลาสอื่น ๆ เอาไว้จำนวนมากอยู่ในคลาส คลาสที่เป็นศูนย์กลางในการทำหน้าที่หลัก ๆ ในกรณีนี้ คือ God Class

God Class เป็นคลาสที่มีความซับซ้อน (Complexity) สูง เมื่อมีความจำเป็นต้องเปลี่ยนแปลงโครงสร้างของคุณลักษณะของคลาสที่ถูกเรียก จะทำให้ต้องเปลี่ยนแปลงคลาส God Class ด้วย ซึ่งหากหน้าที่ของคลาสที่ถูกเรียกอยู่ที่คลาสที่ถูกเรียกเอง ไม่ใช่ที่คลาส God Class การเปลี่ยนแปลงจะทำได้สะดวกขึ้น

2) วิธีการวัด

- พิจารณาที่ระดับคลาสว่า เมทธอดของคลาสมีการเรียกใช้แอคเซสเซอร์ของคลาสอื่น
- พิจารณาที่คลาสว่าเป็นคลาสที่มีความซับซ้อนสูง
- พิจารณาที่คลาสว่าเป็นคลาสที่มีความเข้าคู่กันต่ำระหว่างเมทธอด

3) การกำหนดมาตรวัด

- มาตรวัด Number of Called Public Accessor (NCPA)

คือจำนวนแอคเซสเซอร์เมทธอดของคลาสอื่นที่ถูกเรียกโดยคลาสที่พิจารณา ซึ่งการนับจำนวนเมทธอดของคลาสอื่นจะพิจารณาจากเมสเสจที่ส่งออกจากออบเจกต์ไลฟ์ไลน์ของคลาสที่พิจารณาไปยังออบเจกต์ไลฟ์ไลน์ของคลาสอื่น

- มาตรวัด Response for a Class (RFC) [16] มีคำนิยามดังนี้

Response Set for a Class (RS) หมายถึงเซ็ต (Set) ของเมทธอดที่จะถูกเรียกใช้งานเมื่อเมสเสจของออบเจกต์ไลฟ์ไลน์ของคลาสที่พิจารณาถูกเรียกใช้งาน ซึ่ง

ถ้า $RS = \{M\} \cup i\{Ri\}$ โดยที่ $\{Ri\}$ คือเซ็ตของเมทธอดที่ถูกเรียกโดยเมทธอด i และ $\{M\}$ คือเซ็ตของเมทธอดในคลาสที่พิจารณา

แล้ว $RFC = | RS |$

มาตรวัด RFC นี้ใช้สำหรับวัดความซับซ้อนของคลาส

- มาตรวัด Cohesion measure of Class (COH) [17]

มาตรวัด COH นี้ใช้สำหรับวัดความเข้าคู่กันของเมทธอดภายในคลาส โดยพิจารณาจากความคล้ายคลึงกันระหว่างเมทธอดสองเมทธอดใด ๆ ภายในคลาส ค่าของ COH แสดงดังสมการต่อไปนี้

$$COH = \frac{SBC(C_i, C_i)}{|C_i| * |C_i|}$$

โดยที่ $|C_i|$ มีค่าเท่ากับจำนวนเมทธอดของคลาส C

มาตรวัด Similarity between Class (SBC) ใช้สำหรับวัดความเข้าคู่กันของเมทธอดภายในคลาส โดยพิจารณาจากความคล้ายคลึงกันระหว่างเมทธอดสองเมทธอดใด ๆ ภายในคลาส

ขั้นตอนการคำนวณความคล้ายคลึงกันระหว่างเมทธอด (Similarity between Methods) หรือ SBM จะต้องมีการสร้าง Communication Matrix between methods (CMM) ซึ่งเป็นเมตริกซ์สมมาตรที่มีขนาดในแต่ละมิติเท่ากับจำนวนเมทธอดของคลาสที่พิจารณา โดยที่แต่ละสมาชิกของCMM สามารถอธิบายได้ดังนี้ CMM (i, j) คือค่าความสัมพันธ์ของเมทธอดที่ i และ j ของคลาสที่พิจารณา โดยค่าการหาค่าความสัมพันธ์กันของเมทธอดแบ่งเป็นสองกรณีคือ ในกรณีที่เมทธอดทั้งสองมีการเรียกใช้กันค่าความสัมพันธ์กันจะมีค่าเป็นจำนวนพารามิเตอร์ของเมทธอดที่ถูกเรียกใช้ บวกหนึ่ง ในกรณีที่เมทธอดทั้งสองไม่มีการเรียกใช้กันค่าความสัมพันธ์กันจะเท่ากับ 0 จากนั้นนำค่าที่ได้มาบวกเพิ่มไปด้วยจำนวนคุณลักษณะของคลาสที่เมทธอดทั้งสองใช้ร่วมกัน

หลังจากได้ CMM แล้วจะสามารถนำมาใช้ในการคำนวณ SBM และ SBC ได้ดังต่อไปนี้

$$SBM(m1, m2) = \frac{\sum_{k=1}^{M} CMM(i,k) * CMM(j,k)}{\sqrt{\sum_{k=1}^{M} CMM(i,k)^2 * \sum_{k=1}^{M} CMM(j,k)^2}}$$

โดยที่ M คือจำนวนเมทธอดในคลาส และ k คือเลขจำนวนเต็มใด ๆ

ถ้า $i = id(m1), j = id(m2)$ โดยที่ id หมายถึงลำดับที่ของเมทธอด

ค่าของ SBC แสดงดังสมการต่อไปนี้

$$SBC(Ci, Cj) = \sum_{p=1}^{|Ci|} \sum_{q=1}^{|Cj|} SBM(m_{ip}, m_{jq})$$

โดยที่ p และ q คือเลขจำนวนเต็มใด ๆ

สำหรับการวัด SBC ในงานวิจัยนี้ Ci และ Cj จะหมายถึงคลาสที่พิจารณาคลาสเดียวกัน

4) กลยุทธ์การตรวจจับ

$$GodClass(S) = S' \mid S' \subseteq S, \forall C \in S'$$
$$(NCPA(C) > 1) \wedge (RFC(C) > 9) \wedge (COH(C) < 0.25)$$

กำหนดให้ *C* คือ คลาสที่สงสัยว่ามีข้อบกพร่อง

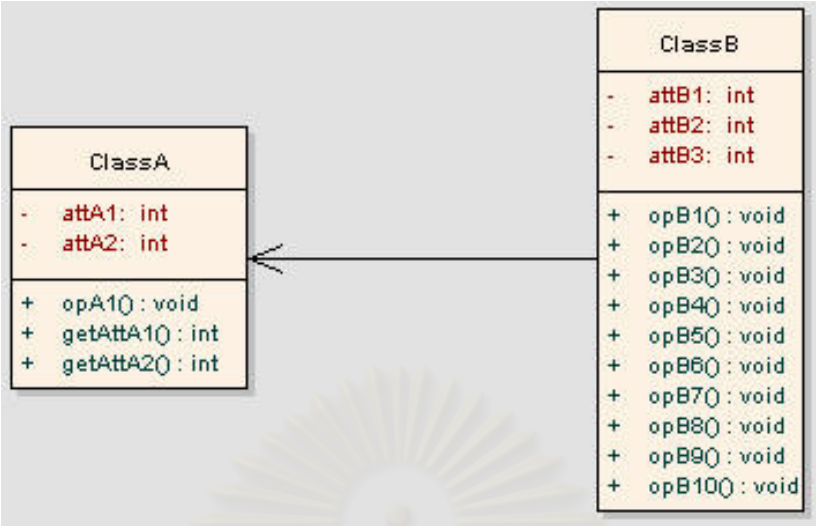
- ข้อกำหนดของค่ามาตรวัดที่บอกถึงข้อบกพร่องคือคลาสที่มีค่าวัดอยู่ในช่วงที่กำหนดในกลยุทธ์ นั่นคือหากคลาสที่พิจารณามีการเรียกใช้งานแอคเซสเซอร์ของคลาสอื่นมากกว่า 1 และคลาสที่พิจารณามีความซับซ้อนมากกว่า 9 และมีความเข้าคู่กันภายในคลาสน้อยกว่า 0.25 แล้ว แสดงว่าคลาสที่พิจารณานั้นมีข้อบกพร่องประเภท God Class
- วิธีการหาช่วงของข้อบกพร่องแสดงรายละเอียดในบทที่ 4

5) วิธีการรีแฟคทอริง

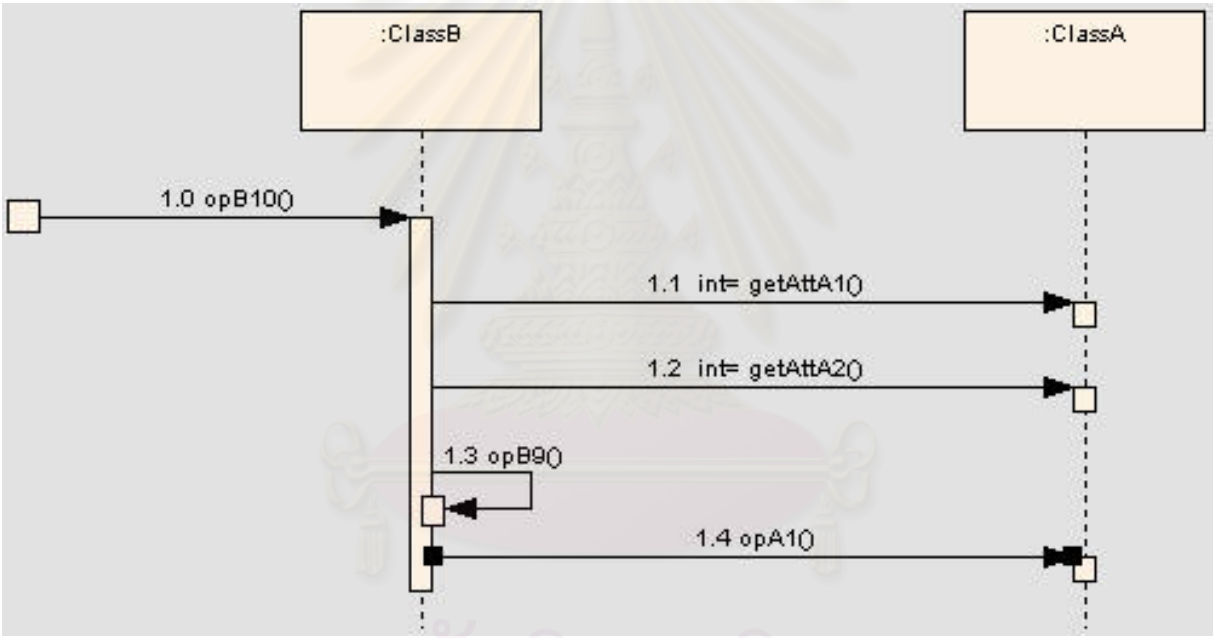
การประยุกต์วิธีรีแฟคทอริงสำหรับ God Class มี 3 วิธีได้แก่การ Move Method, Extract Class, Collapse Hierarchy

- วิธี Move Method ใช้โดยการย้ายเมทธอดที่อยู่ใน God Class ไปยังคลาสอื่น ซึ่งจะทำให้หน้าที่การทำงานของ God Class ลดน้อยลง
- วิธี Move Method ใช้โดยการย้ายเมทธอดที่ถูกเรียกใช้งานโดย God Class มาไว้ที่ God Class แล้วใช้วิธีการ Extract Class เพื่อแยกออกไปสร้างเป็นคลาสใหม่ ซึ่งจะทำให้หน้าที่การทำงานของ God Class ลดน้อยลง
- วิธี Collapse Hierarchy ใช้เมื่อ God Class มีการเรียกใช้งานคลาสอื่นที่เป็นคลาสแม่ ซึ่งวิธีนี้จะพิจารณาว่าหากไม่มีคลาสลูกอื่น ๆ เรียกใช้งานคุณสมบัติของคลาสแม่ก็จะย้ายคุณสมบัติของคลาสแม่ลงไปที่คลาสลูกทั้งหมด หลังจากนั้น God Class จะไม่มีการเรียกใช้งานคลาสอื่นซึ่งเคยเป็นคลาสแม่

ตัวอย่างของลักษณะข้อบกพร่องประเภท God Class ในโมเดลการออกแบบมีดังต่อไปนี้ โดยรูปที่ 3.18 แผนภาพคลาสของระบบที่มีคลาส B ที่เป็น God Class รูปที่ 3.19 แผนภาพซีเควนซ์ ของระบบที่มีคลาส B ที่เป็น God Class



รูปที่ 3.18 แผนภาพคลาสของระบบที่มีคลาสที่เป็น God Class



รูปที่ 3.19 แผนภาพซีเควนซ์ของระบบที่มีคลาสที่เป็น God Class

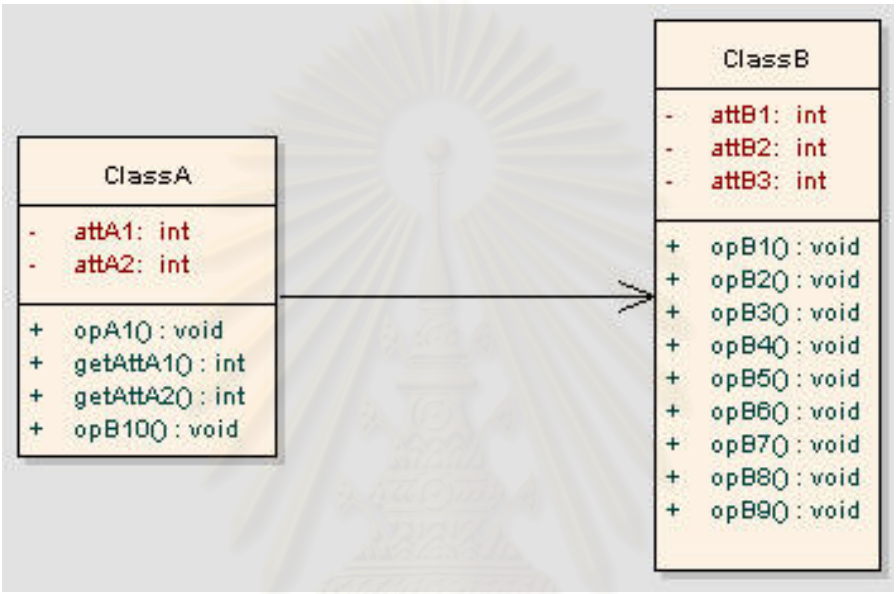
จากรูปที่ 3.18, 3.19 นำมาคำนวณมาตรวัดได้ผลดังตารางที่ 3.9

ตารางที่ 3.9 แสดงผลการคำนวณมาตรวัดของระบบที่มีคลาส B เป็น God Class

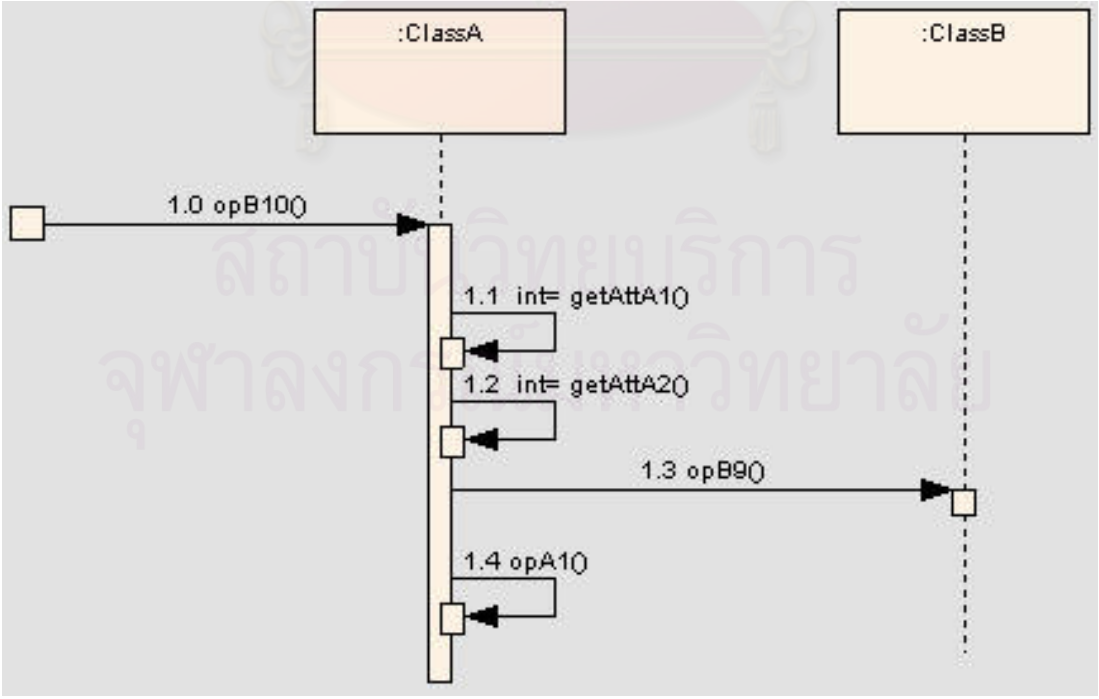
Class	ค่ามาตรวัด NCPA	ค่ามาตรวัด RFC	ค่ามาตรวัด COH
A	0	3	0
B	2	13	0

จากผลที่ได้ในตารางจะเห็นว่า $NCPA_B$ = 2, RFC_B = 13 และ COH_B = 0 แสดงว่าเป็น God Class เมื่อเทียบกับค่าช่วงมาตรวัดในกลยุทธ์ ดังนั้นจะทำการรีแฟคทอริงโดยการ Move Method opB10 จากในคลาส B มาไว้ในคลาส A แทน

จากการประยุกต์ใช้วิธีรีแฟคทอริงได้ผลดังรูปที่ 3.20, 3.21 และค่ามาตรวัดของระบบหลังจากทำรีแฟคทอริงแสดงดังตารางที่ 3.10 ซึ่งจะเห็นว่า $NCPA_B$ = 0, RFC_B = 9 และ COH_B = 0 จากค่ามาตวัดทั้งสามนี้ แสดงว่าคลาส B ไม่ได้มีข้อบกพร่องลักษณะ God Class อีกต่อไป



รูปที่ 3.20 แผนภาพคลาสของระบบหลังการรีแฟคทอริง



รูปที่ 3.21 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง

ตารางที่ 3.10 แสดงผลการคำนวณมาตรวัดของระบบที่มีเมทธอดที่เป็น God Class

Class	ค่ามาตรวัด NCPA	ค่ามาตรวัด RFC	ค่ามาตรวัด COH
A	0	5	0
B	0	9	0

3.2.2.6 Switch Statements

1) คำนิยาม คือ ลักษณะของการใช้ Switch Statements เปรียบเทียบเงื่อนไขเพื่อให้ระบบมีการทำงานที่แตกต่างกันไปตามเงื่อนไขต่าง ๆ ลักษณะการเกิด Switch Statements ทำให้เป็นปัญหาในการเกิดความซ้ำซ้อนของโปรแกรม หากมี Switch Statements อยู่หลายที่ในโปรแกรม เมื่อมีความจำเป็นต้องเพิ่มโค้ดใน Switch Statements ใด มักจะต้องตามไปเพิ่มใน Switch Statements อื่นด้วยเสมอ และลักษณะของ Switch Statements ที่เกิดขึ้นอยู่บ่อยครั้ง คือ การเปรียบเทียบที่มีเงื่อนไขในการเปรียบเทียบเป็นประเภทของออบเจ็กต์ ซึ่งหากมีออบเจกต์ประเภทใหม่เกิดขึ้น จะทำให้ต้องเพิ่มโค้ดใน Switch Statements ในทุก ๆ Switch Statements ด้วย ลักษณะแบบที่ทำให้การเปลี่ยนแปลงของซอฟต์แวร์ยากมากขึ้น

2) วิธีการวัด

- พิจารณาที่ระดับโมเดลการออกแบบว่ามีการใช้งาน Switch Statements อยู่

3) การกำหนดมาตรวัด

- มาตรวัด Number of Switch Statements (NOSS)

คือจำนวน Switch Statements ในโมเดลการออกแบบ ซึ่งจะพิจารณาจำนวนจากคอมไบน์แฟรกเมนต์ (Combined Fragment) ที่อยู่ในแผนภาพซีเควนซ์ที่มีเงื่อนไขมากกว่า 2 เงื่อนไข

4) กลยุทธ์การตรวจจับ

$$SwitchStatements(S) = S' \mid S' \subseteq S, \forall U \in S'$$
$$NOSS > 0$$

กำหนดให้ *U* คือ โมเดลการออกแบบที่สงสัยว่ามีข้อบกพร่อง

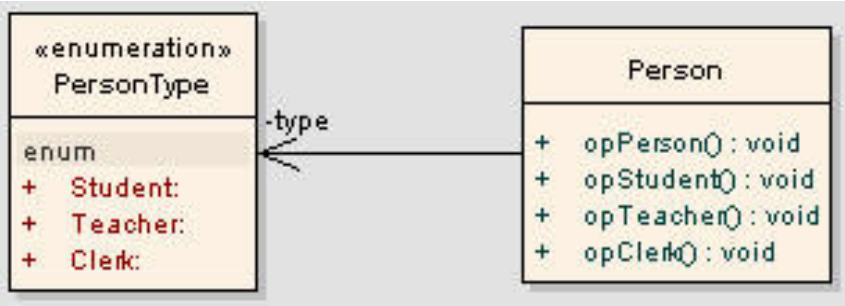
- ข้อกำหนดของค่ามาตรวัดที่บอกถึงข้อบกพร่องคือคลาสที่มีค่าวัดอยู่ในช่วงที่กำหนดในกลยุทธ์ นั่นคือหากโมเดลการออกแบบที่พิจารณามีจำนวนคอมไบน์แฟรกเมนต์ที่มีเงื่อนไขมากกว่า 2 ขึ้นไปอย่างน้อยหนึ่งคอมไบน์แฟรกเมนต์ แสดงว่าโมเดลการออกแบบที่พิจารณานั้นมีข้อบกพร่องประเภท Switch Statements
- งานวิจัยนี้นำมาตรวัด NOSS จาก [13] มาประยุกต์ให้เป็นมาตรวัดสำหรับโมเดลการออกแบบ แต่ยังคงใช้ช่วงของมาตรวัดที่บอกถึงข้อบกพร่องเป็นค่าเดียวกับงานวิจัย [13] ดังนั้นงานวิจัยนี้บทที่ 5 จึงไม่ได้มีการคำนวณหาช่วงของข้อกำหนดของมาตรวัดสำหรับตรวจจับที่ระบุถึงข้อบกพร่องประเภทนี้

5) วิธีการรีแฟคทอริง

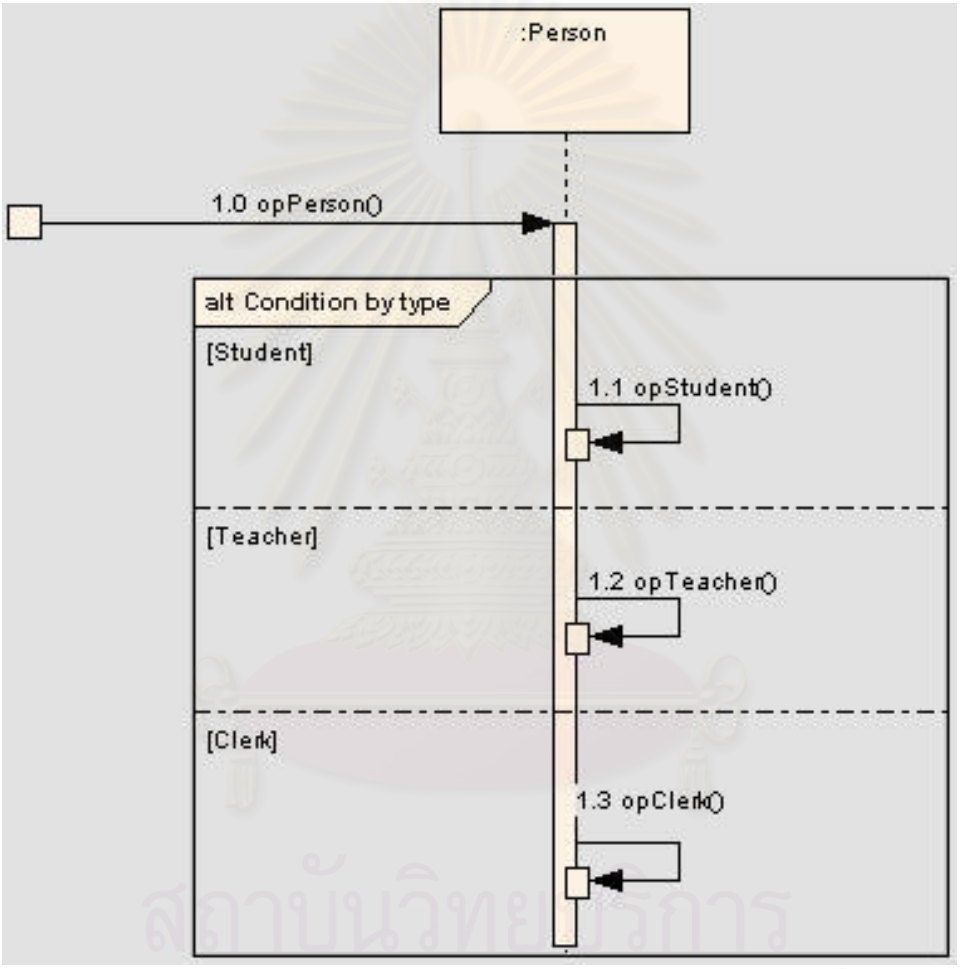
การประยุกต์วิธีรีแฟคทอริงสำหรับ Switch Statements มี 7 วิธีได้แก่การ Extract Method, Move Method, Replace Type Code with Subclasses, Replace Type Code with State/Strategy, Replace Conditional with Polymorphism, Replace Parameter with Explicit Methods, และ Introduce Null Object

- วิธี Extract Method ใช้เพื่อกระจายโค้ดส่วนที่เป็น Switch Statements ออกมาสร้างเป็นเมทธอดใหม่ และทำการย้ายเมทธอดใหม่นี้ไปยังคลาสที่จะทำโพลิมอฟิซึม(Polymorphism) หลังจากนั้นจะใช้วิธี Replace Type Code with Subclasses หรือ Replace Type Code with State/Strategy เพื่อใช้แทนที่โค้ดที่ถูกย้ายไป และใช้วิธี Replace Conditional with Polymorphism แทนการใช้ Switch Statements เดิม
- วิธี Replace Parameter with Explicit Methods จะใช้เมื่อบางเคสสเตทเมนต์ (Case statement) ส่งผลกระทบกับเมทธอดใดเพียงเมทธอดเดียว
- วิธี Introduce null object ประยุกต์ใช้เมื่อพบว่าเงื่อนไขหนึ่งในเคสสเตทเมนต์เป็นค่าว่าง (Null)

ตัวอย่างของลักษณะข้อบกพร่องประเภท Switch Statements ในโมเดลการออกแบบมีดังต่อไปนี้ โดยรูปที่ 3.22 แผนภาพคลาสของระบบที่มี Switch Statements รูปที่ 3.23 แผนภาพซีเควนซ์ ของระบบที่มี Switch Statements



รูปที่ 3.22 แผนภาพคลาสของระบบที่มี Switch Statements

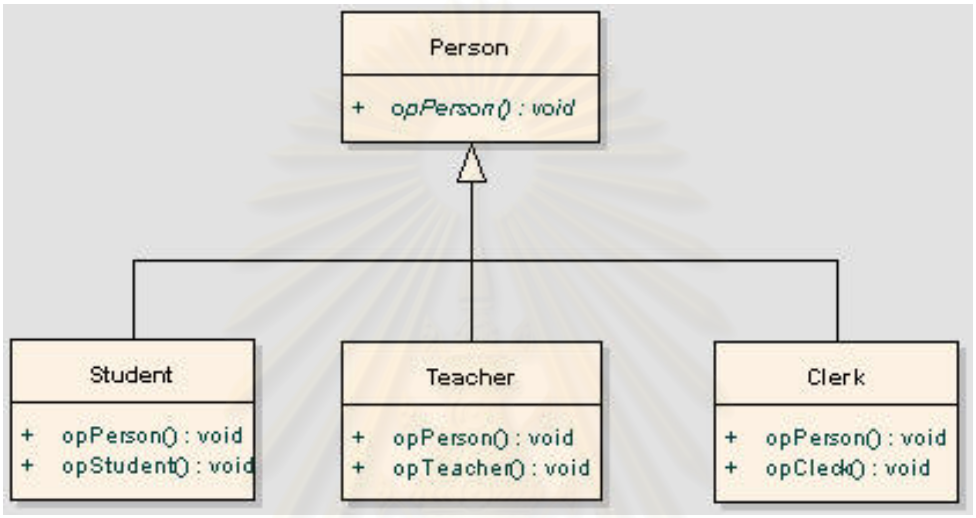


รูปที่ 3.23 แผนภาพซีเควนซ์ของระบบที่มี Switch Statements

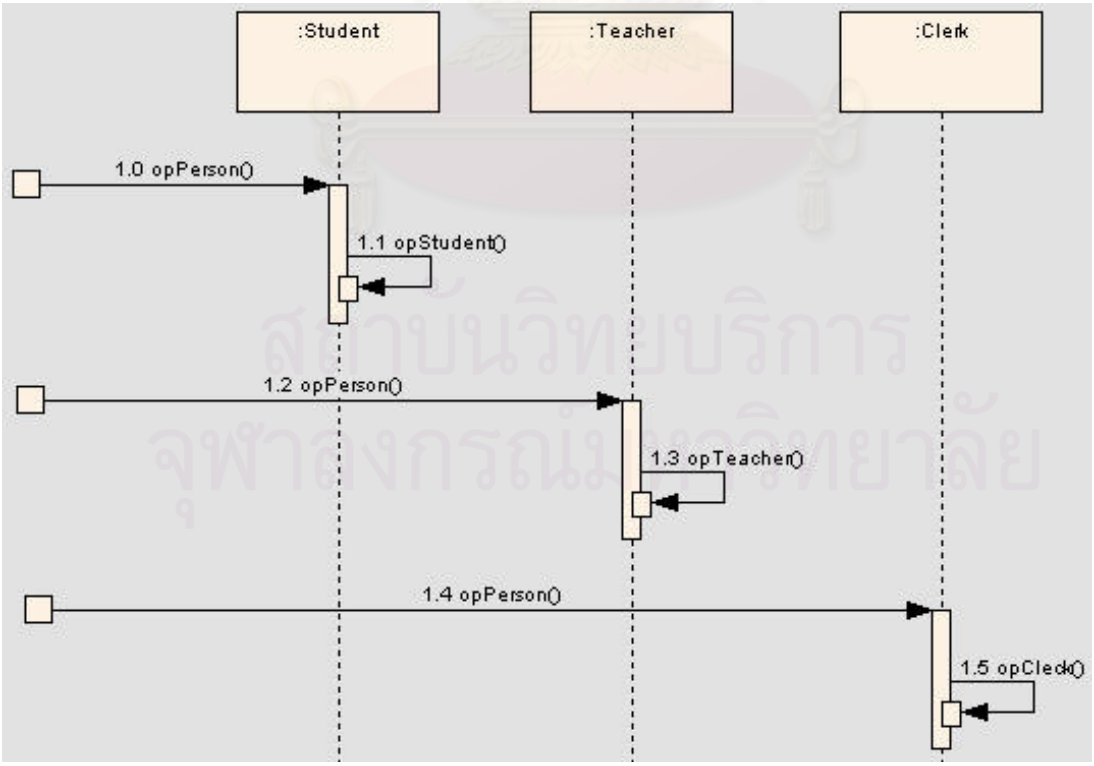
จากรูปที่ 3.22, 3.23 นำมาคำนวณมาตรวัด NOSS ได้มีค่าเป็น 1 ซึ่งมีค่ามากกว่า 0 ตามข้อกำหนดในกลยุทธ์ ดังนั้นใช้วิธีการรีแฟคทอริงที่เริ่มด้วยการ Extract Method ในส่วนของแต่ละกรณีใน Switch Statements ออกมาเป็นเมทธอดต่างหากซึ่งในกรณีนี้ เมทธอด opStudent(), opTeacher() และ opClerk() เป็น เมทธอดที่แยกอยู่อย่างอิสระอยู่แล้ว จึงข้ามไปทำขั้นตอนต่อไป คือ Replace Type Code with Subclasses ได้เลย ในที่นี้คลาสลูก (Subclass) ของคลาส Person ถูกสร้างขึ้นมาใหม่ด้วยกัน 3 คลาส คือ คลาส Teacher, คลาส Student และ คลาส Clerk จากนั้นทำ

Replace Conditional with Polymorphism ต่อโดยการทำเมทธอด opPerson() ของคลาส Person ให้เป็นเมทธอดที่มีหลายรูปร่าง (Polymorphic) แล้วให้ คลาสลูกทั้งสามนำเมทธอดนี้ไปใช้โดยมีลักษณะที่แต่งต่างกันดังรูปที่ 3.25

จากการประยุกต์ใช้วิธีรีแฟคทอริงได้ผลดังรูปที่ 3.24, 3.25 และค่ามาตรวัด NOSS ของระบบหลังจากทำรีแฟคทอริงกลายเป็น 0 ซึ่งไม่อยู่ในช่วงของค่ามาตรวัดที่บอกถึงข้อบกพร่องลักษณะ Switch Statements



รูปที่ 3.24 แผนภาพคลาสของระบบหลังการรีแฟคทอริง



รูปที่ 3.25 แผนภาพซีเควนซ์ของระบบหลังการรีแฟคทอริง

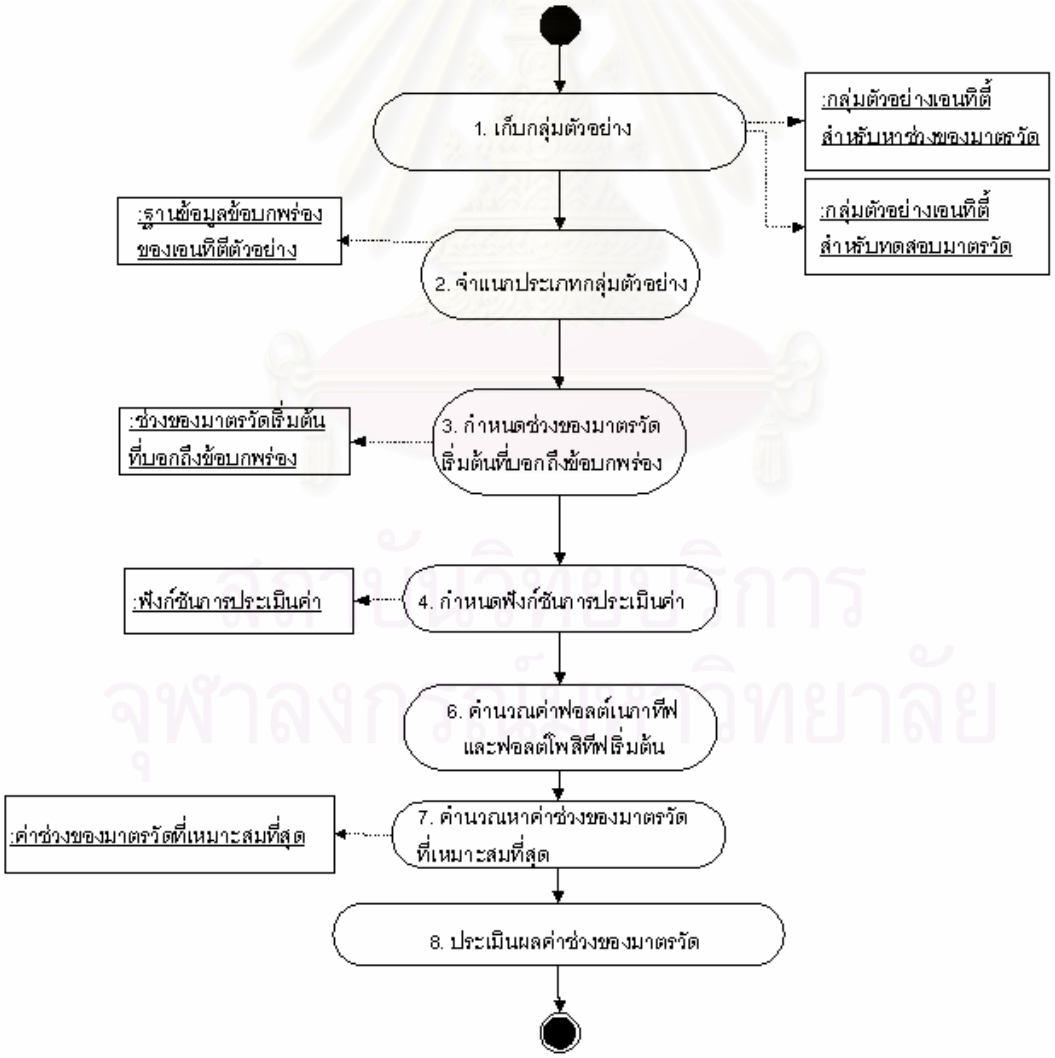
บทที่ 4

การกำหนดช่วงของมาตรวัด

ในบทนี้จะอธิบายถึงวิธีการกำหนดช่วงของมาตรวัดที่ใช้ในงานวิจัยนี้ ที่นำวิธีการของ[15] มาประยุกต์ใช้ รวมถึงผลการคำนวณหาช่วงของมาตรวัดที่เหมาะสมที่สุด

4.1 วิธีการกำหนดช่วงของมาตรวัด

การกำหนดช่วงของมาตรวัดเป็นการหาค่าช่วงของมาตรวัดที่เหมาะสมที่สุด เพื่อนำไปใช้ในกลยุทธ์การตรวจจับข้อบกพร่อง ในวิทยานิพนธ์นี้ได้นำวิธีการหาค่าช่วงของมาตรวัดที่เหมาะสมที่สุดจาก [15] มาประยุกต์ใช้ ขั้นตอนการหาค่าช่วงของมาตรวัดแสดงดังแผนภาพแอคทิวิตีในรูปที่ 4.1



รูปที่ 4.1 ขั้นตอนการหาค่าช่วงของมาตรวัด

จากรูปที่ 4.1 วิธีการตรวจจับข้อบกพร่องมีทั้งหมด 7 ขั้นตอนประกอบด้วย

4.1.1 การเก็บกลุ่มตัวอย่าง

ในขั้นตอนนี้จะเป็นการเก็บระบบตัวอย่างมาทั้งหมด 40 ระบบต่อข้อบกพร่องของโมเดลการออกแบบหนึ่งประเภทโดยจะใช้ระบบตัวอย่างเดียวกับงานวิจัย [18] ซึ่งใช้ระบบตัวอย่างจากหนังสือและงานวิจัยต่าง ๆ และจะแบ่งระบบตัวอย่างออกเป็นสองกลุ่มคือกลุ่มตัวอย่างสำหรับหาช่วงของมาตรวัดหรือเซ็ตทูนนิง 35 ตัวอย่าง และกลุ่มตัวอย่างสำหรับทดสอบมาตรวัดหรือเซ็ตแวลิเดชัน 5 ตัวอย่าง โดยจะนำข้อมูลการแบ่งกลุ่มตัวอย่างสำหรับข้อบกพร่องแต่ละประเภทเก็บไว้ในเอกสารเอ็กซ์เอ็มแอล (XML: Extensible Markup Language) และระบุประเภทของข้อบกพร่องว่าเป็นข้อบกพร่องที่เอนทิตี้แบบใด รูปที่ 4.2 แสดงตัวอย่างเอกสารเอ็กซ์เอ็มแอลที่ได้จากขั้นตอนนี้ โดยข้อบกพร่องในรูปคือ God Class และเป็นข้อบกพร่องที่เอนทิตี้แบบคลาส

```
<ClassDefect name="GodClass">
  <TuningSet>
    <Model file="Ex1.xml"/>
    <Model file="Ex2.xml"/>
    <Model file="Ex3.xml"/>
    <Model file="Ex4.xml"/>
    <Model file="Ex5.xml"/>
    <Model file="Ex6.xml"/>
    <Model file="Ex7.xml"/>
    <Model file="Ex8.xml"/>
    <Model file="Ex9.xml"/>
    <Model file="Ex10.xml"/>
    <Model file="Ex11.xml"/>
    <Model file="Ex12.xml"/>
    <Model file="Ex13.xml"/>
    <Model file="Ex14.xml"/>
    <Model file="Ex15.xml"/>
    <Model file="Ex16.xml"/>
    <Model file="Ex17.xml"/>
    <Model file="Ex18.xml"/>
    <Model file="Ex19.xml"/>
    <Model file="Ex20.xml"/>
    <Model file="Ex21.xml"/>
    <Model file="Ex22.xml"/>
    <Model file="Ex23.xml"/>
    <Model file="Ex24.xml"/>
    <Model file="Ex25.xml"/>
    <Model file="Ex26.xml"/>
    <Model file="Ex27.xml"/>
    <Model file="Ex28.xml"/>
    <Model file="Ex29.xml"/>
    <Model file="Ex30.xml"/>
    <Model file="Ex31.xml"/>
    <Model file="Ex32.xml"/>
    <Model file="Ex33.xml"/>
    <Model file="Ex34.xml"/>
    <Model file="Ex35.xml"/>
  </TuningSet>
  <ValidationSet>
    <Model file="Ex36.xml"/>
    <Model file="Ex37.xml"/>
    <Model file="Ex38.xml"/>
    <Model file="Ex39.xml"/>
    <Model file="Ex40.xml"/>
  </ValidationSet>
</ClassDefect>
```

รูปที่ 4.2 ตัวอย่างเอกสารเอ็กซ์เอ็มแอลสำหรับเก็บข้อมูลกลุ่มตัวอย่างของข้อบกพร่องประเภท God Class

4.1.2 การจำแนกประเภทกลุ่มตัวอย่าง

ในขั้นตอนนี้จะเป็นการจำแนกประเภทว่าเอนทิตี้ในกลุ่มตัวอย่างมีข้อบกพร่องของโมเดลการออกแบบแบบใดโดยผู้เชี่ยวชาญซึ่งในที่นี้ได้แก่ผู้จัดทำวิทยานิพนธ์และเก็บข้อมูลในเอกสารเอ็กซ์เอ็มแอล และในเอกสารเอ็กซ์เอ็มแอลจะเก็บเฉพาะข้อมูลของเอนทิตี้ที่มีข้อบกพร่องเท่านั้น ซึ่งเอนทิตี้ประเภทนี้ก็คือเอนทิตี้ที่เป็นกลุ่มตัวอย่างโพสิทีฟตามงานวิจัย [15] นั่นเอง

ในการระบุการบกพร่องจะต้องระบุประเภทของเอนทิตี้ที่มีข้อบกพร่องให้ตรงกับประเภทของเอนทิตี้ตามแต่ละข้อบกพร่อง ประเภทของเอนทิตี้ในงานวิจัยนี้มี 3 ประเภทได้แก่โมเดลการออกแบบ, คลาส, และเมทธอด การระบุประเภทของเอนทิตี้ในเอกสารเอ็กซ์เอ็มแอลก็จะมี 3 ประเภทคือ ModelDesignEntity, ClassDesignEntity, และ OperationDesignEntity ซึ่งรูปที่ 4.3 แสดงตัวอย่างเอกสารเอ็กซ์เอ็มแอลที่ได้จากขั้นตอนนี้ โดยข้อบกพร่องในรูปคือ God Class ซึ่งเป็นข้อบกพร่องของเอนทิตี้ระดับคลาส ดังนั้นเอนทิตี้ที่มีข้อบกพร่องประเภท God Class ในเอกสารนี้จะอยู่ในประเภท ClassDesignEntity รวมถึงต้องมีการระบุชื่อคลาสในเอกสารเอ็กซ์เอ็มแอลด้วย

```
<DefectEntityDefects>
  <ModelDesignEntity file="Ex9.xml">
    <Defect name="SwitchStatement"/>
  </ModelDesignEntity>
  <ModelDesignEntity file="Ex13.xml">
    <Defect name="SwitchStatement"/>
  </ModelDesignEntity>
  <ModelDesignEntity file="Ex16.xml">
    <Defect name="SwitchStatement"/>
  </ModelDesignEntity>
  <ClassDesignEntity name="Customer" file="Ex1.xml">
    <Defect name="GodClass"/>
  </ClassDesignEntity>
  <ClassDesignEntity name="Bank" file="Ex1.xml">
    <Defect name="MiddleMan"/>
  </ClassDesignEntity>
  <ClassDesignEntity name="PurchaseOrder" file="Ex3.xml">
    <Defect name="MiddleMan"/>
  </ClassDesignEntity>
  <ClassDesignEntity name="RawMaterialOrder" file="Ex3.xml">
    <Defect name="MiddleMan"/>
  </ClassDesignEntity>
  <OperationDesignEntity name="postReturnMaterial" className="ReturnedMaterial" file="Ex37.xml">
    <Defect name="FeatureEnvy"/>
  </OperationDesignEntity>
  <OperationDesignEntity name="showHistoricalStudy" className="Student" file="Ex4.xml">
    <Defect name="MessageChains"/>
  </OperationDesignEntity>
  <OperationDesignEntity name="viewBillInfo" className="Bill" file="Ex8.xml">
    <Defect name="MessageChains"/>
  </OperationDesignEntity>
  <OperationDesignEntity name="displayMemberRental" className="RentalList" file="Ex14.xml">
    <Defect name="MessageChains"/>
  </OperationDesignEntity>
  <OperationDesignEntity name="create_cd_list" className="CD List" file="Ex19.xml">
    <Defect name="MessageChains"/>
  </OperationDesignEntity>
  <OperationDesignEntity name="getSeminarInfo" className="SeminarHistory" file="Ex22.xml">
    <Defect name="MessageChains"/>
  </OperationDesignEntity>
  <OperationDesignEntity name="viewDocumentFlow" className="PurchaseOrder" file="Ex37.xml">
    <Defect name="MessageChains"/>
  </OperationDesignEntity>
  <OperationDesignEntity name="getTransactions" className="Member" file="Ex38.xml">
    <Defect name="MessageChains"/>
  </OperationDesignEntity>
</DefectEntityDefects>
```

รูปที่ 4.3 ตัวอย่างเอกสารเอ็กซ์เอ็มแอลที่เก็บผลการระบุข้อบกพร่องของเอนทิตี้ตัวอย่าง

4.1.3 การกำหนดช่วงของค่ามาตรวัดเริ่มต้นที่บอกถึงข้อบกพร่อง

ในขั้นตอนนี้จะเป็นการกำหนดว่าสำหรับแต่ละกลยุทธ์การตรวจจับข้อบกพร่องนั้น จะกำหนดค่าเริ่มต้นของช่วงค่ามาตรวัดแต่ละมาตรวัดในกลยุทธ์เป็นค่าใด ซึ่งค่าเริ่มต้นนี้จะเป็นอินพุทสำหรับการคำนวณหาค่าช่วงที่เหมาะสมในขั้นตอนที่ 4.1.7 และการกำหนดค่าช่วงเริ่มต้นจะเก็บในเอกสารเอ็กซ์เอ็มแอล

ตัวอย่างเช่นการกำหนดค่าช่วงของมาตรวัดสำหรับกลยุทธ์ตรวจจับข้อบกพร่องประเภท Message Chains กำหนดว่าค่าช่วงของมาตรวัด SOMsgCh เริ่มต้นสำหรับกลยุทธ์เป็น 3 รูปที่ 4.4 แสดงตัวอย่างเอกสารเอ็กซ์เอ็มแอลที่ได้จากขั้นตอนนี้

```
1  <strategy>
2      <thresholdA>3</thresholdA>
3  </strategy>
```

รูปที่ 4.4 ตัวอย่างเอกสารเอ็กซ์เอ็มแอลที่เก็บค่าช่วงของมาตรวัดเริ่มต้นของข้อบกพร่องประเภท Message Chains

4.1.4 การกำหนดฟังก์ชันการประเมินค่า

งานวิจัยนี้จะใช้ฟังก์ชันการประเมินค่าฟังก์ชันเดียวกับที่ใช้ใน [15] ดังแสดงในหัวข้อ 2.2.3

4.1.5 การคำนวณค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟเริ่มต้น

การคำนวณค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟเริ่มต้นนี้ จะคำนวณจากการใช้กลยุทธ์การตรวจจับข้อบกพร่องสำหรับข้อบกพร่องแต่ละประเภทด้วยค่าช่วงของมาตรวัดเริ่มต้น เพื่อนำค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟมาใช้ในการประเมินผลการคำนวณหาค่าช่วงที่เหมาะสมว่าค่าช่วงที่ได้จากทูนนิงแมชชีน ทำให้ค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟลดลงจริงหรือไม่

4.1.6 การคำนวณหาค่าช่วงของมาตรวัดที่เหมาะสมที่สุด

ในขั้นตอนนี้จะเป็นการใช้งานอัลกอริธึมเพื่อคำนวณหาค่าช่วงของมาตรวัดที่เหมาะสมที่สุดสำหรับแต่ละกลยุทธ์การตรวจจับข้อบกพร่อง โดยการคำนวณจะนำกลุ่มตัวอย่างที่เป็นเซ็ตทูนนิงที่ได้จากขั้นตอน 4.1.1, ผลการระบุประเภทของข้อบกพร่องของเอนทิตี้จากขั้นตอน 4.1.2, ค่าช่วงของมาตรวัดเริ่มต้นจากขั้นตอนที่ 4.1.3, ฟังก์ชันการประเมินค่าจากขั้นตอน 4.1.4 มาใช้ในการประมวลผลอัลกอริธึม ซึ่งผลลัพธ์จากกากประมวลผลอัลกอริธึมนี้ จะได้เป็นค่าช่วงของมาตรวัดสำหรับแต่ละมาตรวัดที่อยู่ในกลยุทธ์การตรวจจับข้อบกพร่องแต่ละประเภท

4.1.7 การประเมินผลค่าช่วงของมาตรวัด

ขั้นตอนนี้จะเป็นการประมวลผลค่าช่วงของมาตรวัดที่ได้จากขั้นตอน 4.1.6 โดยจะนำค่าช่วงที่ได้มาเป็นอินพุทในการตรวจจับข้อบกพร่อง ซึ่งจะใช้กลุ่มตัวอย่างที่เป็นเซ็ตแวลิเดชันมาประมวลผล เพื่อตรวจสอบว่าหลังจากใช้ช่วงของค่ามาตรวัดที่ได้จากขั้นตอน 4.1.6 มาตรวจจับข้อบกพร่องของเซ็ตแวลิเดชันแล้ว ค่าของฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟเป็นเท่าใดบ้าง มีค่าลดลงหรือไม่เมื่อเปรียบเทียบกับค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟเริ่มต้นจากขั้นตอนที่ 4.1.5

4.2 ผลการคำนวณหาช่วงของมาตรวัดที่เหมาะสมที่สุด

4.2.1 ผลการคำนวณค่าช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่องประเภท Feature Envy

4.2.1.1 การกำหนดค่าเริ่มต้นสำหรับประมวลผล

สำหรับข้อบกพร่องประเภท Feature Envy นี้ ผู้วิจัยได้กำหนดค่าเริ่มต้นของค่าช่วงมาตรวัดและค่าพารามิเตอร์สำหรับฟังก์ชันการประเมินค่าเป็นดังนี้

1) ค่าช่วงเริ่มต้นสำหรับมาตรวัด

กำหนดค่าเริ่มต้นของช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่องประเภท Feature Envy เป็น NSMsg/ NMsg < 0.5

2) ค่าพารามิเตอร์สำหรับฟังก์ชันการประเมินค่าเป็นดังนี้

- A = 100
- b = 0.7 เนื่องจากต้องการให้น้ำหนักความสำคัญของจำนวนฟอลส์เนกาทีฟมากกว่าจำนวนฟอลส์โพสิทีฟ

4.2.1.2 ผลการคำนวณค่าช่วงที่เหมาะสมสำหรับมาตรวัด

ผลการคำนวณค่าช่วงที่เหมาะสมสำหรับมาตรวัดแสดงดังตารางที่ 4.1 จะเห็นได้ว่าค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟเริ่มต้นสำหรับเซ็ตทูนนิงมีเป็น 1 และ 11 ตามลำดับ หลังจากนั้นทดลองประเมินค่าด้วยอัลกอริทึมฟาสต์ซีมูเลทเต็ดแอนนิลลิง 5 ครั้ง ผลของการคำนวณค่าช่วงของมาตรวัดทั้ง 5 ครั้งทำให้เกิดจำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟไม่แตกต่างกัน และจำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟในเซ็ตทูนนิงเป็น 1 และ 3 ตามลำดับ ซึ่งลดลงเมื่อเปรียบเทียบกับการใช้ค่าช่วงของมาตรวัดเริ่มต้น จากการคำนวณในเซ็ตแวลิเดชันให้ผลว่าจำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟลดลงจากการใช้ค่าช่วงเริ่มต้นเช่นเดียวกัน

ต้นฉบับไม่มีหน้า 63 -64

NO PAGE 63 -64 IN ORIGINAL

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.1 แสดงผลการคำนวณค่าช่วงของมาตรวัดที่ใช้ในการตรวจจับข้อบกพร่องประเภท Feature Envy

				ค่าเริ่มต้น	ผลการคำนวณครั้งที่ 1	ผลการคำนวณครั้งที่ 2	ผลการคำนวณครั้งที่ 3	ผลการคำนวณครั้งที่ 4	ผลการคำนวณครั้งที่ 5
ค่าของมาตรวัด			NSMsg / NMsg	0.5	0.28	0.29	0.31	0.28	0.28
จำนวนในเซ็ตทูนนิง	กลุ่มตัวอย่างโพสิทีฟ	6	กลุ่มตัวอย่างฟอลส์เนกาทีฟ	1	1	1	1	1	1
	กลุ่มตัวอย่างเนกาทีฟ	1085	กลุ่มตัวอย่างฟอลส์โพสิทีฟ	11	3	3	3	3	3
จำนวนในเซ็ตแวลิเดชัน	กลุ่มตัวอย่างโพสิทีฟ	2	กลุ่มตัวอย่างฟอลส์เนกาทีฟ	0	0	0	0	0	0
	กลุ่มตัวอย่างเนกาทีฟ	306	กลุ่มตัวอย่างฟอลส์โพสิทีฟ	5	2	2	2	2	2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.2.1.3 ค่าช่วงสำหรับตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Feature Envy

จากผลการคำนวณช่วงของมาตรวัดสามารถสรุปได้ว่าค่าช่วงที่เหมาะสมในการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Feature Envy ได้แก่ 0.31 ดังนั้นกลยุทธ์ในการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Feature Envy จะเป็นดังนี้

$$FeatureEnvy(S) = S' \mid S' \subseteq S, \forall M \in S'$$
$$(NSMsg(M) / NMsg(M) < 0.31$$

กำหนดให้ *M* คือ เมทธอดที่สงสัยว่ามีข้อบกพร่อง

4.2.2 ผลการคำนวณค่าช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่องประเภท Message Chains

4.2.2.1 การกำหนดค่าเริ่มต้นสำหรับประมวลผล

สำหรับข้อบกพร่องประเภท Message Chains นี้ ผู้วิจัยได้กำหนดค่าเริ่มต้นของค่าช่วงมาตรวัดและค่าพารามิเตอร์สำหรับฟังก์ชันการประเมินค่าเป็นดังนี้

1) ค่าช่วงเริ่มต้นสำหรับมาตรวัด

มาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่องประเภท Message Chains ได้แก่มาตรวัด SOMsgCh โดยจะกำหนดค่าเริ่มต้นของช่วงของมาตรวัดเป็น SOMsgCh ≥ 3

2) ค่าพารามิเตอร์สำหรับฟังก์ชันการประเมินค่าเป็นดังนี้

- A = 100

- b = 0.5 เนื่องจากต้องการให้น้ำหนักความสำคัญของจำนวนฟอลส์เนกาทีฟเท่ากับจำนวนฟอลส์โพสิทีฟ

4.2.2.2 ผลการคำนวณค่าช่วงที่เหมาะสมสำหรับมาตรวัด

ผลการคำนวณค่าช่วงที่เหมาะสมสำหรับมาตรวัดแสดงดังตารางที่ 4.2

ตารางที่ 4.2 แสดงผลการคำนวณค่าช่วงของมาตรวัดที่ใช้ในการตรวจจับข้อบกพร่องประเภท Message Chains

				ค่าเริ่มต้น	ผลการคำนวณครั้งที่ 1	ผลการคำนวณครั้งที่ 2	ผลการคำนวณครั้งที่ 3	ผลการคำนวณครั้งที่ 4	ผลการคำนวณครั้งที่ 5
ค่าของมาตรวัด			SOMsgCh	3	2	2	2	2	2
จำนวนในเซ็ตทูนนิง	กลุ่มตัวอย่างโพสิทีฟ	5	กลุ่มตัวอย่างฟอลส์เนกาทีฟ	3	0	0	0	0	0
	กลุ่มตัวอย่างเนกาทีฟ	1086	กลุ่มตัวอย่างฟอลส์โพสิทีฟ	0	0	0	0	0	0
จำนวนในเซ็ตแวลิเดชัน	กลุ่มตัวอย่างโพสิทีฟ	2	กลุ่มตัวอย่างฟอลส์เนกาทีฟ	1	0	0	0	0	0
	กลุ่มตัวอย่างเนกาทีฟ	306	กลุ่มตัวอย่างฟอลส์โพสิทีฟ	0	0	0	0	0	0

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จากตารางที่ 4.2 ค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟเริ่มต้นสำหรับเซ็ตทูนนิงมีเป็น 3 และ 0 ตามลำดับ หลังจากนั้นเมื่อได้ทดลองประเมินค่าด้วยอัลกอริทึมฟาสต์ซีมูเลทเต็ดแอนนิลลิง 5 ครั้ง ผลของการคำนวณค่าช่วงของมาตรวัดทั้ง 5 ครั้งทำให้เกิดจำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟไม่แตกต่างกัน และจำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟในเซ็ตทูนนิงเป็น 0 และ 0 ตามลำดับ ซึ่งลดลงเมื่อเปรียบเทียบกับการใช้ค่าช่วงของมาตรวัดเริ่มต้น จากการคำนวณในเซ็ตแวลิเดชันให้ผลว่าจำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟลดลงจากการใช้ค่าช่วงเริ่มต้น
เช่นเดียวกัน

4.2.2.3 ค่าช่วงสำหรับตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Message Chains

จากผลการคำนวณช่วงของมาตรวัดสามารถสรุปได้ว่าค่าช่วงที่เหมาะสมในการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Message Chains ได้แก่ 2 ดังนั้นกลยุทธ์ในการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Message Chains จะเป็นดังนี้

$$MessageChains(S) = S' \mid S' \subseteq S, \forall M \in S'$$
$$SOMsgCh \geq 2$$

กำหนดให้ *M* คือ เมทธอดที่สงสัยว่ามีข้อบกพร่อง

4.2.3 ผลการคำนวณค่าช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่องประเภท Middle Man

4.2.3.1 การกำหนดค่าเริ่มต้นสำหรับประมวลผล

สำหรับข้อบกพร่องประเภท Middle Man นี้ ผู้วิจัยได้กำหนดค่าเริ่มต้นของค่าช่วงมาตรวัดและค่าพารามิเตอร์สำหรับฟังก์ชันการประเมินค่าเป็นดังนี้

1) ค่าช่วงเริ่มต้นสำหรับมาตรวัด

มาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่องประเภท Middle Man ได้แก่ มาตรวัด WOD โดยจะกำหนดค่าเริ่มต้นของช่วงของมาตรวัดเป็น WOD ≥ 0.5

2) ค่าพารามิเตอร์สำหรับฟังก์ชันการประเมินค่าเป็นดังนี้

- A = 100

- b = 0.7 เนื่องจากต้องการให้น้ำหนักความสำคัญของจำนวนฟอลส์เนกาทีฟมากกว่าจำนวนฟอลส์โพสิทีฟ

4.2.3.2 ผลการคำนวณค่าช่วงที่เหมาะสมสำหรับมาตรวัด

ผลการคำนวณค่าช่วงที่เหมาะสมสำหรับมาตรวัดแสดงดังตารางที่ 4.3 ซึ่งค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟเริ่มต้นสำหรับเซ็ตทูนนิงมีเป็น 6 และ 8 ตามลำดับ หลังจากนั้นเมื่อได้ทดลองประเมินค่าด้วยอัลกอริทึมฟาสต์ซีมูเลทเต็ดแอนนิลลิง 5 ครั้ง ผลของการคำนวณค่าช่วงของมาตรวัดทั้ง 5 ครั้งทำให้เกิดจำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟไม่แตกต่างกัน และจำนวนฟอลส์เนกาทีฟในเซ็ตทูนนิงและเซ็ตแวลิเดชันเป็น 0 ซึ่งลดลงเมื่อเปรียบเทียบกับการใช้ค่าช่วงของมาตรวัดเริ่มต้น แต่จำนวนฟอลส์โพสิทีฟในเซ็ตทูนนิงและเซ็ตแวลิเดชันเพิ่มขึ้นเมื่อเปรียบเทียบกับการใช้ค่าช่วงของมาตรวัดเริ่มต้น ซึ่งมีสาเหตุมาจากการกำหนดค่าพารามิเตอร์ b ในฟังก์ชันการประเมินค่าให้กับค่าฟอลส์เนกาทีฟมากกว่าค่าฟอลส์โพสิทีฟ

4.2.3.3 ค่าช่วงสำหรับตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Middle Man

จากผลการคำนวณหาช่วงของมาตรวัดสามารถสรุปได้ว่าค่าช่วงที่เหมาะสมในการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Middle Man ได้แก่ 0.43 ดังนั้นกลยุทธ์ในการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Middle Man จะเป็นดังนี้

$$MiddleMan(S) = S' \mid S' \subseteq S, \forall C \in S'$$
$$WOD > 0.43$$

กำหนดให้ *C* คือ คลาสที่สงสัยว่ามีข้อบกพร่อง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.3 แสดงผลการคำนวณค่าช่วงของมาตรวัดที่ใช้ในการตรวจจับข้อบกพร่องประเภท Middle Man

				ค่าเริ่มต้น	ผลการคำนวณครั้งที่ 1	ผลการคำนวณครั้งที่ 2	ผลการคำนวณครั้งที่ 3	ผลการคำนวณครั้งที่ 4	ผลการคำนวณครั้งที่ 5
ค่าของมาตรวัด			WOD	0.5	0.43	0.48	0.43	0.46	0.47
จำนวนในเซ็ตทูนนิง	กลุ่มตัวอย่างโพสิทีฟ	11	กลุ่มตัวอย่างฟอลส์เนกาทีฟ	6	0	0	0	0	0
	กลุ่มตัวอย่างเนกาทีฟ	448	กลุ่มตัวอย่างฟอลส์โพสิทีฟ	8	13	13	13	13	13
จำนวนในเซ็ตแวลิเดชัน	กลุ่มตัวอย่างโพสิทีฟ	3	กลุ่มตัวอย่างฟอลส์เนกาทีฟ	2	0	0	0	0	0
	กลุ่มตัวอย่างเนกาทีฟ	111	กลุ่มตัวอย่างฟอลส์โพสิทีฟ	1	2	2	2	2	2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.2.4 ผลการคำนวณค่าช่วงของมาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่องประเภท God Class

4.2.4.1 การกำหนดค่าเริ่มต้นสำหรับประมวลผล

สำหรับข้อบกพร่องประเภท God Class นี้ ผู้วิจัยได้กำหนดค่าเริ่มต้นของค่าช่วงมาตรวัดและค่าพารามิเตอร์สำหรับฟังก์ชันการประเมินค่าเป็นดังนี้

1) ค่าช่วงเริ่มต้นสำหรับมาตรวัด

มาตรวัดสำหรับกลยุทธ์การตรวจจับข้อบกพร่องประเภท God Class ได้แก่มาตรวัด NCPA, RFC, และ COH โดยจะกำหนดค่าเริ่มต้นของช่วงของมาตรวัดเป็นดังนี้

- NCPA > 2

- RFC > 5

- COH < 1

2) ค่าพารามิเตอร์สำหรับฟังก์ชันการประเมินค่าเป็นดังนี้

- A = 100

- b = 0.7 เนื่องจากต้องการให้น้ำหนักความสำคัญของจำนวนฟอลส์เนกาทีฟมากกว่าจำนวนฟอลส์โพสิทีฟ

4.2.4.2 ผลการคำนวณค่าช่วงที่เหมาะสมสำหรับมาตรวัด

ผลการคำนวณค่าช่วงที่เหมาะสมสำหรับมาตรวัดแสดงดังตารางที่ 4.4

จากตารางที่ 4.4 จะเห็นได้ว่า ค่าฟอลส์เนกาทีฟและค่าฟอลส์โพสิทีฟเริ่มต้นสำหรับเซ็ตทูนนิงมีเป็น 13 และ 0 ตามลำดับ หลังจากนั้นเมื่อได้ทดลองประเมินค่าด้วยอัลกอริทึมฟาสต์ซีมูเลทเต็ดแอนนิลลิง 5 ครั้ง ผลของการคำนวณค่าช่วงของมาตรวัดทั้ง 5 ครั้งทำให้เกิดจำนวนฟอลส์เนกาทีฟและฟอลส์โพสิทีฟไม่แตกต่างกัน และจำนวนฟอลส์เนกาทีฟในเซ็ตทูนนิงและเซ็ตแวลิเดชันเป็น 8 ซึ่งลดลงเมื่อเปรียบเทียบกับการใช้ค่าช่วงของมาตรวัดเริ่มต้น จำนวนฟอลส์เนกาทีฟในเซ็ตแวลิเดชันเป็น 3 ซึ่งไม่น้อยลงเมื่อเทียบกับการคำนวณด้วยค่าช่วงของมาตรวัดเริ่มต้น แต่จำนวนฟอลส์โพสิทีฟในเซ็ตทูนนิงและเซ็ตแวลิเดชันเพิ่มขึ้นเมื่อเปรียบเทียบกับการใช้ค่าช่วงของมาตรวัดเริ่มต้น ซึ่งมีสาเหตุมาจากการกำหนดค่าพารามิเตอร์ b ในฟังก์ชันการประเมินค่าให้กับค่าฟอลส์เนกาทีฟมากกว่าค่าฟอลส์โพสิทีฟ

ตารางที่ 4.4 แสดงผลการคำนวณค่าช่วงของมาตรวัดที่ใช้ในการตรวจจับข้อบกพร่องประเภท God Class

				ค่าเริ่มต้น	ผลการคำนวณ ครั้งที่ 1	ผลการ คำนวณ ครั้งที่ 2	ผลการ คำนวณ ครั้งที่ 3	ผลการ คำนวณ ครั้งที่ 4	ผลการ คำนวณ ครั้งที่ 5
ค่าของมาตรวัด			NCPA	2	1	1	1	1	1
			RFC	5	9	9	9	9	9
			COH	1	0.25	0.25	0.25	0.25	0.25
จำนวนในเซ็ตทูนนิง	กลุ่มตัวอย่างโพสิทีฟ	18	กลุ่มตัวอย่างฟอลส์เนกาทีฟ	13	8	8	8	8	8
	กลุ่มตัวอย่างเนกาทีฟ	437	กลุ่มตัวอย่างฟอลส์โพสิทีฟ	0	5	5	5	5	5
จำนวนในเซ็ตแวลิเดชัน	กลุ่มตัวอย่างโพสิทีฟ	5	กลุ่มตัวอย่างฟอลส์เนกาทีฟ	3	3	3	3	3	3
	กลุ่มตัวอย่างเนกาทีฟ	113	กลุ่มตัวอย่างฟอลส์โพสิทีฟ	0	1	1	1	1	1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.2.4.3 ค่าช่วงสำหรับตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท God Class

จากผลการคำนวณหาช่วงของมาตรวัด สามารถสรุปได้ว่าค่าช่วงที่เหมาะสมในการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท God Class สำหรับมาตรวัด NCPA มีค่าเป็น 1, สำหรับมาตรวัด RFC มีค่าเป็น 9 และสำหรับมาตรวัด COH มีค่าเป็น 0.25 ดังนั้นกลยุทธ์ในการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท God Class จะเป็นดังนี้

$$GodClass(S) = S' \mid S' \subseteq S, \forall C \in S'$$
$$(NCPA > 1) \wedge (RFC > 9) \wedge (COH < 0.25)$$

กำหนดให้ *C* คือ คลาสที่สงสัยว่ามีข้อบกพร่อง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบ

ในบทนี้จะกล่าวถึงการทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบแต่ละประเภทตามขั้นตอนการวิจัยในบทที่ 3 โดยในแต่ละประเภทจะอธิบายเป็น 3 ขั้นตอนคือ การตรวจจับข้อบกพร่องของโมเดลการออกแบบและการวัดค่ามาตรวัดความสามารถการเปลี่ยนแปลงตามขั้นตอน 3.1.4 และ 3.1.5 ของบทที่ 3, การประยุกต์ใช้วิธีรีแฟคทอริงตามขั้นตอน 3.1.6 ของบทที่ 3, และการประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบและการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงหลังประยุกต์ใช้วิธีรีแฟคทอริงตามขั้นตอน 3.1.7 และ 3.1.8 ของบทที่ 3

แผนภาพคลาสที่แสดงในบทนี้จะแสดงเฉพาะส่วนที่มีข้อบกพร่อง แผนภาพที่สมบูรณ์ของระบบและรายละเอียดของโมเดลการออกแบบตัวอย่างที่นำมาใช้ทดสอบแสดงอยู่ในภาคผนวก ก

5.1 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Data Class

5.1.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ

จากกลยุทธ์การตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Data Class ผู้วิจัยได้นำกลยุทธ์การตรวจจับมาทำการทดสอบตรวจจับข้อบกพร่องของโมเดลการออกแบบ 5 ระบบ ได้ผลการคำนวณมาตรวัดแสดงในภาคผนวก ข ซึ่งจะเห็นว่าพบระบบที่มีค่ามาตรวัดที่บอกถึงข้อบกพร่องดังต่อไปนี้

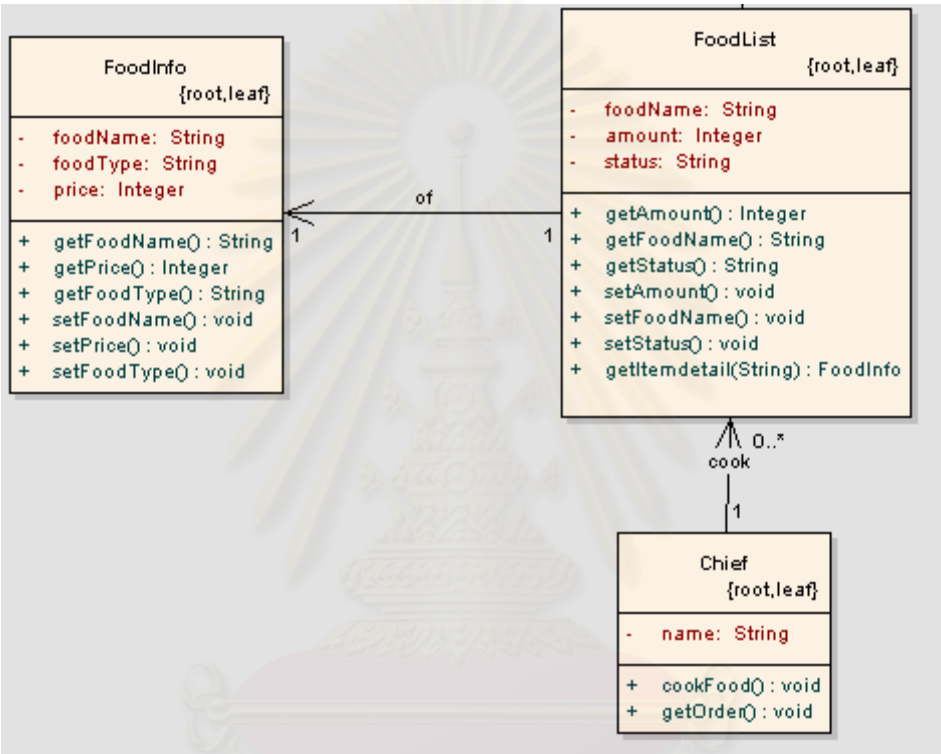
5.1.1.1 คลาส FoodInfo ในระบบตัวอย่างที่ 8 ให้ค่ามาตรวัด NNorm = 0 ซึ่งบอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่อง Data Class และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 1.288
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.992

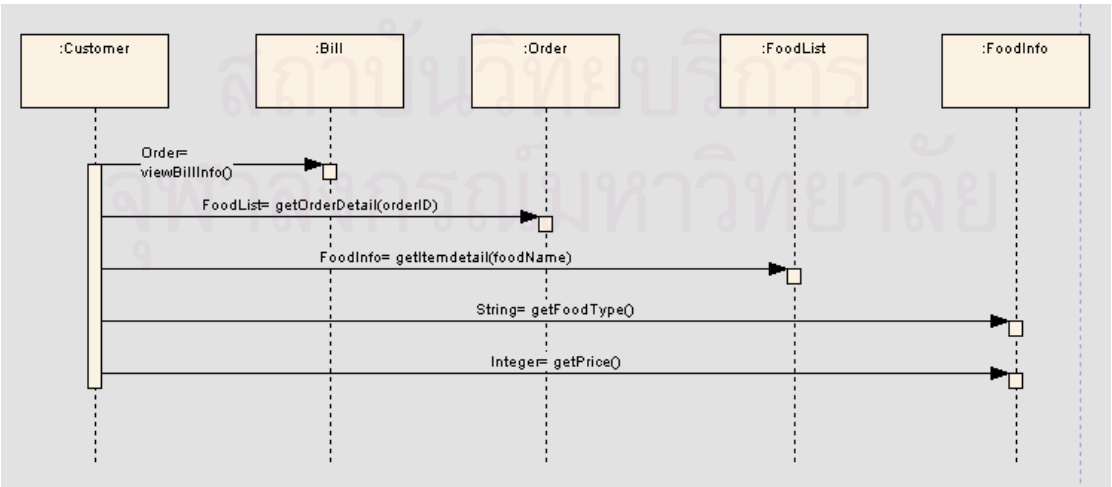
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.888

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.899

รูปที่ 5.1, 5.2 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Data Class ตามลำดับ



รูปที่ 5.1 แผนภาพคลาสของระบบตัวอย่างที่มี Class FoodInfo เป็น Data Class

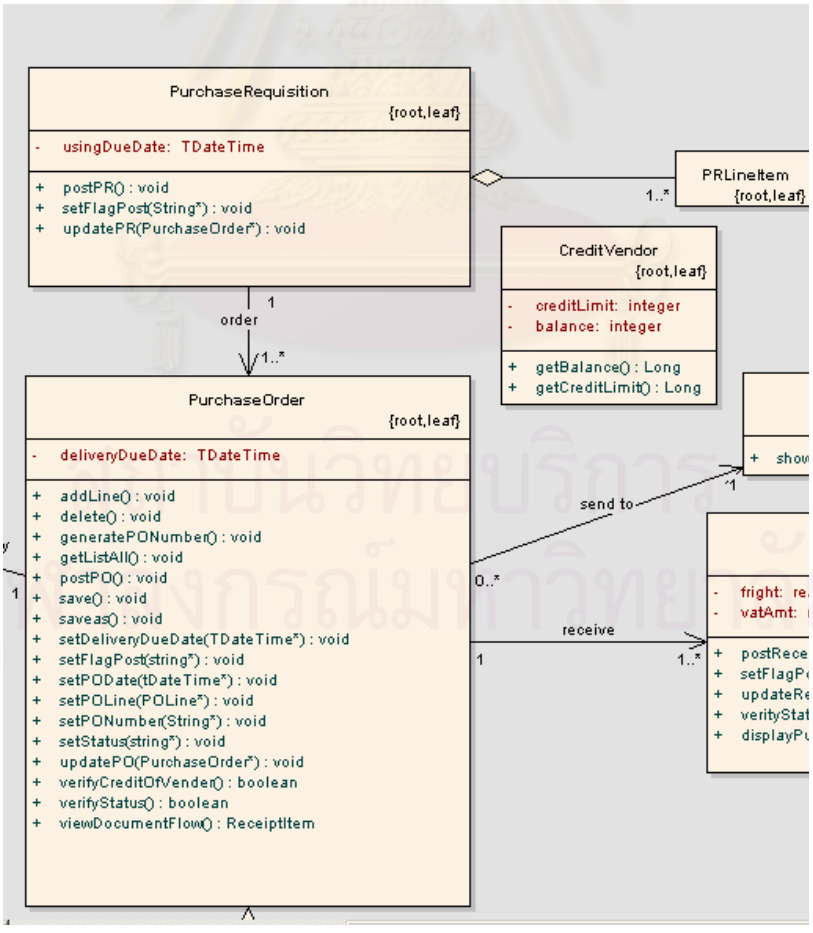


รูปที่ 5.2 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มี Class FoodInfo เป็น Data Class

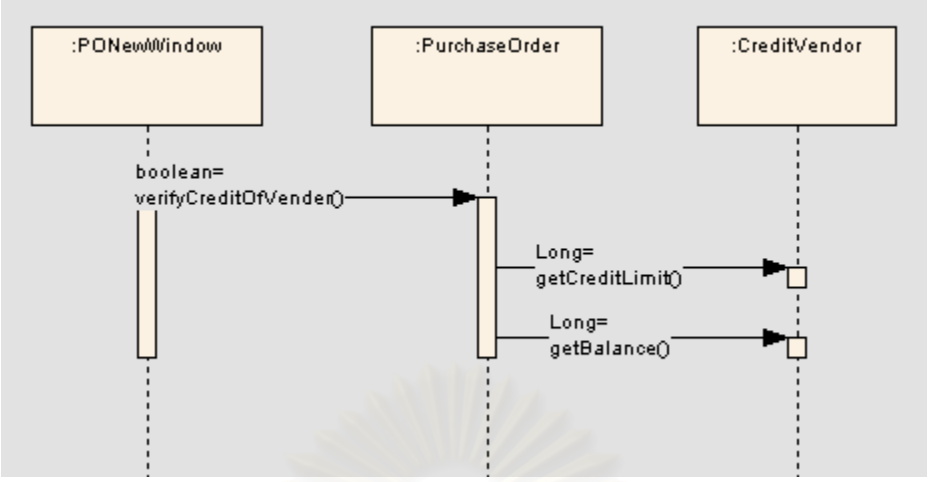
5.1.1.2 คลาส CreditVendor ในระบบตัวอย่างที่ 37 ให้ค่ามาตรวัด NNorm = 0 ซึ่งบอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Data Class และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.832
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.992
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.424
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.872

รูปที่ 5.3, 5.4 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Data Class ตามลำดับ



รูปที่ 5.3 แผนภาพคลาสของระบบตัวอย่างที่มี Class CreditVendor เป็น Data Class

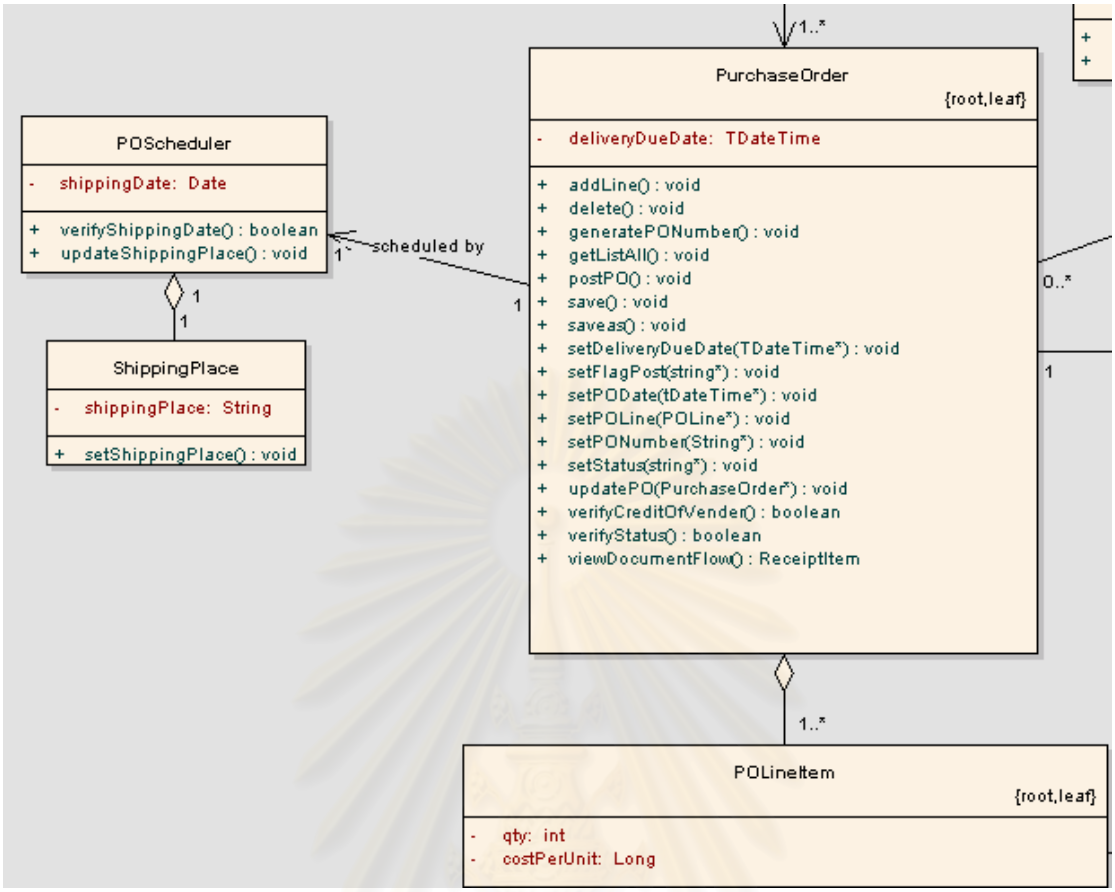


รูปที่ 5.4 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มี Class CreditVendor เป็น Data Class

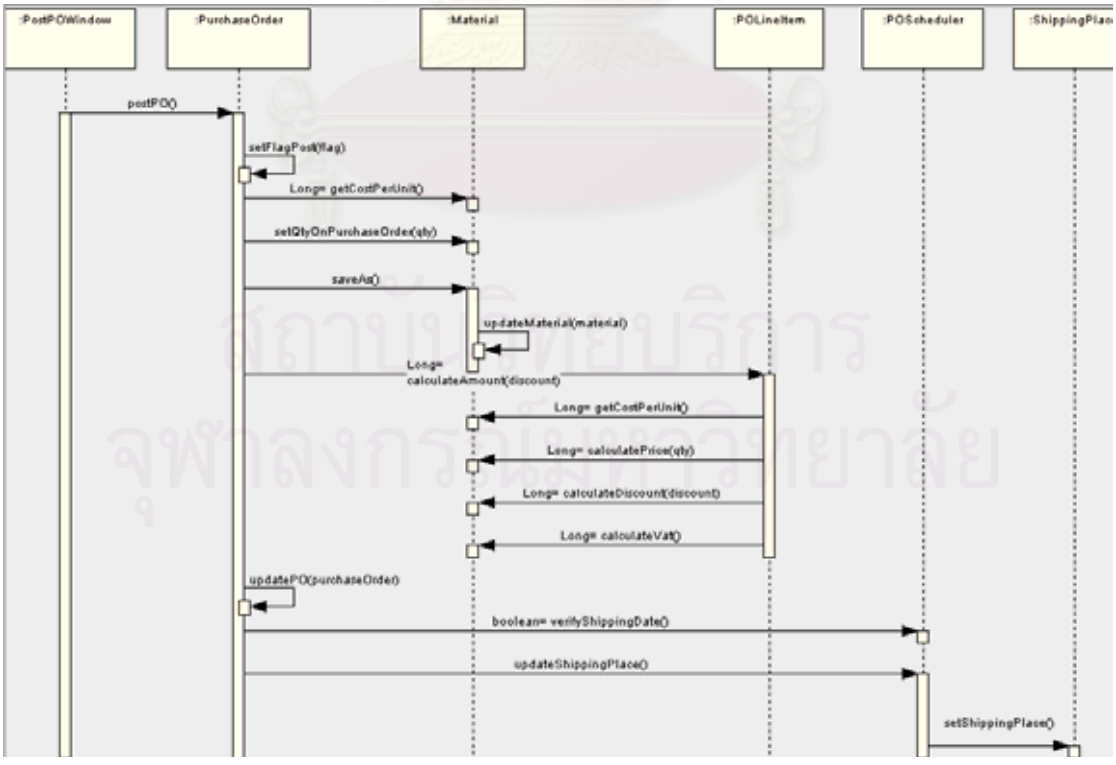
5.1.1.3 คลาส ShippingPlace ในระบบตัวอย่างที่ 37 ให้ค่ามาตรวัด NNorm = 0 ซึ่งบอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Data Class และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.832
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.992
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.424
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.872

รูปที่ 5.5, 5.6 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Data Class ตามลำดับ



รูปที่ 5.5 แผนภาพคลาสของระบบตัวอย่างที่มี Class ShippingPlace เป็น Data Class

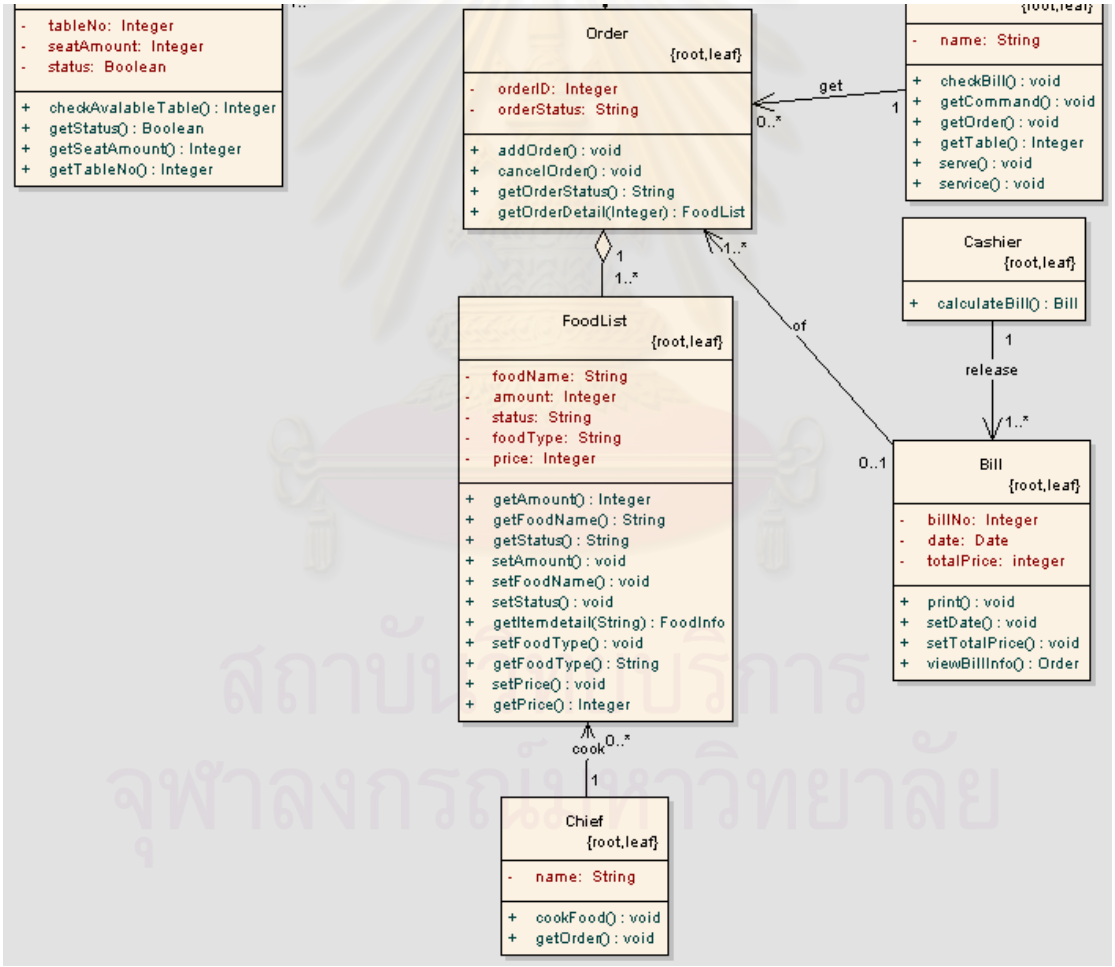


รูปที่ 5.6 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มี Class ShippingPlace เป็น Data Class

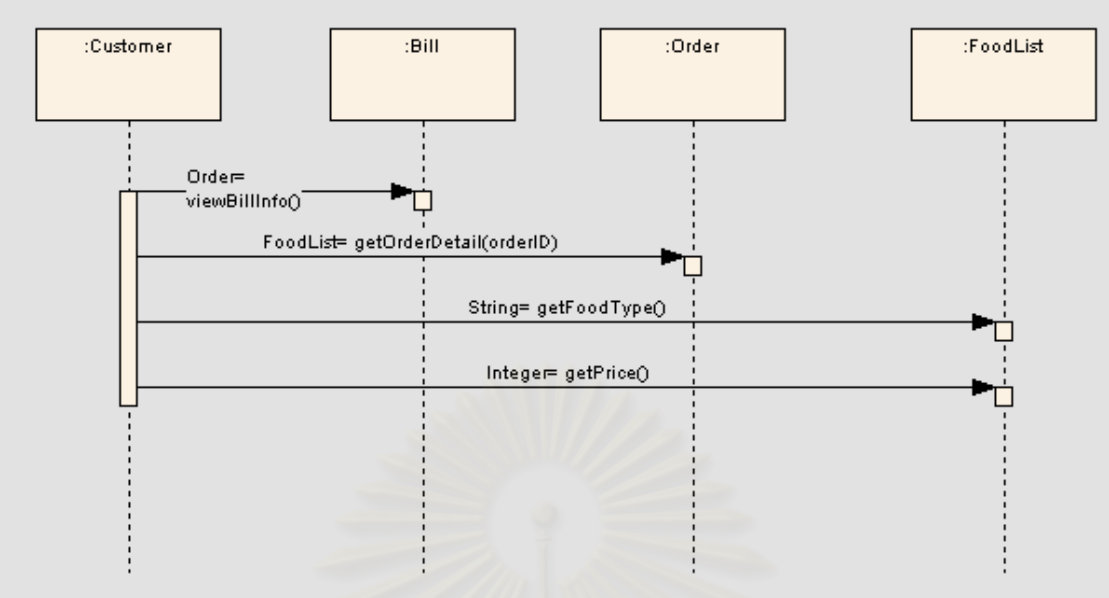
5.1.2 การประยุกต์ใช้วิธีรีแฟคทอริง

วิธีการรีแฟคทอริงที่จะนำมาประยุกต์ใช้จะนำมาจากกลยุทธ์การตรวจจับข้อบกพร่องประเภท Data Class จากบทที่ 3 ซึ่งสำหรับแต่ละคลาสจะมีการใช้วิธีรีแฟคทอริงดังนี้

5.1.2.1 คลาส FoodInfo ในระบบตัวอย่างที่ 8 ประยุกต์ใช้วิธีการ Move Method โดยจากการพิจารณาลักษณะของคลาส FoodInfo และลักษณะการเรียกใช้งานแล้ว จะทำการย้ายเมทธอดทั้งหมดและคุณลักษณะทั้งหมดของคลาส FoodInfo ไปยังคลาส FoodList ซึ่งเป็นผู้เรียกใช้งานคุณสมบัติของคลาส FoodInfo หลังจากนั้นจะลบคลาส FoodInfo ทิ้งเนื่องจากหากเก็บไว้ในระบบอาจจะทำให้เสียค่าบำรุงรักษาโดยไม่จำเป็น รูปที่ 5.7, 5.8 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



รูปที่ 5.7 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส FoodInfo

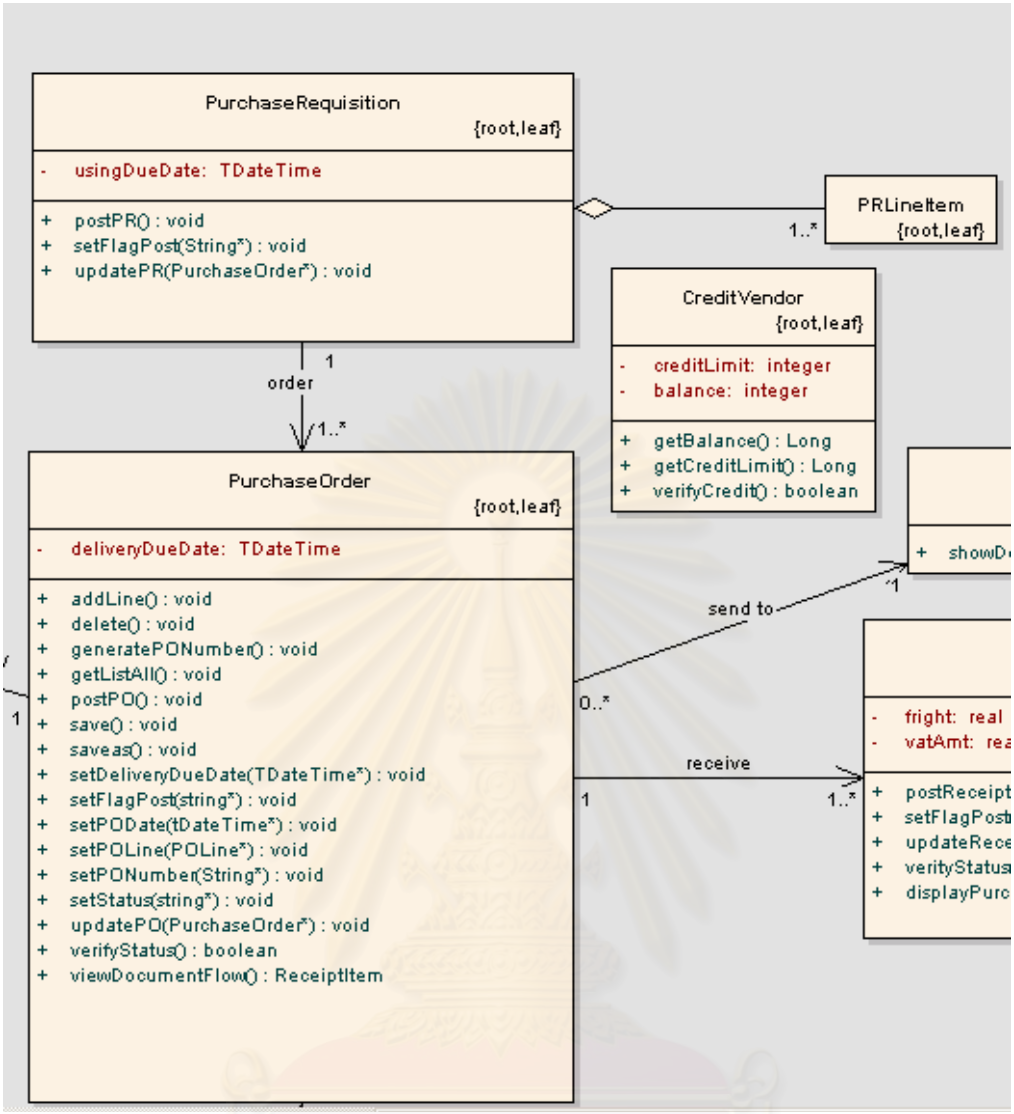


รูปที่ 5.8 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส FoodInfo

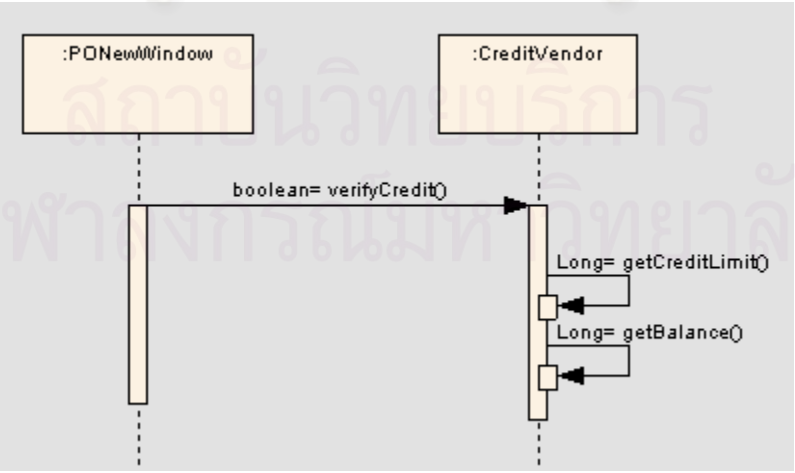
5.1.2.2 คลาส CreditVendor ในระบบตัวอย่างที่ 37 ประยุกต์ใช้วิธีการ Move Method กับเมทธอด verifyCreditOfVendor ของคลาส PurchaseOrder ไปเป็นเมทธอดของคลาส CreditVendor และ Rename Method เป็นชื่อ verifyCredit เพื่อให้ง่ายต่อความเข้าใจ หลังจากนั้นที่แผนภาพซีเควนซ์จะเปลี่ยนการเรียกใช้งานเมทธอด verifyCreditOfVendor ให้ไปเรียกใช้งานเมทธอด verifyCredit ที่ออบเจ็กต์ไลฟ์ไลน์ของคลาส CreditVendor แทน

รูปที่ 5.9, 5.10 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



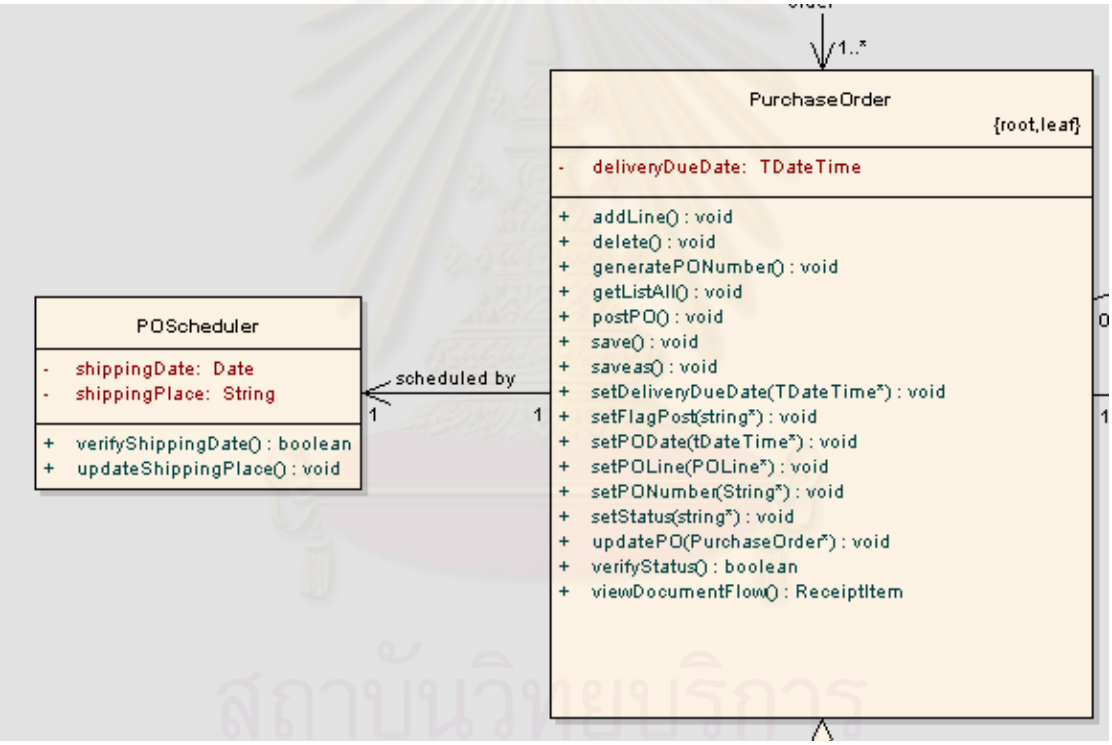
รูปที่ 5.9 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส CreditVendor



รูปที่ 5.10 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส CreditVendor

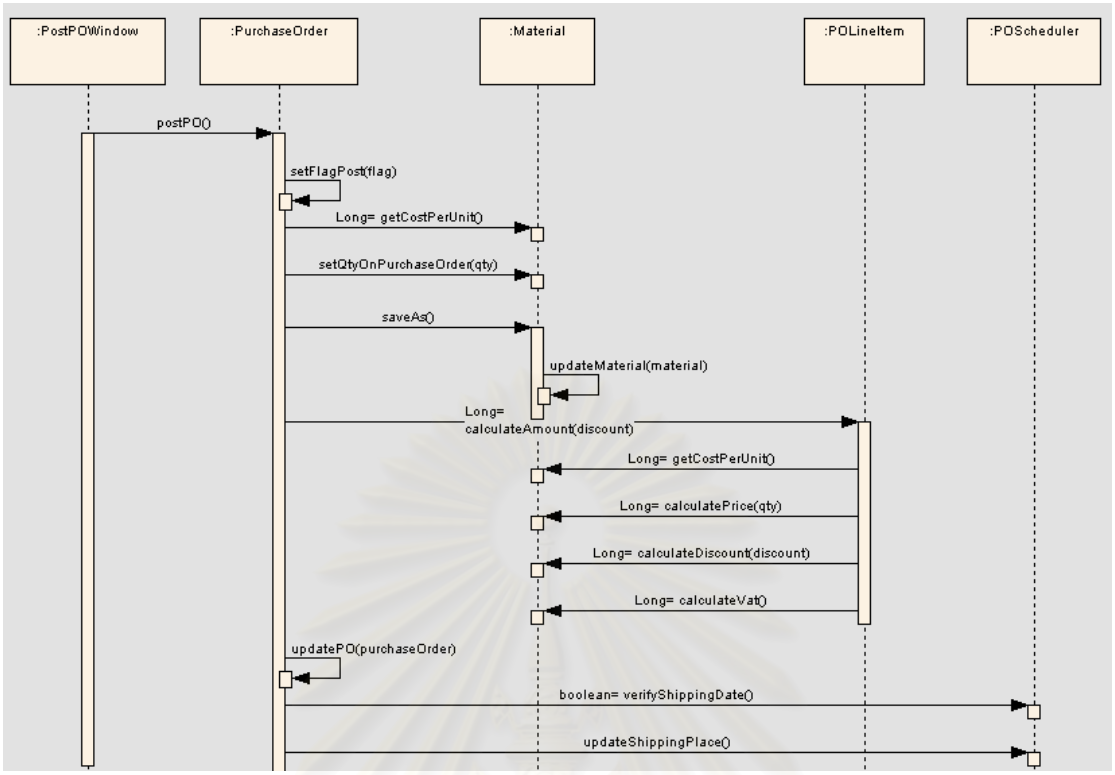
5.1.2.3 คลาส ShippingPlace ในระบบตัวอย่างที่ 37 วิธีรีแฟคทอริงที่เลือกใช้คือการ Move Method แต่เนื่องจากกรณีนี้เมทธอด updateShippingPlace มาเรียกใช้งานเมทธอด setShippingPlace ของคลาส ShippingPlace โดยที่เมทธอด setShippingPlaceเป็นเพียงเมทธอดขนาดเล็ก ผู้วิจัยจึงจะนำวิธีการ Inline Method มาประยุกต์ใช้ด้วย ซึ่งเป็นการทำลายเมทธอด setShippingPlace แล้วย้ายคุณลักษณะ shippingPlace ของคลาส ShippingPlace ไปยังคลาส POScheduler ซึ่งก็คือการย้ายเนื้อหาของเมทธอด setShippingPlace ไปเพิ่มเป็นเนื้อหาของเมทธอด updateShippingPlace ด้วย เนื่องจากคลาส ShippingPlace เป็นคลาสที่ไม่จำเป็นในระบบ หลังจากนั้นจะลบคลาส ShippingPlace ออกจากระบบ

รูปที่ 5.11, 5.12 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



รูปที่ 5.11 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส ShippingPlace

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.12 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส ShippingPlace

5.1.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง

หลังจากประยุกต์ใช้วิธีการรีแฟคทอริงเรียบร้อยแล้ว ผู้วิจัยได้ทำการวัดค่าของมาตรวัดสำหรับตรวจจับข้อบกพร่องที่คลาสที่มีลักษณะเป็น Data Class อีกครั้งหนึ่ง ได้ผลการวัดค่ามาตรวัดเป็นดังนี้

คลาส FoodInfo ในระบบตัวอย่างที่ 8 ไม่เป็น Data Class และไม่อยู่ในระบบแล้ว เนื่องจากถูกย้ายเมทธอดและคุณลักษณะไปไว้ที่คลาส FoodList

หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริง เป็นดังตาราง 6.1 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าที่ดีขึ้น สำหรับโมเดลความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA

ตาราง 5.1 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 8

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	1.288	0.992	0.888	0.899
หลัง	1.288	0.992	0.888	0.908

คลาส CreditVendor ในระบบตัวอย่างที่ 37 มีค่ามาตรวัด NNorm = 1 และไม่เป็น Data Class แล้ว

คลาส ShippingPlace ในระบบตัวอย่างที่ 37 ไม่เป็น Data Class และไม่อยู่ในระบบแล้ว เนื่องจากถูกย้ายเมทธอดและคุณลักษณะไปไว้ที่คลาส POScheduler

หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 6.2 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าที่ดีขึ้น สำหรับโมเดลความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH และ โมเดลความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH

ตาราง 5.2 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	0.832	0.992	0.424	0.872
หลัง	0.946	0.992	0.54	0.872

5.2 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Feature Envy

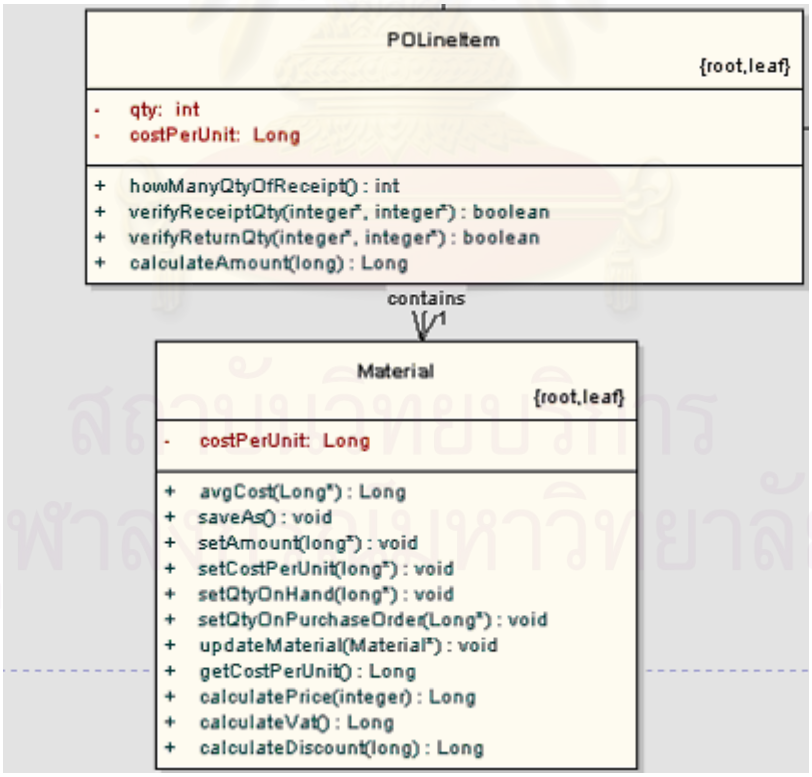
5.2.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ

จากกลยุทธ์การตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Feature Envy ผู้วิจัยได้นำกลยุทธ์การตรวจจับมาทำการทดสอบตรวจจับข้อบกพร่องของโมเดลการออกแบบ 5 ระบบ ได้ผลการคำนวณมาตรวัดแสดงในภาคผนวก ข ซึ่งจะเห็นว่าพบระบบที่มีค่ามาตรวัดที่บอกถึงข้อบกพร่องดังต่อไปนี้

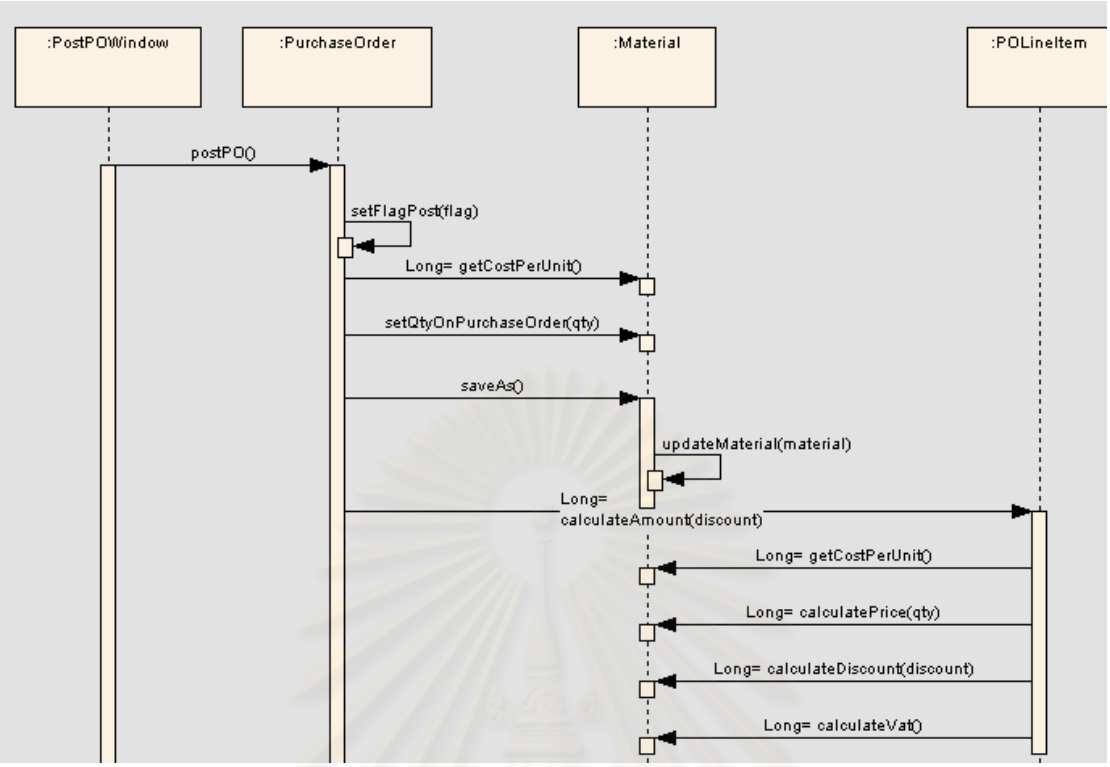
5.2.1.1 เมทธอด calculateAmount ของคลาส POLineItem ในระบบตัวอย่างที่ 37 ให้ค่ามาตรวัด NSMsg / NMsg = 0.25 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Feature Envy และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.832
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.992
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.424
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.872

รูปที่ 5.13, 5.14 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Feature Envy ตามลำดับ



รูปที่ 5.13 แผนภาพคลาสของระบบตัวอย่างที่มีเมทธอด calculateAmount ของคลาส POLineItem เป็น Feature Envy

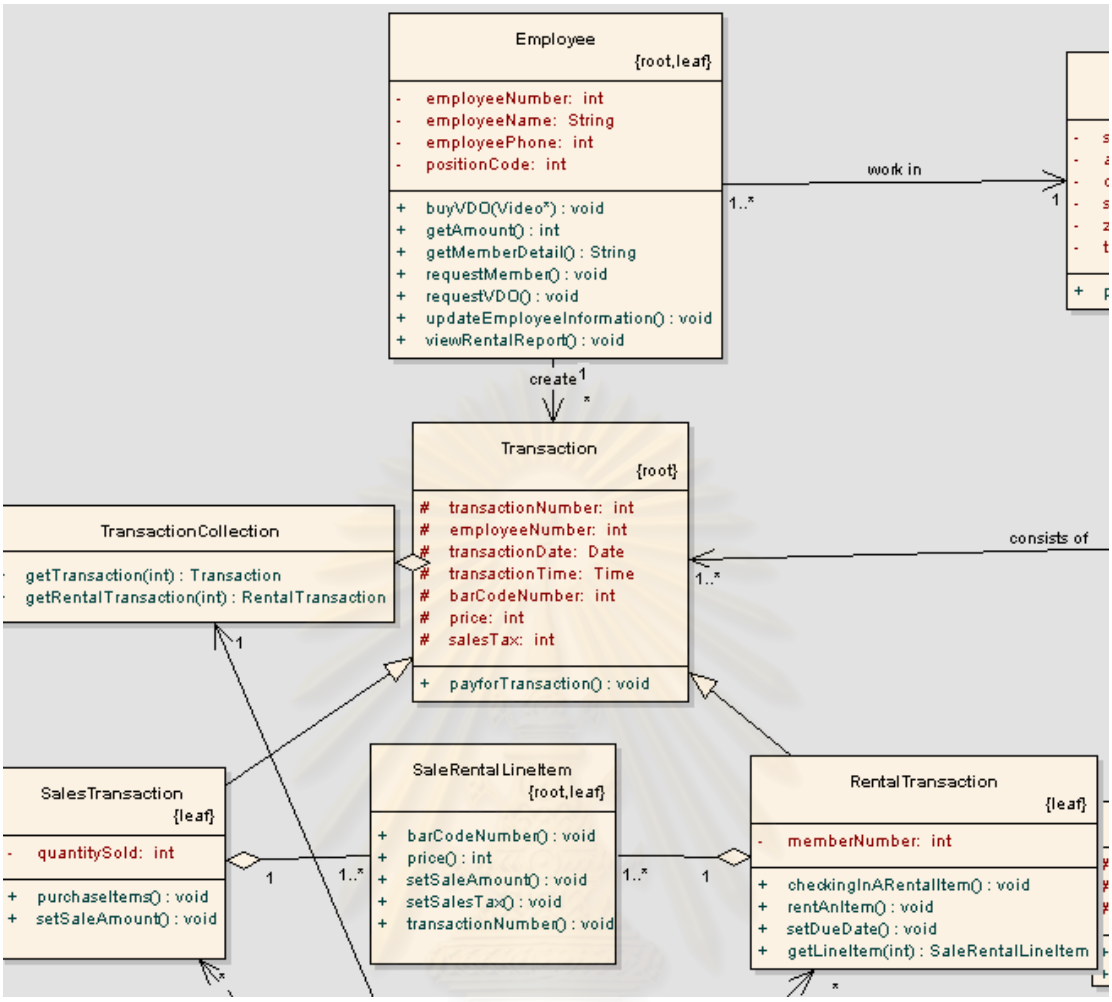


รูปที่ 5.14 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีเมทธอด calculateAmount ของคลาส POLineItem เป็น Feature Envy

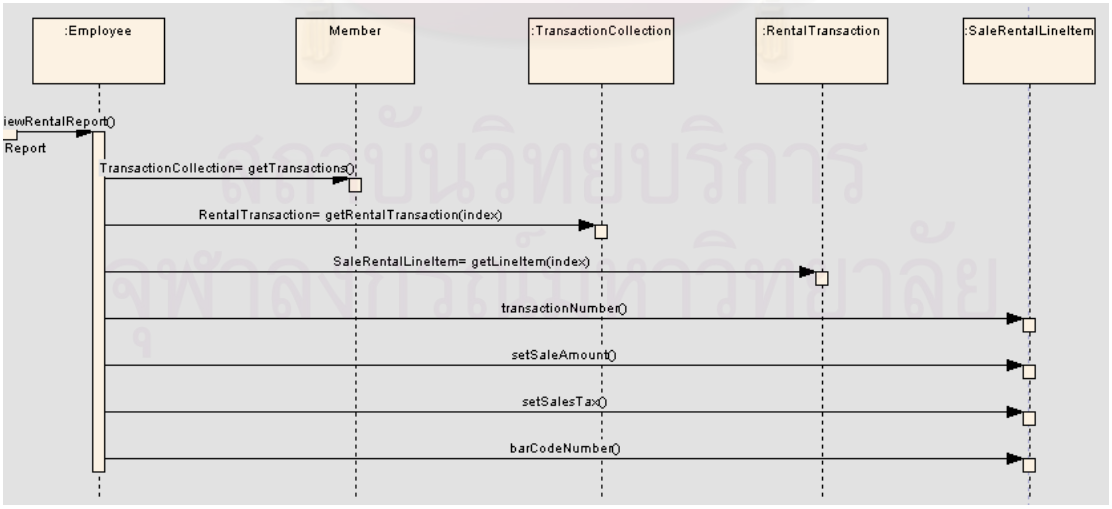
5.2.1.2 เมทธอด viewRentalReport ของคลาส Employee ในระบบตัวอย่างที่ 38 ให้ค่ามาตรวัด NSMsg / NMsg = 0.25 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Feature Envy และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.946
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.912
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.54
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.566

รูปที่ 5.15, 5.16 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Feature Envy ตามลำดับ



รูปที่ 5.15 แผนภาพคลาสของระบบตัวอย่างที่มีเมทธอด viewRentalReport ของคลาส Employee เป็น Feature Envy

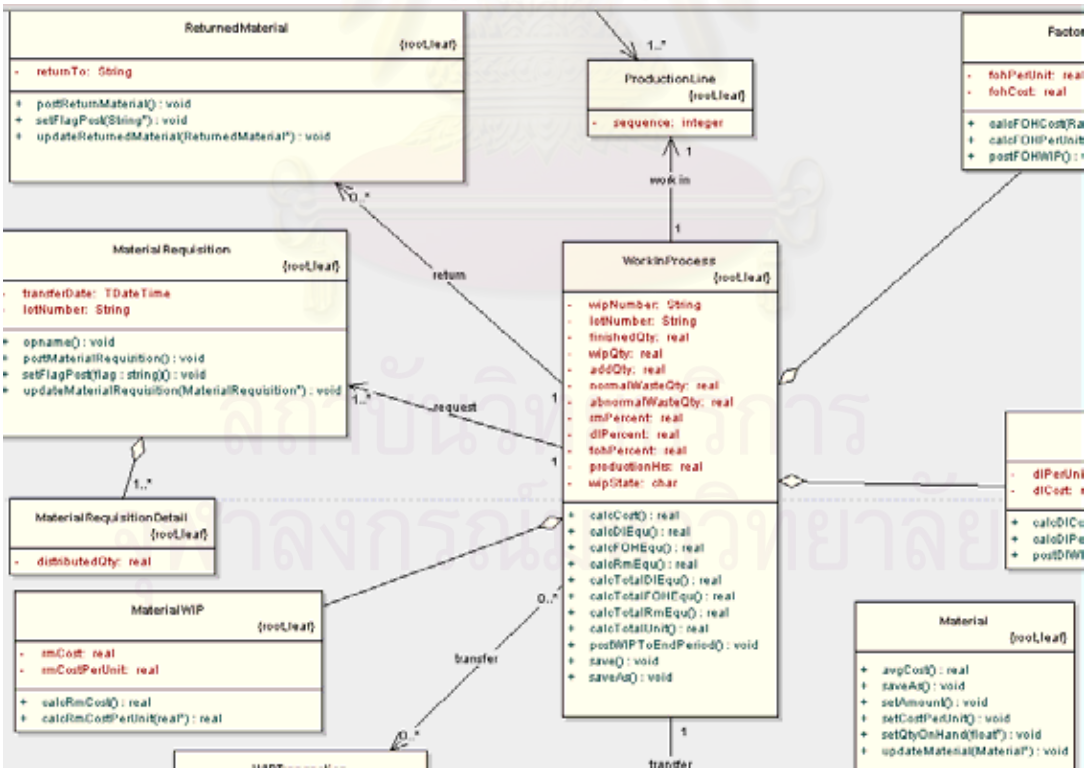


รูปที่ 5.16 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีเมทธอด viewRentalReport ของคลาส Employee เป็น Feature Envy

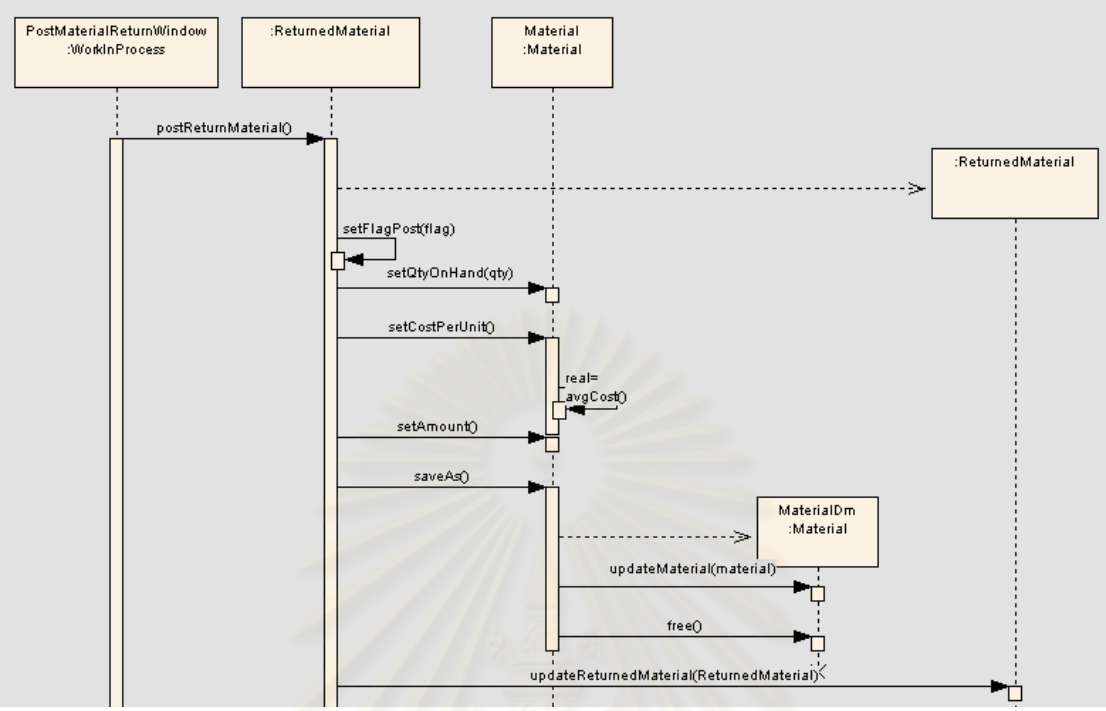
5.2.1.3 เมทธอด postReturnMaterial ของคลาส ReturnedMaterial ในระบบตัวอย่างที่ 39 ให้ค่ามาตรวัด NSMsg / NMsg = 0.25 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Feature Envy และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.946
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.952
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.54
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.611

รูปที่ 5.17, 5.18 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Feature Envy ตามลำดับ



รูปที่ 5.17 แผนภาพคลาสของระบบตัวอย่างที่มีเมทธอด postReturnMaterial ของคลาส ReturnedMaterial เป็น Feature Envy

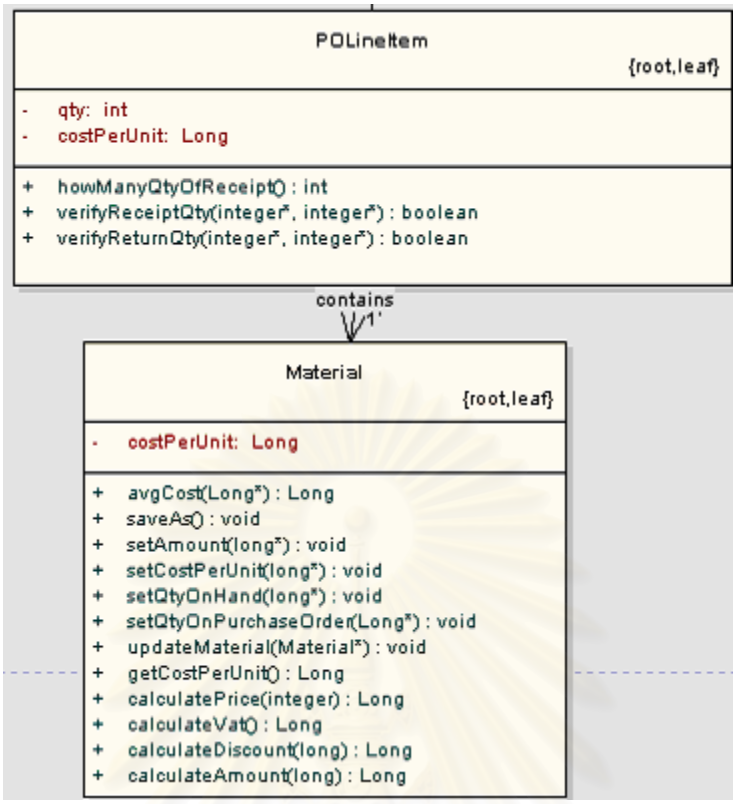


รูปที่ 5.18 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีเมทธอด postReturnMaterial ของคลาส ReturnedMaterial เป็น Feature Envy

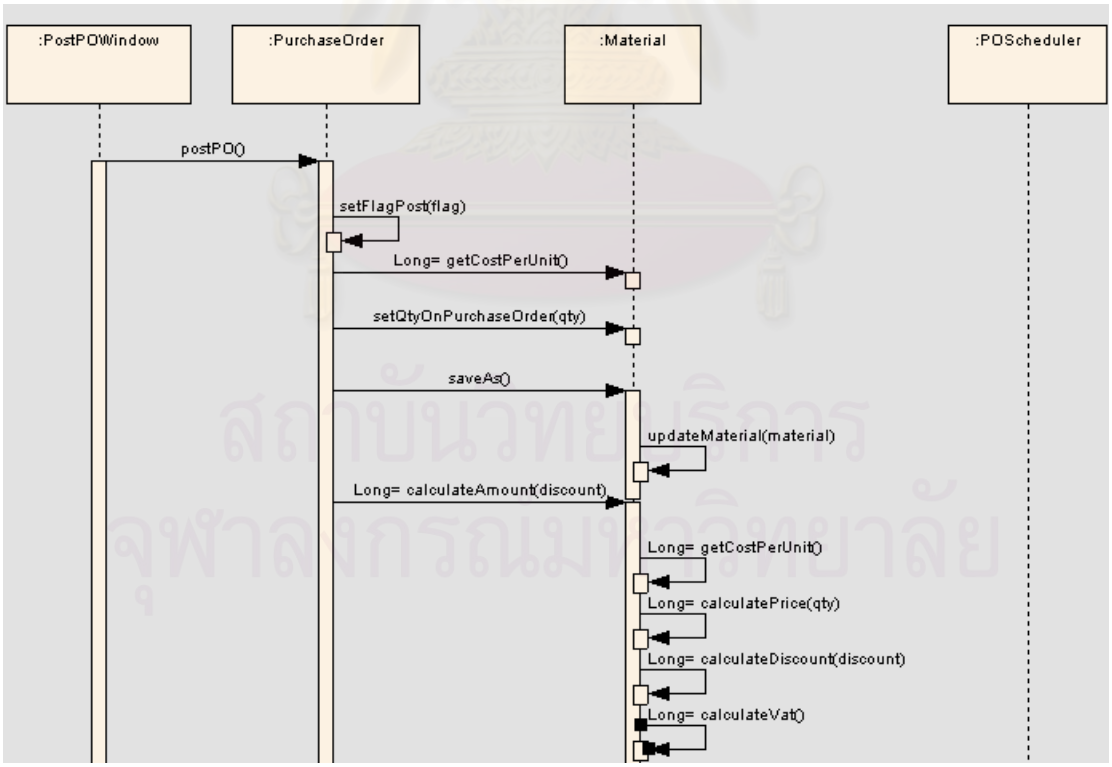
5.2.2 การประยุกต์ใช้วิธีรีแฟคทอริง

วิธีการรีแฟคทอริงที่จะนำมาประยุกต์ใช้จะนำมาจากกลยุทธ์การตรวจจับข้อบกพร่องประเภท Feature Envy จากบทที่ 3 ซึ่งสำหรับแต่ละเมทธอดจะมีการใช้วิธีรีแฟคทอริงดังนี้

5.2.2.1 เมทธอด calculateAmount ของคลาส POLineItem ในระบบตัวอย่างที่ 37 ประยุกต์ใช้วิธีการ Move Method โดยจากการพิจารณาลักษณะการเรียกใช้งานเมทธอดของคลาส Material โดยเมทธอด calculateAmount ของคลาส POLineItem แล้วจะเห็นได้ว่าควรย้ายเมทธอด calculateAmount ของคลาส POLineItem ไปยังคลาส Material รูปที่ 5.19, 5.20 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง

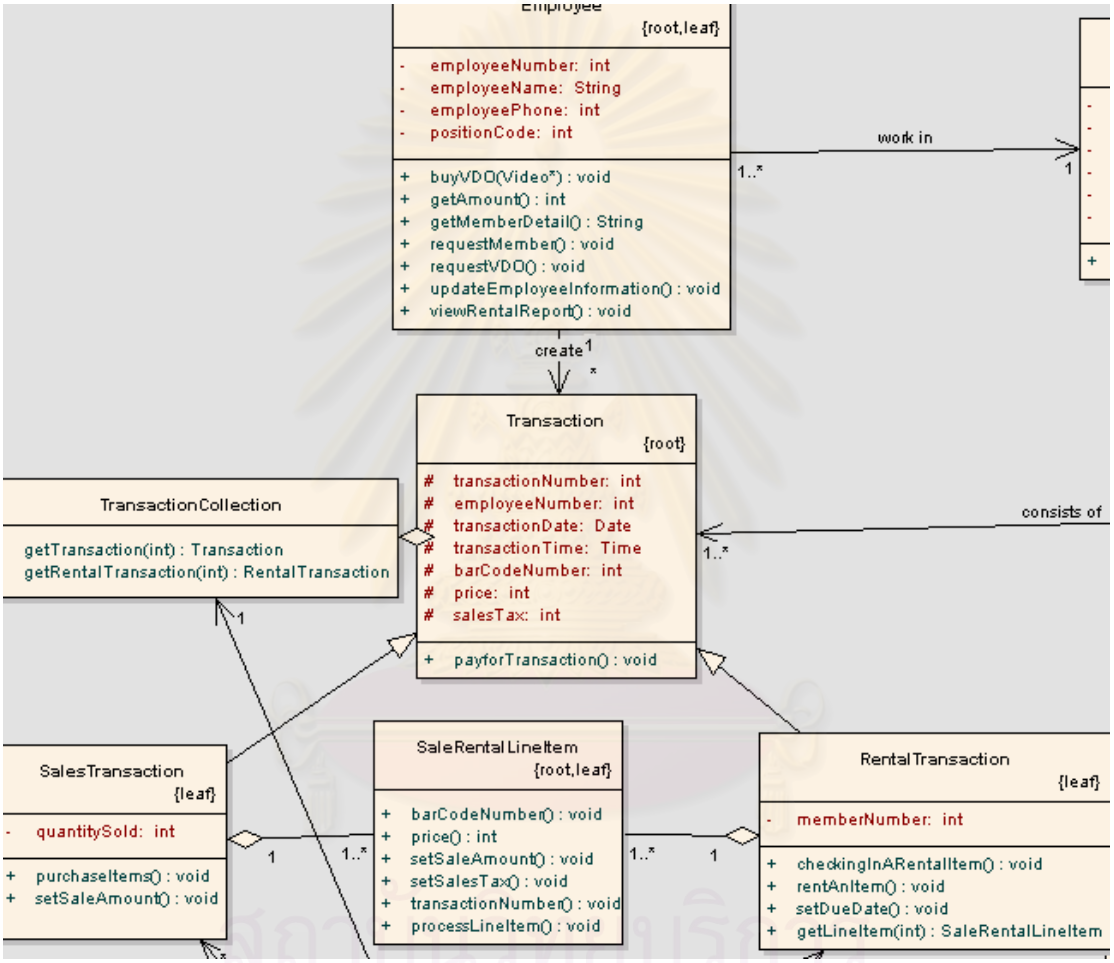


รูปที่ 5.19 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด calculateAmount



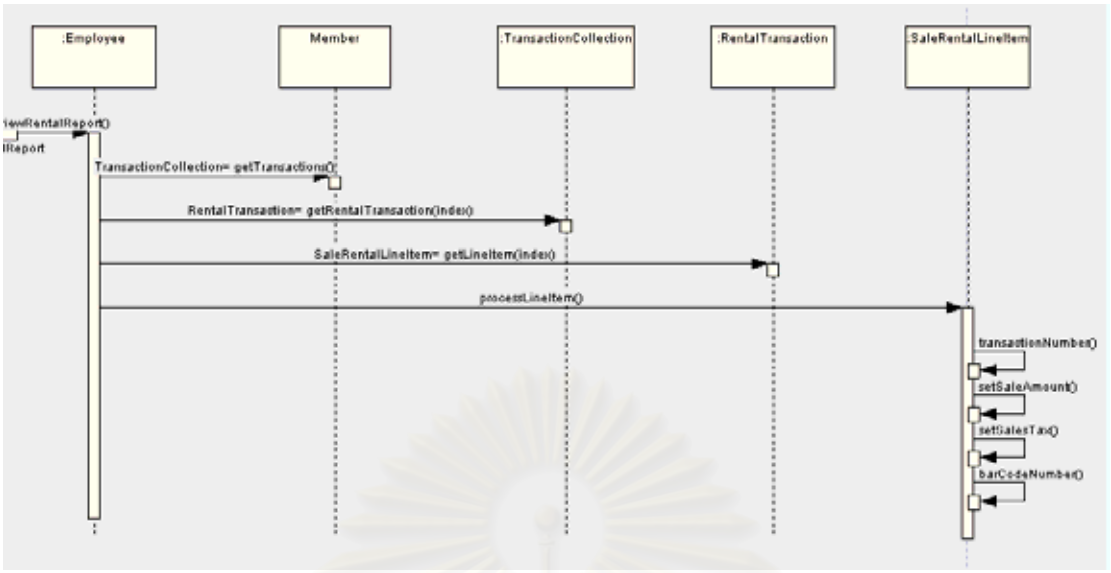
รูปที่ 5.20 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด calculateAmount

5.2.2.2 เมทธอด viewRentalReport ของคลาส Employee ในระบบตัวอย่างที่ 38 ประยุกต์ใช้วิธีการ Extract Method viewRentalReport เป็นอีกเมทธอดคือ เมทธอดชื่อ processSalesLineItem แล้ว ใช้วิธี Move Method เพื่อย้ายเมทธอด processSalesLineItem ไปที่คลาส SalesRentalLineItem รูปที่ 5.21, 5.22 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



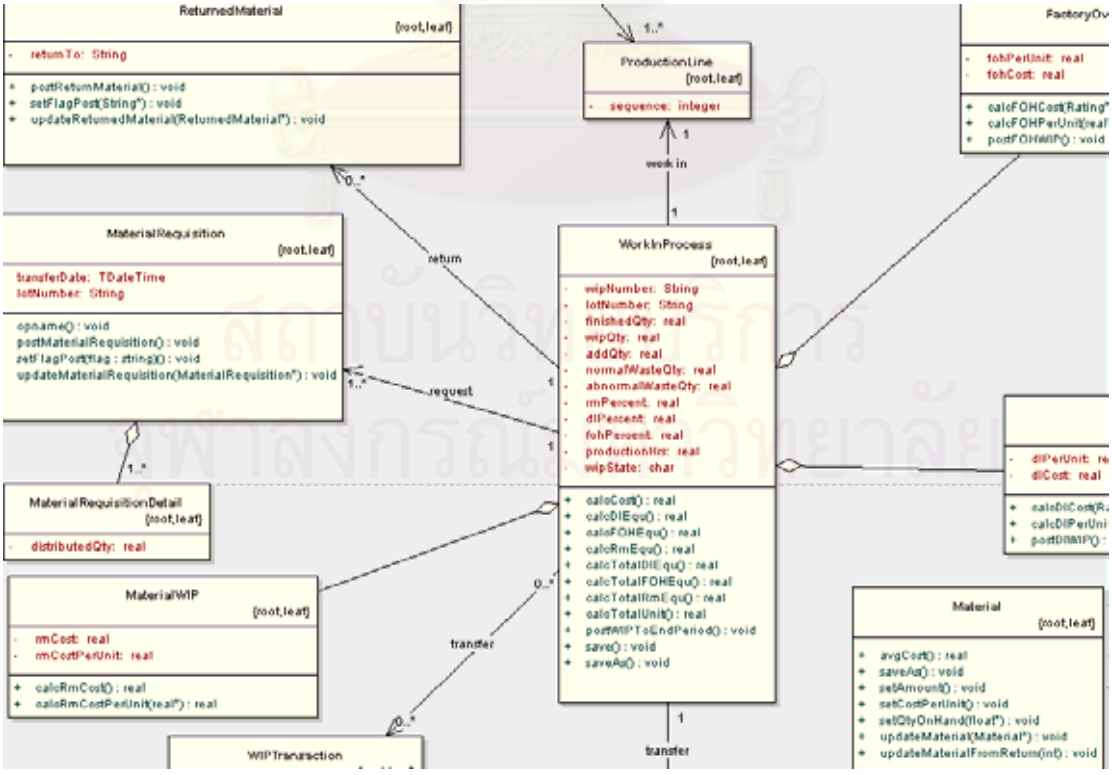
รูปที่ 5.21 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด viewRentalReport

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

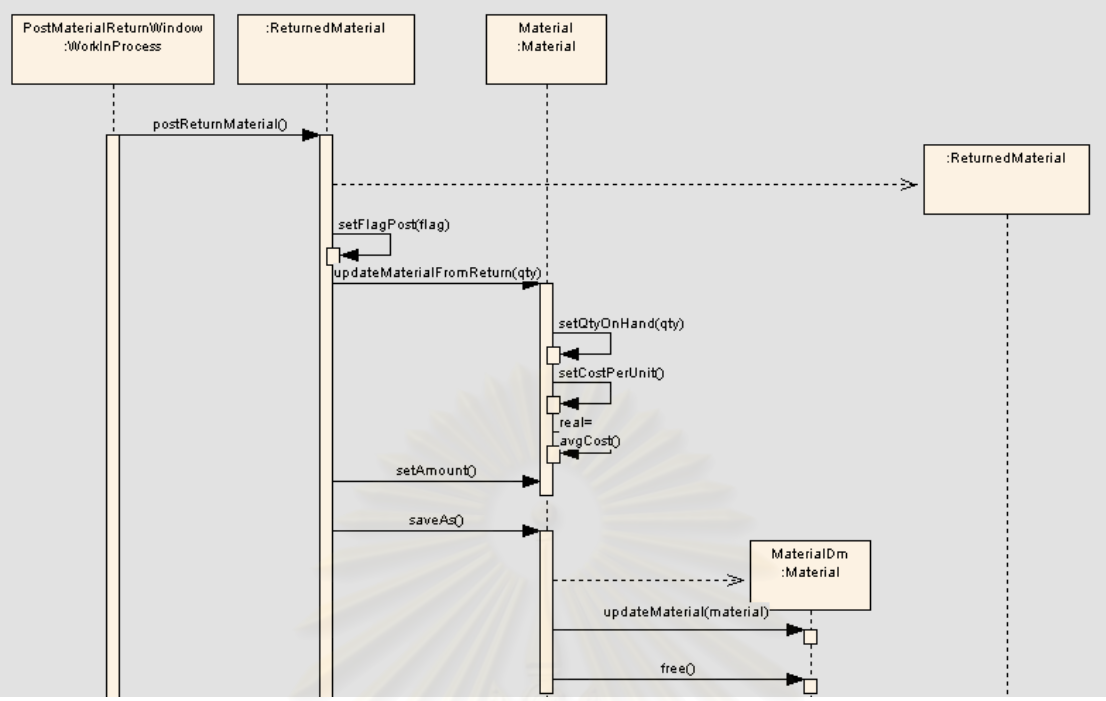


รูปที่ 5.22 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด viewRentalReport

5.2.2.3 เมทธอด postReturnMaterial ของคลาส Returned Material ในระบบตัวอย่างที่ 39 ประยุกต์ใช้วิธีการ Extract Method postReturnMaterial เป็นอีกเมทธอด แล้วย้ายไปเป็นเมทธอดชื่อ updateMaterialFromReturn ที่คลาส Material รูปที่ 5.23, 5.24 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



รูปที่ 5.23 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด postReturnMaterial



รูปที่ 5.24 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด postReturnMaterial

5.2.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง

หลังจากประยุกต์ใช้วิธีการรีแฟคทอริงเรียบร้อยแล้ว ผู้วิจัยได้ทำการวัดค่าของมาตรวัดสำหรับตรวจจับข้อบกพร่องที่เมทธอดที่มีลักษณะเป็น Feature Envy อีกครั้งหนึ่ง ได้ผลการวัดค่ามาตรวัดเป็นดังนี้

เมทธอด calculateAmount ของคลาส POLineItem ในระบบตัวอย่างที่ 37 ซึ่งถูกย้ายไปยังคลาส Material ไม่เป็น Feature Envy แล้ว และค่าของมาตรวัด NSMsg / NMsg เป็นค่าเข้าสู่ค่าอนันต์ ซึ่งในที่นี้หมายถึงจำนวนเมสเสจที่เรียกใช้งานเมทธอดของคลาสอื่นมีค่าเป็น 0 (NMsg มีค่าเป็น 0)

หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.3 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าไม่เปลี่ยนแปลง

ตาราง 5.3 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	0.832	0.992	0.424	0.872
หลัง	0.832	0.992	0.424	0.872

เมทธอด viewRentalReport ของคลาส Employee ในระบบตัวอย่างที่ 38 ซึ่งถูก Extract ออกและย้ายไปยังคลาส SalesRentalLineItem ไม่เป็น Feature Envy แล้ว โดยค่าของมาตรวัด NSMsg / NMsg สำหรับเมทธอด viewRentalReport ของคลาส Employee เป็น 1 ซึ่งไม่อยู่ในช่วงของข้อบกพร่อง และค่าของมาตรวัด NSMsg / NMsg สำหรับเมทธอด processSalesLineItem ที่คลาส SalesRentalLineItem เป็นค่าเข้าสู่ค่าอนันต์ ซึ่งในที่นี้หมายถึงจำนวนเมสเสจที่เรียกใช้งานเมทธอดของคลาสอื่นมีค่าเป็น 0 (NMsg มีค่าเป็น 0)

หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.4 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าไม่เปลี่ยนแปลง

ตาราง 5.4 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 38

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	0.946	0.912	0.54	0.566
หลัง	0.946	0.912	0.54	0.566

เมทธอด postReturnMaterial ของคลาส Returned Material ในระบบตัวอย่างที่ 39 ซึ่งถูก Extract ออกและย้ายไปยังคลาส Material ไม่เป็น Feature Envy แล้ว โดยค่าของมาตรวัด NSMsg / NMsg สำหรับเมทธอด postReturnMaterial ของคลาส Returned Material เป็น 0.33 ซึ่งไม่อยู่ในช่วงของข้อบกพร่อง และค่าของมาตรวัด NSMsg / NMsg สำหรับเมทธอด updateMaterialFromReturn ที่คลาส Material เป็นค่าเข้าสู่ค่าอนันต์ ซึ่งในที่นี้หมายถึงจำนวนเมสเสจที่เรียกใช้งานเมทธอดของคลาสอื่นมีค่าเป็น 0 (NMsg มีค่าเป็น 0)

หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.5 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าไม่เปลี่ยนแปลง

ตาราง 5.5 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 39

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	0.946	0.952	0.54	0.611
หลัง	0.946	0.952	0.54	0.611

5.3 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Message Chains

5.3.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ

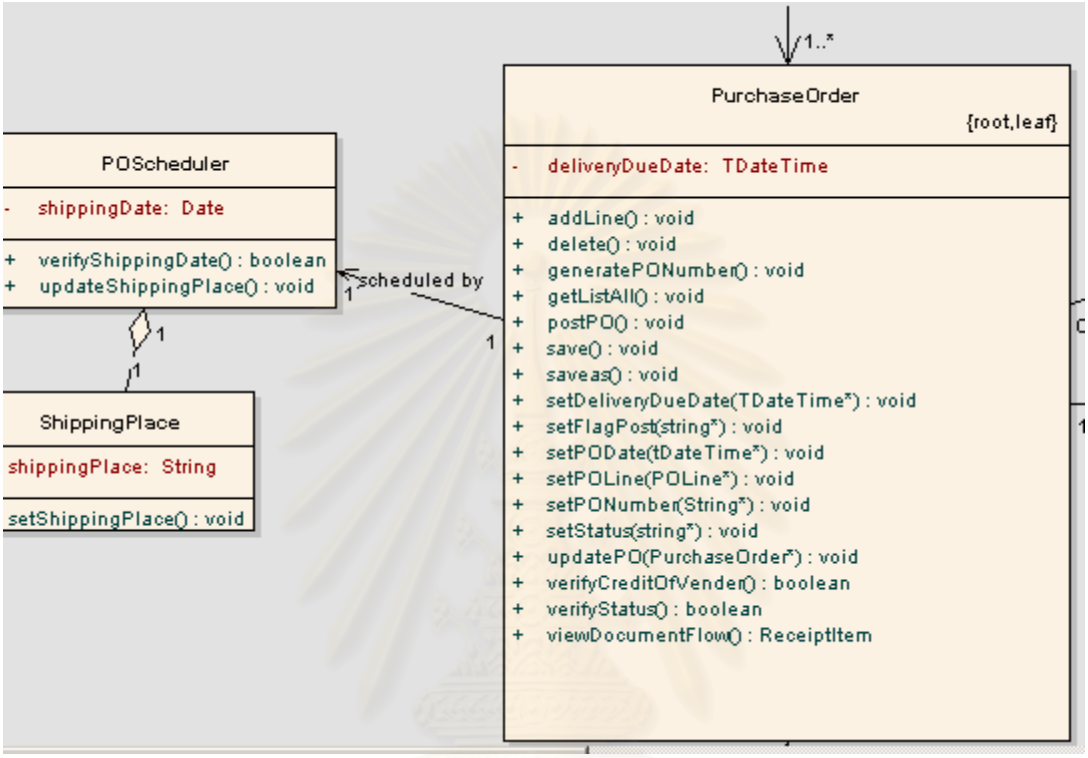
จากกลยุทธ์การตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Message Chains ผู้วิจัยได้นำกลยุทธ์การตรวจจับมาทำการทดสอบตรวจจับข้อบกพร่องของโมเดลการออกแบบ 5 ระบบ ได้ผลการคำนวณมาตรวัดแสดงในภาคผนวก ข ซึ่งจะเห็นว่าพบระบบที่มีค่ามาตรวัดที่บอกถึงข้อบกพร่องดังต่อไปนี้

5.3.1.1 เมทธอด viewDocumentFlow ของคลาส PurchaseOrder ในระบบตัวอย่างที่ 37 ให้ค่ามาตรวัด SOMsgCh = 2 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Message Chains และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

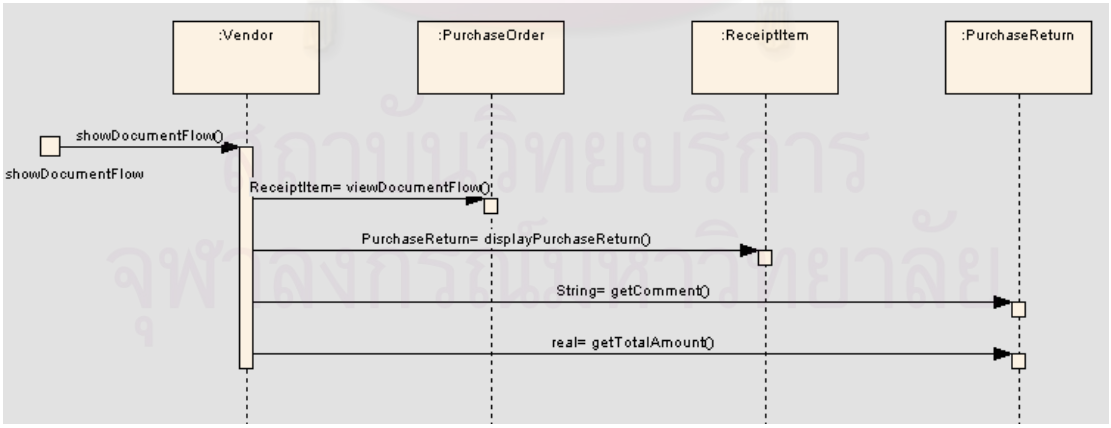
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.832
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.992
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.424

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.872

รูปที่ 5.25, 5.26 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Message Chains ตามลำดับ



รูปที่ 5.25 แผนภาพคลาสของระบบตัวอย่างที่เมทธอด viewDocumentFlow ของคลาส PurchaseOrder มีข้อบกพร่องประเภท Message Chains



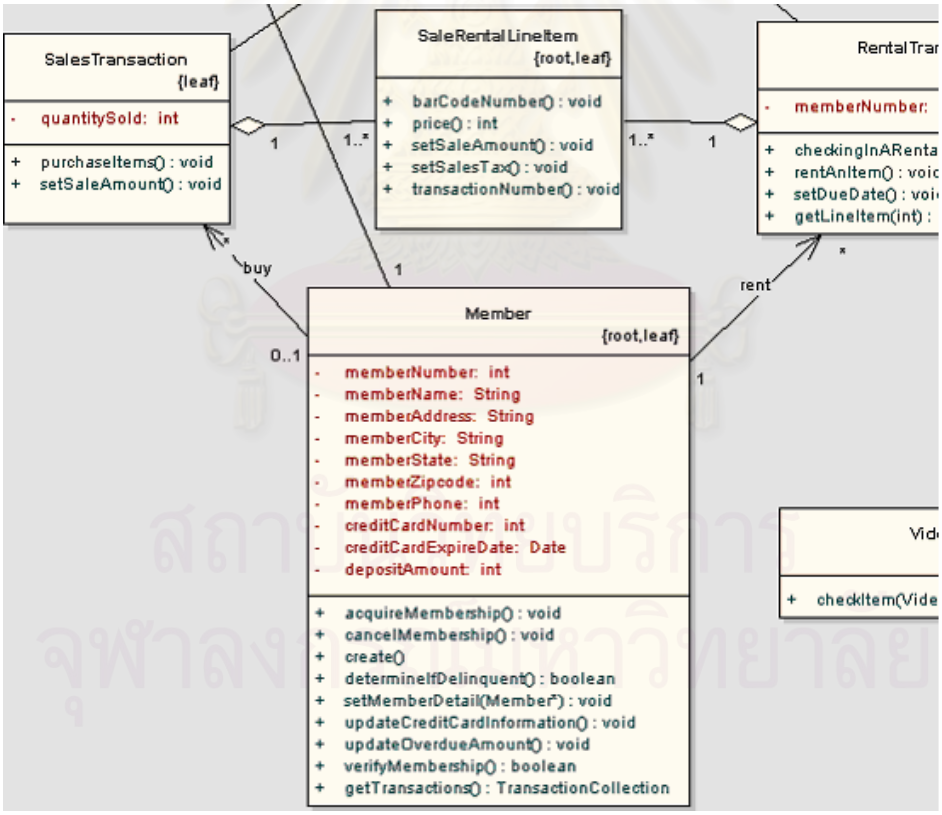
รูปที่ 5.26 แผนภาพซีเควนซ์ของระบบตัวอย่างที่เมทธอด viewDocumentFlow ของคลาส PurchaseOrder มีข้อบกพร่องประเภท Message Chains

5.3.1.2 เมทธอด getTransactions ของคลาส Member ในระบบตัวอย่างที่ 38 ให้ค่ามาตรวัด SOMsgCh = 3 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่อง

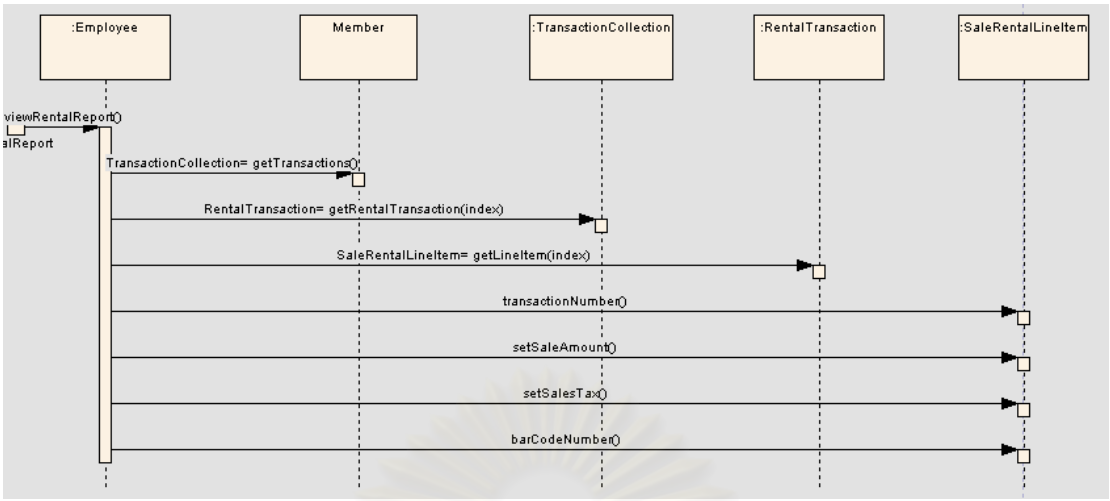
แบบ Message Chains และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.946
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.912
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.54
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.566

รูปที่ 5.27, 5.28 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบที่มีข้อบกพร่องประเภท Message Chains



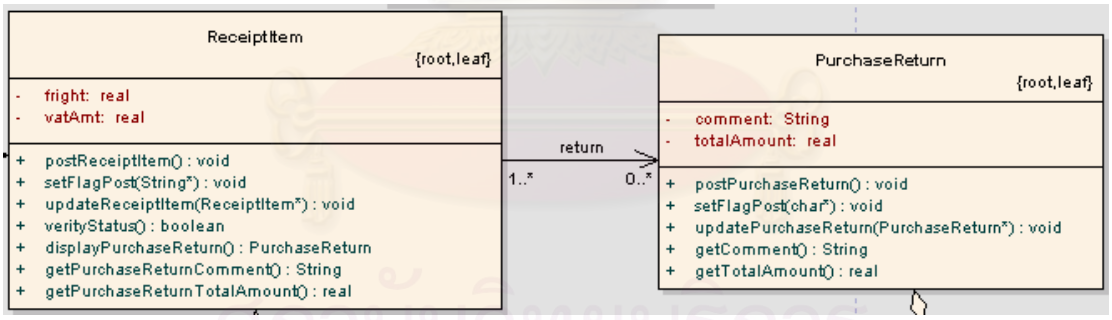
รูปที่ 5.27 แผนภาพคลาสของระบบตัวอย่างที่เมทธอด getTransactions ของคลาส Member มีข้อบกพร่องประเภท Message Chains



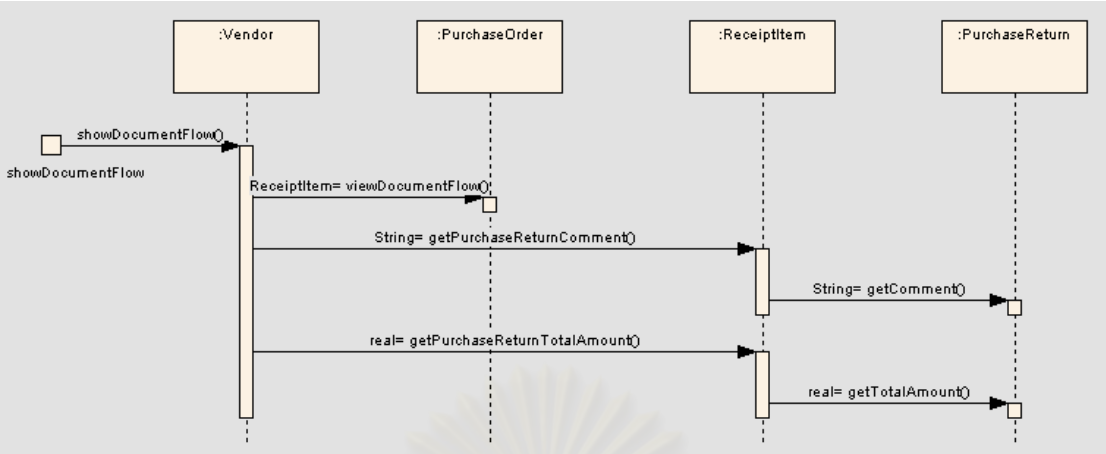
รูปที่ 5.28 แผนภาพซีเควนซ์ของระบบตัวอย่างที่เมทธอด getTransactions ของคลาส Member มีข้อบกพร่องประเภท Message Chains

5.3.2 การประยุกต์ใช้วิธีรีแฟคทอริง

5.3.2.1 เมทธอด viewDocumentFlow ของคลาส PurchaseOrder ในระบบตัวอย่างที่ 37ประยุกต์ใช้วิธีการ Hide Legate โดยจะเพิ่มเมทธอด getPurchaseReturnComment, getPurchaseReturnTotalAmount รูปที่ 5.29, 5.30 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง

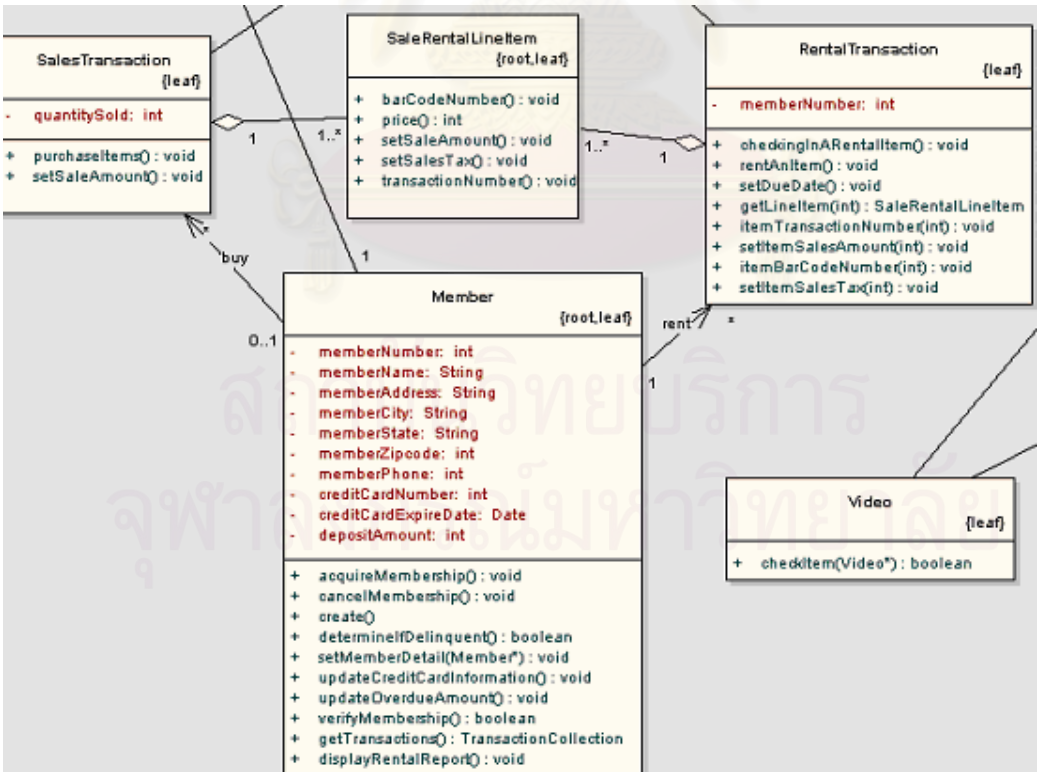


รูปที่ 5.29 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด viewDocumentFlow ของคลาส PurchaseOrder

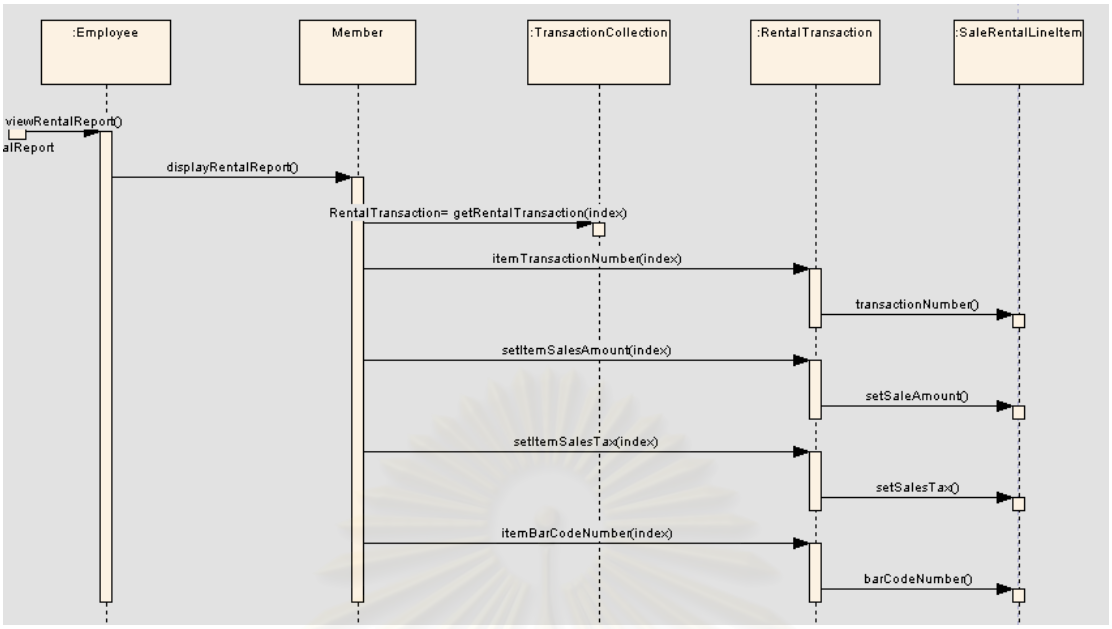


รูปที่ 5.30 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด viewDocumentFlow ของคลาส PurchaseOrder

5.3.2.2 เมทธอด getTransactions ของคลาส Member ในระบบตัวอย่างที่ 38 ประยุกต์ใช้วิธีการ Hide Delegate โดยสร้างเป็นเมทธอดใหม่ชื่อ displayRentalReport ในคลาส Member และสร้างเมทธอดใหม่ที่คลาส RentalTransaction ได้แก่เมทธอด itemTransactionNumber, setItemSalesAmount, setItemSalesTax, itemBarCodeNumber รูปที่ 5.31, 5.32 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



รูปที่ 5.31 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด getTransactions ของคลาส Member



รูปที่ 5.32 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับเมทธอด getTransactions ของคลาส Member

5.3.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง

หลังจากประยุกต์ใช้วิธีการรีแฟคทอริง ผู้วิจัยได้วัดค่าของมาตรวัดสำหรับตรวจจับข้อบกพร่องที่เมทธอดที่มีลักษณะเป็น Message Chains อีกครั้งหนึ่ง ได้ผลการวัดค่ามาตรวัดเป็นดังนี้

เมทธอด viewDocumentFlow ของคลาส PurchaseOrder ในระบบตัวอย่างที่ 37 ไม่ก่อให้เกิด Message Chains แล้ว และค่าของมาตรวัด SOMsgCh เป็น 1

หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.6 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าไม่เปลี่ยนแปลง

ตาราง 5.6 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	0.832	0.992	0.424	0.872
หลัง	0.832	0.992	0.424	0.872

เมทธอด getTransactions ของคลาส Member ในระบบตัวอย่างที่ 38 ไม่ก่อให้เกิด Message Chains แล้ว และค่าของมาตรวัด SOMsgCh เป็น 1

หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.7 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าไม่เปลี่ยนแปลง

ตาราง 5.7 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 38

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	0.946	0.912	0.54	0.566
หลัง	0.946	0.912	0.54	0.566

5.4 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Middle Man

5.4.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ

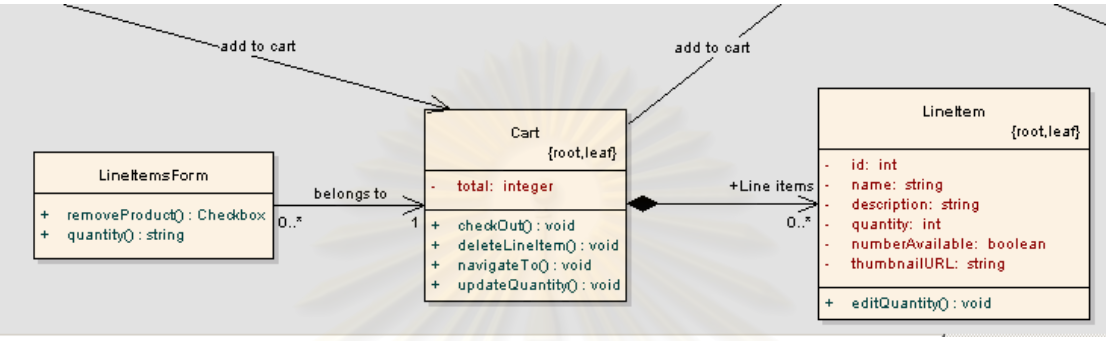
จากกลยุทธ์การตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Middle Man ผู้วิจัยได้นำกลยุทธ์การตรวจจับมาทำการทดสอบตรวจจับข้อบกพร่องของโมเดลการออกแบบ 5 ระบบ ได้ผลการคำนวณมาตรวัดแสดงในภาคผนวก ข ซึ่งจะเห็นว่าพบระบบที่มีค่ามาตรวัดที่บอกถึงข้อบกพร่องดังต่อไปนี้

5.4.1.1 คลาส LineItem ในระบบตัวอย่างที่ 33 ให้ค่ามาตรวัด WOD = 1 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Middle Man และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

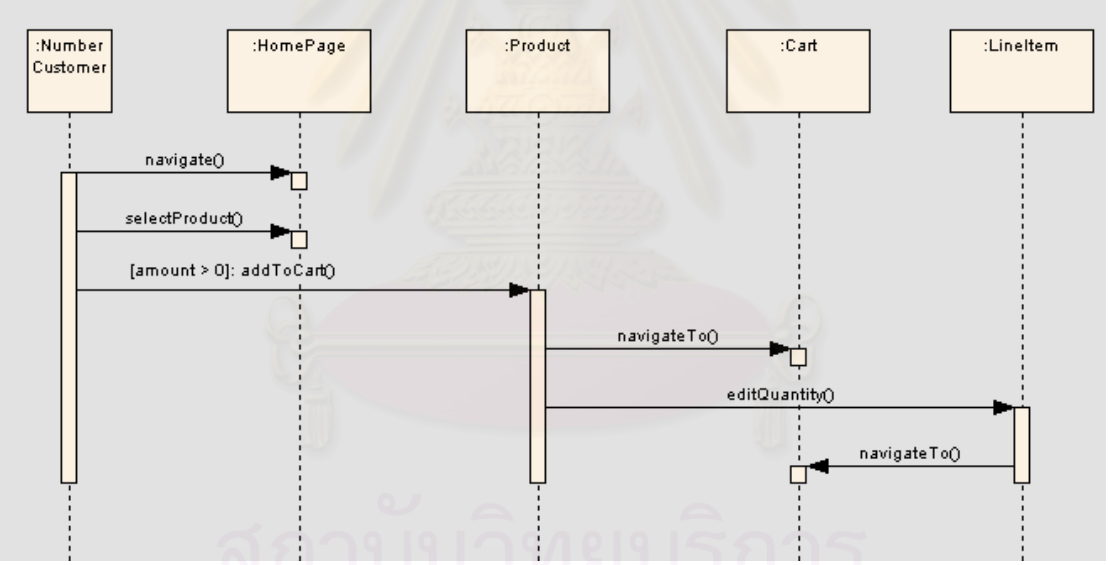
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.946
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.992
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.54

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.818

รูปที่ 5.33, 5.34 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Middle Man ตามลำดับ



รูปที่ 5.33 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส LineItem เป็น Middle Man



รูปที่ 5.34 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส LineItem เป็น Middle Man

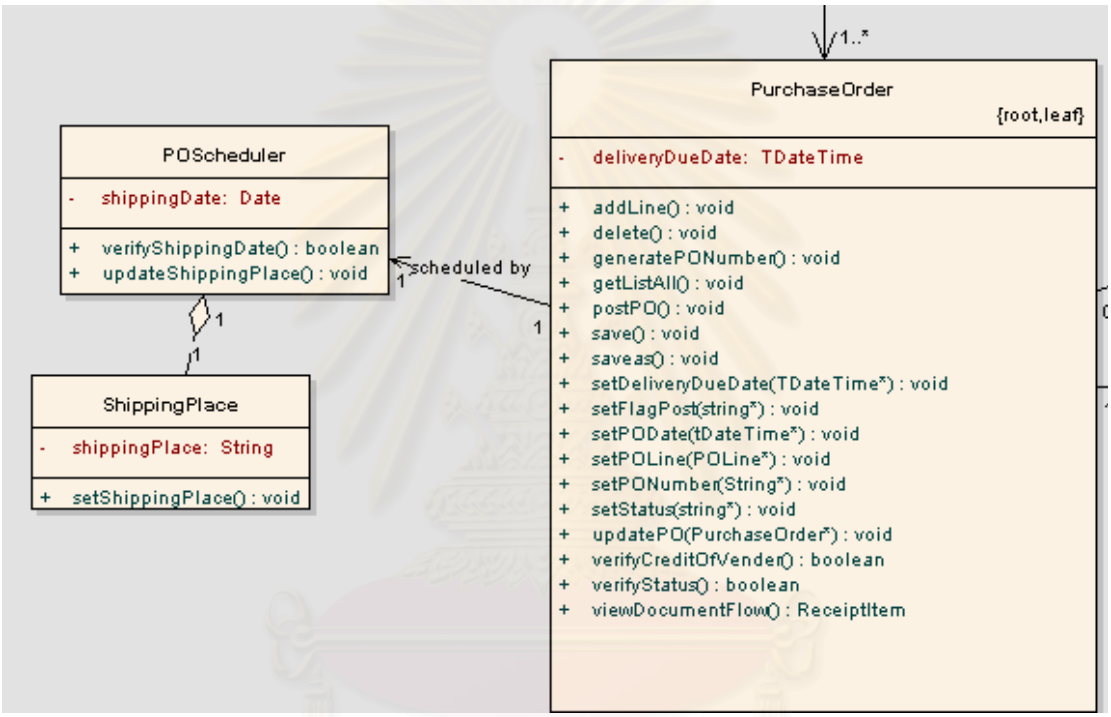
5.4.1.2 คลาส POScheduler ในระบบตัวอย่างที่ 37 ให้ค่ามาตรวัด WOD = 0.5 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Middle Man และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.832
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.992

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.424

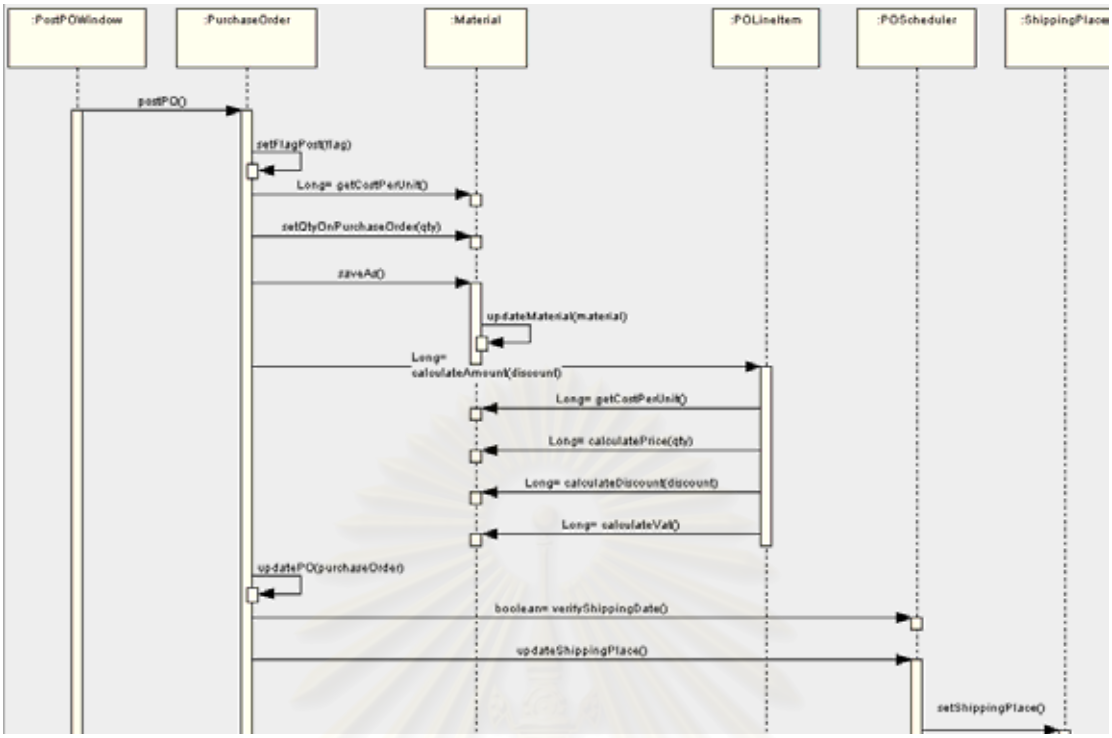
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.872

รูปที่ 5.35, 5.36 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Middle Man ตามลำดับ



รูปที่ 5.35 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส POScheduler เป็น Middle Man

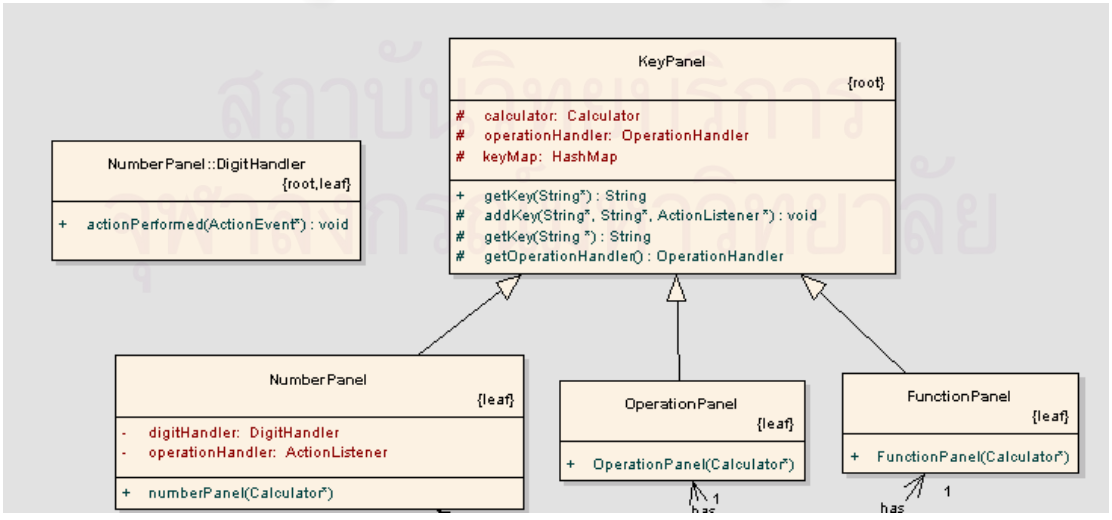
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



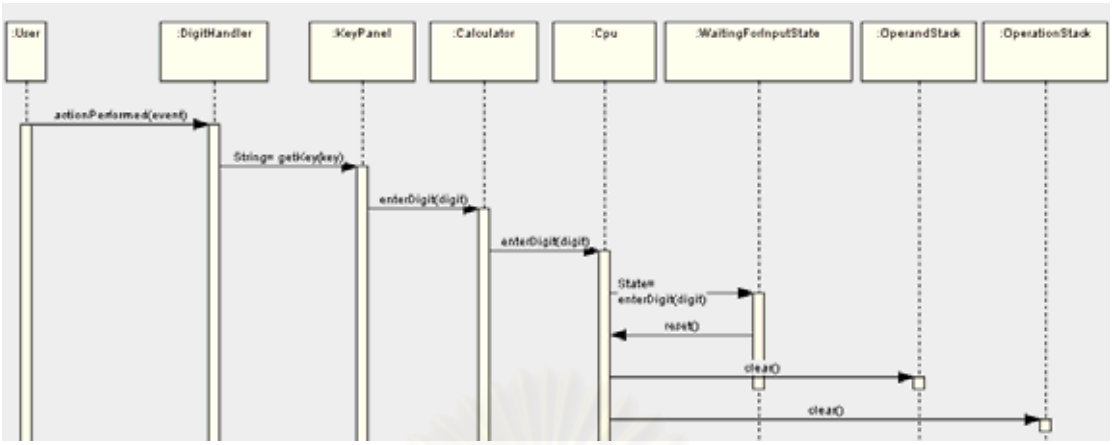
รูปที่ 5.36 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส POScheduler เป็น Middle Man

5.4.1.3 คลาส DigitHandler ในระบบตัวอย่างที่ 40 ให้ค่ามาตรวัด WOD = 1 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Middle Man แต่จากการพิจารณาที่แผนภาพคลาสและแผนภาพซีเควนซ์แล้วพบว่า คลาส DigitHandler เป็นคลาสยูสเซอร์อินเตอร์เฟส (User Interface) จึงไม่ถือว่าผลจากมาตรวัดของคลาสนี้บอกถึงข้อบกพร่อง

รูปที่ 5.37, 5.38 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่ตรวจพบว่าเกิด Middle Man ตามลำดับ



รูปที่ 5.37 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส DigitHandler



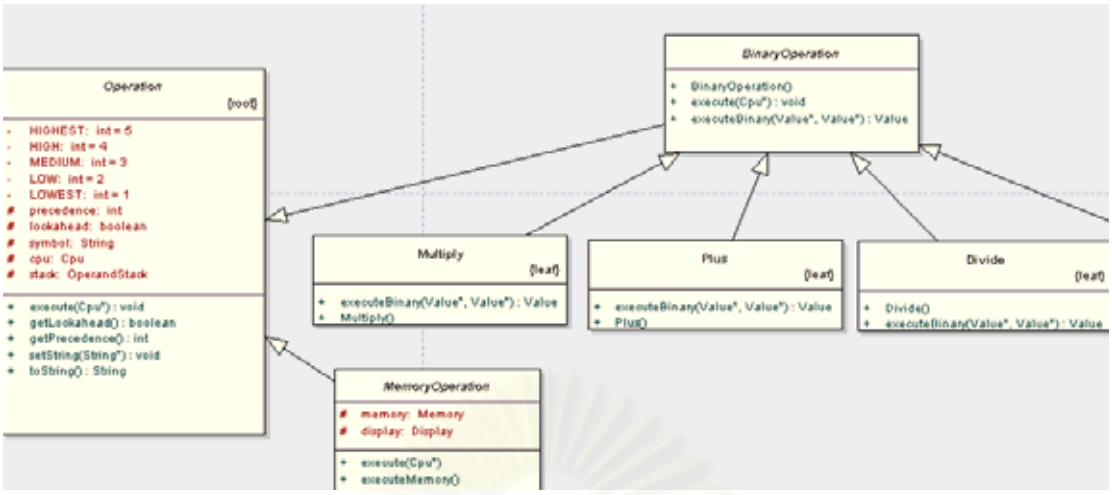
รูปที่ 5.38 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส DigitHandler

5.4.1.4 คลาส BinaryOperation ในระบบตัวอย่างที่ 40 ให้ค่ามาตรวัด WOD = 0.5 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Middle Man และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

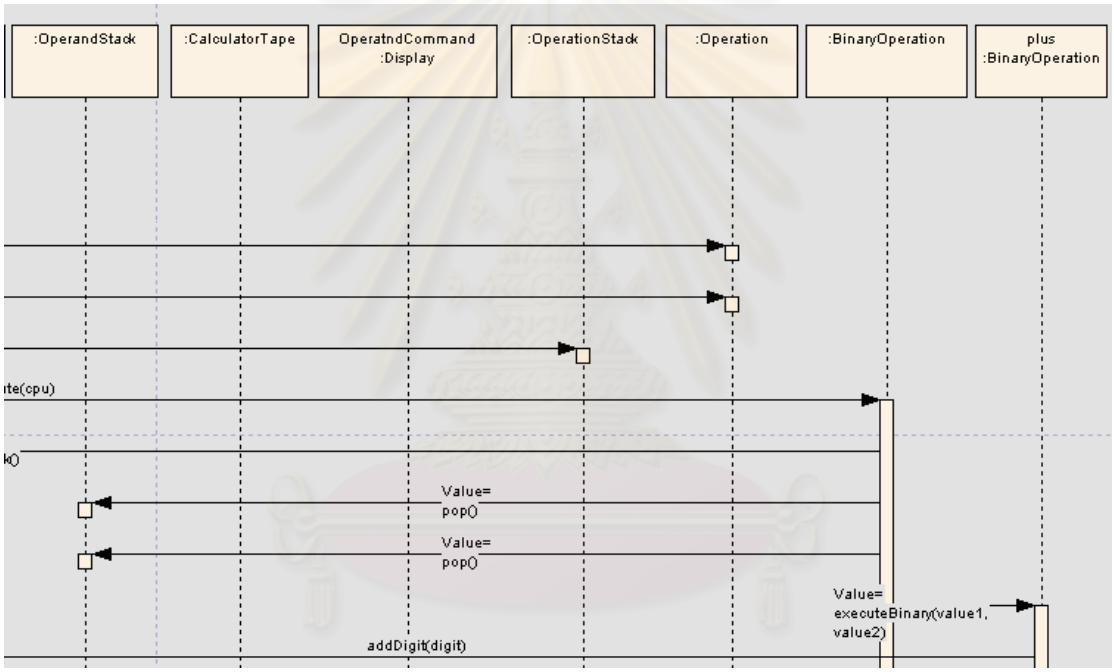
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 1.402
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.832
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 1.004
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.575

รูปที่ 5.39, 5.40 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Middle Man ตามลำดับ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



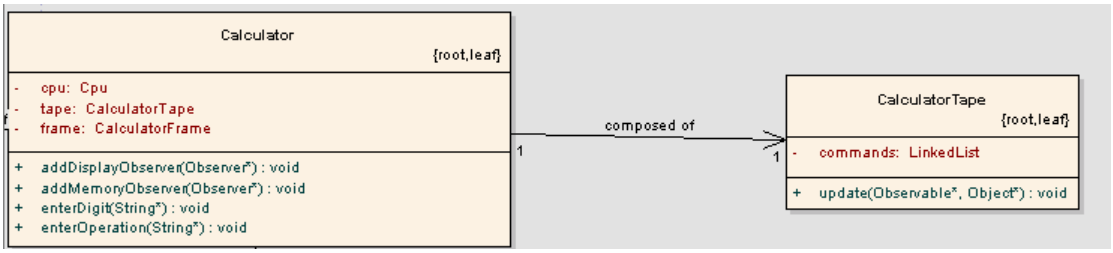
รูปที่ 5.39 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส BinaryOperation เป็น Middle Man



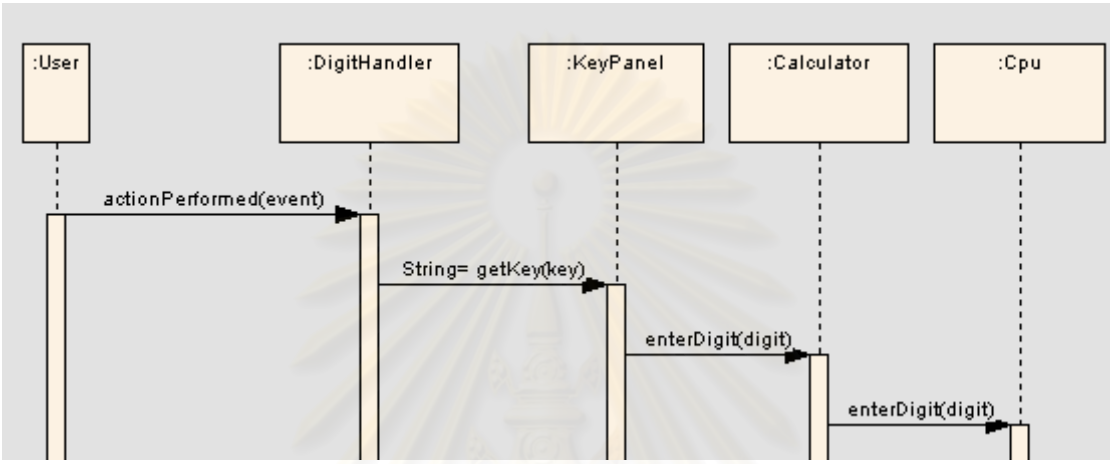
รูปที่ 5.40 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส BinaryOperation เป็น Middle Man

5.4.1.5 คลาส Calculatorในระบบตัวอย่างที่ 40 ให้ค่ามาตรวัด WOD = 0.5 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Middle Man แต่จากการพิจารณาที่แผนภาพคลาสและแผนภาพซีเควนซ์แล้วพบว่า คลาส Calculator เป็นคลาสยูสเซอร์อินเตอร์เฟส (User Interface) จึงไม่ถือว่าผลจากมาตรวัดของคลาสนี้บอกถึงข้อบกพร่อง

รูปที่ 5.41, 5.42 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่ตรวจพบว่าเกิด Middle Man ตามลำดับ



รูปที่ 5.41 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส Calculator เป็น Middle Man



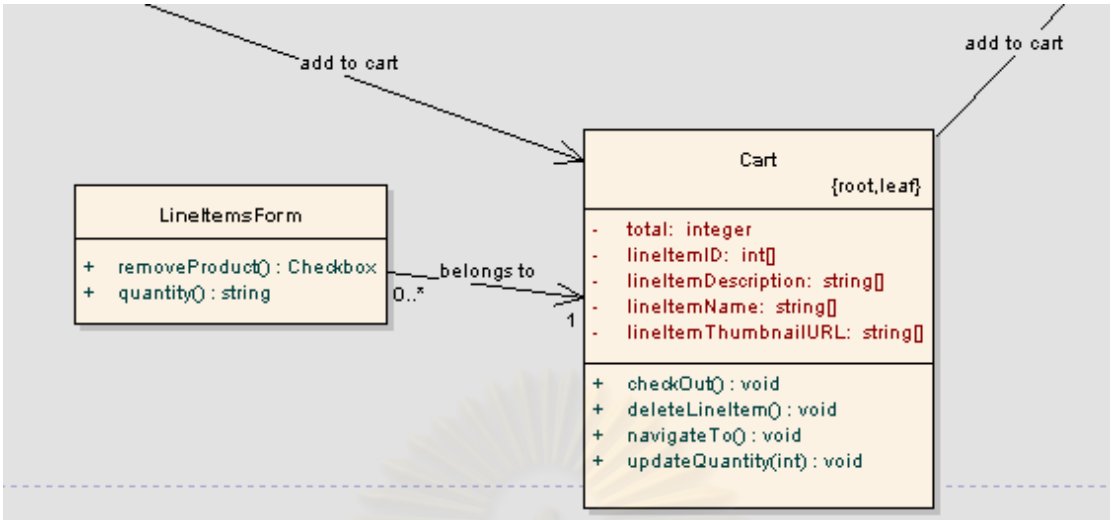
รูปที่ 5.42 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส Calculator เป็น Middle Man

5.4.2 การประยุกต์ใช้วิธีรีแฟคทอริง

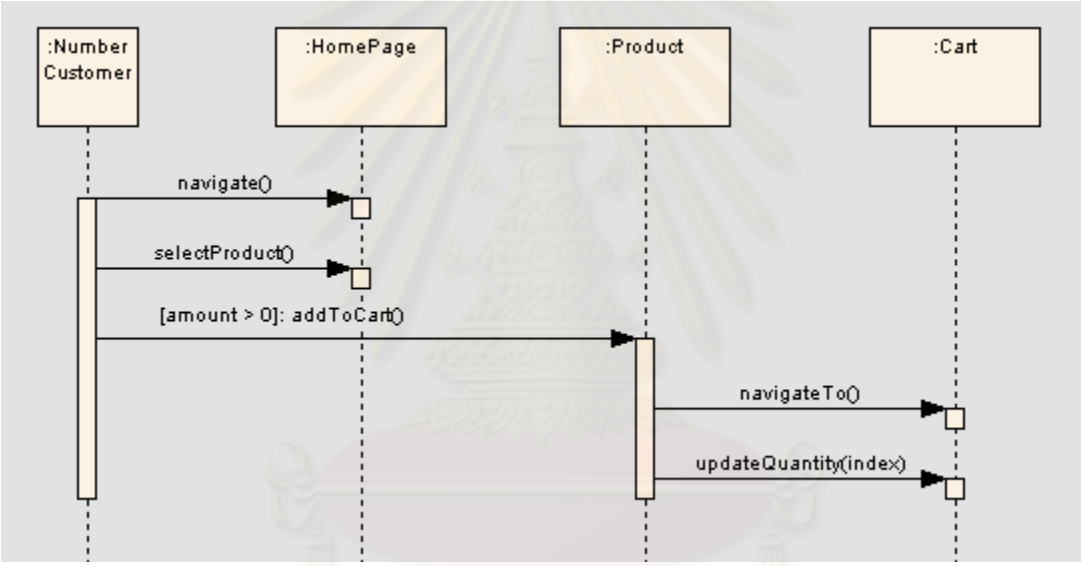
5.4.2.1 คลาส LineItem ในระบบตัวอย่างที่ 33 ประยุกต์ใช้วิธีการ Inline Method รวมเมทธอด editQuantity ของคลาส LineItem รวมกับเมทธอด updateQuantity ของคลาส Cart แล้วใช้วิธี Move Attribute ของคลาส LineItem ไปไว้ที่ Cart

รูปที่ 5.43, 5.44 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

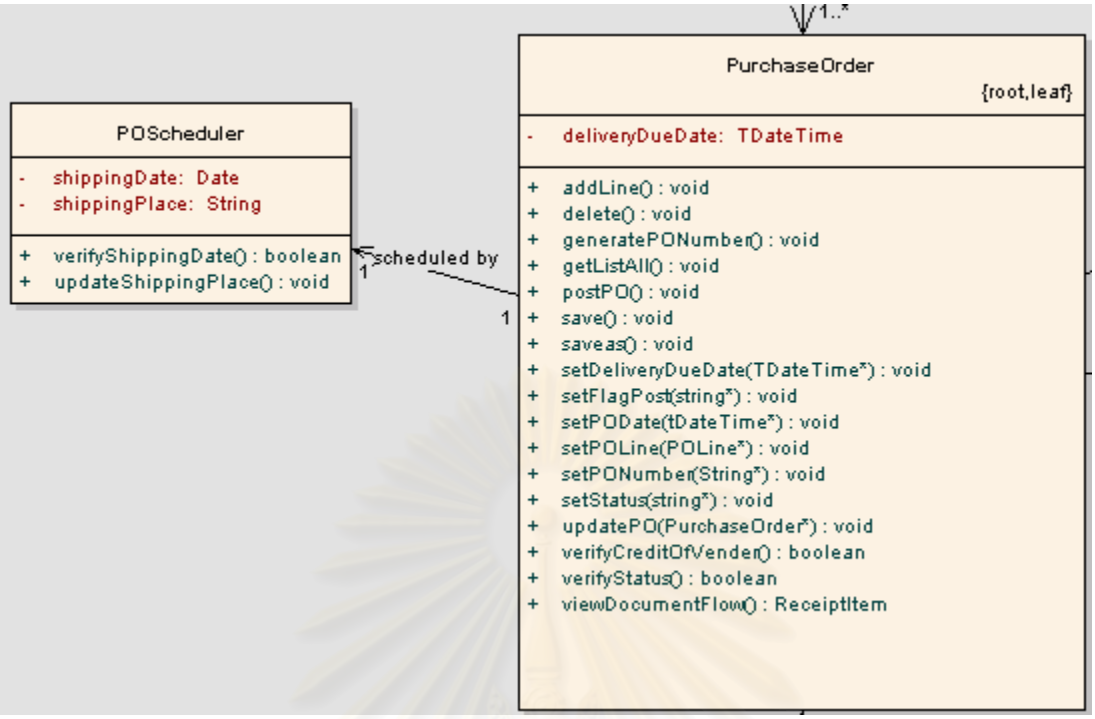


รูปที่ 5.43 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส LineItem

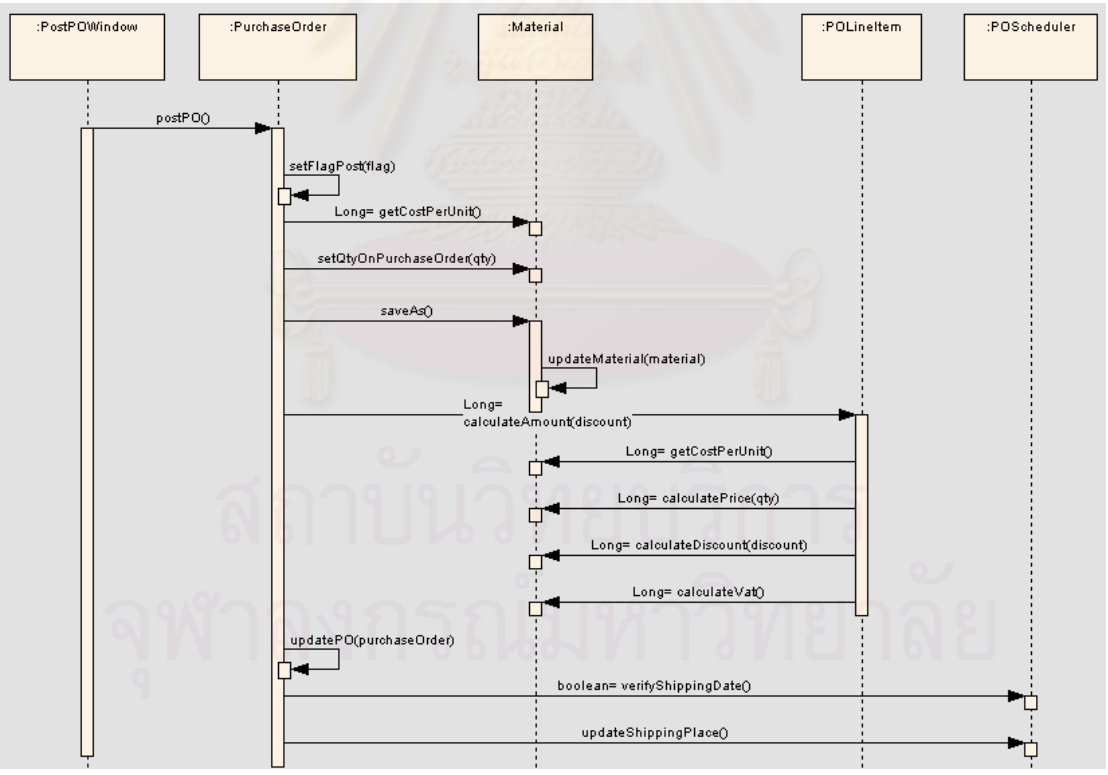


รูปที่ 5.44 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส LineItem

5.4.2.2 คลาส POScheduler ในระบบตัวอย่างที่ 37 ประยุกต์ใช้วิธีการ Inline Method ที่ชื่อ setShippingPlace มาไว้ที่คลาส POScheduler รวมทั้งย้ายคุณลักษณะของคลาส ShippingPlace มาไว้ที่ POScheduler ด้วย ซึ่งรูปที่ 5.45, 5.46 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



รูปที่ 5.45 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส POScheduler

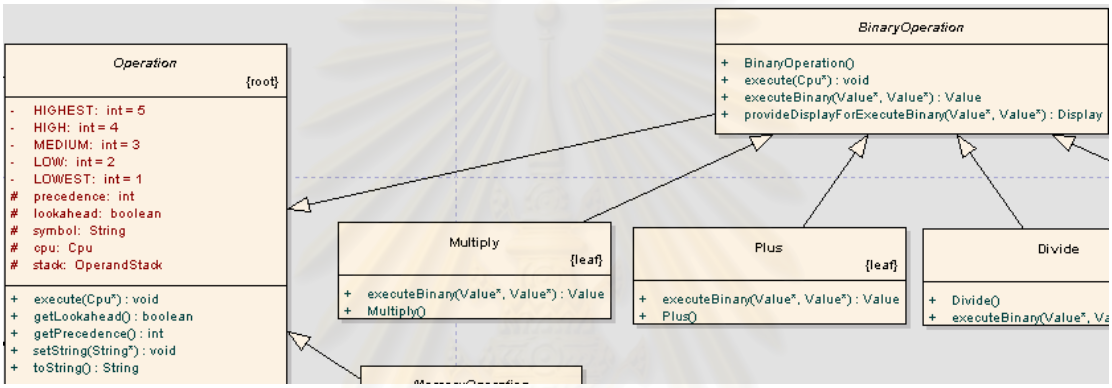


รูปที่ 5.46 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส POScheduler

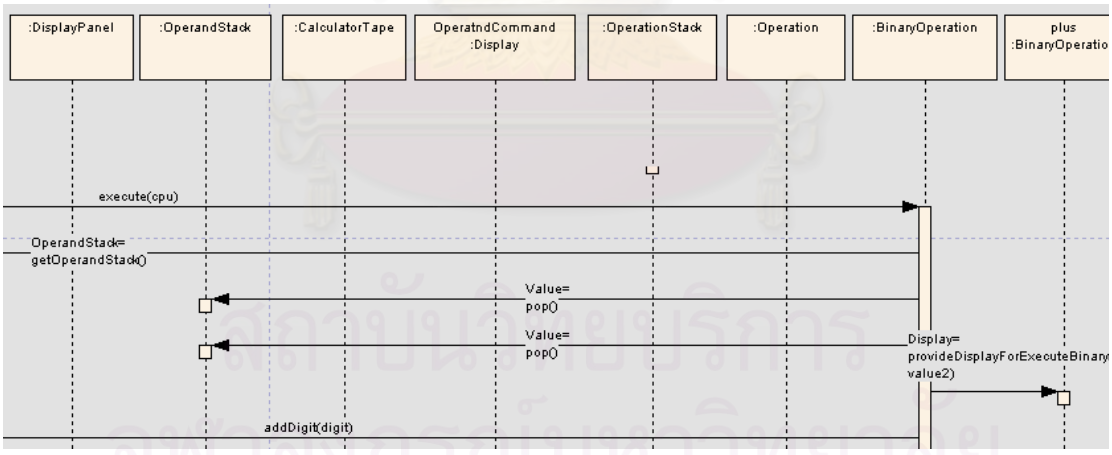
5.4.2.3 คลาส BinaryOperation ในระบบตัวอย่างที่ 40 ประยุกต์ใช้วิธีการ Remove Middle Man โดยที่เมทธอดชื่อ executeBinary ของคลาส BinaryOperation ก่อให้เกิดลักษณะ Middle Man เนื่องจากการที่เมทธอด executeBinary มีการมอบงานให้แก่เมทธอด addDigit ของ

คลาส Display ดังนั้นจึงใช้วิธีการ Remove Middle Man โดยเปลี่ยนให้เมทธอด executeBinary กลายเป็นเมทธอด provideDisplayForExecuteBinary ซึ่งมีเอาท์พุทเป็นออบเจกต์ของคลาส Display แล้วเปลี่ยนการเรียกใช้เมทธอด executeBinary ให้คลาสที่เคยเรียกใช้เมทธอด executeBinary ไปเรียกใช้เมทธอด provideDisplayForExecuteBinary จากคลาส BinaryOperation และเรียกใช้เมทธอด addDigit โดยตรงที่คลาส Display เอง

ซึ่งรูปที่ 5.47, 5.48 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



รูปที่ 5.47 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส BinaryOperation



รูปที่ 5.48 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส BinaryOperation

5.4.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง

หลังจากประยุกต์ใช้วิธีการรีแฟคทอริงเรียบร้อยแล้ว ผู้วิจัยได้ทำการวัดค่าของมาตรวัดสำหรับตรวจจับข้อบกพร่องที่เมทธอดที่มีลักษณะเป็น Middle Man อีกครั้งหนึ่ง ได้ผลการวัดค่ามาตรวัดเป็นดังนี้

คลาส LineItem ในระบบตัวอย่างที่ 33 ไม่ก่อให้เกิด Middle Man แล้วเนื่องจากมีการลบคลาสทิ้ง หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.8 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าที่ดีขึ้นสำหรับโมเดลความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH, โมเดลความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH, และโมเดลความสมบูรณ์ของการเปลี่ยนแปลง = 1.079 – 0.009 *NA

ตาราง 5.8 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 33

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	0.946	0.992	0.54	0.818
หลัง	1.06	0.992	0.656	0.836

คลาส POScheduler ในระบบตัวอย่างที่ 37 ไม่ก่อให้เกิด Middle Man แล้ว และค่าของมาตรวัด WOD เป็น 0 หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.9 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าที่ดีขึ้นสำหรับโมเดลความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH และ โมเดลความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH

ตาราง 5.9 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	0.832	0.992	0.424	0.872
หลัง	0.946	0.992	0.54	0.872

คลาส BinaryOperation ในระบบตัวอย่างที่ 40 ไม่ก่อให้เกิด Middle Man แล้ว และค่าของมาตรวัด WOD เป็น 0 หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการ

เปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.10 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบไม่เปลี่ยนแปลง

ตาราง 5.10 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 40

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	1.402	0.832	1.004	0.575
หลัง	1.402	0.832	1.004	0.575

5.5 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท God Class

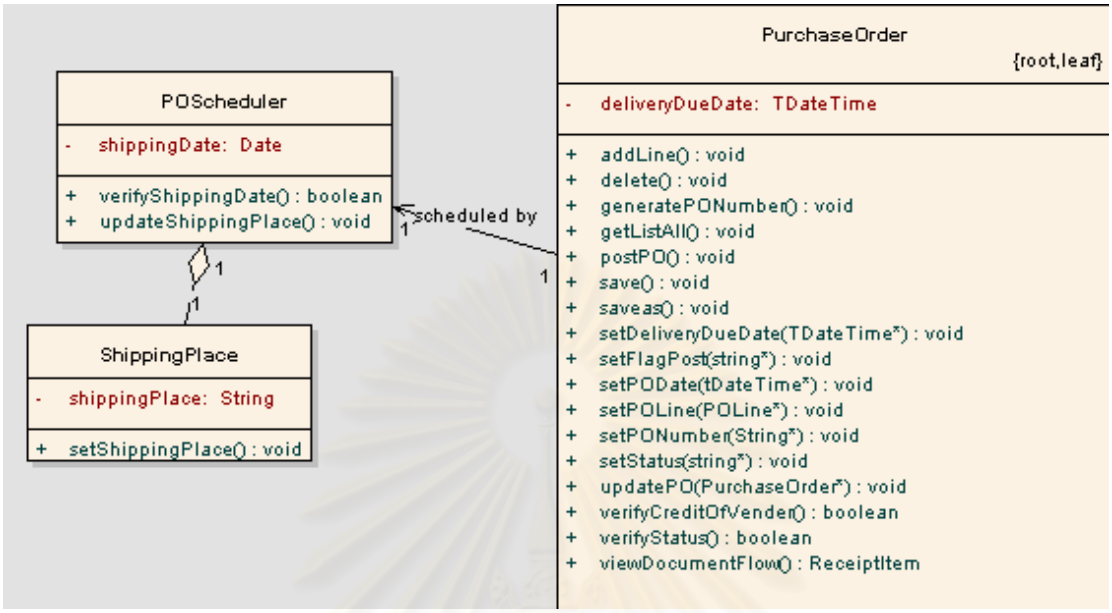
5.5.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ

จากกลยุทธ์การตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท God Class ผู้วิจัยได้นำกลยุทธ์การตรวจจับมาทำการทดสอบตรวจจับข้อบกพร่องของโมเดลการออกแบบ 5 ระบบ ได้ผลการคำนวณมาตรวัดแสดงในภาคผนวก ข ซึ่งจะเห็นว่าพบระบบที่มีค่ามาตรวัดที่บอกถึงข้อบกพร่องดังต่อไปนี้

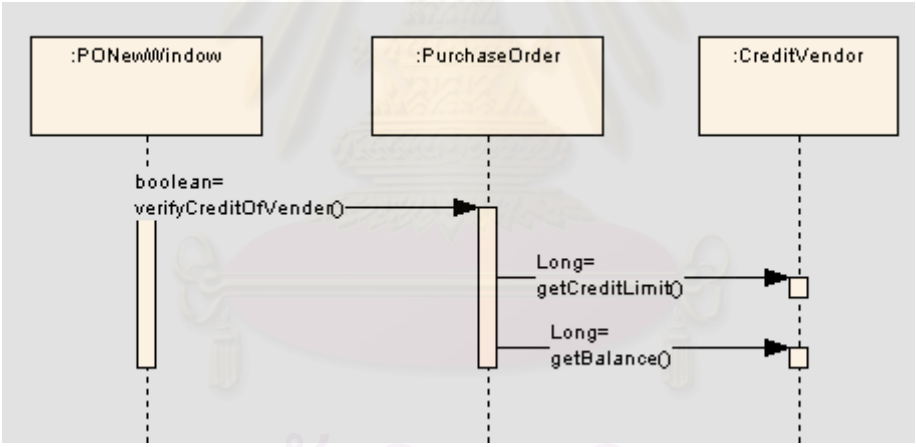
5.5.1.1 คลาส PurchaseOrder ในระบบตัวอย่างที่ 37 ให้ค่ามาตรวัด NCPA = 3, RFC = 25, COH = 0.01 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ God Class และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 0.832
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.992
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 0.424
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.872

รูปที่ 5.49, 5.50, 5.51, 5.52 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด God Class

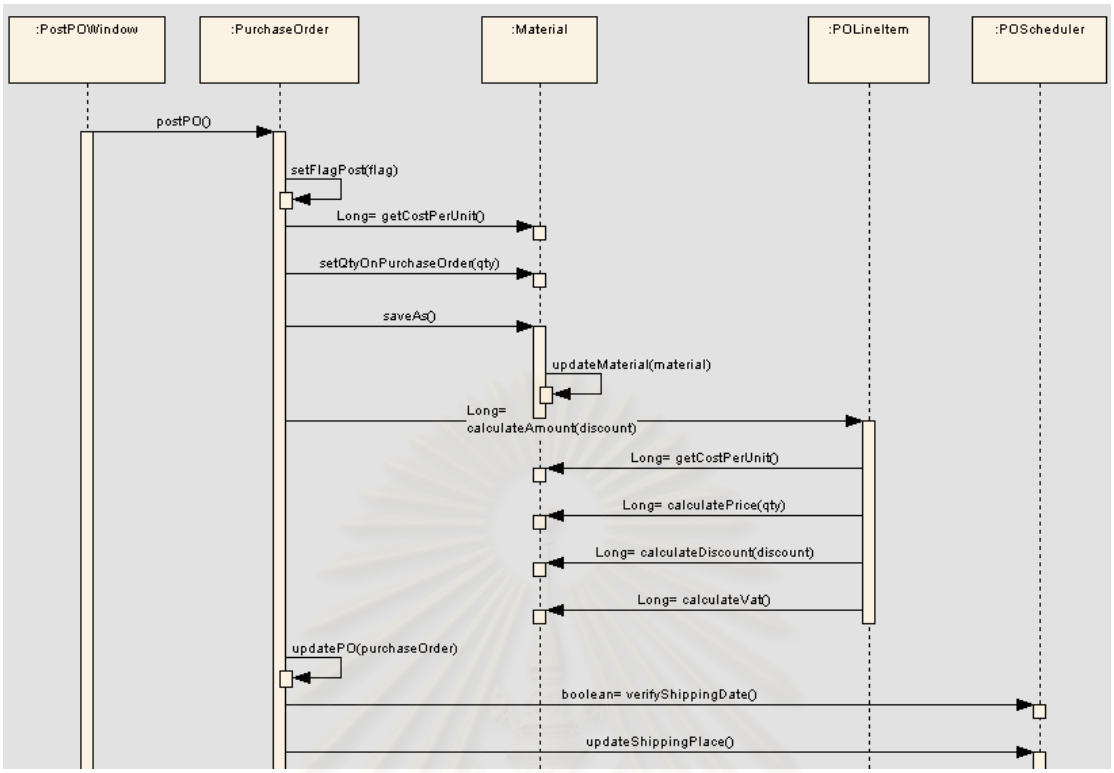


รูปที่ 5.49 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส PurchaseOrder เป็น God Class

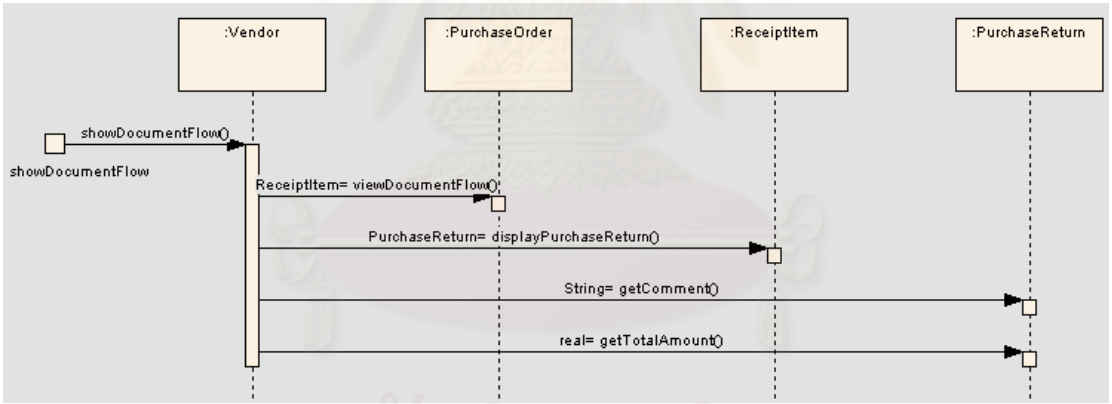


รูปที่ 5.50 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส PurchaseOrder เป็น God Class

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.51 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส PurchaseOrder เป็น God Class



รูปที่ 5.52 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส PurchaseOrder เป็น God Class

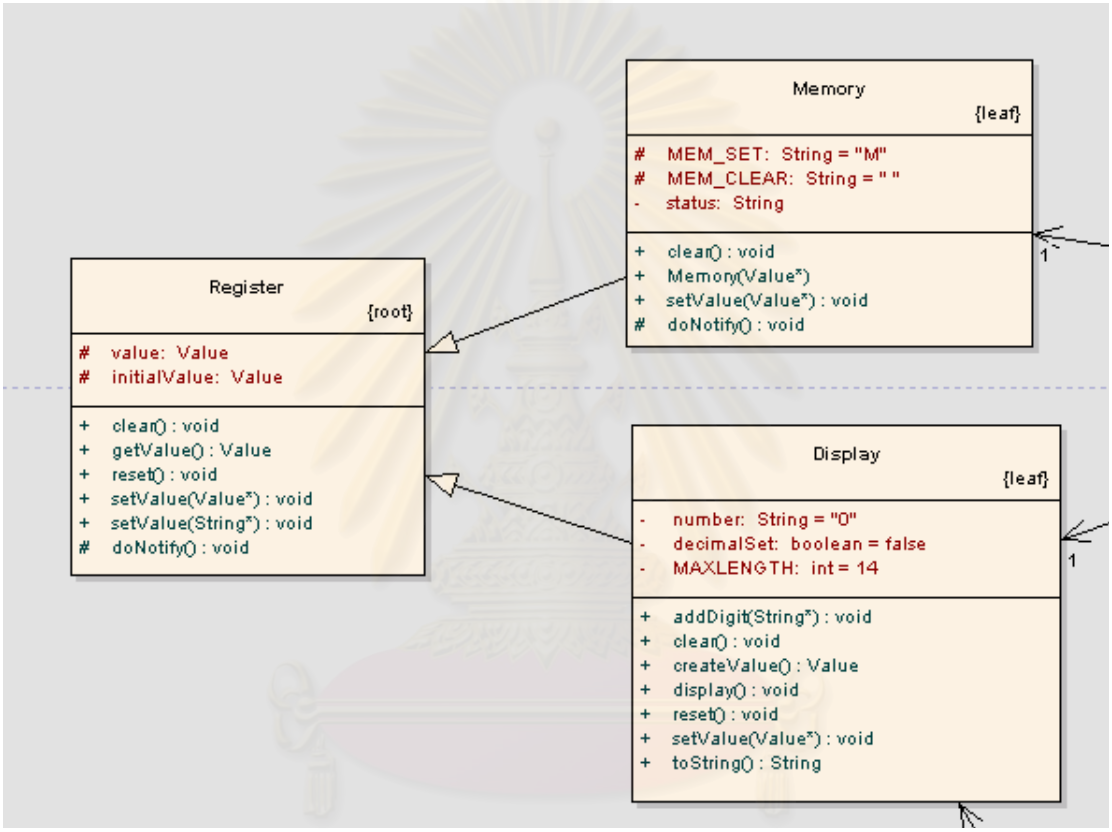
5.5.1.2 คลาส Display ในระบบตัวอย่างที่ 40 ให้ค่ามาตรวัด NCPA = 2, RFC = 13, COH = 0.18 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ God Class และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 1.402
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.832

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 1.004

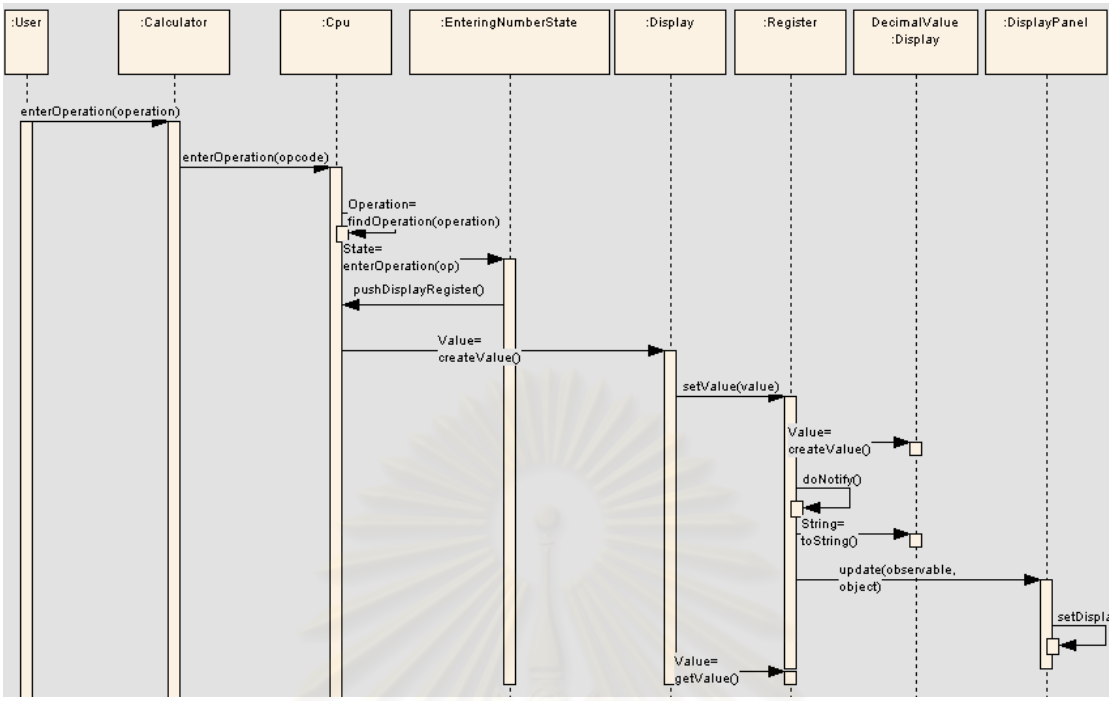
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.575

รูปที่ 5.53, 5.54 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด God Class ตามลำดับ



รูปที่ 5.53 แผนภาพคลาสของระบบตัวอย่างที่มีคลาส Display เป็น God Class

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

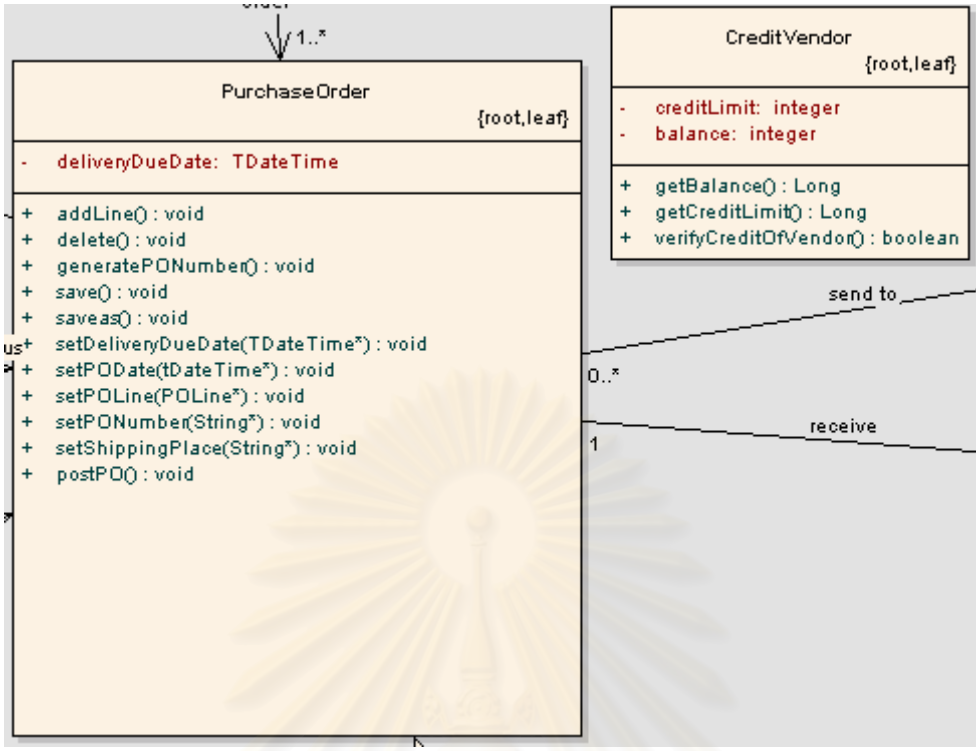


รูปที่ 5.54 แผนภาพซีเควนซ์ของระบบตัวอย่างที่มีคลาส Display เป็น God Class

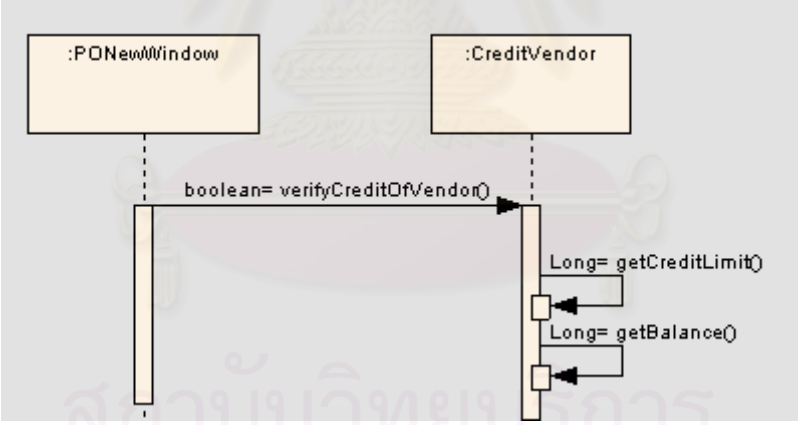
5.5.2 การประยุกต์ใช้วิธีรีแฟคทอริง

5.5.2.1 คลาส PurchaseOrder ในระบบตัวอย่างที่ 37 มีการเรียกใช้แอคเซสเซอร์เมทธอดของคลาส CreditVendor เพื่อนำข้อมูลของ CreditVendor มาประมวลผลที่คลาสตัวเอง จึงประยุกต์ใช้วิธี Move Method เพื่อย้ายเมทธอด verifyCreditOfVendor จากคลาส PurchaseOrder ไปอยู่ที่คลาส CreditVendor แทน เนื่องจากการตรวจสอบควรเป็นหน้าที่ของคลาส CreditVendor ซึ่งรูปที่ 5.55, 5.56 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



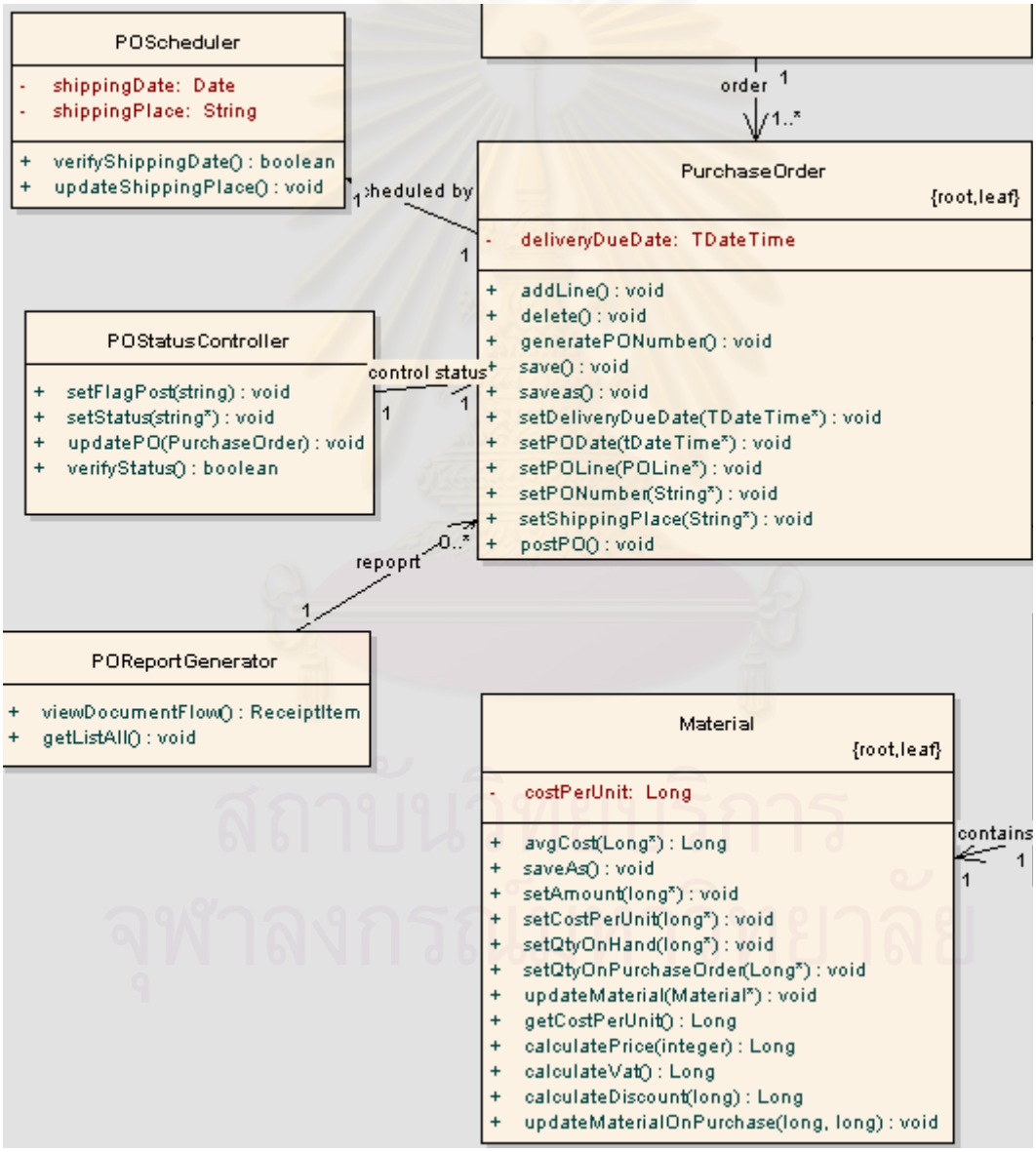
รูปที่ 5.55 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder และ CreditVendor



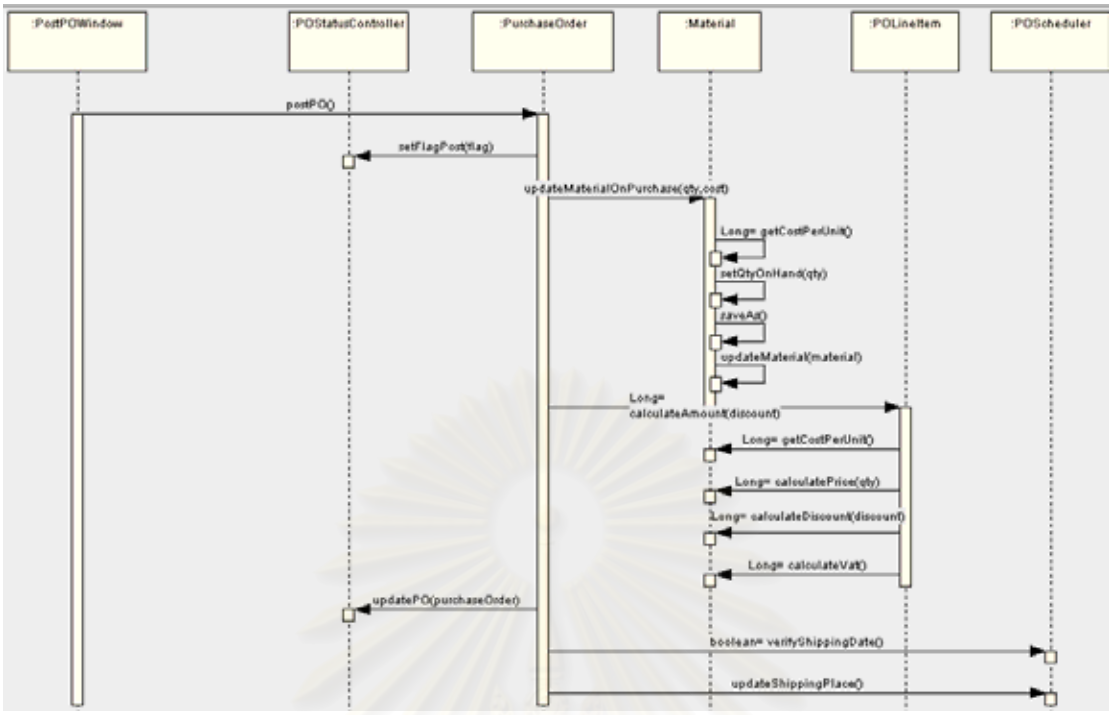
รูปที่ 5.56 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder และ CreditVendor

คลาส PurchaseOrder มีการทำงานในส่วนที่เกี่ยวกับการจัดการสถานะรวมอยู่ในคลาสด้วย จึงใช้วิธี Extract Class ไปสร้างเป็นคลาส POStatusController เพื่อแยกการทำงานในส่วนนี้ออกไปจากคลาส PurchaseOrder และจะเห็นได้ว่าคลาส PurchaseOrder ยังมีการเรียกใช้งานแอคเซสเซอร์ เมทธอดของคลาส Material เมื่อมีการเปลี่ยนแปลงข้อมูลเกี่ยวกับ Material จากการที่เมทธอด PostPO ถูกเรียกใช้งาน ดังนั้นจึงมีการประยุกต์ใช้วิธีรีแฟคทอริงด้วยการ Extract Method เพื่อสร้างเป็นเมทธอด updateMaterialOnPurchase ในคลาส Material ที่มีการทำงาน

เป็นการเรียกใช้แอคเซสเซอร์ เมทธอดของคลาส Material เอง แล้วให้คลาส PurchaseOrder ไปเรียกใช้งานเมทธอด updateMaterialOnPurchase แทนเพื่อลดการทำงานของคลาส PurchaseOrder นอกจากนี้คลาส PurchaseOrder มีการเรียกใช้งานเมทธอด updateShippingPlace ของคลาส POScheduler ซึ่งไปเรียกเมทธอด setShippingPlace ของคลาส ShippingPlace อีกครั้ง จึงประยุกต์ใช้วิธี Inline Method และ Move Attribute ที่ชื่อ shippingPlace ของคลาส ShippingPlace มาอยู่ที่คลาส POScheduler เพื่อรวมการทำงานของการ setShippingPlace มาอยู่ที่คลาส POScheduler และลบคลาส ShippingPlace ทิ้ง รูปที่ 5.57, 5.58 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



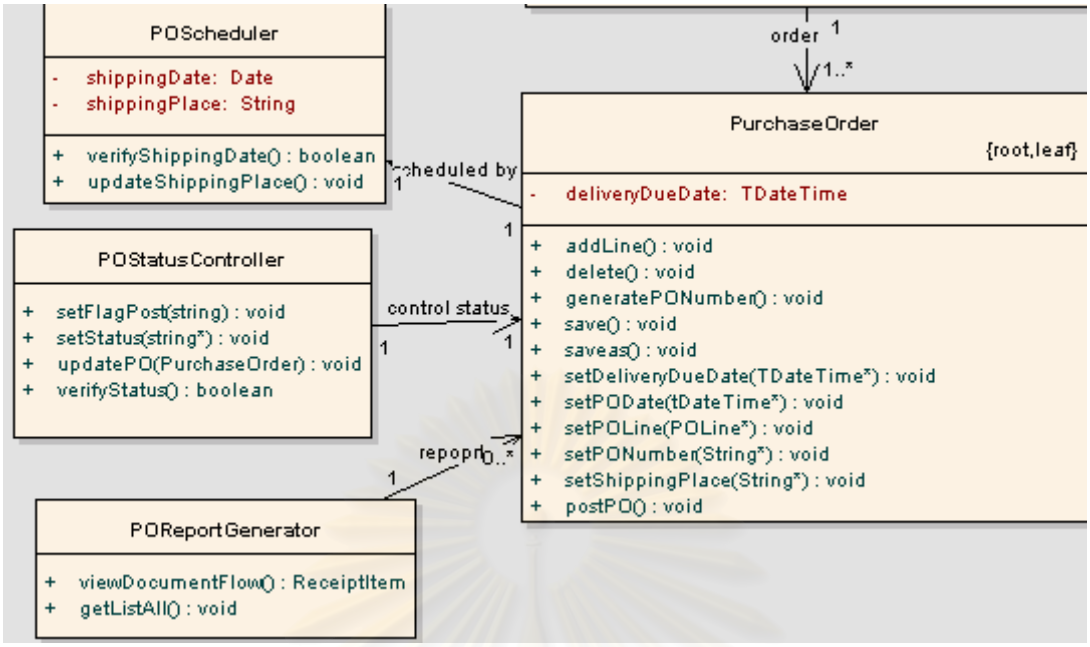
รูปที่ 5.57 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder, POStatusController, Material, และคลาส POScheduler



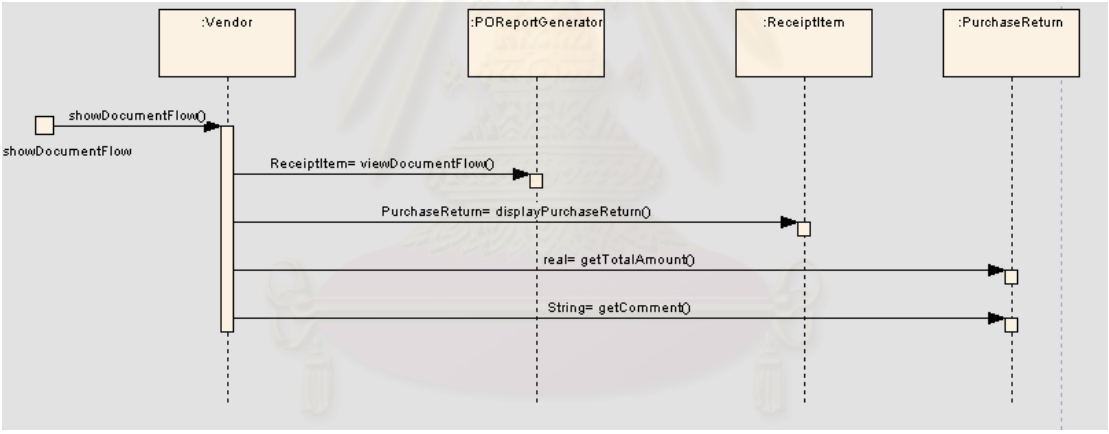
รูปที่ 5.58 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder, POStatusController, Material, และคลาส POScheduler

นอกจากนี้คลาส PurchaseOrder ยังมีการทำงานในส่วนที่เกี่ยวกับรายงาน คือมีเมทธอด viewDocumentFlow ซึ่งสามารถประยุกต์ใช้วิธีรีแฟคทอริง Extract Class เพื่อไปสร้างคลาส POReportGenerator ให้มีเมทธอด viewDocumentFlow ทำงานเกี่ยวกับรายงานแทน และเปลี่ยนให้คลาสที่เคยเรียกใช้เมทธอด viewDocumentFlow ของคลาส PurchaseOrder ให้ไปเรียกที่คลาส POReportGenerator แทน ซึ่งรูปที่ 5.59, 5.60 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

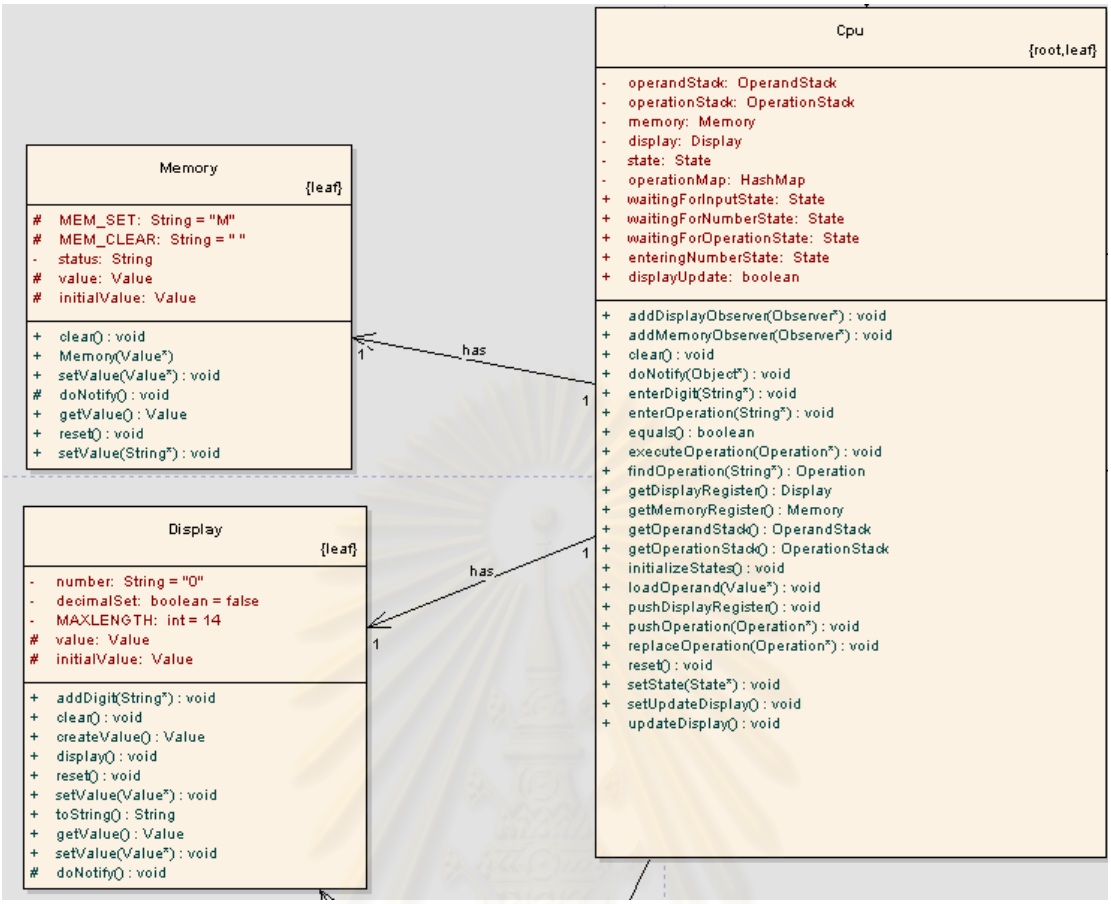


รูปที่ 5.59 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder และ POReportGenerator

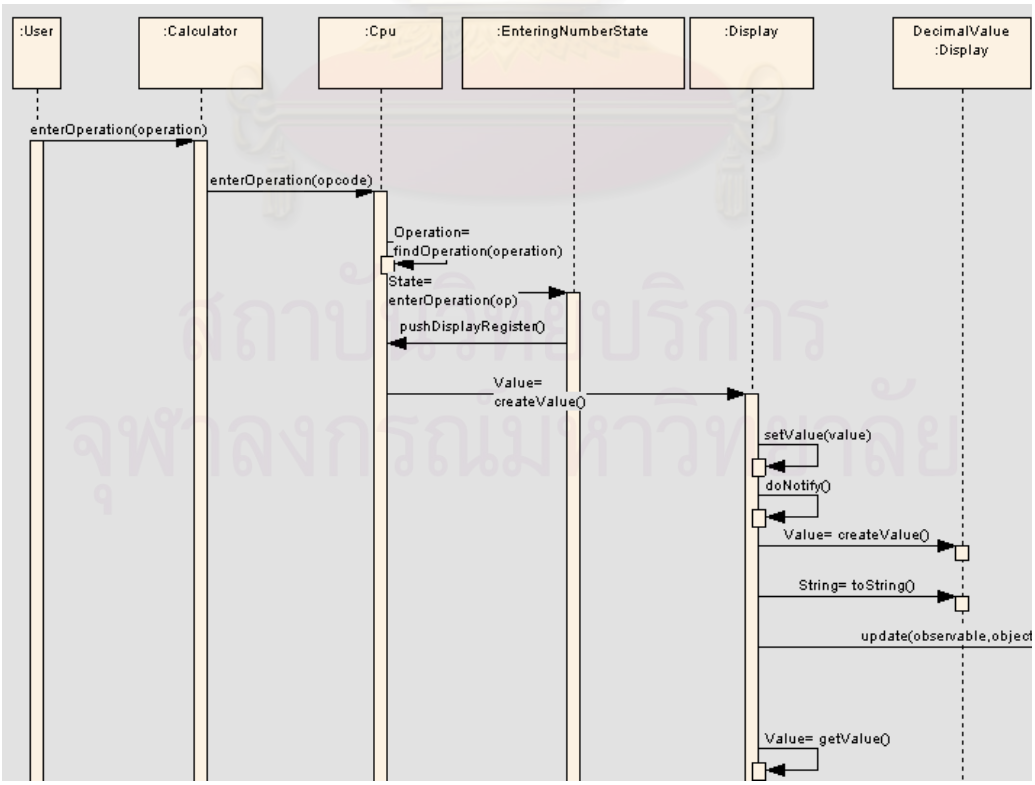


รูปที่ 5.60 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส PurchaseOrder และ POReportGenerator

5.5.2.2 คลาส Display ในระบบตัวอย่างที่ 40 มีการเรียกใช้แอคเซสเซอร์เมทธอดของคลาส Register ซึ่งเป็น Superclass ของตัวเอง และจากลักษณะการเรียกใช้จะเห็นได้ว่าไม่มีคลาสอื่นเรียกใช้งานคลาส Register เลย จึงประยุกต์ใช้วิธีการ Collapse Hierarchy เพื่อกระจายคุณลักษณะและเมทธอดของคลาส Register ไปยังคลาสลูกซึ่งได้แก่คลาส Display และคลาส Memory แล้วลบคลาส Register รวมถึงลำดับชั้นการสืบทอดคุณสมบัติออกจากโมเดล และเปลี่ยนการเรียกใช้งานเมทธอดของคลาส Register ให้ไปเรียกใช้งานที่คลาส Display แทน ซึ่งรูปที่ 5.61, 5.62 แสดงแผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



รูปที่ 5.61 แผนภาพคลาสหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Display



รูปที่ 5.62 แผนภาพซีเควนซ์หลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Display

5.5.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง

หลังจากประยุกต์ใช้วิธีการรีแฟคทอริงเรียบร้อยแล้ว ผู้วิจัยได้ทำการวัดค่าของมาตรวัดสำหรับตรวจจับข้อบกพร่องที่เมทธอดที่มีลักษณะเป็น God Class อีกครั้งหนึ่ง ได้ผลการวัดค่ามาตรวัดเป็นดังนี้

คลาส PurchaseOrder ในระบบตัวอย่างที่ 37 ให้ค่ามาตรวัด NCPA = 0, RFC = 17, COH = 0 ซึ่งไม่ก่อให้เกิดข้อบกพร่องประเภท God Class แล้ว หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.11 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าที่ดีขึ้นสำหรับโมเดลความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH และ 1.004 – 0.116 NAggH

ตาราง 5.11 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 37

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	0.832	0.992	0.424	0.872
หลัง	0.946	0.992	0.54	0.872

คลาส Display ในระบบตัวอย่างที่ 40 ให้ค่ามาตรวัด NCPA = 0, RFC = 14, COH = 0.09 ซึ่งไม่ก่อให้เกิดข้อบกพร่องประเภท God Class แล้ว หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.12 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าที่ดีขึ้นสำหรับโมเดลความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH และค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าที่ดีขึ้นสำหรับโมเดลความสมบูรณ์ของการเปลี่ยนแปลง =1.079 – 0.009 * NA เนื่องจากการรีแฟคทอริงมีการกระจายคุณลักษณะของ Superclass ไปยังคลาสลูกทำให้จำนวนคุณลักษณะโดยรวมเพิ่มขึ้น

ตาราง 5.12 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 40

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	1.402	0.832	1.004	0.575
หลัง	1.402	0.872	1.004	0.557

5.6 การทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Switch Statements

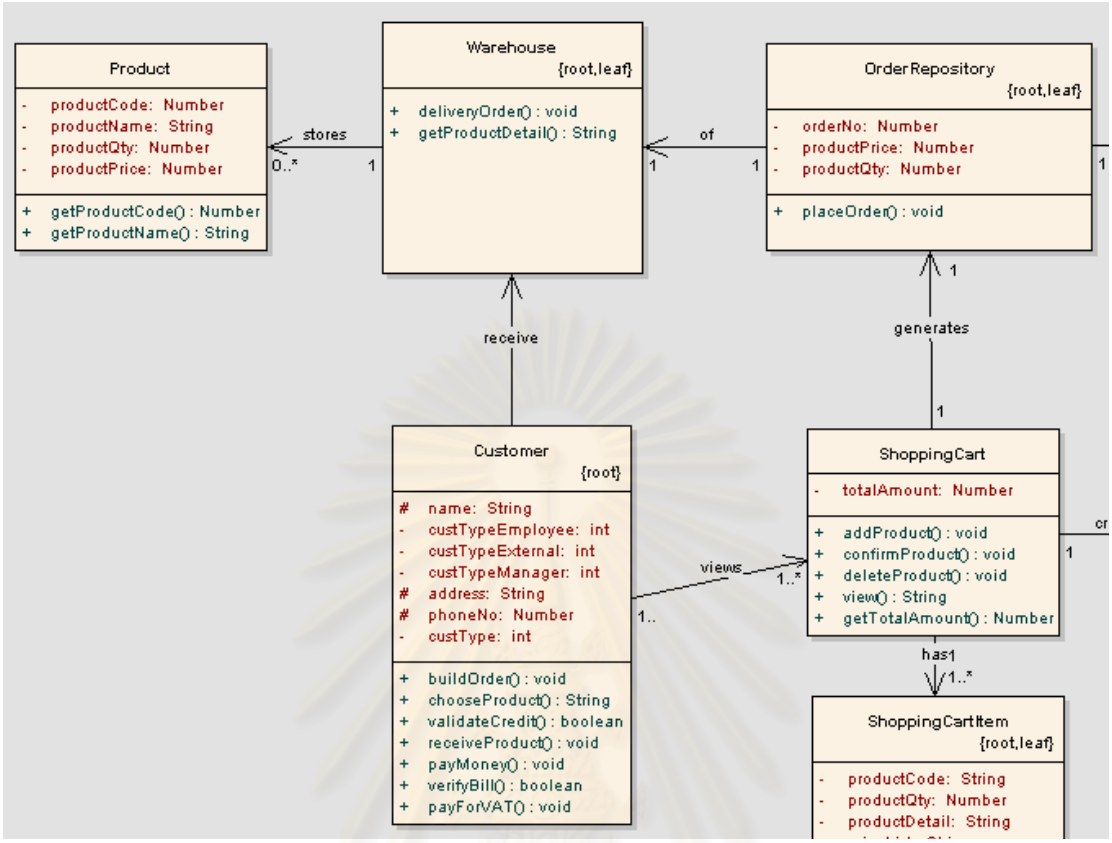
5.6.1 การตรวจจับข้อบกพร่องของโมเดลการออกแบบ

จากกลยุทธ์การตรวจจับข้อบกพร่องของโมเดลการออกแบบประเภท Switch Statements ผู้วิจัยได้นำกลยุทธ์การตรวจจับมาทำการทดสอบตรวจจับข้อบกพร่องของโมเดลการออกแบบ 5 ระบบ ได้ผลการคำนวณมาตรวัดแสดงในภาคผนวก ข ซึ่งจะเห็นว่าพบระบบที่มีค่ามาตรวัดที่บอกถึงข้อบกพร่องดังต่อไปนี้

5.6.1.1 ระบบตัวอย่างที่ 41 ให้ค่ามาตรวัด NOSS = 1 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Switch Statements และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

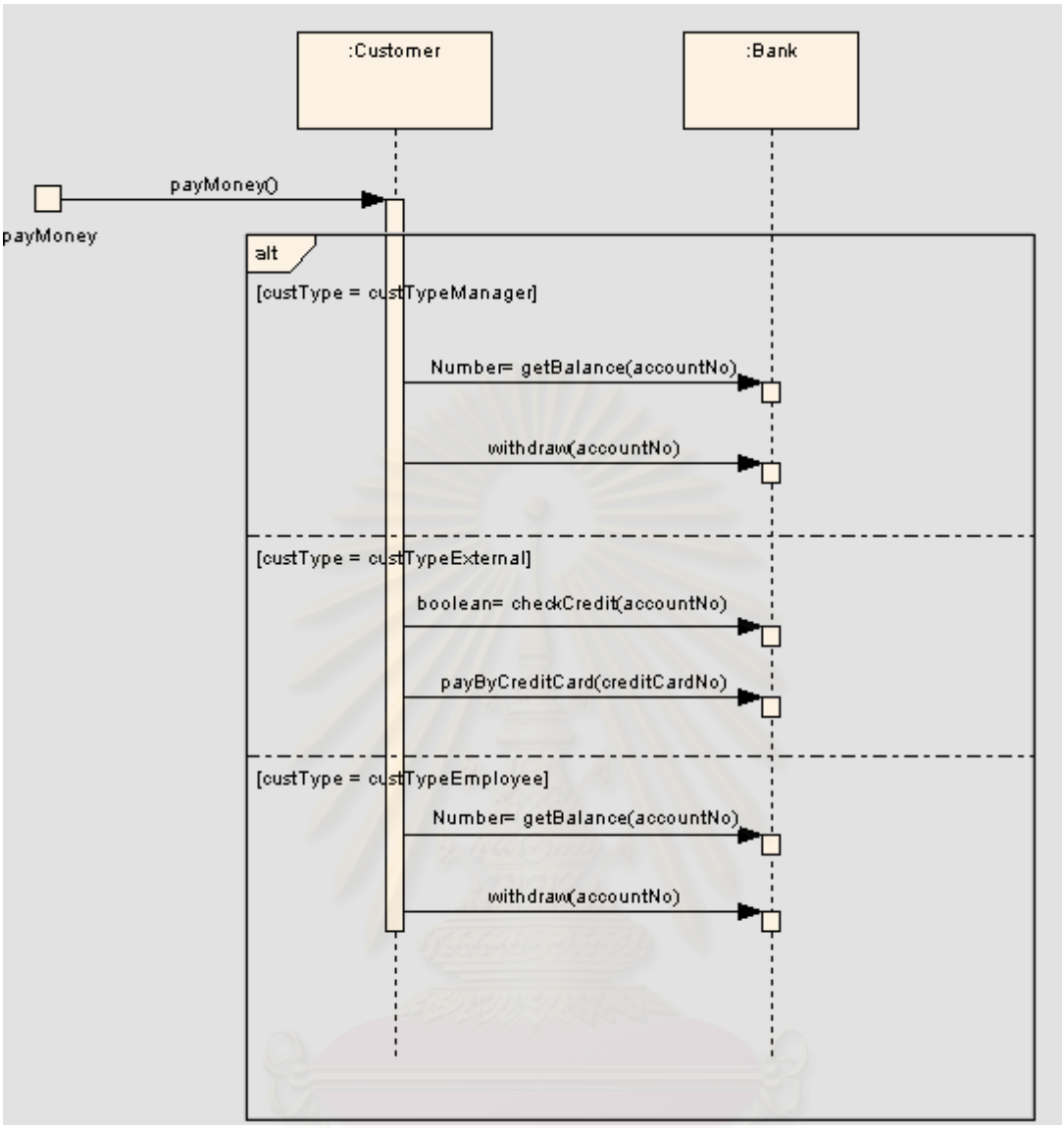
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 1.402
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.952
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 1.004
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.791

รูปที่ 5.63, 5.64 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Switch Statements ตามลำดับ



รูปที่ 5.63 แผนภาพคลาสของระบบตัวอย่างที่ 41 ที่มี Switch Statements

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



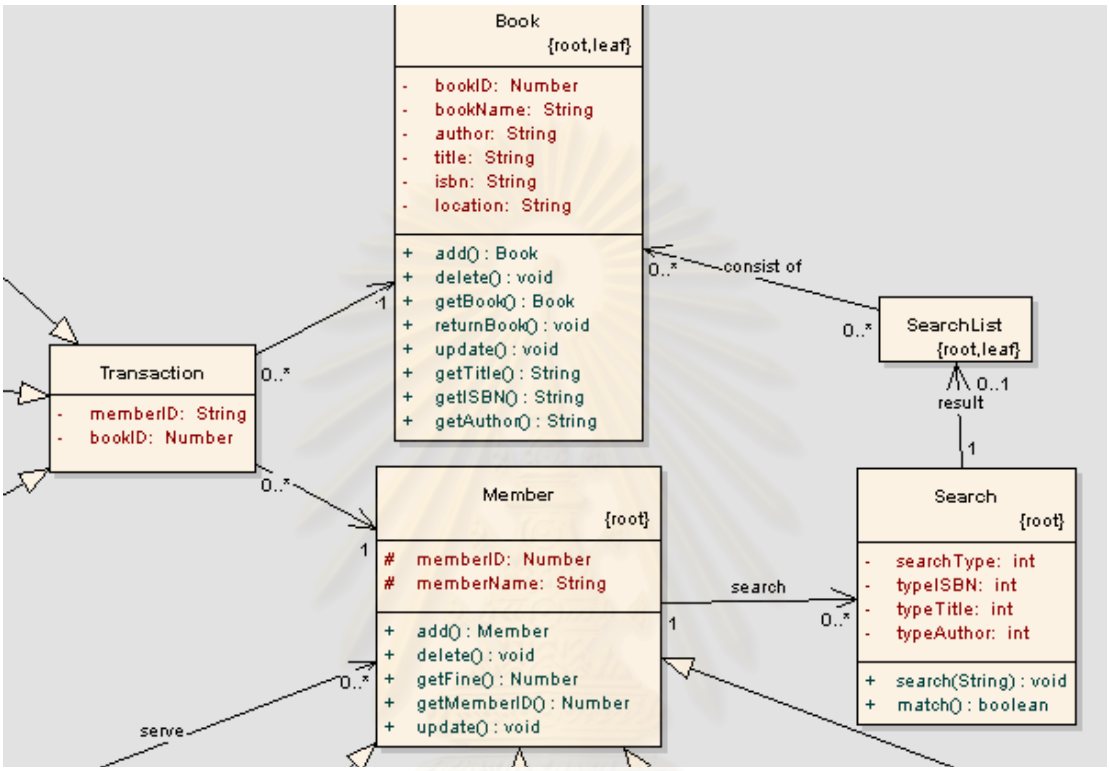
รูปที่ 5.64 แผนภาพซีเควนซ์ของระบบตัวอย่างที่ 41 ที่มี Switch Statements

5.6.1.2 ระบบตัวอย่างที่ 42 ให้ค่ามาตรวัด NOSS = 1 ซึ่งอยู่ในช่วงที่บอกถึงข้อบกพร่องตามกลยุทธ์การตรวจจับข้อบกพร่องแบบ Switch Statements และผลการคำนวณค่ามาตรวัดความสามารถการเปลี่ยนแปลงซอฟต์แวร์ก่อนการรีแฟคทอริงมีค่าดังนี้

- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 1.402 – 0.114 NAggH มีค่าเท่ากับ 1.402
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH มีค่าเท่ากับ 0.912
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.004 – 0.116 * NAggH มีค่าเท่ากับ 1.004

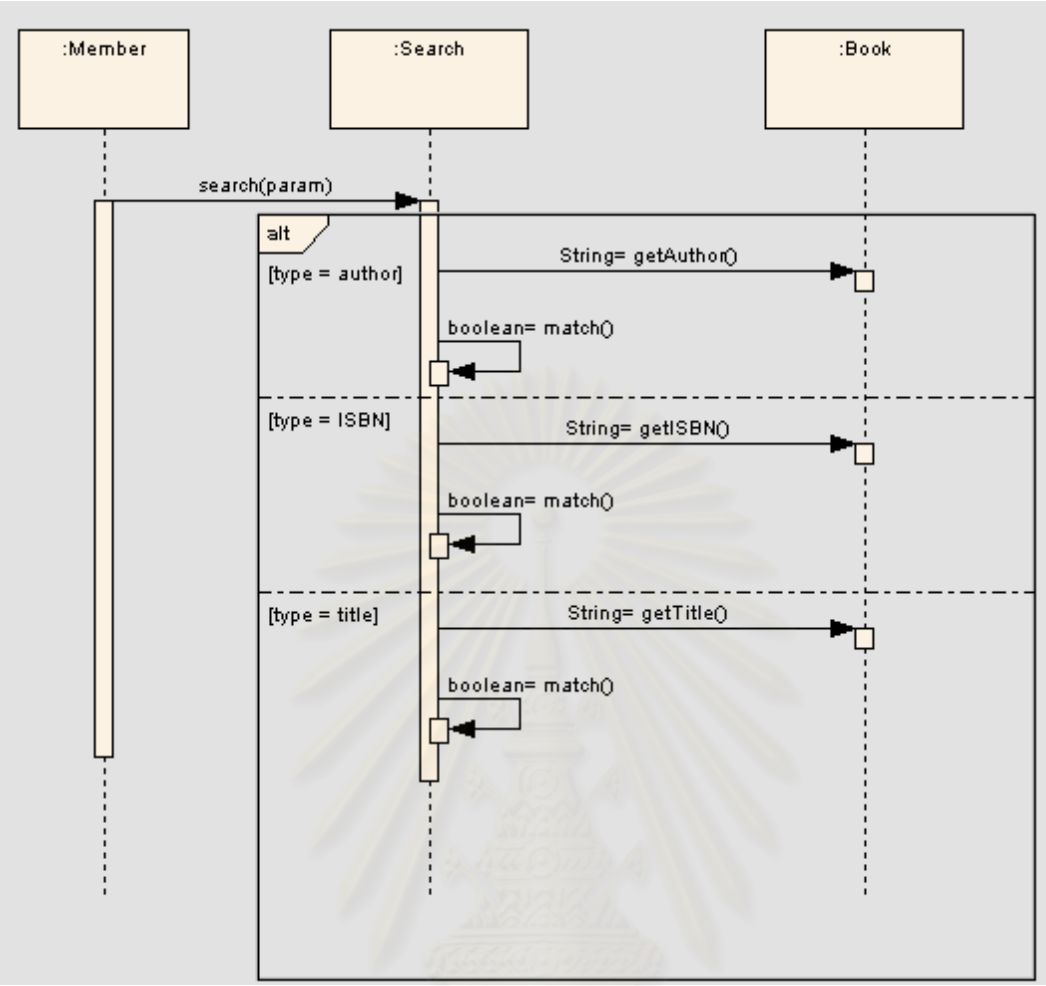
- สำหรับความสามารถการเปลี่ยนแปลงที่วัดจากโมเดล ความสมบูรณ์ของการเปลี่ยนแปลง = 1.079– 0.009 * NA มีค่าเท่ากับ 0.827

รูปที่ 5.65, 5.66 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบในส่วนที่เกิด Switch Statements ตามลำดับ



รูปที่ 5.65 แผนภาพคลาสของระบบตัวอย่างที่ 42 ที่มี Switch Statements

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

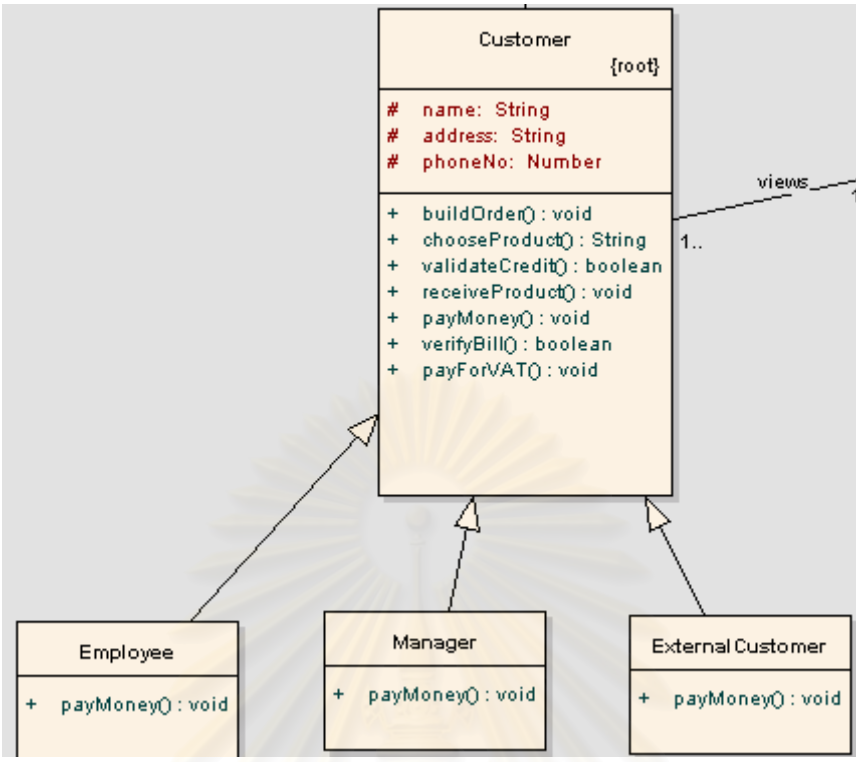


รูปที่ 5.66 แผนภาพซีเควนซ์ของระบบตัวอย่างที่ 42 ที่มี Switch Statements

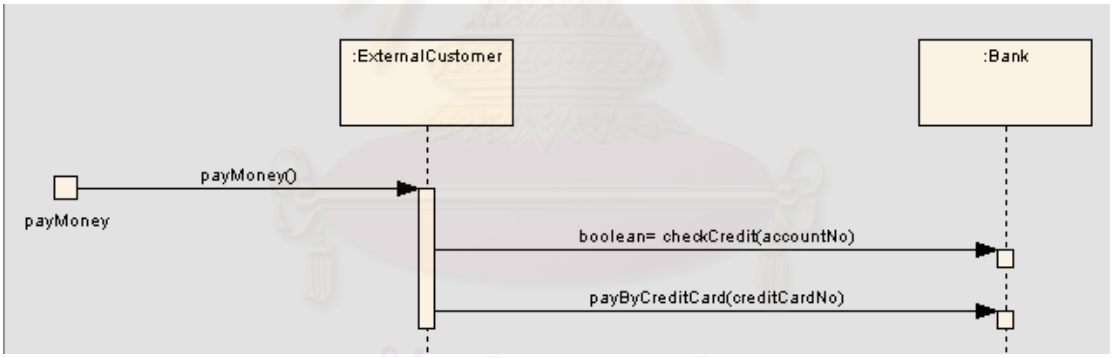
5.6.2 การประยุกต์ใช้วิธีรีแฟคทอริง

5.6.2.1 ระบบตัวอย่างที่ 41 ประยุกต์ใช้ วิธี Extract Method ใช้เพื่อกระจายโค้ดส่วนที่เป็น Switch Statements ในเมทธอด payment ของคลาส Customer ออกมาสร้างเป็นเมทธอดใหม่ และทำการย้ายเมทธอดใหม่นี้ไปยังคลาสที่จะทำโพลิมอฟิซึม หลังจากนั้นจะใช้วิธี Replace Type Code with Subclasses และใช้วิธี Replace Conditional with Polymorphism แทนการใช้ Switch Statements เดิม

รูปที่ 5.67, 5.68 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



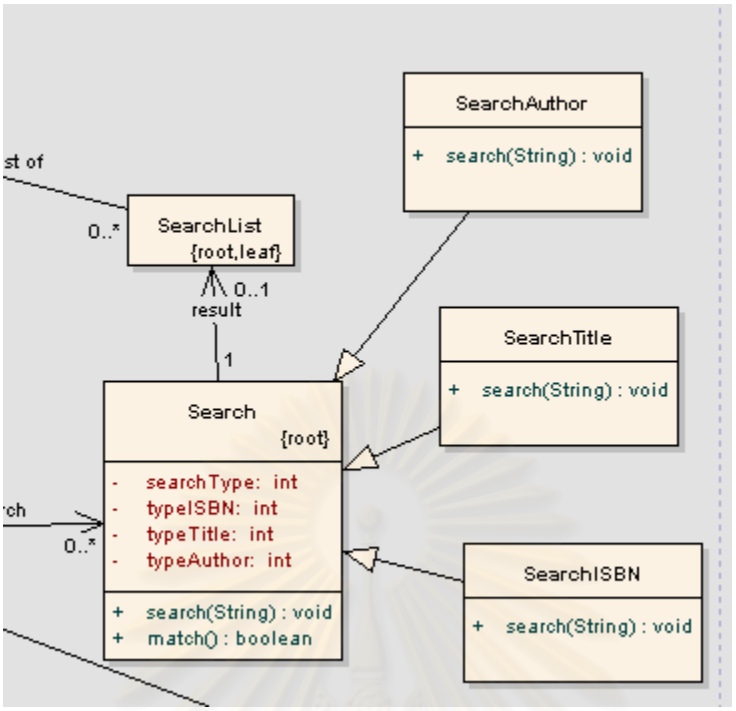
รูปที่ 5.67 แผนภาพคลาสของระบบตัวอย่างหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Customer



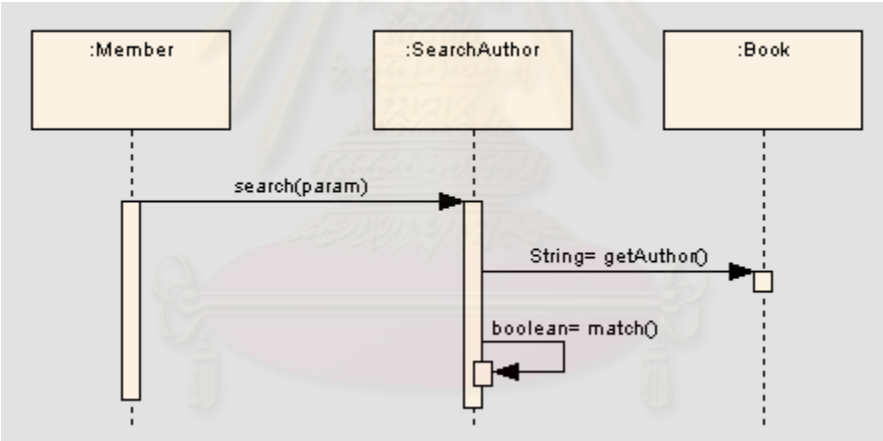
รูปที่ 5.68 แผนภาพซีเควนซ์ของระบบตัวอย่างหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Customer

5.6.2.2 ระบบตัวอย่างที่ 42 ประยุกต์ใช้วิธีการ Extract Method ใช้เพื่อกระจายโค้ดส่วนที่เป็น Switch Statements ในเมทธอด search ของคลาส Search ออกมาสร้างเป็นเมทธอดใหม่ และทำการย้ายเมทธอดใหม่นี้ไปยังคลาสที่จะทำโพลิมอฟิซึม หลังจากนั้นจะใช้วิธี Replace Type Code with Subclasses และใช้วิธี Replace Conditional with Polymorphism แทนการใช้ Switch Statements เดิม

รูปที่ 5.69, 5.70 แผนภาพคลาสและแผนภาพซีเควนซ์ของระบบหลังประยุกต์ใช้วิธีรีแฟคทอริง



รูปที่ 5.69 แผนภาพคลาสของระบบตัวอย่างหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Search



รูปที่ 5.70 แผนภาพซีเควนซ์ของระบบตัวอย่างหลังประยุกต์ใช้วิธีรีแฟคทอริงกับคลาส Search

5.6.3 การประเมินความสามารถของมาตรวัดข้อบกพร่องของโมเดลการออกแบบ และการตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลง

ผู้วิจัยได้ทำการวัดค่าของมาตรวัดสำหรับตรวจจับข้อบกพร่องที่ระบบมีลักษณะเป็นข้อบกพร่องประเภท Switch Statements อีกครั้งหนึ่ง ได้ผลการวัดค่ามาตรวัดเป็นดังนี้

ในระบบตัวอย่างที่ 41 ให้ค่ามาตรวัด NOSS = 0 ไม่ก่อให้เกิดข้อบกพร่องประเภท Switch Statements แล้ว หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการ

ทำรีแฟคทอริงเป็นดังตาราง 5.13 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าที่แย่ลงสำหรับโมเดลความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH ตาราง 5.13 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 41

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	1.402	0.992	1.004	0.836
หลัง	1.402	0.952	1.004	0.836

ในระบบตัวอย่างที่ 42 ให้ค่ามาตรวัด NOSS = 0 ไม่ก่อให้เกิดข้อบกพร่องประเภท Switch Statements แล้ว หลังจากนั้นผู้วิจัยได้ตรวจสอบค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบ และได้เปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงเป็นดังตาราง 5.14 ซึ่งจะเห็นว่าค่ามาตรวัดความสามารถการเปลี่ยนแปลงของระบบมีค่าที่แย่ลงสำหรับโมเดลความถูกต้องของการเปลี่ยนแปลง = 0.992 – 0.040 NGenH ตาราง 5.14 แสดงการเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการทำรีแฟคทอริงสำหรับระบบตัวอย่างที่ 42

	Modifiability Correctness		Modifiability Completeness	
	1.402 – 0.114 *NAggH	0.992 – 0.040 *NGenH	1.004 – 0.116*NAggH	1.079 – 0.009 * NA
ก่อน	1.402	0.912	1.004	0.827
หลัง	1.402	0.872	1.004	0.827

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

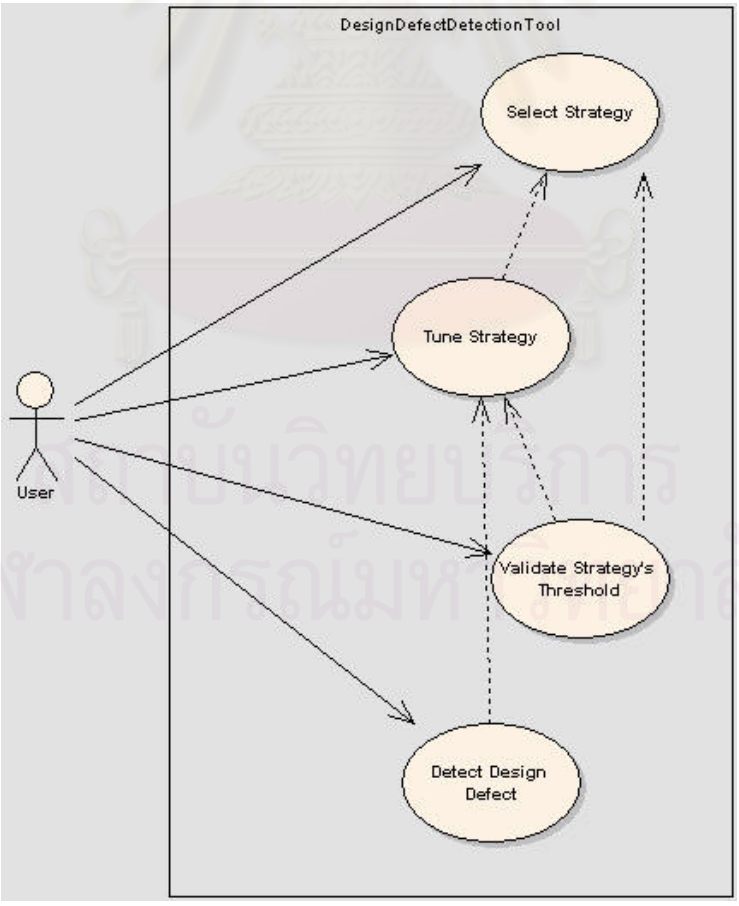
บทที่ 6

การออกแบบและพัฒนาเครื่องมือคำนวณหาช่วงของมาตรวัด และตรวจจับข้อบกพร่อง

ในบทนี้จะกล่าวถึงการออกแบบและพัฒนาเครื่องมือซึ่งใช้ช่วยคำนวณหาช่วงของมาตรวัดและใช้ในการตรวจจับข้อบกพร่องของโมเดลการออกแบบ ซึ่งจะอธิบายด้วยแผนภาพยูสเคส แผนภาพคลาส และแผนภาพแอคทิวิตี้

6.1 การออกแบบและพัฒนาเครื่องมือคำนวณหาช่วงของมาตรวัด และตรวจจับข้อบกพร่อง

6.1.1 ยูสเคสของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ



รูปที่ 6.1 แผนยูสเคส ของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ

ยูสเคสต่างๆ ของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบที่แสดงไว้ในรูปที่ 6.1 มีลำดับการของเหตุการณ์ดังนี้

6.1.1.1 ยูสเคส Select Strategy:

- ยูสเคสนี้เริ่มเมื่อผู้ใช้ (User) เลือกกลยุทธ์
- ระบบทำการจดจำชนิดของกลยุทธ์ที่ผู้ใช้เลือก

6.1.1.2 ยูสเคส Tune Strategy:

- ยูสเคสนี้เริ่มเมื่อผู้ใช้ออกคำสั่งให้ระบบทำการหาค่าช่วงของมาตรวัด
- ระบบทำการหาค่าช่วงของมาตรวัด
- ระบบแสดงค่าช่วงของมาตรวัดที่ใช้ได้จริงที่หาได้

6.1.1.3 ยูสเคส Validate Strategy's Threshold:

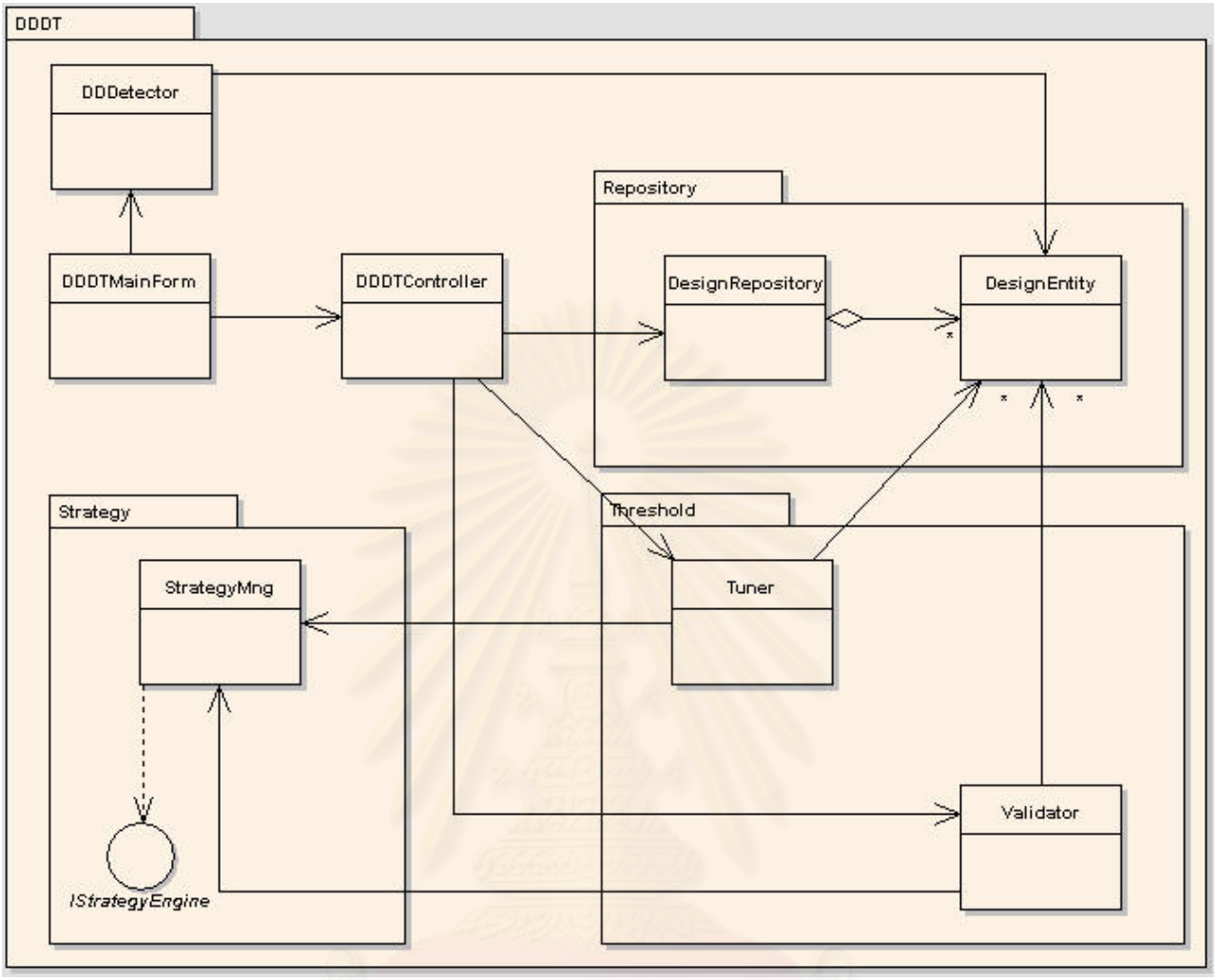
- ยูสเคสนี้เริ่มเมื่อผู้ใช้ออกคำสั่งให้ระบบทำการตรวจสอบค่าช่วงของมาตรวัดที่หามาได้
- ระบบทำการตรวจสอบค่าช่วงของมาตรวัดกับกลยุทธ์ที่สัมพันธ์กัน
- ระบบแสดงค่าฟอลส์เนกาทีฟและฟอลส์โพสิทีฟที่คำนวณออกมาได้

6.1.1.4 ยูสเคส Detect Design Defect:

- ยูสเคสนี้เริ่มเมื่อผู้ใช้ออกคำสั่งให้ระบบทำการตรวจสอบข้อบกพร่องกับเอนทิตีโมเดลการออกแบบโดยใช้ค่าช่วงของมาตรวัดของกลยุทธ์ที่หามาได้
- ให้ระบบทำการตรวจสอบข้อบกพร่องกับเอนทิตีโมเดลการออกแบบโดยใช้ค่าช่วงของมาตรวัดของกลยุทธ์ที่หามาได้
- ระบบทำการจัดเก็บผลลัพธ์ที่ได้จากการตรวจสอบข้อบกพร่องของเอนทิตีโมลเดลการออกแบบในรูปแบบของไฟล์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6.1.2 สถาปัตยกรรมของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ



รูปที่ 6.2 แผนภาพคลาสแสดงสถาปัตยกรรม (Architecture) ของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ

จากรูปที่ 6.2 เครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบจะถูกแบ่งออกเป็น 3 แพคเกจ (Package) หลัก คือ

6.1.2.1 แพคเกจ Repository: แพคเกจนี้รับผิดชอบเกี่ยวกับการจัดเก็บตัวอย่างเอนทิตีโมเดลการออกแบบทั้งหมดที่ระบบจะใช้เป็นกลุ่มตัวอย่างทดลอง

- คลาส DesignRepository: รับผิดชอบการดึงไฟล์ของโมเดลการออกแบบทั้งหมดที่เก็บไว้บนพื้นที่ที่กำหนดบนฮาร์ดดิสก์ (Hard disk) มาเก็บไว้เป็นโมเดลการออกแบบบนหน่วยความจำ (Memory) รวมไปถึงการจัดสร้างเซต (Set) ของเอนทิตีการออกแบบเพื่อที่ถูกนำไปใช้เป็นกลุ่มตัวอย่างเพื่อใช้ในการหาค่าช่วงของมาตรวัดของกลยุทธ์หนึ่งๆด้วย

- คลาส DesignEntity: ออบเจกต์ของคลาสนี้คือเอนทิตี้การออกแบบที่ใช้เป็นตัวอย่างทดลองกับมาตรวัดต่างๆ

6.1.2.2 แพคเกจ Threshold: แพคเกจนี้รับผิดชอบเกี่ยวกับการหาและตรวจสอบค่าช่วงของมาตรวัดของแต่ละกลยุทธ์

- คลาส Tuner: คลาสนี้ทำหน้าที่ปรับปรุงค่าช่วงของมาตรวัดของกลยุทธ์หนึ่งๆจากค่าเริ่มต้นที่เป็นค่าสมมุติให้กลายเป็นค่าที่ใช้ได้จริงในที่สุด โดยการปรับปรุงค่าช่วงเหล่านี้จะใช้อัลกอริทึมฟาสต์ซีมูเลทเต็ดแอนนิลลิง
- คลาส Validator: คลาสนี้รับผิดชอบในการตรวจฟอสส์เนกาทีฟ และ ฟอลส์โพสิทีฟของค่าช่วงของมาตรวัดของแต่กลยุทธ์หนึ่งๆที่ได้ผ่านการปรับปรุงแล้ว

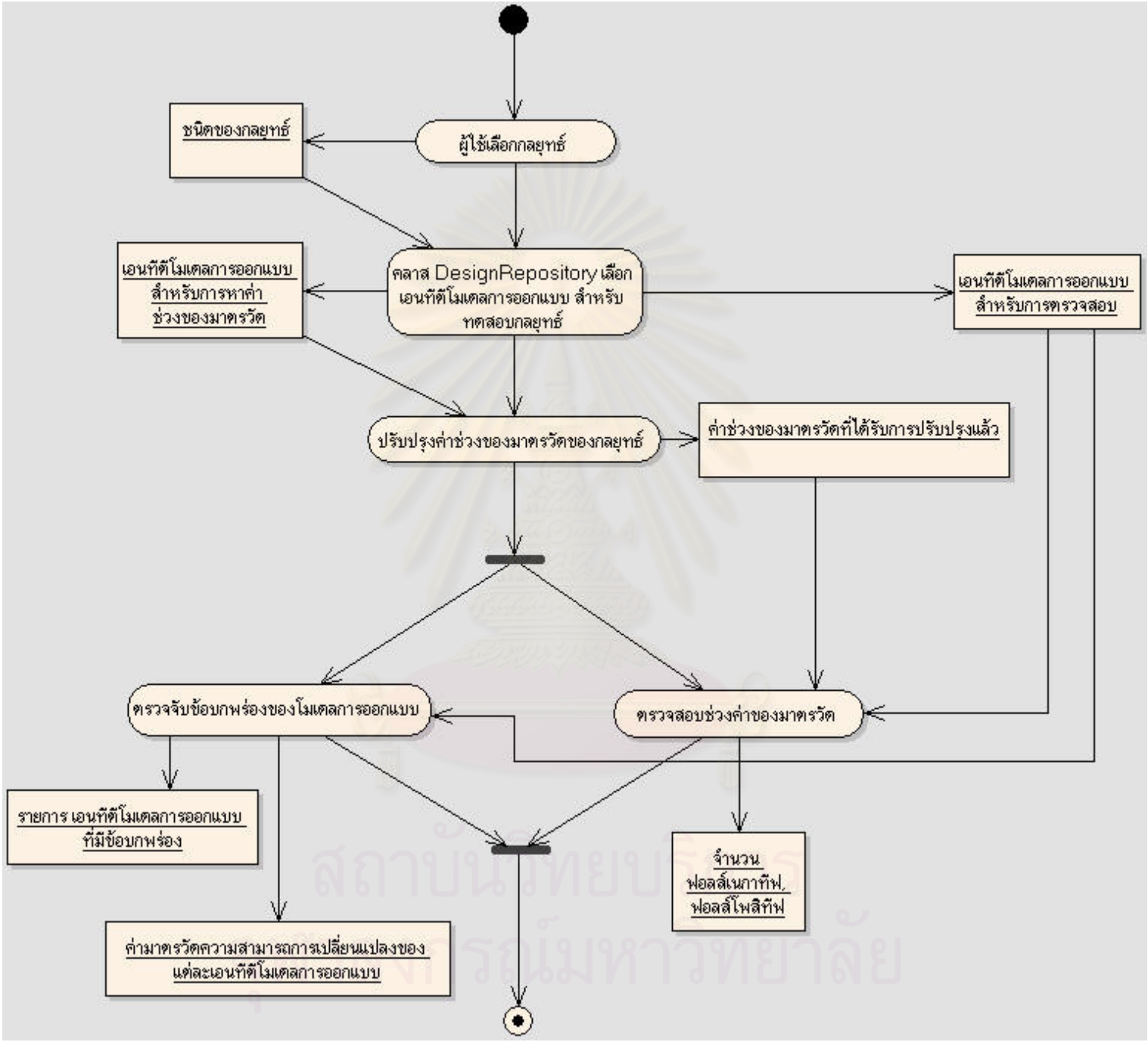
6.1.2.3 แพคเกจ Strategy: แพคเกจนี้รับผิดชอบเกี่ยวกับการเรียกใช้กลยุทธ์ต่างๆซึ่งเป็นปลักอิน (Plug-in) ของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบนี้

- คลาส StategyMng: คลาสนี้รับผิดชอบการดึงปลักอินของกลยุทธ์ต่างๆเข้าสู่หน่วยความจำเพื่อเรียกใช้
- อินเทอร์เฟส (Interface) IStrategyEngine: เป็นอินเทอร์เฟสกลางที่ปลักอินกลยุทธ์ต่างๆจะต้องใช้เพื่อที่เครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบจะสามารถเรียกใช้ได้อย่างถูกต้อง โดยปลักอินกลยุทธ์ที่ใช้ในการทดลองมีด้วยกัน 6 ตัวด้วยกัน ดังนี้
    - ปลักอิน DataClassStrategyEngine
    - ปลักอิน FeatureEnvyStrategyEngine
    - ปลักอิน MessageChainsStrategyEngine
    - ปลักอิน MiddleMaStrategyEngine
    - ปลักอิน GodClassStrategyEngine
    - ปลักอิน SwitchStatementsStrategyEngine

    โดยที่ปลักอินแต่ละตัวจะถูกตั้งชื่อให้สื่อถึงกลยุทธ์ที่ปลักอินนั้นๆเกี่ยวข้อง

    สถาปัตยกรรมของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ

6.1.3 กิจกรรมภายในของเครื่องมือตรวจจับข้อบกพร่องของโมเดลการออกแบบ



รูปที่ 6.3 แผนภาพแอคทิวีตี้แสดงกิจกรรมภายในของเครื่องมือตรวจจับข้อบกพร่อง

กิจกรรมหลักภายในของเครื่องมือตรวจจับข้อบกพร่องมีดังนี้

6.1.3.1 แอคทิวิตี้ ผู้ใช้เลือกกลยุทธ์: ระบบทำการจดจำชนิดของกลยุทธ์ที่ผู้ใช้ได้ทำการเลือก

6.1.3.2 แอคทิวิตี้ คลาส DesignRepository เลือก เอนทีตีโมเดลการออกแบบ สำหรับทดสอบกลยุทธ์: คลาส DesignRepository รับเอาชนิดของกลยุทธ์เข้ามาจากนั้นทำการดึงชุดของเอนทีตีโมเดลการออกแบบตัวอย่างที่เหมาะสมกับกลยุทธ์นั้นออกมาได้ผลลัพธ์เป็นเซ็ตของเอนทีตีโมเดลการออกแบบ 2 ชุด คือ เซ็ตของเอนทีตีโมเดลการออกแบบ สำหรับการหาค่าช่วงของมาตรวัด และ เซ็ตของเอนทีตีโมเดลการออกแบบ สำหรับการตรวจสอบ

6.1.3.3 แอคทิวิตี้ ปรับปรุงค่าช่วงของมาตรวัดของกลยุทธ์: ระบบรับเอา เซ็ตของเอนทีตีโมเดลการออกแบบ สำหรับการหาค่าช่วงของมาตรวัด ที่ได้จากขั้นตอนที่แล้วมาเป็นอินพุทของการหาค่าช่วงของมาตรวัดของกลยุทธ์ ในขั้นตอนนี้ระบบจะทำการปรับปรุงค่าช่วงของมาตรวัดสำหรับกลยุทธ์ที่ผู้ใช้เลือกจากค่าช่วงเริ่มต้นที่ถูกสมมุติขึ้นไปเป็นค่าช่วงมาตรวัดที่ปรับปรุงแล้วที่สามารถใช้ได้จริง

6.1.3.4 แอคทิวิตี้ ตรวจจับข้อบกพร่องของโมเดลการออกแบบ: ในขั้นตอนการทำงานนี้ระบบจะต้องได้ค่าช่วงมาตรวัดที่ใช้ได้จริงมาแล้ว ซึ่งระบบจะใช้ค่านี้เป็นอินพุทร่วมกันกับเซ็ตของเอนทีตีโมเดลการออกแบบ สำหรับการตรวจสอบเพื่อหาค่าฟอลส์เนกาทีฟและฟอลส์โพสิทีฟ ของค่าช่วงมาตรวัดบนเซ็ตของเอนทีตีโมเดลการออกแบบ สำหรับการตรวจสอบ

6.1.3.5 แอคทิวิตี้ ตรวจสอบช่วงค่าของมาตรวัด: ขั้นตอนนี้เป็นขั้นตอนการใช้งานจริงของค่าช่วงมาตรวัดที่ได้จากระบบ โดยระบบสามารถใช้ค่าช่วงมาตรวัดนี้ในการระบุว่าเอนทีตีโมเดลการออกแบบใดมีข้อบกพร่องหรือไม่ ได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 7

บทสรุปและข้อเสนอแนะ

7.1 บทสรุป

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการตรวจจับข้อบกพร่องของโมเดลการออกแบบหรือร่องรอยที่ไม่ดี ที่เกิดขึ้นในขั้นตอนการออกแบบ 6 ประเภทคือ Data Class, Feature Envy, God Class, Message Chains, Middle Man และ Switch Statements โดยพิจารณาที่แผนภาพคลาสและแผนภาพซีเควนซ์ว่ามีข้อบกพร่องหรือร่องรอยที่ไม่ดีใดบ้าง ที่ส่งผลกระทบต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ โดยออกแบบวิธีการตรวจจับข้อบกพร่องหรือร่องรอยที่ไม่ดีด้วยการกำหนดกลยุทธ์เพื่อตรวจจับข้อบกพร่องหรือร่องรอยที่ไม่ดีของโมเดลการออกแบบด้วยมาตรวัดซอฟต์แวร์เชิงวัตถุสำหรับขั้นตอนการออกแบบ

นอกจากนี้ได้พัฒนาเครื่องมือสำหรับหาช่วงค่าของมาตรวัดที่เหมาะสมที่สุดสำหรับกลยุทธ์การตรวจจับด้วยวิธีซีมูเลทเต็ด แอนนิลลิงโดยใช้กลุ่มระบบตัวอย่าง 35 ระบบต่อข้อบกพร่องของโมเดลการออกแบบหนึ่งประเภท และพัฒนาเครื่องมือสำหรับตรวจจับข้อบกพร่องหรือร่องรอยที่ไม่ดี และเสนอวิธีรีแฟคทอริง สำหรับข้อบกพร่องหรือร่องรอยที่ไม่ดีแต่ละประเภท วิทยานิพนธ์ฉบับนี้ประเมินผลกระทบของการเกิดข้อบกพร่องต่าง ๆ ต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ด้วยระบบทดสอบ 5 ระบบต่อข้อบกพร่องของโมเดลการออกแบบหนึ่งประเภท โดยการตรวจจับข้อบกพร่องของโมเดลการออกแบบแต่ละประเภท และเปรียบเทียบค่ามาตรวัดความสามารถในการเปลี่ยนแปลงก่อนและหลังการประยุกต์ใช้วิธีรีแฟคทอริงสำหรับข้อบกพร่องของโมเดลการออกแบบแต่ละประเภท

ผลการทดลองสรุปว่าข้อบกพร่องประเภท Data Class, God Class, และMiddle Man ส่งผลต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ โดยหากมีการประยุกต์ใช้วิธีรีแฟคทอริงที่ถูกต้องเพื่อแก้ไขข้อบกพร่องประเภทดังกล่าวแล้ว สามารถทำให้ค่าของมาตรวัดความสามารถของการเปลี่ยนแปลงดีขึ้น สำหรับการรีแฟคทอริงเพื่อแก้ไขข้อบกพร่องประเภท Feature Envy และ Message Chains ไม่ส่งผลให้ค่าของมาตรวัดความสามารถของการเปลี่ยนแปลงดีขึ้น ส่วนการรีแฟคทอริงเพื่อแก้ไขข้อบกพร่องประเภท Switch Statements ทำให้ค่าของมาตรวัดความสามารถของการเปลี่ยนแปลงลดลง

ในงานวิจัยนี้ได้มีการกำหนดมาตรวัดโดยเลือกมาตรวัดเชิงวัตถุสำหรับโมเดลการออกแบบที่มีผู้เสนออยู่แล้วมาประยุกต์ใช้ ได้แก่มาตรวัด RFC, มาตรวัด COH, มาตรวัด NCPA และมีการเสนอมาตรวัดใหม่ได้แก่ NNorm, NSMsg, NMsg, SOMsgCh, และมาตรวัด WOD มาใช้ในการตรวจจับข้อบกพร่องของโมเดลการออกแบบ

7.2 ข้อจำกัดและข้อเสนอแนะ

1. ผลของการคำนวณมาตรวัดข้อบกพร่องประเภท Middle Man อาจระบุว่าคลาสที่เป็นยูสเซอร์อินเตอร์เฟสมีค่ามาตรวัดอยู่ในช่วงที่เป็นข้อบกพร่อง ซึ่งหากพิจารณาแล้ว ไม่ถือว่าลักษณะการทำงานของคลาสที่เป็นยูสเซอร์อินเตอร์เฟสมีข้อบกพร่องประเภท Middle Man เนื่องจากคลาสที่เป็นยูสเซอร์อินเตอร์เฟสมีความจำเป็นต้องมอบหมายงานให้กับคลาสอื่น

2. การรีแฟคทอริงด้วยวิธี Move Method มีข้อควรระวังที่ต้องพิจารณาด้วยว่าเมทธอดมีการเรียกใช้คุณลักษณะของคลาสเดิมหรือไม่ ซึ่งหากมีการเรียกใช้คุณลักษณะของคลาสเดิม เมื่อย้ายเมทธอดแล้ว การเรียกใช้คุณลักษณะจะต้องสามารถทำได้เหมือนเดิม

3. การรีแฟคทอริงสำหรับข้อบกพร่องบางประเภทอาจทำให้เกิดปัญหาอื่นตามมา เช่นการรีแฟคทอริงเพื่อแก้ไขข้อบกพร่องประเภท Message Chains อาจทำให้เกิดข้อบกพร่องประเภท Middle Man ดังนั้นในการแก้ไขข้อบกพร่องควรพิจารณาลักษณะของข้อบกพร่อง และเปรียบเทียบข้อดีและข้อเสียก่อนทำรีแฟคทอริง

4. จากผลการทดลองสำหรับการรีแฟคทอริงระบบซึ่งไม่ทำให้ความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ดีขึ้นนั้น อาจใช้โมเดลความสามารถของการเปลี่ยนแปลงซอฟต์แวร์โมเดลอื่นเพื่อมาทดสอบการตรวจจับข้อบกพร่องและการรีแฟคทอริงเพิ่มเติม หรือเพิ่มกลุ่มตัวอย่างที่ใช้ทดสอบการตรวจจับข้อบกพร่องให้มากขึ้น

5. จากโมเดลความสามารถของการเปลี่ยนแปลงที่ใช้ในงานวิจัยนี้จะเห็นว่า องค์ประกอบของโมเดลการออกแบบที่ส่งผลต่อความสามารถของการเปลี่ยนแปลงซอฟต์แวร์มีเพียงจำนวนโครงสร้างลำดับชั้นของความสัมพันธ์แบบแอกกริเกชัน จำนวนโครงสร้างลำดับชั้นของความสัมพันธ์แบบเจนเนอรัลไลเซชันและจำนวนคุณลักษณะทั้งหมดของระบบเท่านั้น หากการประยุกต์ใช้วิธีรีแฟคทอริงไม่ส่งผลต่อองค์ประกอบเหล่านี้ การวัดค่าความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ก่อนและหลังการรีแฟคทอริงด้วยโมเดลที่เลือกใช้ในงานวิจัย อาจไม่แสดงให้เห็นถึงความสามารถในการเปลี่ยนแปลงที่ดีขึ้น แต่วิธีการที่ใช้ในงานวิจัยนี้สามารถนำไปใช้เป็นเครื่องมือพื้นฐานสำหรับการหาหลักการปรับปรุงความสามารถของการเปลี่ยนแปลงซอฟต์แวร์ต่อไปได้

6. การตรวจจับข้อบกพร่องจะช่วยแนะนำว่าควรแก้ไขโมเดลการออกแบบที่ใด และแก้ไขด้วยวิธีรีแฟคทอริงแบบใด โดยผลของการปรับปรุงความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ จะขึ้นอยู่กับการเลือกใช้วิธีการรีแฟคทอริงที่เหมาะสม

7. งานวิจัยนี้นำเสนอวิธีการตรวจจับข้อบกพร่องของโมเดลการออกแบบเพื่อปรับปรุงคุณภาพของซอฟต์แวร์ในด้านความสามารถในการเปลี่ยนแปลงซอฟต์แวร์เชิงวัตถุ หากสามารถนำวิธีการตรวจจับข้อบกพร่องไปประยุกต์ใช้กับคุณภาพของซอฟต์แวร์ในด้านอื่น ๆ จะทำให้คุณภาพของการออกแบบและพัฒนาซอฟต์แวร์ได้รับการปรับปรุงให้ดียิ่งขึ้น

8. งานวิจัยนี้พัฒนาเครื่องมือตรวจจับข้อบกพร่องและเสนอแนะวิธีการรีแฟคทอริงสำหรับข้อบกพร่องของโมเดลการออกแบบหรือร่องรอยที่ไม่ดีที่กระทบต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์ โดยการเลือกใช้วิธีรีแฟคทอริงยังขึ้นอยู่กับการตัดสินใจของผู้ใช้งาน หากสามารถพัฒนาเครื่องมือที่ช่วยในการประยุกต์วิธีรีแฟคทอริงได้อย่างอัตโนมัติจะทำให้เครื่องมือนี้สามารถนำไปใช้งานได้อย่างสมบูรณ์มากขึ้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] B. P. Lientz and E. B. Swanson. Software Maintenance Management. Addison Wesley, 1980.

[2] B.W. Boehm, J. R. Brown, and M. Lipow. Quantitative Evaluation of Software Quality. Proceedings of the 2nd international conference on Software engineering (2nd).:592 – 605, 1976.

[3] Alan Dennis, Barbara Haley Wixom and David Tegarden. Systems Analysis and Design with UML Version 2.0 An Object-Oriented Approach. John Wiley & Sons, Inc., 2005.

[4] A. Riel. Object-Oriented Design Heuristics. Addison-Wesley, 1996.

[5] Fowler, M. Refactoring: Improving the Design of Existing Code. United States: Addison-Wesley, 1999.

[6] Fenton, N. E. and Pfleeger, S. L. Software Metrics: A Rigorous and Practical Approach. PWS Publishing Company, 1997.

[7] R. Marinescu. Measurement and Quality in Object-Oriented Design. PhD thesis Politehnica University of Timis¸oara, 2002.

[8] Y. Kataoka, T. Imai, H. Andou, and T. Fukaya. A Quantitative Evaluation of Maintainability Enhancement by Refactoring. Proceedings of the International Conference on Software Maintenance (ICSM'02), 2002.

[9] M. Genero, M^{a}E. Manso, M. Piattini, and G. Cantone. Building UML Class Diagram Maintainability Prediction Models Based on Early Metrics. Proceedings IEEE Computer Society, 2003:263-275.

[10] K. Hoffman. Combinatorial and Integer Optimization. Available from: http://iris.gmu.edu/~khoffman/papers/newcomb1.html

[11] S. Kirkpatick, CD Gelatt, and M. P. Vecchi. Optimization by simulated annealing. Science, 220, No. 4598, 1983:671-. 680.

[12] H. Szu and R. Hartley. Fast simulated annealing. Phys. Lett. A, vol.122, 1987:157-162.

[13] T. Pienlert and P. Muenchaisri. Bad-Smell Detection using Object-Oriented Software Metrics. International Conference Computer Science, Software Engineering, Information Technology, e-Business and Application (CSITeA '04), December 27-29, 2004.

[14] R. Marinescu. Detection Strategies: Metrics-Based Rules for Detecting Design Flaws. Proceedings of the International Conference on Software Maintenance (ICSM04), 2004.

[15] Mihancea, P.F.; Marinescu, R. Towards the Optimization of Automatic Detection of Design Flaws in Object-Oriented Software Systems. Software Maintenance and Reengineering, 2005. CSMR 2005. Ninth European Conference on, Vol., Iss., 21-23 March 2005: 92- 101.

[16] S. R. Chidamber and C. F. Kemerer. A metric suite for object-oriented design. IEEE Transactions of Software Engineering, 25(5):476–493, June 1994.

[17] B. Lee, and C. Wu. Genetic Algorithm Based Restructuring of Object-Oriented Designs Using Metrics. IEICE Transactions on Inormation and System Society, vol. E85-D, NO.7, 2002.

[18] M. Kiewkanya, N. Jindasawat, and P. Muenchaisri. A Methodology for Constructing Maintainability Model of Object-Oriented Design. Proceedings of the Quality Software, Fourth International Conference on (QSIC'04), 2004.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

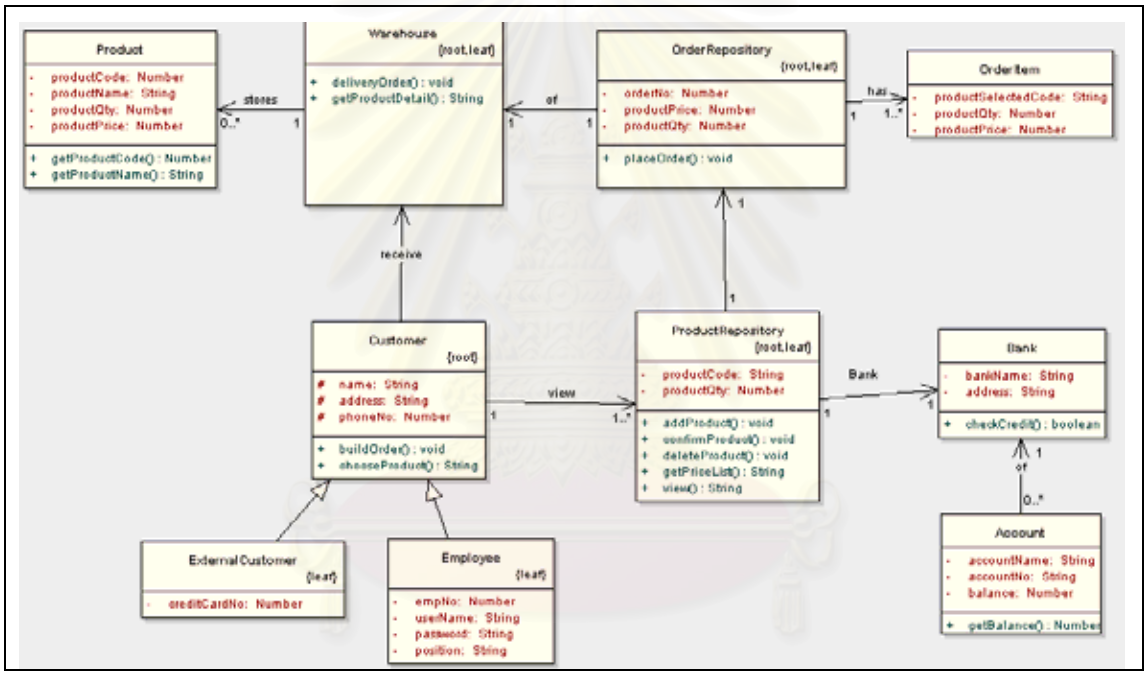
ภาคผนวก ก

ระบบที่ใช้ในงานวิจัย

ในภาคผนวกนี้จะแสดงรายละเอียดเกี่ยวกับระบบตัวอย่างที่ใช้ในงานวิจัย 42 ระบบซึ่งมีรายละเอียดดังนี้

1. ระบบขายสินค้าผ่านเว็บ

ระบบขายสินค้าผ่านเว็บ มีฟังก์ชันการทำงานสำหรับลูกค้า ได้แก่ การเลือกชมสินค้า และการสั่งซื้อสินค้าผ่านบัตรเครดิต หลังจากลูกค้าสั่งรายการซื้อสินค้าเรียบร้อยแล้ว คำสั่งซื้อที่ถูกยืนยันจะถูกส่งไปยัง Warehouse เพื่อจัดการส่งสินค้าให้กับลูกค้าต่อไป

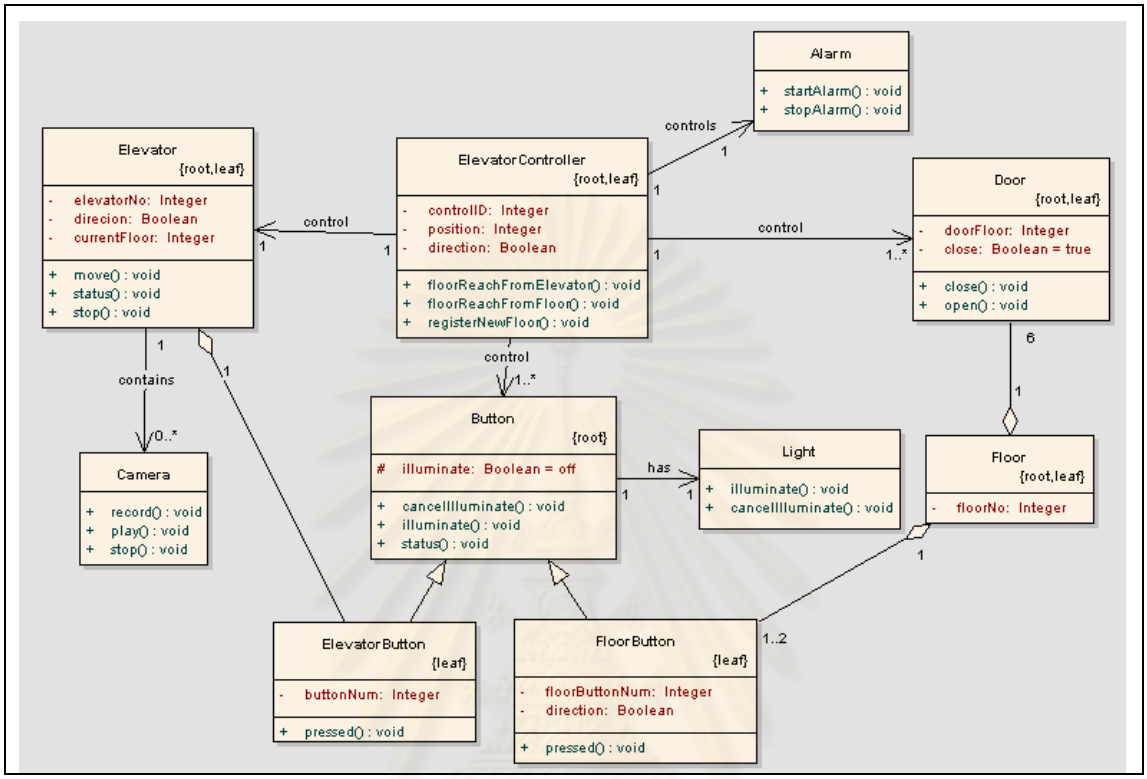


รูปที่ ก.1 แผนภาพคลาสระบบขายสินค้าผ่านเว็บ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2. ระบบลิฟต์

ระบบลิฟต์เป็นระบบที่รองรับความต้องการในการทำงานของลิฟต์ที่ประกอบด้วย การขึ้น-ลงของลิฟต์, การเซ็ตสถานะไฟของปุ่มภายในลิฟต์เมื่อถูกกด และการกดปุ่มเรียกลิฟต์ในแต่ละชั้น

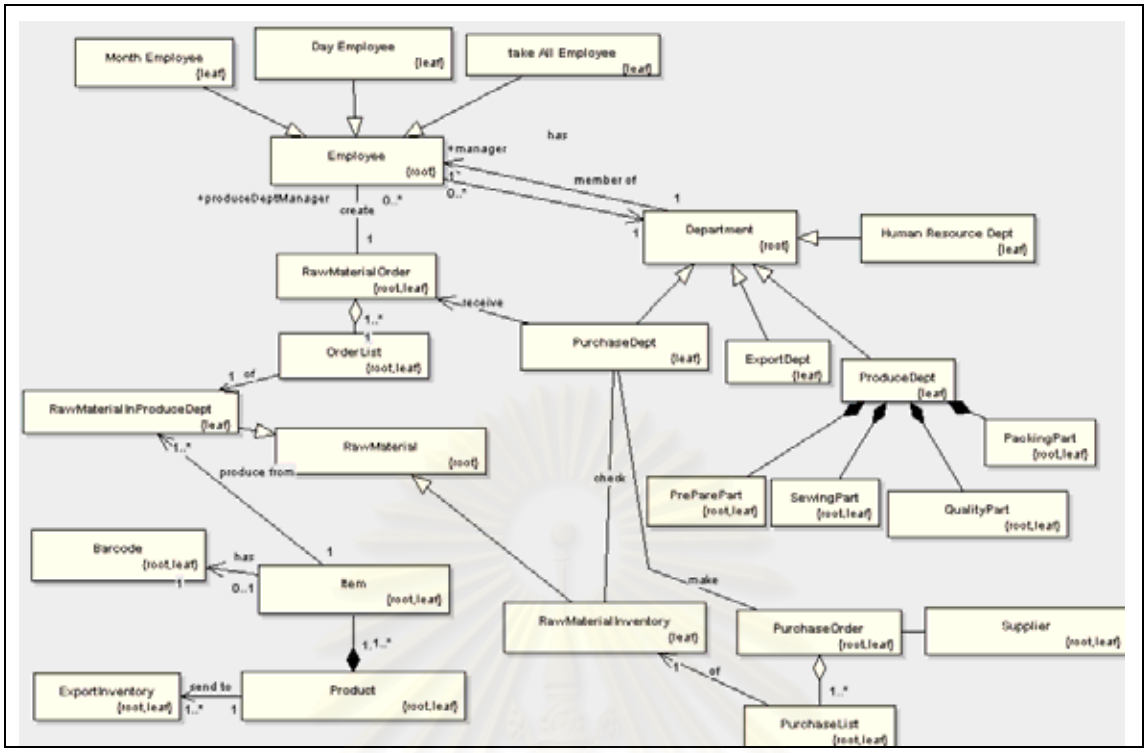


รูปที่ ก.2 แผนภาพคลาสระบบลิฟต์

3. ระบบการผลิตเสื้อผ้าส่งออก

ระบบนี้เป็นระบบที่จัดทำเพื่อช่วยในการทำงานของบริษัทผลิตเสื้อผ้าแห่งหนึ่ง ซึ่งทำการผลิตเสื้อผ้าส่งออกขายไปยังต่างประเทศ การทำงานในบริษัทจะแบ่งออกเป็น 4 แผนก ได้แก่ แผนกจัดซื้อวัตถุดิบ แผนกการผลิต แผนกบริหารทรัพยากรบุคคล และแผนกส่งออก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

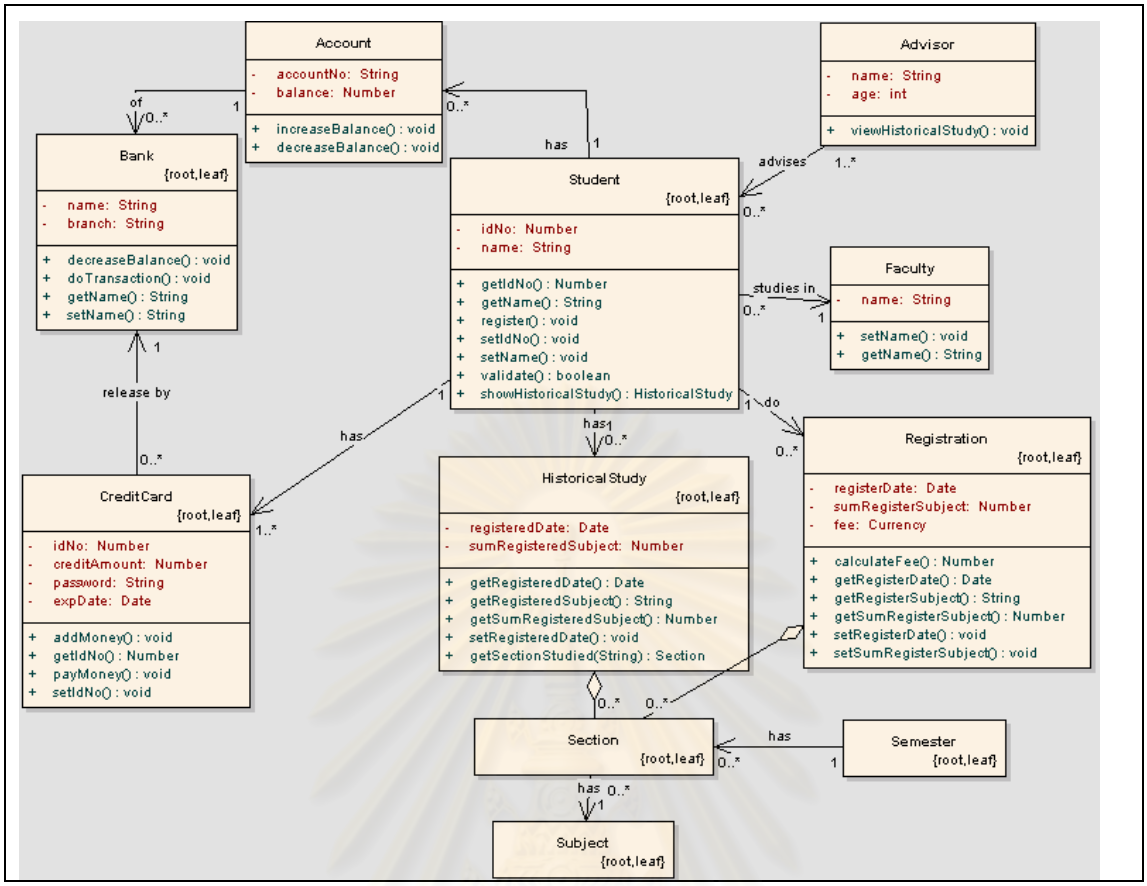


รูปที่ ก.3 แผนภาพคลาสระบบการผลิตเสื้อผ้าส่งออก

4. ระบบการลงทะเบียนเรียนผ่านเว็บ

ระบบการลงทะเบียนผ่านเว็บมีฟังก์ชันในลงทะเบียนเรียน โดยผู้ที่จะเข้าใช้งานระบบได้ต้องผ่านการตรวจสอบผู้เข้าใช้งานก่อน โดยหากเป็นผู้ใช้งานที่ถูกต้องจะสามารถลงทะเบียนได้ และในการลงทะเบียน จะมีการตรวจสอบประวัติการลงทะเบียนว่าจะไม่มีการลงทะเบียนซ้ำ รวมถึงมีฟังก์ชันการติดต่อกับธนาคารเพื่อชำระเงินค่าลงทะเบียน รวมถึงการแสดงรายงานประวัติการลงทะเบียนเรียนด้วย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

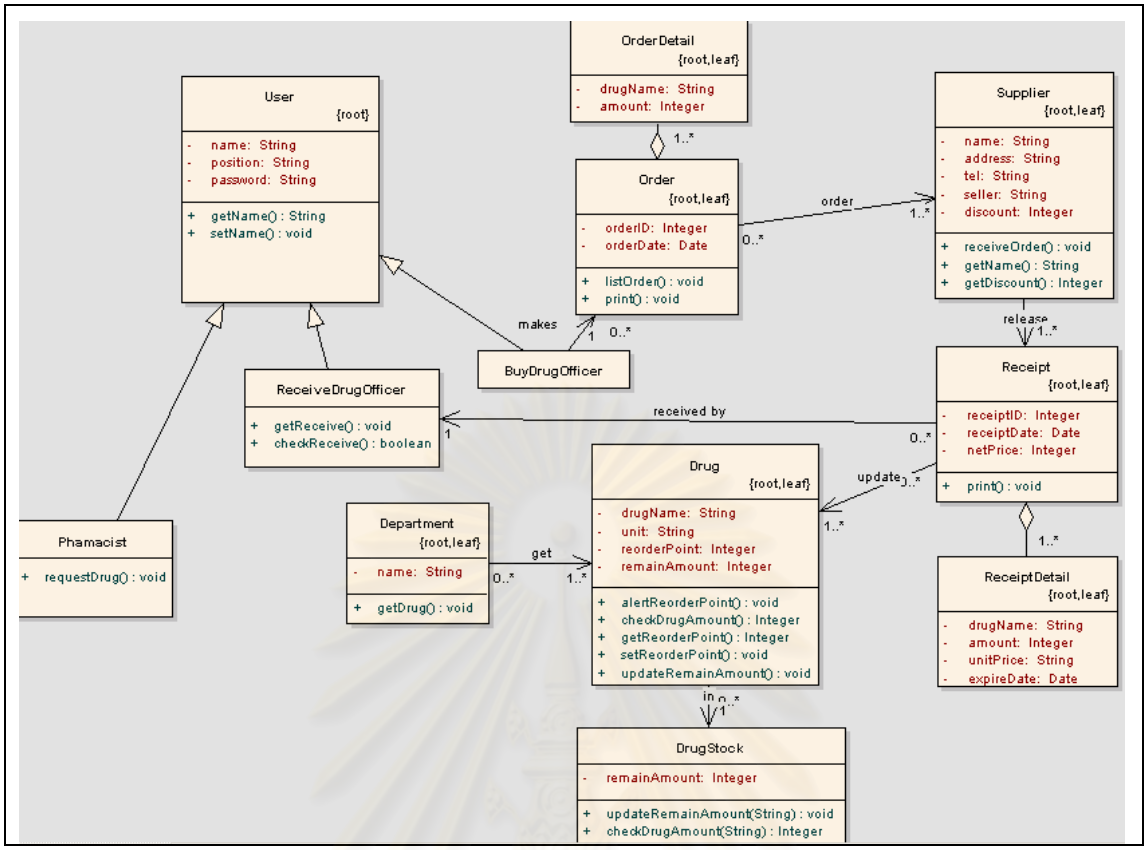


รูปที่ ก.4 แผนภาพคลาสระบบการลงทะเบียนเรียนผ่านเว็บ

5. ระบบสารสนเทศคลังยา

ระบบสารสนเทศคลังยาเป็นระบบที่ใช้เก็บรายการข้อมูลยาประเภทต่างๆ และบันทึกรายการข้อมูลการเบิกจ่าย และซื้อยา โดยจะมีคลังยาเป็นที่เก็บข้อมูลรายการยาว่าปัจจุบันมีจำนวนเท่าใด ในการทำรายการสั่งซื้อจะเกิดขึ้น 2 กรณี คือ กรณีที่รายการยามีจำนวนต่ำกว่าจุดสั่งซื้อ และกรณีที่มีการร้องขอรายการยาจากแผนกอื่นแล้วมีจำนวนไม่เพียงพอ โดยเจ้าพนักงานจะเป็นผู้ออกใบรายการสั่งซื้อไปยังผู้ผลิตยานั้นๆ เมื่อสั่งซื้อยาจากบริษัทขายยาเรียบร้อยแล้วจะนำรายการยาจากใบสั่งซื้อที่บริษัทขายยาออกให้มาแก้ไขข้อมูลในคลังยา

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

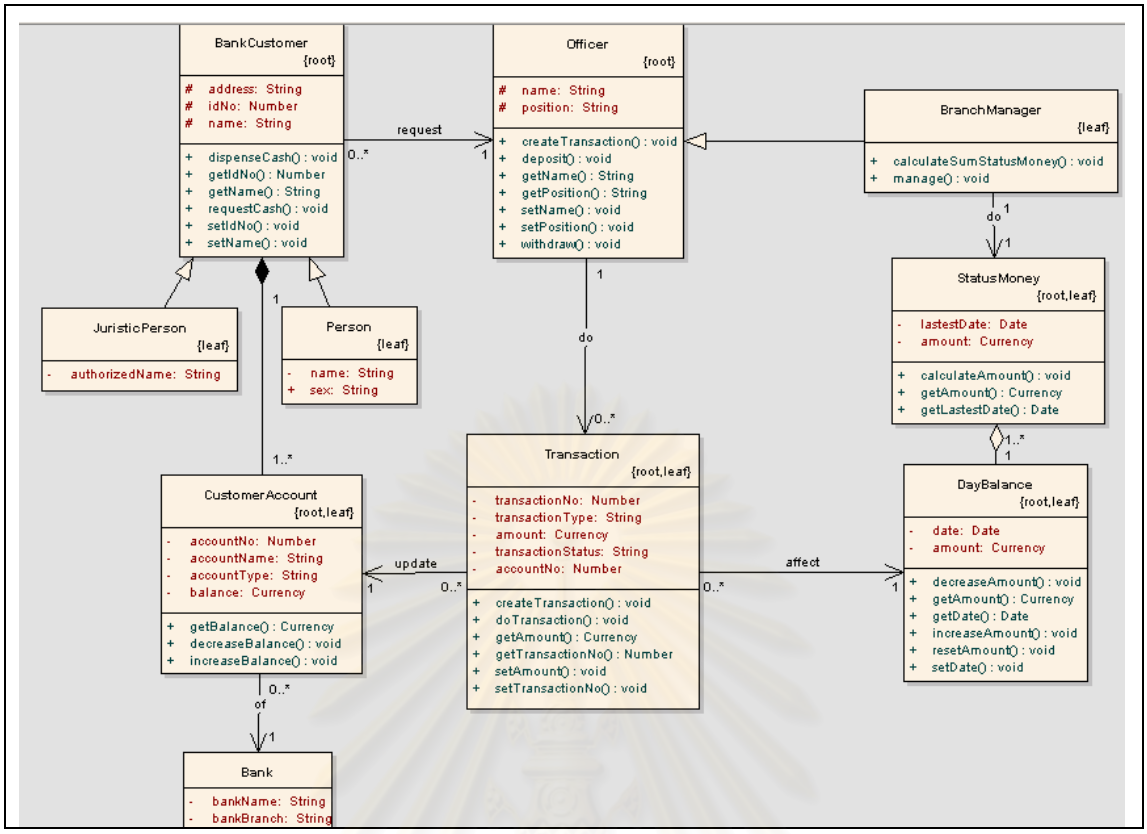


รูปที่ ก.5 แผนภาพคลาสระบบสารสนเทศคลังยา

6. ระบบการฝากถอนคำนวณเงินดำรงฐานะของธนาคารพาณิชย์

สำหรับระบบนี้ ลูกค้าของธนาคารสามารถเข้าใช้งานฟังก์ชันการฝากเงิน และฟังก์ชันการถอนเงินหากเงินในบัญชีมีจำนวนเพียงพอ รวมถึงระบบจะมีฟังก์ชันให้ตรวจสอบสถานะทางการเงิน และฟังก์ชันการปรับยอดเงินของธนาคารเมื่อสิ้นวันด้วย

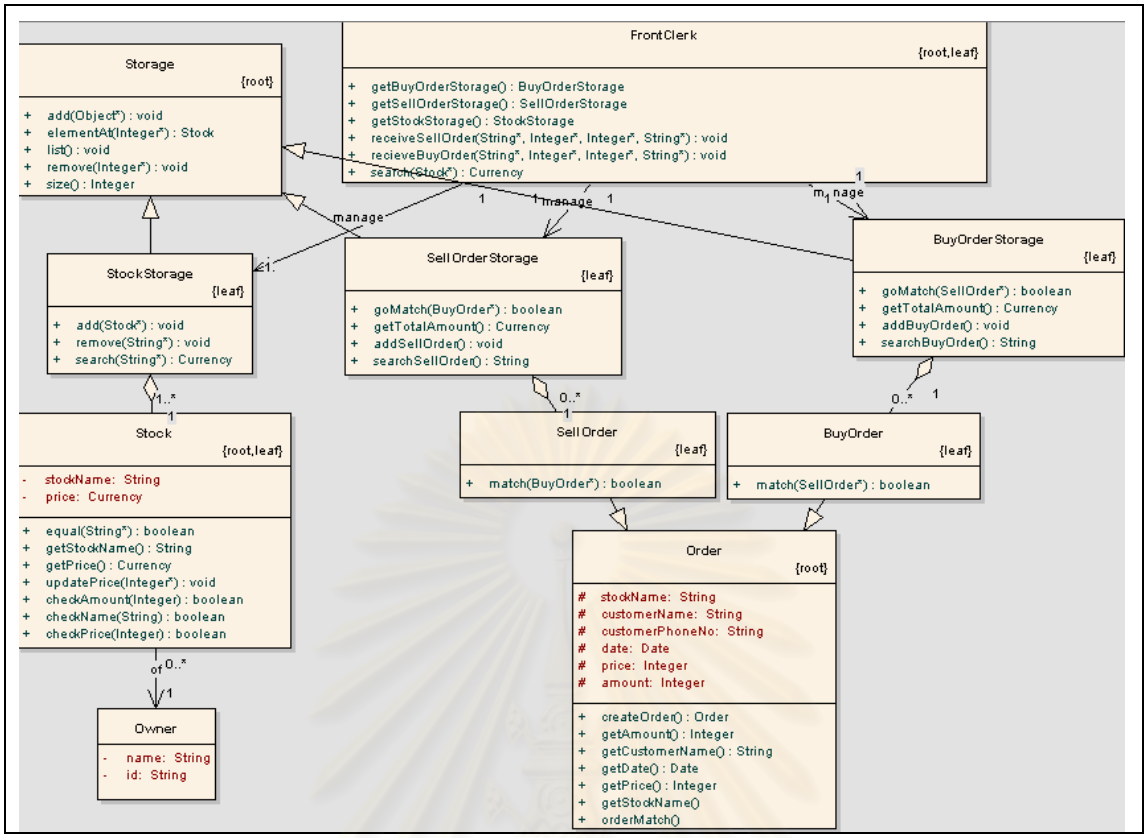
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ก.6 แผนภาพคลาสระบบการฝากถอนคำนวณเงินดำรงฐานะของธนาคารพาณิชย์

7. ระบบซื้อขายหุ้นผ่านโทรศัพท์เคลื่อนที่

ในระบบนี้ผู้ที่ต้องการซื้อขายหุ้นสามารถใช้งานระบบได้ผ่านทางโทรศัพท์เคลื่อนที่ โดยสามารถค้นหารายละเอียดของหุ้นที่ต้องการ รวมทั้งสามารถซื้อและขายหุ้นได้ด้วย ซึ่งหากมีการประกาศซื้อหุ้น ระบบจะตรวจสอบกับในตลาดหุ้นว่ามีผู้ใดประกาศขายหุ้นที่มีลักษณะตรงกันหรือไม่ หากตรงกัน จะส่งข้อมูลหุ้นที่มีการประกาศขายให้แก่ผู้ต้องการซื้อหุ้น หากมีการประกาศขายหุ้นผ่านทางโทรศัพท์เคลื่อนที่ ระบบจะตรวจสอบกับในตลาดหุ้นว่ามีผู้ใดประกาศซื้อหุ้นที่มีลักษณะตรงกันหรือไม่ หากตรงกัน จะส่งข้อมูลหุ้นที่มีการประกาศซื้อให้แก่ผู้ต้องการขายหุ้น

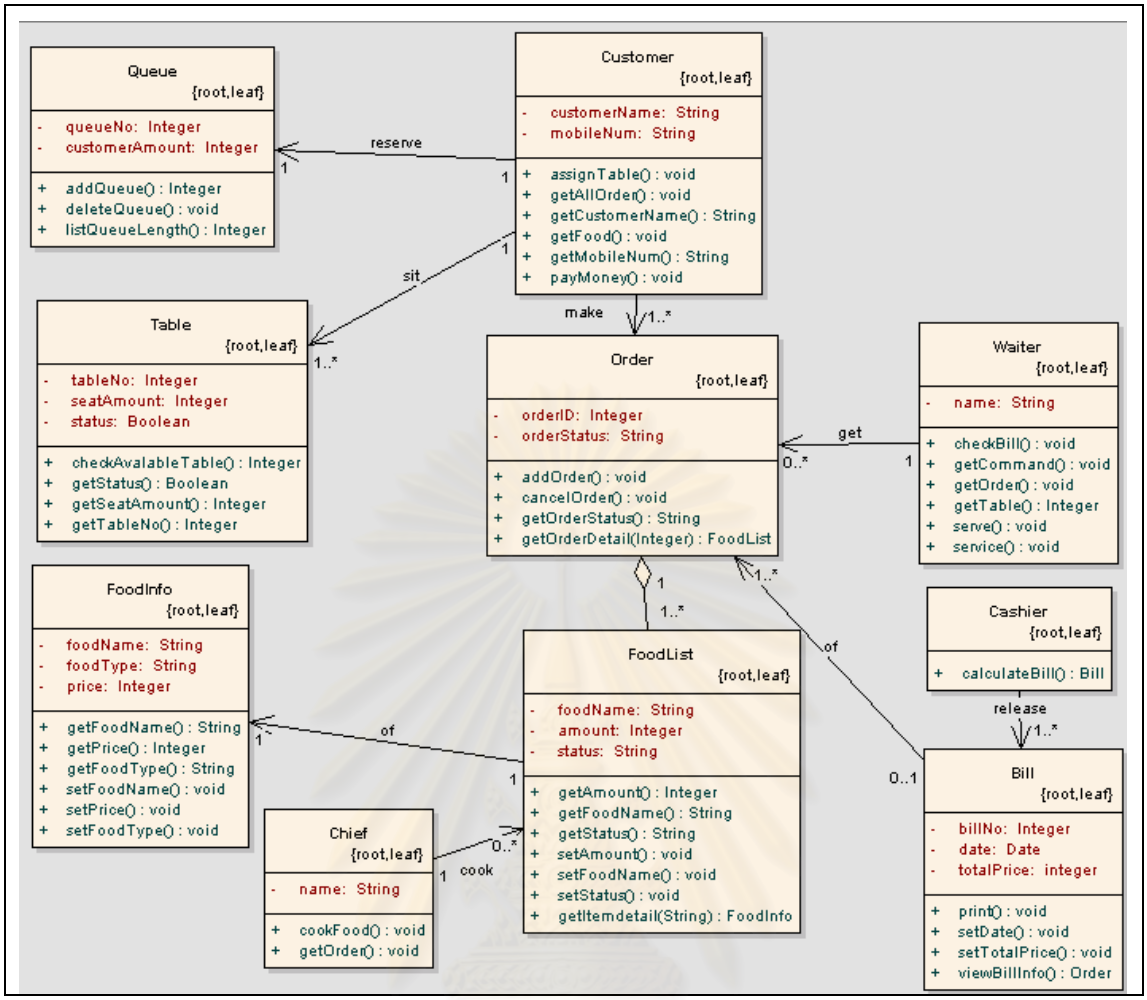


รูปที่ ก.7 แผนภาพคลาสระบบซื้อขายหุ้นผ่านโทรศัพท์เคลื่อนที่

8. ระบบจัดการร้านอาหาร

ระบบจัดการร้านอาหารนี้เป็นระบบที่รองรับฟังก์ชันการทำงานภายในร้านอาหาร เริ่มตั้งแต่การจัดที่นั่งให้กับลูกค้า, การสั่งอาหาร, การให้บริการต่างๆ, การคำนวณค่าอาหาร รวมถึงการจัดคิวให้กับลูกค้าในกรณีที่ที่นั่งภายในร้านเต็ม

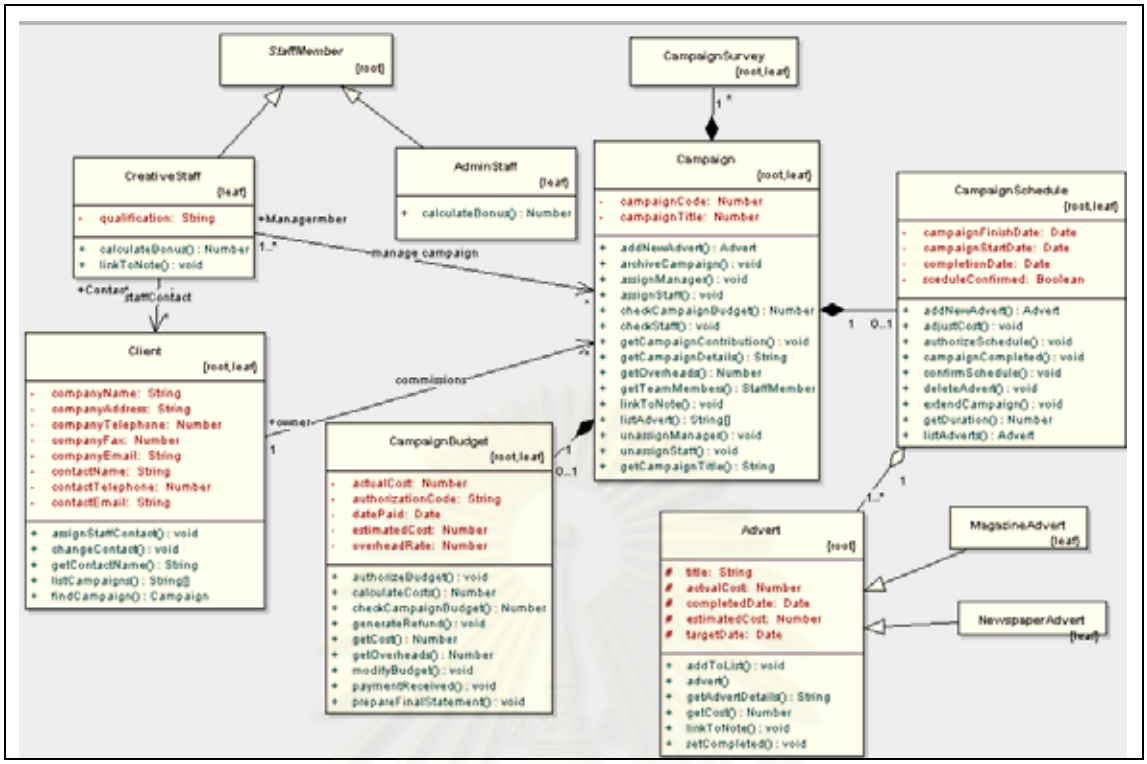
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ก.8 แผนภาพคลาสระบบจัดการร้านอาหาร

9. ระบบงานโฆษณา

ระบบนี้เป็นระบบสำหรับจัดการการทำงานภายในบริษัทรับทำโฆษณา ซึ่งมีสื่ออยู่หลายแขนงด้วยกัน และแสดงการทำงานระหว่างบริษัทโฆษณาและลูกค้า ทางบริษัทจะจัดโครงการที่ลูกค้าต้องการ เรียกว่าแคมเปญ (Campaign) แต่ละแคมเปญจะมีทีมงานซึ่งคัดเลือกมาจากบุคลากรของบริษัท และพนักงานของบริษัทหนึ่งคนสามารถทำงานได้หลายแคมเปญ ในช่วงเวลาเดียวกัน เมื่อโครงการหนึ่งเสร็จสิ้นก็จะมีการคำนวณค่าใช้จ่าย

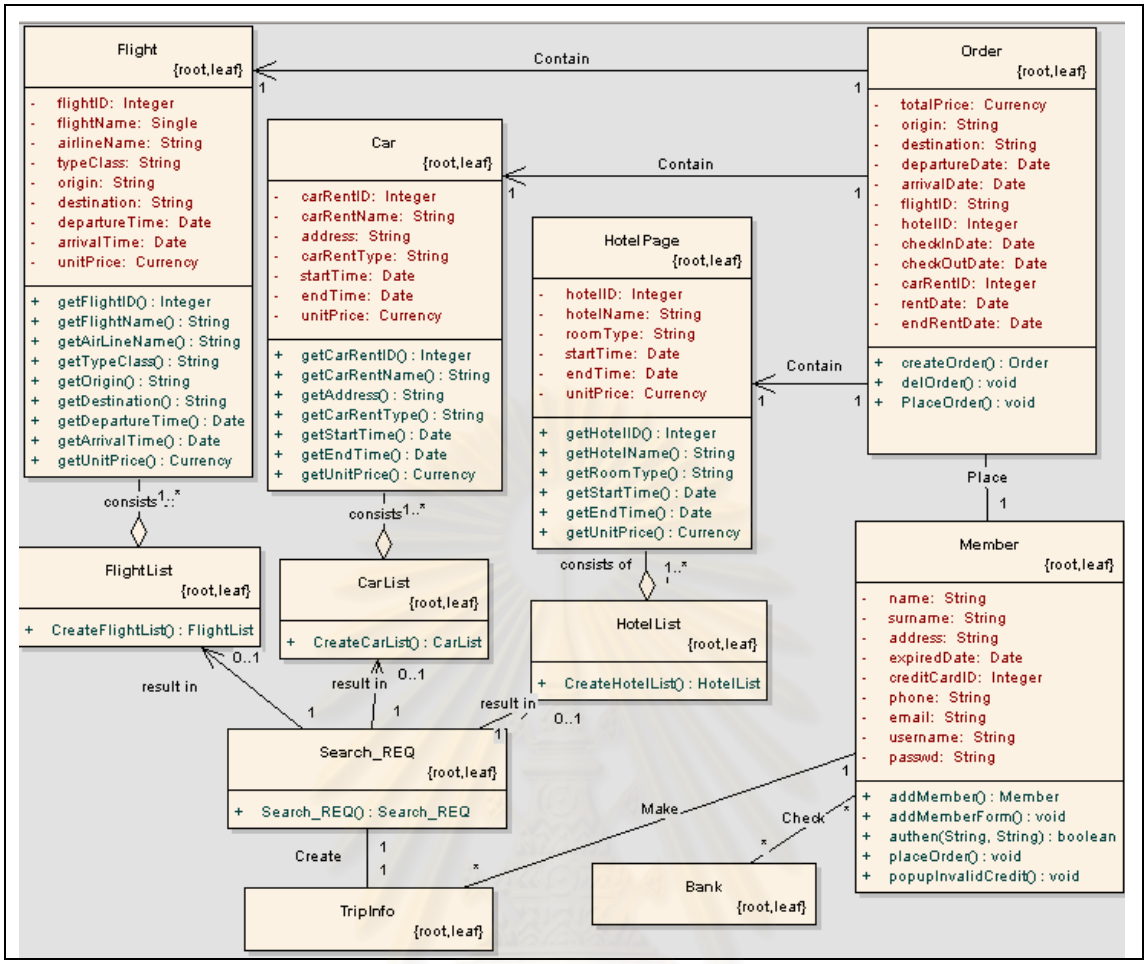


รูปที่ ก.9 แผนภาพคลาสระบบงานโฆษณา

10. ระบบจองท่องเที่ยว

ระบบจองท่องเที่ยว เป็นระบบที่สามารถให้ลูกค้าจองทัวร์ผ่านอินเทอร์เน็ตได้ โดยมีฟังก์ชันการทำงานดังนี้

1. ผู้ใช้สามารถระบุวันไปและวันกลับ พร้อมทั้งจำนวนคนเดินทาง โดยระบบจะจัดการหาข้อมูลที่เหมาะสมกับวันเดินทางให้แก่ผู้ใช้เอง
2. เมื่อผู้ใช้กำหนดข้อมูลการเดินทางที่ต้องการไปเรียบร้อยแล้ว ระบบจะทำการเลือกเที่ยวบินที่เหมาะสมกับความต้องการของลูกค้าทั้งหมดที่เป็นไปได้ และแสดงรายละเอียดของเที่ยวบินนั้นๆ พร้อมจำนวนที่นั่งที่เหลือและราคาในแต่ละแพ็คเกจ ให้ลูกค้าเลือก
3. สำหรับโรงแรมและรถเช่าก็จะทำในลักษณะเดียวกัน โดยระบบจะทำการรวมค่าใช้จ่ายทั้งหมดที่รวมไว้ จนถึงขณะนั้นมาให้ด้วย
4. ระบบจะเก็บเงินลูกค้าผ่านบัตรเครดิตโดยลูกค้าที่สามารถซื้อแพ็คเกจได้ คือลูกค้าที่เป็นสมาชิก โดยระบบจะต้องมีการตรวจสอบความถูกต้องก่อนการเข้าสู่ระบบด้วย หากลูกค้าไม่ใช่สมาชิกก็สามารถทำการลงทะเบียนได้ โดยทำการกรอกข้อมูลส่วนตัว

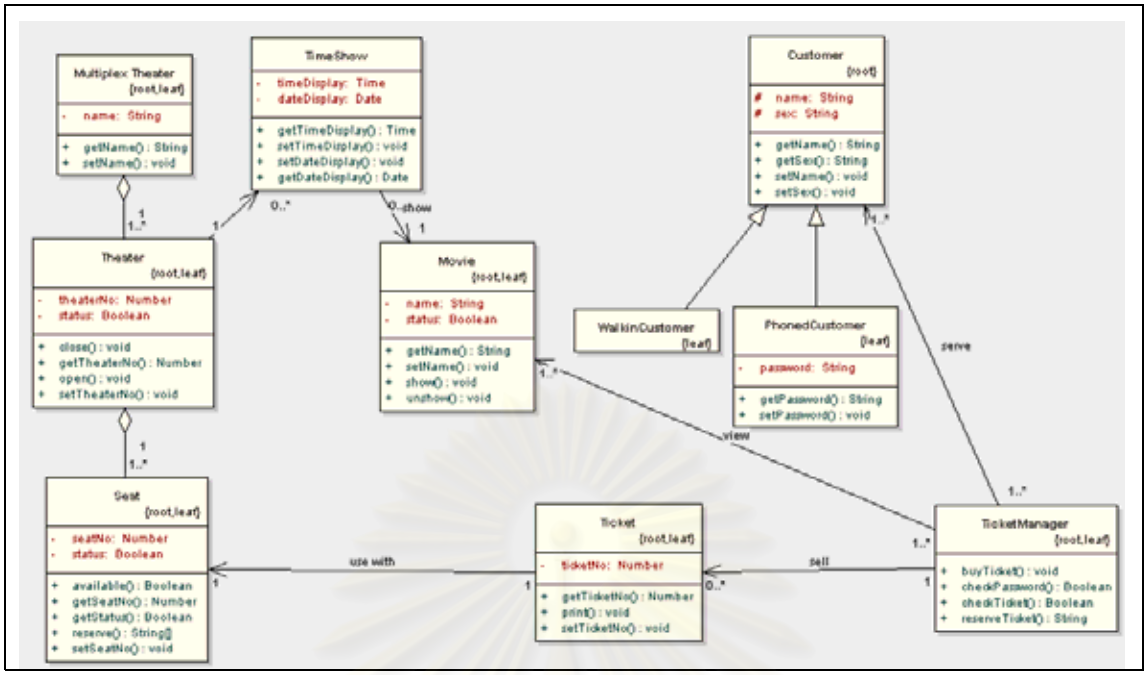


รูปที่ ก.10 แผนภาพคลาสระบบจองท่องเที่ยว

11. ระบบโรงภาพยนตร์แบบ Multiplex

โรงภาพยนตร์ระบบ Multiplex คือโรงภาพยนตร์ที่มีหลายโรงฉายย่อยอยู่ภายใน การซื้อตั๋วสามารถทำได้ที่หน้าโรงภาพยนตร์ ส่วนการจองตั๋วล่วงหน้าสามารถทำได้ผ่านทางโทรศัพท์ โดยผู้รับจองจะแจ้งรหัสแก่ผู้จอง เพื่อให้ผู้จองนำมาแจ้งแก่เจ้าหน้าที่หน้าโรงเพื่อแสดงตัวเข้ารับบัตรชม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

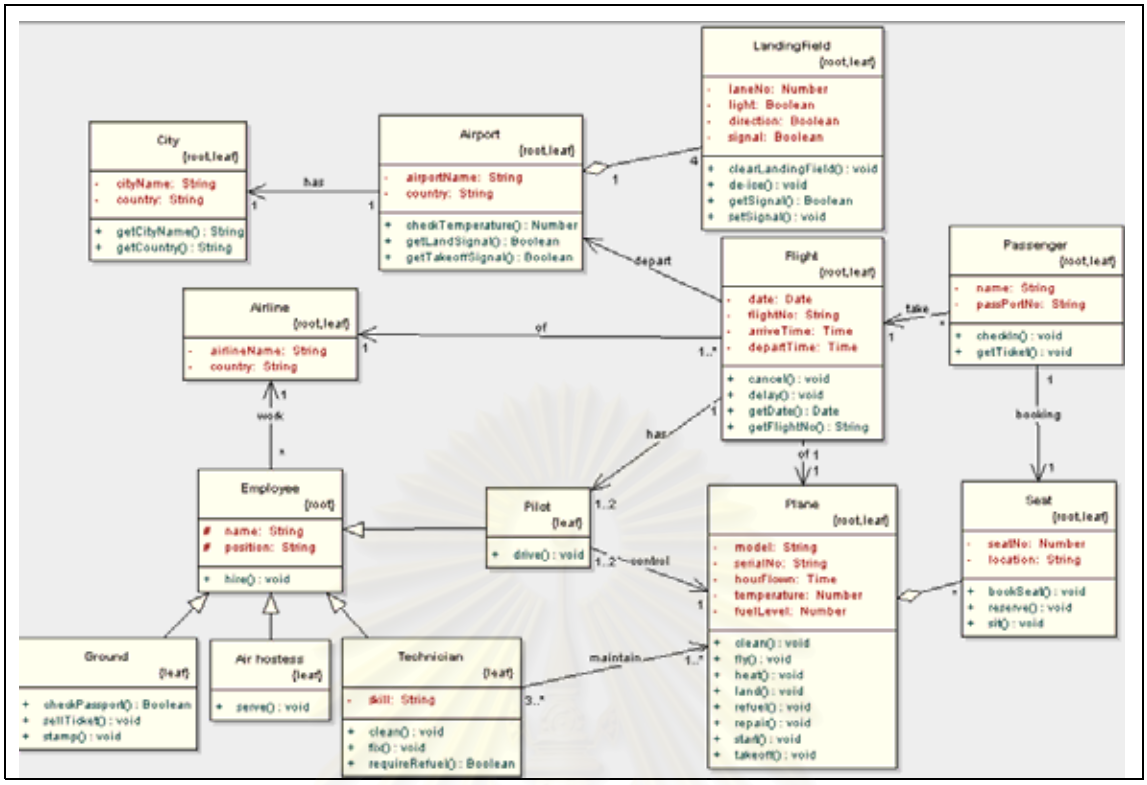


รูปที่ ก.11 แผนภาพคลาสระบบโรงภาพยนตร์แบบ Multiplex

12. ระบบขนส่งทางอากาศ

ระบบนี้เป็นการขนส่งผู้โดยสารระหว่างเมือง โดยสายการบินของแต่ละประเทศจะมีเที่ยวบินของตัวเองรับส่งผู้โดยสารระหว่างสนามบินของเมืองต่างๆ โดยพนักงานภาคพื้นดินของสายการบินนั้นจะเป็นผู้ขายตั๋วให้กับผู้โดยสาร และมีพนักงานซ่อมเครื่องบินประจำสายการบินนั้นๆ เครื่องบินแต่ละลำจะมีพนักงานขับเครื่องบินประจำเครื่องบินแต่ละเที่ยวบิน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

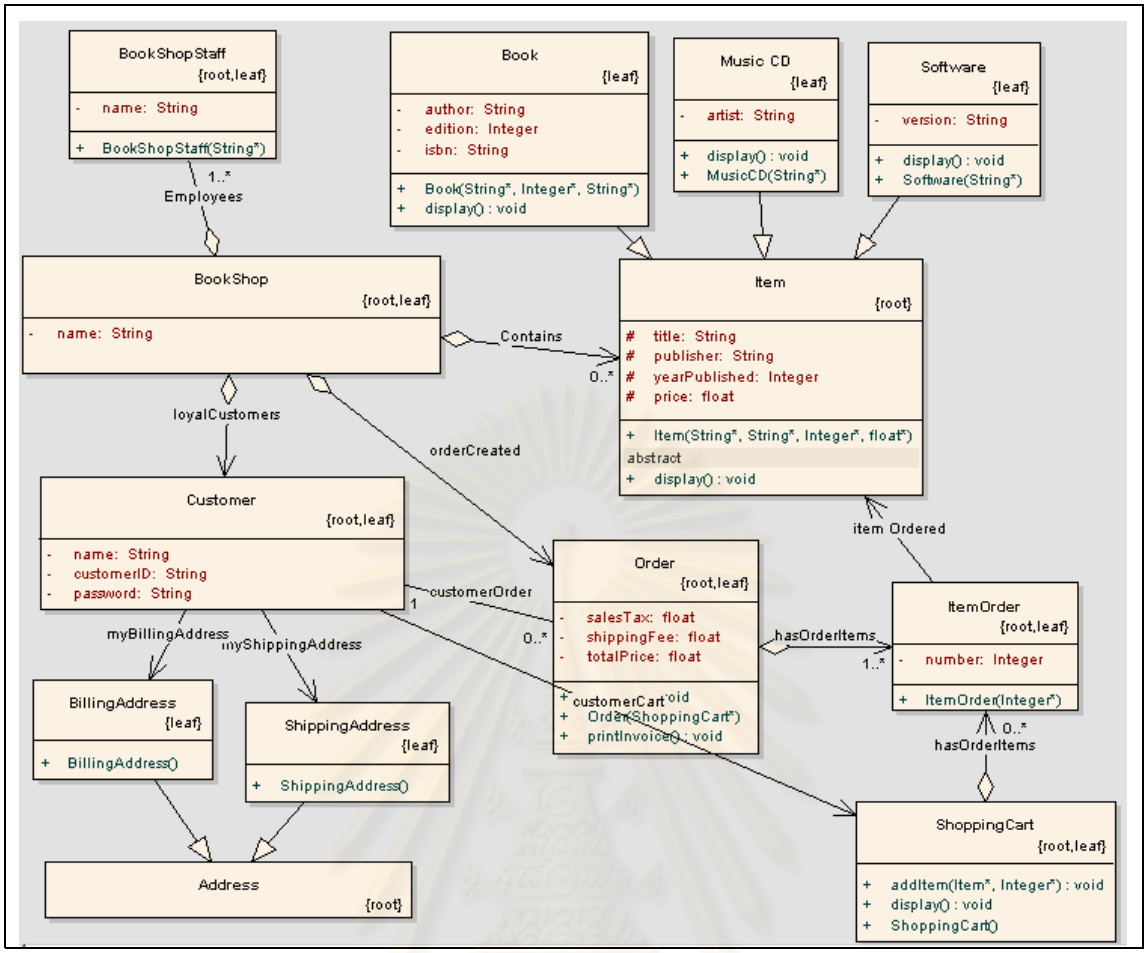


รูปที่ ก.12 แผนภาพคลาสระบบขนส่งทางอากาศ

13. ระบบร้านหนังสือออนไลน์

ระบบขายหนังสือออนไลน์ เป็นร้านขายหนังสือผ่านเว็บไซต์ นอกจากจะขายหนังสือแล้ว ยังมีการขายซีดีเพลง และซอฟต์แวร์อีกด้วย โดยลูกค้าจะต้องกรอกข้อมูลส่วนตัวเพื่อสมัครเป็นสมาชิกของทางร้านก่อนเข้าไปสั่งซื้อสินค้า และในการสั่งซื้อสินค้าลูกค้าต้องกรอกที่อยู่เพื่อส่งสินค้าไปตามที่อยู่ที่ได้ระบุไว้

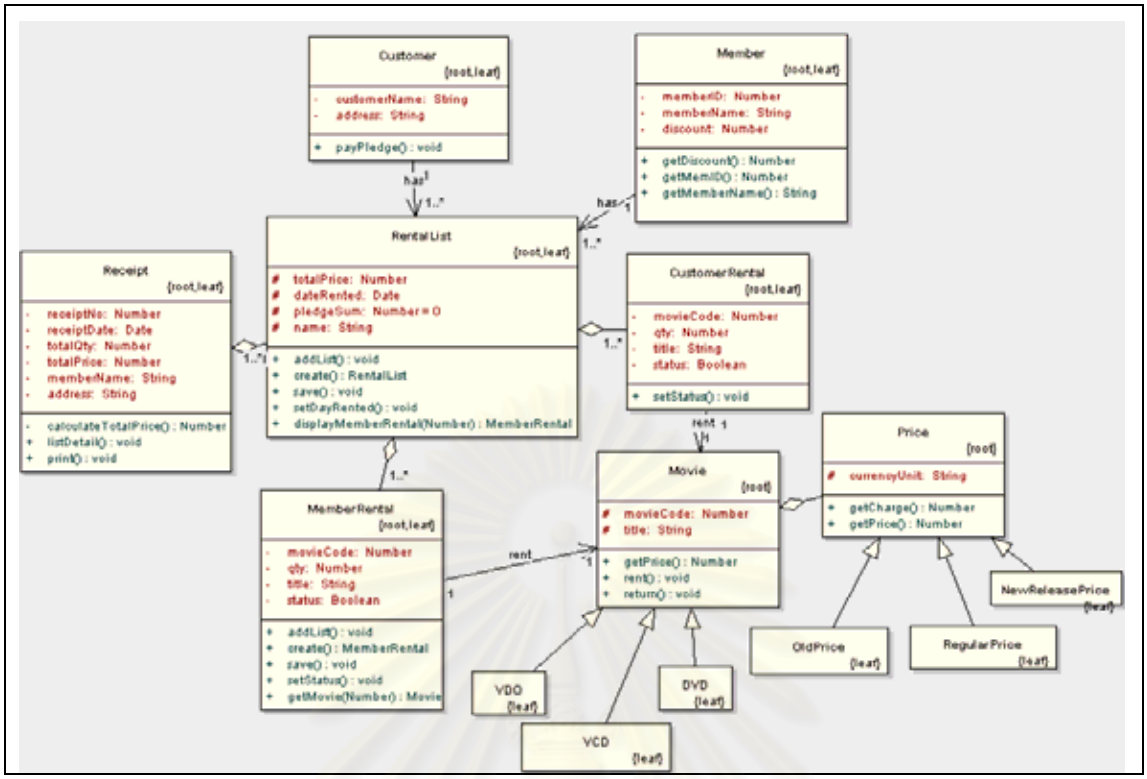
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ก.13 แผนภาพคลาสระบบร้านหนังสือออนไลน์

14. ระบบเช่าวีดีโอ

ระบบนี้เป็นระบบสำหรับการเช่า-คืนภาพยนตร์ของร้านเช่าวีดีโอ โดยรายการเช่าจะแบ่งออกเป็นรายการเช่า-คืน ของสมาชิก และรายการเช่า-คืนของลูกค้าทั่วไป เนื่องจากว่าสมาชิกจะมีส่วนลดในการเช่าภาพยนตร์เรื่องต่างๆ ภายในร้านเช่าวีดีโอ ส่วนลูกค้าทั่วไปเมื่อมีการยืมภาพยนตร์ต่างๆ จะต้องเสียค่ามัดจำก่อน เมื่อสมาชิกและลูกค้าทั่วไปนำภาพยนตร์มาคืนให้กับทางร้าน ทางร้านจึงจะออกไปเสร็จคำนวณค่าใช้จ่ายที่สมาชิกและลูกค้าต้องชำระให้ โดยภาพยนตร์จะแบ่งออกเป็น วีดีโอ, วีซีดี และดีวีดี และจำแนกค่าเช่าตามความใหม่ของภาพยนตร์

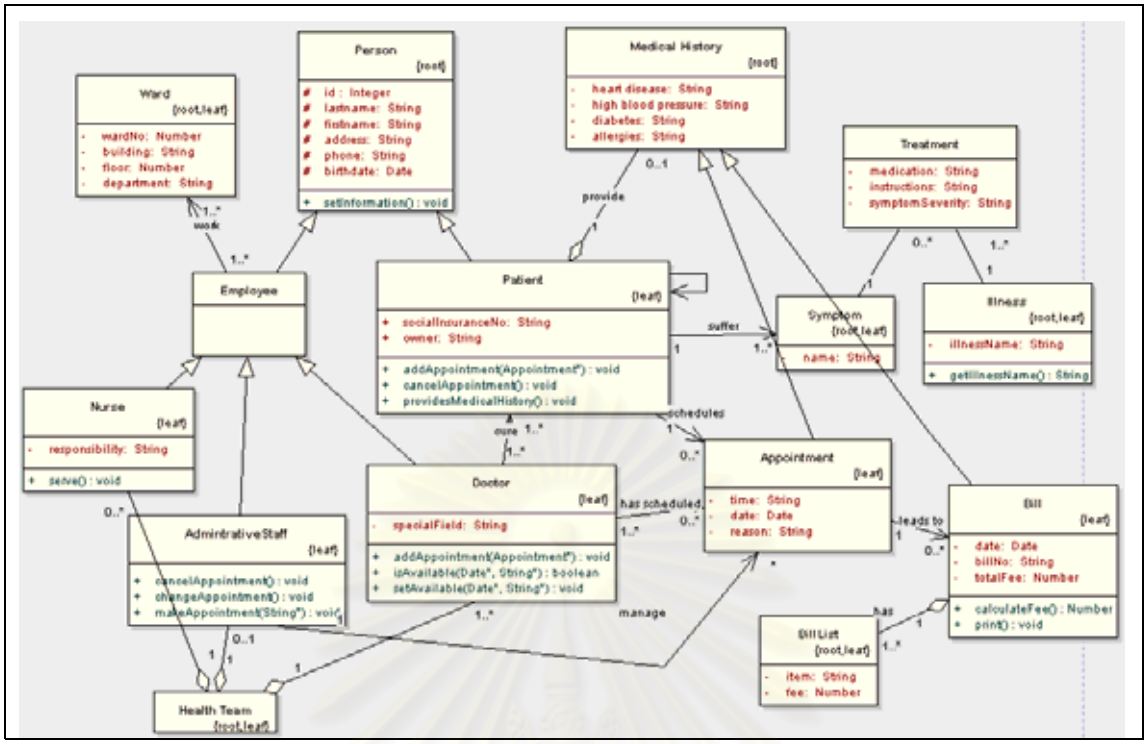


รูปที่ ก.14 แผนภาพคลาสระบบเช่าวีดีโอ

15. ระบบโรงพยาบาล

ระบบนี้เป็นการจำลองความสัมพันธ์ระหว่างบุคคลต่าง ๆ ในโรงพยาบาล ซึ่งประกอบด้วย แพทย์ พยาบาล เจ้าหน้าที่ และคนไข้ โดยแพทย์ พยาบาล และเจ้าหน้าที่ จะเป็นสมาชิกของทีมสุขภาพ เพื่อช่วยในการดูแลคนไข้ โดยระบบจะมีการเก็บประวัติของคนไข้ เก็บอาการป่วยของคนไข้ เพื่อใช้ประกอบการวินิจฉัยโรคของแพทย์ การพบแพทย์ทุกครั้ง คนไข้จะต้องทำการนัดหมายก่อนล่วงหน้า เมื่อคนไข้มาพบแพทย์ตามนัด และได้รับการตรวจรักษาเสร็จแล้ว จะมีการออกบิลเพื่อเก็บค่าใช้จ่ายจากคนไข้

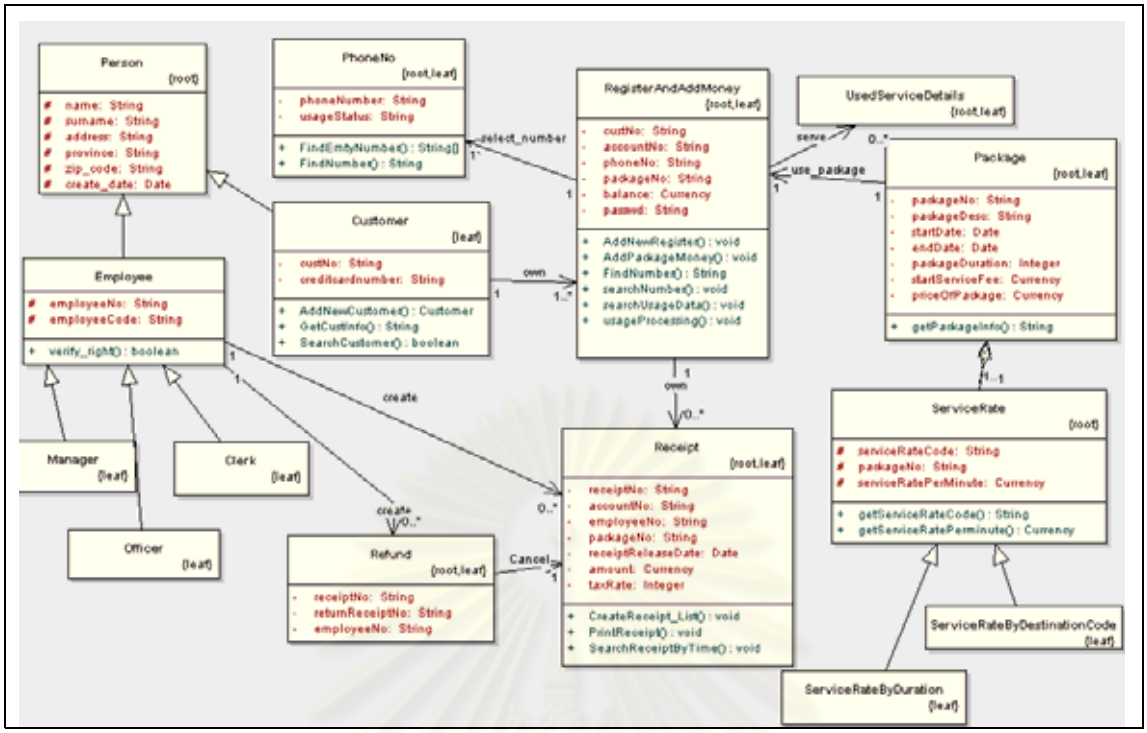
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ก.15 แผนภาพคลาสระบบโรงพยาบาล

16. ระบบชำระค่าบริการโทรศัพท์ล่วงหน้า

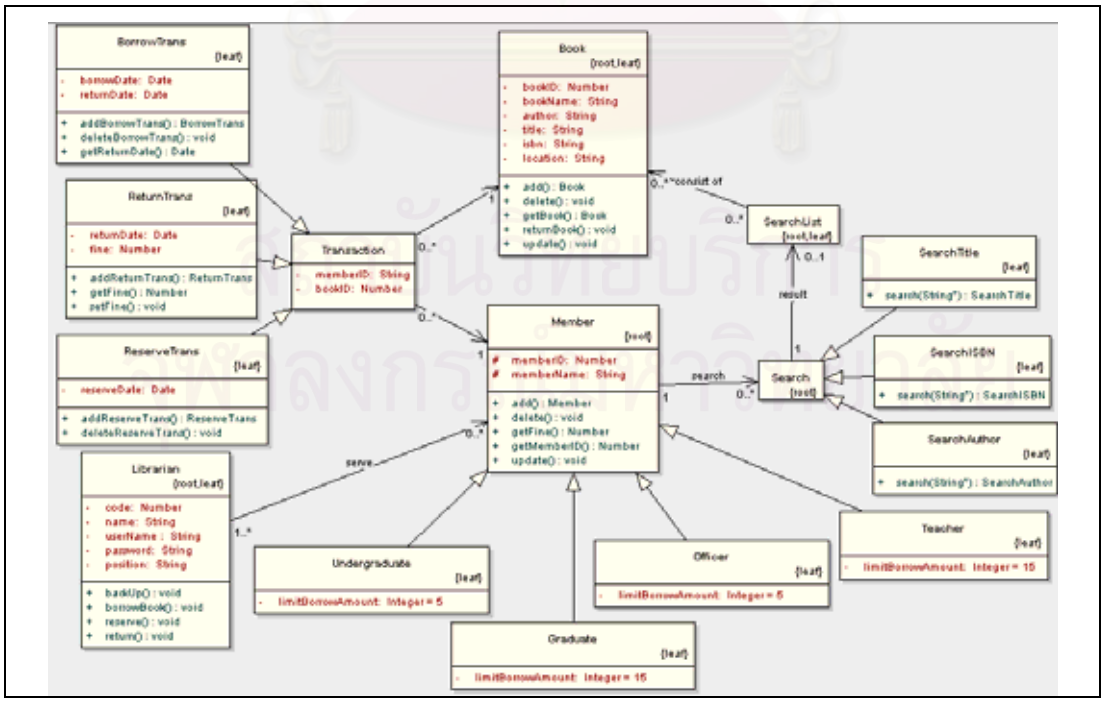
เป็นระบบที่สามารถให้บริการลูกค้าที่ต้องการชำระค่าบริการโทรศัพท์เคลื่อนที่ล่วงหน้าก่อนการใช้โทรศัพท์ (คล้ายระบบ one-two-call หรือ dprompt) โดยบริษัทจะมีแพ็คเกจให้ลูกค้าเลือกได้ 2 แบบ คือแบบคิดค่าบริการตามช่วงเวลาที่ใช้โทร และแบบคิดค่าบริการตามรหัสปลายทางที่โทรไปหา ระบบสามารถจะออกใบเสร็จรวมถึงสามารถยกเลิกใบเสร็จใบนั้นๆ เพื่อคืนเงินแก่ลูกค้าได้ โดยระบบจ่ายเงินล่วงหน้าจะยอมให้ลูกค้ามาทำการจ่ายเงินที่ศูนย์บริการ และพนักงานจะทำการเติมเงินให้ลูกค้าเข้าระบบ ในกรณีที่เป็นลูกค้าใหม่ลูกค้าจะต้องเสียค่าเปิดเบอร์และเติมเงินพร้อมทั้งเลือกหมายเลขโทรศัพท์ โดยพนักงานของบริษัทแบ่งออกได้เป็น สามกลุ่ม ได้แก่ ผู้จัดการ, เจ้าหน้าที่ และ เสมียน ในส่วนของการจัดการข้อมูลลูกค้าเจ้าหน้าที่จะเป็นผู้มีหน้าที่รับผิดชอบเปลี่ยนแปลง ส่วนของรายงานแสดงข้อมูลทางธุรกิจและจัดการข้อมูลแพ็คเกจการให้บริการจะรับผิดชอบโดยผู้จัดการเท่านั้น ส่วนเสมียนจะสามารถทำการจดทะเบียนลูกค้าใหม่, เติมเงิน และดูยอดการใช้บริการได้เท่านั้น



รูปที่ ก.16 แผนภาพคลาสระบบชำระค่าบริการล่วงหน้า

17. ระบบห้องสมุดใหญ่

ระบบห้องสมุดสนับสนุนงานการบริการห้องสมุดให้มีประสิทธิภาพการสืบค้นให้รวดเร็วยิ่งขึ้น ให้บริการยืม-คืนหนังสือ, จองหนังสือ, ตรวจสอบรายชื่อหนังสือที่สมาชิกค้างส่งได้, คำนวณค่าปรับหนังสือเกินกำหนด และช่วยให้การแก้ไขข้อมูลต่างๆ ที่เกี่ยวกับระบบสามารถทำได้ง่าย



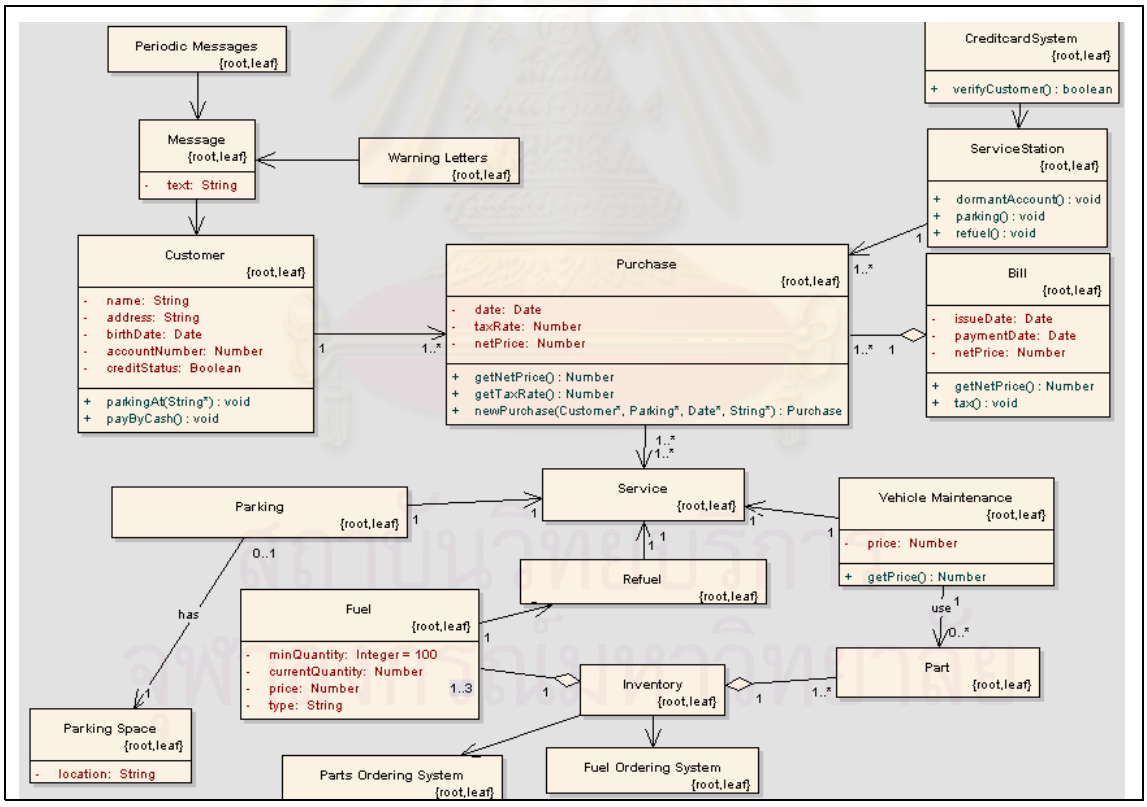
รูปที่ ก.17 แผนภาพคลาสระบบห้องสมุดใหญ่

18. ระบบสถานีบริการหลวง

สถานีบริการหลวงมีบริการให้แก่ลูกค้า 3 ประเภท คือ บริการเติมน้ำมัน, บริการตรวจสอบเครื่องยนต์ และบริการที่จอดรถ โดยที่ลูกค้าสามารถชำระค่าบริการได้ทันที หรือจะเก็บบิลไว้ชำระเมื่อถึงสิ้นเดือนก็ได้ ในการชำระค่าบริการของลูกค้าสามารถชำระค่าบริการด้วยเงินสด, บัตรเครดิต หรือเช็ค

ระบบสามารถออกรายงานให้แก่ผู้จัดการสถานีบริการเพื่อใช้ในการควบคุมสินค้าและบริการ โดยระบบต้องสามารถเตือนให้ผู้จัดการสถานีทราบได้ด้วยว่าจะต้องสั่งซื้อสินค้าและน้ำมันเมื่ออยู่ในระดับที่ไม่เพียงพอต่อการให้บริการ

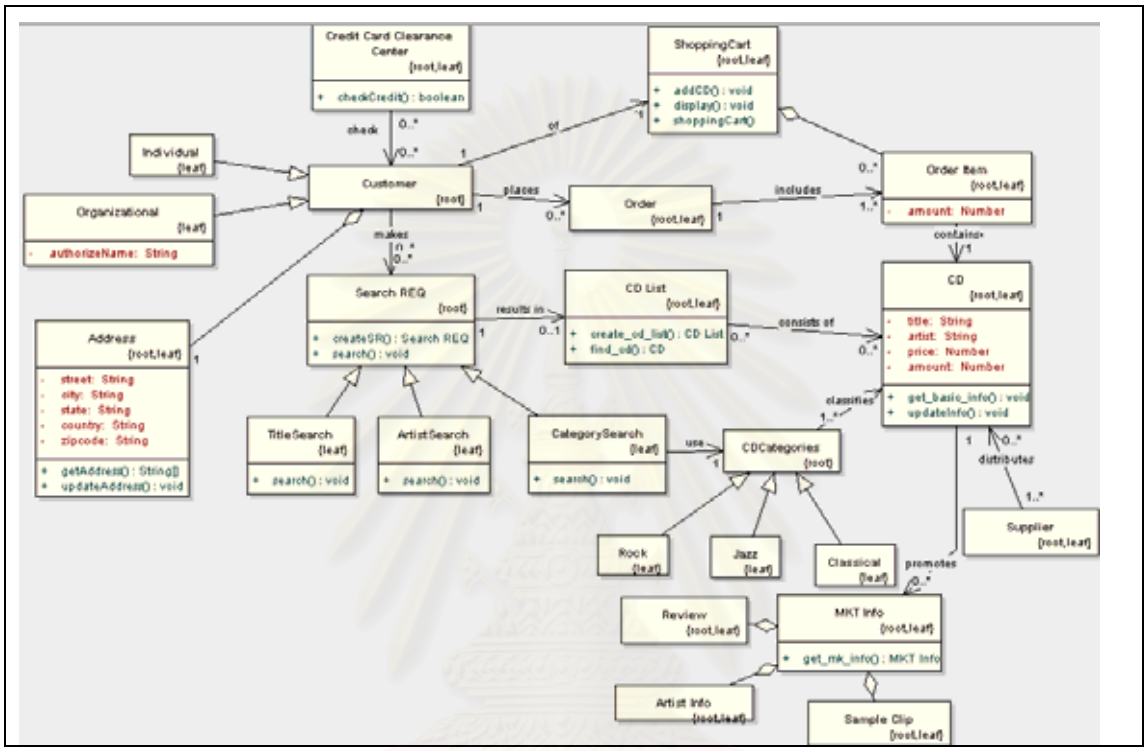
การแจ้งค่าบริการแก่ลูกค้า จะส่งบิลไปเก็บค่าบริการในวันแรกของเดือน ถ้าหากบิลนั้นค้างชำระเป็นระยะเวลา 90 วัน จะทำการยกเลิกเครดิตของลูกค้าคนนั้นทันที และ ระบบยังสามารถตรวจสอบยอดค่าใช้จ่ายของลูกค้าที่มียอดค้างชำระ พร้อมกับส่งจดหมายเตือนไปยังลูกค้าด้วย



รูปที่ ก.18 แผนภาพคลาสระบบสถานีบริการหลวง

19. ระบบขายซีดีออนไลน์

ระบบนี้เป็นระบบจัดการการขายซีดีเพลง ทางอินเทอร์เนต โดยลูกค้าสามารถสั่งซื้อสินค้าผ่านทางอินเทอร์เนต ได้โดยผ่านทางเว็บไซต์ของระบบ โดยมีฟังก์ชันรองรับการทำงานดังนี้ ฟังก์ชันการค้นหาข้อมูลสินค้า, ฟังก์ชันการสั่งซื้อสินค้า และฟังก์ชันการโฆษณาส่งเสริมสินค้า ลูกค้าทุกคนจะต้องทำการกรอกข้อมูลส่วนตัว และสามารถสั่งซื้อสินค้าโดยใช้บัตรเครดิต



รูปที่ ก.19 แผนภาพคลาสระบบขายซีดีออนไลน์

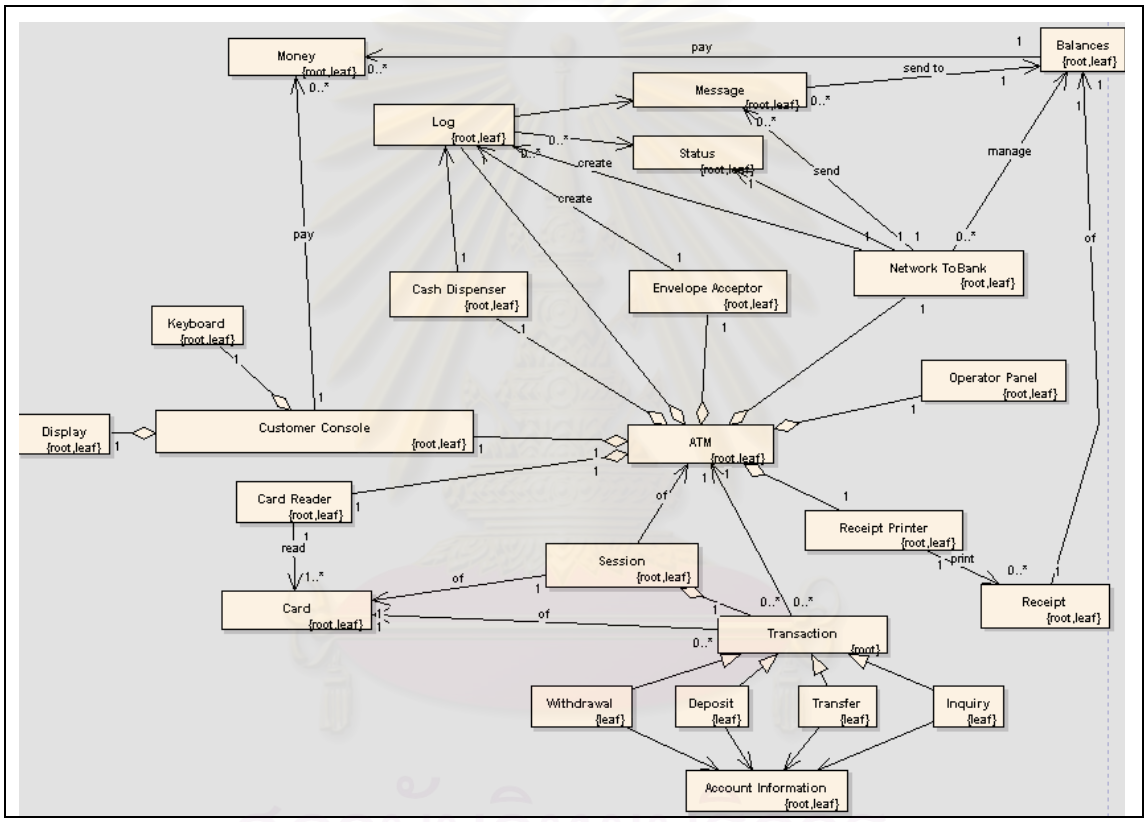
20. ระบบ ATM

ระบบ เอทีเอ็ม ประกอบด้วย ส่วนของการอ่านบัตร เอทีเอ็ม , customer console ที่ประกอบด้วยแป้นพิมพ์และหน้าจอเพื่อติดต่อกับลูกค้า, ช่องสำหรับฝากเงิน, ช่องจ่ายเงินสด, เครื่องพิมพ์สำหรับพิมพ์รายการ และ สวิตซ์สำหรับการปิด-เปิด เครื่อง เอทีเอ็ม เมื่อต้องการใช้บริการ ลูกค้าต้องสอดบัตร เอทีเอ็ม และกดรหัสส่วนตัว (PIN: Personal identification number) ทั้งหมายเลขบัตร เอทีเอ็ม และรหัสจะถูกส่งไปยังธนาคารเพื่อทำการตรวจสอบความ

เครื่อง เอทีเอ็ม จะต้องติดต่อกับธนาคารในการทำรายการทุก ๆ รายการเพื่อให้ธนาคารทำการตรวจสอบความถูกต้องและอนุมัติให้เครื่อง เอทีเอ็ม ทำรายการได้ในขณะนั้น ดังนั้นรายการจะเสร็จสมบูรณ์ได้เมื่อธนาคารอนุมัติ ยกเว้นในกรณีของการฝากเงิน จะมีการส่งข้อความจากเครื่อง เอทีเอ็ม ไปยังธนาคารเพื่อแจ้งให้ทราบว่าลูกค้าได้สอดซองสำหรับฝากเงิน (ซึ่งการสอดซองฝากเงินจะต้องทำภายในระยะเวลาที่กำหนด หากลูกค้าสอดซองฝากเงินไม่ทันหรือกดปุ่ม

ยกเลิกจะไม่มีการส่งข้อความจากเครื่อง เอทีเอ็ม ไปยังธนาคาร และจะไม่ถือว่ามีการฝากเงินสำหรับลูกค้าคนดังกล่าว) เมื่อทำรายการเสร็จสิ้นทั้งหมดแล้ว เครื่อง เอทีเอ็ม จะพิมพ์ใบแจ้งรายการให้ลูกค้าทราบ

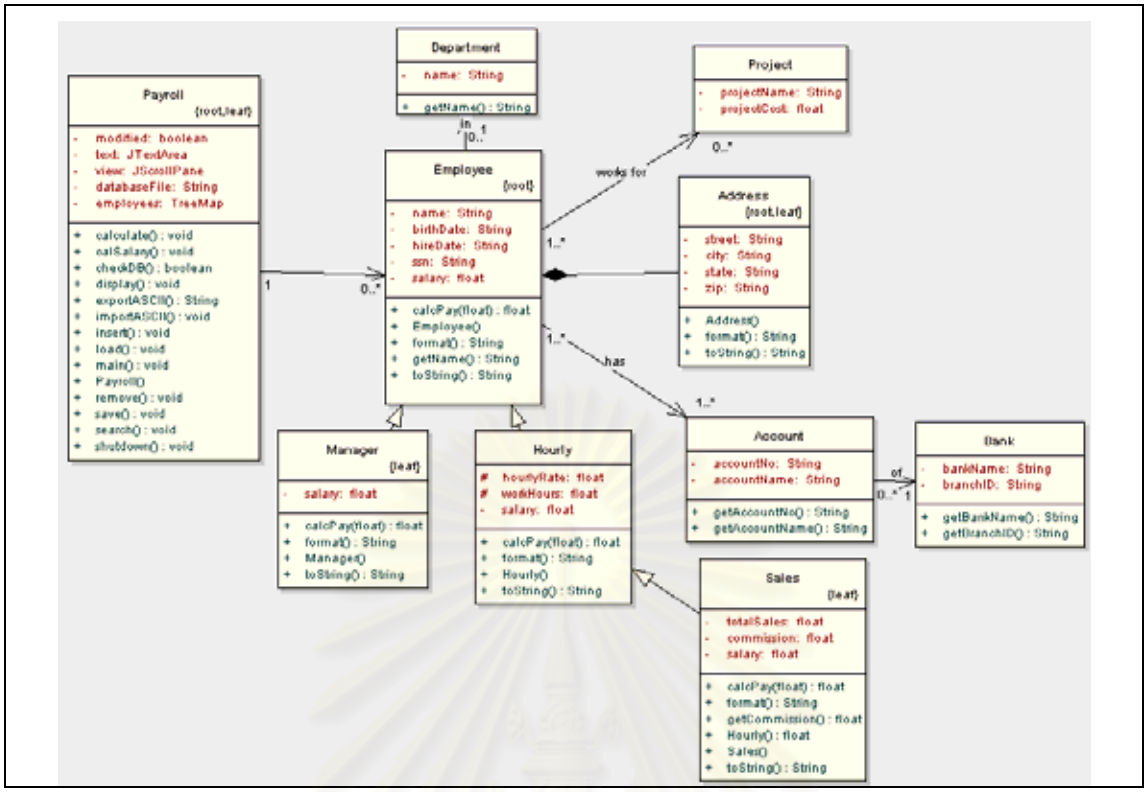
เครื่อง เอทีเอ็ม จะมีสวิตซ์สำหรับให้เจ้าหน้าที่ทำการปิด-เปิด เครื่อง ภายหลังการเปิดเครื่อง จะมีการตรวจสอบว่าเจ้าหน้าที่เป็นผู้มีสิทธิ์ในการจัดการเครื่อง เอทีเอ็ม แล้วจึงป้อนจำนวนเงินที่จะนำมาใส่ในเครื่อง การปิดเครื่อง เอทีเอ็ม จะทำได้เมื่อในขณะนั้นไม่มีการให้บริการแก่ลูกค้า เมื่อสวิตซ์ของเครื่องถูกปิด เครื่องจะปิดแล้วเจ้าหน้าที่จะทำการเก็บซองฝากเงิน และทำการใส่เงินในเครื่องตามจำนวนที่กำหนด ใส่กระดาษสำหรับพิมพ์รายการและอื่น ๆ



รูปที่ ก.20 แผนภาพคลาสระบบ ATM

21. ระบบจ่ายเงิน

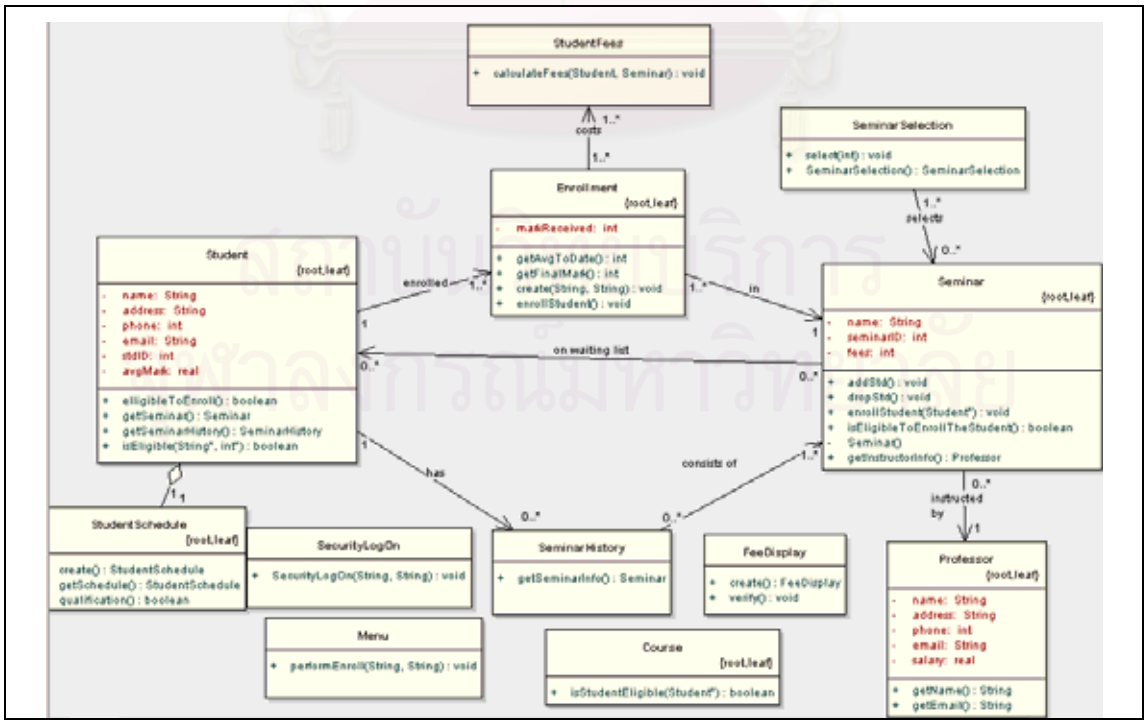
ระบบการจ่ายเงิน เป็นระบบเพื่อคำนวณเงินเดือนให้กับพนักงานภายในบริษัท ซึ่งมีผู้จัดการและพนักงานขาย โดยการคำนวณเงินเดือนของพนักงานขายนั้น จะทำการคำนวณตามจำนวนเวลาที่ทำงาน และอัตราเงินเดือนในแต่ละช่วงเวลาจะไม่เท่ากัน ขึ้นอยู่กับช่วงเวลาที่พนักงานขายเข้าทำงาน



รูปที่ ก.21 แผนภาพคลาสระบบจ่ายเงิน

22. ระบบลงทะเบียนสัมมนา

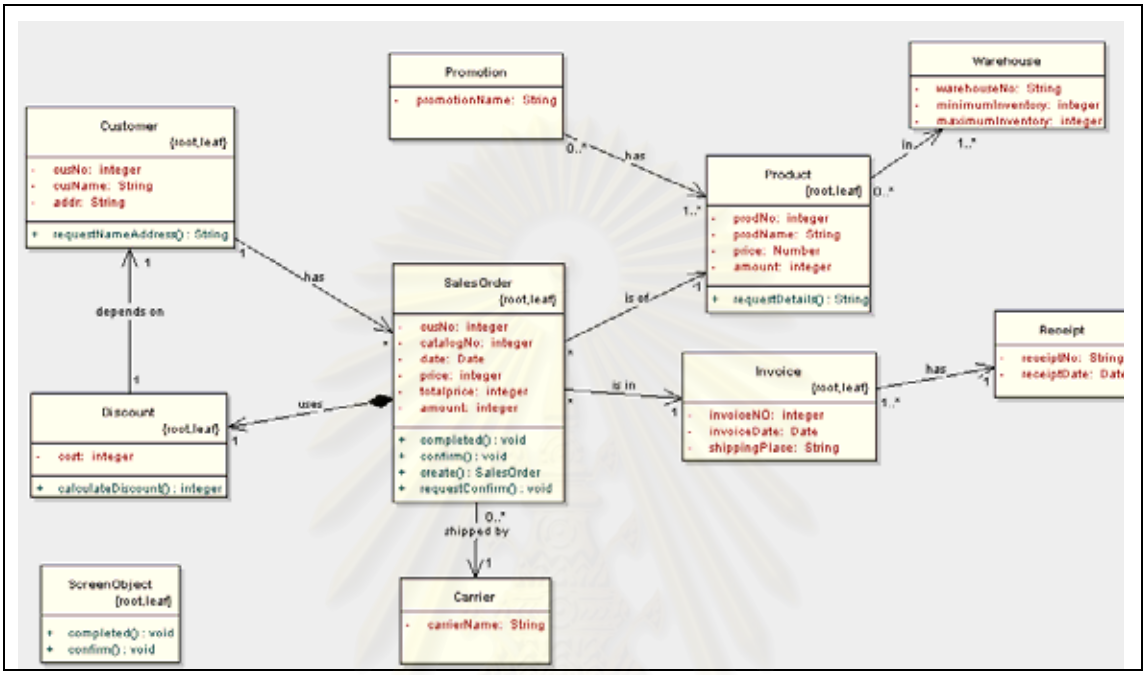
ระบบมีหน้าที่สำหรับรับลงทะเบียนเรียนวิชาสัมมนาของนักศึกษา โดยนักเรียนต้องลงทะเบียนเรียนวิชาสัมมนาอย่างน้อย 1 ตัว ซึ่งวิชาสัมมนาแต่ละตัวจะมีผู้สอนเพียงคนเดียวเท่านั้น



รูปที่ ก.22 แผนภาพคลาสระบบลงทะเบียนสัมมนา

23. ระบบส่วนลดของการขาย

ระบบนี้เป็นระบบการขายสินค้า ให้แก่ลูกค้า โดยจะมีการให้ส่วนลดแก่ลูกค้า โดยส่วนลดจะขึ้นอยู่กับลูกค้าและสินค้าที่ซื้อ และภายหลังการขายสินค้า จะต้องมีการออกใบเสร็จรับเงินแก่ลูกค้าด้วย

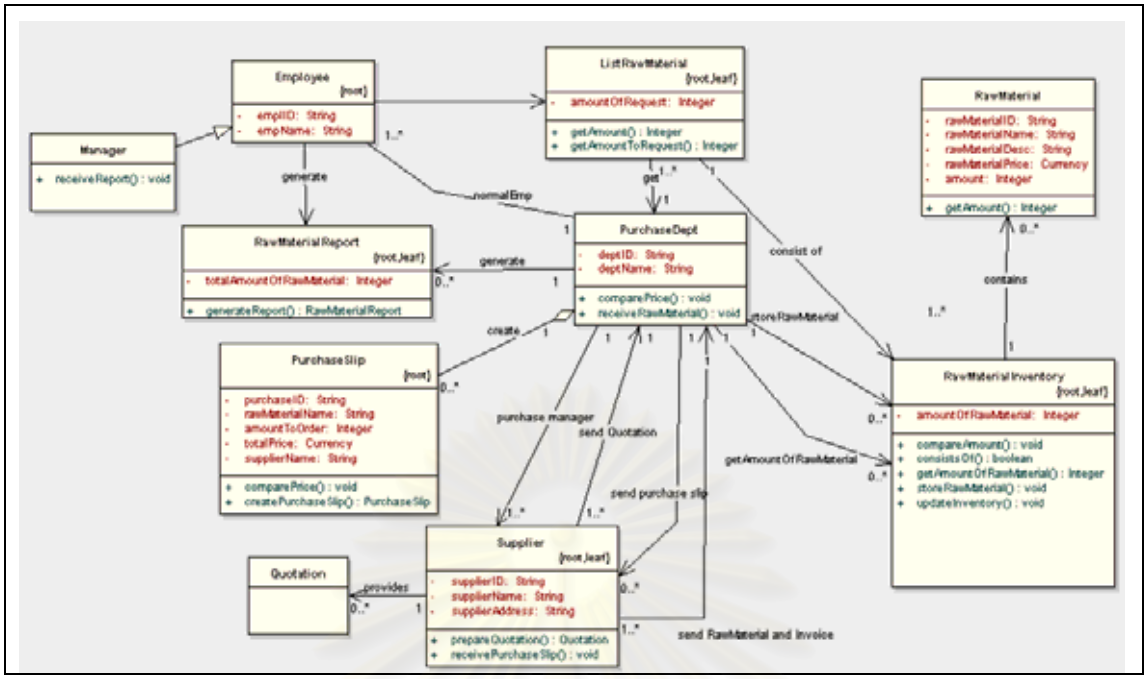


รูปที่ ก.23 แผนภาพคลาสระบบส่วนลดของการขาย

24. ระบบจัดซื้อวัตถุดิบ

ระบบการจัดซื้อวัตถุดิบนี้เป็นระบบที่นำมาใช้แก้ปัญหาการดำเนินงานของบริษัทหนึ่ง ซึ่งมีการจัดทำเอกสารต่าง ๆ ไม่มีรูปแบบ จึงเกิดความล่าช้าในการจัดทำเอกสาร และการสืบค้นราคาวัตถุดิบในการสั่งซื้อแต่ละครั้งไม่สะดวก เกิดความล่าช้า รวมทั้งไม่มีการบันทึกการเบิกจ่ายวัตถุดิบ ระบบที่จะมีหน้าที่ดังต่อไปนี้

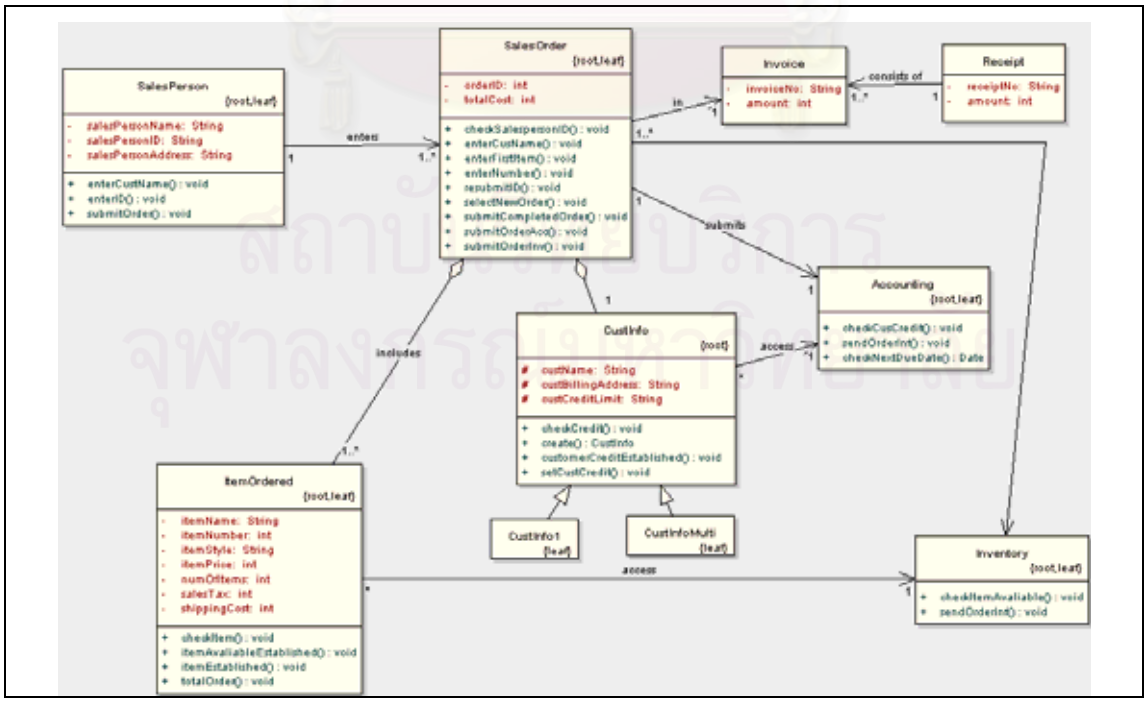
1. ระบบสามารถตรวจสอบยอดคงเหลือของวัตถุดิบในคลังวัตถุดิบได้
2. ระบบสามารถตรวจสอบยอดของวัตถุดิบที่ต้องจัดซื้อได้
3. ระบบสามารถจัดเก็บข้อมูลของผู้จำหน่ายวัตถุดิบให้กับทางปริษัทได้
4. ระบบสามารถสร้างใบสั่งซื้อวัตถุดิบจากผู้จำหน่ายวัตถุดิบได้
5. ระบบสามารถสร้างรายงานของวัตถุดิบคงเหลือในคลังได้
6. ระบบสามารถเพิ่มและแก้ไขข้อมูลต่างๆของวัตถุดิบในคลังวัตถุดิบได้



รูปที่ ก.24 แผนภาพคลาสระบบจัดซื้อวัตถุดิบ

25. ระบบขายสินค้า

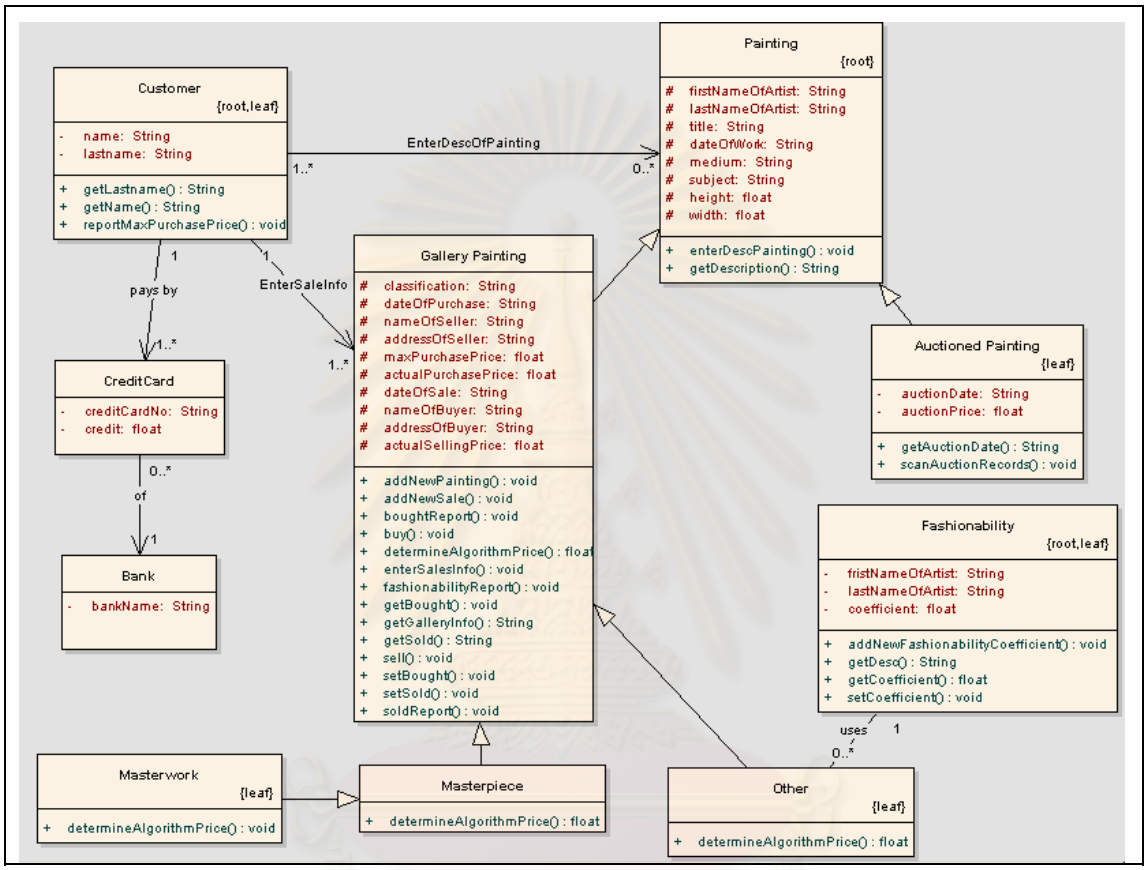
ระบบขายสินค้าเป็นระบบที่ลูกค้าต้องมายื่นรายการสั่งซื้อให้แก่พนักงานขาย จากนั้นฝ่ายบัญชีจะตรวจสอบเครดิตของลูกค้าว่ามีเพียงพอที่จะสั่งซื้อสินค้าได้หรือไม่ นอกจากนั้นระบบยังสามารถตรวจสอบรายการสินค้าในใบสั่งซื้อว่ามีสินค้าอยู่ในคลังพอที่จะขายให้ลูกค้าได้หรือไม่



รูปที่ ก.25 แผนภาพคลาสระบบขายสินค้า

26. ระบบตัวแทนจำหน่ายรูปวาด

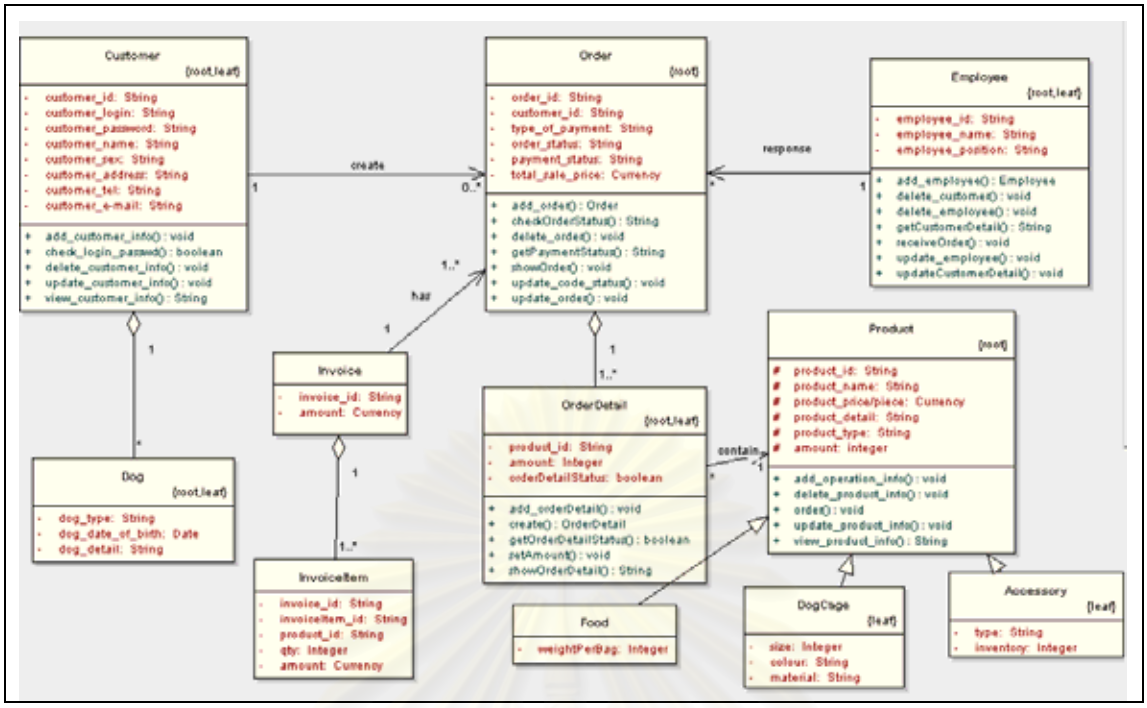
ระบบสามารถรองรับการทำงานของร้านตัวแทนจำหน่ายรูปวาดที่จะเป็นตัวกลางในการซื้อ-ขายรูปภาพ โดยลูกค้าของร้านสามารถจะนำรูปภาพมาขายให้กับทางร้าน หรือมาซื้อรูปภาพจากทางร้านก็ได้ นอกจากนั้น ระบบยังรองรับการทำงานในส่วนของการประมูลรูปภาพนอกเหนือไปจากการซื้อ-ขายภาพทั่วไปด้วย



รูปที่ ก.26 แผนภาพคลาสระบบตัวแทนจำหน่ายรูปวาด

27. ระบบสั่งอุปกรณ์สุนัขออนไลน์

ระบบนี้เป็นระบบสั่งซื้ออุปกรณ์สุนัขผ่านทางอินเตอร์เน็ต เมื่อลูกค้าทำการลงทะเบียนเป็นสมาชิกของระบบ ระบบจะเก็บข้อมูลของลูกค้าเอาไว้ในฐานข้อมูล ดังนั้นลูกค้าจะสามารถสั่งซื้อของผ่านทางเว็บไซต์ของระบบ สามารถตรวจสอบรายการสั่งซื้อของว่าสั่งสิ่งใดไปบ้าง และยังสามารถตรวจสอบสถานะของสินค้าเช่น ขณะสินค้าได้ทำการส่งออกจากระบบเรียบร้อยแล้ว เป็นต้น

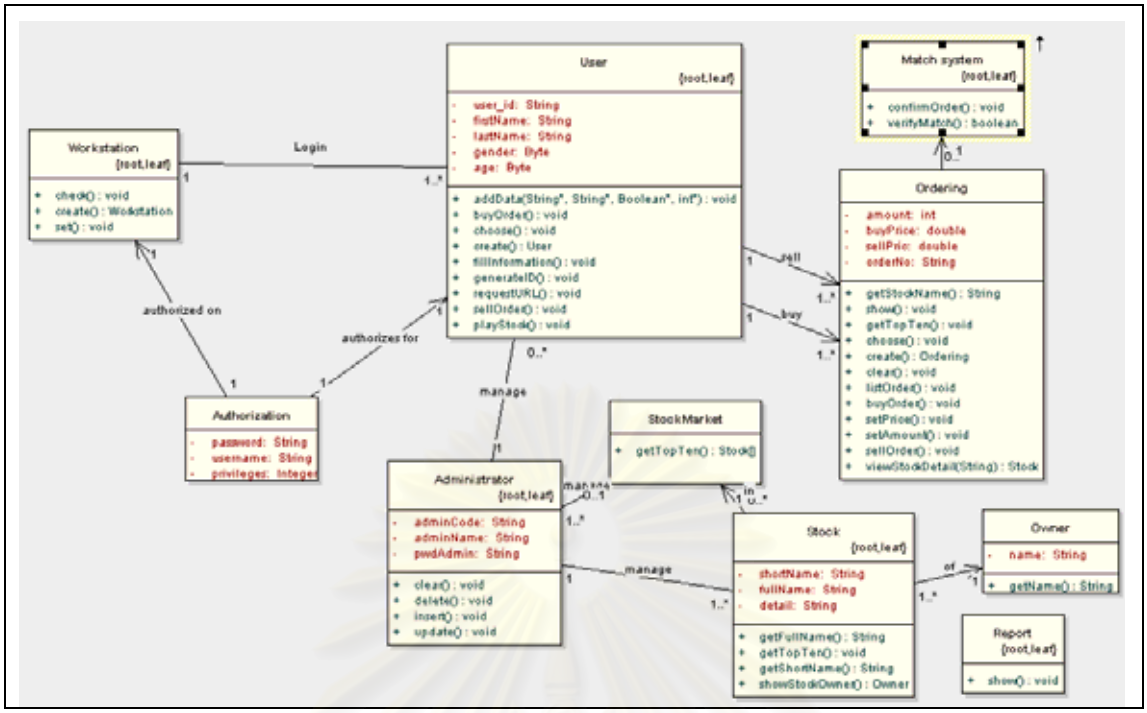


รูปที่ ก.27 แผนภาพคลาสระบบสั่งอุปกรณ์สุนัขออนไลน์

28. ระบบหุ้น

การซื้อขายหุ้นสามารถทำได้โดย ผู้เล่นจะต้องทำการสมัครสมาชิกในกรณีที่ยังไม่เป็นสมาชิก กรณีที่ผู้เล่นเป็นสมาชิกอยู่แล้วจะทำการล็อกอิน(Login) เพื่อเข้าสู่หน้าจอเล่นเกม โดยทางฝั่ง Server จะทำการตรวจสอบรหัสผู้ใช้งานเพื่อเข้าสู่หน้าจอของผู้เล่นคนนั้น การเล่นหุ้น ผู้เล่นสามารถทำการซื้อ-ขายหุ้นผ่านอินเตอร์เน็ตโดยสามารถส่งคำสั่งซื้อและคำสั่งขายไป โดยทางเซิร์ฟเวอร์จะ ทำการนำคำสั่งซื้อและคำสั่งขายของผู้เล่นแต่ละคนมาทำการจับคู่กัน และเกิดการซื้อ-ขายกันขึ้น โดยประกอบด้วย ราคาซื้อ, ราคาขาย และจำนวนหุ้นที่ทำการซื้อขาย การดึงข้อมูลหุ้นจากตลาดหลักทรัพย์จะทำโดยอัตโนมัติ และจะทำการยกเลิกคำสั่งซื้อ-คำสั่งขายทุกวัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

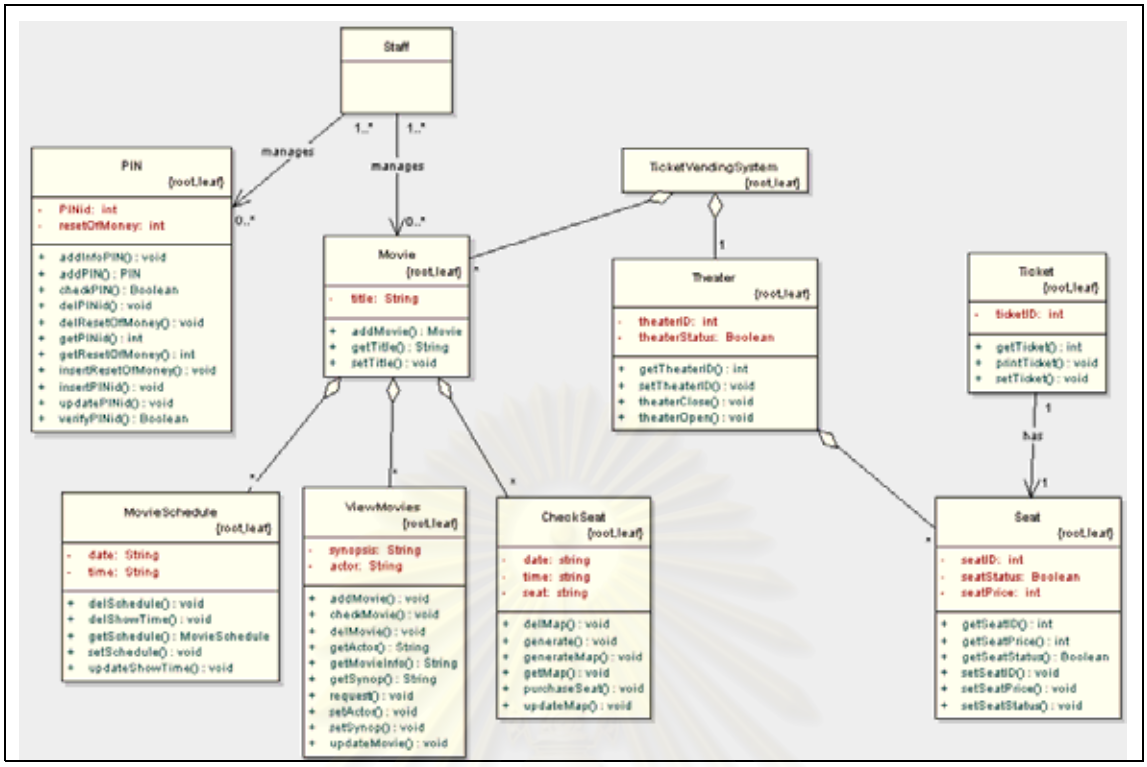


รูปที่ ก.28 แผนภาพคลาสระบบหุ้น

29. ระบบขายตั๋วหนังออนไลน์

ระบบนี้สามารถขายตั๋วหนังแบบออนไลน์ โดยลูกค้าจะต้องทำการซื้อรหัสส่วนตัวด้วยจำนวนเงินตามที่ต้องการ ภายหลังการซื้อตั๋วหนังออนไลน์ ตัวเลขที่แสดงจำนวนเงินในรหัสส่วนตัวนี้ จะลดลงตามราคาตั๋วที่ซื้อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

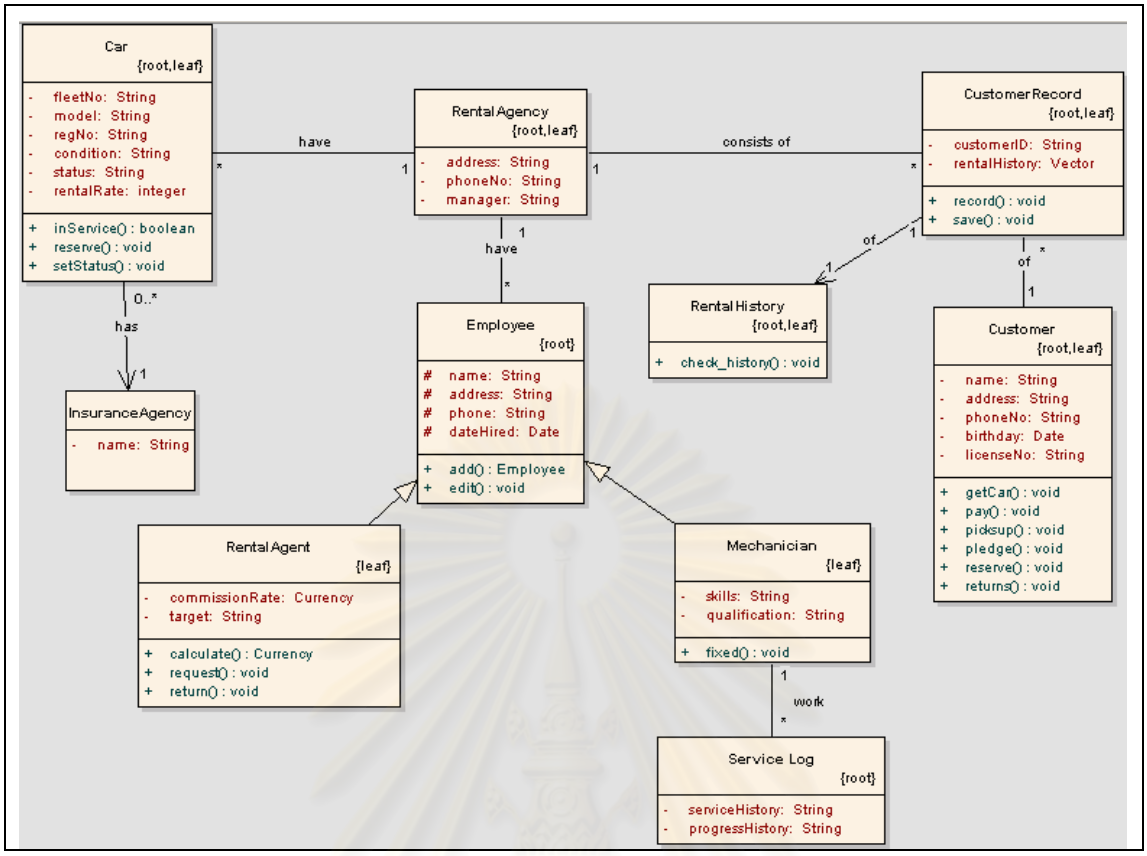


รูปที่ ก.29 แผนภาพคลาสระบบขายตั๋วหนังออนไลน์

30. ระบบเช่ารถ

ระบบเช่ารถสามารถทำฟังก์ชันหลักๆ ได้ดังนี้

1. ลูกค้าสามารถจองรถ โดยก่อนรับรถ ลูกค้าสามารถจอง โดยติดต่อบริษัทตัวแทน และทำการร้องขอ บริษัทตัวแทนจะรับหรือปฏิเสธคำร้องขอโดยอยู่บนพื้นฐานของเงื่อนไขต่างๆ เช่น สถานะของรถว่างหรือไม่ หรือประวัติการเช่ารถของลูกค้า ถ้าการของได้รับการยอมรับ บริษัทตัวแทนจะกรอกฟอร์มซึ่งประกอบด้วยรายละเอียดลูกค้า การจองจะสมบูรณ์โดยการวางเงินมัดจำ
2. ลูกค้ามารับรถ เมื่อลูกค้ามาถึงบริษัทตัวแทน บริษัทจะจัดรถที่ลูกค้าร้องขอไว้ให้ เมื่อลูกค้าจ่ายเงินค่าเช่ารถครบสมบูรณ์ลูกค้าจะได้รับรถไป
3. ลูกค้านำรถมาคืน ลูกค้านำรถมาคืนเมื่อถึงวันที่กำหนดคืนในสัญญาการเช่ารถ

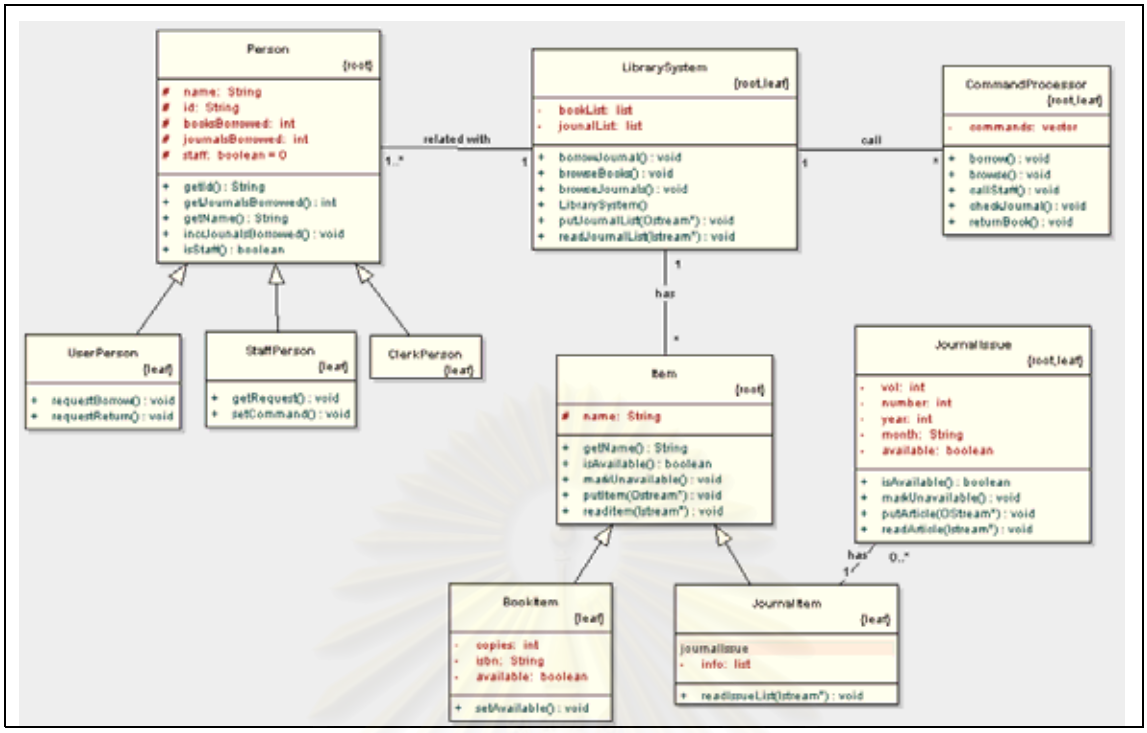


รูปที่ ก.30 แผนภาพคลาสระบบเช่ารถ

31. ระบบ Library

ระบบจะประกอบไปด้วยผู้ยืมหนังสือ เจ้าหน้าที่ให้บริการ และประกอบด้วยการบริการให้การยืมหนังสือ และการคืนหนังสือ โดยมีรายละเอียดดังนี้ ในการยืมหนังสือนั้น เจ้าหน้าที่จะมีหน้าที่ให้บริการยืมหนังสือ ถ้าหนังสือที่ผู้ยืมต้องการยืมยังไม่มีผู้อื่นยืมไปในขณะนั้น ผู้ยืมจะสามารถยืมหนังสือได้ ส่วนในกรณีของการคืนหนังสือ ก็จะทำการเพิ่มข้อมูลของรายการหนังสือที่นำมาคืนจนหมด ถึงจะสิ้นสุดขั้นตอนการคืนหนังสือ

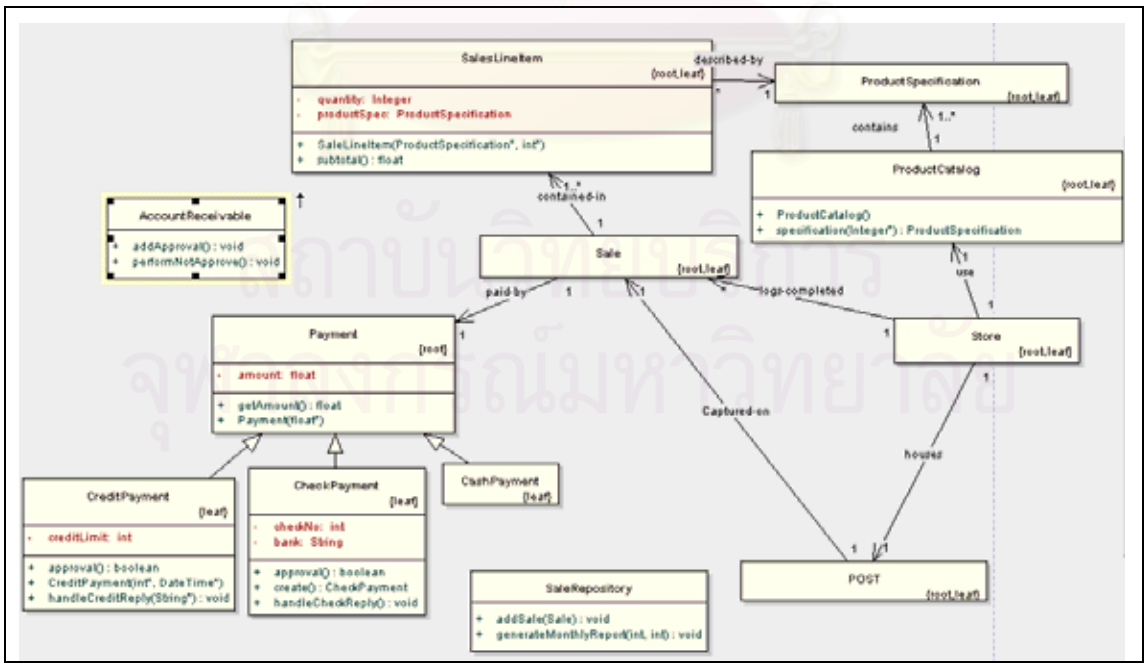
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ ก.31 แผนภาพคลาสระบบ Library

32. ระบบคงคลังสินค้า

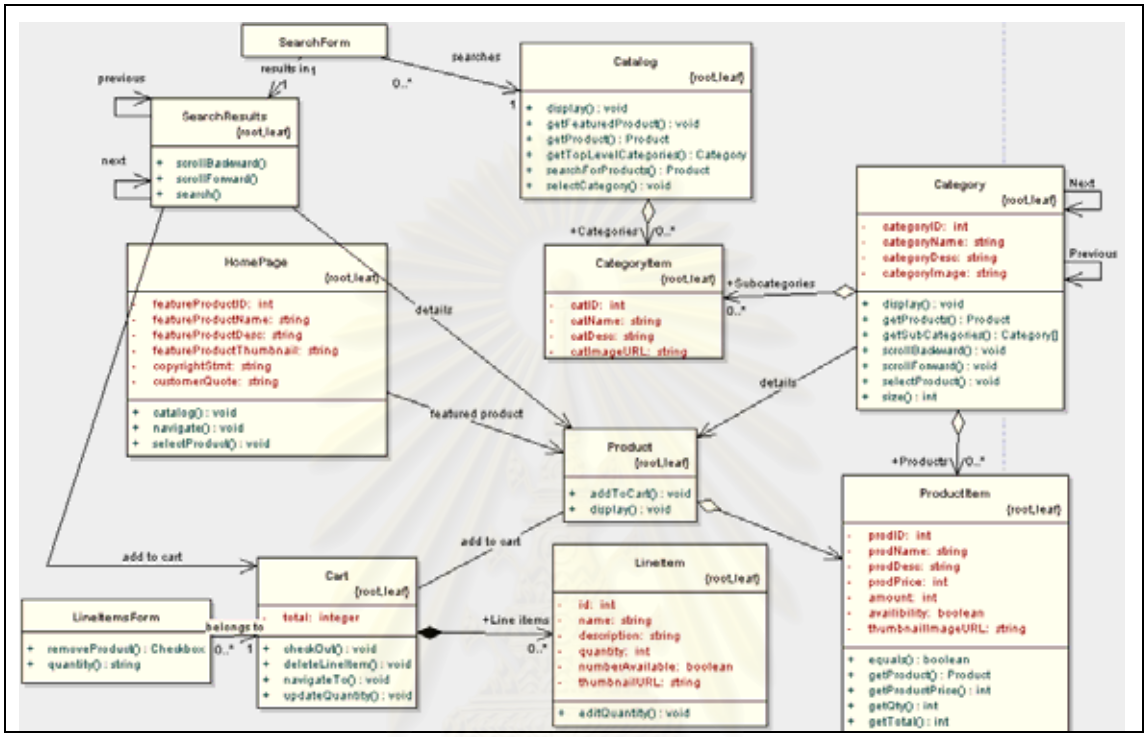
ระบบคงคลังสินค้า เป็นระบบการขายสินค้าภายในองค์กร โดยจะเก็บข้อมูลสินค้าและรายละเอียดของสินค้าภายในคงคลังทั้งหมด ซึ่งในการขายสินค้าทุกครั้ง จะต้องทำการบันทึกข้อมูลจำนวนสินค้าทุกครั้ง และลูกค้าสามารถจ่ายเงินได้ 3 วิธี คือ เครดิต, เช็ค และ เงินสด



รูปที่ ก.32 แผนภาพคลาสระบบคงคลังสินค้า

33. ระบบสินค้าบนเว็บ

ระบบนี้เป็นระบบสำหรับการซื้อ-ขายสินค้าผ่านทางอินเตอร์เน็ตโดยให้ลูกค้าสามารถทำการสั่งซื้อสินค้าผ่านหน้าจอเว็บบราวเซอร์ได้ ซึ่งบริการที่ระบบนี้สนับสนุน ได้แก่ การค้นหารายการสินค้า และการสั่งซื้อสินค้า โดยจะมี ตะกร้าคอยเก็บข้อมูลของสินค้าที่ลูกค้าเลือกไว้

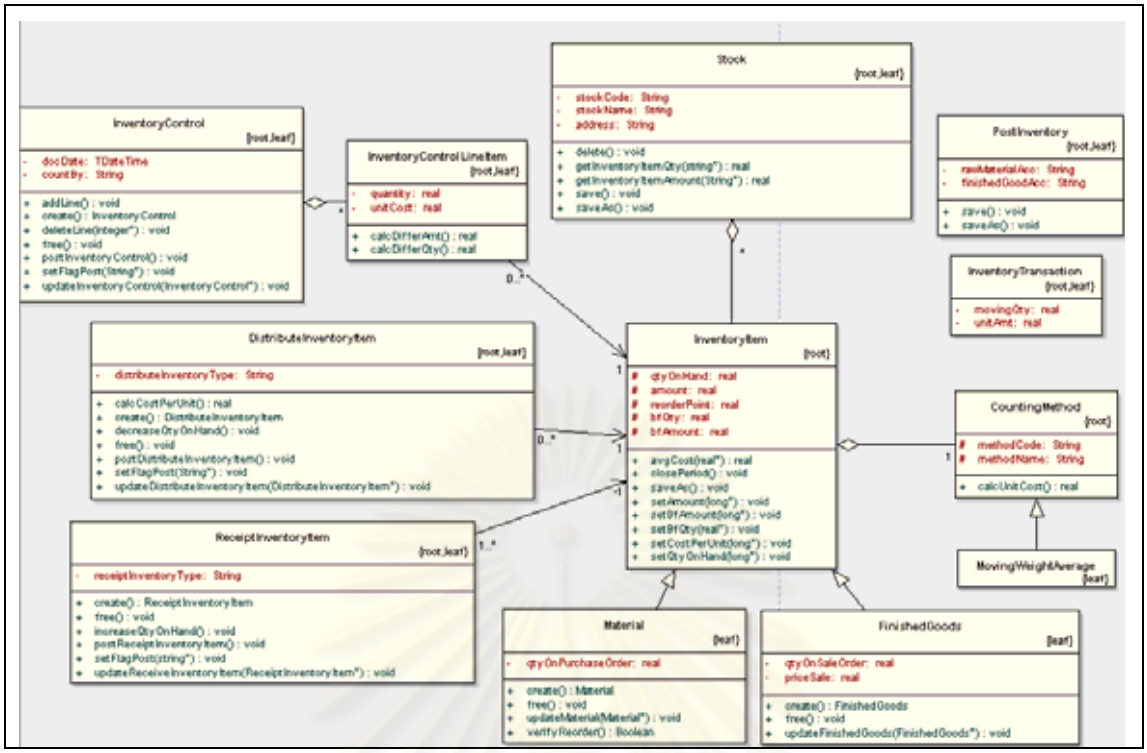


รูปที่ ก.33 แผนภาพคลาสระบบสินค้าบนเว็บ

34. ระบบกลุ่มงานสินค้าคงคลัง

กลุ่มงานสินค้าคงคลัง ประกอบด้วยงานสำหรับการควบคุมการรับ-จ่ายสินค้าและวัตถุดิบ จัดทำบัญชีย่อยสินค้าเพื่อใช้ในการควบคุมความเคลื่อนไหวของสินค้าคงคลัง รวมทั้งปรับปรุงปริมาณสินค้าคงคลังจากการตรวจนับ โดยมีงานต่างๆ ดังนี้

1. การติดตั้งงานสินค้าคงคลัง ติดตั้งข้อมูลบัญชีคุมยอดที่ใช้ในการผ่านรายการบัญชีสำหรับระบบงานสินค้าคงคลัง และข้อมูลคลังสินค้า
2. การทำบัญชีย่อยสินค้าและวัตถุดิบ บันทึกข้อมูลรายละเอียดสินค้าและวัตถุดิบภายในคลังสินค้าแต่ละชนิด
3. การรับสินค้าและวัตถุดิบเข้าคลังสินค้า บันทึกการรับสินค้าและวัตถุดิบเข้าคลังสินค้าแต่ละชนิด
4. การจำหน่ายสินค้าและเบิกวัตถุดิบออกจากคลังสินค้า
5. การตรวจนับสินค้า
6. การสร้างรายงานสินค้าคงคลัง

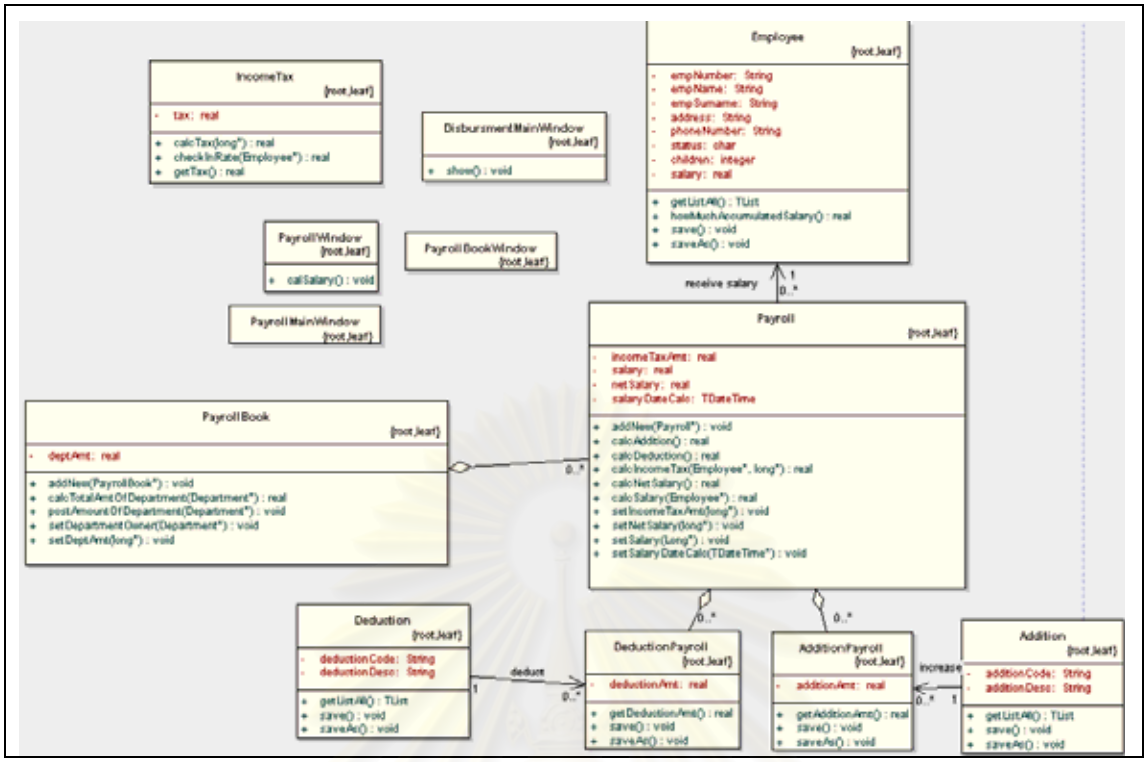


รูปที่ ก.34 แผนภาพคลาสระบบกลุ่มงานสินค้าคงคลัง

35. ระบบงานเงินเดือน

ระบบงานเงินเดือน ประกอบด้วยการจัดการข้อมูลพนักงานในส่วนหักและส่วนเงินเพิ่ม เพื่อใช้ในการคำนวณเงินเดือนและทำการคำนวณเงินเดือนรวมทั้งการทำบัญชีเงินได้พนักงานซึ่งเป็นบัญชีย่อยของกลุ่มงานนี้ มีฟังก์ชันการทำงานดังนี้

1. การติดตั้งงานเงินเดือน ทำหน้าที่ติดตั้งข้อมูลเพิ่มและหักเงินเดือนขององค์กรสำหรับระบบงานเงินเดือน
2. การสร้างและปรับปรุงข้อมูลเงินเดือนพนักงาน ทำหน้าที่บันทึกรายละเอียดด้านเงินเดือนพนักงาน โดยรับข้อมูลจากพนักงานที่ดูแลเรื่องเงินเดือน
3. การคำนวณเงินเดือนพนักงาน ทำหน้าที่คำนวณเงินเดือนพนักงานและจัดทำบัญชีเงินได้พนักงาน รวมทั้งการทำการบันทึกรายการบัญชีเพื่อเบิกจ่ายเงินเดือนโดยใช้ระบบงานย่อยเช็คสั่งจ่าย เพื่อจัดทำใบสำคัญสั่งจ่ายอื่นๆ
4. การสร้างรายงานเงินเดือน ทำหน้าที่จัดทำรายงานด้านเงินเดือน



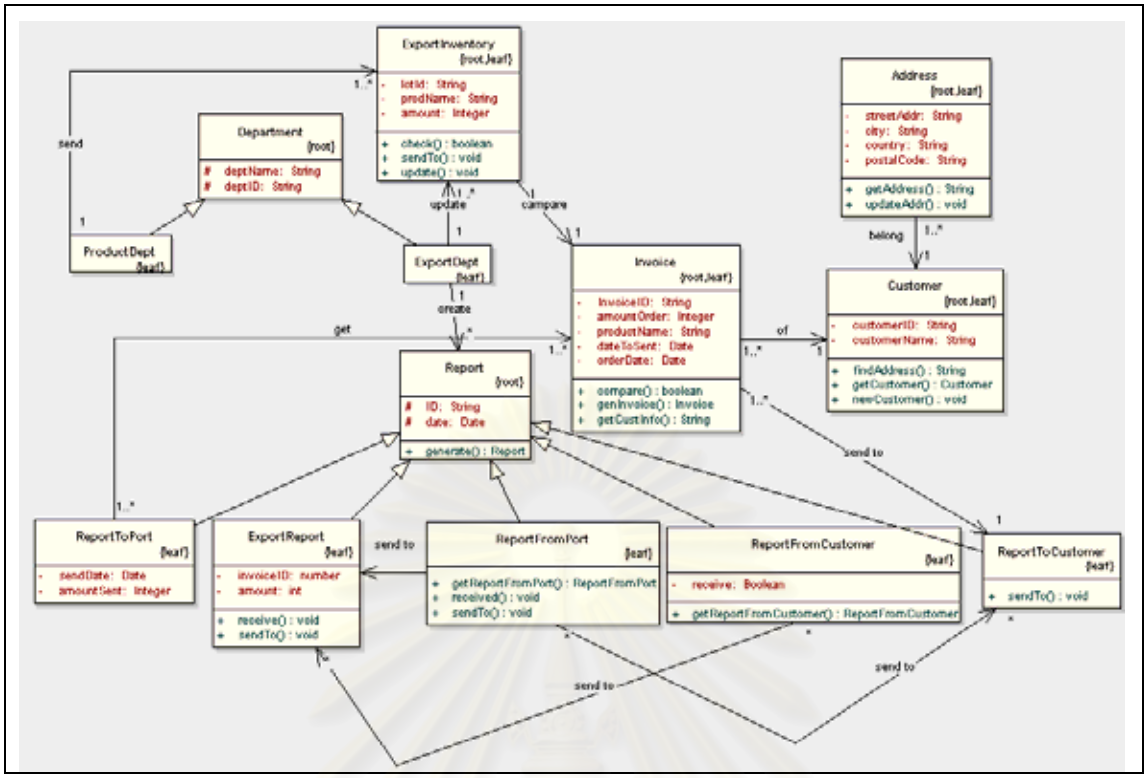
รูปที่ ก.35 แผนภาพคลาส ระบบงานเงินเดือน

36. ระบบการส่งออก

ระบบสินค้าคงคลังและการส่งออกนี้เป็นระบบที่นำมาใช้แก้ปัญหาการดำเนินงานของบริษัทหนึ่ง ซึ่งการตรวจสอบจำนวนสินค้าที่ผลิตเสร็จเพื่อรอการส่งออกมีความล่าช้าเพราะไม่มีระบบบันทึกจำนวนสินค้าในคลังสินค้า และการติดต่อกับกรมศุลกากรยังใช้เอกสารในการติดต่อทำให้เกิดมีความล่าช้าในการติดต่อ ระบบมีหน้าที่ดังต่อไปนี้

1. ระบบสามารถบันทึกจำนวนสินค้าเข้าออกในคลังสินค้าได้
2. ระบบสามารถตรวจสอบจำนวนสินค้าในคลังสินค้ากับจำนวนสินค้าที่ลูกค้าสั่งได้
3. ระบบสามารถออกเอกสารรายละเอียดการส่งสินค้าได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

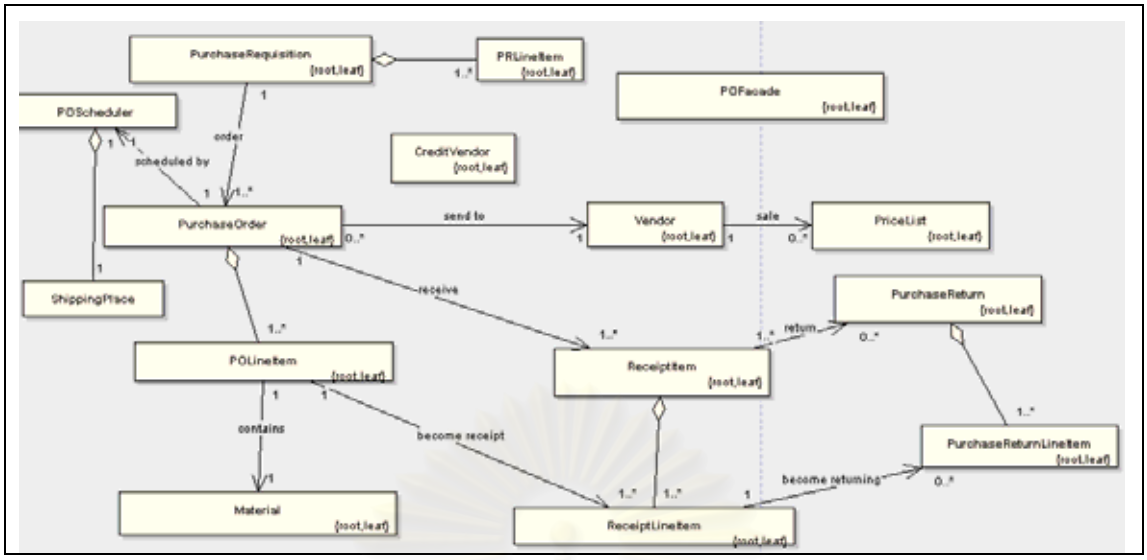


รูปที่ ก.36 แผนภาพคลาสระบบการส่งออก

37. ระบบกลุ่มงานสั่งซื้อ

งานสั่งซื้อ ประกอบด้วยการรับรายการความต้องการสั่งซื้อ การจัดทำรายการสั่งซื้อส่งให้กับผู้ขาย และการแจ้งปริมาณการสั่งซื้อให้กับงานสินค้าคงคลัง โดยมีฟังก์ชันการทำงานดังนี้

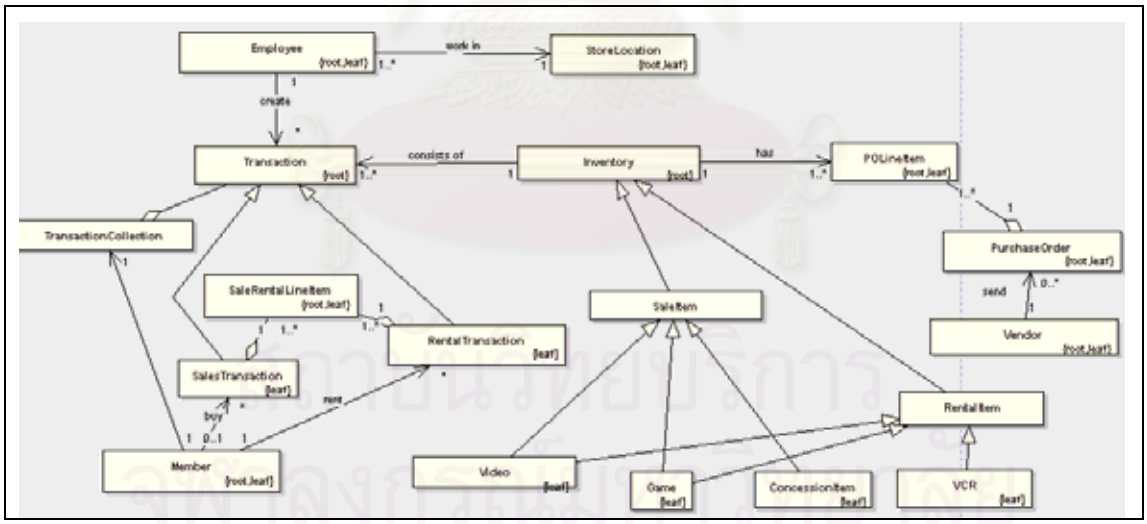
1. การติดตั้งค่าเริ่มต้นงานสั่งซื้อ ทำหน้าที่ติดตั้งเลขที่เริ่มต้นของเอกสารของงานสั่งซื้อที่ใช้ในองค์การ คือ ใบเสนอซื้อ ใบสั่งซื้อ ใบตรวจรับสินค้า และใบส่งสินค้า
2. การสร้างและปรับปรุงข้อมูลผู้ขาย ทำหน้าที่ป้อนข้อมูลผู้ขายและรายการสินค้าขาย
3. การรับความต้องการซื้อ ทำหน้าที่บันทึกรายการเสนอซื้อจากคลังสินค้าโดยรับข้อมูลจากคลังสินค้า และทำการพิมพ์เอกสารใบเสนอซื้อ
4. การสั่งซื้อ ทำหน้าที่บันทึกรายการสั่งซื้อ ทำการตรวจสอบข้อมูลเจ้าหนี้จากงานเจ้าหนี้กรณีเป็นการซื้อเชื่อ และทำการส่งข้อมูลปริมาณการสั่งซื้อให้ระบบงานย่อยสินค้าเพื่อปรับปรุงปริมาณสินค้าที่กำลังสั่งซื้อ ทำการพิมพ์เอกสารใบสั่งซื้อส่งให้ผู้ขาย
5. การตรวจรับวัตถุดิบ ทำหน้าที่บันทึกรายการตรวจรับวัตถุดิบจากการสั่งซื้อ
6. การจัดทำรายการคืนสินค้า ทำหน้าที่บันทึกรายการส่งคืนสินค้า และทำการส่งข้อมูลปริมาณการส่งคืนให้ระบบงานย่อยคลังสินค้า เพื่อปรับปรุงปริมาณสินค้า



รูปที่ ก.37 แผนภาพคลาสระบบกลุ่มงานสั่งซื้อ

38. ระบบร้านเช่าวีดีโอ

ร้านค้าวีดีโอนี้ มีบริการสำหรับซื้อ ได้แก่วีดีโอ, เกมส์ และคอนเซสชัน (Concession) และบริการสำหรับเช่าวีดีโอ, เกมส์ และวีซีอาร์ ลูกค้าที่สามารถมาใช้บริการเช่ารายการสินค้าต่างๆ ภายในร้าน จะต้องเป็นสมาชิกของร้านก่อนจึงจะสามารถเช่าสินค้าได้ โดยจะทำการเก็บข้อมูลการซื้อและเช่ารายการสินค้าทุกชนิดภายในร้านทุกครั้งด้วย



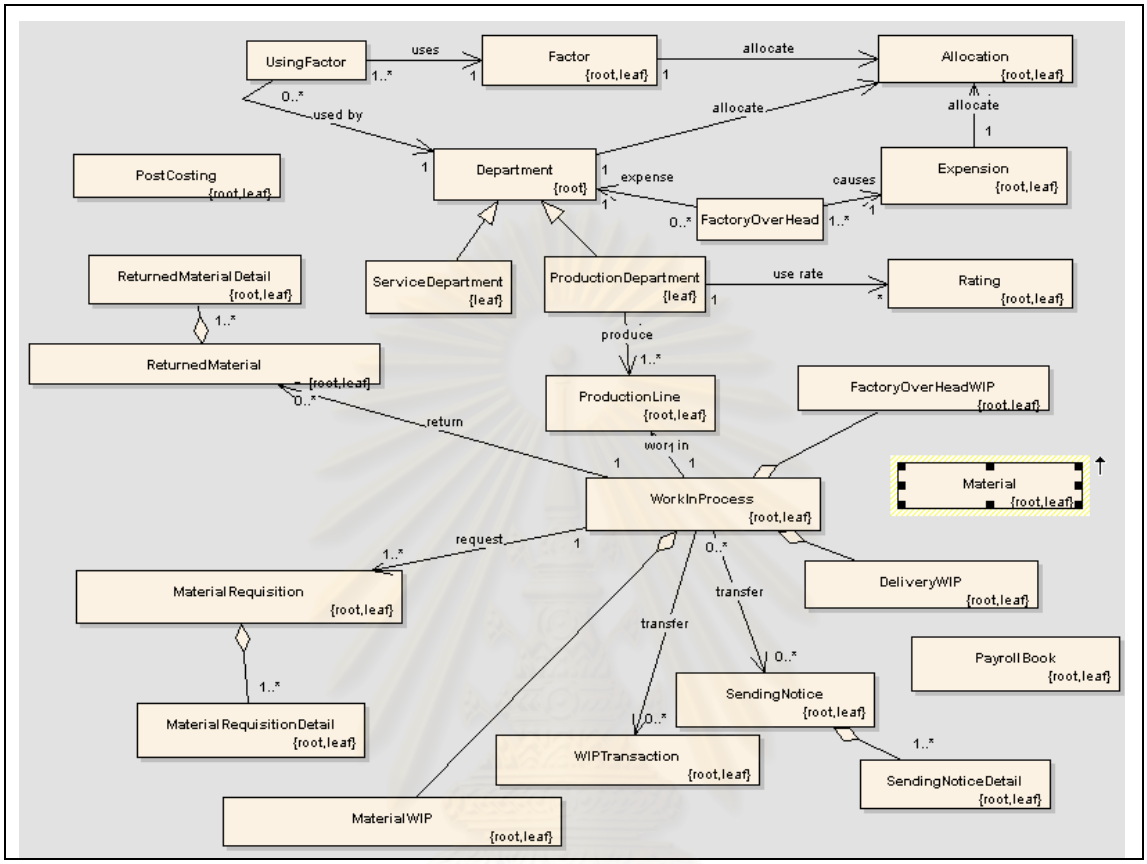
รูปที่ ก.38 แผนภาพคลาสระบบร้านเช่าวีดีโอ

39. ระบบงานต้นทุนการผลิต

ระบบต้นทุนการผลิต เป็นระบบที่นำวัตถุดิบมาเข้าสายงานการผลิตสินค้า โดยมีฟังก์ชันการทำงานดังนี้

1. มีการเบิกวัตถุดิบเข้าคลังสินค้า เพื่อทำการผลิตสินค้า

2. รับวัตถุดิบที่เหลือจากการผลิตกลับคืน
3. จัดส่งสินค้าไปยังลูกค้า
4. ทำระบบบัญชีการคำนวณเงินเดือนของพนักงานภายในบริษัท

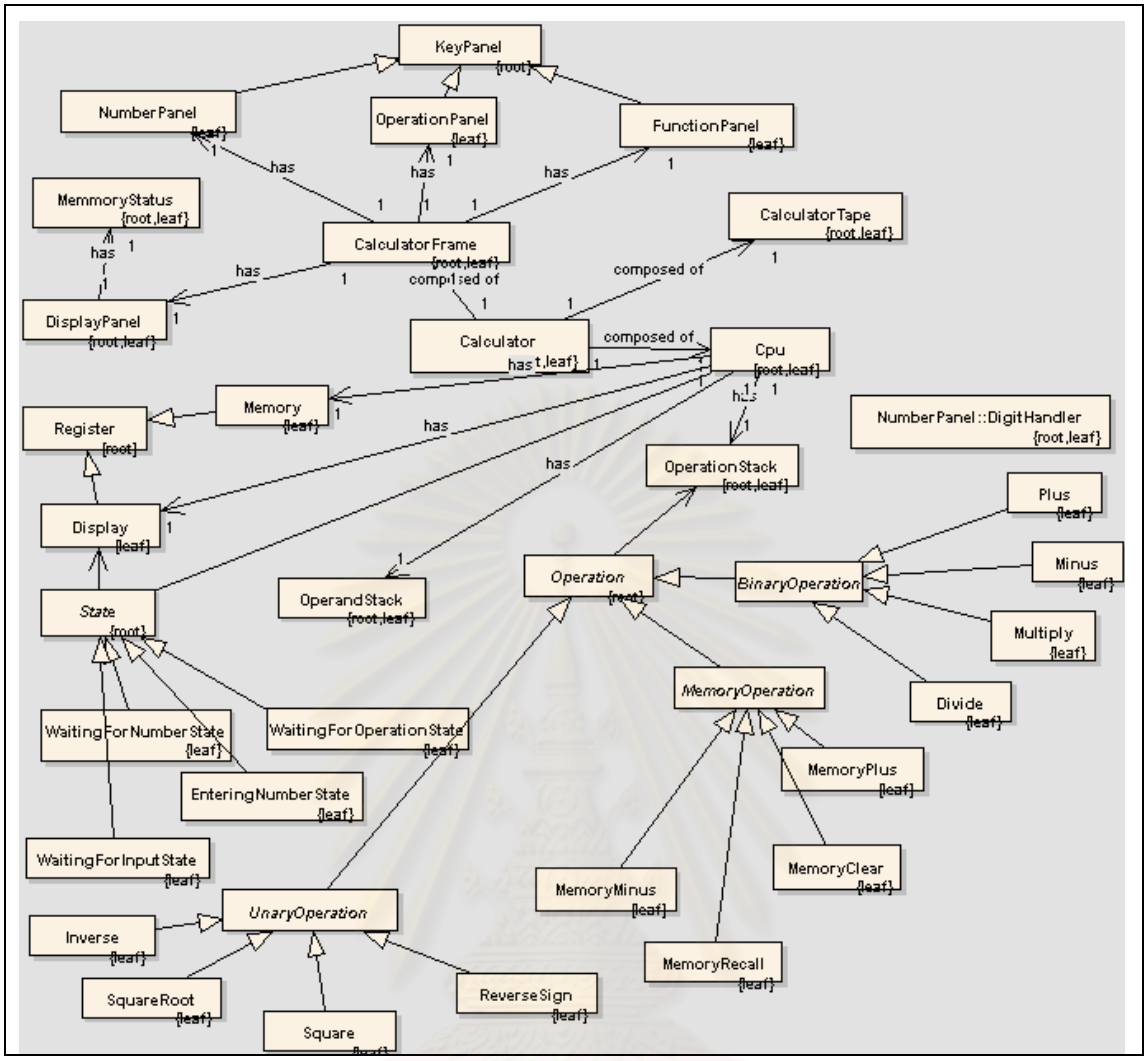


รูปที่ ก.39 แผนภาพคลาสระบบงานต้นทุนการผลิต

40. ระบบ Calculator

ระบบเครื่องคิดเลขเป็นระบบที่แสดงถึงโครงสร้างของเครื่องคิดเลข และส่วนประกอบต่างๆ ภายในเครื่องคิดเลข โดยมีฟังก์ชันการทำงานดังนี้

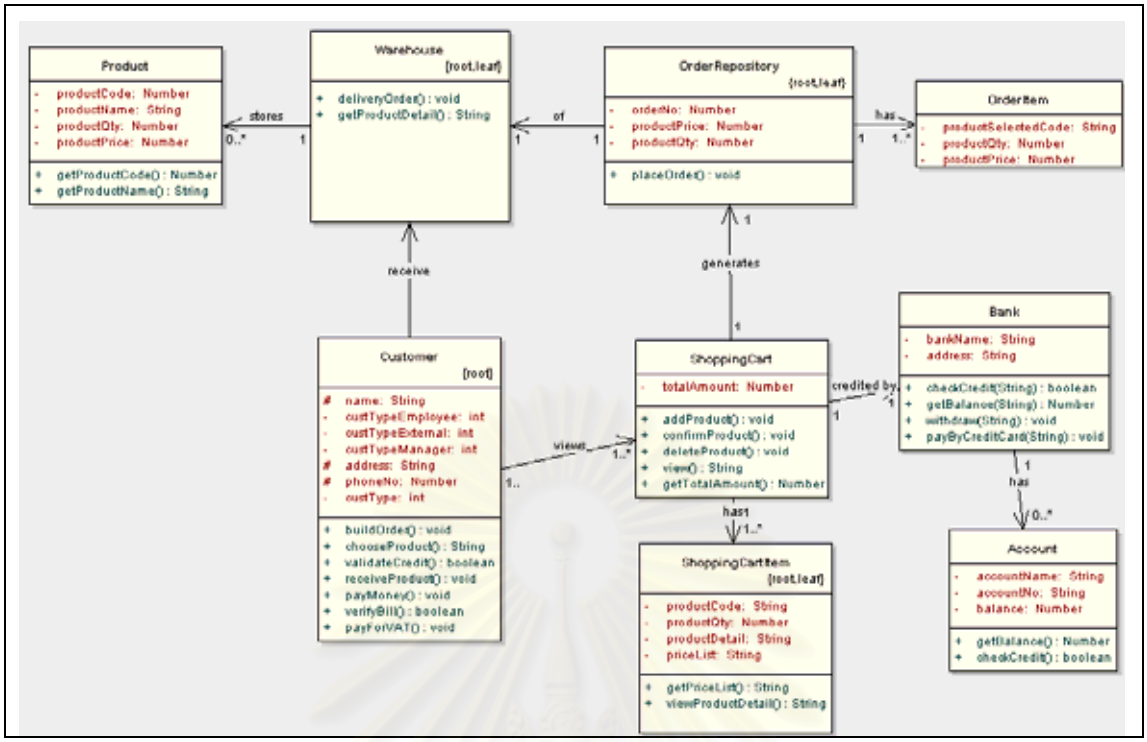
1. สามารถคำนวณไบนารีโอเปอร์เรชัน (Binary operation) ได้
2. สามารถคำนวณยูนารีโอเปอร์เรชัน (Unary operation) ได้



รูปที่ ก.40 แผนภาพคลาสระบบ Calculator

41. ระบบซื้อขายสินค้าผ่านเว็บ

ระบบขายสินค้าผ่านเว็บ มีฟังก์ชันให้ลูกค้าเลือกชมสินค้า และสั่งซื้อสินค้าผ่านบัตรเครดิต หลังจากลูกค้าสั่งรายการซื้อสินค้าเรียบร้อยแล้ว คำสั่งซื้อที่ถูกยืนยันจะถูกส่งไปยังคลังสินค้า เพื่อจัดการส่งสินค้าให้กับลูกค้าต่อไป

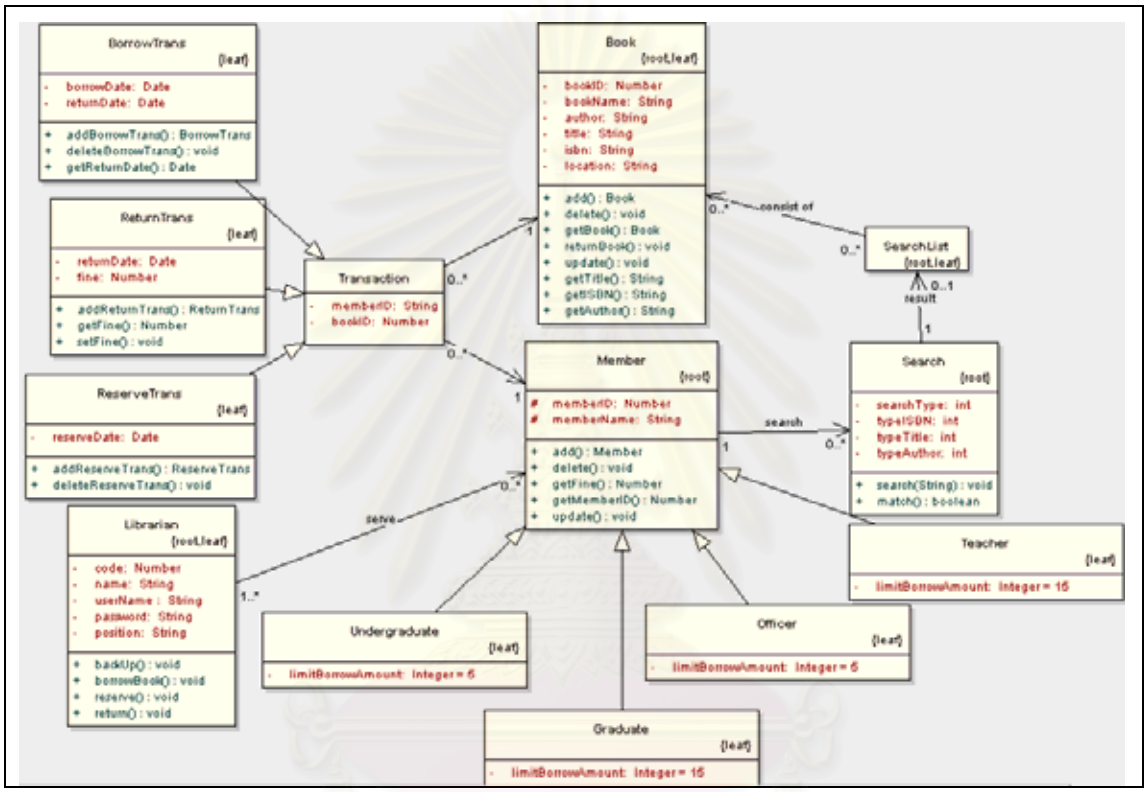


รูปที่ ก.41 แผนภาพคลาสระบบซื้อขายสินค้าผ่านเว็บ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

42. ระบบห้องสมุด

ระบบห้องสมุดพัฒนาเพื่อสนับสนุนงานการบริการห้องสมุดให้มีประสิทธิภาพการสืบค้นให้รวดเร็วยิ่งขึ้น ในการสืบค้นสามารถสืบค้นได้ด้วยรหัสไอเอสบีเอ็น (ISBN), ชื่อผู้แต่ง, และชื่อหนังสือ นอกจากนี้ระบบห้องสมุดยังให้บริการยืม-คืนหนังสือ, การจองหนังสือ, การตรวจสอบรายชื่อหนังสือที่สมาชิกค้างส่งได้, การคำนวณค่าปรับหนังสือเกินกำหนด และช่วยให้การแก้ไขข้อมูลต่างๆ ที่เกี่ยวกับระบบสามารถทำได้ง่าย



รูปที่ ก.42 แผนภาพคลาสระบบห้องสมุด

ในงานวิจัยนี้ได้แบ่งกลุ่มตัวอย่างสำหรับใช้เป็นเซ็ตทูนนิงเพื่อหาค่าช่วงที่เหมาะสมของมาตรวัดและเซ็ตแวลิเดชันสำหรับทดสอบกลยุทธ์การตรวจจับข้อบกพร่อง โดยแสดงในตารางที่ ก.1 ซึ่งจากตาราง T หมายถึงการใช้ระบบตัวอย่างนั้นเป็นเซ็ตทูนนิงและ V หมายถึงการใช้ระบบตัวอย่างนั้นเป็นเซ็ตแวลิเดชัน ส่วนช่องที่ว่างหมายถึงระบบตัวอย่างไม่ได้ถูกใช้งาน

ตารางที่ ก.1 การแบ่งกลุ่มตัวอย่างเป็นเซ็ตทูนนิงและเซ็ตแวลิเดชันสำหรับข้อบกพร่องแต่ละประเภท

ระบบตัวอย่าง	Data Class	Feature Envy	Message Chains	Middle Man	God Class	Switch Statements
1	T	T	T	T	T	
2	T	T	T	T	T	V
3	T	T	T	T	T	V
4	T	T	T	T	T	V
5	T	T	T	T	T	T
6	T	T	T	T	T	T
7	T	T	T	T	T	T
8	V	T	T	T	T	T
9	T	T	T	T	T	T
10	T	T	T	T	T	T
11	T	T	T	T	T	T
12	T	T	T	T	T	T
13	T	T	T	T	T	T
14	T	T	T	T	T	T
15	T	T	T	T	T	T
16	T	T	T	T	T	T
17	T	T	T	T	T	
18	T	T	T	T	T	T
19	T	T	T	T	T	T
20	T	T	T	T	T	T
21	T	T	T	T	T	T
22	T	T	T	T	T	T
23	T	T	T	T	T	T
24	T	T	T	T	T	T
25	T	T	T	T	T	T
26	T	T	T	T	T	T
27	T	T	T	T	T	T
28	T	T	T	T	T	T
29	T	T	T	T	T	T
30	T	T	T	T	T	T

ตารางที่ ก.1(ต่อ) การแบ่งกลุ่มตัวอย่างเป็นเซ็ตทูนนิงและเซ็ตแวลิเดชันสำหรับข้อบกพร่องแต่ละประเภท

ระบบตัวอย่าง	Data Class	Feature Envy	Message Chains	Middle Man	God Class	Switch Statements
31	T	T	T	T	T	T
32	T	T	T	T	T	T
33	T	T	T	V	T	T
34	T	T	T	T	T	T
35	T	T	T	T	T	T
36	T	V	V	T	V	T
37	V	V	V	V	V	T
38	V	V	V	V	V	T
39	V	V	V	V	V	T
40	V	V	V	V	V	T
41						V
42						V

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

ผลการทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบ

ในบทนี้จะแสดงผลการทดสอบการตรวจจับข้อบกพร่องของโมเดลการออกแบบแต่ละประเภท โดยจะแสดงค่าของมาตรวัดสำหรับข้อบกพร่องแต่ละเภท ก่อนและหลังการทำรีแฟคทอริง ดังมีรายละเอียดดังนี้

1.ผลการทดสอบการตรวจจับข้อบกพร่องประเภท Data Class

คลาสจะมีข้อบกพร่องประเภท Data Class เมื่อมีค่ามาตรวัด NNorm เท่ากับ 0 จากตารางค่ามาตรวัดที่มีค่าเป็นลบหมายถึงคลาสนั้นไม่มีเมทธอดใด ๆ เลยหรือไม่มีแอคเซสเซอร์เมทธอดเลย

ค่ามาตรวัดก่อนทำรีแฟคทอริงมีดังนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.1 แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex8.xml	FoodList	1	Not bad	20	1	0	1.288
Ex8.xml	Waiter	-1	Not bad	20	1	0	1.288
Ex8.xml	Order	3	Not bad	20	1	0	1.288
Ex8.xml	Queue	-1	Not bad	20	1	0	1.288
Ex8.xml	Bill	2	Not bad	20	1	0	1.288
Ex8.xml	Table	1	Not bad	20	1	0	1.288
Ex8.xml	Cashier	-1	Not bad	20	1	0	1.288
Ex8.xml	Customer	4	Not bad	20	1	0	1.288
Ex8.xml	FoodInfo	0	Bad	20	1	0	1.288
Ex8.xml	Chief	-1	Not bad	20	1	0	1.288
Ex37.xml	Material	9	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturnLineItem	-1	Not bad	23	5	0	0.832
Ex37.xml	ReceiptItem	-1	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	16	Not bad	23	5	0	0.832
Ex37.xml	ReceiptLineItem	-1	Not bad	23	5	0	0.832
Ex37.xml	CreditVendor	0	Bad	23	5	0	0.832

ตารางที่ ข.1(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	POScheduler	-1	Not bad	23	5	0	0.832
Ex37.xml	Vendor	-1	Not bad	23	5	0	0.832
Ex37.xml	POFacade	-1	Not bad	23	5	0	0.832
Ex37.xml	PONewWindow	-1	Not bad	23	5	0	0.832
Ex37.xml	POLineItem	-1	Not bad	23	5	0	0.832
Ex37.xml	PostRecWindow	-1	Not bad	23	5	0	0.832
Ex37.xml	PurchaseRequisition	-1	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	3	Not bad	23	5	0	0.832
Ex37.xml	PostPRWindow	-1	Not bad	23	5	0	0.832
Ex37.xml	PriceList	-1	Not bad	23	5	0	0.832
Ex37.xml	PurchasingDocumentControl	-1	Not bad	23	5	0	0.832
Ex37.xml	PRLineItem	-1	Not bad	23	5	0	0.832
Ex37.xml	ShippingPlace	0	Bad	23	5	0	0.832
Ex37.xml	PostPOWindow	-1	Not bad	23	5	0	0.832
Ex37.xml	PostReturnWindow	-1	Not bad	23	5	0	0.832

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.1(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Member	-1	Not bad	57	4	2	0.946
Ex38.xml	RentalTransaction	-1	Not bad	57	4	2	0.946
Ex38.xml	POLineItem	-1	Not bad	57	4	2	0.946
Ex38.xml	Vendor	-1	Not bad	57	4	2	0.946
Ex38.xml	Video	-1	Not bad	57	4	2	0.946
Ex38.xml	PurchaseOrder	-1	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	-1	Not bad	57	4	2	0.946
Ex38.xml	StoreLocation	-1	Not bad	57	4	2	0.946
Ex38.xml	RentalItem	-1	Not bad	57	4	2	0.946
Ex38.xml	SalesTransaction	-1	Not bad	57	4	2	0.946
Ex38.xml	Transaction	-1	Not bad	57	4	2	0.946
Ex38.xml	TransactionCollection	-1	Not bad	57	4	2	0.946
Ex38.xml	ConcessionItem	-1	Not bad	57	4	2	0.946
Ex38.xml	Inventory	-1	Not bad	57	4	2	0.946
Ex38.xml	Game	-1	Not bad	57	4	2	0.946
Ex38.xml	Employee	-1	Not bad	57	4	2	0.946

ตารางที่ ข.1(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	SaleItem	-1	Not bad	57	4	2	0.946
Ex39.xml	FactoryOverHead	-1	Not bad	52	4	1	0.946
Ex39.xml	PostCosting	-1	Not bad	52	4	1	0.946
Ex39.xml	SendingNotice	-1	Not bad	52	4	1	0.946
Ex39.xml	WorkInProcess	-1	Not bad	52	4	1	0.946
Ex39.xml	ProductionDepartment	-1	Not bad	52	4	1	0.946
Ex39.xml	DeliveryWIP	-1	Not bad	52	4	1	0.946
Ex39.xml	Factor	-1	Not bad	52	4	1	0.946
Ex39.xml	Department	-1	Not bad	52	4	1	0.946
Ex39.xml	PayrollBook	-1	Not bad	52	4	1	0.946
Ex39.xml	ReturnedMaterial	-1	Not bad	52	4	1	0.946
Ex39.xml	FactoryOverHead	-1	Not bad	52	4	1	0.946
Ex39.xml	Allocation	-1	Not bad	52	4	1	0.946
Ex39.xml	Material	-1	Not bad	52	4	1	0.946
Ex39.xml	ProductionLine	-1	Not bad	52	4	1	0.946
Ex39.xml	FactoryOverHeadWIP	-1	Not bad	52	4	1	0.946

ตารางที่ ข.1(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex39.xml	Rating	-1	Not bad	52	4	1	0.946
Ex39.xml	MaterialRequisitionDetail	-1	Not bad	52	4	1	0.946
Ex39.xml	UsingFactor	-1	Not bad	52	4	1	0.946
Ex39.xml	MaterialWIP	-1	Not bad	52	4	1	0.946
Ex39.xml	ReturnedMaterialDetail	-1	Not bad	52	4	1	0.946
Ex39.xml	Expension	-1	Not bad	52	4	1	0.946
Ex39.xml	ServiceDepartment	-1	Not bad	52	4	1	0.946
Ex39.xml	WIPTransaction	-1	Not bad	52	4	1	0.946
Ex39.xml	MaterialRequisition	-1	Not bad	52	4	1	0.946
Ex39.xml	SendingNoticeDetail	-1	Not bad	52	4	1	0.946
Ex40.xml	Inverse	-1	Not bad	56	0	4	1.402
Ex40.xml	Operation	3	Not bad	56	0	4	1.402
Ex40.xml	Divide	-1	Not bad	56	0	4	1.402
Ex40.xml	MemoryPlus	-1	Not bad	56	0	4	1.402
Ex40.xml	Minus	-1	Not bad	56	0	4	1.402
Ex40.xml	OperationPanel	-1	Not bad	56	0	4	1.402

ตารางที่ ข.1(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	State	-1	Not bad	56	0	4	1.402
Ex40.xml	Plus	-1	Not bad	56	0	4	1.402
Ex40.xml	EnteringNumberState	-1	Not bad	56	0	4	1.402
Ex40.xml	OperationHandler	-1	Not bad	56	0	4	1.402
Ex40.xml	OperationStack	-1	Not bad	56	0	4	1.402
Ex40.xml	FunctionPanel	-1	Not bad	56	0	4	1.402
Ex40.xml	Display	-1	Not bad	56	0	4	1.402
Ex40.xml	DigitHandler	-1	Not bad	56	0	4	1.402
Ex40.xml	MemoryRecall	-1	Not bad	56	0	4	1.402
Ex40.xml	Square	-1	Not bad	56	0	4	1.402
Ex40.xml	CalculatorFrame	-1	Not bad	56	0	4	1.402
Ex40.xml	ReverseSign	-1	Not bad	56	0	4	1.402
Ex40.xml	BinaryOperation	-1	Not bad	56	0	4	1.402
Ex40.xml	WaitingForOperationState	-1	Not bad	56	0	4	1.402
Ex40.xml	WaitingForNumberState	-1	Not bad	56	0	4	1.402
Ex40.xml	MemmoryStatus	-1	Not bad	56	0	4	1.402

ตารางที่ ข.1(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	Cpu	19	Not bad	56	0	4	1.402
Ex40.xml	KeyPanel	3	Not bad	56	0	4	1.402
Ex40.xml	KeyHandler	-1	Not bad	56	0	4	1.402
Ex40.xml	SquareRoot	-1	Not bad	56	0	4	1.402
Ex40.xml	Multiply	-1	Not bad	56	0	4	1.402
Ex40.xml	Memory	-1	Not bad	56	0	4	1.402
Ex40.xml	UnaryOperation	-1	Not bad	56	0	4	1.402
Ex40.xml	WaitingForInputState	-1	Not bad	56	0	4	1.402
Ex40.xml	Calculator	-1	Not bad	56	0	4	1.402
Ex40.xml	MemoryOperation	-1	Not bad	56	0	4	1.402
Ex40.xml	NumberPanel	-1	Not bad	56	0	4	1.402
Ex40.xml	MemoryMinus	-1	Not bad	56	0	4	1.402
Ex40.xml	CalculatorTape	-1	Not bad	56	0	4	1.402
Ex40.xml	Register	3	Not bad	56	0	4	1.402
Ex40.xml	OperandStack	-1	Not bad	56	0	4	1.402
Ex40.xml	DisplayPanel	2	Not bad	56	0	4	1.402
Ex40.xml	MemoryClear	-1	Not bad	56	0	4	1.402

ตารางที่ ข.1(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex8.xml	FoodList	1	Not bad	20	1	0	1.288
Ex8.xml	Waiter	-1	Not bad	20	1	0	1.288
Ex8.xml	Order	3	Not bad	20	1	0	1.288
Ex8.xml	Queue	-1	Not bad	20	1	0	1.288
Ex8.xml	Bill	2	Not bad	20	1	0	1.288
Ex8.xml	Table	1	Not bad	20	1	0	1.288
Ex8.xml	Cashier	-1	Not bad	20	1	0	1.288
Ex8.xml	Customer	4	Not bad	20	1	0	1.288
Ex8.xml	FoodInfo	0	Bad	20	1	0	1.288
Ex8.xml	Chief	-1	Not bad	20	1	0	1.288
Ex37.xml	Material	9	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturnLineItem	-1	Not bad	23	5	0	0.832
Ex37.xml	ReceiptItem	-1	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	16	Not bad	23	5	0	0.832
Ex37.xml	ReceiptLineItem	-1	Not bad	23	5	0	0.832
Ex37.xml	CreditVendor	0	Bad	23	5	0	0.832

ตารางที่ ข.1(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex8.xml	FoodList	1	Not bad	20	1	0	1.288
Ex8.xml	Waiter	-1	Not bad	20	1	0	1.288
Ex8.xml	Order	3	Not bad	20	1	0	1.288
Ex8.xml	Queue	-1	Not bad	20	1	0	1.288
Ex8.xml	Bill	2	Not bad	20	1	0	1.288
Ex8.xml	Table	1	Not bad	20	1	0	1.288
Ex8.xml	Cashier	-1	Not bad	20	1	0	1.288
Ex8.xml	Customer	4	Not bad	20	1	0	1.288
Ex8.xml	FoodInfo	0	Bad	20	1	0	1.288
Ex8.xml	Chief	-1	Not bad	20	1	0	1.288
Ex37.xml	Material	9	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturnLineItem	-1	Not bad	23	5	0	0.832
Ex37.xml	ReceiptItem	-1	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	16	Not bad	23	5	0	0.832
Ex37.xml	ReceiptLineItem	-1	Not bad	23	5	0	0.832
Ex37.xml	CreditVendor	0	Bad	23	5	0	0.832

ค่ามาตรวัดหลังทำรีแฟคทอริง

ตารางที่ ข.2 แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex8.xml	FoodList	1	Not bad	1.288	0.992	0.888	0.908
Ex8.xml	Waiter	-1	Not bad	1.288	0.992	0.888	0.908
Ex8.xml	Order	3	Not bad	1.288	0.992	0.888	0.908
Ex8.xml	Queue	-1	Not bad	1.288	0.992	0.888	0.908
Ex8.xml	Bill	2	Not bad	1.288	0.992	0.888	0.908
Ex8.xml	Table	1	Not bad	1.288	0.992	0.888	0.908
Ex8.xml	Cashier	-1	Not bad	1.288	0.992	0.888	0.908
Ex8.xml	Customer	4	Not bad	1.288	0.992	0.888	0.908
Ex8.xml	Chief	-1	Not bad	1.288	0.992	0.888	0.908
Ex37.xml	Material	9	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseReturnLineItem	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	ReceiptItem	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseOrder	15	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	ReceiptLineItem	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	CreditVendor	1	Not bad	23	4	0	0.946

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.2(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Data Class หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		Nnorm		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	POScheduler	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	Vendor	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POFacade	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PONewWindow	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POLineItem	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostRecWindow	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseRequisition	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseReturn	3	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostPRWindow	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PriceList	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchasingDocumentControl	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PRLineItem	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostPOWindow	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostReturnWindow	-1	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POScheduler	-1	Not bad	0.946	0.992	0.54	0.872

2.ผลการทดสอบการตรวจจับข้อบกพร่องประเภท Feature Envy

ค่ามาตรวัดก่อนทำรีแฟคทอริงแสดงดังตารางที่ ข.3 ค่ามาตรวัดที่เป็น 9999 หมายถึงค่ามาตรวัดเป็นค่าเข้าสู่ค่าอนันต์ (ค่ามาตรวัด NMsg มีค่าเป็น 0)

ตารางที่ ข.3 แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	Material	avgCost	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	saveAs	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	setAmount	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	setCostPerUnit	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	setQtyOnHand	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	setQtyOnPurchaseOrder	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	updateMaterial	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	getCostPerUnit	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	calculatePrice	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	calculateVat	9999	Not bad	23	5	0	0.832
Ex37.xml	Material	calculateDiscount	9999	Not bad	23	5	0	0.832
Ex37.xml	ReceiptItem	postReceiptItem	0.4	Not bad	23	5	0	0.832
Ex37.xml	ReceiptItem	setFlagPost	9999	Not bad	23	5	0	0.832
Ex37.xml	ReceiptItem	updateReceiptItem	9999	Not bad	23	5	0	0.832

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.3(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	ReceiptItem	verityStatus	9999	Not bad	23	5	0	0.832
Ex37.xml	ReceiptItem	displayPurchaseReturn	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	addLine	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	delete	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	generatePONumber	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	getListAll	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	postPO	0.666666667	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	save	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	saveas	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setDeliveryDueDate	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setFlagPost	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setPODate	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setPOLine	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setPONumber	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setStatus	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	updatePO	9999	Not bad	23	5	0	0.832

ตารางที่ ข.3(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	PurchaseOrder	verifyCreditOfVender	0.5	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	verifyStatus	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	viewDocumentFlow	9999	Not bad	23	5	0	0.832
Ex37.xml	ReceiptLineItem	howManyQtyOfReturn	9999	Not bad	23	5	0	0.832
Ex37.xml	ReceiptLineItem	verifyReturnQty	9999	Not bad	23	5	0	0.832
Ex37.xml	CreditVendor	getBalance	9999	Not bad	23	5	0	0.832
Ex37.xml	CreditVendor	getCreditLimit	9999	Not bad	23	5	0	0.832
Ex37.xml	POScheduler	verifyShippingDate	9999	Not bad	23	5	0	0.832
Ex37.xml	POScheduler	updateShippingPlace	1	Not bad	23	5	0	0.832
Ex37.xml	Vendor	showDocumentFlow	0.5	Not bad	23	5	0	0.832
Ex37.xml	POFacade	getAllPurchasingDocument	9999	Not bad	23	5	0	0.832
Ex37.xml	POLineItem	howManyQtyOfReceipt	9999	Not bad	23	5	0	0.832
Ex37.xml	POLineItem	verifyReceiptQty	9999	Not bad	23	5	0	0.832
Ex37.xml	POLineItem	verifyReturnQty	9999	Not bad	23	5	0	0.832
Ex37.xml	POLineItem	calculateAmount	0.25	Bad	23	5	0	0.832
Ex37.xml	PurchaseRequisition	postPR	9999	Not bad	23	5	0	0.832

ตารางที่ ข.3(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	PurchaseRequisition	setFlagPost	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseRequisition	updatePR	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	postPurchaseReturn	0.666666667	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	setFlagPost	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	updatePurchaseReturn	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	getComment	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	getTotalAmount	9999	Not bad	23	5	0	0.832
Ex37.xml	PriceList	getListAll	9999	Not bad	23	5	0	0.832
Ex37.xml	PriceList	getProductOfVender	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchasingDocumentControl	save	9999	Not bad	23	5	0	0.832
Ex37.xml	PurchasingDocumentControl	saveAs	9999	Not bad	23	5	0	0.832
Ex37.xml	ShippingPlace	setShippingPlace	9999	Not bad	23	5	0	0.832
Ex38.xml	VCR	getTimesRented	9999	Not bad	57	4	2	0.946
Ex38.xml	Member	acquireMembership	9999	Not bad	57	4	2	0.946
Ex38.xml	Member	cancelMembership	9999	Not bad	57	4	2	0.946
Ex38.xml	Member	create	9999	Not bad	57	4	2	0.946

ตารางที่ ข.3(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Member	determineIfDelinquent	9999	Not bad	57	4	2	0.946
Ex38.xml	Member	setMemberDetail	9999	Not bad	57	4	2	0.946
Ex38.xml	Member	updateCreditCardInformation	9999	Not bad	57	4	2	0.946
Ex38.xml	Member	updateOverdueAmount	9999	Not bad	57	4	2	0.946
Ex38.xml	Member	verifyMembership	9999	Not bad	57	4	2	0.946
Ex38.xml	Member	getTransactions	9999	Not bad	57	4	2	0.946
Ex38.xml	RentalTransaction	checkingInARentalItem	9999	Not bad	57	4	2	0.946
Ex38.xml	RentalTransaction	rentAnItem	9999	Not bad	57	4	2	0.946
Ex38.xml	RentalTransaction	setDueDate	9999	Not bad	57	4	2	0.946
Ex38.xml	RentalTransaction	getLineItem	9999	Not bad	57	4	2	0.946
Ex38.xml	POLineItem	computeItemTotalCost	9999	Not bad	57	4	2	0.946
Ex38.xml	Vendor	addNewVendorInformation	9999	Not bad	57	4	2	0.946
Ex38.xml	Vendor	changeExistingVendorInformation	9999	Not bad	57	4	2	0.946
Ex38.xml	Vendor	provideVendorInformation	9999	Not bad	57	4	2	0.946
Ex38.xml	Video	checkItem	9999	Not bad	57	4	2	0.946
Ex38.xml	PurchaseOrder	computePurchaseOrderTotalCost	9999	Not bad	57	4	2	0.946

ตารางที่ ข.3(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	SaleRentalLineItem	barCodeNumber	9999	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	price	9999	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	setSaleAmount	9999	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	setSalesTax	9999	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	transactionNumber	9999	Not bad	57	4	2	0.946
Ex38.xml	StoreLocation	provideStoreInformation	9999	Not bad	57	4	2	0.946
Ex38.xml	RentalItem	updateRentalInformation	9999	Not bad	57	4	2	0.946
Ex38.xml	SalesTransaction	purchaseItems	9999	Not bad	57	4	2	0.946
Ex38.xml	SalesTransaction	setSaleAmount	1	Not bad	57	4	2	0.946
Ex38.xml	Transaction	payforTransaction	9999	Not bad	57	4	2	0.946
Ex38.xml	TransactionCollection	getTransaction	9999	Not bad	57	4	2	0.946
Ex38.xml	TransactionCollection	getRentalTransaction	9999	Not bad	57	4	2	0.946
Ex38.xml	Inventory	addNewInventoryItem	9999	Not bad	57	4	2	0.946
Ex38.xml	Inventory	changeInventoryItemInformation	9999	Not bad	57	4	2	0.946
Ex38.xml	Inventory	deleteInventoryItem	9999	Not bad	57	4	2	0.946
Ex38.xml	Inventory	inquireAboutAvailableInventory	9999	Not bad	57	4	2	0.946

ตารางที่ ข.3(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Inventory	orderInventory	9999	Not bad	57	4	2	0.946
Ex38.xml	Inventory	updateQuantityOnOrder	9999	Not bad	57	4	2	0.946
Ex38.xml	Employee	buyVDO	9999	Not bad	57	4	2	0.946
Ex38.xml	Employee	getAmount	1	Not bad	57	4	2	0.946
Ex38.xml	Employee	getMemberDetail	1	Not bad	57	4	2	0.946
Ex38.xml	Employee	requestMember	9999	Not bad	57	4	2	0.946
Ex38.xml	Employee	requestVDO	1	Not bad	57	4	2	0.946
Ex38.xml	Employee	updateEmployeeInformation	9999	Not bad	57	4	2	0.946
Ex38.xml	Employee	viewRentalReport	0.25	Bad	57	4	2	0.946
Ex38.xml	SaleItem	updateInventoryQtyOnHand	9999	Not bad	57	4	2	0.946
Ex38.xml	SaleItem	updateQuantitySold	9999	Not bad	57	4	2	0.946
Ex39.xml	PostCosting	save	9999	Not bad	52	4	1	0.946
Ex39.xml	PostCosting	saveAs	9999	Not bad	52	4	1	0.946
Ex39.xml	SendingNotice	postSendingNotice	9999	Not bad	52	4	1	0.946
Ex39.xml	WorkInProcess	calcCost	9999	Not bad	52	4	1	0.946
Ex39.xml	WorkInProcess	calcDlEqu	9999	Not bad	52	4	1	0.946

ตารางที่ ข.3(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex39.xml	Department	saveAs	9999	Not bad	52	4	1	0.946
Ex39.xml	PayrollBook	postAmountOfDepartment	1	Not bad	52	4	1	0.946
Ex39.xml	PayrollBook	setFlagPost	9999	Not bad	52	4	1	0.946
Ex39.xml	PayrollBook	updatePayrollBook	9999	Not bad	52	4	1	0.946
Ex39.xml	ReturnedMaterial	postReturnMaterial	0.25	Bad	52	4	1	0.946
Ex39.xml	ReturnedMaterial	setFlagPost	9999	Not bad	52	4	1	0.946
Ex39.xml	ReturnedMaterial	updateReturnedMaterial	9999	Not bad	52	4	1	0.946
Ex39.xml	Allocation	allocateCommonFoh	9999	Not bad	52	4	1	0.946
Ex39.xml	Allocation	allocateExpension	9999	Not bad	52	4	1	0.946
Ex39.xml	Allocation	allocateServiceFoh	9999	Not bad	52	4	1	0.946
Ex39.xml	Allocation	getListAll	9999	Not bad	52	4	1	0.946
Ex39.xml	Allocation	save	9999	Not bad	52	4	1	0.946
Ex39.xml	Allocation	saveAs	9999	Not bad	52	4	1	0.946
Ex39.xml	Material	avgCost	9999	Not bad	52	4	1	0.946
Ex39.xml	Material	saveAs	1	Not bad	52	4	1	0.946
Ex39.xml	Material	setAmount	9999	Not bad	52	4	1	0.946

ค่ามาตรวัดหลังทำรีแฟคทอริง

ตารางที่ ข.4 แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex36.xml	ExportReport	receive	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	ExportReport	sendTo	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	ExportInventory	check	1	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	ExportInventory	sendTo	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	ExportInventory	update	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	Report	generate	1	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	ReportFromPort	getReportFromPort	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	ReportFromPort	received	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	ReportFromPort	sendTo	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	Address	getAddress	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	Address	updateAddr	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	ReportFromCustomer	getReportFromCustomer	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	ReportToCustomer	sendTo	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	Invoice	compare	9999	Not bad	1.288	0.912	0.888	0.872

ตารางที่ ข.4(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex36.xml	Invoice	genInvoice	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	Invoice	getCustInfo	1	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	Customer	findAddress	1	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	Customer	getCustomer	9999	Not bad	1.288	0.912	0.888	0.872
Ex36.xml	Customer	newCustomer	9999	Not bad	1.288	0.912	0.888	0.872
Ex37.xml	Material	avgCost	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	saveAs	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	setAmount	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	setCostPerUnit	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	setQtyOnHand	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	setQtyOnPurchaseOrder	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	updateMaterial	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	getCostPerUnit	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	calculatePrice	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	calculateVat	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	calculateDiscount	9999	Not bad	0.832	0.992	0.424	0.872

ตารางที่ ข.4(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	Material	calculateAmount	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	postReceiptItem	0.4	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	setFlagPost	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	updateReceiptItem	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	verityStatus	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	displayPurchaseReturn	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	addLine	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	delete	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	generatePONumber	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	getListAll	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	postPO	0.5	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	save	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	saveas	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setDeliveryDueDate	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setFlagPost	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setPODate	9999	Not bad	0.832	0.992	0.424	0.872

ตารางที่ ข.4(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	PurchaseOrder	setPOLine	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setPONumber	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setStatus	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	updatePO	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	verifyCreditOfVender	0.5	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	verifyStatus	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	viewDocumentFlow	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptLineItem	howManyQtyOfReturn	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptLineItem	verifyReturnQty	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	CreditVendor	getBalance	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	CreditVendor	getCreditLimit	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POScheduler	verifyShippingDate	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POScheduler	updateShippingPlace	1	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Vendor	showDocumentFlow	0.5	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POFacade	getAllPurchasingDocument	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POLineItem	howManyQtyOfReceipt	9999	Not bad	0.832	0.992	0.424	0.872

ตารางที่ ข.4(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	POLineItem	verifyReceiptQty	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POLineItem	verifyReturnQty	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseRequisition	postPR	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseRequisition	setFlagPost	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseRequisition	updatePR	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	postPurchaseReturn	0.666666667	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	setFlagPost	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	updatePurchaseReturn	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	getComment	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	getTotalAmount	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PriceList	getListAll	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PriceList	getProductOfVender	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchasingDocumentControl	save	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchasingDocumentControl	saveAs	9999	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ShippingPlace	setShippingPlace	9999	Not bad	0.832	0.992	0.424	0.872
Ex38.xml	VCR	getTimesRented	9999	Not bad	0.946	0.912	0.54	0.566

ตารางที่ ข.4(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Member	acquireMembership	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	cancelMembership	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	create	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	determineIfDelinquent	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	setMemberDetail	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	updateCreditCardInformation	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	updateOverdueAmount	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	verifyMembership	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	getTransactions	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	checkingInARentalItem	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	rentAnItem	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	setDueDate	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	getLineItem	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	POLineItem	computeItemTotalCost	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Vendor	addNewVendorInformation	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Vendor	changeExistingVendorInformation	9999	Not bad	0.946	0.912	0.54	0.566

ตารางที่ ข.4(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Vendor	provideVendorInformation	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Video	checkItem	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	PurchaseOrder	computePurchaseOrderTotalCost	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	barCodeNumber	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	price	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	setSaleAmount	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	setSalesTax	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	transactionNumber	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	processLineItem	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	StoreLocation	provideStoreInformation	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalItem	updateRentalInformation	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SalesTransaction	purchaseItems	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SalesTransaction	setSaleAmount	1	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Transaction	payforTransaction	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	TransactionCollection	getTransaction	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	TransactionCollection	getRentalTransaction	9999	Not bad	0.946	0.912	0.54	0.566

ตารางที่ ข.4(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Inventory	addNewInventoryItem	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	changeInventoryItemInformation	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	deleteInventoryItem	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	inquireAboutAvailableInventory	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	orderInventory	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	updateQuantityOnOrder	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Employee	buyVDO	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Employee	getAmount	1	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Employee	getMemberDetail	1	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Employee	requestMember	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Employee	requestVDO	1	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Employee	updateEmployeeInformation	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Employee	viewRentalReport	1	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleItem	updateInventoryQtyOnHand	9999	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleItem	updateQuantitySold	9999	Not bad	0.946	0.912	0.54	0.566
Ex39.xml	PostCosting	save	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Department	getListAll	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Department	save	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Department	saveAs	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	PayrollBook	postAmountOfDepartment	1	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	PayrollBook	setFlagPost	9999	Not bad	0.946	0.952	0.54	0.611

ตารางที่ ข.4(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex39.xml	ReturnedMaterial	updateReturnedMaterial	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Allocation	allocateCommonFoh	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Allocation	allocateExpension	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Allocation	allocateServiceFoh	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Allocation	getListAll	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Allocation	save	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Allocation	saveAs	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Material	avgCost	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Material	saveAs	1	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Material	setAmount	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Material	setCostPerUnit	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Material	setQtyOnHand	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Material	updateMaterial	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Material	updateMaterialFromReturn	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	FactoryOverHeadWIP	calcFOHCost	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	FactoryOverHeadWIP	calcFOHPerUnit	9999	Not bad	0.946	0.952	0.54	0.611

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.4(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Feature Envy หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			NSMsg/ NMsg		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex39.xml	FactoryOverHeadWIP	postFOHWIP	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Rating	calcDlRate	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Rating	calcFohRate	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	MaterialWIP	calcRmCost	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	MaterialWIP	calcRmCostPerUnit	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Expension	save	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	Expension	saveAs	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	WIPTransaction	postWIPToWIP	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	MaterialRequisition	opname	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	MaterialRequisition	postMaterialRequisition	0.333333333	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	MaterialRequisition	setFlagPost(flag : string)	9999	Not bad	0.946	0.952	0.54	0.611
Ex39.xml	MaterialRequisition	updateMaterialRequisition	9999	Not bad	0.946	0.952	0.54	0.611

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3.ผลการทดสอบการตรวจจับข้อบกพร่องประเภท Message Chains

ค่ามาตรวัดก่อนทำรีแฟคทอริง

ตารางที่ ข.5 แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	PurchaseOrder	postPO	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	save	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	saveas	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setDeliveryDueDate	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setFlagPost	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setPODate	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setPOLine	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setPONumber	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	setStatus	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	updatePO	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	verifyCreditOfVender	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	verifyStatus	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseOrder	viewDocumentFlow	2	Bad	23	5	0	0.832

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.5(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	ReceiptLineItem	howManyQtyOfReturn	0	Not bad	23	5	0	0.832
Ex37.xml	ReceiptLineItem	verifyReturnQty	0	Not bad	23	5	0	0.832
Ex37.xml	CreditVendor	getBalance	0	Not bad	23	5	0	0.832
Ex37.xml	CreditVendor	getCreditLimit	0	Not bad	23	5	0	0.832
Ex37.xml	POScheduler	verifyShippingDate	0	Not bad	23	5	0	0.832
Ex37.xml	POScheduler	updateShippingPlace	0	Not bad	23	5	0	0.832
Ex37.xml	Vendor	showDocumentFlow	0	Not bad	23	5	0	0.832
Ex37.xml	POFacade	getAllPurchasingDocument	0	Not bad	23	5	0	0.832
Ex37.xml	POLineItem	howManyQtyOfReceipt	0	Not bad	23	5	0	0.832
Ex37.xml	POLineItem	verifyReceiptQty	0	Not bad	23	5	0	0.832
Ex37.xml	POLineItem	verifyReturnQty	0	Not bad	23	5	0	0.832
Ex37.xml	POLineItem	calculateAmount	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseRequisition	postPR	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseRequisition	setFlagPost	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseRequisition	updatePR	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	postPurchaseReturn	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	setFlagPost	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	updatePurchaseReturn	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	getComment	0	Not bad	23	5	0	0.832
Ex37.xml	PurchaseReturn	getTotalAmount	0	Not bad	23	5	0	0.832
Ex37.xml	PriceList	getListAll	0	Not bad	23	5	0	0.832
Ex37.xml	PriceList	getProductOfVender	0	Not bad	23	5	0	0.832

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.5(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	PurchasingDocumentControl	save	0	Not bad	23	5	0	0.832
Ex37.xml	PurchasingDocumentControl	saveAs	0	Not bad	23	5	0	0.832
Ex37.xml	ShippingPlace	setShippingPlace	0	Not bad	23	5	0	0.832
Ex38.xml	VCR	getTimesRented	0	Not bad	57	4	2	0.946
Ex38.xml	Member	acquireMembership	0	Not bad	57	4	2	0.946
Ex38.xml	Member	cancelMembership	0	Not bad	57	4	2	0.946
Ex38.xml	Member	create	0	Not bad	57	4	2	0.946
Ex38.xml	Member	determineIfDelinquent	0	Not bad	57	4	2	0.946
Ex38.xml	Member	setMemberDetail	0	Not bad	57	4	2	0.946
Ex38.xml	Member	updateCreditCardInformation	0	Not bad	57	4	2	0.946
Ex38.xml	Member	updateOverdueAmount	0	Not bad	57	4	2	0.946
Ex38.xml	Member	verifyMembership	0	Not bad	57	4	2	0.946
Ex38.xml	Member	getTransactions	3	Bad	57	4	2	0.946
Ex38.xml	RentalTransaction	checkingInARentalItem	0	Not bad	57	4	2	0.946
Ex38.xml	RentalTransaction	rentAnItem	0	Not bad	57	4	2	0.946
Ex38.xml	RentalTransaction	setDueDate	0	Not bad	57	4	2	0.946
Ex38.xml	RentalTransaction	getLineItem	0	Not bad	57	4	2	0.946
Ex38.xml	POLineItem	computeItemTotalCost	0	Not bad	57	4	2	0.946
Ex38.xml	Vendor	addNewVendorInformation	0	Not bad	57	4	2	0.946
Ex38.xml	Vendor	changeExistingVendorInformation	0	Not bad	57	4	2	0.946
Ex38.xml	Vendor	provideVendorInformation	0	Not bad	57	4	2	0.946

ตารางที่ ข.5(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Video	checkItem	0	Not bad	57	4	2	0.946
Ex38.xml	PurchaseOrder	computePurchaseOrderTotalCost	0	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	barCodeNumber	0	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	price	0	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	setSaleAmount	0	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	setSalesTax	0	Not bad	57	4	2	0.946
Ex38.xml	SaleRentalLineItem	transactionNumber	0	Not bad	57	4	2	0.946
Ex38.xml	StoreLocation	provideStoreInformation	0	Not bad	57	4	2	0.946
Ex38.xml	RentalItem	updateRentalInformation	0	Not bad	57	4	2	0.946
Ex38.xml	SalesTransaction	purchaseItems	0	Not bad	57	4	2	0.946
Ex38.xml	SalesTransaction	setSaleAmount	0	Not bad	57	4	2	0.946
Ex38.xml	Transaction	payforTransaction	0	Not bad	57	4	2	0.946
Ex38.xml	TransactionCollection	getTransaction	0	Not bad	57	4	2	0.946
Ex38.xml	TransactionCollection	getRentalTransaction	0	Not bad	57	4	2	0.946
Ex38.xml	Inventory	addNewInventoryItem	0	Not bad	57	4	2	0.946
Ex38.xml	Inventory	changeInventoryItemInformation	0	Not bad	57	4	2	0.946
Ex38.xml	Inventory	deleteInventoryItem	0	Not bad	57	4	2	0.946
Ex38.xml	Inventory	inquireAboutAvailableInventory	0	Not bad	57	4	2	0.946
Ex38.xml	Inventory	orderInventory	0	Not bad	57	4	2	0.946
Ex38.xml	Inventory	updateQuantityOnOrder	0	Not bad	57	4	2	0.946
Ex38.xml	Employee	buyVDO	0	Not bad	57	4	2	0.946
Ex38.xml	Employee	getAmount	0	Not bad	57	4	2	0.946

ตารางที่ ข.5(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Employee	getMemberDetail	0	Not bad	57	4	2	0.946
Ex38.xml	Employee	requestMember	0	Not bad	57	4	2	0.946
Ex38.xml	Employee	requestVDO	0	Not bad	57	4	2	0.946
Ex38.xml	Employee	updateEmployeeInformation	0	Not bad	57	4	2	0.946
Ex38.xml	Employee	viewRentalReport	0	Not bad	57	4	2	0.946
Ex38.xml	SaleItem	updateInventoryQtyOnHand	0	Not bad	57	4	2	0.946
Ex38.xml	SaleItem	updateQuantitySold	0	Not bad	57	4	2	0.946

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ค่ามาตรวัดหลังทำรีแฟคทอริง

ตารางที่ ข.6 แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	Material	avgCost	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	saveAs	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	setAmount	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	setCostPerUnit	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	setQtyOnHand	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	setQtyOnPurchaseOrder	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	updateMaterial	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	getCostPerUnit	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	calculatePrice	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	calculateVat	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Material	calculateDiscount	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	postReceiptItem	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	setFlagPost	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	updateReceiptItem	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	verityStatus	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	displayPurchaseReturn	0	Not bad	0.832	0.992	0.424	0.872

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.6(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	ReceiptItem	getPurchaseReturnComment	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	getPurchaseReturnTotalAmount	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	addLine	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	delete	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	generatePONumber	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	getListAll	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	postPO	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	save	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	saveas	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setDeliveryDueDate	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setFlagPost	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setPODate	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setPOLine	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setPONumber	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setShippingPlace	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	setStatus	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	updatePO	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	verifyCreditOfVender	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	verifyStatus	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	viewDocumentFlow	1	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptLineItem	howManyQtyOfReturn	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptLineItem	verifyReturnQty	0	Not bad	0.832	0.992	0.424	0.872

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.6(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	CreditVendor	getBalance	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	CreditVendor	getCreditLimit	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Vendor	showDocumentFlow	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POFacade	getAllPurchasingDocument	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POLineItem	howManyQtyOfReceipt	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POLineItem	verifyReceiptQty	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POLineItem	verifyReturnQty	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POLineItem	calculateAmount	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseRequisition	postPR	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseRequisition	setFlagPost	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseRequisition	updatePR	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	postPurchaseReturn	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	setFlagPost	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	updatePurchaseReturn	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	getComment	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	getTotalAmount	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PriceList	getListAll	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PriceList	getProductOfVender	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchasingDocumentControl	save	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchasingDocumentControl	saveAs	0	Not bad	0.832	0.992	0.424	0.872
Ex38.xml	VCR	getTimesRented	0	Not bad	0.946	0.912	0.54	0.566

ตารางที่ ข.6(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Member	acquireMembership	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	cancelMembership	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	create	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	determineIfDelinquent	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	setMemberDetail	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	updateCreditCardInformation	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	updateOverdueAmount	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	verifyMembership	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	getTransactions	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Member	displayRentalReport	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	checkingInARentalItem	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	rentAnItem	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	setDueDate	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	getLineItem	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	itemTransactionNumber	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	setItemSalesAmount	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	itemBarCodeNumber	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalTransaction	setItemSalesTax	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	POLineItem	computeItemTotalCost	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Vendor	addNewVendorInformation	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Vendor	changeExistingVendorInformation	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Vendor	provideVendorInformation	0	Not bad	0.946	0.912	0.54	0.566

ตารางที่ ข.6(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Video	checkItem	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	PurchaseOrder	computePurchaseOrderTotalCost	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	barCodeNumber	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	price	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	setSaleAmount	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	setSalesTax	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SaleRentalLineItem	transactionNumber	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	StoreLocation	provideStoreInformation	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	RentalItem	updateRentalInformation	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SalesTransaction	purchaseItems	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	SalesTransaction	setSaleAmount	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Transaction	payforTransaction	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	TransactionCollection	getTransaction	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	TransactionCollection	getRentalTransaction	1	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	addNewInventoryItem	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	changeInventoryItemInformation	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	deleteInventoryItem	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	inquireAboutAvailableInventory	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	orderInventory	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Inventory	updateQuantityOnOrder	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Employee	buyVDO	0	Not bad	0.946	0.912	0.54	0.566
Ex38.xml	Employee	getAmount	0	Not bad	0.946	0.912	0.54	0.566

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.6(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Message Chains หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ชื่อเมทธอด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
			SOMsgCh		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex38.xml	Employee	getMemberDetail	0	Not bad	0.946	0.912	0.54	Ex38.xml
Ex38.xml	Employee	requestMember	0	Not bad	0.946	0.912	0.54	Ex38.xml
Ex38.xml	Employee	requestVDO	0	Not bad	0.946	0.912	0.54	Ex38.xml
Ex38.xml	Employee	updateEmployeeInformation	0	Not bad	0.946	0.912	0.54	Ex38.xml
Ex38.xml	Employee	viewRentalReport	0	Not bad	0.946	0.912	0.54	Ex38.xml
Ex38.xml	SaleItem	updateInventoryQtyOnHand	0	Not bad	0.946	0.912	0.54	Ex38.xml
Ex38.xml	SaleItem	updateQuantitySold	0	Not bad	0.946	0.912	0.54	Ex38.xml

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.ผลการทดสอบการตรวจจับข้อบกพร่องประเภท Middle Man

ค่ามาตรวัดก่อนทำรีแฟคทอริง

ตารางที่ ข.7 แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		WOD		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex33.xml	SearchForm	0	Not bad	0.946	0.992	0.54	0.818
Ex33.xml	ProductItem	0	Not bad	0.946	0.992	0.54	0.818
Ex33.xml	LineItem	1	Bad	0.946	0.992	0.54	0.818
Ex33.xml	SearchResults	0	Not bad	0.946	0.992	0.54	0.818
Ex33.xml	HomePage	0.333333333	Not bad	0.946	0.992	0.54	0.818
Ex33.xml	Cart	0	Not bad	0.946	0.992	0.54	0.818
Ex33.xml	Product	0.5	Bad	0.946	0.992	0.54	0.818
Ex33.xml	Catalog	0.166666667	Not bad	0.946	0.992	0.54	0.818
Ex33.xml	LineItemsForm	0	Not bad	0.946	0.992	0.54	0.818
Ex33.xml	CategoryItem	0	Not bad	0.946	0.992	0.54	0.818
Ex33.xml	Category	0.142857143	Not bad	0.946	0.992	0.54	0.818
Ex37.xml	Material	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturnLineItem	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	0	Not bad	0.832	0.992	0.424	0.872

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.7(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		WOD		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	ReceiptLineItem	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	CreditVendor	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POScheduler	0.5	Bad	0.832	0.992	0.424	0.872
Ex37.xml	Vendor	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POFacade	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PONewWindow	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POLineItem	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PostRecWindow	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseRequisition	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturn	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PostPRWindow	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PriceList	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchasingDocumentControl	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PRLineItem	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ShippingPlace	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PostPOWindow	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PostReturnWindow	0	Not bad	0.832	0.992	0.424	0.872
Ex40.xml	Inverse	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Operation	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Divide	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryPlus	0	Not bad	1.402	0.832	1.004	0.575

ตารางที่ ข.7(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		WOD		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	Minus	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperationPanel	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	State	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Plus	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	EnteringNumberState	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperationHandler	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperationStack	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	FunctionPanel	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Display	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	DigitHandler	1	Bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryRecall	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Square	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	CalculatorFrame	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	ReverseSign	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	BinaryOperation	0.5	Bad	1.402	0.832	1.004	0.575
Ex40.xml	WaitingForOperationState	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	WaitingForNumberState	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemmoryStatus	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Cpu	0.045454545	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	KeyPanel	0.25	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	KeyHandler	0	Not bad	1.402	0.832	1.004	0.575

ตารางที่ ข.7(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		WOD		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	SquareRoot	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Multiply	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Memory	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	UnaryOperation	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	WaitingForInputState	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Calculator	0.5	Bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryOperation	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	NumberPanel	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryMinus	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	CalculatorTape	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Register	0.166666667	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperandStack	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	DisplayPanel	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryClear	0	Not bad	1.402	0.832	1.004	0.575

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ค่ามาตรวัดหลังทำรีแฟคทอริง

ตารางที่ ข.8 แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		WOD		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex33.xml	SearchForm	0	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	ProductItem	0	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	SearchResults	0	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	HomePage	0.333333333	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	Cart	0	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	Product	0	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	Catalog	0.166666667	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	LineItemsForm	0	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	CategoryItem	0	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	Category	0.142857143	Not bad	1.06	0.992	0.656	0.836
Ex37.xml	Material	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseReturnLineItem	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	ReceiptItem	0	Not bad	0.946	0.992	0.54	0.872
Ex33.xml	SearchForm	0	Not bad	1.06	0.992	0.656	0.836
Ex33.xml	ProductItem	0	Not bad	1.06	0.992	0.656	0.836
Ex37.xml	PurchaseOrder	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	ReceiptLineItem	0	Not bad	0.946	0.992	0.54	0.872

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.8(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		WOD		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	CreditVendor	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POScheduler	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	Vendor	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POFacade	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PONewWindow	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POLineItem	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostRecWindow	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseRequisition	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseReturn	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostPRWindow	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PriceList	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchasingDocumentControl	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PRLineItem	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostPOWindow	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostReturnWindow	0	Not bad	0.946	0.992	0.54	0.872
Ex40.xml	Inverse	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Operation	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Divide	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryPlus	0	Not bad	1.402	0.832	1.004	0.575

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.8(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		WOD		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	Minus	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperationPanel	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	State	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Plus	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	EnteringNumberState	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperationHandler	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperationStack	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	FunctionPanel	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Display	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	DigitHandler	1	Bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryRecall	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Square	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	CalculatorFrame	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	ReverseSign	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	BinaryOperation	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	WaitingForOperationState	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	WaitingForNumberState	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemmoryStatus	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Cpu	0.045454545	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	KeyPanel	0.25	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	KeyHandler	0	Not bad	1.402	0.832	1.004	0.575

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.8(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี Middle Man หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		WOD		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	SquareRoot	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Multiply	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Memory	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	UnaryOperation	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	WaitingForInputState	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Calculator	0.5	Bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryOperation	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	NumberPanel	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryMinus	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	CalculatorTape	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Register	0.166666667	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperandStack	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	DisplayPanel	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryClear	0	Not bad	1.402	0.832	1.004	0.575

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5.ผลการทดสอบการตรวจจับข้อบกพร่องประเภท God Class

ค่ามาตรวัดก่อนทำรีแฟคทอริง

ตารางที่ ข.9 แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	ค่ามาตรวัด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		NCPA	RFC	COH		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	Material	0	11	0.029578077	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseReturnLineItem	0	0	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptItem	1	10	0.04	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseOrder	3	25	0.01384083	Bad	0.832	0.992	0.424	0.872
Ex37.xml	ReceiptLineItem	0	2	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	CreditVendor	0	2	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POScheduler	1	3	0.25	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	Vendor	2	5	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POFacade	0	1	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PONewWindow	0	1	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	POLineItem	1	8	0.0625	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PostRecWindow	0	1	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchaseRequisition	0	3	0.111111111	Not bad	0.832	0.992	0.424	0.872

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.9(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	ค่ามาตรวัด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		NCPA	RFC	COH		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	PurchaseReturn	0	8	0.04	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PostPRWindow	0	1	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PriceList	0	2	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PurchasingDocumentControl	0	2	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PRLineItem	0	0	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	ShippingPlace	0	1	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PostPOWindow	0	1	0	Not bad	0.832	0.992	0.424	0.872
Ex37.xml	PostReturnWindow	0	1	0	Not bad	0.832	0.992	0.424	0.872
Ex40.xml	Inverse	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Operation	0	5	0.04	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Divide	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryPlus	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Minus	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperationPanel	0	1	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	State	0	4	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Plus	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	EnteringNumberState	0	6	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperationHandler	0	1	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperationStack	0	5	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	FunctionPanel	0	1	0	Not bad	1.402	0.832	1.004	0.575

ตารางที่ ข.9(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	ค่ามาตรวัด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		NCPA	RFC	COH		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	Display	2	13	0.183172138	Bad	1.402	0.832	1.004	0.575
Ex40.xml	DigitHandler	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryRecall	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Square	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	CalculatorFrame	0	3	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	ReverseSign	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	BinaryOperation	1	8	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	WaitingForOperationState	0	4	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	WaitingForNumberState	0	4	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemmoryStatus	0	1	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Cpu	1	35	0.051652893	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	KeyPanel	0	5	0.0625	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	KeyHandler	0	1	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	SquareRoot	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Multiply	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Memory	0	4	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	UnaryOperation	0	3	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	WaitingForInputState	0	6	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Calculator	0	6	0.25	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryOperation	0	3	0	Not bad	1.402	0.832	1.004	0.575

ตารางที่ ข.9(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class ก่อนทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	ค่ามาตรวัด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		NCPA	RFC	COH		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	NumberPanel	0	1	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryMinus	0	2	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	CalculatorTape	0	2	1	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	Register	0	9	0.110307081	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	OperandStack	0	6	0	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	DisplayPanel	0	3	0.111111111	Not bad	1.402	0.832	1.004	0.575
Ex40.xml	MemoryClear	0	2	0	Not bad	1.402	0.832	1.004	0.575

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ค่ามาตรวัดหลังทำรีแฟคทอริง

ตารางที่ ข.10 แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	ค่ามาตรวัด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		NCPA	RFC	COH		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	Material	0	12	0.025	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseReturnLineItem	0	0	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POScheduler	0	2	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	ReceiptItem	1	10	0.04	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseOrder	0	17	0.008264463	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	ReceiptLineItem	0	2	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	ShippingPlace	0	0	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	CreditVendor	0	3	0.111111111	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	Vendor	2	5	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POFacade	0	1	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PONewWindow	0	1	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POStatusController	0	4	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POLineItem	1	8	0.0625	Not bad	0.946	0.992	0.54	0.872

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ ข.10(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	ค่ามาตรวัด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		NCPA	RFC	COH		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex37.xml	PostRecWindow	0	1	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseRequisition	0	3	0.111111111	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	POReportGenerator	0	2	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchaseReturn	0	8	0.04	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostPRWindow	0	1	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PriceList	0	2	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PurchasingDocumentControl	0	2	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PRLineItem	0	0	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostPOWindow	0	1	0	Not bad	0.946	0.992	0.54	0.872
Ex37.xml	PostReturnWindow	0	1	0	Not bad	0.946	0.992	0.54	0.872
Ex40.xml	Inverse	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	Operation	0	5	0.04	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	Divide	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	MemoryPlus	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	Minus	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	OperationPanel	0	1	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	State	0	4	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	Plus	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	EnteringNumberState	0	6	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	OperationHandler	0	1	0	Not bad	1.402	0.872	1.004	0.557

ตารางที่ ข.10(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	ค่ามาตรวัด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		NCPA	RFC	COH		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	OperationStack	0	5	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	FunctionPanel	0	1	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	Display	0	14	0.089854347	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	DigitHandler	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	MemoryRecall	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	Square	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	CalculatorFrame	0	3	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	ReverseSign	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	BinaryOperation	1	8	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	WaitingForOperationState	0	4	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	WaitingForNumberState	0	4	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	MemmoryStatus	0	1	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	Cpu	1	35	0.051652893	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	KeyPanel	0	5	0.0625	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	KeyHandler	0	1	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	SquareRoot	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	Multiply	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	Memory	0	7	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	UnaryOperation	0	3	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	WaitingForInputState	0	6	0	Not bad	1.402	0.872	1.004	0.557

ตารางที่ ข.10(ต่อ) แสดงผลการคำนวณมาตรวัดของระบบที่มี God Class หลังทำรีแฟคทอริง

โมเดล	ชื่อคลาส	ค่ามาตรวัด	ค่ามาตรวัด	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
		NCPA	RFC	COH		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex40.xml	Calculator	0	6	0.25	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	MemoryOperation	0	3	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	NumberPanel	0	1	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	MemoryMinus	0	2	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	CalculatorTape	0	2	1	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	OperandStack	0	6	0	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	DisplayPanel	0	3	0.111111111	Not bad	1.402	0.872	1.004	0.557
Ex40.xml	MemoryClear	0	2	0	Not bad	1.402	0.872	1.004	0.557

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6.ผลการทดสอบการตรวจจับข้อบกพร่องประเภท Switch Statements

ค่ามาตรวัดก่อนทำรีแฟคทอริง

ตารางที่ ข.11 แสดงผลการคำนวณมาตรวัดของระบบที่มี Switch Statements ก่อนทำรีแฟคทอริง

โมเดล	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
	NOSS		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex41.xml	1	Bad	1.402	0.992	1.004	0.836
Ex45.xml	1	Bad	1.402	0.912	1.004	0.827

ค่ามาตรวัดหลังทำรีแฟคทอริง

ตารางที่ ข.12 แสดงผลการคำนวณมาตรวัดของระบบที่มี Switch Statements หลังทำรีแฟคทอริง

โมเดล	ค่ามาตรวัด	Bad/ Not Bad	ความถูกต้องของการเปลี่ยนแปลง		ความสมบูรณ์ของการเปลี่ยนแปลง	
	NOSS		1.402 – 0.114 NAggH	0.992 – 0.040 NGenH	1.004 – 0.116 * NAggH	1.079– 0.009 * NA
Ex41.xml	0	Not bad	1.402	0.952	1.004	0.836
Ex45.xml	0	Not bad	1.402	0.872	1.004	0.827

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวธันยวัต จันทร์เปีย เกิดเมื่อวันที่ 21 กรกฎาคม พ.ศ. 2522 สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต (วท.บ.) สาขาศาสตร์คอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ เมื่อปีการศึกษา 2542 และเข้าศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2547

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย