การพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

นายอติโรจน์ สืบวงศ์คล้าย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2550
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DEVELOPMENT OF A FRAMEWORK FOR OPTIMIZING OBJECT-ORIENTED DESIGN MODEL

Mr. Atiroj Suebwongklay

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 2007

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ
โดย	นายอติโรจน์ สืบวงศ์คล้าย
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์)

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

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ นครทิพย์ พร้อมพูล)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

นายอติโรจน์ สืบวงศ์คล้าย: การพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ. (DEVELOPMENT OF A FRAMEWORK FOR OPTIMIZING OBJECT-ORIENTED DESIGN) อ. ที่ปรึกษา: รศ.ดร.พรศิริ หมื่นไชยศรี, 108 หน้า.

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบซอฟต์แวร์เชิงวัตถุ โดยประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งในการหารูปแบบที่เหมาะสม ซึ่งก็คือรูปแบบที่มีคุณภาพด้านที่สนใจที่ดีที่สุดของโมเดลการออกแบบหนึ่งๆ ในกระบวนการหารูปแบบที่เหมาะสมโครงร่างจะวัดคุณภาพของโมเดลการออกแบบด้วยฟังก์ชันการประเมินค่าที่สอดคล้องกับคุณภาพด้านที่ต้องการปรับปรุง ซึ่งฟังก์ชันการประเมินค่าที่ใช้ในโครงร่างมีลักษณะเป็นปลั๊กอินเพื่อทำให้โครงร่างนี้มีความยืดหยุ่นในการปรับปรุงคุณภาพหลายๆ ด้านของโมเดลการออกแบบใดๆ

งานวิจัยนี้ทดสอบการปรับปรุงคุณภาพของโมเดลการออกแบบ 3 ประเภทได้แก่ การลดจำนวนของร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวี, การลดจำนวนร่องรอยที่ไม่ดีแบบมิดเดิลแมนและ การปรับปรุงการเกาะกลุ่มกันและเข้าคู่กันของคลาสต่างๆ โดยทดสอบกับโมเดลการออกแบบตัวอย่าง 3 โมเดล ซึ่งแต่ละโมเดลใช้สำหรับการทดสอบคุณภาพในแต่ละด้าน การทดสอบได้ทำการวัดคุณภาพของโมเดลการออกแบบด้วยฟังก์ชันการประเมินค่าที่สอดคล้องกับคุณภาพแต่ละด้านก่อนและหลังการปรับปรุงคุณภาพโมเดลการออกแบบของโครงร่าง ผลการทดสอบพบว่าโครงร่างที่นำเสนอสามารถปรับปรุงคุณภาพของโมเดลการออกแบบทั้ง 3 ประเภทได้ดีขึ้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.... วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....

สาขาวิชา....วิทยาศาสตร์คอมพิวเตอร์.....ลายมือชื่ออาจารย์ที่ปรึกษา.....

ปีการศึกษา .....2550.....

4771461021 : MAJOR COMPUTER SCIENCE

KEY WORD: OBJECT-ORIENTED SOFTWARE DESIGN / COMBINATORIAL OPTIMIZATION / SIMULATED ANNEALING ALGORITHM / LOCAL OPTIMIZATION / REFACTORING / DESIGN FLAWS/ BAD-SMELL / OBJECT-ORIENTED METRICS

ATIROJ SUEBWONGKLAY: DEVELOPMENT OF A FRAMEWORK FOR OPTIMIZING OBJECT-ORIENTED DESIGN. THESIS ADVISOR: ASSOC PROF PORNSIRI MUENCHAISRI, Ph.D., 108 pp.

This thesis proposes development of a framework for optimizing object-oriented software design model. The developed framework uses simulated annealing algorithm with local optimization technique for finding the configuration of an object-oriented design model which has the optimum value for a specified quality attribute of design model. The framework measures the quality of the design model by using the cost function which is corresponding to the required quality. The cost function is implemented as the plug-in for the framework in order to provide the flexibility for optimizing design model in various types of quality attributes.

The framework was evaluated by conducting three experiments to improve three types of design model's quality. Those included the reduction of Feature Envy Bad-Smell, the reduction of Middle Man Bad-Smell and the improvement of Cohesion and Coupling of classes. The experiments were made on three design models. Each model was used for testing a different quality type. The cost functions for each required quality before and after optimizing the design model's quality by the framework were collected. The result shows that the framework can improve three types of quality of the design models.

Department.... Computer Engineering...Student's signature...อติโรจน์ สืบวงศ์คลัย.....
Field of study.... Computer Science.....Advisor's signature....ผศ.ดร. พรศิริ.....
Academic year ...2007.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือจาก รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี อาจารย์ที่ปรึกษาวิทยานิพนธ์ของข้าพเจ้า ขอกราบขอบพระคุณอาจารย์ที่ให้คำแนะนำ และข้อเสนอแนะต่างๆ ตลอดระยะเวลาของการทำวิทยานิพนธ์ของข้าพเจ้า จนสำเร็จลุล่วงได้ด้วยดี

ขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์ เป็นประธานกรรมการ ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ และผู้ช่วยศาสตราจารย์ นครทิพย์ พร้อมพูล เป็นกรรมการสอบวิทยานิพนธ์ ซึ่งได้สละเวลาและให้คำแนะนำต่างๆ ในการสอบวิทยานิพนธ์ของข้าพเจ้าอย่างดียิ่ง

ขอขอบคุณ คุณธันยวัต จันทร์เปียสำหรับคำแนะนำ คำปรึกษาและความช่วยเหลือในทุกๆ ด้านในการทำวิทยานิพนธ์เล่มนี้

ท้ายสุดนี้ขอกราบขอบพระคุณบิดา มารดา และเพื่อนๆ ทุกคนที่คอยเป็นกำลังใจและให้ความสนับสนุนมาโดยตลอด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ฎ
สารบัญภาพ ..... ฎ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขตงานวิจัย 2
1.4 ขั้นตอนการวิจัย 3
1.5 ประโยชน์ที่จะได้รับ 4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5
2.1 ทฤษฎีที่เกี่ยวข้อง 5
2.1.1 โมเดลการออกแบบที่แสดงด้วยแผนภาพยูเอ็มแอล (Design Model with UML Diagram) [1] 5
2.1.2 การวัดซอฟต์แวร์ (Software Measurement) [2] 7
2.1.3 วิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่ (Combinatorial Optimization) [3] 8
2.1.4 วิธีซีมูเล็ทเต็ดแอนนิลลิง (Simulated Annealing Algorithm: SA) [4] 9
2.1.5 วิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิง (Adaptive Simulated Annealing Algorithm: ASA)[5] [6] [7] 11
2.1.6 ซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง (Simulated Annealing with Local Optimization: SALO) [8] 15

2.1.7 คุณสมบัติการเข้าคู่กัน (Coupling) และการเกาะกลุ่มกัน (Cohesion) ของคลาส [9] 18

2.1.8 ร่องรอยที่ไม่ดี (Bad-Smell) [10]..... 18

2.1.9 ระบบพิกัดไฮเปอร์สเฟียริคอล (Hyperspherical coordinates) [11] 19

2.2 งานวิจัยที่เกี่ยวข้อง 20

2.2.1 งานวิจัย “A Stochastic Approach to Automated Design Improvement” [12] 20

2.2.2 งานวิจัย “Genetic Algorithm Based Restructuring of Object-Oriented Designs Using Metrics” [9] 20

2.2.3 งานวิจัย “การตรวจจับข้อบกพร่องในขั้นตอนการออกแบบเพื่อปรับปรุงความสามารถในการเปลี่ยนแปลงซอฟต์แวร์เชิงวัตถุ” [13]..... 24

บทที่ 3 การพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ 26

3.1 ขั้นตอนการพัฒนาโครงร่าง..... 27

3.1.1 การกำหนดรูปแบบของโมเดลการออกแบบที่จะใช้ในโครงร่าง..... 27

3.1.2 การกำหนดวิธีการเปลี่ยนแปลงโมเดลการออกแบบ..... 30

3.1.3 การกำหนดวิธีการประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งในงานการปรับปรุงคุณภาพของโมเดลการออกแบบ 30

3.1.4 การพัฒนาโครงร่างที่ใช้ปรับปรุงคุณภาพของโมเดลการออกแบบ 30

3.1.5 การพัฒนาปลั๊กอินการปรับปรุงโมเดลการออกแบบ 43

3.1.6 การพัฒนาเครื่องมือเพื่อทดสอบโครงร่างที่ได้สร้างขึ้น 45

3.1.7 การตรวจสอบผลลัพธ์การหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุของเครื่องมือที่ได้จัดทำขึ้น 45

3.2 การประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง..... 46

3.2.1 ลักษณะอินพุทของวิธีซีมูเล็ทเต็ดแอนนิลลิงในโครงร่าง..... 46

3.2.2 การกำหนดช่วงของค่าที่เป็นไปได้ให้กับอาเรย์คอลยูสกราฟแต่ละช่อง..... 46

3.2.3 การใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงกับพารามิเตอร์ที่เป็นจำนวนเต็ม 46
3.2.4 วิธีที่เลือกใช้ในการสุ่มหาเวกเตอร์ของการหาค่าเหมาะสมเฉพาะแห่ง..... 47
3.2.5 เงื่อนไขการจบการทำงานของวิธีซีมูเล็ทเต็ดแอนนิลลิงที่เลือกใช้..... 48
3.3 ตัวอย่างกระบวนวิธีในการเปลี่ยนแปลงโมเดลการออกแบบ..... 48
บทที่ 4 โครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ..... 53
4.1 โครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ..... 53
4.1.1 UMLModel 54
4.1.2 UMLIOMng 55
4.1.3 UMLIOEngines..... 55
4.1.4 OptimizationMng..... 55
4.1.5 OptimizerEngines 56
4.1.6 UMLOptimizer..... 56
4.1.7 UMLCostFnEngines 56
4.1.8 UMLMetricMng 57
4.1.9 UMLMetrics 57
4.2 ไฟล์คอนฟิกูเรชัน..... 57
4.3 สภาพแวดล้อมในการพัฒนา 58
บทที่ 5 การทดสอบการปรับปรุงคุณภาพของโมเดลการออกแบบด้วยโครงร่าง 59
5.1 การทดสอบฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบ 60
5.2 การทดสอบฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบ 73
5.3 การทดสอบฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบ 82
บทที่ 6 บทสรุปและข้อเสนอแนะ 93

บทสรุป 93
ข้อจำกัดและข้อเสนอแนะ 94
รายการอ้างอิง 96
ภาคผนวก 98
ประวัติผู้เขียนวิทยานิพนธ์ 108

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 5.1 ค่ามาตรวัด *NSMsg* และ *NMsg* ของเมทธอดต่างๆ ในโมเดลต้นแบบของระบบที่ 1 62
ตารางที่ 5.2 ค่ามาตรวัด *NSMsg* และ *NMsg* ของเมทธอดต่างๆ ในโมเดลที่ถูกปรับปรุงแล้วของระบบที่ 1 63
ตารางที่ 5.3 ค่ามาตรวัด *WOD* ของคลาสต่างๆ ในโมเดลต้นแบบของระบบที่ 2 74
ตารางที่ 5.4 ค่ามาตรวัด *WOD* ของคลาสต่างๆ ในโมเดลที่ถูกปรับปรุงแล้วของระบบที่ 2 75
ตารางที่ 5.5 ค่ามาตรวัด *COH(C)* และ *COPIA (C)* ของคลาสต่างๆ ในโมเดลต้นแบบของระบบที่ 3 83
ตารางที่ 5.6 ค่ามาตรวัด *COH(C)* และ *COPIA (C)* ของคลาสต่างๆ ในโมเดลที่ถูกปรับปรุงแล้วของระบบที่ 3 84

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 2.1 ตัวอย่างแผนภาพคลาส 6
รูปที่ 2.2 ตัวอย่างแผนภาพซีเควนซ์ 7
รูปที่ 2.3 รหัสเทียมของวิธีซีมูเล็ทเต็ดแอนนิลลิงทั่วๆ ไป 10
รูปที่ 2.4 ตัวอย่างกราฟ (Graph) ของฟังก์ชันการประเมินค่าใดๆ 11
รูปที่ 2.5 กราฟฟังก์ชันความน่าจะเป็นของวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิง 14
รูปที่ 2.6 รหัสเทียมของซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง 16
รูปที่ 2.7 รหัสเทียมของฟังก์ชันการหาค่าเหมาะสมเฉพาะแห่ง 17
รูปที่ 2.8 รูปแบบการหาพิกัดคาร์ทีเซียนจากพิกัดไฮเปอร์สเฟียริคอล 19
รูปที่ 2.9 ตัวอย่างบางส่วนของคอลยูสกราฟที่สอดคล้องกับโมเดลการออกแบบในรูปที่ 2.1 21
รูปที่ 2.10 ตัวอย่างอาเรย์ ที่สอดคล้องกับกราฟในรูปที่ 2.9 21
รูปที่ 3.1 ขั้นตอนการพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ 26
รูปที่ 3.2 รูปแบบของแผนภาพคลาสที่เป็นไปตามข้อกำหนด 28
รูปที่ 3.3 รูปแบบของแผนภาพซีเควนซ์ที่สัมพันธ์กับ แผนภาพคลาสในรูปที่ 3.2 ที่เป็นไปตามข้อกำหนด 29
รูปที่ 3.4 ส่วนประกอบของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ 31
รูปที่ 3.5 การทำงานของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ 32
รูปที่ 3.6 คอลยูสกราฟที่สร้างมาจากข้อมูลของโมเดลการออกแบบในรูปที่ 3.2 และรูปที่ 3.3 33
รูปที่ 3.7 อาเรย์ที่สอดคล้องกับกราฟในรูปที่ 3.6 34
รูปที่ 3.8 ตัวอย่างคลาสที่ได้จากการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบด้วยวิธีของ [9] 34
รูปที่ 3.9 รูปแบบของการระบุโรลในโมเดลการออกแบบผลลัพธ์ 35
รูปที่ 3.10 รูปแบบของการระบุบริบทการถูกเรียกใช้ของข้อมูลจากแผนภาพคลาสต้นแบบในรูปที่ 3.2 36
รูปที่ 3.11 รูปแบบของการระบุบริบทการถูกเรียกใช้ของข้อมูลจากแผนภาพซีเควนซ์ต้นแบบในรูปที่ 3.3 37

รูปที่ 3.12 ตัวอย่างแผนภาพคลาสที่ได้จากการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบด้วยวิธีที่ปรับปรุงขึ้น 39
รูปที่ 3.13 ตัวอย่างแผนภาพซีเควนซ์ที่ได้จากการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบด้วยวิธีที่ปรับปรุงขึ้น 39
รูปที่ 3.14 อาเรย์ที่สอดคล้องกับรูปที่ 3.8 41
รูปที่ 3.16 อาเรย์ตั้งต้นของการทำงาน 49
รูปที่ 3.17 รูปแบบของสถานะที่เข้ามาเป็นอินพุท 49
รูปที่ 3.18 แผนภาพคลาสหลังจากการทำงานรอบที่ 1 49
รูปที่ 3.19 แผนภาพซีเควนซ์หลังจากการทำงานรอบที่ 1 49
รูปที่ 3.20 อาเรย์ตั้งต้นของการทำงานรอบที่ 2 50
รูปที่ 3.21 แผนภาพคลาสหลังจากการทำงานรอบที่ 2 50
รูปที่ 3.22 แผนภาพซีเควนซ์หลังจากการทำงานรอบที่ 2 50
รูปที่ 3.23 อาเรย์ตั้งต้นของการทำงานรอบที่ 3 51
รูปที่ 3.24 แผนภาพคลาสหลังจากการทำงานรอบที่ 3 51
รูปที่ 3.25 แผนภาพซีเควนซ์หลังจากการทำงานรอบที่ 3 51
รูปที่ 4.1 การทำงานของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ 53
รูปที่ 4.2 ส่วนประกอบของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ 54
รูปที่ 4.3 ลำดับชั้นความเป็นองค์ประกอบกันของออบเจ็กต์ต่างๆ ของคอมโพเนนท์ 55
UMLModel 55
รูปที่ 4.4 เนื้อหาของไฟล์คอนฟิกูเรชันของวิธีซีมูเล็ทเต็ดแอนนิลลิงที่พัฒนาขึ้น 58
รูปที่ 5.1 แผนภาพคลาสต้นแบบของระบบที่ 1 60
รูปที่ 5.2 แผนภาพซีเควนซ์ Register ต้นแบบของระบบที่ 1 64
รูปที่ 5.3 แผนภาพซีเควนซ์ RentCar ต้นแบบของระบบที่ 1 65
รูปที่ 5.4 แผนภาพซีเควนซ์ ReturnCar ต้นแบบของระบบที่ 1 66
รูปที่ 5.5 แผนภาพคลาสที่ถูกปรับปรุงแล้วของระบบที่ 1 67
รูปที่ 5.6 แผนภาพซีเควนซ์ Register ที่ถูกปรับปรุงแล้วของระบบที่ 1 68
รูปที่ 5.7 แผนภาพซีเควนซ์ RentCar ที่ถูกปรับปรุงแล้วของระบบที่ 1 69
รูปที่ 5.7 แผนภาพซีเควนซ์ RentCar ที่ถูกปรับปรุงแล้วของระบบที่ 1 (ต่อ) 70

รูปที่ 5.8 แผนภาพซีเควนซ์ ReturnCar ที่ถูกปรับปรุงแล้วของระบบที่ 1 71
รูปที่ 5.8 แผนภาพซีเควนซ์ ReturnCar ที่ถูกปรับปรุงแล้วของระบบที่ 1 (ต่อ)..... 72
รูปที่ 5.9แผนภาพคลาสต้นแบบของระบบที่ 2 73
รูปที่ 5.10 แผนภาพซีเควนซ์ Shopping ต้นแบบของระบบที่ 2..... 76
รูปที่ 5.11 แผนภาพซีเควนซ์ PlaceOrder ต้นแบบของระบบที่ 2 77
รูปที่ 5.12 แผนภาพคลาสที่ถูกปรับปรุงแล้วของระบบที่ 2..... 78
รูปที่ 5.13 แผนภาพซีเควนซ์ Shopping ที่ถูกปรับปรุงแล้วของระบบที่ 2 79
รูปที่ 5.14 แผนภาพซีเควนซ์ PlaceOrder ที่ถูกปรับปรุงแล้วของระบบที่ 2..... 80
รูปที่ 5.14 แผนภาพซีเควนซ์ PlaceOrder ที่ถูกปรับปรุงแล้วของระบบที่ 2 (ต่อ)..... 81
รูปที่ 5.15 แผนภาพคลาสต้นแบบของระบบที่ 3 85
รูปที่ 5.16 แผนภาพซีเควนซ์ ReturnGoods ต้นแบบของระบบที่ 3 86
รูปที่ 5.17 แผนภาพซีเควนซ์ GetApproval ต้นแบบของระบบที่ 3 87
รูปที่ 5.18 แผนภาพคลาสที่ถูกปรับปรุงแล้วของระบบที่ 3..... 88
รูปที่ 5.19 แผนภาพซีเควนซ์ ReturnGoods ที่ถูกปรับปรุงแล้วของระบบที่ 3..... 89
รูปที่ 5.19 แผนภาพซีเควนซ์ ReturnGoods ที่ถูกปรับปรุงแล้วของระบบที่ 3 (ต่อ)..... 90
รูปที่ 5.20 แผนภาพซีเควนซ์ GetApproval ที่ถูกปรับปรุงแล้วของระบบที่ 3..... 91
รูปที่ 5.20 แผนภาพซีเควนซ์ GetApproval ที่ถูกปรับปรุงแล้วของระบบที่ 3 (ต่อ)..... 92
รูปที่ ก.1 โปรแกรมประยุกต์ที่ใช้ทดสอบโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ..... 99
รูปที่ ก.2 หน้าจอสำหรับเลือกปลั๊กอินที่ใช้ประเมินคุณภาพ..... 99
รูปที่ ก.3 หน้าจอสำหรับเลือกไฟล์ของโมเดลการออกแบบต้นทาง..... 100
รูปที่ ก.4 หน้าจอสำหรับเลือกตำแหน่งสำหรับบันทึกบันทึกโมเดลการออกแบบผลลัพธ์ที่ได้ 101
รูปที่ ก.5 รายละเอียดของโมเดลการออกแบบต้นทางและของโมเดลการออกแบบผลลัพธ์ที่บันทึกในแฟ้มข้อมูลภายนอก..... 101

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP) เป็นเทคนิคที่ได้รับความนิยมกันอย่างแพร่หลายในปัจจุบัน เนื่องจากการเขียนโปรแกรมในลักษณะนี้จะสนับสนุนคุณสมบัติทางด้าน การสืบทอด (Inheritance) การห่อหุ้ม (Encapsulation) และการมีหลายรูปลักษณ์ (Polymorphism) ของคลาส (Class) การออกแบบซอฟต์แวร์เชิงวัตถุ (Object-Oriented Software) เพื่อนำไปใช้อ้างอิงในขั้นตอนการเขียนโปรแกรมนั้นนิยมอธิบายด้วย ภาษายูเอ็มแอล (UML: Unified Modeling Language) ที่เขียนอยู่ในรูปแบบของแผนภาพยูเอ็มแอล (UML Diagram) โดยที่การออกแบบที่ดีในขั้นตอนของการออกแบบนั้นจะนำมาซึ่งคุณสมบัติที่ดีของระบบ เช่น การสนับสนุนการนำกลับมาใช้ใหม่ (Reusability) ได้สูง หรือ สามารถบำรุงรักษา (Maintainability) ได้ง่าย ซึ่งการออกแบบที่ไม่ดีอาจทำให้ซอฟต์แวร์ที่ได้สูญเสียคุณสมบัติที่ดีที่ต้องการไปได้

ในขั้นตอนการบำรุงรักษา (Maintenance phase) ซึ่งเกิดขึ้นได้เสมอกับซอฟต์แวร์ตัวหนึ่งๆ ก็สามารถที่จะทำให้โมเดลการออกแบบ (Design Model) ที่ดี ที่เคยมีอยู่ของซอฟต์แวร์นั้นๆ เปลี่ยนเป็นไม่ดีหรือดีน้อยลงได้ ถ้าทำการเปลี่ยนแปลงรหัสต้นทาง (Source code) และ/หรือ โมเดลการออกแบบ โดยที่ไม่ได้พิจารณาผลกระทบจากการเปลี่ยนแปลงรหัสต้นทางและโมเดลการออกแบบในขั้นตอนการบำรุงรักษานั้นๆ ให้ดี

การแก้ไขโมเดลการออกแบบที่ไม่ดีที่เกิดในขั้นตอนการออกแบบและขั้นตอนการบำรุงรักษานี้สามารถแก้ไขได้ด้วยการเปลี่ยนโครงสร้างโมเดลการออกแบบใหม่ (Design Restructuring) โดยอาจทำโดยผู้เชี่ยวชาญหรือทำอัตโนมัติโดยใช้เครื่องมือช่วยเหลือ ในปัจจุบันมีงานวิจัยหลายงานที่รองรับการเปลี่ยนโครงสร้างโมเดลการออกแบบใหม่อย่างอัตโนมัติ ส่วนหนึ่งของงานวิจัยดังกล่าวใช้หลักการวัด (Measurement Principle) โดยใช้มาตรวัด (Metric) ต่างๆ ที่กำหนดขึ้นมาเพื่อวัดคุณภาพของโมเดลการออกแบบ จากนั้นจึงทำการเปลี่ยนโครงสร้างของโมเดลการออกแบบใหม่ด้วยวิธีเฉพาะของตนเองที่สัมพันธ์กับมาตรวัดที่กำหนด และนำมาตรวัดชุดเดิมมาวัดคุณภาพของโมเดลการออกแบบอีกครั้งเพื่อตรวจสอบว่าหลังจากกระบวนการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบแล้วทำให้คุณภาพของโมเดลการออกแบบดีขึ้นหรือไม่

จากสมมติฐานในเรื่องความสัมพันธ์ของมาตรวัดต่างๆ กับวิธีการเปลี่ยนโครงสร้างโมเดลการออกแบบใหม่ ทำให้เกิดข้อจำกัดในเรื่องของคุณภาพของโมเดลการออกแบบ เนื่องจากงานวิจัยต่างๆ อาจใช้มาตรวัดที่ไม่เหมือนกัน ดังนั้นคุณภาพที่ถูกปรับปรุงโดยงานวิจัยแต่ละงานจึงอาจเป็นคนละด้านกัน และถูกผูกติดไปกับงานวิจัยงานนั้นๆ โดยที่ผู้ที่ใช้เครื่องมือช่วยเหลือในการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบที่สอดคล้องกับงานวิจัยดังกล่าว จะไม่สามารถเลือกได้หรือเลือกได้ในจำนวนจำกัดว่าคุณภาพในด้านใดบ้างที่ผู้ใช้ต้องการให้เครื่องมือทำการปรับปรุงให้ดีขึ้น

งานวิจัยนี้จึงนำเสนอโครงร่าง (Framework) ในการปรับปรุงคุณภาพของโมเดลการออกแบบที่ถูกอธิบายด้วยภาษายูเอ็มแอล โดยโครงร่างนี้เปิดกว้างสำหรับการประยุกต์ใช้มาตรวัดชนิดต่างๆ ได้อย่างมีอิสระ โดยที่ผู้ใช้สามารถกำหนดชุดของมาตรวัดใดๆ ที่เหมาะสมที่ใช้ในการระบุคุณภาพของโมเดลการออกแบบเองได้ ซึ่งงานวิจัยนี้สามารถนำไปประยุกต์ใช้ได้ทั้งในขั้นตอนของการออกแบบและขั้นตอนการบำรุงรักษา โดยไม่ขึ้นกับภาษาโปรแกรม (Programming Language) ใดๆ

1.2 วัตถุประสงค์

1.2.1 เพื่อประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง ในการเปลี่ยนโครงสร้างของโมเดลการออกแบบซอฟต์แวร์ให้ดีขึ้นโดยพิจารณาจากมาตรวัด

1.2.2 เพื่อออกแบบและพัฒนาโครงร่างที่ใช้ในการปรับปรุงคุณภาพของโมเดลการออกแบบเชิงวัตถุ

1.3 ขอบเขตงานวิจัย

1.3.1 งานวิจัยนี้ทำการปลี่ยนแปลงโมเดลการออกแบบในส่วนของแผนภาพคลาส (Class Diagram)

1.3.2 งานวิจัยนี้ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง (Simulated Annealing with Local Optimization) ในการปรับปรุงคุณภาพโมเดลการออกแบบ

1.3.3 งานวิจัยนี้สามารถใช้กับมาตรวัดใดๆ ที่สามารถวัดได้ในแผนภาพคลาส

1.3.4 โครงร่างที่ถูกสร้างในงานวิจัยนี้รับอินพุทเป็นโมเดลการออกแบบในรูปแบบของแฟ้มข้อความ (Text File) และหลังจากที่ได้ทำการปรับปรุงคุณภาพของโมเดล

การออกแบบแล้วโครงร่างนี้ก็จะให้เอาท์พุทออกมาเป็นแฟ้มข้อความที่สอดคล้องกับโมเดลการออกแบบที่ถูกปรับปรุงแล้ว

1.3.5 โครงร่างที่ถูกสร้างในงานวิจัยนี้สามารถปรับปรุงคุณภาพของโมเดลการออกแบบได้ โดยการใช้ร่วมกับปลั๊กอิน (Plug-in) มาตรวัดใดๆ ที่พัฒนาขึ้น โดยปลั๊กอินเหล่านี้ถูกใช้ได้มากกว่า 1 ตัวในการปรับปรุงคุณภาพของโมเดลการออกแบบครั้งหนึ่งๆ

1.3.6 โครงร่างที่ถูกสร้างในงานวิจัยนี้จะเลือกใช้ค่ามาตรวัดสองตัวที่ระบุถึงปริมาณการเข้าคู่กัน (Coupling) และปริมาณการเกาะกลุ่มกัน (Cohesion) ซึ่งถือเป็นมาตรวัดโดยปริยาย (Default Metrics) มาใช้ในกรณีที่ไม่ได้มีการลงทะเบียนปลั๊กอินมาตรวัดใดๆ เลยกับโครงร่าง ในการปรับปรุงคุณภาพของโมเดลการออกแบบครั้งหนึ่งๆ

1.3.7 โครงร่างที่ถูกสร้างในงานวิจัยนี้จะถูกทดสอบโดยใช้มาตรวัดอย่างน้อย 3 ตัว กับโมเดลการออกแบบของระบบซอฟต์แวร์ใดๆ อย่างน้อย 3 ระบบ โดยแต่ละระบบจะมีคลาสอย่างน้อย 10 คลาสขึ้นไป

1.3.8 งานวิจัยนี้ไม่รวมไปถึงขั้นตอนการแปลงอัตโนมัติระหว่างเอกสารการออกแบบหรือรหัสต้นทางไปเป็นโมเดลการออกแบบที่ใช้ได้กับโครงร่าง (Framework)

1.3.9 พัฒนาเครื่องมือและใช้เครื่องมือบนระบบปฏิบัติการวินโดว์ (Windows) ตั้งแต่รุ่น 2000 ขึ้นไป

1.4 ขั้นตอนการวิจัย

1.4.1 ศึกษาวิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง

1.4.2 ศึกษาคุณลักษณะของมาตรวัดที่ใช้กับโมเดลการออกแบบเชิงวัตถุที่มีอยู่ในปัจจุบัน

1.4.3 กำหนดโมเดลของโมเดลการออกแบบที่เป็นกลางที่จะใช้ในการหาค่ามาตรวัด

1.4.4 ศึกษาวิธีการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบที่เหมาะสม

1.4.5 นิยามส่วนการคำนวณฟังก์ชันการประเมินค่า (Cost Function) ของวิธีซีมูเล็ท-เต็ดแอนนิลลิงที่เหมาะสม

1.4.6 ศึกษาวิธีประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งกับงานปรับปรุงคุณภาพโมเดลการออกแบบ

1.4.7 จัดทำโครงร่างและตัวอย่างของปลั๊กอินมาตรวัด

1.4.8 ออกแบบและพัฒนาเครื่องมือที่ใช้ในการปรับปรุงคุณภาพของโมเดลการออกแบบ

1.4.9 ประเมินคุณภาพของโครงร่างที่นำเสนอ

1.4.10 สรุปผลการวิจัย และข้อเสนอแนะ

1.4.11 จัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่จะได้รับ

1.5.1 สามารถปรับปรุงคุณภาพของโมเดลการออกแบบเชิงวัตถุด้วยวิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งโดยประเมินคุณภาพด้วยฟังก์ชันการประเมินค่าสำหรับคุณภาพในด้านต่างๆ

1.5.2 โครงร่างที่ได้สามารถทำการปรับปรุงโมเดลการออกแบบได้ทั้งในขั้นตอนการออกแบบและขั้นตอนการบำรุงรักษา

1.5.3 ได้เครื่องมือที่จัดทำขึ้นเพื่อทำการปรับปรุงคุณภาพของโมเดลการออกแบบเชิงวัตถุ

1.5.4 ได้แนวคิดที่จะทำการปรังปรุงคุณภาพของโมเดลการออกแบบเชิงวัตถุกับแผนภาพยูเอ็มแอล (UML Diagram) อื่นๆ นอกเหนือจากแผนภาพคลาส

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

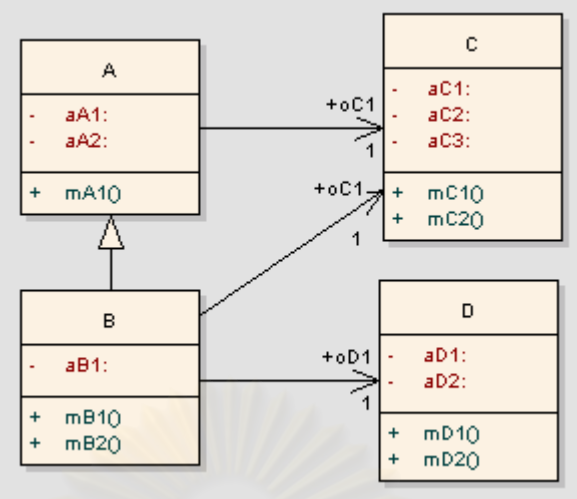
ทฤษฎีที่เกี่ยวข้องในงานวิจัย ได้แก่ โมเดลการออกแบบที่แสดงด้วยแผนภาพยูเอ็มแอล, การวัดซอฟต์แวร์, วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง ซึ่งมีรายละเอียดดังนี้

2.1.1 โมเดลการออกแบบที่แสดงด้วยแผนภาพยูเอ็มแอล (Design Model with UML Diagram) [1]

แผนภาพยูเอ็มแอลคือแผนภาพที่นิยมใช้ในการอธิบายระบบซอฟต์แวร์เชิงวัตถุ แผนภาพนี้ถูกเขียนด้วยภาษายูเอ็มแอลที่มีคุณสมบัติเป็นกึ่งรูปนัย (Semi-Formal) รูปแบบของภาษายูเอ็มแอล จะมีสัญลักษณ์ที่มีความหมายในตัวเอาไว้ประกอบกันเป็นแผนภาพเพื่อใช้อธิบายบริบทหนึ่งๆ การที่จะอธิบายระบบใดๆ ด้วยวิธีนี้อาจจะต้องใช้แผนภาพหลายๆ แบบประกอบกัน ในงานวิจัยนี้เกี่ยวข้องกับแผนภาพยูเอ็มแอลเพียงสองแบบเท่านั้นคือ แผนภาพคลาส (Class Diagram) และ แผนภาพซีเควนซ์ (Sequence Diagram)

2.1.1.1 แผนภาพคลาส

รูปที่ 2.1 แสดงแผนภาพคลาสของระบบซอฟต์แวร์หนึ่งที่ประกอบไปด้วยคลาส A, B, C และ D โดยที่แต่ละคลาสในรูปจะถูกแสดงด้วยสี่เหลี่ยมที่มีส่วนประกอบอยู่ 3 ส่วน ส่วนบนสุดคือชื่อคลาส ที่เหลือเป็นการแสดงสมาชิก (Member) ภายในคลาสหนึ่งคลาส โดยที่ส่วนกลางคือแอตทริบิวต์ (Attribute) ในคลาส และ ส่วนล่างสุดคือเมทธอด (Method) ของคลาสนั้นๆ รูปที่ 2.1 แสดงความสัมพันธ์ระหว่างคลาสแบบที่นิยมใช้กัน โดยความสัมพันธ์ระหว่างคลาส A กับ B เป็นแบบเจนเนอรัลไลเซชัน (Generalization) แสดงการสืบทอดกันกล่าวคือ คลาส B รับเอาคุณสมบัติต่างๆ ซึ่งก็คือแอตทริบิวต์และเมทธอดของคลาส A มาเป็นของตนเอง ส่วนเส้นความสัมพันธ์เส้นอื่นที่เหลือแสดงถึงความสัมพันธ์แบบ แอสโซซิเอชัน (Association) ซึ่งมีความหมายว่าคลาสหนึ่งเรียกใช้เมทธอดหรือแอตทริบิวต์ของอีกคลาสหนึ่ง เช่นจากรูปคลาส B เรียกใช้เมทธอดหรือแอตทริบิวต์ของคลาส D เป็นต้น



รูปที่ 2.1 ตัวอย่างแผนภาพคลาส

นอกจากนี้ที่ปลายทั้ง 2 ด้านของความสัมพันธ์แบบแอสโซซิเอชันยังสามารถบอกข้อมูลสำคัญได้โดยในจำนวนนั้นมีอยู่ 2 ชนิดที่น่าสนใจคือ โรล (Role) และ มัลติพลิซิตี (Multiplicity)

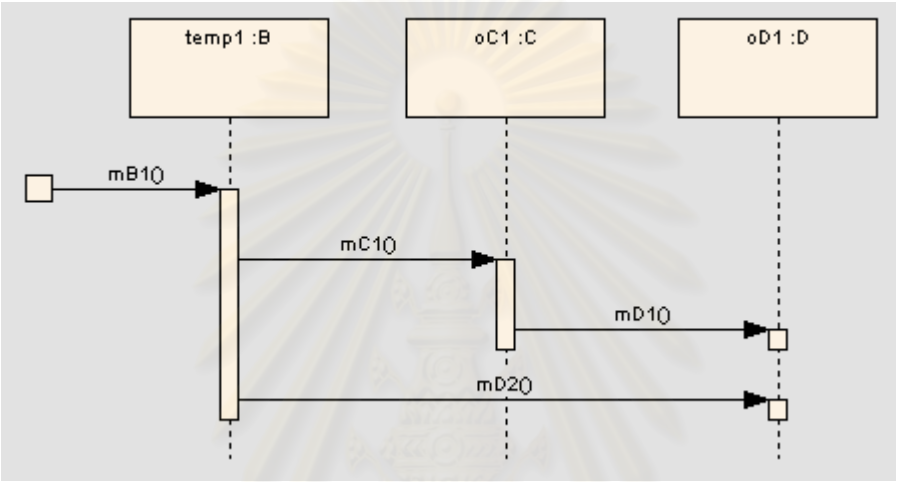
โดยที่การระบุโรลนั้นจะเป็นการกำหนดชื่อของออบเจ็กต์ (Object) ของคลาสฝั่งที่อยู่ติดกับปลายความสัมพันธ์แบบแอสโซซิเอชันที่ถูกระบุโรล ที่ถูกเรียกใช้โดยคลาสที่ติดกับปลายความสัมพันธ์แบบแอสโซซิเอชันด้านตรงข้าม ซึ่งสามารถตีความได้ว่าจะมีแอตทริบิวต์ชื่อเดียวกันกับโรลมีชนิดเป็นออบเจ็กต์ของคลาสด้านที่อยู่ติดกับปลายความสัมพันธ์ด้านที่ถูกระบุโรล โดยแอตทริบิวต์ตัวนี้จะเป็นสมาชิกของคลาสที่อยู่ติดกับปลายความสัมพันธ์ด้านตรงข้าม

การระบุมัลติพลิซิตีนั้นจะเป็นการบอกปริมาณของออบเจ็กต์ของคลาสที่อยู่ติดกับปลายความสัมพันธ์แบบแอสโซซิเอชันที่ถูกระบุมัลติพลิซิตีที่จะเกิดขึ้นทั้งหมดภายใต้ขอบเขตของความสัมพันธ์นั้น การระบุมัลติพลิซิตีที่นิยมใช้มีได้หลายแบบเช่น

- บอกเป็นจำนวนที่แน่นอน เช่น 1, 2, 3 เช่นนี้จะเป็นการบอกจำนวนที่แน่นอนของออบเจ็กต์ของคลาสที่อยู่ติดกับปลายความสัมพันธ์แบบแอสโซซิเอชันที่ถูกระบุมัลติพลิซิตีโดยที่ถ้าจำนวนออบเจ็กต์ที่ถูกระบุมีมากกว่า 1 หมายความว่าออบเจ็กต์เหล่านั้นจะถูกบรรจุในอาเรย์ (Array) หรือ คอลเลคชันคลาส (Collection Class) อื่นๆ อีกที
- บอกเป็นขอบเขตของจำนวน จะหมายความว่าออบเจ็กต์ที่เกิดขึ้นอาจมีจำนวนที่ไม่แน่นอนแต่จะมีจำนวนที่น้อยที่สุดไม่น้อยไปกว่าค่าตัวเลขตัวแรกและจะมีจำนวนไม่เกินค่าตัวเลขตัวที่ 2 อย่างเช่น
  - 0..1ออบเจ็กต์จะเกิดขึ้นหรือไม่เกิดก็ได้แต่ถ้าเกิดขึ้นจะต้องมีเพียงออบเจ็กต์เดียว

- 1..* ออบเจ็กต์จะต้องเกิดขึ้นอย่างน้อย 1 ออบเจ็กต์ และสามารถเกิดขึ้นได้ในจำนวนที่ไม่จำกัด
- 0..* ออบเจ็กต์จะเกิดขึ้นหรือไม่เกิดก็ได้และสามารถเกิดขึ้นได้ในจำนวนที่ไม่จำกัด

2.1.1.2 แผนภาพซีเควนซ์



รูปที่ 2.2 ตัวอย่างแผนภาพซีเควนซ์

รูปที่ 2.2 แสดงแผนภาพซีเควนซ์ซึ่งประกอบไปด้วยออบเจ็กต์ของคลาสต่างๆ และลำดับการส่งเมสเสจ (Message) ระหว่างออบเจ็กต์ซึ่งก็คือการขอเรียกใช้เมทธอดของอีกออบเจ็กต์หนึ่งนั่นเอง จากรูปออบเจ็กต์ temp1 ซึ่งเป็นออบเจ็กต์ของคลาส B ส่งเมสเสจไปยังออบเจ็กต์ oC1 ซึ่งเป็นออบเจ็กต์ของคลาส C เพื่อขอใช้เมทธอด mC1 ของออบเจ็กต์ oC1 ซึ่งภายในเมทธอด mC1 ก็ได้มีการส่งเมสเสจไปยังออบเจ็กต์ oD1 ซึ่งเป็นออบเจ็กต์ของคลาส D เพื่อขอใช้เมทธอด mD1 อีกที ถัดมาเป็นการส่งเมสเสจจากออบเจ็กต์ temp1 ไปยังออบเจ็กต์ oD1 เพื่อขอใช้เมทธอด mD2 และเครื่องหมายสี่เหลี่ยมขนาดเล็กที่อยู่ทางด้านซ้ายสุดของรูปที่ 2.2 เรียกว่า ไดอะแกรมเกท (Diagram Gate) ทำหน้าที่เป็นจุดรับหรือส่งเมสเสจที่เราไม่ต้องการระบุถึงผู้รับหรือผู้ส่ง

2.1.2 การวัดซอฟต์แวร์ (Software Measurement) [2]

การวัดเป็นพื้นฐานสำคัญสำหรับทุกศาสตร์ทางวิศวกรรม ซึ่งเป็นกระบวนการกำหนดค่าตัวเลข หรือสัญลักษณ์ ให้กับแอตทริบิวต์ (Attributes) ของเอนทิตี (Entities) หรือสิ่งที่อยู่ในโลกของความเป็นจริงที่เราสนใจ เพื่อบรรยายเอนทิตีนั้นให้อยู่ในรูปของกฎที่นิยามไว้ชัดเจน

สำหรับการวัดในทางวิศวกรรมซอฟต์แวร์ (Software Engineering) นั้น สามารถแบ่งได้เป็น 2 ประเภท คือ

- การวัดทางตรง (Direct Measurement) เป็นการวัดเฉพาะคุณลักษณะของสิ่งที่เราสนใจ โดยไม่นำคุณลักษณะ หรือ เอนทิตีอื่นมาเกี่ยวข้อง ผลที่ได้จากการวัดทางตรงทำให้ทราบลักษณะในด้านโครงสร้างของซอฟต์แวร์ เช่น การวัดความยาวของซอร์สโค้ด สามารถวัดได้จากการนับจำนวนบรรทัดทั้งหมดของโปรแกรม เป็นต้น

- การวัดทางอ้อม (Indirect Measurement) เป็นการวัดโดยการนำการวัดทางตรงมาประกอบการวัด เนื่องจากคุณลักษณะบางอย่างไม่สามารถวัดได้โดยตรง ผลของการวัดทางอ้อมคือ คุณภาพของซอฟต์แวร์ในด้านต่างๆ เช่น ความสามารถในการทดสอบ (Testability) ความสามารถในการบำรุงรักษา ความสามารถในการทำความเข้าใจระบบ (Understandability) เป็นต้น การที่จะวัดเพื่อทราบถึงคุณภาพของซอฟต์แวร์ในด้านใดด้านหนึ่งนั้น จะต้องกำหนดเกณฑ์ (Criteria) หรือปัจจัย (Factor) ในการวัด และแต่ละเกณฑ์จะมีการกำหนดมาตรวัดที่เกี่ยวข้องที่วัดได้จากการวัดทางตรงหรือจากการวัดทางอ้อม

การวัดทั้ง 2 ประเภทจำเป็นต้องใช้มาตรวัด เพื่อวัดคุณลักษณะของสิ่งที่เราสนใจ สำหรับการวัดซอฟต์แวร์เชิงวัตถุ มีมาตรวัดที่ใช้กันอยู่มากมาย ตัวอย่างเช่น จำนวนเมทธอดต่อคลาส (Weighted Methods per Class: WMC), ระดับความลึกของการถ่ายทอดคุณสมบัติ (Depth of Inheritance Hierarchy: DIT), จำนวนคลาสลูก (Number of Children: NOC), การเข้าคู่กันระหว่างวัตถุ (Coupling between Objects: CBO) และการขาดการเกาะกลุ่มกันของเมทธอด (Lack of Cohesion Method: LCOM) เป็นต้น

2.1.3 **วิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่** (Combinatorial Optimization) [3]

ปัญหาค่าเหมาะสมเชิงวิธีจัดหมู่เป็นปัญหาการหาค่าสูงสุดหรือต่ำสุดมีส่วนประกอบ 2 ส่วนด้วยกันคือ

- กลุ่มของพารามิเตอร์ (Parameter Set)

- ฟังก์ชันเป้าหมาย (Objective Function) ที่รับรูปแบบการจัดหมู่กันของพารามิเตอร์เป็นอินพุท

โดยเราต้องการหารูปแบบการจัดหมู่กันของค่าของพารามิเตอร์ที่ทำให้ค่าที่ได้จากฟังก์ชันเป้าหมายมีค่าต่ำหรือสูงที่สุดที่เป็นไปได้ ซึ่งรูปแบบการจัดหมู่กันของพารามิเตอร์ที่ได้ถือเป็นคำตอบของปัญหาประเภทนี้

2.1.4 **วิธีซีมูเล็ทเต็ดแอนนิลลิง** (Simulated Annealing Algorithm: SA) [4]

วิธีซีมูเล็ทเต็ดแอนนิลลิงเป็นวิธีที่ใช้การสุ่มในการแก้ปัญหาค่าเหมาะสมเชิงวิธีจัดหมู่โดยเลียนแบบวิธีมาจากกระบวนการอบเหนียวซึ่งเป็นการเพิ่มความร้อนให้กับวัตถุและทำให้เย็นลงอย่างช้าๆ เพื่อทำให้รูปแบบผลึกในวัตถุนั้นเรียงตัวกันอย่างมีระเบียบเป็นการเพิ่มความแข็งแรงให้กับวัตถุนั้นๆ โดยวิธีการนี้จะทำการสุ่มหารูปแบบการจัดหมู่กันของค่าพารามิเตอร์หรือเรียกว่าสถานะ (State) ต่างๆ มาแล้วทำการทดลองใส่เป็นอินพุทของฟังก์ชันเป้าหมายหรือเรียกว่าฟังก์ชันการประเมินค่า (Cost Function) ค่าที่เป็นผลลัพธ์จากฟังก์ชันนี้จะถูกเก็บไว้เพื่อนำไปเปรียบเทียบกับค่าผลลัพธ์จากสถานะแบบอื่นที่ทำการสุ่มมาอีก เพื่อหาสถานะที่ทำให้ได้ผลลัพธ์จากฟังก์ชันการประเมินค่าที่ดีที่สุด (ต่ำที่สุด หรือ สูงที่สุด ขึ้นอยู่กับลักษณะของฟังก์ชันการประเมินค่าที่เลือกใช้) ที่เป็นไปได้ตามข้อจำกัดทางด้านเวลาที่ถูกกำหนด วิธีซีมูเล็ทเต็ดแอนนิลลิงแบบที่ใช้ในการหาค่าต่ำสุดของฟังก์ชันการประเมินค่า สามารถอธิบายได้เป็นรหัสเทียม (Pseudo Code) รูปที่ 2.3

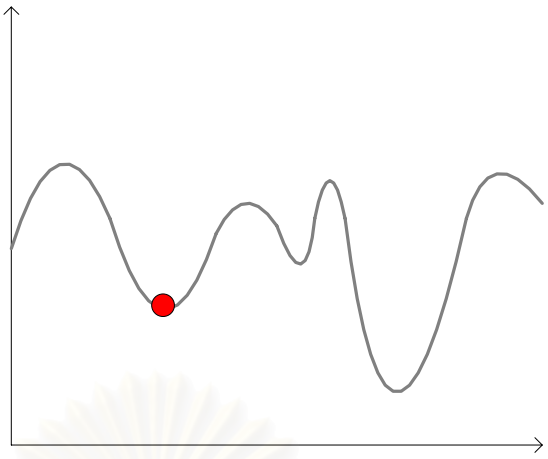
จากรหัสเทียมในรูปที่ 2.3 สามารถอธิบายวิธีซีมูเล็ทเต็ดแอนนิลลิงได้ว่า ในขั้นแรกจะต้องกำหนดสถานะเริ่มต้นพร้อมกับอุณหภูมิเริ่มต้นให้กับสภาพแวดล้อม ต่อมาก็จะเริ่มต้นการทำวงรอบใน (Inner Loop) ทำการเลือกสถานะใกล้เคียง (Neighbor State) โดยวิธีการสุ่ม (เปรียบเทียบได้กับการกระตุ้นวัตถุด้วยวิธีใดๆ ในกระบวนการทางฟิสิกส์) ของสถานะเดิมมาเพื่อทำการคำนวณค่าผลลัพธ์ (เปรียบเทียบได้กับค่าพลังงานสะสมของวัตถุในปัจจุบันในกระบวนการทางฟิสิกส์) จากฟังก์ชันการประเมินค่าของทั้ง 2 สถานะ ถ้าผลลัพธ์ของสถานะใหม่มีค่าน้อยกว่าก็ย่อมแสดงว่าสถานะใหม่นี้เป็นที่ต้องการมากกว่าสถานะเดิม แต่ถ้าผลลัพธ์ของสถานะใหม่มีค่ามากกว่าหรือเท่ากับผลลัพธ์ของสถานะเดิม วิธีซีมูเล็ทเต็ดแอนนิลลิงนี้ก็จะยังเปิดโอกาสในการยอมรับสถานะใหม่ที่ไม่ได้ดีกว่าสถานะเดิมนี้ได้ โดยจะคำนวณหาความน่าจะเป็นในการยอมรับสถานะใหม่นี้ต่อไป เมื่อสถานะหนึ่งถูกยอมรับแล้ว จึงมาตรวจสอบว่าอุณหภูมิของวัตถุมีความสมดุลหรือยังถ้าสมดุลแล้วก็จะหลุดออกจากวงรอบใน และทำการลดอุณหภูมิของสภาพแวดล้อมลงตามอัตราการลดอุณหภูมิที่ถูกกำหนดไว้ใน ฟังก์ชัน ***reduce_temperatue*** เมื่อถึงภาวะมั่นคงของระบบแล้วก็ให้หลุดออกจากกระบวนการทั้งหมดได้

```
BEGIN

  select an initial state x
  select an initial temperature T > 0

  REPEAT

    REPEAT

      y := generate_state(x)
      d := cost_funtion(y) – cost_funtion(x)

      IF d < 0 THEN
         x := y

      ELSE

      IF rand (0, l) < exp (- d / T) THEN
         x := y
      ENDIF

    UNTIL is_thermal_equilibrium() = True

    T : = reduce_temperature(T)

  UNTIL is_stable() = True

END
```

รูปที่ 2.3 รหัสเทียมของวิธีซีมูเล็ทเต็ดแอนนิลลิงทั่วๆ ไป

โดยที่

- ***generate_state:*** เป็นฟังก์ชันการสร้างสถานะใหม่จากสถานะเดิมที่ใส่เป็นอินพุท
- ***cost_function:*** คือฟังก์ชันการประเมินค่า
- ***rand:*** คือฟังก์ชันการสุ่มเลขจำนวนจริง (จากรูปที่ 2.3 ทำการสุ่มตั้งแต่ 0 ถึง 1)
- ***exp:*** คือฟังก์ชันการยกกำลังแบบเอ็กซ์โพเนนเชียล (Exponential)
- ***is_thermal_equilibrium:*** คือฟังก์ชันที่ล้อเลียนการตรวจสอบความสมดุลของอุณหภูมิ (Thermal Equilibrium) ในการเปลี่ยนแปลงสถานะในช่วงอุณหภูมิคงที่หนึ่งๆ
- ***reduce_temperatue:*** คือฟังก์ชันการลดอุณหภูมิ
- ***is_stable:*** เป็นฟังก์ชันการตรวจสอบความมั่นคงของระบบ



รูปที่ 2.4 ตัวอย่างกราฟ (Graph) ของฟังก์ชันการประเมินค่าใดๆ

สาเหตุที่ไม่สามารถเพิกเฉยกับสถานะใหม่ได้ในทันทีเมื่อพบว่ามีผลลัพธ์จากฟังก์ชันการประเมินของสถานะใหม่นี้ ที่ไม่ดีไปกว่าของสถานะเดิม คือ ประโยชน์ในการไม่ยึดติดกับภาวะที่ดีที่สุดเฉพาะแห่ง (Local Optima) เพื่อโอกาสที่จะได้ไปหาสถานะถัดๆ ไปที่มีความใกล้เคียงกับภาวะที่ดีที่สุดของทั้งหมดได้ จากรูปที่ 2.4 จุดที่อยู่ในกราฟแสดงถึงภาวะที่ดีที่สุดเฉพาะแห่งซึ่งไม่ได้เป็นภาวะที่ดีที่สุดของทั้งหมดของฟังก์ชันการประเมินค่าหนึ่งๆ

จากรูปที่ 2.3 จะเห็นว่าอุณหภูมิมีผลต่อการยอมรับสถานะใหม่ กล่าวคือยิ่งค่าอุณหภูมิของสภาพแวดล้อมลดลงเท่าใด ความน่าจะเป็นในการกระโดดออกจากภาวะที่ดีที่สุดเฉพาะแห่ง ก็ยิ่งจะน้อยลงไปเรื่อยๆ ดังนั้นอุณหภูมิเริ่มต้นที่จะถูกกำหนดจะต้องมีค่าสูงพอเพื่อสร้างโอกาสในการกระโดดออกจากภาวะที่ดีที่สุดเฉพาะแห่งของฟังก์ชันการประเมินค่าได้ วิธีการที่ใช้ในการประเมินการยอมรับสถานะใหม่นี้เป็นวิธีการมาตรฐานที่เรียกว่า วิธีเมโทรโพลิส (Metropolis Algorithm)

2.1.5 **วิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิง** (Adaptive Simulated Annealing Algorithm: ASA)[5] [6] [7]

วิธีซีมูเล็ทเต็ดแอนนิลลิงพัฒนาขึ้นมาหลายวิธีด้วยกัน ส่วนมากแต่ละวิธีที่เกิดขึ้นจะมุ่งเน้นทำการปรับปรุงในส่วนของฟังก์ชัน ***reduce_temperatue*** คือการทำให้จำนวนวงรอบในกระบวนการมีจำนวนน้อยที่สุดนั่นเอง

ทั้งนี้ฟังก์ชัน ***reduce_temperatue*** จะมีความสัมพันธ์กับฟังก์ชัน ***generate_state*** โดยตรง กล่าวคือถ้าสถานะใหม่ๆ ที่สร้างขึ้นมาไม่สอดคล้องกับอัตราการลดลงของอุณหภูมิแล้วจะไม่สามารถรับประกันการเข้าสู่ค่าใกล้เคียงของภาวะที่ดีที่สุดของทั้งหมดได้

วิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิงในที่นี้ใช้ฟังก์ชัน ***reduce_temperatue*** ที่ทำให้เกิดอัตราการลดลงของอุณหภูมิเร็วกว่าวิธีซีมูเล็ทเต็ดแอนนิลลิงแบบอื่นๆ และยิ่งแสดงประสิทธิภาพอย่างเห็นได้ชัดในกรณีที่ฟังก์ชันการประเมินค่ามีมิติ (Dimension) มากๆ กล่าวคือมีจำนวนพารามิเตอร์มากๆ แต่วิธีนี้สามารถใช้ได้กับระบบที่มีข้อจำกัดคือช่วงของค่า (Range of Value) ของพารามิเตอร์แต่ละตัวต้องเป็นจำนวนจำกัด คือต้องสามารถระบุค่าที่น้อยที่สุดและค่าที่มากที่สุดที่เป็นไปได้ของพารามิเตอร์แต่ละตัว

วิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิงจะพิจารณาพารามิเตอร์แต่ละตัวแยกจากกัน โดยการเปลี่ยนแปลงหนึ่งครั้งจากสถานะเดิมไปเป็นสถานะใหม่จะถูกกระทำที่พารามิเตอร์เพียงหนึ่งตัวเท่านั้น โดยค่าพารามิเตอร์ตัวอื่นๆ ยังคงค่าเดิมไม่เปลี่ยนแปลง ซึ่งวิธีนี้จะมีการหาค่าพารามิเตอร์ใหม่แต่ละตัวโดยขึ้นกับความไว (Sensitivity) ของพารามิเตอร์ที่มิตินั้นๆ เอง ดังนี้

$$p_{i(k+1)} = p_{ik} + q_{ik}(B - A), \qquad \text{สมการที่ 2.1}$$

โดยที่

- p_i แสดงถึงค่าของพารามิเตอร์ลำดับ(มิติ)ที่ i ของฟังก์ชันการประเมิน
- k แสดงถึงจำนวนรอบในกระบวนการสุ่มเพื่อหาสถานะใหม่
- A เป็นค่าที่น้อยที่สุดของพารามิเตอร์ในมิติที่ i
- B เป็นค่าที่มากที่สุดของพารามิเตอร์ในมิติที่ i
- q เป็นค่าโดยสุ่มของพารามิเตอร์ในมิติที่ i โดยที่

$$q_i = \mathrm{sgn}\left(u_i - \frac{1}{2}\right)\left(\left(1 + \frac{1}{Tg_i}\right)^{|2u_i - 1|} - 1\right)Tg_i, \qquad \text{สมการที่ 2.2}$$

โดยที่

- sgn เป็นฟังก์ชันที่ให้ค่าออกมาเป็น
  - -1 เมื่ออินพุทของฟังก์ชันน้อยกว่า 0
  - 0 เมื่ออินพุทของฟังก์ชันเท่ากับ 0

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- 1 เมื่ออินพุทของฟังก์ชันมากกว่า 0

- u_i ของพารามิเตอร์ในมิติที่ i มีค่าเป็น 0 หรือ 1 ขึ้นอยู่กับการสุ่ม
- Tg_i เป็นค่าอุณหภูมิย่อยที่ใช้ในการควบคุมการสร้างค่าใหม่จากค่าเดิม ของพารามิเตอร์ในมิติที่ i

วิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิงจะแยกอุณหภูมิที่ใช้ออกเป็น 2 ประเภทคือ อุณหภูมิย่อยในการควบคุมการสร้างพารามิเตอร์ใหม่ซึ่งจะใช้สัญลักษณ์แทนว่า Tg กับ อุณหภูมิที่ใช้ในการยอมรับสถานะใหม่ซึ่งจะใช้สัญลักษณ์แทนว่า Ta อุณหภูมิทั้ง 2 แบบจะสามารถสร้างออกมาได้จากฟังก์ชันที่มีรูปร่างหน้าตาคล้ายกันดังนี้

$$T_{i(k+1)} = T_{i0} \exp(-Ck^{\frac{1}{N}}), \qquad \text{สมการที่ 2.3}$$

โดยที่ถ้าเป็นอุณหภูมิย่อยในการควบคุมการสร้างพารามิเตอร์ใหม่ (Tg) ค่า k จะหมายถึงจำนวนรอบในการสร้างพารามิเตอร์ใหม่ แต่กับอุณหภูมิที่ใช้ในการยอมรับสถานะใหม่ (Ta) ค่า k จะหมายถึงจำนวนสถานะที่ถูกยอมรับมาแล้วในอดีตทั้งหมด ค่า C และค่า N ในสมการที่ 2.3 คือค่าคงที่ที่เหมาะสมกับระบบหนึ่งๆ และ ค่าจำนวนพารามิเตอร์ (มิติ) ของฟังก์ชันการประเมินค่าตามลำดับ

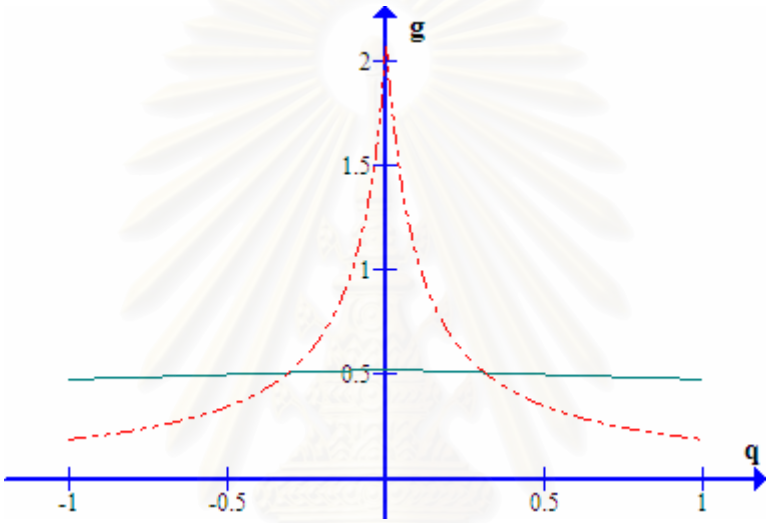
เนื่องจากวิธีซีมูเล็ทเต็ดแอนนิลลิงทุกๆ วิธี เป็นการสุ่มหาสถานะใหม่แล้วตรวจสอบว่าดีกว่าสถานะที่มีอยู่เดิมหรือไม่ ดังนั้นวิธีซีมูเล็ทเต็ดแอนนิลลิงวิธีหนึ่งจึงจะต้องเกี่ยวข้องกับฟังก์ชันความน่าจะเป็น (Probability Density Function: PDF) แบบต่างๆ ที่วิธีนั้นเลือกใช้ สำหรับวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิงนี้ เลือกใช้ฟังก์ชันความน่าจะเป็นที่อธิบายได้ดังนี้

$$g_{i(q_i)} = \frac{1}{2(|q_i| + Tg_i)\ln(1+\frac{1}{Tg_i})}, \qquad \text{สมการที่ 2.4}$$

โดยที่

- g_i คือฟังก์ชันความน่าจะเป็นในการสุ่มเลือกค่า q_i ของพารามิเตอร์ในมิติที่ i ตามสมการที่ 2.2
- ฟังก์ชันนี้ต้องถูกจำกัดค่าโดเมน (Domain) ไว้ที่ช่วง -1 ถึง 1 จึงจะถือว่าเป็นฟังก์ชันความน่าจะเป็น กล่าวคือ q_i จะต้องมีค่าอยู่ในช่วง -1 ถึง 1 เท่านั้น

การอธิบายพฤติกรรมของการสุ่มเลือกค่าใหม่ให้กับพารามิเตอร์ในมิติใดๆ ของวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิง สามารถูกอธิบายได้จากกราฟของ สมการที่ 2.4 ดังรูปที่ 2.5



รูปที่ 2.5 กราฟฟังก์ชันความน่าจะเป็นของวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิง

รูปที่ 2.5 แสดงลักษณะเส้นกราฟของ g_i สองเส้นโดยกราฟที่แสดงโดยเส้นทึบคือเส้นกราฟของ g_i ที่ถูกกำหนดค่า Tg ให้เท่ากับ 10 ส่วนเส้นกราฟที่แสดงโดยใช้เส้นประ คือเส้นกราฟของ g_i ที่ถูกกำหนดค่า Tg ให้เท่ากับ 0.1 จากลักษณะเส้นกราฟทั้งสองประกอบกับสมการที่ 2.1 และสมการที่ 2.2 พบว่า

- เมื่อเริ่มต้นการทำงาน (สภาวะอุณหภูมิย่อยในการควบคุมการสร้างพารามิเตอร์ (Tg) สูง) วิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิงมีโอกาสกำหนดค่าให้กับ $p_{i(k+1)}$ ได้เป็นค่าในช่วงกว้างเต็มพื้นที่จากค่าที่มากที่สุดไปยังค่าที่น้อยที่สุดของ p_i เนื่องจากเส้นกราฟของฟังก์ชันความน่าจะเป็นยังให้ค่าที่เกือบเท่ากันในทุกๆ ค่าของโดเมน กล่าวคือค่าของ q_i ($-1 \le q_i \le 1$) มีโอกาสเกิดขึ้นได้เท่าๆ กัน ทำให้ $p_{i(k+1)}$ มีโอกาสขยับออกไปห่าง p_{ik} ได้เต็มช่วงของตนเอง

- เมื่อทำงานไปได้ระยะหนึ่งจนอุณหภูมิย่อยในการควบคุมการสร้างพารามิเตอร์ ลดต่ำลงเหลือค่าน้อย วิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิงมีโอกาสสูงในการสุ่มกำหนดค่าให้กับ $p_{i(k+1)}$ ได้ใกล้เคียงกับค่า p_{ik} เนื่องจาก ความน่าจะเป็นที่จะสุ่มได้ q_i ที่มีค่าใกล้เคียงกับ 0 มีโอกาสสูงกว่าค่าอื่น

2.1.6 **ซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง** (Simulated Annealing with Local Optimization: SALO) [8]

วิธีการนี้เป็นการนำเอาวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิงมาเพิ่มเติมคุณสมบัติการหาค่าเหมาะสมเฉพาะแห่ง (Local Optimization)

เนื่องจากวิธีซีมูเล็ทเต็ดแอนนิลลิงมีข้อเด่นในเรื่องความสามารถในการกระโดดออกจากภาวะที่ดีที่สุดเฉพาะแห่งได้ แต่มีข้อด้อยในการปรับละเอียด (Fine Tune) เช่นเดียวกับการแก้ปัญหาค่าเหมาะสมเชิงวิธีจัดหมู่โดยสุ่มวิธีอื่น กล่าวคือเมื่อสถานะใดๆ ถูกสร้างขึ้นมาโดยฟังก์ชัน ***generate_state*** และมีความใกล้เคียงกับภาวะการณ์ดีที่สุด ที่สามารถใช้เป็นคำตอบของปัญหาค่าเหมาะสมเชิงวิธีจัดหมู่ที่สนใจได้ สถานะถัดจากนั้นที่เกิดขึ้นในรอบถัดไปอาจกระโดดหนีไปไกลจากสถานะที่ควรจะเป็นคำตอบนั้นๆ ได้ ทำให้การที่จะกลับมายังสถานะที่ควรจะเป็นคำตอบนี้ ต้องใช้การเรียกฟังก์ชัน ***generate_state*** อีกหลายครั้งกล่าวคือใช้จำนวนรอบในการทำงานมากนั่นเอง

การหาค่าเหมาะสมเฉพาะแห่งมีข้อดีในเรื่องของการทำการปรับละเอียด แต่มีข้อเสียในเรื่องการยึดติดกับค่าภาวะการณ์ดีที่สุดเฉพาะแห่ง ดังนั้นการนำเอาทั้ง 2 วิธีการมารวมกันจึงเป็นการลบข้อด้อยของกันและกัน จำนวนรอบในการทำงานของวิธีซีมูเล็ทเต็ดแอนนิลลิงก็จะน้อยกว่าการทำแบบปกติ

กำหนดให้ ***local_optimize*** เป็นฟังก์ชันในการหาค่าเหมาะสมเฉพาะแห่งเมื่อนำมาใช้ในวิธีซีมูเล็ทเต็ดแอนนิลลิง ฟังก์ชันนี้จะถูกเรียกถัดจากฟังก์ชัน ***generate_state*** ดังในรูปที่ 2.6 และรายละเอียดของ ***local_optimize*** ที่มีเป้าหมายในการหาค่าต่ำสุดจะแสดงไว้ในรูปที่ 2.7

จากรูปที่ 2.7 การหาค่าเหมาะสมเฉพาะแห่งสร้างจากแนวความคิดเรื่องการใช้เวกเตอร์ (Vector) ตัวหนึ่งมาบวกกับสถานะปัจจุบัน ทำให้ได้สถานะใหม่ ซึ่งสถานะปัจจุบันสามารถถูกมองเป็นจุดใดๆ ในปริภูมิ (Space) ขนาด n มิติได้ โดยที่

- ถ้าค่าจากฟังก์ชันการประเมินที่มีสถานะใหม่นี้เป็นอินพุท ดีกว่าค่าจากฟังก์ชันการประเมินที่ได้จากสถานะเดิม ก็จะถือว่าทิศทางของเวกเตอร์ดังกล่าวมีโอกาสที่จะชี้ไปยัง

สถานะที่ดีกว่าขึ้นไปได้เรื่อยๆ ดังนั้นจึงทำการเพิ่มขนาดของเวกเตอร์ตัวเดิม จากนั้นก็ทำการบวกซ้ำเข้าไปในสถานะปัจจุบันอีกทีเพื่อหาสถานะถัดไป

- ในกรณีที่สถานะใหม่ที่ได้จากเวกเตอร์ดังกล่าว ไม่ได้เป็นสถานะที่ดีกว่าปัจจุบัน ก็แสดงว่าสุ่มมาผิดทาง เวกเตอร์จะถูกลดขนาดลงจากนั้นก็จะทำการสุ่มเปลี่ยนทิศทางใหม่โดยให้เวกเตอร์คงขนาด (ที่ถูกลดไปแล้ว) เดิมไว้

```
BEGIN

  select an initial state x
  select an initial temperature T > 0

  REPEAT

    REPEAT

      temp_y := generate_state(x)
      y := local_optimize(temp_y)
      d := cost_funtion(y) – cost_funtion(x)

      IF d < 0 THEN
        x : = y
      ELSE
      IF rand (0, l) < exp (- d / T) THEN
        x : = y
      ENDIF

    UNTIL is_thermal_equilibrium() = True

    T : = reduce_temperature(T)

  UNTIL is_stable() = True

END
```

รูปที่ 2.6 รหัสเทียมของซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
local_optimize(x)
BEGIN

   initialize_vector(V)
   U := 0

   WHILE |V| ≤ THESHOLD

      WHILE cost_funtion(x + V) > cost_funtion(x) and iter < MAXITER

        V := random_vector(V)
        iter := iter + 1

       ENDWHILE

       IF cost_funtion(x + V) > cost_funtion(x) THEN
         V := V / 2
       ELSE
       IF iter = 0 THEN
         x := x + V
         U := U + V
         V := 2U
       ELSE
       IF cost_funtion(x + U + V) < cost_funtion(x)  THEN
          x := x + U + V
         U := U + V
         V := 2U
       ELSE
         x := x + V
         U := V
         V := 2U
       ENDIF

    ENDWHILE

local_optimize :=  x

END
```

รูปที่ 2.7 รหัสเทียมของฟังก์ชันการหาค่าเหมาะสมเฉพาะแห่ง

โดยที่

- ***x:*** เป็นสถานะใดๆ ที่จะถูกนำมาหาค่าเหมาะสมเฉพาะแห่ง
- ***V:*** เป็นเวกเตอร์ที่เอาไว้ใช้ในการบวกกับสถานะเดิมกลายเป็นสถานะใหม่
- ***U:*** เป็นเวกเตอร์ที่เอาไว้ใช้เก็บเวกเตอร์บวกสะสมของ ***V***

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- ***initialize_vector*:** เป็นฟังก์ชันการกำหนดค่าเริ่มต้นทั้งขนาดและทิศทางให้กับเวกเตอร์
- **THESHOLD:** เป็นค่าคงที่ ที่ใช้ระบุขนาดของ V ที่น้อยที่สุดที่ยอมรับได้ในการหาค่าเหมาะสมเฉพาะแห่ง
- **MAXITER:** เป็นค่าคงที่ ที่ใช้กำหนดจำนวนรอบที่มากที่สุดในการสุ่มหาทิศทางของ V ที่จะทำให้เกิดสถานะใหม่ที่ดีกว่าสถานะเดิมที่มีอยู่

โดยสรุปแล้ววิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งคือวิธีอะแดพทีฟ-ซีมูเล็ทเต็ดแอนนิลลิงที่ถูกปรับปรุงประสิทธิภาพให้มีวงรอบการทำงานน้อยลงโดยความช่วยเหลือของการหาเหมาะสมเฉพาะแห่ง ดังนี้จึงทำให้เวลาที่ใช้ในการทำงานสั้นลงด้วย แต่วิธีนี้ก็มีข้อจำกัดเช่นเดียวกับวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิงคือข้อมูลที่จะนำมาใช้ในการหาค่าเหมาะสมจะต้องมีช่วงจำกัดที่สามารถระบุได้

2.1.7 คุณสมบัติการเข้าคู่กัน (Coupling) และการเกาะกลุ่มกัน (Cohesion) ของคลาส [9]

การเข้าคู่กันของคลาส หมายถึงดีกรี (Degree) ของการขึ้นต่อกันระหว่างคลาสต่าง ๆ ในโมเดลการออกแบบใด ๆ

การเกาะกลุ่มของคลาส หมายถึงดีกรี (Degree) ของความสัมพันธ์ระหว่างกันของสมาชิกภายในคลาส นั่นคือการที่เมทธอดและแอตทริบิวต์ของคลาสใด ๆ มีความสัมพันธ์กัน

โมเดลการออกแบบที่ดีควรมีคลาสที่มีการเข้าคู่กันระหว่างคลาสต่ำและมีการเกาะกลุ่มกันภายในคลาสสูง

2.1.8 ร่องรอยที่ไม่ดี (Bad-Smell) [10]

ร่องรอยที่ไม่ดี หมายถึงลักษณะของการออกแบบหรือเขียนรหัสต้นทางที่ไม่ดี ซึ่งจะส่งผลให้การบำรุงรักษาซอฟต์แวร์ตัวนั้นเป็นไปได้ยาก ร่องรอยที่ไม่ดีสำหรับโปรแกรมเชิงวัตถุที่ถูกกำหนดไว้มีอยู่หลายประเภท แต่เฉพาะที่เกี่ยวข้องกับงานวิจัยนี้มีอยู่สองแบบด้วยกันคือ

- ฟีเจอร์เอนวี (Feature Envy) คือ ลักษณะที่เมทธอดมีการเรียกใช้คุณสมบัติในคลาสอื่นมากกว่าคลาสเจ้าของเมทธอดเอง
- มิดเดิลแมน (Middle Man) คือ คลาสที่ทำหน้าที่เป็นตัวกลางการติดต่อระหว่างออบเจ็กต์โดยที่ตนเองไม่มีฟังก์ชันงานในตัวเองมากนัก

การกำจัดร่องรอยที่ไม่ดีสามารถทำได้โดยการเปลี่ยนแปลงโครงสร้างของโมเดลการออกแบบหรือรหัสต้นทางให้ได้คุณภาพของโมเดลการออกแบบหรือรหัสต้นทางที่ดีขึ้น โดยที่กระบวนการเปลี่ยนแปลงนี้ นิยมเรียกว่าการทำรีแฟคทอริง (Refactoring) โดยมีเงื่อนไขว่าการเปลี่ยนแปลงนี้จะต้องไม่ทำให้พฤติกรรมของระบบมีการเปลี่ยนแปลงเมื่อมองจากภายนอก

2.1.9 **ระบบพิกัดไฮเปอร์สเฟียริคอล** (Hyperspherical Coordinates) [11]

คือระบบพิกัดทรงกลมใน n ในมิติ ซึ่งแต่ละพิกัดสามารถระบุได้โดยใช้ ระยะห่างจากพิกัดนั้นไปยังศูนย์กลางของระบบพิกัดประกอบกันกับมุมที่พิกัดนั้นทำกับระนาบของแกนต่างๆ ในพิกัดทรงกลมนั้น ตัวอย่างการบอกพิกัดในระบบทรงกลม 3 มิติ คือ (r,θ_1,θ_2) โดยที่

- r : เป็นระยะทางจากพิกัดนั้นไปยังศูนย์กลางของระบบพิกัด หรือรัศมีของทรงกลมใน 3 มิติ
- θ_1 : เป็นมุมที่เส้นตรงที่ออกจากจุดศูนย์กลางของระบบพิกัดไปยังพิกัดที่สนใจ ทำกับแกน Z โดยที่มีค่าได้ตั้งแต่ 0 ถึง π
- θ_2 : เป็นมุมที่โปรเจคชันไปยังระนาบ XY ของเส้นตรงที่ออกจากจุดศูนย์กลางของระบบพิกัดไปยังพิกัดที่สนใจ ทำกับแกน X 0 ถึง 2π

โดยที่เราจะมี θ ได้ n-1 ตัวในระบบพิกัดทรงกลมใน n มิติ

การแปลงระบบพิกัดทรงกลม n มิตินี้ให้เป็นระบบพิกัดคาร์ทีเซียน (Cartesian Coordinates) n มิติ จะสามารถอธิบายเป็นลักษณะทั่วๆ ไปดังรูปที่ 2.8 นี้

$$
\begin{aligned}
x_1 &= r\cos(\theta_1)\\
x_2 &= r\sin(\theta_1)\cos(\theta_2)\\
x_3 &= r\sin(\theta_1)\sin(\theta_2)\cos(\theta_3)\\
&\ldots\\
x_{n-1} &= r\sin(\theta_1)\ldots\sin(\theta_{n-2})\cos(\theta_{n-1})\\
x_n &= r\sin(\theta_1)\ldots\sin(\theta_{n-2})\sin(\theta_{n-1})
\end{aligned}
$$

รูปที่ 2.8 รูปแบบการหาพิกัดคาร์ทีเซียนจากพิกัดไฮเปอร์สเฟียริคอล

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัย "A Stochastic Approach to Automated Design Improvement" [12]

เป็นงานวิจัยที่นำเสนอการใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงแบบปกติในการทำรีแฟคทอริง (Refactoring) ที่รหัสต้นทางที่นิยามไว้ใน [10] โดยที่วิธีการรีแฟคทอริงหนึ่งๆ จะใช้มาตรวัดที่ไม่เหมือนกัน ในงานวิจัยได้เสนอตัวอย่างเครื่องมือในการทำรีแฟคทอริง 2 แบบคือวิธี pullUpMethod และ วิธี pullDownMethod อัตโนมัติ โดยใช้มาตรวัดที่กำหนด โดยงานนี้ทางผู้วิจัยคาดหวังจะทำกับรีแฟคทอริงวิธีอื่นเพิ่มมากขึ้นต่อไปในอนาคต

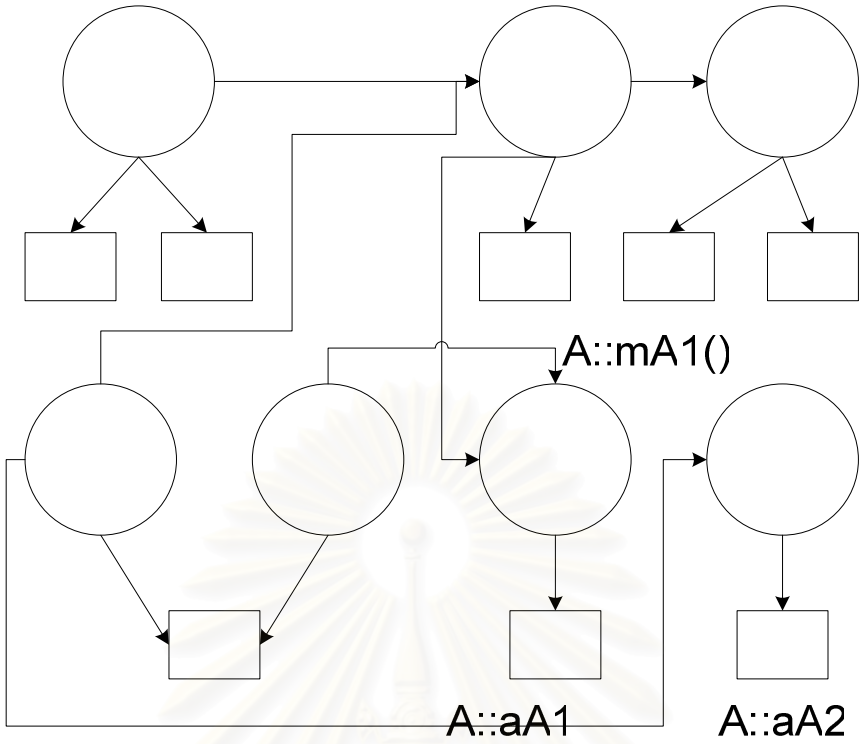
ผู้จัดทำวิทยานิพนธ์ได้นำแนวคิดการปรับปรุงคุณภาพของซอฟท์แวร์ด้วยวิธีซีมูเล็ทเต็ด-แอนนิลลิงมาประยุกต์ใช้

2.2.2 งานวิจัย "Genetic Algorithm Based Restructuring of Object-Oriented Designs Using Metrics" [9]

เป็นงานวิจัยที่นำเสนอการทำการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบเชิงวัตถุด้วยวิธีการเจนเนติก (Genetic Algorithm) โดยมีการกำหนดรูปแบบของโมเดลการออกแบบ (Design Model) ที่จะใช้ในการทำการปรับปรุงโดยเอื้อต่อการหาค่าเหมาะสมของ มาตรวัดการเข้าคู่กัน (Coupling) และ มาตรวัดการเกาะกลุ่มกัน (Cohesion) ที่ดีที่สุดจากวิธีเจนเนติกข้างต้น โดยงานวิจัยนี้สามารถทำการเปลี่ยนแปลงคลาสต่างๆ ที่มีอยู่ในโมเดลการออกแบบ ให้มีคุณภาพที่ดีขึ้นได้

ในงานวิจัยนี้ได้ทำการกำหนดคอลยูสกราฟ(Call-Use Graph: CUG) และ คลาสแอสโซซิ-เอชันกราฟ (Class-Association Graph: CAG) ขึ้นมาเพื่อใช้อธิบายโมเดลการออกแบบที่มีอยู่ และกราฟทั้งสองนี้เป็นโครงสร้างที่เหมาะสมกับวิธีเจนเนติก ตัวอย่างเช่น

จากรูปที่ 2.9 โหนดวงกลมจะใช้แทนเมทธอด โหนดสี่เหลี่ยมจะใช้แทนแอตทริบิวต์ ลูกศรระหว่างโหนดจะถือเป็นการเรียกกันระหว่างเมทธอดหรือการที่เมทธอดไปใช้แอตทริบิวต์ใดๆ การที่กราฟนี้มีจำนวนโหนดทั้งหมดเป็นอย่างไรนั้น สามารถดูได้จากเมทธอดและแอตทริบิวต์ของคลาสต่างๆ ในโมเดลการออกแบบหรือรหัสต้นทาง แต่ข้อมูลการเรียกกันระหว่างเมทธอดและการที่เมทธอดไปใช้แอตทริบิวต์ใดๆ นั้นต้องดูจากรหัสต้นทางเพียงอย่างเดียว



รูปที่ 2.9 ตัวอย่างบางส่วนของคอลยูสกราฟที่สอดคล้องกับโมเดลการออกแบบในรูปที่ 2.1

กราฟการเรียกและการใช้สามารถอธิบายได้อีกรูปแบบคือเป็นอาเรย์ (Array) ของตัวเลข ซึ่งเป็นลักษณะที่เหมาะสมกับการนำไปหาค่าเหมาะสมโดยวิธีเจนเนติก ซึ่งเป็นรูปแบบหนึ่งของวิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่ แต่ละช่องของอาเรย์จะบอกข้อมูลของเมทธอดและแอตทริบิวต์ใดๆ ในโมเดลการออกแบบ เช่นกำหนดให้เมื่อนำเมทธอดและแอตทริบิวต์ทุกๆ ตัวในโมเดลการออกแบบมาเรียงกันในลำดับดังนี้

B::mB1() B::mB2()

mA1, mB1, mB2, mC1, mC2, mD1, mD2, aA1, aA2, aB1, aC1, aC2, aC3, aD1, aD2

ดังนั้นอาเรย์ที่ใช้อธิบายกราฟในรูปที่ 2.9 สามารถเขียนได้เป็นดังรูปที่ 2.10

B::aB1

1	2	2	3	3	4	4	1	1	2	3	3	3	4

รูปที่ 2.10 ตัวอย่างอาเรย์ ที่สอดคล้องกับกราฟในรูปที่ 2.9

โดยให้คลาสทั้งหมดในระบบคือ A, B, C และ D ถูกแทนเป็นตัวเลขตั้งแต่ 1 – 4 ตามลำดับ ดังนั้นอาเรย์ที่ได้นี้จะสามารถใช้อธิบายว่าเมทธอดและแอตทริบิวต์แต่ละตัวเป็นสมาชิกของคลาสใดบ้าง

นอกจากนี้งานวิจัยนี้ยังนำเสนอมาตรวัดใหม่ 3 ตัวที่เอาไว้วัดปริมาณการเข้าคู่กันและการเกาะกลุ่มกันของคลาสต่างๆ ที่มีในโมเดลการออกแบบ คือ COH (Cohesion), COP_{IH} (Inheritance Coupling) และ COP_{IA} (Interaction Coupling)

นิยามของ *COH(D)* เป็นดังสมการที่ 2.5

$$COH(D) = \sum_{C_i \in D} COH(C_i) \qquad \text{สมการที่ 2.5}$$

โดยที่

- D คือโมเดลการออกแบบ
- C_i คือคลาสแต่ละตัวภายในโมเดลการออกแบบ
- $COH(C_i)$ คือมาตรวัดที่ใช้วัดการเกาะกลุ่มกันของเมทธอดภายในคลาสคลาสหนึ่ง

นิยามของ *COH(C)* เป็นดังสมการที่ 2.6

$$COH(C_i) = \frac{SBC(C_i, C_i)}{|C_i| * |C_i|} \qquad \text{สมการที่ 2.6}$$

โดยที่

- $|C|$ มีค่าเท่ากับจำนวนเมทธอดของคลาส C
- *SBC* (Similarity between Class) คือมาตรวัดที่ใช้สำหรับวัดความเข้าคู่กันของเมทธอดภายในคลาส โดยพิจารณาจากความคล้ายคลึงกันระหว่างเมทธอดสองเมทธอดใดๆ ภายในคลาส

ขั้นตอนการคำนวณความคล้ายคลึงกันระหว่างเมทธอด (Similarity between Methods) หรือการคำนวณมาตรวัด *SBM* จะต้องมีการสร้าง Communication Matrix between Methods *(CMM)* ซึ่งเป็นเมตริกซ์สมมาตรที่มีขนาดในแต่ละมิติเท่ากับจำนวนเมทธอดทั้งหมดของโมเดลการออกแบบที่พิจารณา โดยที่แต่ละสมาชิกของ *CMM* สามารถอธิบายได้ดังนี้ *CMM (i, j)* คือค่าความสัมพันธ์ของเมทธอดในแถวที่ i และคอลัมน์ที่ j ของเมตริกซ์ โดยการหาค่าน้ำหนักของการ

เรียกใช้กันของเมทธอดแบ่งเป็น 2 กรณีคือ ในกรณีที่เมทธอดทั้งสองมีการเรียกใช้กัน ค่าน้ำหนักจะมีค่าเป็นจำนวนพารามิเตอร์ของเมทธอดที่ถูกเรียกใช้บวกหนึ่ง ในกรณีที่เมทธอดทั้งสองไม่มีการเรียกใช้กัน ค่าน้ำหนักจะเท่ากับ 0 จากนั้นนำค่าที่ได้มาบวกเพิ่มไปด้วยจำนวนแอตทริบิวต์ใดๆ ที่เมทธอดทั้งสองใช้ร่วมกัน

หลังจากได้ *CMM* แล้วจะสามารถนำมาใช้ในการคำนวณ *SBM* และ *SBC* ได้ดังนี้

$$SBM(m_1, m_2) = \frac{\sum_{k=1}^{M} CMM(i,k) * CMM(j,k)}{\sqrt{\sum_{k=1}^{M} CMM(i,k)^2 * \sum_{k=1}^{M} CMM(j,k)^2}} \quad \text{สมการที่ 2.7}$$

โดยที่

- M คือจำนวนเมทธอดในโมเดลการออกแบบ

และนิยามของ *SBC* เป็นดังนี้

$$SBC(C_i, C_j) = \sum_{p=1}^{|C_i|} \sum_{q=1}^{|C_j|} SBM(m_{ip}, m_{jq}) \quad \text{สมการที่ 2.8}$$

นิยามของ $COP_{IA}(D)$ เป็นดังนี้

$$COP_{IA}(D) = \sum_{C_i \in D} COP_{IA}(C_i) \quad \text{สมการที่ 2.9}$$

โดยที่

- D คือโมเดลการออกแบบ
- $COP_{IA}(C)$ คือมาตรวัดที่ใช้วัดการเข้าคู่กันของคลาสในโมเดลการออกแบบ

นิยามของ $COP_{IA}(C)$ เป็นดังนี้

$$COP_{IA}(C_i) = \sum_{C_j \in Int(C_i)} SBC(C_i, C_j) \quad \text{สมการที่ 2.10}$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

โดยที่

- *Int(C_i)* หมายถึงเซ็ตของคลาสที่มีสมาชิกอย่างน้อยหนึ่งตัวถูกเรียกใช้จากคลาส C_i

ผู้จัดทำวิทยานิพนธ์ได้นำเอาคอลยูสกราฟมาประยุกต์ใช้ในการสร้างโครงร่างเพื่อทำการเปลี่ยนแปลงแผนภาพคลาสและแผนภาพซีเควนซ์ต่างๆ ในโมเดลการออกแบบบเพื่อปรับปรุงคุณภาพของโมเดลการออกแบบนั้นให้ดีขึ้น โดยมี มาตรวัด COH และ COP_{IA} เป็นปลั๊กอินการปรับปรุงโมเดลการออกแบบโดยปริยาย (Default)

2.2.3 งานวิจัย "การตรวจจับข้อบกพร่องในขั้นตอนการออกแบบเพื่อปรับปรุงความสามารถในการเปลี่ยนแปลงซอฟต์แวร์เชิงวัตถุ" [13]

งานวิจัยนี้นำเสนอวิธีการตรวจจับข้อบกพร่องของโมเดลการออกแบบ (Design Flaw) และร่องรอยที่ไม่ดีของโมเดลการออกแบบ 6 ประเภทได้แก่ ดาต้าคลาส (Data Class), ฟีเจอร์เอน-วี (Feature Envy), เมสเสจเชน (Message Chains), มิดเดิลแมน (Middle Man), ก๊อดคลาส (God Class), และ สวิตช์เสตทเมนท์ (Switch Statements) โดยพิจารณาว่ามีข้อบกพร่องและร่องรอยที่ไม่ดีประเภทใดบ้างที่ส่งผลกระทบต่อความสามารถในการเปลี่ยนแปลงซอฟต์แวร์เชิงวัตถุ ในงานวิจัยมีการออกแบบกลยุทธ์ (Strategy) การตรวจจับข้อบกพร่องและร่องรอยที่ไม่ดีด้วยมาตรวัดเชิงวัตถุสำหรับโมเดลการออกแบบ โดยทำการตรวจจับข้อบกพร่องที่แผนภาพคลาสและแผนภาพซีเควนซ์ นอกจากนั้นยังนำเสนอวิธีการรีแฟคทอริงเพื่อแก้ไขข้อบกพร่องและร่องรอยที่ไม่ดีแต่ละประเภท พร้อมทั้งนำเสนอวิธีการหาช่วงของมาตรวัดที่บอกถึงข้อบกพร่องและร่องรอยที่ไม่ดี โดยประยุกต์ใช้วิธีซีมูเลทเต็ดแอนนิลลิงเพื่อหาค่าช่วงที่เหมาะสมที่สุด

ในการพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุได้นำเอากลยุทธ์การตรวจจับร่องรอยที่ไม่ดีของโมเดลการออกแบบที่เป็นร่องรอยที่ไม่ดีจากงานวิจัยนี้มาประยุกต์ใช้ 2 แบบคือ กลยุทธ์การตรวจจับฟีเจอร์เอนวี และ กลยุทธ์การตรวจจับมิดเดิลแมน เพื่อเป็นอีก 2 ตัวอย่างของปลั๊กอินการปรับปรุงโมเดลการออกแบบ

กลยุทธ์การตรวจจับเมทธอดที่มีร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีถูกนิยามเป็นดังอสมการที่ 2.1

$$NSMsg(M) / NMsg(M) < 0.31 \qquad \text{อสมการที่ 2.1}$$

โดยที่

- M คือ เมทธอดใดๆ ในโมเดลการออกแบบ และ M จะมีร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีก็ต่อเมื่อทำให้ อสมการนี้เป็นจริง

- *NSMsg* (Number of Self Message) คือมาตรวัด ที่วัดจำนวนเมทธอดของคลาสตัวเองที่ถูกเรียกโดยเมทธอดที่พิจารณา โดยจะวัดจากเมสเสจในแผนภาพซีเควนซ์ โดยถ้าเป็นเมสเสจของเมทธอดเดียวกันจะนับเพียงหนึ่งครั้ง

- *NMsg* (Number of Message) คือมาตรวัด ที่วัดจำนวนเมทธอดของคลาสอื่นที่ถูกเรียกโดยเมทธอดที่พิจารณา โดยจะวัดจากเมสเสจในแผนภาพซีเควนซ์ โดยถ้าเป็นเมสเสจของเมทธอดเดียวกันจะนับเพียงหนึ่งครั้ง

อสมการที่ 2.1 มีความหมายว่าถ้าอัตราส่วนระหว่าง *NSMsg* ต่อ *NMsg* ของเมทธอดใดๆ มีค่าน้อยกว่า 0.31 แสดงว่าเมทธอดนั้นมีร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวี

กลยุทธ์การตรวจจับคลาสที่มีร่องรอยที่ไม่ดีแบบมิดเดิลแมนถูกนิยามดังอสมการที่ 2.2

$$WOD(C) > 0.43 \qquad \text{อสมการที่ 2.2}$$

โดยที่

- C คือ คลาสใดๆ ในโมเดลการออกแบบ และ C จะมีร่องรอยที่ไม่ดีแบบมิดเดิลแมนก็ต่อเมื่อทำให้ อสมการนี้เป็นจริง

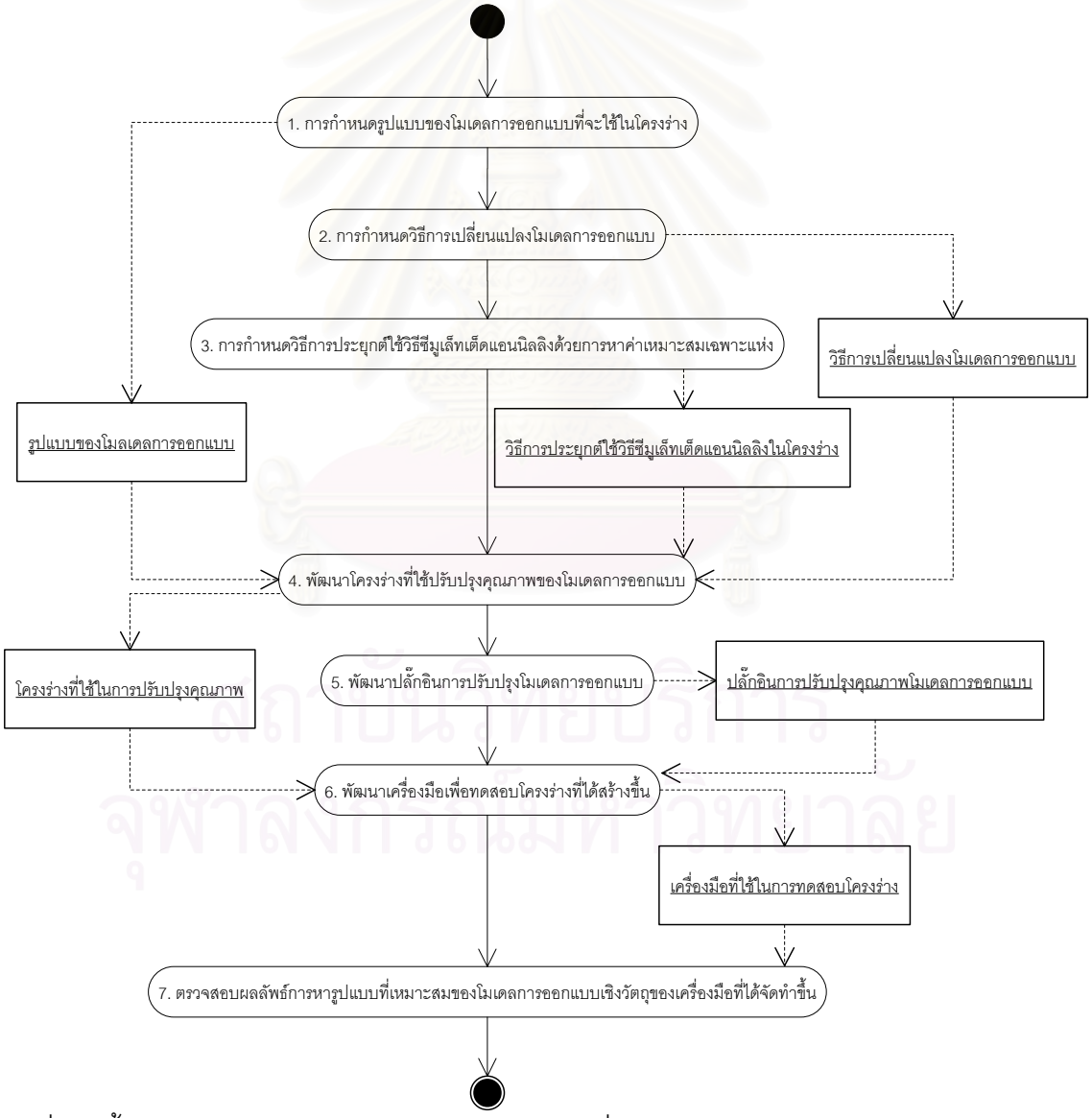
- *WOD* (Weight of Delegation) คือมาตรวัด ที่วัดอัตราส่วนระหว่างจำนวนเมทธอดของคลาสที่พิจารณาที่ทำหน้าที่มอบหมายงานให้กับคลาสอื่น ต่อจำนวนเมทธอดทั้งหมดของคลาสที่ไม่รวมคอนสตรัคเตอร์ ซึ่งการนับจำนวนเมทธอดของคลาสว่าเป็นเมทธอดที่ทำหน้าที่มอบหมายงานให้กับคลาสอื่น จะพิจารณาจากเส้นของเมสเสจในแผนภาพซีเควนซ์ว่าเมื่อเมสเสจของเมทธอดของออบเจ็กต์ถูกเรียกแล้ว ได้มีการเรียกใช้งานเมทธอดของคลาสอื่นต่อไปอีกหนึ่งเมสเสจเท่านั้นใช่หรือไม่ ถ้าใช่ แสดงว่าเมสเสจนั้นไม่มีหน้าที่อื่นอีกนอกจากการมอบหมายงานให้กับคลาสอื่น

อสมการที่ 2.2 มีความหมายว่าค่า *WOD* ของคลาสใดๆ มีค่ามากกว่า 0.43 แสดงว่าคลาสนั้นมีร่องรอยที่ไม่ดีแบบมิดเดิลแมน

บทที่ 3

การพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

เนื้อหาของบทนี้จะเป็นการอธิบายถึงรายละเอียดของการพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุรวมไปถึงเทคนิคต่างๆ ที่ถูกนำมาใช้ โดยจะเริ่มจากขั้นตอนการพัฒนาโครงร่างตามด้วยลักษณะของโมเดลการออกแบบรวมไปถึงข้อกำหนดของโมเดลการออกแบบ และสุดท้ายจะเป็นการอธิบายถึงการประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงภายในโครงร่าง



รูปที่ 3.1 ขั้นตอนการพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

3.1 ขั้นตอนการพัฒนาโครงร่าง

จากรูปที่ 3.1 ขั้นตอนในการทำวิจัยเพื่อพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ ประกอบไปด้วย 7 ขั้นตอน โดยที่ผลลัพธ์ของขั้นตอนทั้งหมดจะถูกอธิบายในหัวข้อถัดๆ ไปในบทนี้

3.1.1 การกำหนดรูปแบบของโมเดลการออกแบบที่จะใช้ในโครงร่าง

ขั้นตอนนี้เป็นการกำหนดรูปแบบของโมเดลการออกแบบที่จะใช้เพื่อการหารูปแบบที่เหมาะสม โดยคุณสมบัติที่ควรจะมีของรูปแบบโมเดลการออกแบบที่จะนำมาใช้ มีดังนี้

- เป็นรูปแบบของโมเดลการออกแบบที่ใช้กันอย่างแพร่หลายในปัจจุบัน เพราะจะทำให้สามารถนำโครงร่างไปใช้ได้อย่างกว้างขวาง
- เป็นรูปแบบที่นำไปประยุกต์ใช้กับวิธีซีมูเล็ทเต็ดแอนนิลลิงได้ เนื่องจากงานวิจัยนี้จะทำการเปลี่ยนแปลงโมเดลการออกแบบโดยใช้วิธีซีมูเล็ทเต็ดแอนนิลลิง

เนื่องจากปัจจุบันการอธิบายโมเดลการออกแบบด้วยภาษายูเอ็มแอลเป็นรูปแบบมาตรฐานที่นิยมใช้กันโดยทั่วไป ดังนั้นผู้จัดทำวิทยานิพนธ์จึงได้เลือกใช้รูปแบบของโมเดลยูเอ็มแอลเป็นอินพุทของโครงร่างที่จะพัฒนาขึ้น แต่ด้วยรูปแบบของโมเดลยูเอ็มแอลเองยังไม่เพียงพอกับการนำไปใช้กับวิธีซีมูเล็ทเต็ดแอนนิลลิงซึ่งต้องการอินพุทในรูปแบบของอาเรย์ข้อมูล ดังนั้นคอลยูสกราฟจึงเป็นโครงสร้างข้อมูลอีกอย่างหนึ่งที่ถูกนำเอามาใช้ในการเชื่อมต่อเพื่อทำให้วิธีซีมูเล็ทเต็ดแอนนิลลิงสามารถทำงานกับโมเดลยูเอ็มแอลเพื่อปรับปรุงคุณภาพได้

ในส่วนของโมเดลยูเอ็มแอลจะใช้เฉพาะแผนภาพคลาสและแผนภาพซีเควนซ์เพราะมีรายละเอียดอยู่ในระดับที่ค่อนข้างชัดเจนใกล้เคียงกับรหัสต้นทาง ทำให้ง่ายในการตีความเพื่อเปลี่ยนแปลงรูปแบบ

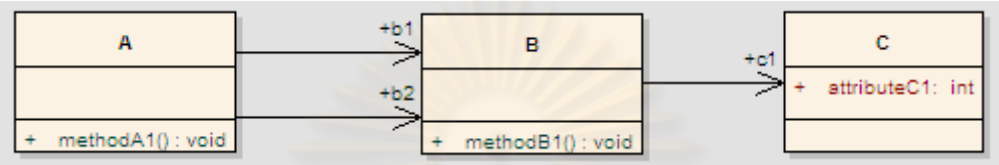
ซึ่งแผนภาพยูเอ็มแอลที่จะในมาใช้ในโครงร่างมีข้อกำหนดดังนี้

3.1.1.1 ข้อกำหนดของแผนภาพคลาสที่จะถูกใช้ในโครงร่าง

1. ที่ปลายของความสัมพันธ์แบบแอสโซซิเอชันทุกๆ ความสัมพันธ์ด้านที่มีลูกศรกำกับจะต้องถูกกำกับด้วยชื่อของโรลปลายทางเสมอ รูปที่ 3.2 แสดงตัวอย่างของข้อกำหนดนี้ จากรูปถ้าพิจารณาความสัมพันธ์ระหว่างคลาส A และ B จะเห็นว่าความสัมพันธ์แบบแอสโซซิเอชันจากคลาส A ไปยังคลาส B สองความสัมพันธ์ ที่ปลายด้านที่มีลูกศรระบุชื่อของโรลไว้ทั้งคู่ โดยที่สามารถตีความได้เป็น ออบเจ็กต์ชื่อ b1 และ b2 เป็นออบเจ็กต์ของคลาส B

และต่างก็เป็นแอตทริบิวต์ของคลาส A ทั้งคู่ การทำดังนี้ทำให้ข้อมูลการระบุแอตทริบิวต์แบบที่เป็นโรลของคลาสไม่สูญหาย

2. โรลทุกตัวในโมเดลการออกแบบจะต้องมีชื่อไม่ซ้ำกัน เนื่องจากชื่อของโรลจะถูกนำไประบุเป็น บริบทการถูกเรียกใช้ (Called Context) ซึ่งจะถูกอธิบายอีกทีในหัวข้อ 3.1.4.3.2



รูปที่ 3.2 รูปแบบของแผนภาพคลาสที่เป็นไปตามข้อกำหนด

3.1.1.2 ข้อกำหนดของแผนภาพซีเควนซ์ที่จะถูกใช้ในโครงร่าง

1. พฤติกรรมของเมทธอดแต่ละตัวที่กำหนดในแต่ละคลาสจะต้องอธิบายได้ด้วยเมสเสจที่สัมพันธ์กับเมทธอดนั้นๆ เช่น ในรูปที่ 3.3 เมทธอด A::methodA1 มีการอธิบายลำดับการทำงานด้วยเมสเสจว่ามีการเรียกเมทธอด B::methodB1 ของออบเจ็กต์ b1 แล้วจึงไปเรียกใช้เมทธอดเดียวกันของออบเจ็กต์ b2 ขึ้นทำงาน ซึ่งออบเจ็กต์ทั้งคู่ถือเป็นแอตทริบิวต์ของคลาส A จากข้อกำหนดนี้ทำให้ข้อมูลการเรียกใช้กันระหว่างเมทธอดและแอตทริบิวต์ครบถ้วน

2. พฤติกรรมของเมทธอดแต่ละตัวจะต้องถูกอธิบายเพียงหนึ่งครั้งเท่านั้นเช่น ได้มีการอธิบายพฤติกรรมภายในของเมทธอด B::methodB1 ด้วยเมสเสจไปแล้วเมื่อตอนที่ เมทธอด A::methodA1 เรียกใช้เมทธอด B::methodB1 ของออบเจ็กต์ b1 ดังนั้นในลำดับถัดมาเมื่อ เมทธอด A::methodA1 เรียกใช้ เมทธอด B::methodB1 ของออบเจ็กต์ b2 จึงมีเมสเสจอ้างถึงเฉยๆ โดยจะต้องไม่มีการเขียนเมสเสจใดๆ ที่เป็นเมสเสจลูกที่แสดงพฤติกรรมของ เมทธอด B::methodB1 อีก ดังแสดงไว้ในรูปที่ 3.3 เพื่อป้องกันความผิดพลาดที่เกิดจากการอธิบายพฤติกรรมของเมทธอดเดียวกันหลายครั้งแต่อธิบายไม่ตรงกัน

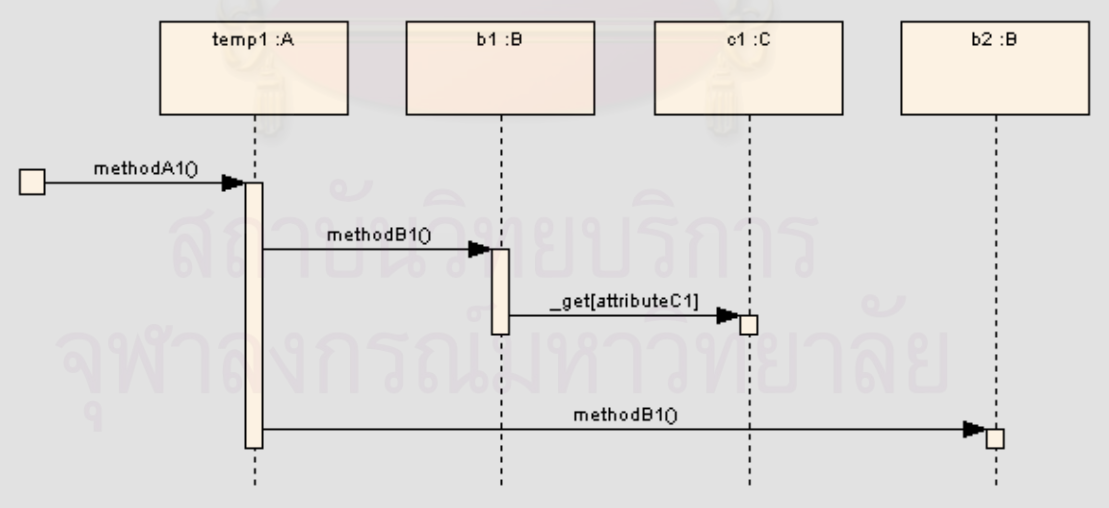
3. แผนภาพซีเควนซ์จะต้องเริ่มต้นด้วยเมสเสจที่ถูกส่งออกจากไดอะแกรมเกท (Diagram Gate) เพียงอันเดียวเสมอ ซึ่งโดยภาพรวมแล้วแผนภาพซีเควนซ์นี้จะเป็นการอธิบายพฤติกรรมของเมทธอดที่สัมพันธ์กับเมสเสจที่ถูกส่งออกจากไดอะแกรมเกทนี้

4. ออบเจ็กต์แรกที่รับเมสเสจมาจากไดอะแกรมเกทจะต้องเป็นออบเจ็กต์ชั่วคราว (Temporary Object) ซึ่งถือเป็นตัวแปรชั่วคราว (Temporary Variable) เท่านั้น กล่าวคือจะต้องไม่ใช่ออบเจ็กต์ที่เป็นแอตทริบิวต์แบบโรลของคลาสใดๆ ในระบบ เนื่องจากแผนภาพซีเควนซ์จะทำหน้าที่อธิบายพฤติกรรมของเมทธอดที่สัมพันธ์กับเมสเสจแรกในแผนภาพดังที่กล่าวไปในข้อ 3 ดังนั้นการระบุออบเจ็กต์ที่ถูกเรียกขึ้นเพื่อให้บริการเมทธอดดังกล่าวจะไม่อยู่ในขอบเขตของการอธิบายพฤติกรรมของเมทธอดนั้น ในที่นี้จึงกำหนดให้เป็นออบเจ็กต์ชั่วคราวเพื่อความสอดคล้องกันของการเขียนแผนภาพซีเควนซ์ทั้งหมดในโมเดลการออกแบบ

5. การเรียกใช้แอตทริบิวต์ใดๆ ของคลาสจะกำหนดโดยเมสเสจที่ถูกตั้งชื่อเป็นพิเศษสองแบบ คือ

   - _get[<*AttributeName*>]
   - _set[<*AttributeName*>]

   เพื่อเป็นการแสดงการเข้าถึงเพื่อขอรับค่าหรือขอเปลี่ยนแปลงค่าของแอตทริบิวต์ใดๆ โดยที่ [<*AttributeName*>] หมายถึงชื่อของแอตทริบิวต์ที่กำลังจะทำงานด้วย ข้อกำหนดนี้เป็นการเพิ่มข้อมูลการเรียกใช้แอตทริบิวต์จากเมทธอดให้กับโมเดลยูเอ็มแอล ดังแสดงไว้ในรูปที่ 3.3



รูปที่ 3.3 รูปแบบของแผนภาพซีเควนซ์ที่สัมพันธ์กับ แผนภาพคลาสในรูปที่ 3.2 ที่เป็นไปตามข้อกำหนด

3.1.2 การกำหนดวิธีการเปลี่ยนแปลงโมเดลการออกแบบ

ในขั้นตอนนี้วิธีการเปลี่ยนแปลงโครงสร้างของโมเดลการออกแบบจะถูกกำหนดขึ้นโดยที่วิธีการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบนี้จะต้องไม่ทำให้พฤติกรรมของโมเดลการออกแบบเปลี่ยนแปลงไป หลังจากถูกเปลี่ยนแปลงแล้ว

วิธีการเปลี่ยนแปลงโมเดลการออกแบบที่เลือกใช้คือการย้ายเมทธอด และแอตทริบิวต์แต่ละตัวไปยังคลาสต่างๆ ที่จะเกิดขึ้นในโมเดลการออกแบบที่จะเกิดขึ้นใหม่ โดยที่ไม่ได้เปลี่ยนแปลงลำดับการถูกเรียกใช้ของเมทธอด และแอตทริบิวต์ต่างๆ เหล่านั้น ด้วยเหตุนี้การเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบวิธีนี้จึงไม่ทำให้พฤติกรรมของโมเดลการออกแบบเปลี่ยนแปลงไป

3.1.3 การกำหนดวิธีการประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งในงานการปรับปรุงคุณภาพของโมเดลการออกแบบ

ขั้นตอนนี้เป็นการกำหนดรายละเอียดของวิธีการประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งที่จะนำมาใช้จริงเนื่องจากวิธีซีมูเล็ทเต็ดแอนนิลลิงที่ถูกอธิบายในหัวข้อที่ 2.1.5 และ 2.1.6 นั้นยังไม่ได้ระบุรายละเอียดที่แน่นอนในระดับที่สามารถนำไปพัฒนาเป็นรหัสต้นทางได้ในทันที รวมไปถึงการกำหนดรายละเอียดของฟังก์ชันที่ถูกกำหนดคุณสมบัติไว้อย่างคร่าวๆ เท่านั้นโดยผู้คิดค้นวิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง วิธีการประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งจะถูกอธิบายโดยละเอียดในหัวข้อ 3.2

3.1.4 การพัฒนาโครงร่างที่ใช้ปรับปรุงคุณภาพของโมเดลการออกแบบ

ขั้นตอนนี้เป็นการออกแบบสถาปัตยกรรมของโครงร่างที่จะพัฒนาขึ้น โดยโครงร่างที่จะถูกพัฒนาควรจะมีความยืดหยุ่นสูงในการเปลี่ยนแปลงส่วนต่างๆ ภายในโครงร่างเพื่อการพัฒนาให้โครงร่างสามารถนำไปประยุกต์ใช้กับการปรับปรุงคุณภาพของโมเดลการออกแบบได้ในสภาพแวดล้อมและข้อจำกัดที่ต่างกัน

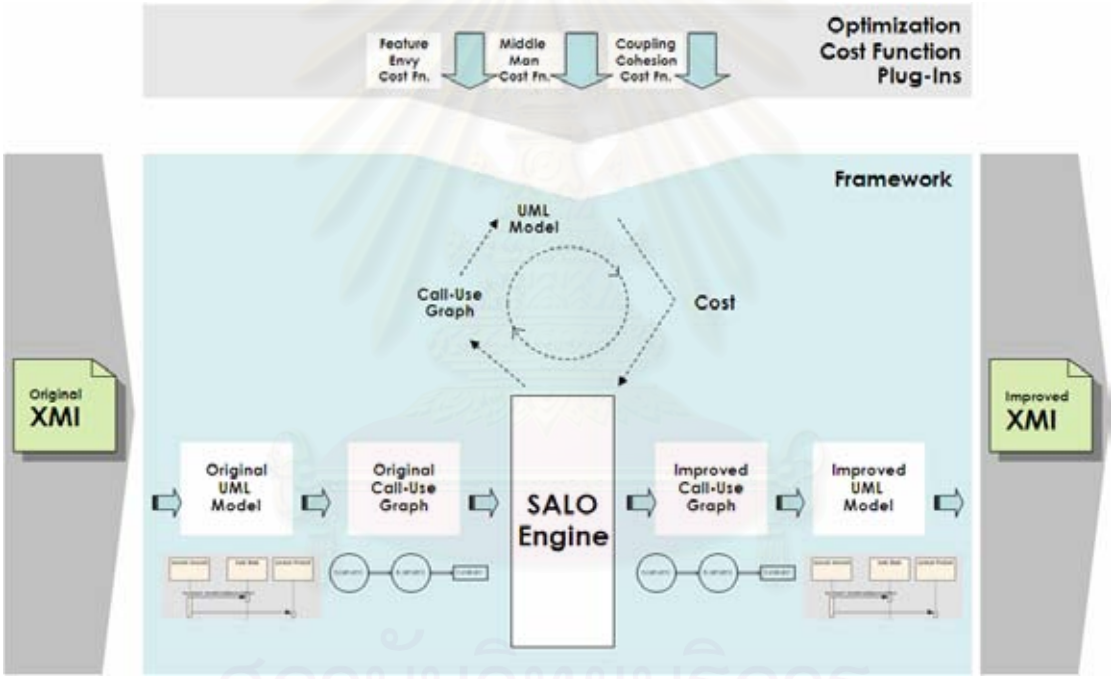
3.1.4.1 ภาพรวมของโครงร่าง

รูปที่ 3.4 แสดงส่วนประกอบต่างๆ ของโครงร่าง จากรูปจะเห็นได้ว่าโครงร่างถูกออกแบบมาให้ทำงานกับไฟล์นามสกุล XMI (XML Metadata Interchange) ซึ่งเป็นรูปแบบของไฟล์ที่ใช้อธิบายและจัดเก็บโมเดลยูเอ็มแอลที่เครื่องมือที่ใช้สร้างโมเดลยูเอ็มแอลส่วนมากสนับสนุน

โครงร่างจะรับไฟล์ XMI เป็นอินพุทและให้ XMI ที่ถูกปรับปรุงแล้วเป็นเอาท์พุท ภายในโครงร่างจะเกิดการแปลงกลับไปมาระหว่างโครงสร้างข้อมูลสองแบบด้วยกันคือโมเดลยูเอ็มแอล

และคอลยูสกราฟเนื่องจากส่วนวิธีการประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งจะต้องรับอินพุทในรูปแบบของอาเรย์ซึ่งก็คือคอลยูสกราฟเพื่อสุ่มเปลี่ยนแปลงค่าของอาเรย์นั้น แต่ในส่วนของปลั๊กอินเพื่อปรับปรุงโมเดลการออกแบบต่างๆ ที่แสดงในด้านบนสุดของรูปที่ 3.4 จะรับอินพุทเป็นโมเดลยูเอ็มแอลเพื่อทำการวัดคุณภาพของโมเดลยูเอ็มแอลนั้น ดังนั้นการแปลงกลับไปมาดังกล่าวจะเกิดขึ้นในแต่ละวงรอบการทำงานของวิธีซีมูเล็ทเต็ดแอนนิลลิงจนกว่าจะได้รูปแบบเหมาะสมของโมเดลยูเอ็มแอลที่เป็นอินพุท

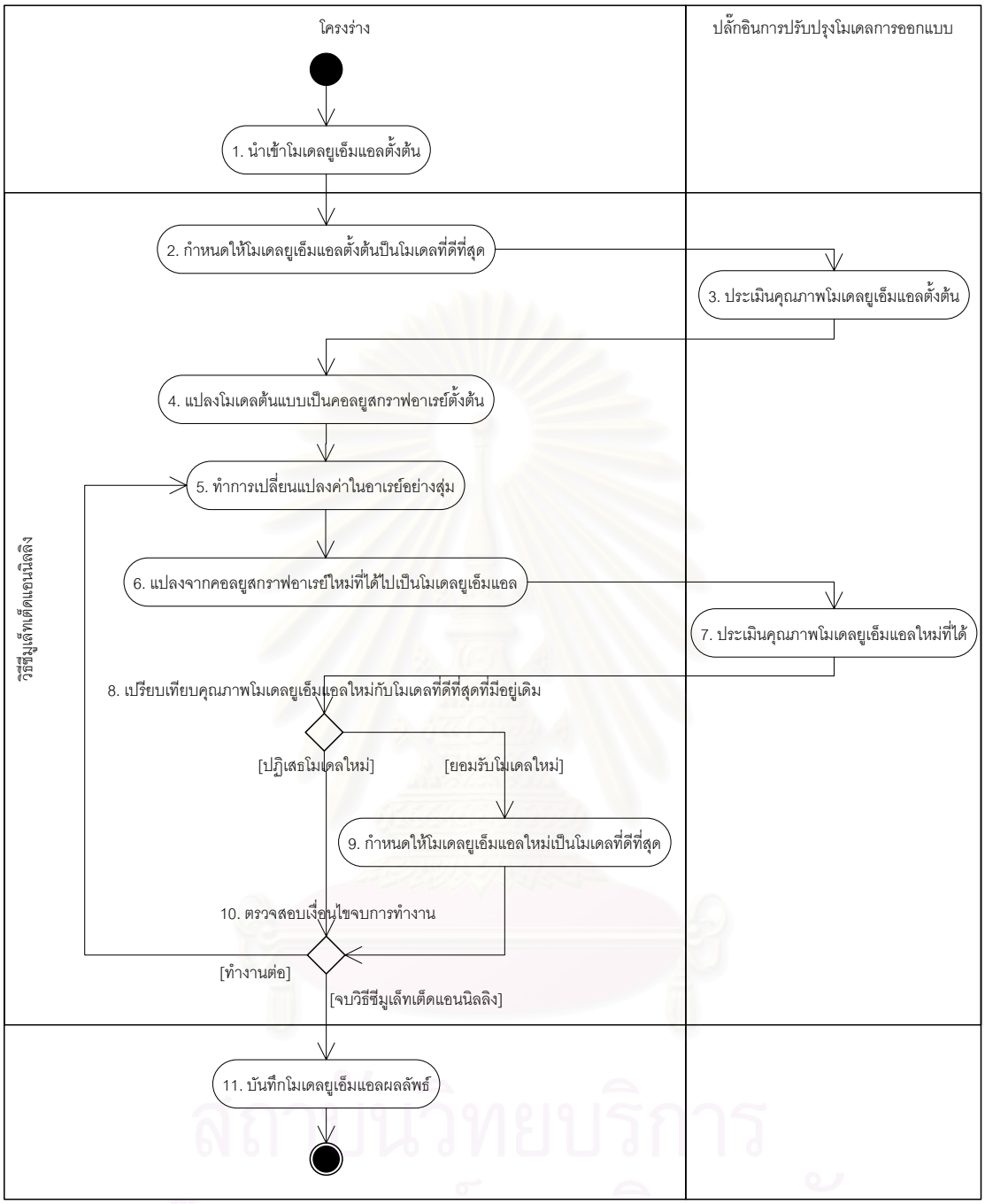
สถาปัตยกรรมของโครงร่างจะถูกอธิบายอย่างละเอียดอีกครั้งในบทที่ 4 ส่วนพฤติกรรมการทำงานของโครงร่างจะถูกอธิบายในหัวข้อถัดไป



รูปที่ 3.4 ส่วนประกอบของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

3.1.4.2 วิธีการทำงานของโครงร่าง

หัวข้อนี้จะอธิบายถึงการนำเอาโมเดลยูเอ็มแอลมาใช้ในโครงร่าง รูปที่ 3.5 ได้อธิบายถึงหลักการทำงานและการเปลี่ยนแบบของโมเดลการออกแบบในโครงร่างเพื่อการทำความเข้าใจได้ง่ายขึ้น



รูปที่ 3.5 การทำงานของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

จากรูปที่ 3.5 จะเห็นได้ว่าการที่จะให้โมเดลการออกแบบที่ถูกนำเข้ามาสามารถใช้ในการทำงานร่วมกับส่วนของการหารูปแบบที่เหมาะสมด้วยวิธีซีมูเล็ทเต็ดแอนนิลลิงได้นั้น โมเดลยูเอ็มแอลที่ถูกนำเข้ามานั้นจะถูกแปลงไปเป็นคอลยูสกราฟอาเรย์ และในระหว่างกระบวนการเปลี่ยนแปลงรูปแบบของโมเดลการออกแบบ เมื่อวิธีซีมูเล็ทเต็ดแอนนิลลิงได้ทำการสุ่มรูปแบบของคอลยูสกราฟขึ้นมาใหม่แต่ละครั้งก็จำเป็นจะต้องส่งให้ปลั๊กอินการปรับปรุงโมเดลการออกแบบ ที่

ทำหน้าที่เป็นฟังก์ชันการประเมิน ไปประเมินค่าคุณภาพของโมเดลการออกแบบนั้นๆ ด้วยมาตรวัดต่างๆ ที่ตัวปลั๊กอินตัวนั้นกำหนดไว้ แต่เนื่องจากโครงร่างนี้เปิดโอกาสให้นักพัฒนาทั้งหลายเป็นผู้สร้างปลั๊กอินใหม่ๆ ขึ้นมาได้ ดังนั้นโมเดลการออกแบบที่จะถูกส่งไปให้ปลั๊กอินทำการประเมินค่าคุณภาพจึงควรถูกอธิบายในรูปแบบของโมเดลยูเอ็มแอลซึ่งเป็นรูปแบบที่นักพัฒนาทั่วไปคุ้นเคย จากสาเหตุนี้คอลยูสกราฟจึงถูกแปลงกลับไปเป็นโมเดลยูเอ็มแอลอีกครั้ง

จากความต้องการข้างต้น กฎเกณฑ์ของการแปลงจากโมเดลยูเอ็มแอลไปเป็นคอลยูส-กราฟ และจากคอลยูสกราฟไปเป็นโมเดลยูเอ็มแอล จึงถูกกำหนดขึ้นมา โดยที่รายละเอียดของกฎเกณฑ์ที่ว่านี้จะถูกอธิบายอย่างละเอียดในหัวข้อถัดๆ ไป

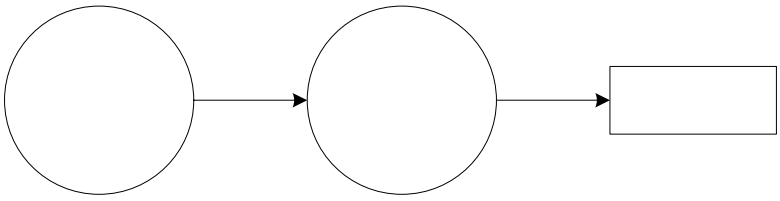
3.1.4.3 การแปลงระหว่างโมเดลยูเอ็มแอลและคอลยูสกราฟ

3.1.4.3.1 การแปลงจากโมเดลยูเอ็มแอลไปเป็นคอลยูสกราฟ

คอลยูสกราฟที่มีรายละเอียดดังที่แสดงไว้ในหัวข้อ 2.2.2 สามารถถูกสร้างขึ้นมาได้โดยใช้ข้อมูลจากโมเดลยูเอ็มแอลที่มีอยู่ได้

- **การสร้างคอลยูสกราฟจากโมเดลยูเอ็มแอล**
    1. สร้างโหนดต่าง ๆ ของกราฟจากทุกๆ เมทธอดและแอตทริบิวต์ที่ไม่ใช่โรล ที่ถูกระบุในแผนภาพคลาสต่างๆ ในโมเดลการออกแบบ
    2. สร้างเส้นเชื่อมต่อระหว่างโหนดโดยอาศัยข้อมูลการเรียกใช้กันระหว่างเมทธอดหรือระหว่างเมทธอดกับแอตทริบิวต์จากแผนภาพซีเควนซ์ต่างๆ ในโมเดลการออกแบบ

จากข้อมูลของโมเดลการออกแบบในรูปที่ 3.2 และรูปที่ 3.3 เมื่อนำมาสร้างคอลยูสกราฟจะได้ผลลัพธ์ดังนี้



รูปที่ 3.6 คอลยูสกราฟที่สร้างมาจากข้อมูลของโมเดลการออกแบบในรูปที่ 3.2 และรูปที่ 3.3

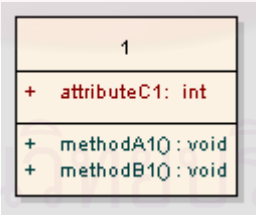
ซึ่งคอลยูสกราฟในรูปที่ 3.6 สามารถถูกเขียนในรูปแบบของอาเรย์ได้ดังรูปที่ 3.7 โดยที่มีข้อกำหนดคืออาเรย์ช่องที่ 1, 2, และ 3 จะใช้สำหรับแสดงค่าข้อมูลของ methodA1, methodB1 และ attributeC1 ตามลำดับ และกำหนดให้คลาส A, B และ C ถูกระบุด้วยหมายเลข 1, 2 และ 3 ตามลำดับ จากข้อกำหนดดังกล่าวประกอบกับค่าที่ถูกกำหนดในแต่ละช่องของอาเรย์ที่ได้นี้ จะเป็นตัวบอกข้อมูลว่าเมทธอดหรือแอตทริบิวต์ตัวหนึ่ง ๆ เป็นสมาชิกของคลาสใดในโมเดลการออกแบบ

1	2	3

รูปที่ 3.7 อาเรย์ที่สอดคล้องกับกราฟในรูปที่ 3.6

3.1.4.3.2 การแปลงจากคอลยูสกราฟกลับไปเป็นโมเดลยูเอ็มแอล

วิธีการปรับปรุงคุณภาพของโมเดลการออกแบบที่ถูกคิดค้นขึ้นมาใน [9] นั้นไม่ได้นำเรื่องของบริบทการถูกเรียกใช้ (Called Context) ของข้อมูลมาพิจารณา จึงทำให้ข้อมูลบางตัวไม่ได้ถูกแสดงในแผนภาพคลาสหลังจากกระบวนการปรับปรุงคุณภาพแล้ว เช่น จากรูปที่ 3.7 ถ้าค่าของอาเรย์ในแต่ละช่องถูกกำหนดให้เป็นค่าเดียวกัน อย่างเช่นมีค่าเป็น 1 คือเมทธอดและแอตทริบิวต์ทุกตัวในโมเดลการออกแบบถูกย้ายให้ไปรวมอยู่ในคลาสหมายเลข 1 ทั้งหมด ดังนั้นทั้งโมเดลการออกแบบจะเหลือคลาสอยู่คลาสเดียวในที่นี้ก็คือคลาสหมายเลข 1



รูปที่ 3.8 ตัวอย่างคลาสที่ได้จากการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบด้วยวิธีของ [9]

จากรูปที่ 3.8 จะเห็นได้ว่าปริมาณข้อมูลเดิมของโมเดลการออกแบบลดลงเนื่องจากว่าในโมเดลการออกแบบผลลัพธ์คงเหลือข้อมูลของ attributeC1 ที่มีชนิดเป็น int เพียงแค่ตัวเดียว ซึ่งเมื่อพิจารณาจากโมเดลต้นแบบแล้วข้อมูล attributeC1 ปรากฏอยู่ในโมเดลถึง 2 ตัว ก็คือ attributeC1 ของออบเจ็กต์ c1 ที่เป็นแอตทริบิวต์ของออบเจ็กต์ b1 และ attributeC1 ของ

ออบเจ็กต์ c1 ที่เป็นแอตทริบิวต์ของออบเจ็กต์ b2 ซึ่งทั้ง b1 และ b2 ต่างก็เป็นแอตทริบิวต์ของคลาส A ทั้งคู่

ด้วยเหตุนี้ผู้จัดทำวิทยานิพนธ์จึงได้ปรับปรุงกระบวนการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบที่ถูกกำหนดภายใน [9] ให้เหมาะสำหรับการทำงานของโครงร่างโดยการนำเอาบริบทการถูกเรียกใช้เข้ามาเกี่ยวข้องกับกระบวนการการเปลี่ยนแปลง โดยจะกำหนดรายการของบริบทการถูกเรียกใช้ให้กับทุกเมทธอด, แอตทริบิวต์, โรล และเมสเสจ ที่ได้ในโมเดลผลลัพธ์ ดังแสดงไว้ในรูปที่ 3.9 และรูปที่ 3.10 เพื่อป้องกันการสูญหายของข้อมูลของโมเดลผลลัพธ์ และนอกจากนั้นผู้ใช้ที่เป็นนักพัฒนาโปรแกรมคอมพิวเตอร์สามารถที่จะระบุที่มาและบริบทของการถูกเรียกใช้ของทุกเมทธอด, แอตทริบิวต์ และเมสเสจ ของโมเดลผลลัพธ์ได้ โดยมีรูปแบบที่จะได้อธิบายต่อไป

การแปลงคอลยูสกราฟกลับไปเป็นโมเดลยูเอ็มแอลดังที่กล่าวไปแล้วนี้ ไม่ได้คำนึงถึงความหมายของคลาสเดิมที่มีอยู่ในโมเดลต้นแบบ ดังที่เห็นได้จากคลาสทุกตัวที่ปรากฏในโมเดลผลลัพธ์จะถูกตั้งชื่อเป็นหมายเลขที่ไม่มีความเกี่ยวข้องกับความหมายของชื่อคลาสในโมเดลต้นแบบเลย การที่พฤติกรรมของระบบยังคงเหมือนเดิมได้นั้นเกิดจากการที่โมเดลการออกแบบผลลัพธ์ยังคงมีเมทธอด, แอตทริบิวต์ และลำดับการเรียกใช้กันของทั้งเมทธอดและแอตทริบิวต์เหมือนกับกับโมเดลต้นแบบทุกประการ

รูปแบบการระบุบริบทการถูกเรียกใช้ในโมเดลการออกแบบ

บริบทการถูกเรียกใช้ คือสภาวะแวดล้อมหนึ่งๆ ที่ข้อมูลตัวใดๆ ถูกเรียกใช้ และจากแนวความคิดที่ว่าบริบทการถูกเรียกใช้ข้อมูลควรจะถูกระบุอย่างละเอียดให้กับทุก เมทธอด, แอตทริบิวต์, โรล และเมสเสจ ทุกตัว เพื่อป้องกันการสูญหายของข้อมูล ดังนั้นในงานวิจัยนี้จึงได้มีการกำหนดรูปแบบการระบุบริบทการถูกเรียกใช้ โดยมีนิยามเป็น

{[*T_*]<*ObjectName*>}

รูปที่ 3.9 รูปแบบของการระบุโรลในโมเดลการออกแบบผลลัพธ์

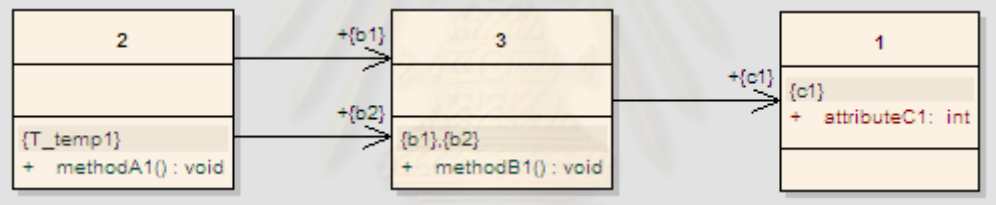
โดยที่

- <*ObjectName*> หมายถึงชื่อของออบเจ็กต์ที่ถูกเรียกใช้ในแผนภาพซีเควนซ์ในโมเดลต้นแบบ โดยถ้าออบเจ็กต์ตัวใดๆ มีชื่อตรงกันกับโรลที่ถูกระบุที่ปลายความสัมพันธ์แบบ

แอสโซซิเอชันในแผนภาพคลาสจะถือว่าเป็นแอตทริบิวต์ของคลาสที่อยู่ติดกับปลายความสัมพันธ์อีกด้านหนึ่ง แต่ถ้าออบเจ็กต์ตัวนั้นๆ ไม่มีชื่อตรงกันกับโรลใดๆ ในแผนภาพคลาส ก็จะถือว่าออบเจ็กต์ตัวนั้นเป็นออบเจ็กต์ชั่วคราว

- *T_* เป็นส่วนระบุว่าบริบทการถูกเรียกใช้นั้นๆ เป็นบริบทการถูกเรียกใช้ของออบเจ็กต์ชั่วคราว
- รายการของบริบทการถูกเรียกใช้คือ ชุดของบริบทการถูกเรียกใช้ที่ถูกนำมารวมกันโดยคั่นระหว่างบริบทการถูกเรียกใช้แต่ละตัวด้วยเครื่องหมายจุลภาค

การนำเอาบริบทการถูกเรียกใช้มาระบุให้กับโมเดลการออกแบบในรูป 3.2 และรูป 3.3 นั้นสามารถแสดงได้ดัง รูปที่ 3.10 และรูปที่ 3.11



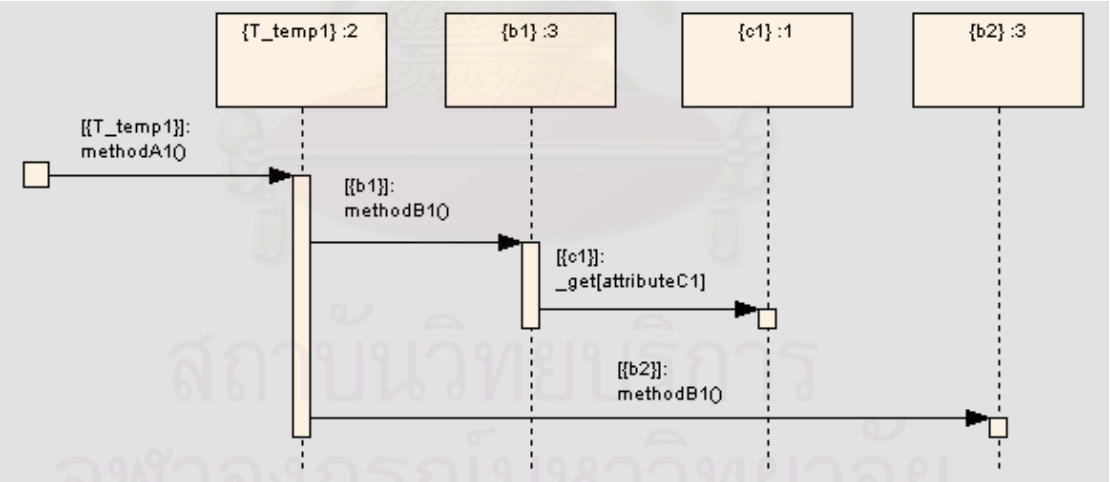
รูปที่ 3.10 รูปแบบของการระบุบริบทการถูกเรียกใช้ของข้อมูลจากแผนภาพคลาสต้นแบบในรูปที่ 3.2

รายละเอียดของการระบุรายการบริบทการถูกเรียกใช้ของแผนภาพคลาสในรูปที่ 3.10 สามารถถูกอธิบายได้เป็นข้อๆ ดังนี้

- เมทธอด methodA1 ถูกเรียกใช้ในบริบทของออบเจ็กต์ temp1 ซึ่งเป็นออบเจ็กต์ชั่วคราว
- เมทธอด methodB1 ถูกเรียกใช้ใน 2 บริบทคือ บริบทของออบเจ็กต์ b1 และ b2 ซึ่งทั้งคู่ไม่ใช่ออบเจ็กต์ชั่วคราว
- แอตทริบิวต์ attributeC1 ถูกเรียกใช้ในบริบทของออบเจ็กต์ c1
- ทุกๆ โรลจะถูกเปลี่ยนชื่อให้กลายเป็นบริบทการถูกเรียกใช้ที่สอดคล้องกับออบเจ็กต์ (แอตทริบิวต์) ของแต่ละโรล

รายละเอียดของการระบุรายการบริบทการถูกเรียกใช้ของแผนภาพซีเควนซ์ในรูปที่ 3.11 สามารถถูกอธิบายได้เป็นข้อๆ ดังนี้

- ออบเจ็กต์แต่ละตัวจะถูกเปลี่ยนชื่อให้กลายเป็นบริบทการถูกเรียกใช้ที่สอดคล้องกับออบเจ็กต์นั้น ๆ
- เมสเสจ methodA1() จากไดอะแกรมเกทไปยังออบเจ็กต์ {T_temp1} ถูกเรียกใช้ในบริบทของออบเจ็กต์ temp1 ซึ่งเป็นออบเจ็กต์ชั่วคราว
- เมสเสจ methodB1() จากออบเจ็กต์ {T_temp1} ไปยังออบเจ็กต์ {b1} ถูกเรียกใช้ในบริบทของออบเจ็กต์ b1
- เมสเสจ _get[attributeC1] จากออบเจ็กต์ {b1} ไปยังออบเจ็กต์ {c1} ถูกเรียกใช้ในบริบทของออบเจ็กต์ c1
- เมสเสจ methodB1() จากออบเจ็กต์ {T_temp1} ไปยังออบเจ็กต์ {b2} ถูกเรียกใช้ในบริบทของออบเจ็กต์ b2



รูปที่ 3.11 รูปแบบของการระบุบริบทการถูกเรียกใช้ของข้อมูลจากแผนภาพซีเควนซ์ต้นแบบในรูปที่ 3.3

ในการใช้งานจริง รายการบริบทการถูกเรียกใช้จะถูกระบุให้กับส่วนต่างๆ ของโมเดลยูเอ็มแอลดังนี้

- สำหรับเมทธอด รายการบริบทการถูกเรียกใช้จะถูกระบุให้เป็นสเตอรีโอไทป์ (Stereotype) ของเมทธอดนั้น ๆ
- สำหรับแอตทริบิวต์ รายการบริบทการถูกเรียกใช้จะถูกระบุให้เป็นสเตอรีโอไทป์ของแอตทริบิวต์นั้น ๆ
- สำหรับโรล บริบทการถูกเรียกใช้จะถูกระบุให้เป็นชื่อของโรลนั้น ๆ
- สำหรับออบเจ็กต์ในแผนภาพซีเควนซ์ บริบทการถูกเรียกใช้จะถูกระบุให้เป็นชื่อของออบเจ็กต์นั้น ๆ
- สำหรับเมสเสจ บริบทการถูกเรียกใช้จะถูกระบุให้เป็นการ์ดคอนดิชัน (Guard Condition) ของเมสเสจนั้น ๆ

รูปที่ 3.10 และ รูปที่ 3.11 ใช้ข้อกำหนดในการระบุบริบทการถูกเรียกใช้ด้านบน โดยที่แผนภาพทั้งสองถูกแสดงผลในรูปแบบของโปรแกรมประยุกต์ Sparx Enterprise Architect

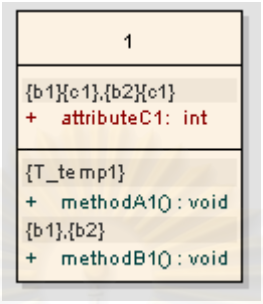
รูปแบบการระบุบริบทการถูกเรียกใช้แบบประกอบ (Compound Called Context) ในโมเดลผลลัพธ์

บริบทการถูกเรียกใช้แบบประกอบ คือบริบทการถูกเรียกใช้ที่ถูกประกอบมาจากบริบทการถูกเรียกใช้อื่นๆ ถูกกำหนดได้โดยการนำบริบทการถูกเรียกใช้ต่างๆ มาเขียนต่อกัน โดยบริบทการถูกเรียกใช้ที่เป็นส่วนประกอบตัวที่อยู่นำหน้าจะแสดงถึงการเรียกใช้บริบทการถูกเรียกใช้ที่ตามมาในลำดับที่ติดกันอีกที โดยอ้างอิงจากโมเดลต้นแบบ เช่น

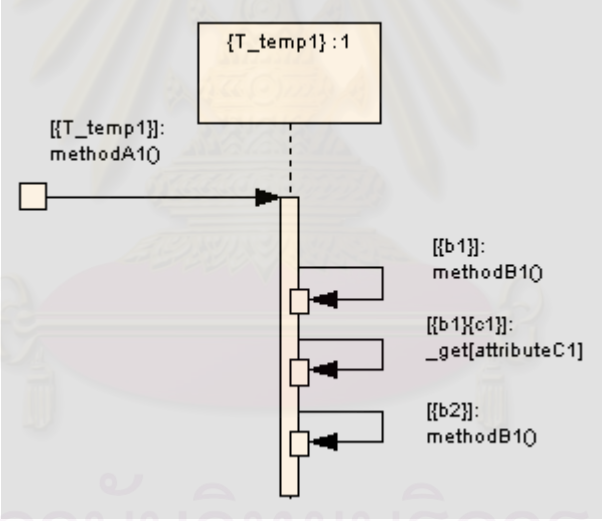
- {b1}{c1} เป็นการระบุบริบทการถูกเรียกใช้แบบประกอบที่แสดงถึงการที่บริบทการถูกเรียกใช้{b1} ไปเรียกใช้บริบทการถูกเรียกใช้ {c1}
- {b2}{c1} เป็นการระบุบริบทการถูกเรียกใช้แบบประกอบที่แสดงถึงการที่บริบทการถูกเรียกใช้{b2} ไปเรียกใช้บริบทการถูกเรียกใช้ {c1}

ประโยชน์ของการใช้บริบทการถูกเรียกใช้แบบประกอบ จะถูกใช้ก็ต่อเมื่อลักษณะทางกายภาพของโมเดลยูเอ็มแอลผลลัพธ์ที่ได้จากกระบวนการเปลี่ยนแปลงโครงสร้างเพื่อปรับปรุงคุณภาพของโมเดลการออกแบบ ไม่สามารถแสดงได้ถึงปริมาณข้อมูลจริงที่มีอยู่ในระบบ กล่าวคือการเรียกใช้กันของสมาชิกของเมทธอดและแอตทริบิวต์ที่ไม่ได้เป็นสมาชิกของคลาสเดียวกันในโมเดลต้นแบบแต่ถูกย้ายมาให้อยู่ในคลาสเดียวกันในโมเดลผลลัพธ์ ดังปัญหาที่แสดงด้วยรูปที่ 3.8

และจากปัญหานี้ เราสามารถนำบริบทการถูกเรียกใช้แบบประกอบมาระบุเพื่อแก้ไขปัญหาได้ดังรูปที่ 3.12 และรูปที่ 3.13



รูปที่ 3.12 ตัวอย่างแผนภาพคลาสที่ได้จากการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบด้วยวิธีที่ปรับปรุงขึ้น



รูปที่ 3.13 ตัวอย่างแผนภาพซีเควนซ์ที่ได้จากการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบด้วยวิธีที่ปรับปรุงขึ้น

แอตทริบิวต์ attributeC1 ในรูปที่ 3.12 และ เมสเสจ _get[attributeC1] ในรูปที่ 3.13 ถูกระบุบริบทการถูกเรียกใช้เป็นแบบบริบทการถูกเรียกใช้แบบประกอบ

ในอีกมุมมองหนึ่ง ปริมาณข้อมูลที่แท้จริงของสมาชิกที่ถูกระบุด้วยบริบทการถูกเรียกใช้แบบประกอบนั้นสามารถตีความได้เป็นการนำเอามัลติพลิซิตีของแต่ละบริบทย่อยในบริบทการถูกเรียกใช้แบบประกอบนั้นมาคูณกัน

การระบุบริบทการถูกเรียกใช้ในแผนภาพคลาสและแผนภาพซีเควนซ์

การระบุบริบทการถูกเรียกใช้ให้กับองค์ประกอบส่วนต่างๆ ในแผนภาพทั้งสองจะมีจุดประสงค์ในการระบุที่แตกต่างกันดังนี้

- การระบุบริบทการถูกเรียกใช้ให้กับองค์ประกอบต่างๆ ในแผนภาพคลาสจะเป็นการระบุบริบทการถูกเรียกใช้ทั้งหมดที่เป็นไปได้ขององค์ประกอบตัวนั้นๆ เช่นจากรูปที่ 3.12 แอตทริบิวต์ attributeC1 ถูกระบุรายการของบริบทการถูกเรียกใช้เป็น {b1}{c1},{b2}{c1} หมายความว่า แอตทริบิวต์ attributeC1 จะถูกเรียกใช้ทั้งหมด 2 บริบทคือ {b1}{c1} และ {b2}{c1} ในระบบที่ถูกอธิบายด้วยโมเดลการออกแบบนี้
- การระบุบริบทการถูกเรียกใช้ให้กับองค์ประกอบต่างๆ ในแผนภาพซีเควนซ์จะเป็นการระบุบริบทการถูกเรียกใช้ในจุดเวลาที่กำลังพิจารณา เช่นจากรูปที่ 3.13 จะพบเมสเสจที่มีชื่อว่า methodB1() อยู่สองตัวด้วยกัน แต่ถูกระบุบริบทการถูกเรียกใช้ที่แตกต่างกัน คือ เมสเสจ methodB1() ตัวที่อยู่ข้างบนถูกระบุว่าถูกเรียกใช้ในบริบท {b1} ในขณะที่เมสเสจตัวด้านล่างถูกระบุว่าถูกเรียกใช้ในบริบท {b2}

การแปลความหมายของบริบทการถูกเรียกใช้ที่ระบุในโมเดลการออกแบบผลลัพธ์

รูปที่ 3.12 และ รูปที่ 3.13 แสดงถึงโมเดลการออกแบบผลลัพธ์ที่ถูกเปลี่ยนแปลงแล้วด้วยวิธีการที่ปรับปรุงขึ้นมาใหม่ โดยการใช้บริบทการถูกเรียกใช้มาประกอบ ซึ่งรายการของบริบทการถูกเรียกใช้ที่ผูกติดอยู่กับ เมทธอด, แอตทริบิวต์, โรล หรือ เมสเสจ จะเป็นตัวบอกพฤติกรรมของสิ่งเหล่านั้นเช่น

- จากแผนภาพคลาสของโมเดลผลลัพธ์ที่ได้ในรูปที่ 3.12 จะเห็นว่ามีรายการของบริบทการถูกเรียกใช้ที่ระบุให้กับ methodB1 คือ {b1},{b2} ดังนั้นเราสามารถอธิบายได้ว่า methodB1 ถูกเรียกใช้ในบริบทที่แตกต่างกัน 2 บริบทในโมเดลต้นแบบ ได้แก่
  - เรียกจากออบเจ็กต์ b1 ซึ่งเป็นแอตทริบิวต์ของคลาส A
  - เรียกจากออบเจ็กต์ b2 ซึ่งเป็นแอตทริบิวต์ของคลาส A

- จากแผนภาพคลาสของโมเดลผลลัพธ์ที่ได้ในรูปที่ 3.12 จะเห็นว่ามีรายการของบริบทการถูกเรียกใช้ที่ระบุให้กับ attributeC1 คือ {b1}{c1},{b2}{c1} โดยที่สมาชิกของรายการบริบทการถูกเรียกใช้รายการนี้ต่างก็เป็นบริบทการถูกเรียกใช้แบบประกอบทั้งคู่โดยที่

บริบทการถูกเรียกใช้เหล่านี้แสดงว่าเคยมี attributeC1 ที่แตกต่างกันอยู่ 2 ตัวในโมเดลต้นแบบ คือ

- attributeC1 ของออบเจ็กต์ c1 ที่ออบเจ็กต์ b1 เรียกใช้และเป็นเจ้าของ
- attributeC1 ของออบเจ็กต์ c1 ที่ออบเจ็กต์ b2 เรียกใช้และเป็นเจ้าของ

- จากแผนภาพซีเควนซ์ของโมเดลผลลัพธ์ที่ได้ในรูปที่ 3.13 ความสัมพันธ์ของบริบทการถูกเรียกใช้ของเมทธอด methodB1 กับ attributeC1 สามารถอธิบายได้ว่า
  - ถ้าเมทธอด methodB1 ถูกเรียกใช้ภายใต้บริบทการถูกเรียกใช้ที่ระบุได้ด้วย {b1} แล้วเมทธอดนี้จะไปเรียกใช้แอตทริบิวต์ attributeC1 ภายใต้บริบทการถูกเรียกใช้ที่ระบุได้ด้วย {b1}{c1} ซึ่งจะเห็นว่า {b1} เป็นส่วนประกอบทางด้านซ้ายของ {b1}{c1}
  - ถ้าเมทธอด methodB1 ถูกเรียกใช้ภายใต้บริบทการถูกเรียกใช้ที่ระบุได้ด้วย {b2} แล้วเมทธอดนี้จะไปเรียกใช้แอตทริบิวต์ attributeC1 ภายใต้บริบทการถูกเรียกใช้ที่ระบุได้ด้วย {b2}{c1} ซึ่งจะเห็นว่า {b2} เป็นส่วนประกอบทางด้านซ้ายของ {b2}{c1}

ขั้นตอนในการแปลงจากคอลยูสกราฟไปเป็นโมเดลยูเอ็มแอล

ในหัวข้อนี้จะอธิบายลำดับของวิธีการแปลงคอลยูสกราฟกลับมาเป็นโมเดลยูเอ็มแอล โดยจะใช้โมเดลตัวอย่างจากรูปที่ 3.2 รูปที่ 3.3 โดยถ้ากำหนดอาเรย์ของคอลยูสกราฟในรูปที่ 3.14 เป็นโจทย์ในการแปลง จะได้โมเดลผลลัพธ์ดังแสดงไว้ในรูปที่ 3.12 และรูปที่ 3.13

1	1	1

รูปที่ 3.14 อาเรย์ที่สอดคล้องกับรูปที่ 3.8

ข้อกำหนดในรูปที่ 3.14 คืออาเรย์ช่องที่ 1, 2, และ 3 จะใช้สำหรับแสดงค่าข้อมูลของ methodA1, methodB1 และ attributeC1 ตามลำดับ

- **การสร้างแผนภาพคลาสจากคอลยูสกราฟ**

1. จากข้อมูลของคอลยูสกราฟแบบอาเรย์ในรูปที่ 3.14 นำเมทธอดและแอตทริบิวต์ทุกตัวที่มีหมายเลขคลาสเดียวกัน มารวมกันเป็นสมาชิกของคลาสใหม่ที่ถูกสร้างขึ้นและคลาสใหม่นี้จะมีชื่อคลาสที่เป็นหมายเลขของคลาสนั้นๆ
2. ระบุรายการของบริบทการถูกเรียกใช้ให้กับสมาชิกแต่ละตัวในคลาสจากข้อมูลของคอลยูสกราฟและโมเดลต้นแบบ
3. การกำหนดความสัมพันธ์แบบแอสโซซิเอชันระหว่างคลาสสองคลาส จะดูจากการที่สมาชิกในคลาสต้นทางหนึ่งตัวไปเรียกสมาชิกในคลาสปลายทาง จากนั้นให้ระบุชื่อโรลที่ปลายด้านที่มีลูกศรของความสัมพันธ์ให้มีค่าเท่ากับบริบทการถูกเรียกใช้ของสมาชิกของคลาสที่เรียกใช้กันคู่นั้น โดยดูข้อมูลจากโมเดลต้นแบบ โดยที่คลาสต้นทางและคลาสปลายทางคู่หนึ่งๆ อาจจะมีความสัมพันธ์แบบแอสโซซิเอชันมากกว่าหนึ่งความสัมพันธ์ได้ โดยให้แยกตามโรลปลายทางที่แตกต่างกัน ทั้งนี้บริบทการถูกเรียกใช้ที่จะถูกนำมาระบุเป็นชื่อโรลจะต้องไม่ประกอบด้วยบริบทการถูกเรียกใช้ของออบเจ็กต์ชั่วคราว
4. นำคลาสทั้งหมดที่สร้างมาใส่ลงไปในแผนภาพคลาสที่ถูกสร้างขึ้นมาใหม่แผนภาพเดียว

- **การสร้างแผนภาพซีเควนซ์จากคอลยูสกราฟ**

1. สร้างแผนภาพซีเควนซ์ขึ้นมาใหม่โดย หนึ่งแผนภาพซีเควนซ์ในโมเดลต้นแบบจะถูกนำมาสร้างเป็นหนึ่งแผนภาพซีเควนซ์ในโมเดลผลลัพธ์
2. สร้างเมสเสจแต่ละตัวในแผนภาพซีเควนซ์ที่ถูกสร้างขึ้นมาใหม่ โดยเมสเสจแต่ละตัวที่ถูกสร้างขึ้นมาจะต้องเข้าคู่กันกับเมสเสจในแผนภาพซีเควนซ์ต้นแบบกล่าวคือจะต้องมีชื่อ และลำดับในการเกิดของเมสเสจเป็นไปตามที่แผนภาพซีเควนซ์ต้นแบบระบุไว้
3. จากนั้นระบุบริบทการถูกเรียกใช้ให้กับเมสเสจที่เกิดใหม่โดยใช้ข้อมูลจากคอลยูสกราฟและโมเดลต้นแบบ
4. สร้างไดอะแกรมเกทเป็นตัวส่งเมสเสจตัวแรกที่อยู่ซ้ายสุดในแผนภาพซีเควนซ์
5. สร้างออบเจ็กต์ที่เป็นผู้รับเมสเสจโดยชื่อของออบเจ็กต์นี้จะมีค่าเท่ากับบริบทการถูกเรียกใช้ของตัวเมสเสจเอง

6. กำหนดออบเจ็กต์ที่เป็นผู้ส่งเมสเสจให้เป็นออบเจ็กต์ผู้รับเมสเสจของเมสเสจที่เรียกใช้เมสเสจนี้อีกที

โดยสรุปแล้วโมเดลผลลัพธ์ที่ได้มาดังตัวอย่างที่แสดงไว้ในรูปที่ 3.12 และรูปที่ 3.13 นั้นเป็นโมเดลที่เอาไว้ใช้ชี้นำในการหาโมเดลผลลัพธ์ในขั้นสมบูรณ์พอที่จะเอาไปใช้ได้เลยอีกที โดยเพื่อการนี้ผู้ใช้จะต้องทำการพิจารณาโมเดลผลลัพธ์ที่ได้จากโครงร่างในเรื่องดังนี้

- การตีความบริบทการถูกเรียกใช้โดยที่บริบทการถูกเรียกใช้อาจถูกแปลงไปเป็นพารามิเตอร์ของเมทธอดหรือกลายเป็นอาเรย์ของแอตทริบิวต์หนึ่งๆ ทั้งนี้ขึ้นอยู่กับการตัดสินใจของผู้ใช้ซึ่งเป็นนักพัฒนาโปรแกรม
- การเปลี่ยนชื่อของเมทธอดหรือแอตทริบิวต์ที่เป็นสมาชิกของคลาสเดียวกันแต่มีชื่อซ้ำกัน
- การตั้งชื่อคลาสใหม่ให้เหมาะสม

3.1.5 การพัฒนาปลั๊กอินการปรับปรุงโมเดลการออกแบบ

ในขั้นตอนนี้เป็นการพัฒนาปลั๊กอินที่โครงร่างจะมาเรียกใช้เพื่อการประเมินคุณภาพของโมเดลการออกแบบในแต่ละรอบของการสุ่มเปลี่ยนแปลงโมเดลการออกแบบตามวิธีซีมูเล็ทเต็ด-แอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง หรือสามารถอธิบายได้อีกอย่างว่าปลั๊กอินการปรับปรุงการออกแบบแต่ละตัวคือ ฟังก์ชันการประเมิน ของวิธีซีมูเล็ทเต็ดแอนนิลลิงนั่นเอง

ฟังก์ชันการประเมินของวิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง ที่พัฒนาขึ้นสำหรับโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุนี้ จะมีลักษณะเป็นปลั๊กอิน ซึ่งมีข้อดีคือสามารถถอดเปลี่ยนได้ง่ายรวมไปถึงความสามารถในการเปลี่ยนฟังก์ชันการประเมินได้ในขณะรันไทม์

ฟังก์ชันการประเมินที่ผู้จัดทำวิทยานิพนธ์ได้จัดทำขึ้นมาเป็นชนิดที่มีค่าเหมาะสมเป็นค่าที่น้อยที่สุดเหมือนกันทั้งหมด โดยที่มีอยู่ 3 ฟังก์ชันด้วยกันดังนี้

- ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบเพื่อเป็นตัวแทนของตัวอย่างการทำฟังก์ชันการประเมินสำหรับปรับปรุงร่องรอยที่ไม่ดีในระดับเมทธอด

- ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบเพื่อเป็นตัวแทนของตัวอย่างการทำฟังก์ชันการประเมินสำหรับปรับปรุงร่องรอยที่ไม่ดีในระดับคลาส
- ฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน (Cohesion) และการเข้าคู่กัน (Coupling) ของคลาสภายในโมเดลการออกแบบเพื่อเป็นฟังก์ชันการประเมินโดยปริยายของโครงร่าง

โดยที่ฟังก์ชันการประเมินตัวสุดท้ายนั้นถือเป็นฟังก์ชันโดยปริยายในโครงร่างนี้ กล่าวคือถ้าไม่ถูกกำหนดเป็นอย่างอื่นแล้วโครงร่างจะเลือกใช้ฟังก์ชันตัวนี้ ส่วนฟังก์ชันการประเมินอีก 2 ฟังก์ชันที่เหลือผู้จัดทำวิทยานิพนธ์ตั้งใจจะให้เป็นตัวอย่างของแนวทางการลดข้อบกพร่องและร่องรอยที่ไม่ดีประเภทอื่นที่ถูกกำหนดขึ้นในปัจจุบัน

3.1.5.1 ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบ

โดยการนิยามคุณภาพของโมเดลการออกแบบที่ถูกระบุด้วยร่องรอยที่ไม่ดีแบบฟีเจอร์เอน-วีเป็นจำนวนของเมทธอดที่มีร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีในโมเดลการออกแบบ โดยที่ถ้าในโมเดลการออกแบบมีจำนวนเมทธอดประเภทนี้น้อยลงจะแสดงว่ามีคุณภาพในด้านนี้ดีขึ้น

และโดยอ้างอิงจาก [13] กลยุทธ์การตรวจจับเมทธอดที่มีร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีถูกนิยามดังอสมการที่ 2.1 และจากนิยามฟังก์ชันการประเมินนี้จะถูกกำหนดเป็นการนับจำนวนของเมทธอดภายในโมเดลการออกแบบที่ทำให้อสมการที่ 2.1 เป็นจริง

3.1.5.2 ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบ

โดยการนิยามคุณภาพของโมเดลการออกแบบที่ถูกระบุด้วยร่องรอยที่ไม่ดีแบบมิดเดิลแมนเป็นจำนวนของคลาสมีร่องรอยที่ไม่ดีแบบมิดเดิลแมนในโมเดลการออกแบบโดยที่ถ้าในโมเดลการออกแบบมีจำนวนคลาสประเภทนี้น้อยลงจะแสดงว่ามีคุณภาพในด้านนี้ดีขึ้น

และโดยอ้างอิงจาก [13] กลยุทธ์การตรวจจับเมทธอดที่มีร่องรอยที่ไม่ดีแบบมิดเดิลแมนถูกนิยามดังอสมการที่ 2.2 และจากนิยามนี้ฟังก์ชันการประเมินนี้จะถูกกำหนดเป็นการนับจำนวนของคลาสภายในโมเดลการออกแบบที่ทำให้อสมการที่ 2.2 เป็นจริง

3.1.5.3 ฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน (Cohesion) และการเข้าคู่กัน (Coupling) ของคลาสภายในโมเดลการออกแบบ

โดยอ้างอิงจาก [9] และด้วยข้อจำกัดของโครงร่างที่จะไม่พิจารณาความสำคัญของความสัมพันธ์แบบเจนเนอรัลไลเซชัน ฟังก์ชันการประเมินนี้จะถูกกำหนดดังสมการที่ 3.1

$$f(D) = \frac{w_1}{COH(D)} + w_2 COP_{IA}(D) \qquad \text{สมการที่ 3.1}$$

โดยที่

- D แทนโมเดลการออกแบบ
- $COH(D)$ คือมาตรวัดที่ใช้วัดการเกาะกลุ่มกันของคลาสในโมเดลการออกแบบที่นิยามไว้ในสมการที่ 2.5
- $COP_{IA}(D)$ คือมาตรวัดที่ใช้วัดการเข้าคู่กันของคลาสในโมเดลการออกแบบที่นิยามไว้ในสมการที่ 2.9
- w_1 คือน้ำหนักของการเกาะกลุ่มกัน
- w_2 คือน้ำหนักของความเข้าคู่กัน

3.1.6 การพัฒนาเครื่องมือเพื่อทดสอบโครงร่างที่ได้สร้างขึ้น

หลังจากที่โครงร่างและปลั๊กอินได้พัฒนาขึ้น จำเป็นที่จะต้องมีตัวอย่างของโปรแกรมประยุกต์ที่นำเอาโครงร่างที่สมบูรณ์แล้วมาใช้เพื่อทดสอบประสิทธิภาพของโครงร่าง โดยที่รายละเอียดของเครื่องมือและวิธีการใช้จะอธิบายในภาคผนวก ก

3.1.7 การตรวจสอบผลลัพธ์การหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุของเครื่องมือที่ได้จัดทำขึ้น

ขั้นตอนนี้เป็นการประเมินโครงร่างที่ใช้ปรับปรุงคุณภาพของโมเดลการออกแบบที่พัฒนาขึ้นโดยใช้เครื่องมือที่ถูกสร้างขึ้นในหัวข้อที่แล้ว จากนั้นทำการสรุปผลการทดสอบ โดยที่ผลการทดลองต่างๆ จะอธิบายอย่างละเอียดในบทที่ 5

3.2 การประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง

3.2.1 ลักษณะอินพุทของวิธีซีมูเล็ทเต็ดแอนนิลลิงในโครงร่าง

การนำโมเดลการออกแบบที่มีอยู่เข้าไปใช้กับส่วนการปรับปรุงโครงสร้างโมเดลการออกแบบ จะต้องทำการเปลี่ยนแปลงโมเดลการออกแบบที่มีอยู่จากรูปแบบของโมเดลยูเอ็มแอลไปเป็นคอลยูสกราฟที่อธิบายด้วยรูปแบบของอาเรย์เสียก่อน และจากข้อจำกัดของวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิง ซึ่งเป็นพื้นฐานของวิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งที่เลือกใช้ที่ว่า ช่วงค่าของแต่ละพารามิเตอร์จะต้องเป็นจำนวนจำกัด ดังนั้นเมื่อนำเอาข้อจำกัดนี้มาประยุกต์ใช้ในโครงร่างก็จะทำให้ค่าช่วงของแต่ละช่องของอาเรย์ ต้องเป็นจำนวนจำกัดไปด้วย เพราะอาเรย์นี้จะถูกนำไปใช้เป็นอินพุทของวิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง โดยที่จะถือว่าแต่ละช่องของอาเรย์คือพารามิเตอร์ในมิติใดๆ ของวิธีซีมูเล็ตเต็ดแอนนิลลิง

3.2.2 การกำหนดช่วงของค่าที่เป็นไปได้ให้กับอาเรย์คอลยูสกราฟแต่ละช่อง

จากการที่จำนวนช่องของอาเรย์จะมีจำนวนเท่ากับจำนวนโหนดของคอลยูสกราฟ หรือจำนวนเมทธอด และแอตทริบิวต์ภายในโมเดลการออกแบบ และจากการที่จำนวนคลาสที่มากที่สุดที่จะเป็นไปได้จะมีจำนวนเท่ากับจำนวนเมทธอด และแอตทริบิวต์รวมกัน ซึ่งเท่ากับจำนวนช่องของอาเรย์ และจำนวนคลาสที่น้อยที่สุดที่จะเป็นไปได้คือ 1 ดังนั้นค่าช่วงของอาเรย์แต่ละช่องจะมีค่าเป็นได้ตั้งแต่ 1 ถึง จำนวนช่องของอาเรย์ ตัวอย่างเช่น ช่วงค่าที่เป็นไปได้ของสมาชิกของอาเรย์ในรูปที่ 3.7 แต่ละช่องคือ 1 ถึง 3

3.2.3 การใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงกับพารามิเตอร์ที่เป็นจำนวนเต็ม

จากสมการที่ 2.4 และรูปที่ 2.5 ซึ่งอธิบายฟังก์ชันความน่าจะเป็นในการสุ่มเลือกค่าใหม่ให้กับพารามิเตอร์ของฟังก์ชันการประเมินภายใต้วิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิง จะเห็นว่ากราฟฟังก์ชันความน่าจะเป็นนี้ เป็นกราฟต่อเนื่องในช่วงที่สนใจ แต่เนื่องจากพารามิเตอร์ในทุกๆ มิติ ที่ส่วนของการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบจะต้องทำการเปลี่ยนค่า มีชนิดของข้อมูลเป็นเลขจำนวนเต็ม ดังนั้นค่าของพารามิเตอร์ในมิติต่างๆ ที่ถูกสุ่มออกมาจะถูกปัดให้กลายเป็นจำนวนเต็ม

วิธีการปัดเลขทศนิยมที่โครงร่างเลือกใช้จะไม่ใช่เป็นการปัดทศนิยมให้เป็นเลขจำนวนเต็มตามปกติ แต่จะเป็นการเลือกใช้ค่าขอบที่เป็นจำนวนเต็มของทศนิยมค่านั้นที่ทำให้ฟังก์ชันการประเมินให้ค่าที่ดีกว่าออกมา ตัวอย่างเช่นถ้าฟังก์ชันการประเมินมีลักษณะเป็นดังสมการที่ 3.2

$$f(x_1, x_2) = x_1^2 + x_2^2$$ สมการที่ 3.2

โดยมีเงื่อนไขคือ

- x_1 และ x_2 เป็นพารามิเตอร์ในมิติที่ 1 และ 2 ของฟังก์ชันการประเมินนี้ตามลำดับ
- ในวงรอบปัจจุบันของการทำงานเป็นจะเป็นการสุ่มเปลี่ยนค่าของพารามิเตอร์ในมิติที่ 2
- ค่าของสถานะปัจจุบันคือ $x_1 = 1$, $x_2 = 2$
- ค่าของ x_2 ที่ถูกสุ่มออกมาใหม่มีค่าเป็น 1.489

จากเงื่อนไขดังกล่าวส่วนของการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบของโครงร่างจะทำการทดลองค่าจำนวนเต็ม 2 ค่าที่มีค่าใกล้เคียงกับ 1.489 คือค่าขอบบนซึ่งเท่ากับ 2 และค่าขอบล่างซึ่งเท่ากับ 1 จากนั้นทำการแทนค่าทั้งสองลงไปในฟังก์ชันการประเมินได้เป็น

$$f(1,2) = 5 \text{ และ } f(1,1) = 2$$

จะเห็นว่าค่าขอบบนคือ 1 จะให้ค่าฟังก์ชันการประเมินที่ดีกว่า ดังนั้นส่วนของการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบจะทำการเลือกค่า 1 มากำหนดให้กับ x_2 ซึ่งวิธีการนี้เป็นวิธีการหาค่าเหมาะสมเฉพาะแห่งวิธีหนึ่ง

3.2.4 วิธีที่เลือกใช้ในการสุ่มหาเวกเตอร์ของการหาค่าเหมาะสมเฉพาะแห่ง

จากหัวข้อ 2.1.6 ที่แสดงถึงรายละเอียดของซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งที่จะต้องมีฟังก์ชัน ***local_optimize*** ที่รับเวกเตอร์ 1 ตัวเป็นอินพุท และมีหน้าที่สุ่มหาเวกเตอร์ใหม่ที่ขนาดเท่ากับเวกเตอร์ที่เป็นอินพุทแต่มีทิศทางที่ต่างกัน ผู้จัดทำวิทยานิพนธ์ได้เลือกใช้ ระบบพิกัดไฮเปอร์สเฟียริคอล ที่ได้แสดงรายละเอียดไว้ในหัวข้อ 2.1.9 มาเป็นเครื่องมือในการสุ่มทิศทางของเวกเตอร์ใหม่ โดยที่จะเลือกสุ่มหามุม θ ทุกๆ มุมเมื่อเสร็จแล้ว จากนั้นก็จะทำการแปลงพิกัดพิกัดไฮเปอร์สเฟียริคอล ที่ได้ไปเป็นพิกัดคาร์ทีเซียน ด้วยวิธีที่แสดงไว้ในหัวข้อ 2.1.9

การที่เวกเตอร์ตัวใหม่ถูกสุ่มออกมาด้วยวิธีการดังที่ได้กล่าวไปแล้วนั้น โอกาสที่คอมโพเนนท์ของเวกเตอร์ตัวใดตัวหนึ่งจะมีค่าสูง จนทำให้เมื่อนำไปบวกกับสถานะปัจจุบันดังรูปที่ 2.7 แล้วทำให้พารามิเตอร์ของสถานะใหม่ที่ได้บางตัวมีค่าเกินช่วงค่าของตนเอง ดังนั้นส่วนของการทำการหาค่าเหมาะสมเฉพาะแห่งนี้จะต้องตรวจสอบความผิดพลาดในกรณีนี้อยู่ตลอดเวลา โดยหากพบว่า ค่าพารามิเตอร์ตัวใดๆ ของสถานะที่กำลังจะถูกใส่เป็นอินพุทให้กับฟังก์ชันการประเมินมีค่าที่เกินขอบเขตช่วงของตนเองแล้ว จะต้องสั่งให้ฟังก์ชันการประเมินให้ค่ากลับมาเป็นค่าที่ไม่ดีที่สุดเท่าที่จะเป็นไปได้ ตัวอย่างเช่น จากสมการที่ 3.2 ถ้ากำหนดเงื่อนไขคือ x_1 และ x_2 มีค่าเป็นได้

ตั้งแต่ 1 ถึง 3 แต่หลังจากเอาเวกเตอร์ที่ได้มาใหม่มาบวกกับสถานะเดิมแล้ว สถานะใหม่ที่ได้แสดงได้เป็น $x_1 = 6$, $x_2 = 1$ พบว่าค่าของ x_1 มีค่ามากกว่า 3 ซึ่งเป็นค่าที่มากที่สุดที่ x_1 จะเป็นไปได้ ดังนั้นฟังก์ชันการประเมินจะต้องให้ค่ากลับมาเป็นค่าที่ไม่ดีที่สุดในกรณีที่ใช้เปรียบเทียบกับค่าอื่น อย่างเช่น ∞

3.2.5 เงื่อนไขการจบการทำงานของวิธีซีมูเล็ทเต็ดแอนนิลลิงที่เลือกใช้

เงื่อนไขมาตรฐานสำหรับการจบการทำงานของวิธีซีมูเล็ทเต็ดแอนนิลลิงที่พัฒนาขึ้นเลือกใช้คือ จะหยุดทำงานเมื่อได้ทำการลดค่าของอุณหภูมิของการยอมรับของวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิงจนถึงค่าที่ต่ำมากค่าหนึ่ง ซึ่งในที่นี้คือ 4.94066e-324 ซึ่งเป็นค่าจำนวนบวกที่เล็กที่สุดของ .NET Framework ที่ถูกใช้ในการพัฒนาโครงร่าง

นอกจากนั้นโครงร่างนี้ยังเปิดโอกาสให้ผู้ใช้กำหนดเงื่อนไขการทำงานเพิ่มเติมจากเงื่อนไขมาตรฐานที่ได้กล่าวไปแล้วข้างต้น โดยที่เงื่อนไขที่ว่านั้นคือ

- จำนวนวนรอบการทำงานที่มากที่สุด
- จำนวนการยอมรับสถานะใหม่มากที่สุด

โดยเงื่อนไขทั้งสองนี้ผู้ใช้สามารถกำหนดได้ภายในไฟล์คอนฟิกูเรชัน (Configuration File) ภายนอก ดังที่จะได้แสดงในภาคผนวก ก โดยที่เงื่อนไขทั้งสองสามารถถูกกำหนดเป็นจำนวนบวกเพื่อที่โครงร่างจะได้นำค่าเหล่านั้นไปใช้หรือถ้าถูกกำหนดเป็นจำนวนลบโครงร่างจะไม่นำเงื่อนไขทั้งสองมาพิจารณา

3.3 ตัวอย่างกระบวนวิธีในการเปลี่ยนแปลงโมเดลการออกแบบ

หัวข้อนี้จะแสดงตัวอย่างการเปลี่ยนรูปร่างของโมเดลการออกแบบในระหว่างกระบวนการทำงานของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุที่พัฒนาขึ้น โดยมีโมเดลการออกแบบตั้งต้นดังแสดงในรูปที่ 3.2 และรูปที่ 3.3 โดยจะเลือกใช้ฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบ

โดยวัตถุประสงค์ของการแสดงตัวอย่างในหัวข้อนี้คือ แสดงให้เห็นถึงรูปแบบของโมเดลการออกแบบที่ฟังก์ชันการประเมินที่อยู่ในรูปของปลั๊กอินจะได้รับไปในการวนรอบแต่ละรอบ โดยจะแสดงให้ดู 3 รอบการทำงานโดยมีอาเรย์ตั้งต้นเป็นดังรูป 3.16

2	3	1

รูปที่ 3.16 อาเรย์ตั้งต้นของการทำงาน

กำหนดให้สมาชิกของอาเรย์ในช่องที่ 1, 2 และ 3 แสดงค่าหมายเลขคลาสที่เป็นเจ้าของของ เมทธอด methodA1, methodB1 และ attributeC1 ตามลำดับ

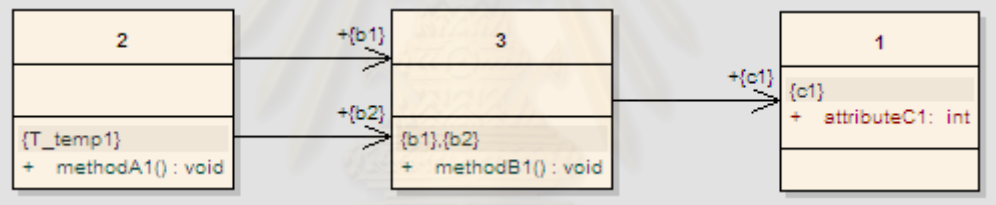
การเข้าเรียกใช้ฟังก์ชันการประเมินรอบที่1

- รูปแบบของสถานะที่เข้ามาเป็นอินพุทแสดงดังรูปที่ 3.17

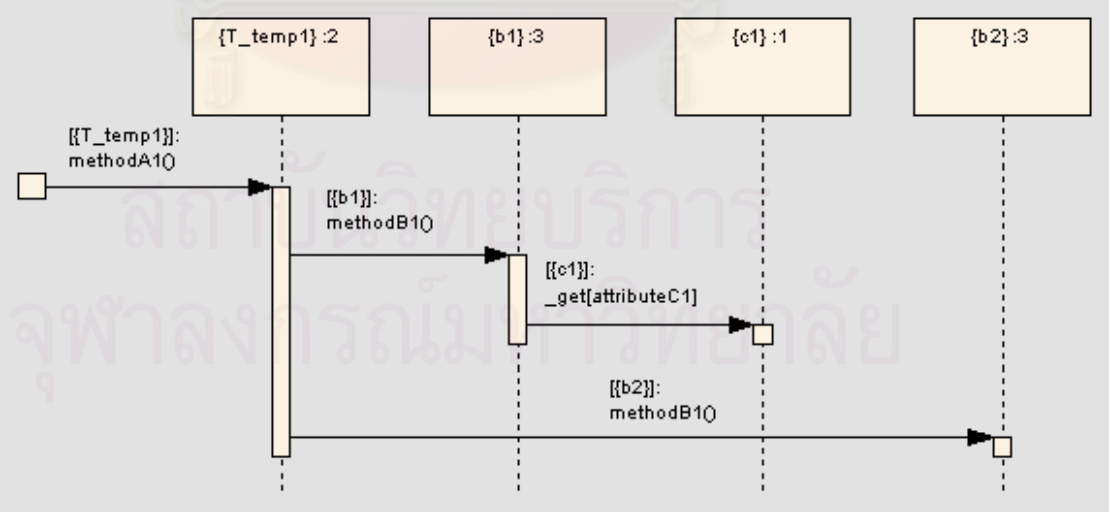
2	3	1

รูปที่ 3.17 รูปแบบของสถานะที่เข้ามาเป็นอินพุท

- โมเดลยูเอ็มแอลที่สร้างได้จากสถานะนี้แสดงดังรูปที่ 3.18 และ 3.19



รูปที่ 3.18 แผนภาพคลาสหลังจากการทำงานรอบที่ 1



รูปที่ 3.19 แผนภาพซีเควนซ์หลังจากการทำงานรอบที่ 1

- ค่าที่ฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบคำนวณได้จากโมเดลการออกแบบนี้เท่ากับ 0.7 ซึ่งเป็นค่าประเมินของโมเดลต้นแบบ

การเข้าเรียกใช้ฟังก์ชันการประเมินรอบที่2

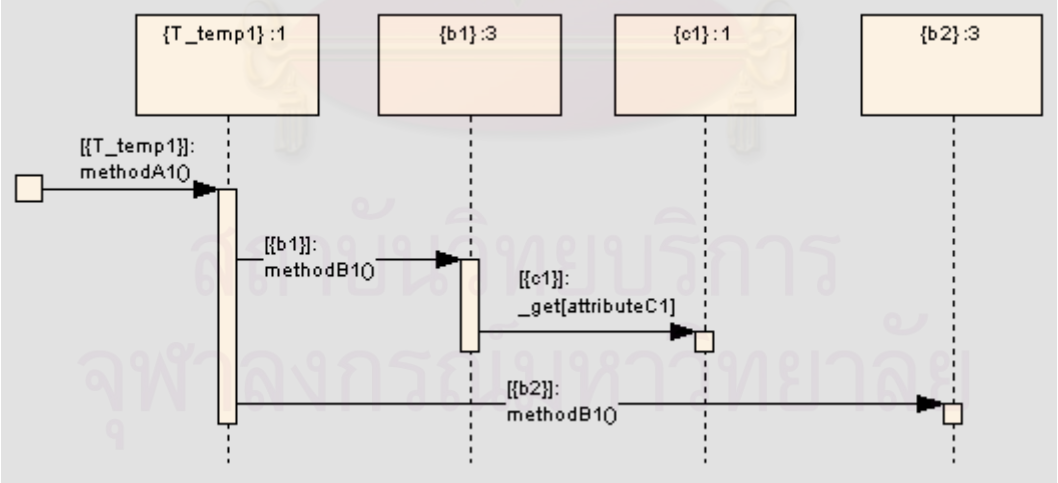
- รูปแบบของสถานะที่เข้ามาเป็นอินพุทแสดงดังรูปที่ 3.20 เกิดจากการสุ่มเปลี่ยนสมาชิกของอาเรย์ในช่องที่ 1

1	3	1

รูปที่ 3.20 อาเรย์ตั้งต้นของการทำงานรอบที่ 2

- โมเดลยูเอ็มแอลที่สร้างได้จากสถานะนี้แสดงดังรูปที่ 3.21 และ 3.22

รูปที่ 3.21 แผนภาพคลาสหลังจากการทำงานรอบที่ 2



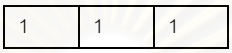
รูปที่ 3.22 แผนภาพซีเควนซ์หลังจากการทำงานรอบที่ 2

- ค่าที่ฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบคำนวณได้จากโมเดลการออกแบบนี้เท่ากับ 1.1 ซึ่งเป็นค่าประเมินที่ไม่

ดีเท่ากับค่าประเมินของโมเดลต้นแบบ แต่สมมุติให้สถานะในรูปแบบนี้ถูกยอมรับจากวิธีซีมูเล็ทเต็ดแอนนิลลิง

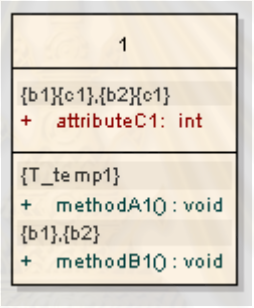
การเข้าเรียกใช้ฟังก์ชันการประเมินรอบที่3

- รูปแบบของสถานะที่เข้ามาเป็นอินพุทแสดงดังรูปที่ 3.23 เกิดจากการสุ่มเปลี่ยนสมาชิกของอาเรย์ในช่องที่ 2

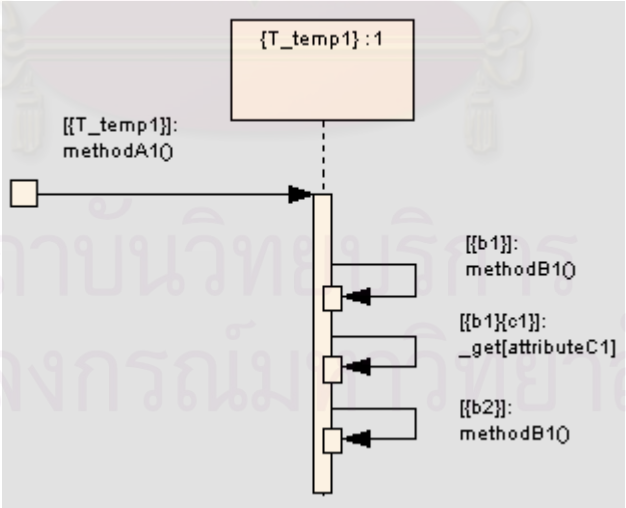


รูปที่ 3.23 อาเรย์ตั้งต้นของการทำงานรอบที่ 3

- โมเดลยูเอ็มแอลที่สร้างได้จากสถานะนี้แสดงดังรูปที่ 3.24 และ 3.25



รูปที่ 3.24 แผนภาพคลาสหลังจากการทำงานรอบที่ 3



รูปที่ 3.25 แผนภาพซีเควนซ์หลังจากการทำงานรอบที่ 3

- ค่าที่ฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบคำนวณได้จากโมเดลการออกแบบนี้เท่ากับ 0.6 ซึ่งเป็นค่าประเมินที่ดี

มากกว่าค่าประเมินของโมเดลต้นแบบ และเป็นค่าที่ดีที่สุดที่ได้จากโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุที่ได้พัฒนาขึ้นอีกด้วย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

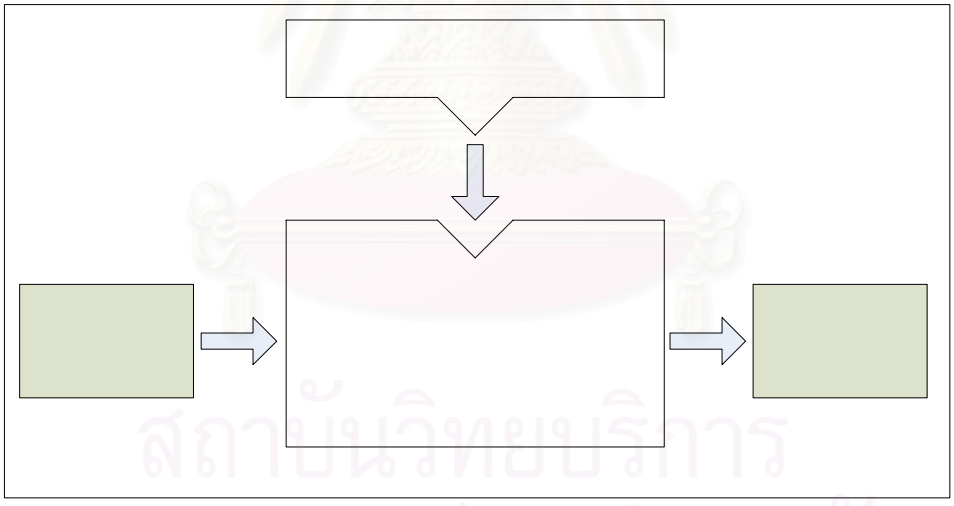
บทที่ 4

โครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

เนื้อหาในบทนี้จะเป็นการอธิบายถึงองค์ประกอบด้านสถาปัตยกรรมของโครงร่างที่ได้พัฒนาขึ้น รวมไปถึงวิธีการปรับแต่งค่าจากไฟล์คอนฟิกูเรชัน และสภาพแวดล้อมในการพัฒนาโปรแกรมประยุกต์ โดยรายละเอียดของอินเทอร์เฟซ (Interface) ของโครงร่างจะถูกอธิบายอีกทีในภาคผนวก ข

4.1 โครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

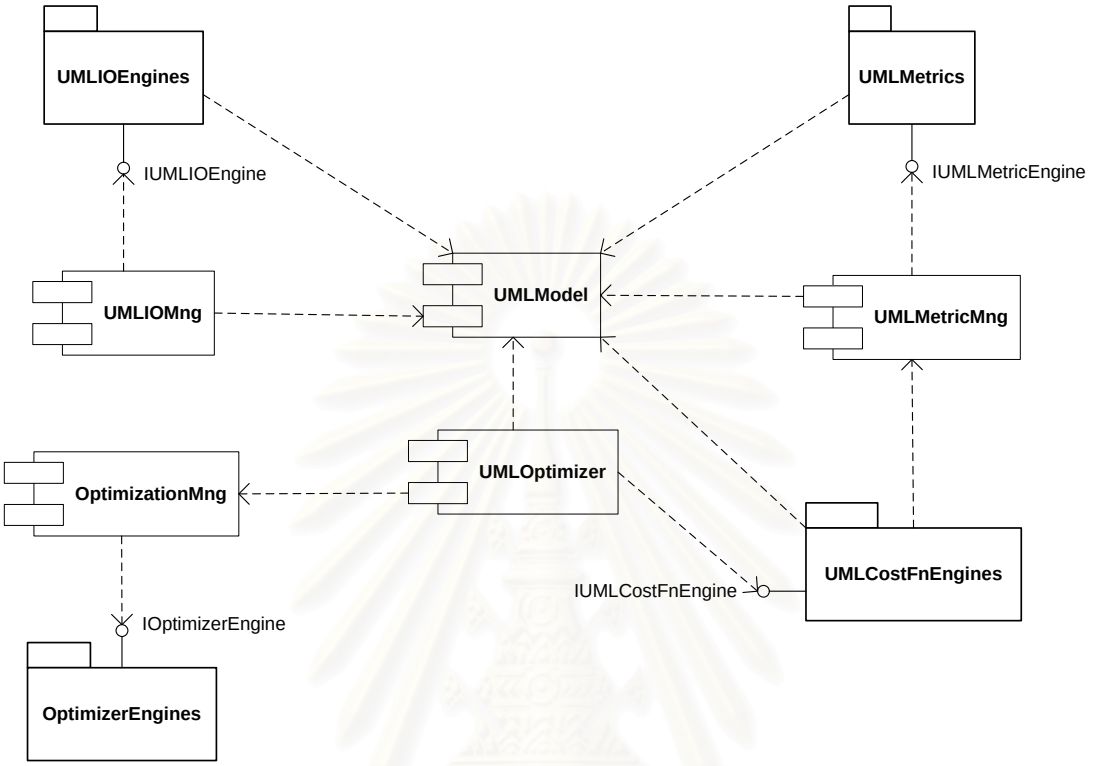
หน้าที่ของโครงร่างที่พัฒนาขึ้นนี้คือ การปรับปรุงโมเดลการออกแบบที่อธิบายด้วยภาษายูเอ็มแอลให้มีคุณภาพในด้านที่สนใจให้ดีขึ้น โดยที่มาตรวัดของคุณภาพเหล่านี้สามารถพัฒนาขึ้นเป็นปลั๊กอินต่างๆ ไว้สำหรับใช้งานกับโครงร่าง



รูปที่ 4.1 ภาพรวมการทำงานของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

รูปที่ 4.1 อธิบายถึงภาพรวมของการทำงานของโครงร่างนี้ โดยโครงร่างจะนำเอาโมเดลยูเอ็มแอลที่มีลักษณะดังที่ได้อธิบายไปในหัวข้อ 3.1.1 โครงร่างจะทำการเปลี่ยนแปลงโครงสร้างของโมเดลการออกแบบนั้นด้วยวิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่ง เพื่อ

ปรับปรุงคุณภาพในด้านที่สนใจ โดยการประเมินปริมาณของคุณภาพของโมเดลการออกแบบในการเปลี่ยนแปลงโครงสร้างของโมเดลแต่ละรอบจะกระทำโดยปลั๊กอินการปรับปรุงที่เลือกใช้



รูปที่ 4.2 ส่วนประกอบของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

รูปที่ 4.2 แสดงส่วนประกอบของโครงร่างที่ได้พัฒนาขึ้นมาโดยที่ส่วนต่างๆ มีรายละเอียดดังนี้

4.1.1 UMLModel

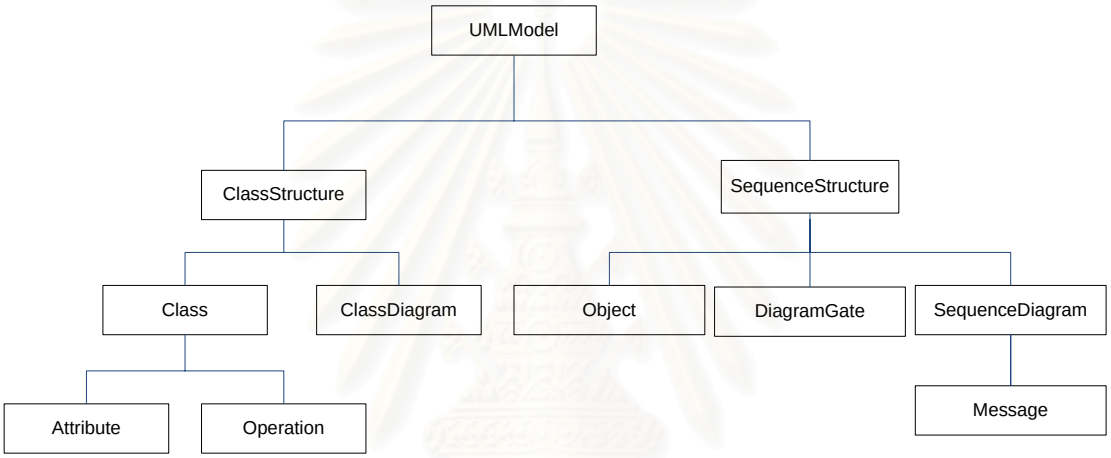
UMLModel เป็นส่วนของโครงสร้างข้อมูลกลางที่ใช้ส่งผ่านกันระหว่างแต่ละคอมโพเนนท์ในโครงสร้าง โดยที่ UMLModel จะทำหน้าที่เป็นตัวแทนของโมเดลยูเอ็มแอลที่นำเข้าเป็นอินพุทของโครงร่าง ออบเจ็กต์โมเดลของคอมโพเนนท์นี้ได้อธิบายไว้ดังรูปที่ 4.3 จากรูปจะเห็นได้ว่า UMLModel มีโครงสร้างและความสัมพันธ์ระหว่างแต่ละส่วนย่อยคล้ายกับส่วนประกอบต่างๆ ในภาษายูเอ็มแอล

ประโยชน์หลักที่โครงร่างจะได้จาก UMLModel คือการได้ใช้ออบเจ็กต์ของโมเดลยูเอ็มแอลที่เป็นกลางโดยไม่ต้องคำนึงถึงส่วนที่แตกต่างกันของรูปแบบของโมเดลยูเอ็มแอลที่ผลิต

ออกมาจากเครื่องมือที่ใช้ในการสร้างโมเดลการออกแบบจากค่ายที่แตกต่างกัน เช่น Rational Rose หรือ Sparx Enterprise Architect เป็นต้น

4.1.2 UMLIOMng

คอมโพเนนท์นี้เป็นผู้ทำหน้าที่ควบคุมการแปลงโมเดลยูเอ็มแอลที่มีรูปแบบเฉพาะตัวที่ได้จากเครื่องมือต่างๆ ดังที่ได้ยกตัวอย่างไว้ในหัวข้อที่ 4.1.1 ให้กลายเป็นโมเดลยูเอ็มแอลแบบที่โครงร่างจะทำงานด้วยได้ ซึ่งก็คือ UMLModel ในหัวข้อที่ 4.1.1 นั่นเอง



รูปที่ 4.3 ลำดับชั้นความเป็นองค์ประกอบกันของออบเจ็กต์ต่างๆ ของคอมโพเนนท์ UMLModel

4.1.3 UMLIOEngines

เป็นแพคเกจของปลั๊กอินที่ถูกควบคุมการทำงานจาก UMLIOMng ผ่านทางอินเทอร์เฟส (Interface) IUMLIOEngine โดยที่ปลั๊กอินเหล่านี้ทำหน้าที่ในการแปลงโมเดลยูเอ็มแอลที่สร้างมาจากเครื่องมือต่างๆ ให้อยู่ในรูปแบบของ UMLModel

ปลั๊กอินที่พัฒนาขึ้นมาเพื่อใช้ในการทดลองของงานวิจัยนี้ ชื่อว่า EA (Enterprise Architect) ทำหน้าที่แปลงโมเดลยูเอ็มแอลที่ถูกผลิตโดยเครื่องมือ Sparx Enterprise Architect ให้กลายเป็น UMLModel

4.1.4 OptimizationMng

คอมโพเนนท์นี้เป็นผู้ทำหน้าที่ควบคุมการหารูปแบบที่เหมาะสมด้วยวิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่ สำหรับโมเดลการออกแบบที่เป็นอินพุทของโครงร่าง

4.1.5 OptimizerEngines

เป็นแพคเกจของปลั๊กอินที่ถูกควบคุมการทำงานจาก OptimizationMng ผ่านทางอินเทอร์เฟส IOptimizerEngine โดยที่ปลั๊กอินเหล่านี้จะกระทำการสุ่มรูปแบบที่ถูกเปลี่ยนแปลงไปต่างๆ ด้วยวิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่ของโมเดลการออกแบบขึ้นมา จากนั้นจะได้นำรูปแบบของโมเดลนั้นไปประเมินค่าคุณภาพด้วยฟังก์ชันการประเมิน และทำอย่างนี้ไปเรื่อยๆ จนกว่าจะได้รูปแบบของโมเดลที่ดีที่สุดที่เป็นไปได้

ปลั๊กอินที่พัฒนาขึ้นมาเพื่อใช้ในการทดลองของงานวิจัยนี้ ชื่อว่า SALO (Simulated Annealing with Local Optimization) ซึ่งเป็นการนำเอาวิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งมาพัฒนาเพื่อใช้งานจริง

4.1.6 UMLOptimizer

เป็นส่วนควบคุมระดับบนสุดในโครงสร้าง ทำหน้าที่สั่งการและคอยเชื่อมต่อให้ส่วนอื่นๆ ในโครงร่างทำงานร่วมกันได้อย่างราบรื่น

4.1.7 UMLCostFnEngines

เป็นแพคเกจของปลั๊กอินที่ใช้ในการประเมินคุณภาพของโมเดลการออกแบบ ปลั๊กอินเหล่านี้ถูกควบคุมการทำงานจาก UMLOptimizer ผ่านทางอินเทอร์เฟส IUMLCostFnEngine และปลั๊กอินเหล่านี้จะทำหน้าที่เป็นฟังก์ชันการประเมินของ OptimizerEngine ตัวหนึ่งๆ

ปลั๊กอินที่พัฒนาขึ้นมาเพื่อใช้ในการทดลองของงานวิจัยนี้ มีอยู่ 3 ตัวด้วยกันคือ

- FnOptimizeFeatureEnvy ทำหน้าที่เป็นฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบ
- FnOptimizeMiddleMan ทำหน้าที่เป็นฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบ
- FnOptimizeCouplingCohesion ทำหน้าที่เป็นฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบ

4.1.8 UMLMetricMng

เป็นส่วนของเครื่องมือช่วยหลือเพื่อใช้ควบคุมการวัดค่าต่างจาก UMLModel ด้วยมาตรวัดต่างๆ ที่พัฒนาขึ้นโดยมีลักษณะเป็นปลั๊กอินเช่นเดียวกัน ซึ่ง UMLCostFnEngine ทั้ง 3 ตัวในหัวข้อที่แล้ว ได้ใช้บริการของคอมโพเนนท์นี้เพื่อหาค่ามาตรวัดต่างๆ ที่ต้องการ จากโมเดลการออกแบบที่อยู่ในรูปของ UMLModel

4.1.9 UMLMetrics

เป็นแพคเกจของปลั๊กอินที่ใช้ในการวัดค่ามาตรวัดต่างๆ ของโมเดลการออกแบบ ปลั๊กอินเหล่านี้ถูกควบคุมการทำงานจาก UMLMetricMng ผ่านทางอินเทอร์เฟส IUMLMetricEngine

ปลั๊กอินรูปแบบนี้ที่พัฒนาขึ้นมาเพื่อใช้ในการทดลองของงานวิจัยนี้ มีอยู่ 6 ตัวด้วยกันคือ

- COH_D ทำหน้าที่วัดค่ามาตรวัด $COH(D)$ ที่นิยามในสมการที่ 2.5
- COPA_D ทำหน้าที่วัดค่ามาตรวัด $COP_{IA}(D)$ ที่นิยามในสมการที่ 2.9
- SBC ทำหน้าที่วัดค่ามาตรวัด SBC ที่นิยามในสมการที่ 2.8
- NSMsg ทำหน้าที่วัดค่ามาตรวัด $NSMsg$ ที่อ้างถึงในอสมการที่ 2.1
- NMsg ทำหน้าที่วัดค่ามาตรวัด $NMsg$ ที่อ้างถึงในอสมการที่ 2.1
- WOD ทำหน้าที่วัดค่ามาตรวัด WOD ที่อ้างถึงในอสมการที่ 2.2

4.2 ไฟล์คอนฟิกูเรชัน

จากรูปที่ 4.4 แสดงให้เห็นว่าค่าที่ถูกปรับแต่งได้ในไฟล์คอนฟิกูเรชันมีดังนี้

- ***IsMinimizationMode:*** เป็นตัวกำหนดว่าค่าที่ดีที่สุดที่วิธีซีมูเล็ทเต็ดแอนนิลลิงกำลังค้นหาเป็นค่าที่มากหรือน้อยที่สุด
- ***InitialAcceptanceTemperature:*** เป็นตัวกำหนดอุณหภูมิเริ่มต้นของการยอมรับของวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิง
- ***AcceptanceTemperatureReductionFactor***: ค่า C ในสมการที่ 2.3 สำหรับการลดอุณหภูมิการยอมรับสถานะใหม่
- ***MaximumAcceptance:*** จำนวนวนรอบการทำงานที่มากที่สุดที่อธิบายไว้ในหัวข้อ 3.2.5

- ***MaximumTryStep:*** จำนวนการยอมรับสถานะใหม่มากที่สุดที่ถูกอธิบายไว้ในหัวข้อ 3.2.5
- ***InitialGenerationTemperature:*** เป็นตัวกำหนดอุณหภูมิเริ่มต้นของการสุ่มเลือกค่าพารามิเตอร์ใหม่สำหรับวิธีอะแดพทีฟซีมูเล็ทเต็ดแอนนิลลิง
- ***GenerationTemperatureReductionFactor:*** ค่า C ในสมการที่ 2.3 สำหรับการลดอุณหภูมิการสุ่มเลือกค่าพารามิเตอร์ใหม่

โดยค่าต่างๆ ที่ได้แสดงในรูปที่ 4.4 จะถูกใช้เป็นค่าโดยปริยายในโครงร่าง ส่วนค่าที่เหลืออีก 3 ค่าที่ไม่ได้กล่าวถึงจะอยู่นอกเหนือจากการปรับแต่งของผู้ใช้เพราะโครงร่างจะกำหนดค่าตามลักษณะของโมเดลการออกแบบที่เป็นอินพุทให้ตอนรันไทม์เสมอ

```
<SALO.Config>
        <IsMinimizationMode>true</IsMinimizationMode>
        <InitialAcceptanceTemperature>1000</InitialAcceptanceTemperature>
        <AcceptanceTemperatureReductionFactor>0.7</AcceptanceTemperatureReductionFactor>
        <MaximumAcceptance>-1</MaximumAcceptance>
        <MaximumTryStep>500000</MaximumTryStep>
        <ParameterConfigs>
                <ParameterConfig>
                        <InitialGenerationTemperature>1000</InitialGenerationTemperature>
                        <GenerationTemperatureReductionFactor>0.7</GenerationTemperatureReductionFactor>
                        <MaxValue>5</MaxValue>
                        <MinValue>-5</MinValue>
                        <IsParameterIntegerType>true</IsParameterIntegerType>
                </ParameterConfig>
        </ParameterConfigs>
</SALO.Config>
```

รูปที่ 4.4 เนื้อหาของไฟล์คอนฟิกูเรชันของวิธีซีมูเล็ทเต็ดแอนนิลลิงที่พัฒนาขึ้น

4.3 สภาพแวดล้อมในการพัฒนา

โครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ พัฒนาขึ้นมาเพื่อให้ทำงานภายใต้สภาพแวดล้อมของ.NET Framework 2.0 ซึ่งปัจจุบันถูกใช้อย่างแพร่หลายบนระบบปฏิบัติการวินโดว์ โดยใช้ภาษา C# เป็นภาษาที่ใช้เขียนรหัสต้นทางของโครงร่าง

บทที่ 5

การทดสอบการปรับปรุงคุณภาพของโมเดลการออกแบบด้วยโครงร่าง

ในบทนี้เป็นการทดสอบโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุที่พัฒนาขึ้นกับโมเดลการออกแบบที่ผู้จัดทำวิทยานิพนธ์ได้ทำการคัดเลือกมาโดยที่โมเดลการออกแบบแต่ละตัวนั้นจะมีคุณภาพที่ไม่ดีสอดคล้องกับฟังก์ชันการประเมินที่มีอยู่ โดยที่วัตถุประสงค์ของการทดลองนี้คือการพิสูจน์ว่า โครงร่างที่พัฒนาขึ้นนี้สามารถทำการปรับปรุงคุณภาพของโมเดลการออกแบบตัวอย่างได้จริง

ในการทดสอบการประยุกต์ใช้วิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่โดยทั่วไปนั้น วิธีที่นิยมทำกันคือ ทดลองประยุกต์ใช้วิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่กับโจทย์ตัวอย่างที่รู้ค่าเหมาะสมค่าจริงกันอยู่แล้ว หลังการทดลองให้นำค่าเหมาะสมจากวิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่ที่ได้มาเปรียบเทียบกับค่าเหมาะสมจริงของโจทย์ตัวอย่างว่ามีความใกล้เคียงกันมากน้อยเพียงใด ถ้าค่าทั้ง 2 ใกล้เคียงกันอย่างยอมรับได้ในบริบทที่กำลังสนใจ ก็จะถือว่าวิธีการหาค่าเหมาะสมเชิงวิธีจัด-หมู่นั้นสามารถนำไปใช้กับโจทย์ตัวอย่างอื่นๆ ได้อีก

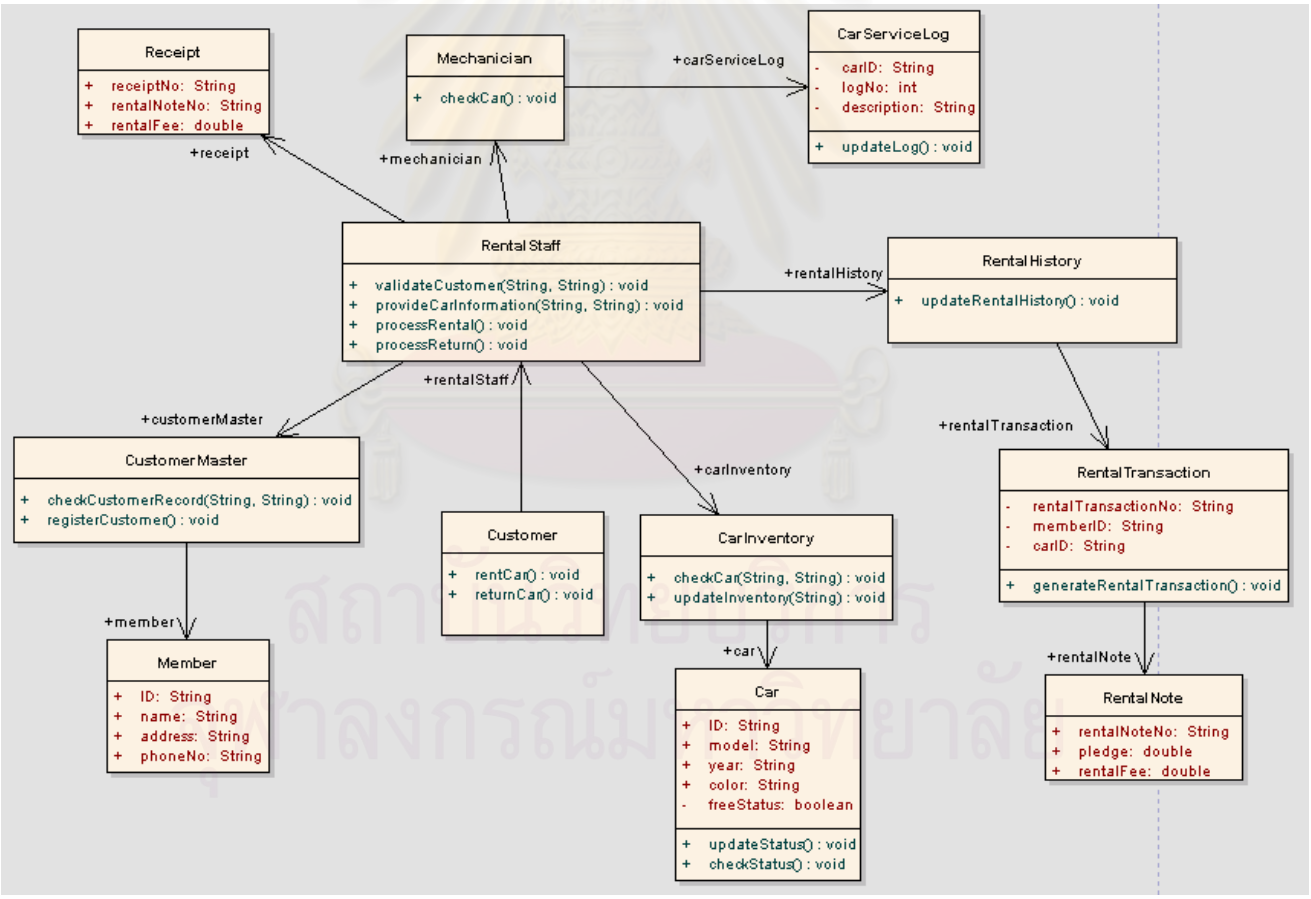
ในการทดลองนี้ผู้จัดทำวิทยานิพนธ์ได้ทำการคัดเลือกโมเดลการออกแบบของระบบตัวอย่างมา 3 ระบบโดยที่

- ระบบที่ 1 มีร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีอยู่ในระบบ จะใช้ในการทดสอบฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบ ซึ่งฟังก์ชันการประเมินนี้สามารถระบุได้ว่าค่าเหมาะสมค่าจริงของระบบที่นำมาทดลองนั้นเป็น 0 นั่นคือร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีอยู่เลย ดังนั้นการทดลองกับระบบนี้จึงเป็นการทดสอบว่าโครงร่างสามารถกำจัดร่องรอยที่ไม่ดีของระบบออกไปได้หมดหรือไม่
- ระบบที่ 2 มีร่องรอยที่ไม่ดีแบบมิดเดิลแมนอยู่ในระบบ จะใช้ในการทดสอบฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบ ซึ่งฟังก์ชันการประเมินนี้สามารถระบุได้ว่าค่าเหมาะสมค่าจริงของระบบที่นำมาทดลองนั้นเป็น 0 นั่นคือร่องรอยที่ไม่ดีแบบมิดเดิลแมนอยู่เลย ดังนั้นการทดลองกับระบบนี้จึงเป็นการทดสอบว่าโครงร่างสามารถกำจัดร่องรอยที่ไม่ดีของระบบออกไปได้หมดหรือไม่
- ระบบที่ 3 มีข้อเสียในเรื่องของการมีการเข้าคู่กันสูง จะถูกใช้ทดสอบฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบ ซึ่ง

ฟังก์ชันการประเมินนี้ไม่สามารถระบุได้ว่าค่าเหมาะสมค่าจริงของระบบที่นำมาทดลองนั้นเป็นค่าใด ดังนั้นการทดลองกับระบบนี้จึงเป็นการทดสอบว่าโครงร่างสามารถเพิ่มคุณภาพให้กับระบบได้หรือไม่

เนื่องจากด้วยลักษณะการทำงานของโครงร่าง โมเดลผลลัพธ์ที่ได้จะยังไม่ใช่โมเดลผลลัพธ์ที่สมบูรณ์ที่สามารถนำเอาไปใช้ได้เลย โดยสิ่งที่ขาดไปคือชื่อของคลาสและของออบเจ็กต์ที่สื่อความหมาย ซึ่งผู้ใช้จะต้องเป็นผู้ระบุอีกทีภายหลัง ดังนั้นการทดลองกับตัวอย่างระบบทั้งสามนี้จะนำความหมายที่อธิบายได้เป็นภาษาธรรมชาติของทั้งโมเดลต้นแบบและโมเดลผลลัพธ์ที่ได้ มาพิจารณา

5.1 การทดสอบฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบ



รูปที่ 5.1 แผนภาพคลาสต้นแบบของระบบที่ 1

การทดลองนี้เป็นการทดสอบการปรับปรุงคุณภาพของโมเดลการออกแบบด้วยโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุกับโมเดลการออกแบบระบบที่ 1 โดยใช้ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบ ซึ่งเป็นการนับจำนวนเมทธอดที่ทำให้อสมการที่ 2.1 เป็นจริง เป็นฟังก์ชันในการวัดคุณภาพ โดยค่าที่น้อยกว่าที่ได้จากฟังก์ชันนี้จะแสดงถึงคุณภาพของโมเดลการออกแบบที่ดีกว่า

ระบบที่ 1 นี้เป็นระบบการเช่ารถยนต์ โดยผู้ที่จะเช่ารถยนต์ได้จะต้องเป็นสมาชิกเท่านั้น ในการเช่ารถยนต์ สมาชิกจะติดต่อกับพนักงานเพื่อตรวจสอบว่ามีรถรุ่นที่ต้องการหรือไม่ หากมีรถตามที่ต้องการแล้ว พนักงานจะบันทึกการเช่ารถยนต์และออกเอกสารการเช่ารถยนต์ให้กับสมาชิก รวมทั้งบันทึกค่ามัดจำรถยนต์ไว้ในเอกสารการเช่ารถยนต์ด้วย หลังจากนั้น เมื่อสมาชิกนำรถยนต์มาคืน พนักงานจะตรวจสอบเอกสารการเช่ารถยนต์และคืนเงินค่ามัดจำให้กับสมาชิกพร้อมกับออกใบเสร็จรับเงินให้กับสมาชิก รวมถึงมีการแจ้งให้ช่างตรวจสอบสภาพรถ และบันทึกลงในประวัติการให้บริการของรถยนต์ด้วย

โมเดลต้นแบบของการทดลองนี้แสดงในรูปที่ 5.1 ถึงรูปที่ 5.4 โดยที่ตารางที่ 5.1 แสดงรายละเอียดค่าของมาตรวัด ***NSMsg*** และ ***NMsg*** ที่กระทำกับเมทธอดทุกๆ ตัวในโมเดลต้นแบบ และจากข้อมูลในตารางดังกล่าวจะเห็นได้ว่าค่าของฟังก์ชันการประเมินจะเท่ากับ 10 ก็คือในโมเดลต้นแบบมีเมทธอดที่มีร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีอยู่ 10 เมทธอดด้วยกันดังระบุไว้ในสดมภ์สุดท้ายของตาราง

เมื่อทำการทดลองปรับปรุงคุณภาพโมเดลการออกแบบด้วยโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุที่พัฒนาขึ้นแล้วจะได้โมเดลผลลัพธ์ดังแสดงได้ในรูปที่ 5.5 ถึงรูปที่ 5.8 โดยที่ตารางที่ 5.2 แสดงรายละเอียดค่าของมาตรวัด ***NSMsg*** และ ***NMsg*** ที่กระทำกับเมทธอดทุกๆ ตัวในโมเดลการออกแบบผลลัพธ์ และจากข้อมูลในตารางดังกล่าวจะเห็นได้ว่าค่าของฟังก์ชันการประเมินจะเท่ากับ 0

จากตารางที่ 5.1 และตารางที่ 5.2 จะเห็นได้ว่ามีค่าพิเศษอยู่สองแบบคือ NaN กับ Infinity โดยมีความหมายคือ การหาร 0 ด้วย 0 และ การหาร จำนวนใดๆ ด้วย 0 ตามลำดับ ในการทดลองนี้จะสมมติให้ NaN มีค่าเสมือนเป็น 1

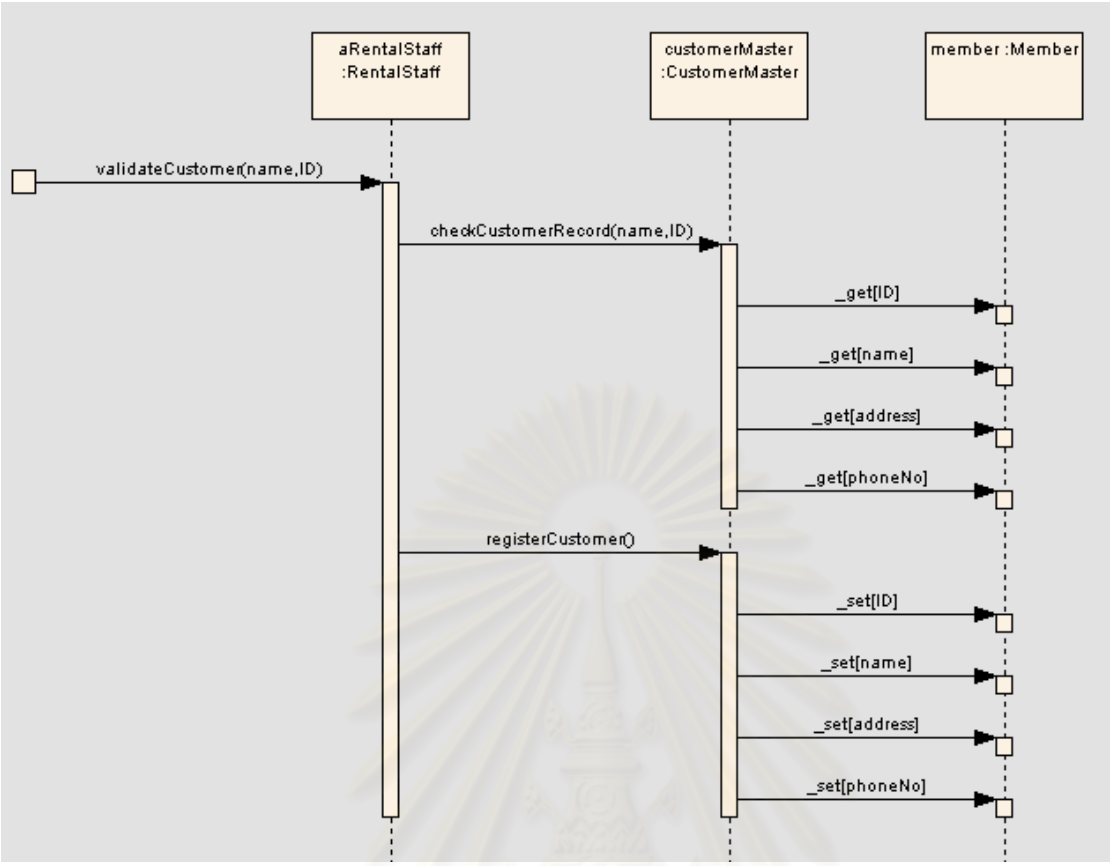
ตารางที่ 5.1 ค่ามาตรวัด *NSMsg* และ *NMsg* ของเมทธอดต่างๆ ในโมเดลต้นแบบของระบบที่ 1

Class	*Method*	*NSMsg*	*NMsg*	*NSMsg/NMsg*	*FeatureEnvy*
CarServiceLog	updateLog	0	0	NaN	FALSE
RentalHistory	updateRentalHistory	0	2	0	TRUE
Customer	rentCar	0	2	0	TRUE
Customer	returnCar	0	1	0	TRUE
Car	updateStatus	0	0	NaN	FALSE
Car	checkStatus	0	0	NaN	FALSE
RentalStaff	validateCustomer	0	2	0	TRUE
RentalStaff	provideCarInformation	0	1	0	TRUE
RentalStaff	processRental	0	1	0	TRUE
RentalStaff	processReturn	0	2	0	TRUE
RentalTransaction	generateRentalTransaction	0	0	NaN	FALSE
Mechanician	checkCar	0	1	0	TRUE
CarInventory	checkCar	0	1	0	TRUE
CarInventory	updateInventory	0	1	0	TRUE
CustomerMaster	checkCustomerRecord	0	0	NaN	FALSE
CustomerMaster	registerCustomer	0	0	NaN	FALSE

ตารางที่ 5.2 ค่ามาตรวัด *NSMsg* และ *NMsg* ของเมทธอดต่างๆ ในโมเดลที่ถูกปรับปรุงแล้วของระบบที่ 1

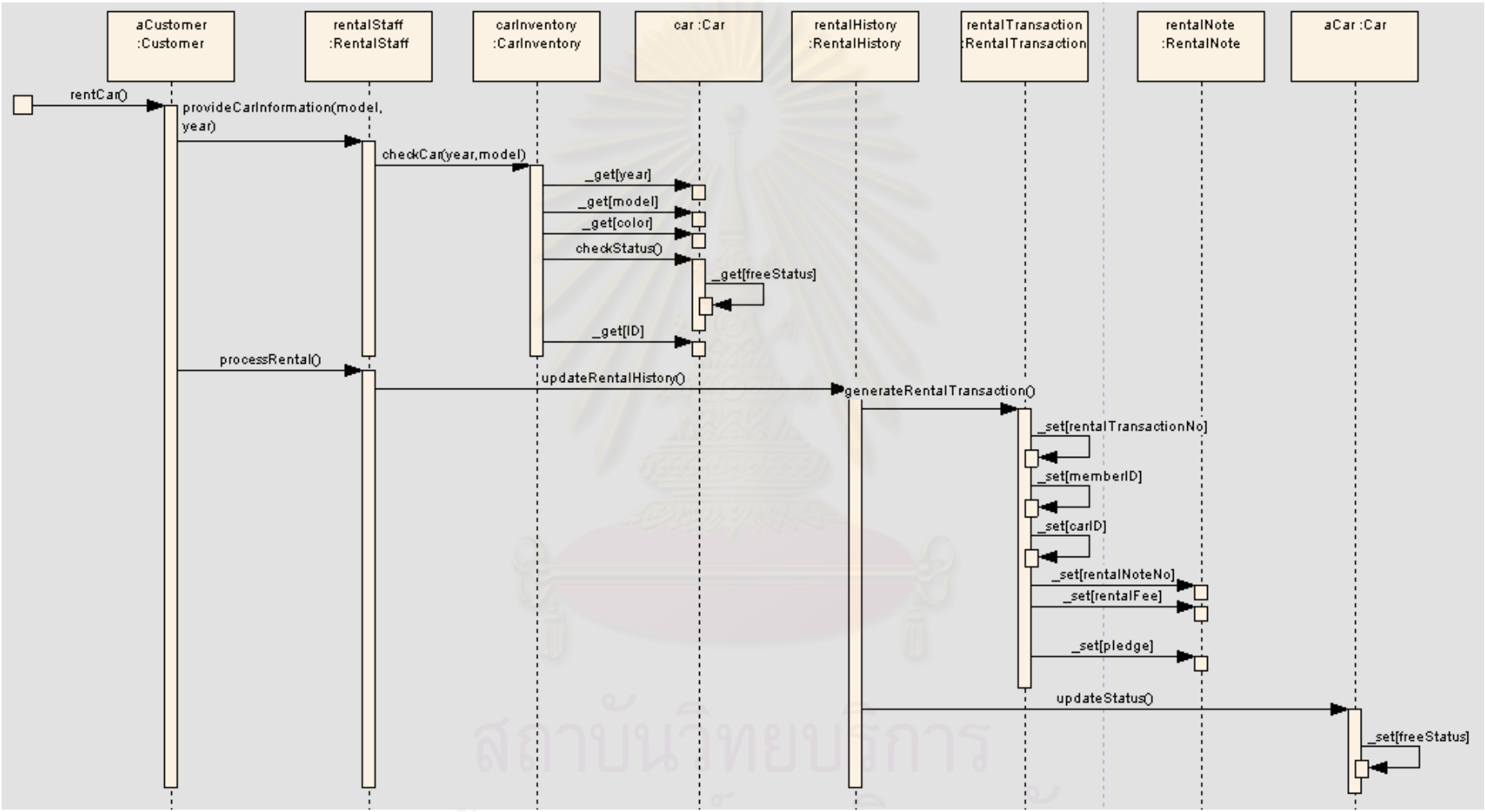
Class	*Operation*	*NSMsg*	*NMsg*	*NSMsg/NMsg*	*FeatureEnvy*
19	validateCustomer	1	1	1	FALSE
19	checkCustomerRecord	0	0	NaN	FALSE
19	provideCarInformation	1	0	Infinity	FALSE
19	checkCar	1	0	Infinity	FALSE
19	checkStatus	0	0	NaN	FALSE
21	registerCustomer	0	0	NaN	FALSE
23	rentCar	1	1	1	FALSE
23	processRental	1	0	Infinity	FALSE
23	updateRentalHistory	1	1	1	FALSE
23	updateStatus	0	0	NaN	FALSE
23	returnCar	1	0	Infinity	FALSE
23	processReturn	1	1	1	FALSE
23	updateInventory	1	0	Infinity	FALSE
26	checkCar	1	0	Infinity	FALSE
26	updateLog	0	0	NaN	FALSE
36	generateRentalTransaction	0	0	NaN	FALSE

จากการทดลองจะเห็นว่า ค่าของฟังก์ชันการประเมินที่ได้มีค่าลดลงจาก 10 เป็น 0 โดยในภาพรวมแล้วสามารถเพิ่มค่า ***NSMsg*** และ ลดค่า ***NMsg*** ของแต่ละเมทธอดในระบบได้ดังที่คาดหวัง จึงเป็นการแสดงว่าโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุสามารถปรับปรุงโมเดลการออกแบบด้วยฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบเป็นผลสำเร็จ

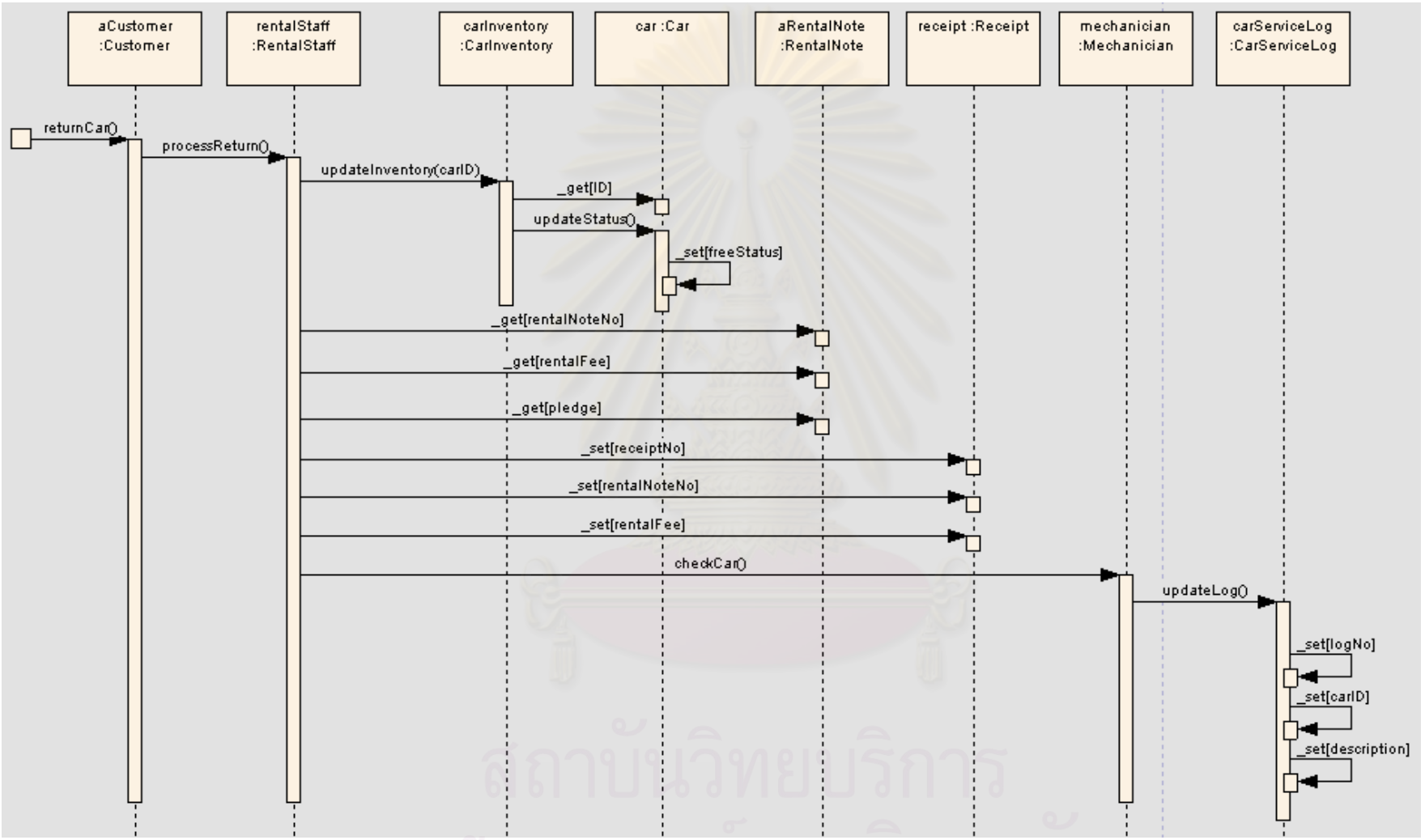


รูปที่ 5.2 แผนภาพซีเควนซ์ Register ต้นแบบของระบบที่ 1

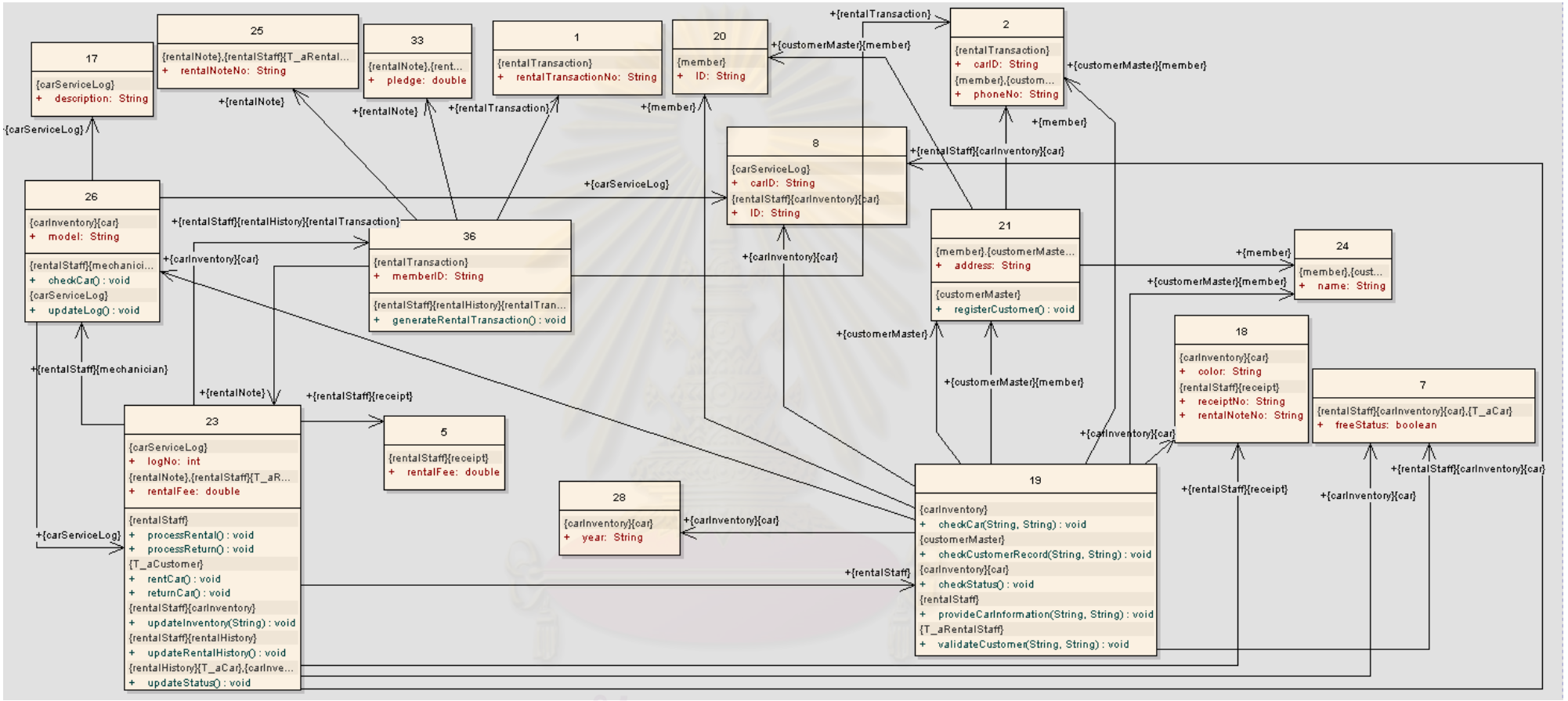
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.3 แผนภาพซีเควนซ์ RentCar ต้นแบบของระบบที่ 1

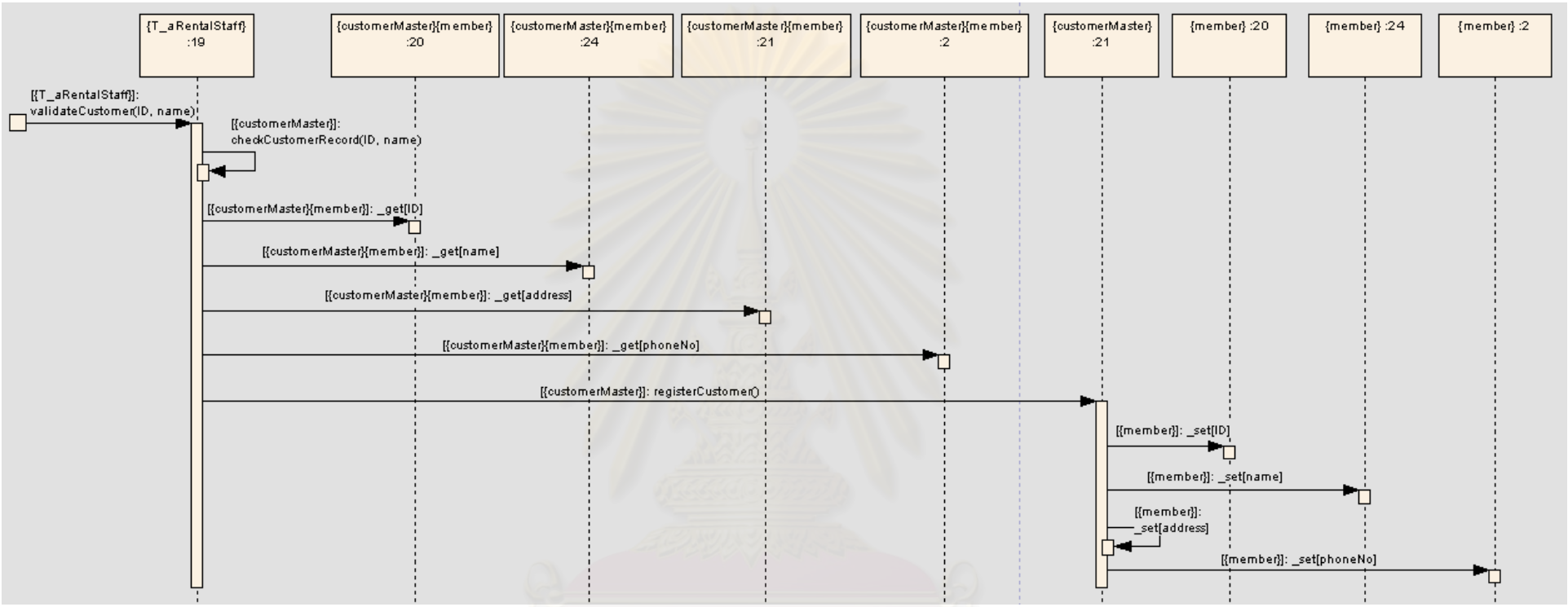


รูปที่ 5.4 แผนภาพซีเควนซ์ ReturnCar ต้นแบบของระบบที่ 1



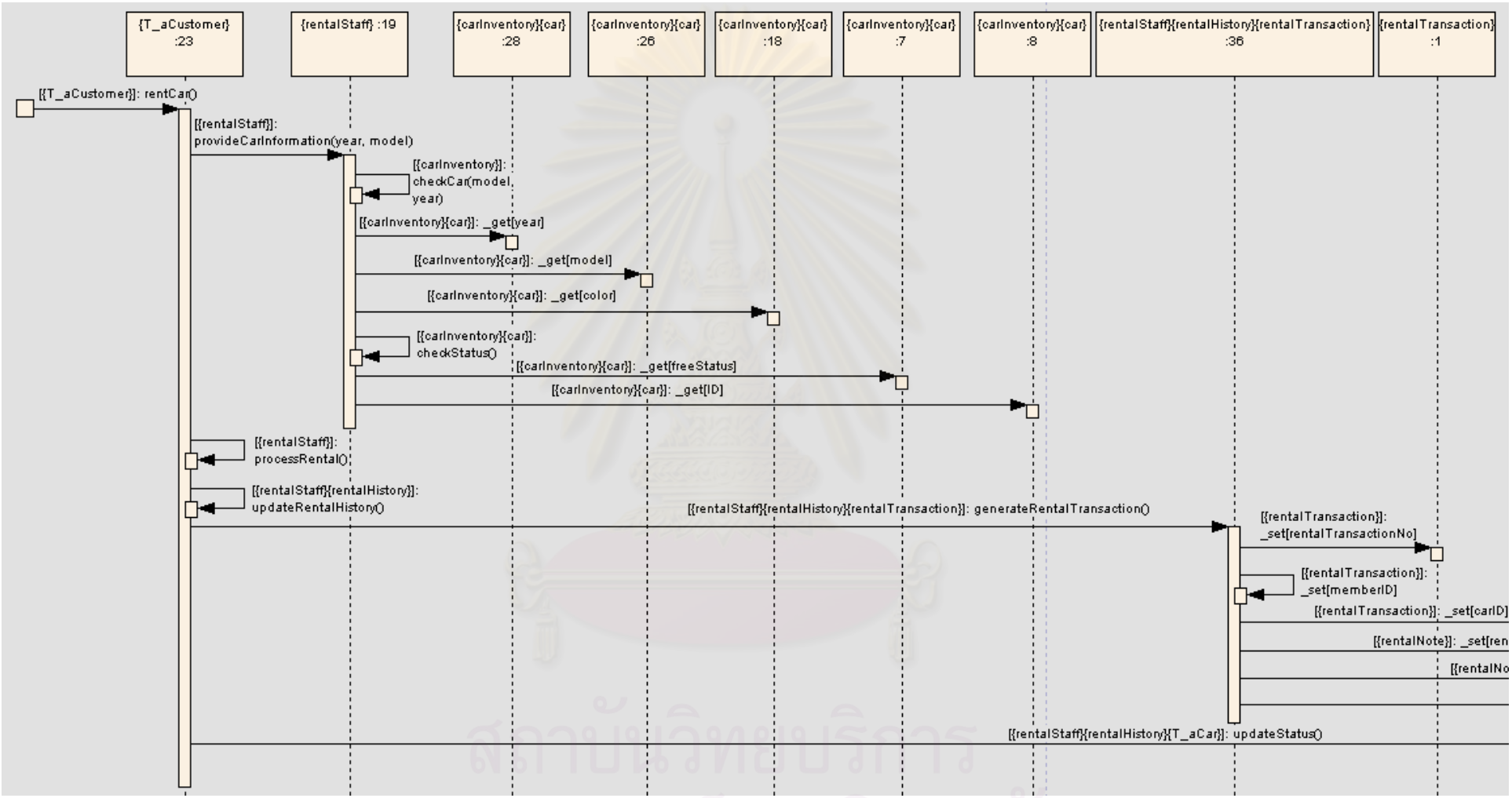
รูปที่ 5.5 แผนภาพคลาสที่ถูกปรับปรุงแล้วของระบบที่ 1

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

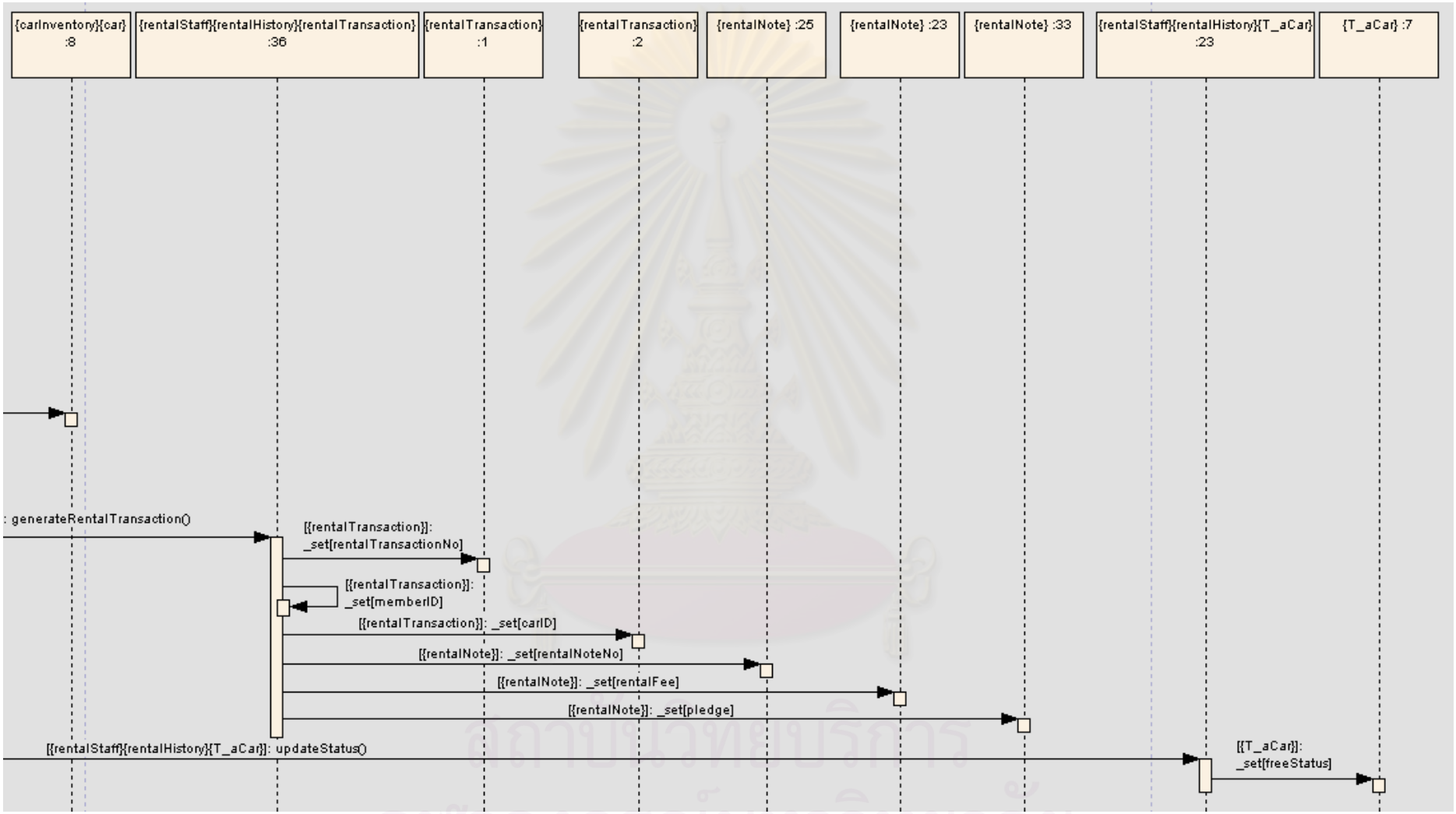


รูปที่ 5.6 แผนภาพซีเควนซ์ Register ที่ถูกปรับปรุงแล้วของระบบที่ 1

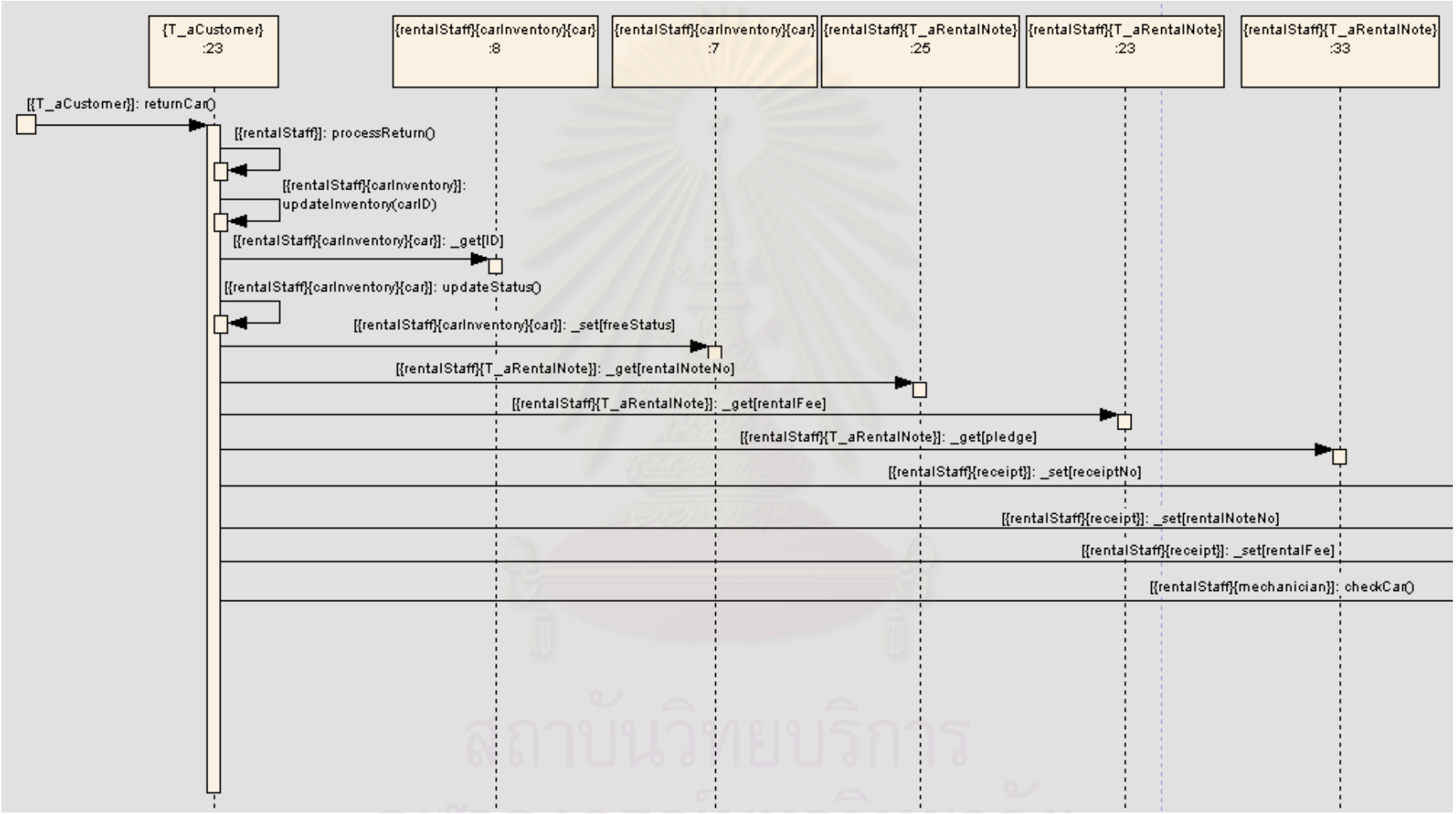
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



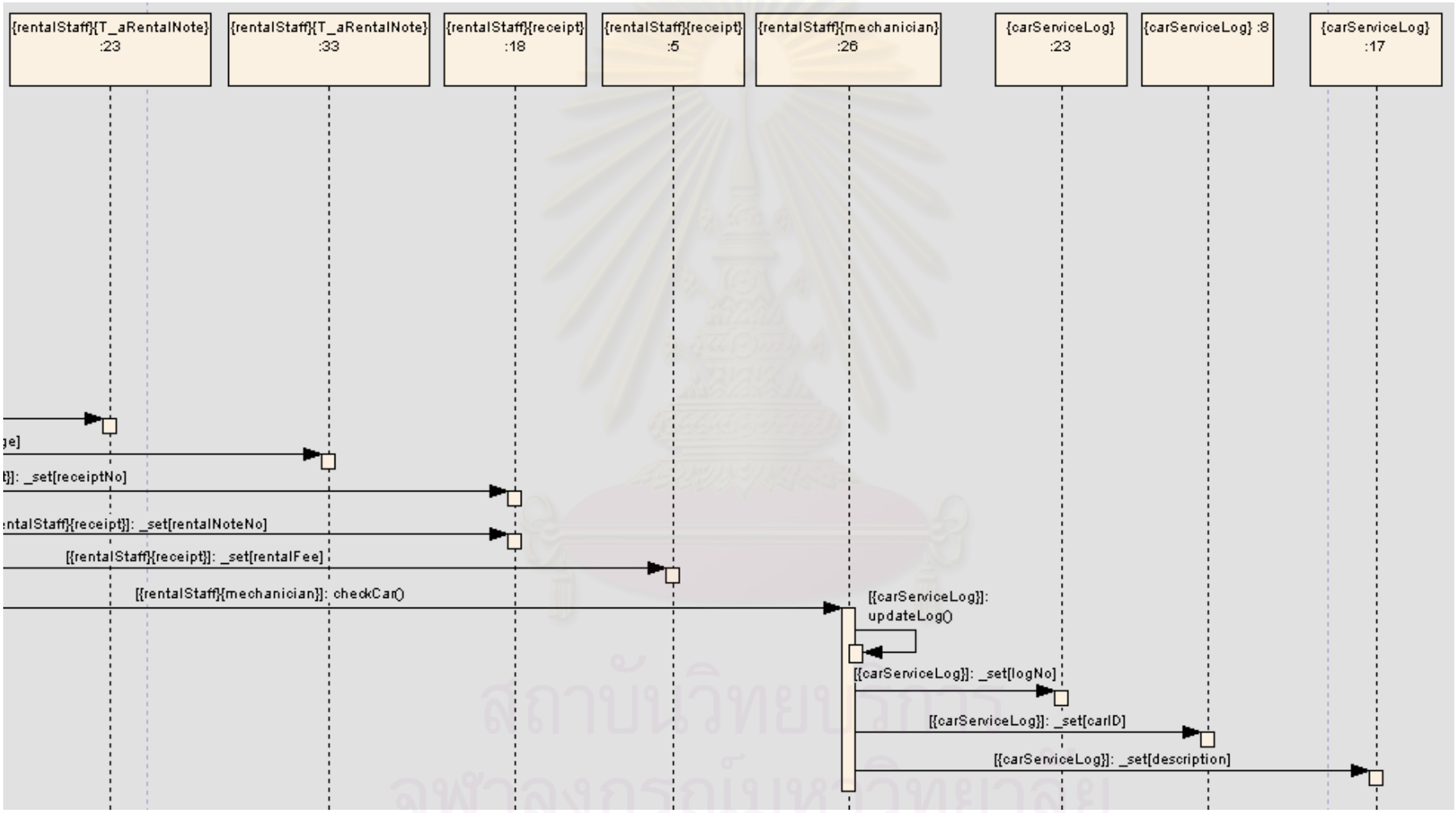
รูปที่ 5.7 แผนภาพซีเควนซ์ RentCar ที่ถูกปรับปรุงแล้วของระบบที่ 1



รูปที่ 5.7 แผนภาพซีเควนซ์ RentCar ที่ถูกปรับปรุงแล้วของระบบที่ 1 (ต่อ)

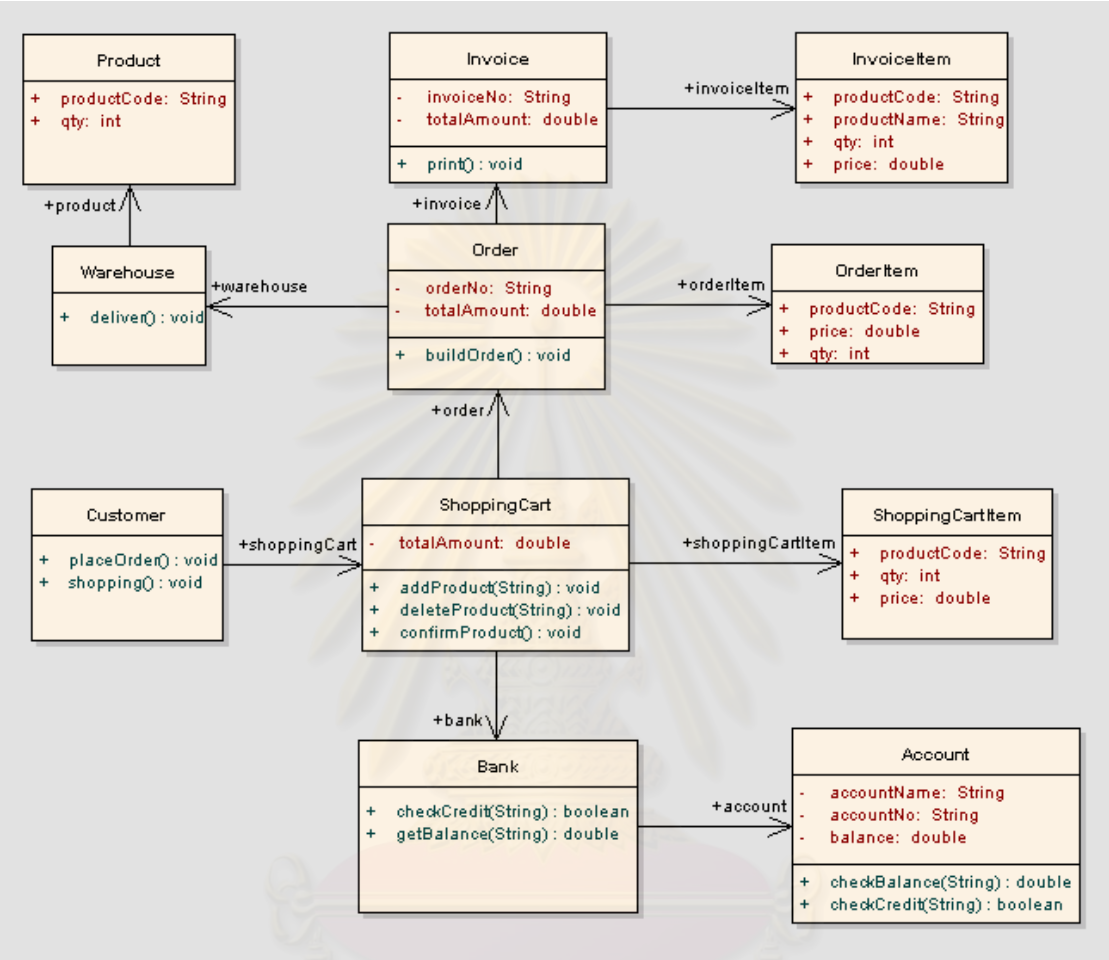


รูปที่ 5.8 แผนภาพซีเควนซ์ ReturnCar ที่ถูกปรับปรุงแล้วของระบบที่ 1



รูปที่ 5.8 แผนภาพซีเควนซ์ ReturnCar ที่ถูกปรับปรุงแล้วของระบบที่ 1 (ต่อ)

5.2 การทดสอบฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบ



รูปที่ 5.9แผนภาพคลาสต้นแบบของระบบที่ 2

การทดลองนี้เป็นการทดสอบการปรับปรุงคุณภาพของโมเดลการออกแบบด้วยโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุกับโมเดลการออกแบบระบบที่ 2 โดยใช้ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบ ซึ่งเป็นการนับจำนวนคลาสที่ทำให้อสมการที่ 2.2 เป็นจริง เป็นฟังก์ชันในการวัดคุณภาพ โดยค่าที่น้อยกว่าที่ได้จากฟังก์ชันนี้จะแสดงถึงคุณภาพของโมเดลการออกแบบที่ดีกว่า

ระบบที่ 2 นี้เป็นระบบขายสินค้าผ่านเว็บ โดยการหักบัญชีบัตรเครดิต ลูกค้าสามารถเลือกสินค้าใส่ตะกร้า และสร้างใบสั่งซื้อสินค้าผ่านระบบได้ โดยระบบจะต้องติดต่อธนาคารเพื่อ

ตรวจสอบวงเงินในบัตรเครดิตก่อน หลังจากนั้นระบบจึงจะออกใบกำกับภาษีและส่งคำสั่งไปยังคลังสินค้าเพื่อดำเนินการจัดส่งสินค้าให้กับลูกค้าต่อไป

โมเดลต้นแบบของการทดลองนี้ถูกแสดงในรูปที่ 5.9 ถึงรูปที่ 5.11 โดยที่ตารางที่ 5.3 แสดงรายละเอียดค่าของมาตรวัด *WOD* ที่กระทำกับคลาสทุกๆ ตัวในโมเดลต้นแบบ และจากข้อมูลในตารางดังกล่าวจะเห็นได้ว่าค่าของฟังก์ชันการประเมินจะเท่ากับ 2 นั่นก็คือในโมเดลต้นแบบมีคลาสที่มีร่องรอยที่ไม่ดีแบบมิดเดิลแมนอยู่ 2 คลาสด้วยกันดังระบุไว้ในสดมภ์สุดท้ายของตาราง

เมื่อทำการทดลองปรับปรุงคุณภาพโมเดลการออกแบบด้วยโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุที่พัฒนาขึ้นแล้วจะได้โมเดลผลลัพธ์ดังแสดงได้ในรูปที่ 5.12 ถึงรูปที่ 5.14 โดยที่ตารางที่ 5.4 แสดงรายละเอียดค่าของมาตรวัด *WOD* ที่กระทำกับคลาสทุกๆ ตัวในโมเดลการออกแบบผลลัพธ์ และจากข้อมูลในตารางดังกล่าวจะเห็นได้ว่าค่าของฟังก์ชันการประเมินจะเท่ากับ 0

ตารางที่ 5.3 ค่ามาตรวัด *WOD* ของคลาสต่างๆ ในโมเดลต้นแบบของระบบที่ 2

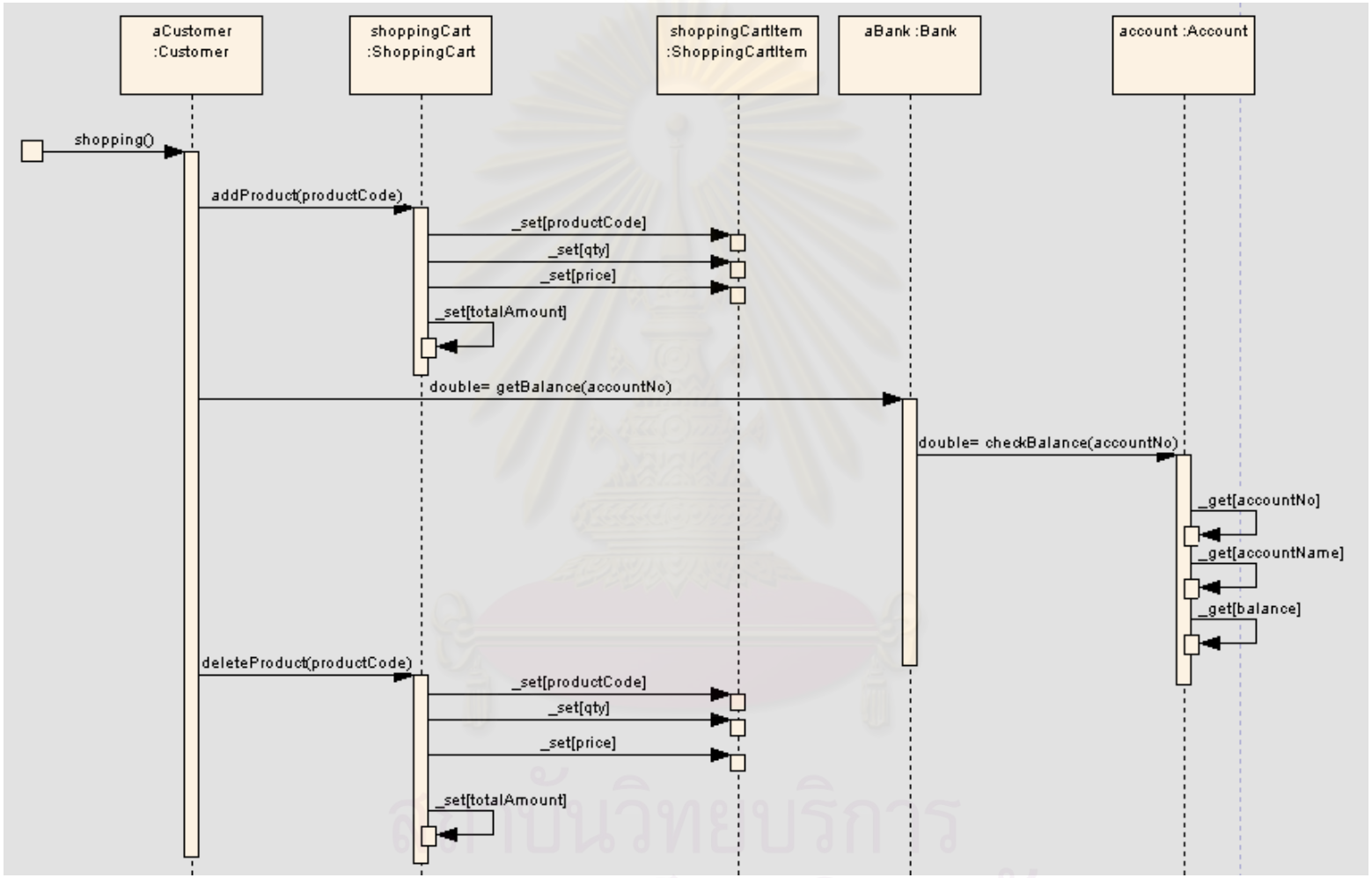
Class	***WOD***	***MiddleMan***
Product	0	FALSE
Order	0	FALSE
InvoiceItem	0	FALSE
Account	0	FALSE
OrderItem	0	FALSE
Bank	1	TRUE
Invoice	0	FALSE
ShoppingCartItem	0	FALSE
Customer	0.5	TRUE
ShoppingCart	0.333333333	FALSE
Warehouse	0	FALSE

จากการทดลองจะเห็นว่าค่าของฟังก์ชันการประเมินที่ได้มีค่าลดลงจาก 2 เป็น 0 โดยสามารถลดปริมาณ *WOD* ของแต่ละคลาสที่มีในระบบได้ จึงเป็นการแสดงว่าโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุสามารถปรับปรุงโมเดลการออกแบบด้วยฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบเป็นผลสำเร็จ

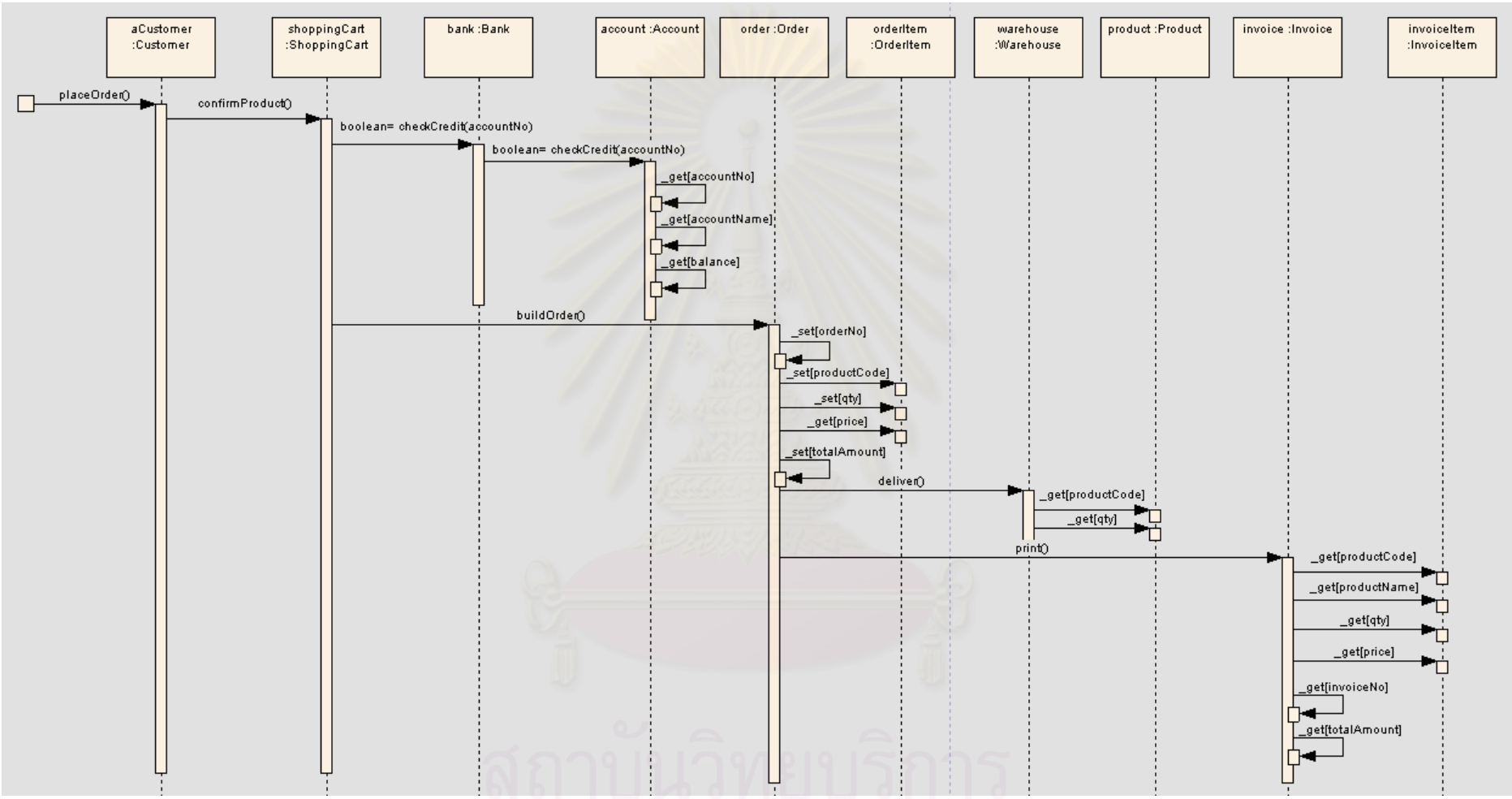
ตารางที่ 5.4 ค่ามาตรวัด *WOD* ของคลาสต่างๆ ในโมเดลที่ถูกปรับปรุงแล้วของระบบที่ 2

Class	***WOD***	***MiddleMan***
21	0	FALSE
20	0	FALSE
25	0	FALSE
13	0	FALSE
14	0	FALSE
10	0.333333333	FALSE
7	0	FALSE
15	0	FALSE
4	0	FALSE
9	0.333333333	FALSE
31	0	FALSE
11	0	FALSE
23	0	FALSE
19	0	FALSE
28	0	FALSE
3	0	FALSE
12	0	FALSE
22	0	FALSE
8	0	FALSE
26	0	FALSE

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

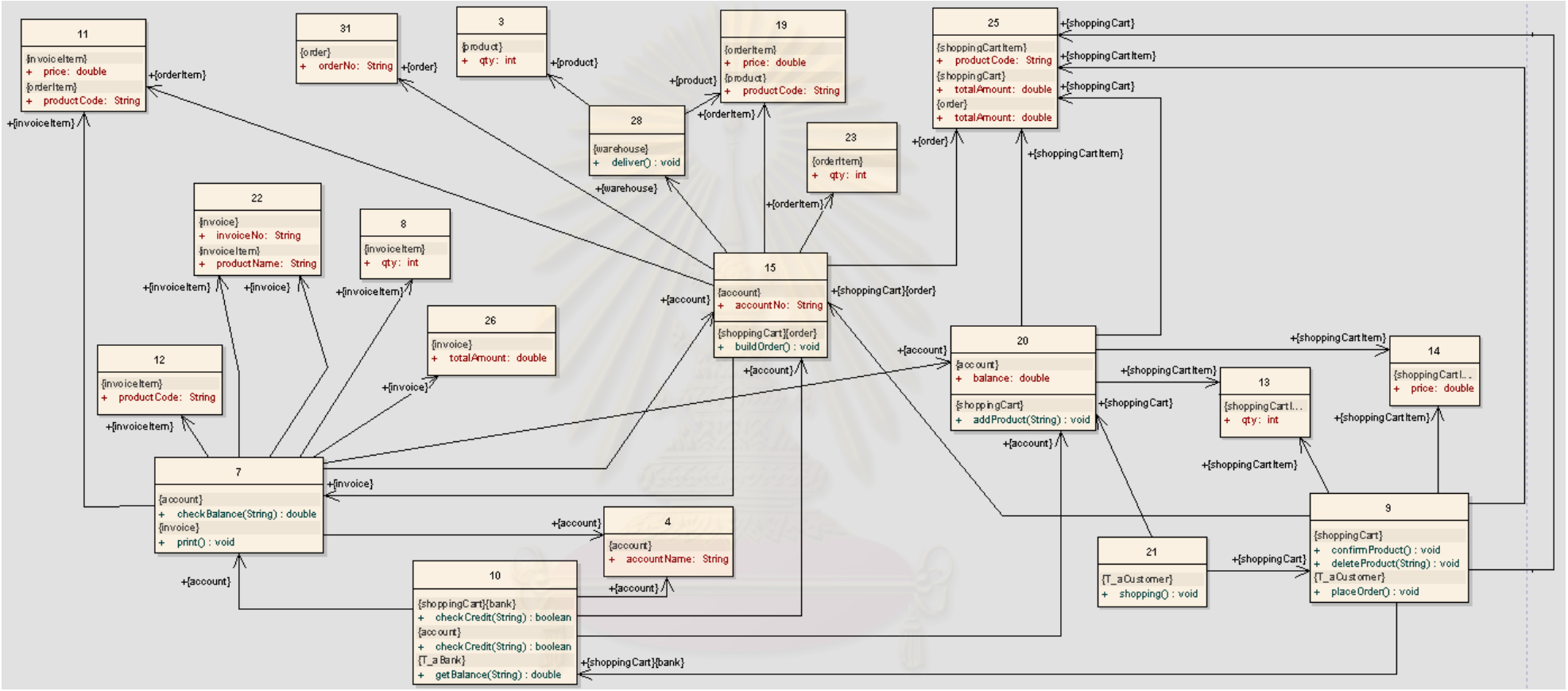


รูปที่ 5.10 แผนภาพซีเควนซ์ Shopping ต้นแบบของระบบที่ 2



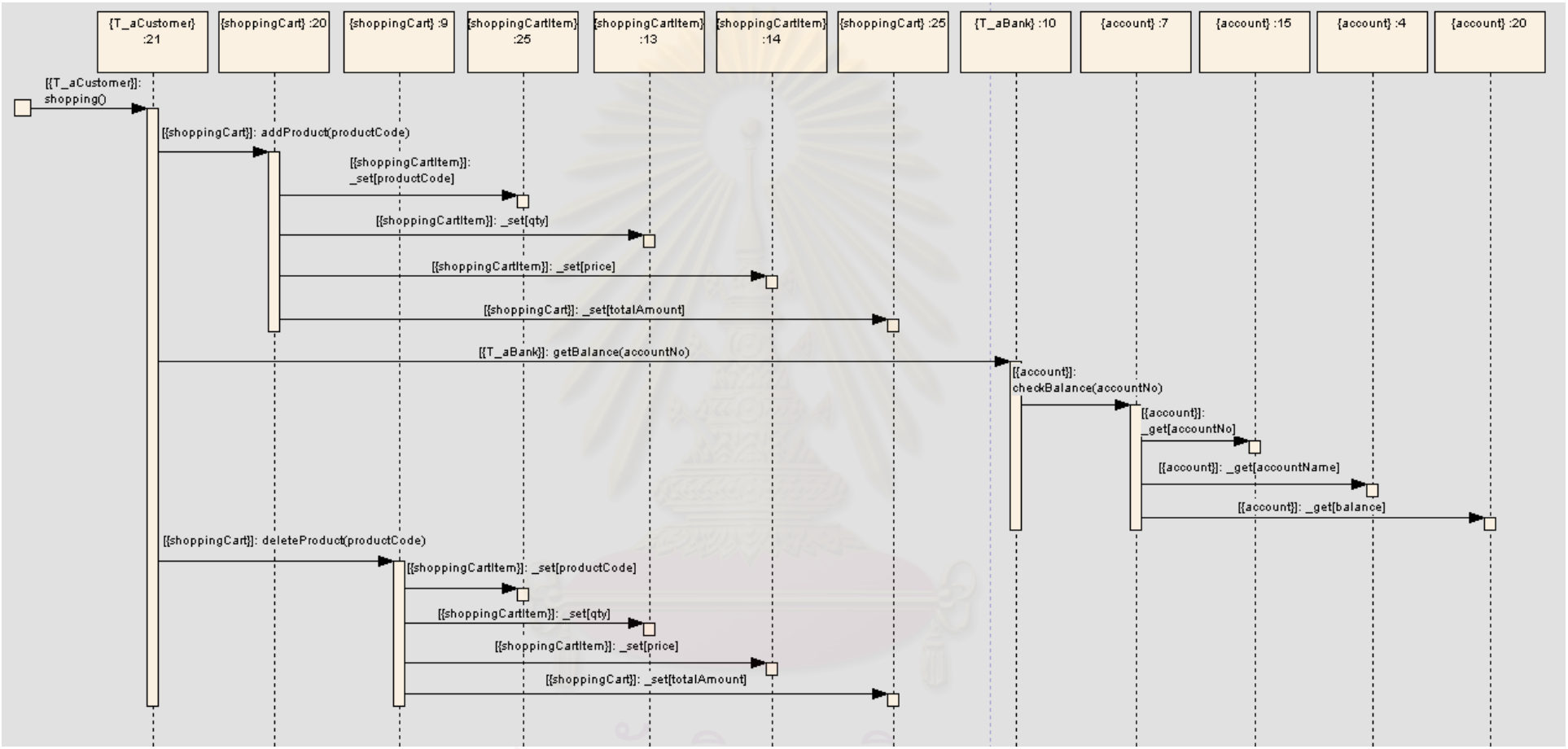
รูปที่ 5.11 แผนภาพซีเควนซ์ PlaceOrder ต้นแบบของระบบที่ 2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



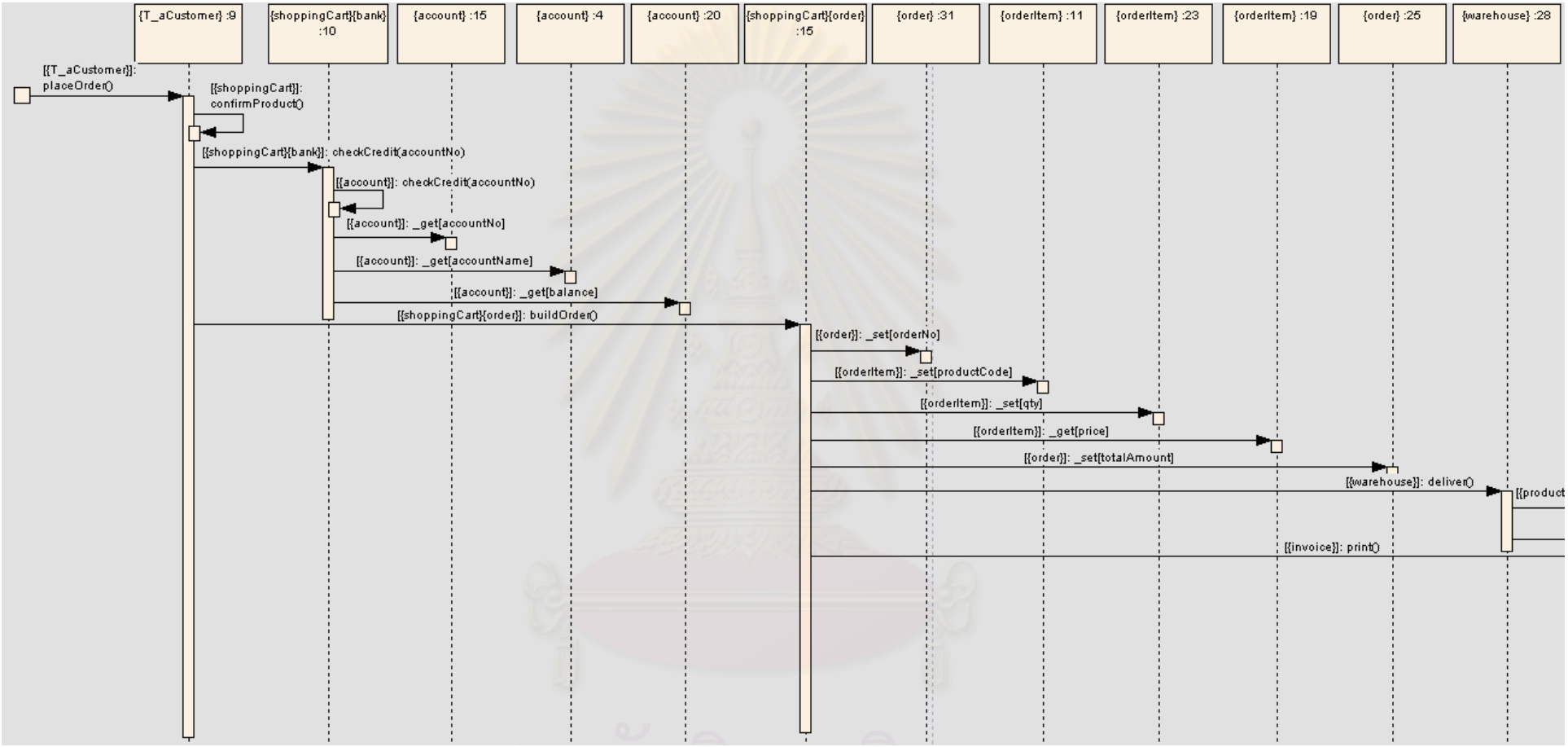
รูปที่ 5.12 แผนภาพคลาสที่ถูกปรับปรุงแล้วของระบบที่ 2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



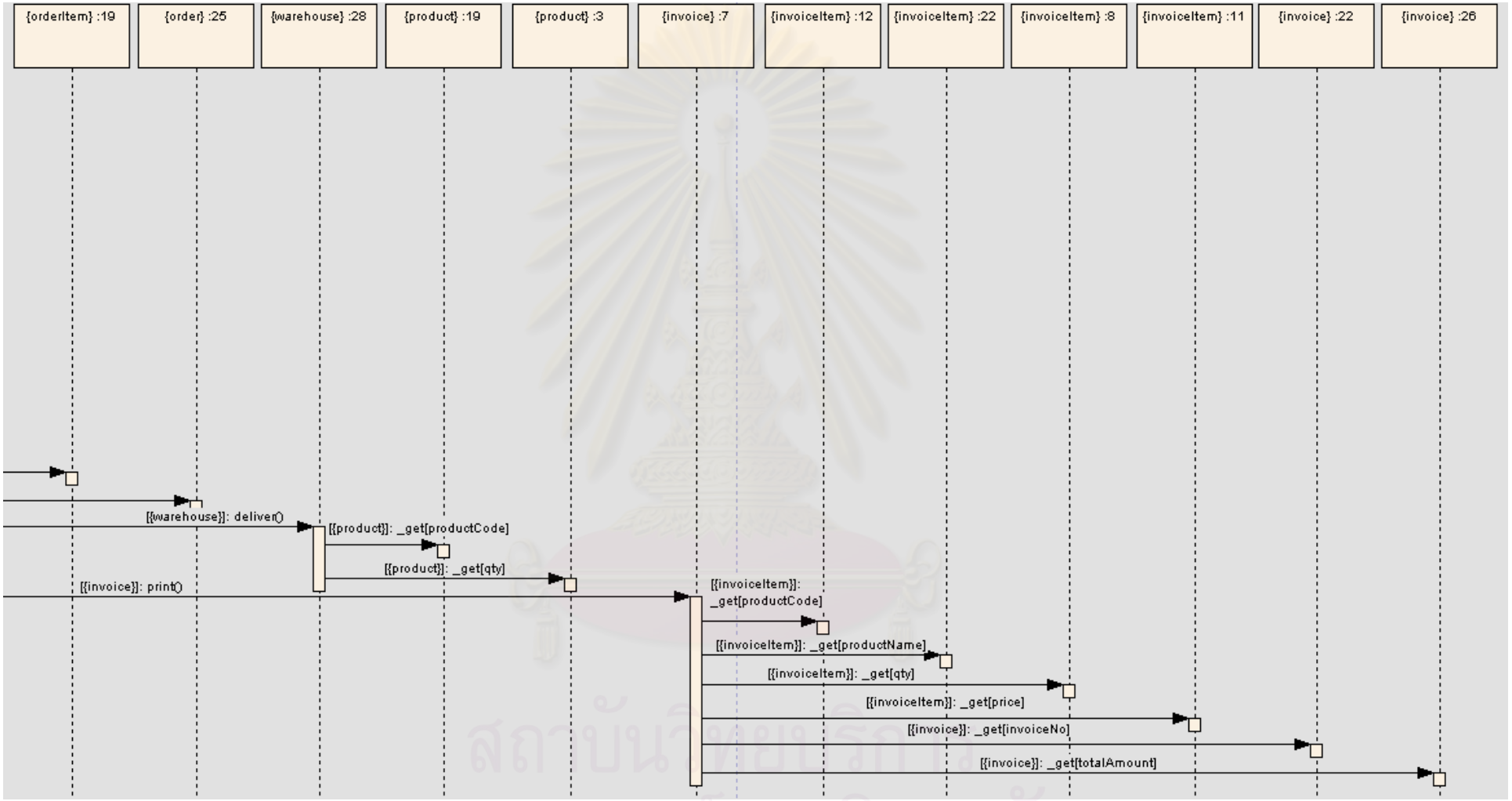
รูปที่ 5.13 แผนภาพซีเควนซ์ Shopping ที่ถูกปรับปรุงแล้วของระบบที่ 2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.14 แผนภาพซีเควนซ์ PlaceOrder ที่ถูกปรับปรุงแล้วของระบบที่ 2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.14 แผนภาพซีเควนซ์ PlaceOrder ที่ถูกปรับปรุงแล้วของระบบที่ 2 (ต่อ)

5.3 การทดสอบฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบ

การทดลองนี้เป็นการทดสอบการปรับปรุงคุณภาพของโมเดลการออกแบบระบบที่ 3 ด้วยโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ โดยใช้ฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบที่ถูกนิยามไว้ที่สมการที่ 3.1 เป็นฟังก์ชันในการวัดคุณภาพ โดยค่าที่น้อยกว่าที่ได้จากฟังก์ชันนี้จะแสดงถึงคุณภาพของโมเดลการออกแบบที่ดีกว่า

ระบบที่ 3 นี้เป็นระบบสำหรับการรับคืนสินค้าจากลูกค้า โดยลูกค้าจะแจ้งความต้องการคืนสินค้าให้แก่ผู้แทนขาย ซึ่งผู้แทนขายจะตรวจสอบข้อมูลการซื้อสินค้าในใบกำกับภาษี และสร้างเอกสารการคืนสินค้าในระบบ โดยเจ้าหน้าที่สินเชื่อจะดำเนินการแจ้งให้ผู้มีอำนาจเข้ามาอนุมัติเอกสารการคืนสินค้า หลังจากเอกสารการคืนสินค้าได้รับการอนุมัติแล้ว ระบบจะสร้างใบลดหนี้เพื่อใช้ในการคืนเงินให้กับลูกค้าต่อไป

การทดลองนี้ได้ทำการกำหนดค่าของ *w1* และ *w2* ของสมการที่ 3.1 ให้เป็น 0.6 และ 0.4 ตามลำดับโดยอ้างอิงจาก [9] จากโมเดลการออกแบบต้นแบบที่ถูกแสดงในรูปที่ 5.15 ถึงรูปที่ 5.17

โดยที่ค่ามาตรวัดต่างๆ ของโมเดลต้นแบบมีค่าดังนี้

- *COH(D)* มีค่าเป็น 6.83178149045208
- $COP_{IA}(D)$ มีค่าเป็น 4.57453046337253

ดังนั้นเมื่อแทนค่าลงในสมการที่ 3.1 แล้วจะได้ค่าของฟังก์ชันการประเมินเป็น 1.9176370083236

โดยรายละเอียดของมาตรวัด *COH(C)* และ $COP_{IA}(C)$ ของโมเดลต้นแบบจะแสดงไว้ที่ตารางที่ 5.5

เมื่อทำการทดลองปรับปรุงคุณภาพโมเดลการออกแบบด้วยโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุที่พัฒนาขึ้นแล้วจะได้โมเดลผลลัพธ์ดังแสดงได้ในรูปที่ 5.18 ถึงรูปที่ 5.20

โดยที่ค่ามาตรวัดต่างๆ ของโมเดลผลลัพธ์มีค่าดังนี้

- *COH(D)* มีค่าเป็น 3.39511888696783
- $COP_{IA}(D)$ มีค่าเป็น 0.242720506978449

ดังนั้นเมื่อแทนค่าลงในสมการที่ 3.1 แล้วจะได้ค่าของฟังก์ชันการประเมินเป็น 0.273812500224115

โดยรายละเอียดของมาตรวัด *COH(C)* และ $COP_{IA}(C)$ ของโมเดลผลลัพธ์จะถูกแสดงไว้ที่ตารางที่ 5.6

ตารางที่ 5.5 ค่ามาตรวัด *COH(C)* และ $COP_{IA}(C)$ ของคลาสต่างๆ ในโมเดลต้นแบบของระบบที่ 3

Class	*COH(C)*	$COP_{IA}(C)$
SalesRep	1	1.346716
CreditNoteItem	N/A	0
CreditStaff	1	0.57735
CreditManager	1	1.676387
CreditNote	1	0
Product	N/A	0
InvoiceItem	N/A	0
ReturnOrder	0.9782281	0.174078
ReturnOrderItem	N/A	0
Invoice	0.853553391	0
Customer	1	0.8

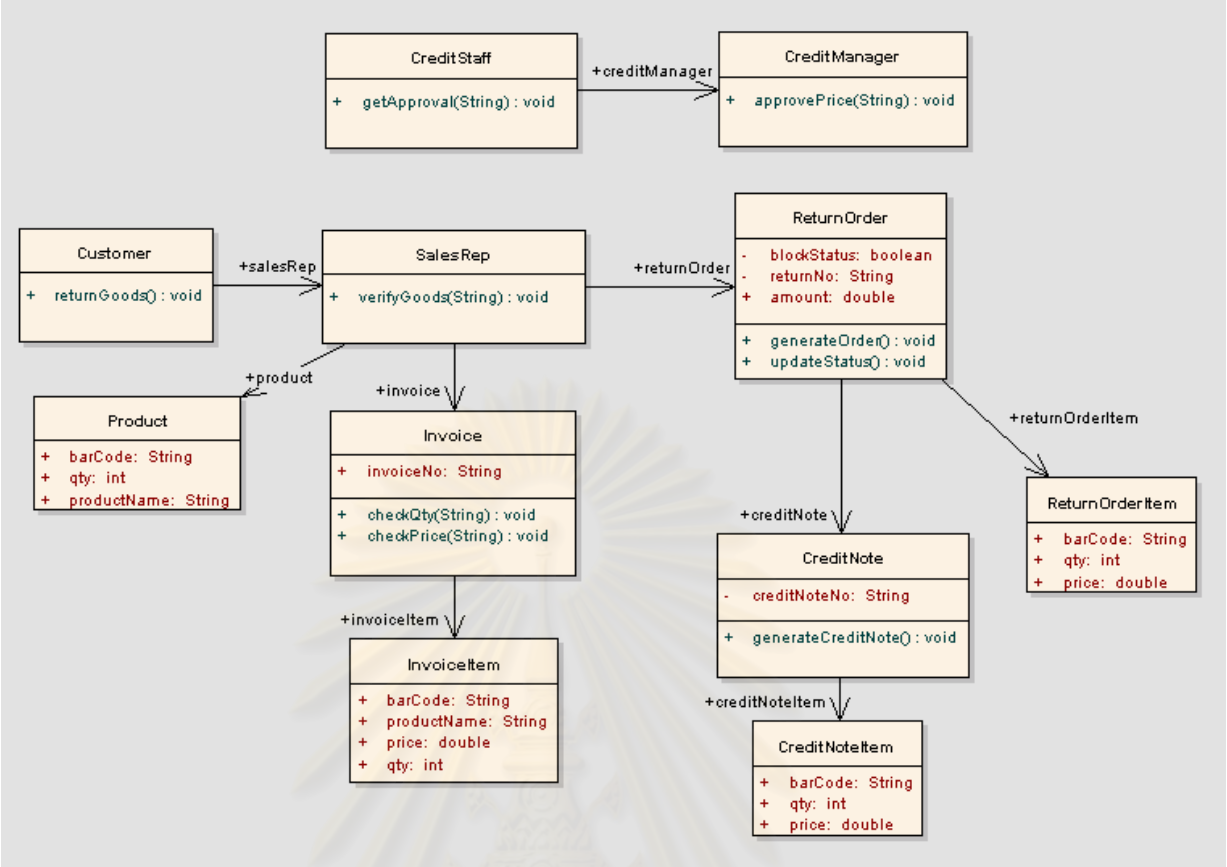
ค่า N/A ในตารางที่ 5.5 และ 5.6 จะหมายความถึงคลาสที่ไม่มีเมทธอดเลยจึงไม่สามารถคำนวณค่ามาตรวัด *SBM* ที่เป็นพื้นฐานของมาตรวัดตัวที่เหลือที่ใช้ในการทดลองได้ ในการทดลองนี้สมมติให้ N/A มีค่าเสมือนเป็น 0

ตารางที่ 5.6 ค่ามาตรวัด *COH(C)* และ $COP_{IA}(C)$ ของคลาสต่างๆ ในโมเดลที่ถูกปรับปรุงแล้วของระบบที่ 3

Class	*COH(C)*	$COP_{IA}(C)$
13	0.754812104	0.11113
7	N/A	0
12	N/A	0
4	N/A	0
21	N/A	0
6	N/A	0
24	1	0
25	1	0
11	N/A	0
8	N/A	0
9	0.640306783	0.13159
14	N/A	0

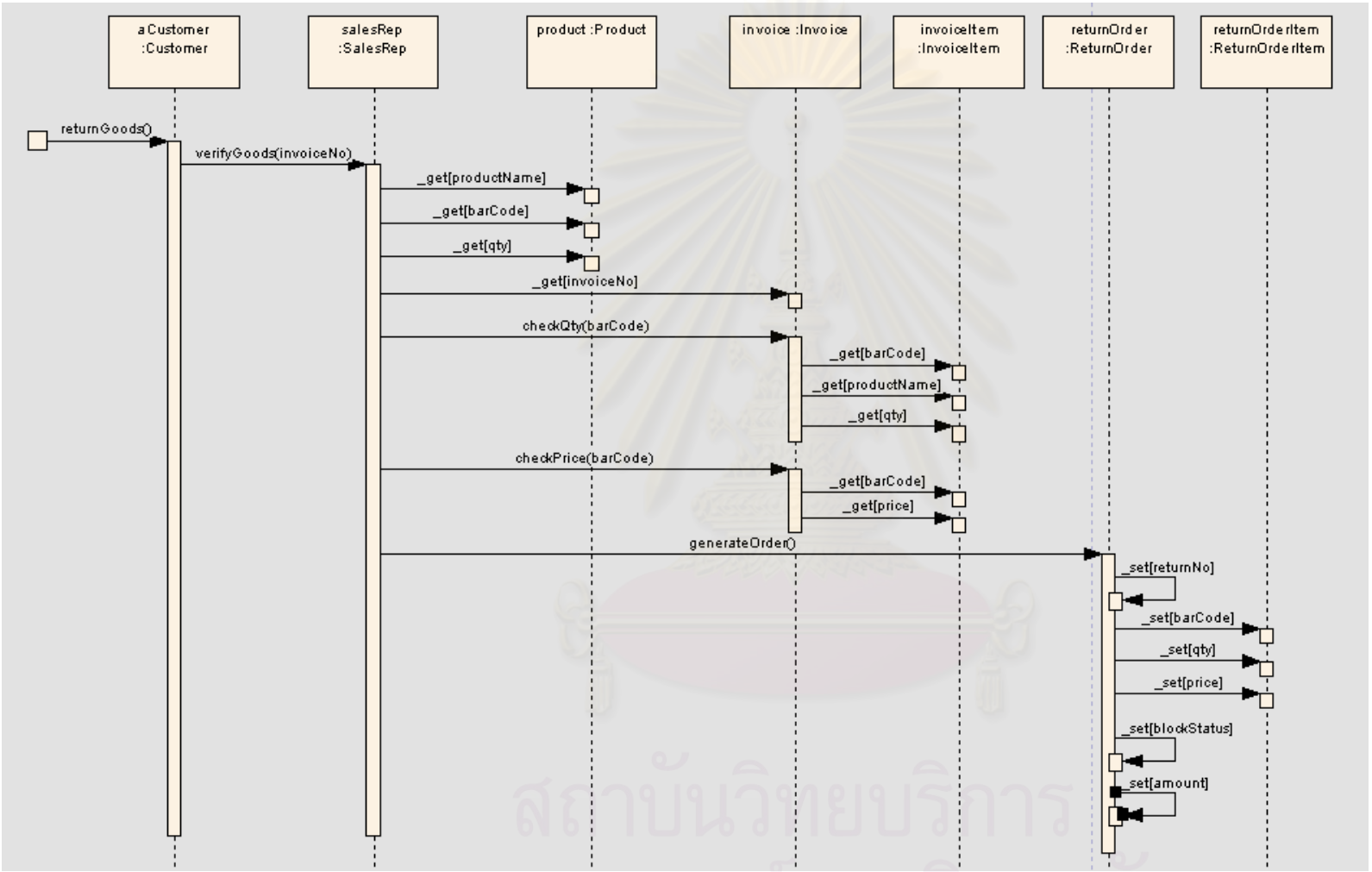
การปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันสามารถทำได้โดยการเพิ่มการเกาะกลุ่มกัน และ/หรือ ลดการเข้าคู่กันในแต่ละคลาส จากตารางที่ 5.6 พบว่าผลลัพธ์ที่ได้จากโครงร่างนั้นจะเป็นการลดการเข้าคู่กันเสียเป็นส่วนมาก หากผู้ใช้ต้องการให้ความสำคัญกับการเพิ่มการเกาะกลุ่มกันในแต่ละคลาสมากกว่านี้ก็สามารถทำได้โดยการเพิ่มค่า *w1* และ ลดค่า *w2* ที่เลือกมาใช้ในการทดลองนี้

จากการทดลองจะเห็นว่าค่าของฟังก์ชันการประเมินที่ได้มีค่าลดลงจาก 1.9176370083236 เป็น 0.273812500224115 เป็นการแสดงว่าโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุสามารถปรับปรุงโมเดลการออกแบบด้วยฟังก์ชันการประเมินเพื่อปรับปรุงการเกาะกลุ่มกัน และการเข้าคู่กันของคลาสภายในโมเดลการออกแบบเป็นผลสำเร็จ

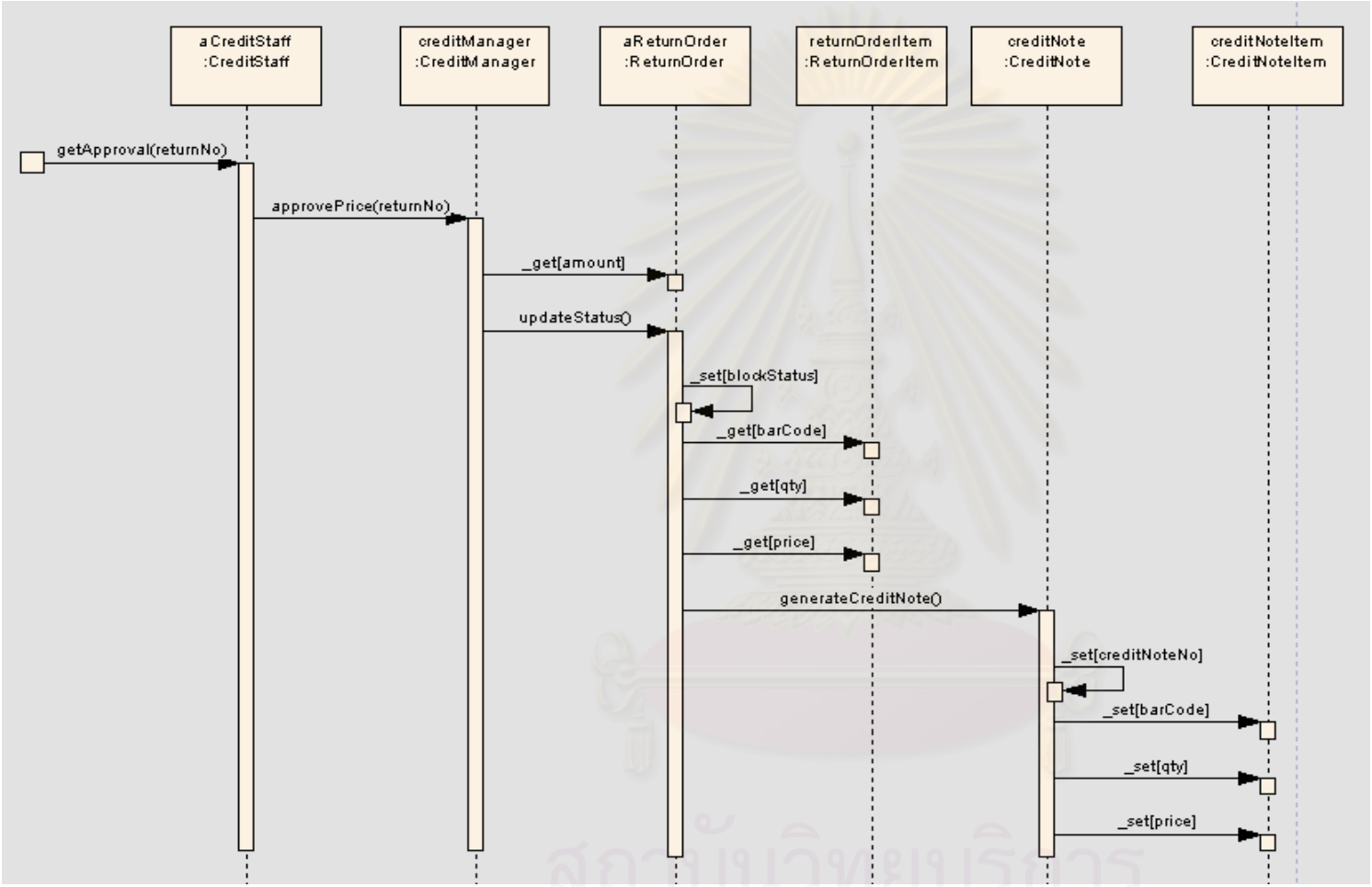


รูปที่ 5.15 แผนภาพคลาสต้นแบบของระบบที่ 3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

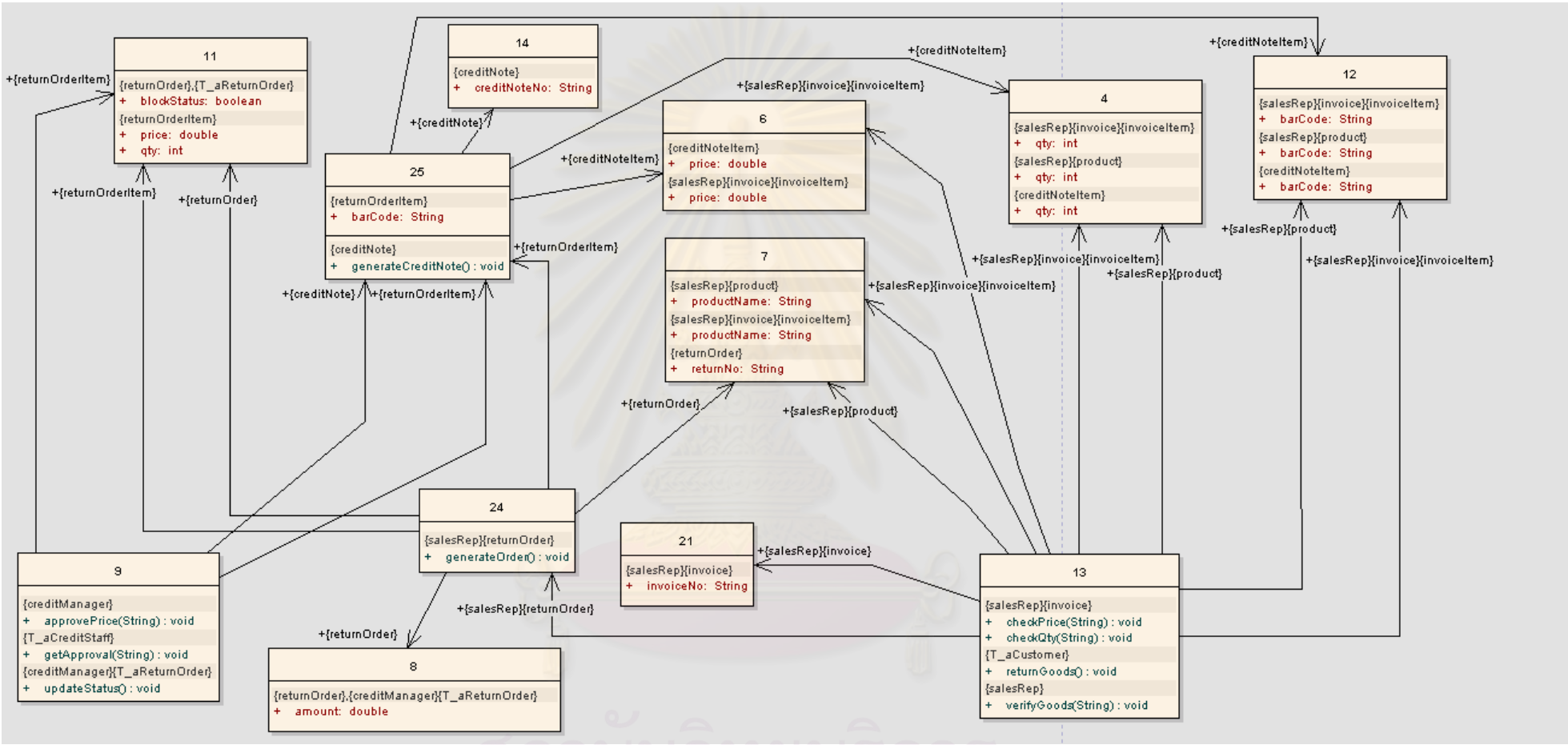


รูปที่ 5.16 แผนภาพซีเควนซ์ ReturnGoods ต้นแบบของระบบที่ 3



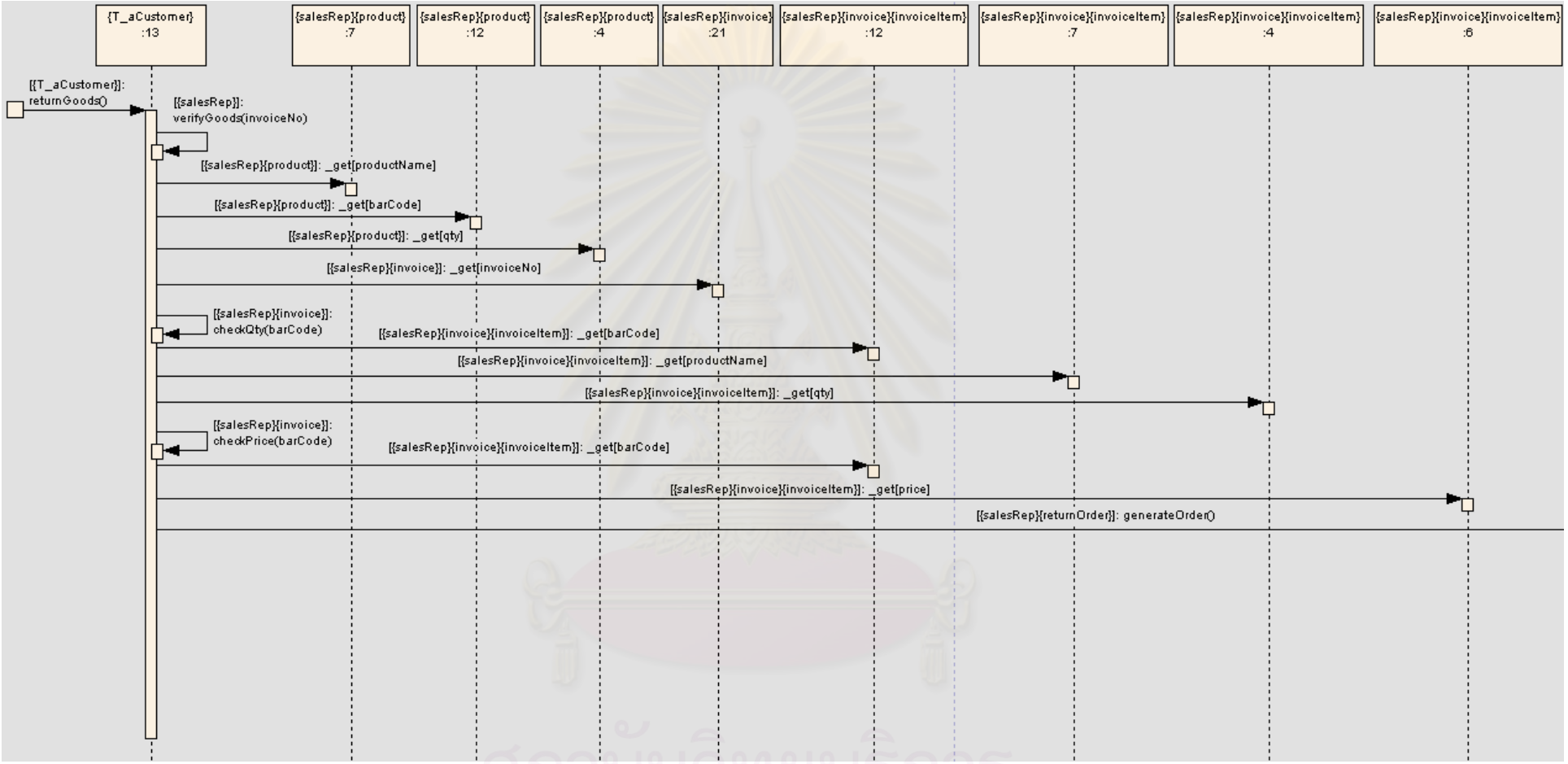
รูปที่ 5.17 แผนภาพซีเควนซ์ GetApproval ต้นแบบของระบบที่ 3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



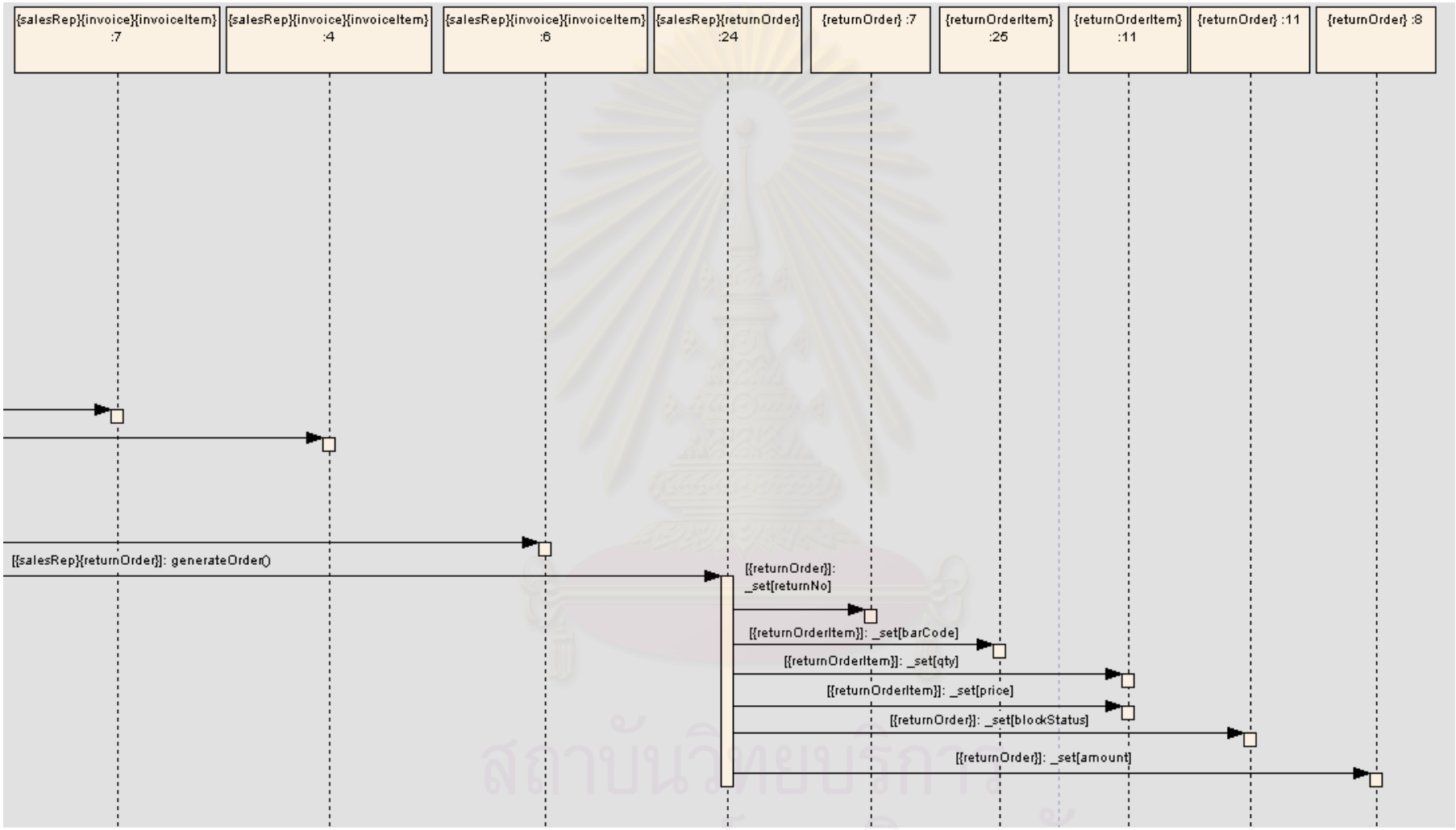
รูปที่ 5.18 แผนภาพคลาสที่ถูกปรับปรุงแล้วของระบบที่ 3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

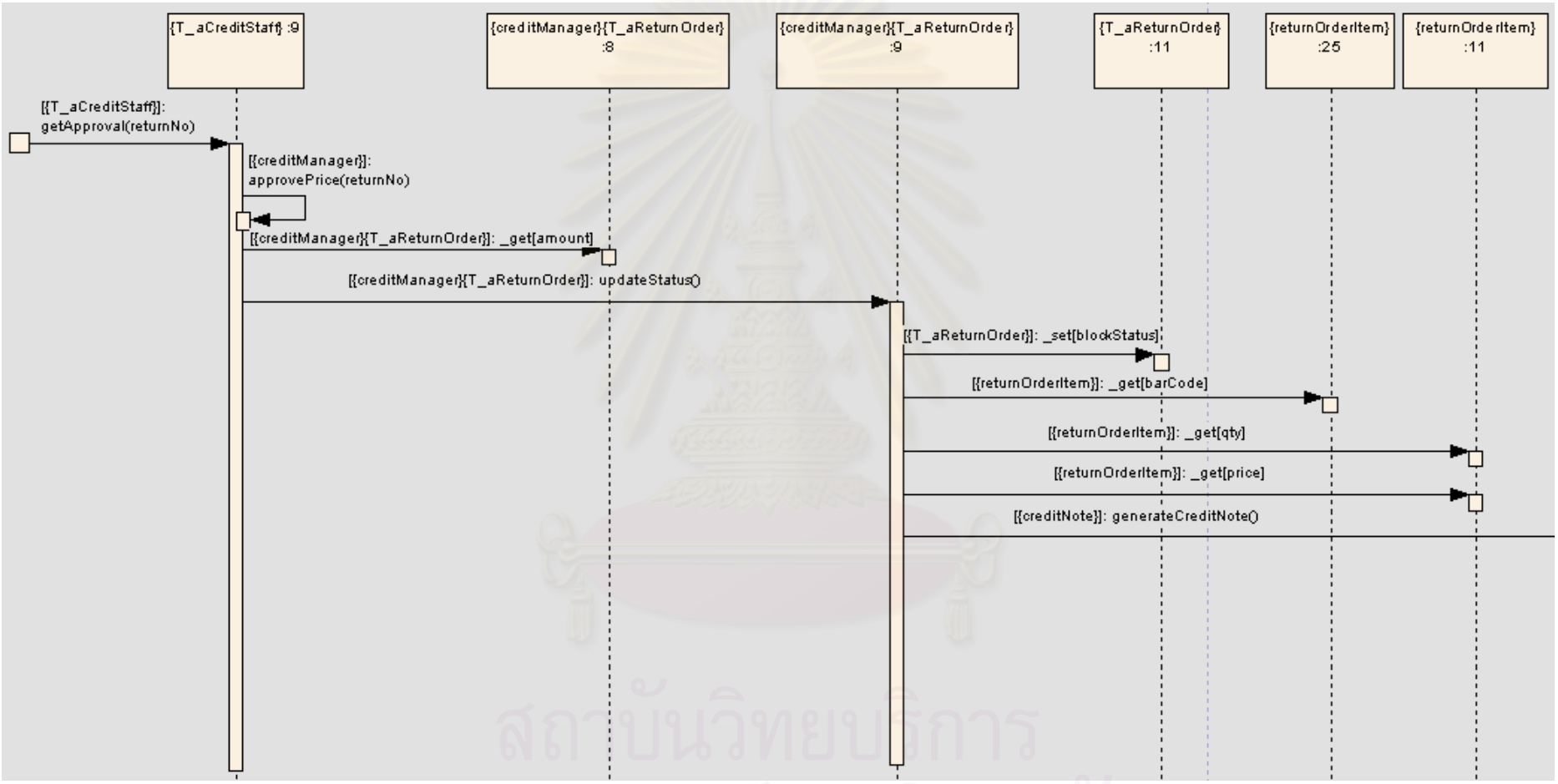


รูปที่ 5.19 แผนภาพซีเควนซ์ ReturnGoods ที่ถูกปรับปรุงแล้วของระบบที่ 3

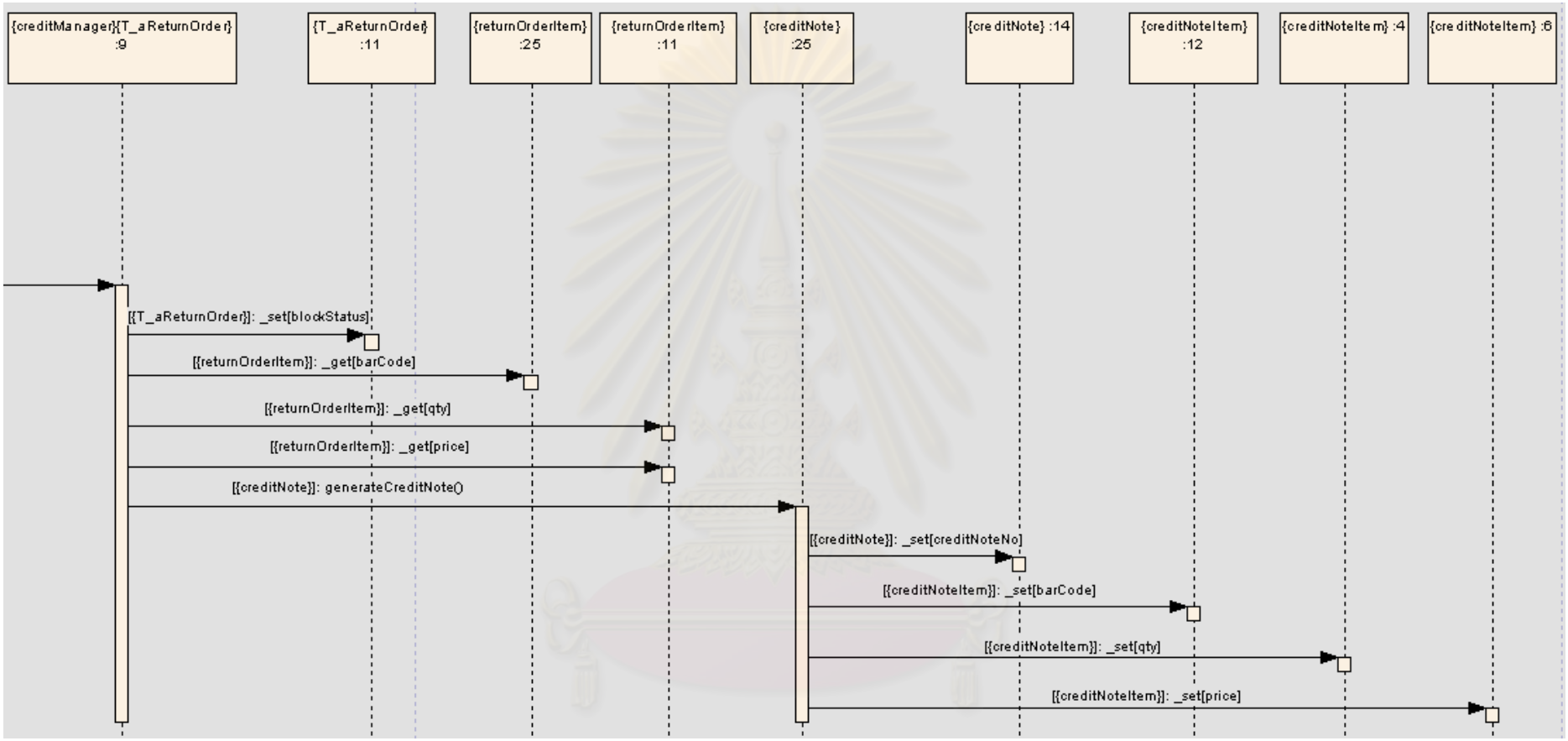
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.19 แผนภาพซีเควนซ์ ReturnGoods ที่ถูกปรับปรุงแล้วของระบบที่ 3 (ต่อ)



รูปที่ 5.20 แผนภาพซีเควนซ์ GetApproval ที่ถูกปรับปรุงแล้วของระบบที่ 3



รูปที่ 5.20 แผนภาพซีเควนซ์ GetApproval ที่ถูกปรับปรุงแล้วของระบบที่ 3 (ต่อ)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

บทสรุปและข้อเสนอแนะ

บทสรุป

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนาโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุโดยพิจารณาที่แผนภาพคลาสและแผนภาพซีเควนซ์ ในวิทยานิพนธ์ได้ประยุกต์ใช้วิธีซีมูเล็ทเต็ดแอนนิลลิงด้วยการหาค่าเหมาะสมเฉพาะแห่งเพื่อทำให้ได้รูปแบบที่มีคุณภาพในด้านที่สนใจที่ดีที่สุดสำหรับโมเดลการออกแบบใด ๆ โดยคุณภาพของโมเดลการออกแบบสามารถที่จะวัดได้ด้วยฟังก์ชันการประเมินค่าที่เป็นปลั๊กอินของโครงร่าง ภายในกระบวนการปรับปรุงคุณภาพโมเดลการออกแบบนั้น โครงร่างจะทำการย้ายเมทธอดและแอตทริบิวต์ต่างๆ ที่มีในโมเดลการออกแบบให้ไปเป็นสมาชิกของคลาสต่างๆ แล้วจึงทำการประเมินค่าคุณภาพของการเปลี่ยนแปลงรอบนั้นๆ เมื่อได้ค่าของโมเดลการออกแบบที่ดีที่สุดเท่าที่จะทำได้แล้วกระบวนการปรับปรุงคุณภาพจึงจะหยุดทำงานภายใต้ข้อกำหนดในไฟล์คอนฟิกูเรชัน

ในวิทยานิพนธ์ฉบับนี้ได้จัดทำฟังก์ชันการประเมินค่า 3 ฟังก์ชันด้วยกันคือ

- ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบ
- ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบ
- ฟังก์ชันการประเมินเพื่อเพิ่มปริมาณการเกาะกลุ่มกัน และลดการเข้าคู่กัน ของคลาสภายในโมเดลการออกแบบ

ผู้วิจัยได้ทำการทดสอบโครงร่างที่ใช้ฟังก์ชันทั้ง 3 กับโมเดลการออกแบบของระบบตัวอย่าง 3 ระบบที่เลือกมาให้เหมาะกับการทดสอบแต่ละฟังก์ชัน และทำการเปรียบเทียบคุณภาพแต่ละด้านด้วยฟังก์ชันการประเมินก่อนและหลังการปรับปรุงคุณภาพโมเดลการออกแบบของโครงร่าง

ผลการทดลองสรุปได้ว่า โครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุสามารถใช้ฟังก์ชันการประเมินค่าทั้ง 3 ในการปรับปรุงคุณภาพของโมเดลการออกแบบได้เป็นผลสำเร็จ และในการทดสอบฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีภายในโมเดลการออกแบบ และฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบ สามารถที่จะระบุได้ว่าโมเดลการออกแบบที่ได้มา มีค่าคุณภาพเป็นค่าที่ดีที่สุดในบริบทของฟังก์ชันการประเมินนั้นๆ กล่าวคือโครงร่างที่ใช้ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดี

แบบฟีเจอร์เอนวีภายในโมเดลการออกแบบสามารถกำจัดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวีออกจากโมเดลการออกแบบได้ทั้งหมด และ โครงร่างที่ใช้ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบมิดเดิลแมนภายในโมเดลการออกแบบสามารถกำจัดร่องรอยที่ไม่ดีแบบมิดเดิลแมนออกจากโมเดลการออกแบบได้ทั้งหมดเช่นกัน

โครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุนี้มีลักษณะเด่นคือ

- สามารถหารูปแบบที่เหมาะสมของโมเดลยูเอ็มแอลที่แผนภาพคลาสและแผนภาพซีเควนซ์
- สามารถเปลี่ยนไปใช้ฟังก์ชันการประเมินหลายๆ อย่างได้ง่ายเนื่องจากโครงร่างนี้รองรับการติดต่อจากฟังก์ชันการประเมินในลักษณะที่เป็นปลั๊กอิน
- รูปแบบของมาตรวัดที่ใช้ สนับสนุนให้การนำกลับมาใช้ใหม่ทำได้ง่าย
- มีลักษณะเป็นโครงร่างที่สนับสนุนให้การพัฒนาปลั๊กอินซึ่งก็คือ ฟังก์ชันการประเมินและมาตรวัดต่างๆ ขึ้นมาใหม่เป็นไปโดยง่าย
- ไม่ขึ้นกับเครื่องมือที่ใช้ผลิตโมเดลยูเอ็มแอลค่ายต่างๆ

ข้อจำกัดและข้อเสนอแนะ

1. จากการทดลองด้วยฟังก์ชันการประเมินค่าแบบใดๆ จะพบว่าโมเดลผลลัพธ์ที่ได้จะมีคุณภาพที่วัดได้ด้วยฟังก์ชันการประเมินค่านั้นๆ ดีขึ้น แต่ก็อาจจะทำให้คุณภาพด้านอื่นๆ ลดลงได้เหมือนกัน ดังนั้นการจัดทำฟังก์ชันการประเมินใหม่ๆ โดยคำนึงถึงคุณภาพหลายๆ ด้าน ตัวอย่างเช่น ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีแบบฟีเจอร์เอนวี และ มิดเดิลแมนภายในโมเดลการออกแบบ อาจทำให้โมเดลผลลัพธ์ที่ได้มีคุณภาพดีขึ้นในหลายๆ ด้านด้วยเช่นกัน

2. ฟังก์ชันการประเมินเพื่อลดร่องรอยที่ไม่ดีที่ถูกนำมาใช้เป็นตัวอย่างในการทดลองในงานวิจัยนี้เป็นเพียง 2 ตัวอย่างในการประเมินร่องรอยที่ไม่ดีเพียง 2 แบบเท่านั้น การประยุกต์ใช้ฟังก์ชันการประเมินในการตรวจจับร่องรอยที่ไม่ดีชนิดอื่นๆ ก็สามารถกระทำได้

3. ค่าคงที่ที่เลือกใช้ในโครงร่างเช่น ค่าที่สามารถปรับแต่งได้ในไฟล์คอนฟิกูเรชันเป็นค่าที่ถูกเลือกมาจากการทดลอง ด้วยทดสอบว่าการใช้ค่าคงที่ใดในการใช้โครงร่างปรับปรุงคุณภาพของโมเดลการออกแบบตัวอย่าง แล้วทำให้โมเดลการออกแบบตัวอย่างนั้นมีค่าคุณภาพที่ดีที่สุด ค่าคงที่เหล่านี้อาจสามารถถูกกำหนดให้เป็นค่าเหมาะสมมากกว่าที่ใช้ในวิทยานิพนธ์ฉบับนี้ได้ โดยการใช้วิธีการหาค่าเหมาะสมเชิงวิธีจัดหมู่อีกทีหนึ่ง

4. ความสัมพันธ์แบบเจนเนอรัลไลเซชัน เป็นความสัมพันธ์ที่พบได้ทั่วไปในโมเดลยูเอ็มแอลดังนั้นการปรับปรุงให้โครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุนี้ สามารถทำงานได้กับความสัมพันธ์ประเภทดังกล่าวได้จะเป็นการเปิดให้โอกาสในการนำโครงร่างนี้ไปใช้ได้จริงในหลายๆ สภาพแวดล้อมมีมากขึ้น

5. คลาสของโมเดลยูเอ็มแอลที่ผ่านการปรับปรุงจากโครงร่างจะมีชื่อคลาสเป็นตัวเลขและสมาชิกแต่ละตัวของคลาสจะถูกกำกับด้วยบริบทการถูกเรียกใช้ จึงต้องอาศัยผู้เชี่ยวชาญในการกำหนดชื่อของคลาสใหม่ประกอบกับการแปลความหมายของบริบทการถูกเรียกใช้อีกที เพื่อทำให้โมเดลยูเอ็มแอลนั้นมีความสมบูรณ์พอที่จะนำไปใช้พัฒนารหัสต้นทางได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] UML Resource Page. [online] Available from: http://www.uml.org

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

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

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

[5] Ingber, L. Very Fast Simulated Re-Annealing. J. Mathl. Comput. Modelling, 12, 1989: 967-973.

[6] Ingber, L. Adaptive simulated annealing (ASA): Lessons learned. [online] Available from:ftp.caltech.edu:/pub/ingber/asa.Z. Software package documentation, 1995.

[7] Ingber, L. “Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison. J. Mathl. Comput. Modelling, 16(11)(1992):87-100.

[8] Desai, R. and Patil, R. SALO: Combining Simulated Annealing and Local Optimization for Efficient Global Optimization. Proceedings of the 9th Florida AI Research Symposium (FLAIRS-’96), Key West, FL: 233-237, June 1996.

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

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

[11] Hyperspherical coordinates. [online] Available from: http://en.wikipedia.org/wiki/Hypersphere

[12] Mark O’Keeffe and Mel ´O Cinn´eide. A Stochastic Approach to Automated Design Improvement. ACM SIGMOD International Conference on Management of Data, 1991.

[13] Tanyawat Chanpia. Defects Detection at Design Phase for Improving Object-Oriented Software Modifiability. Master's Thesis, Department of Computer Engineering, Chulalongkorn University, 2006

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

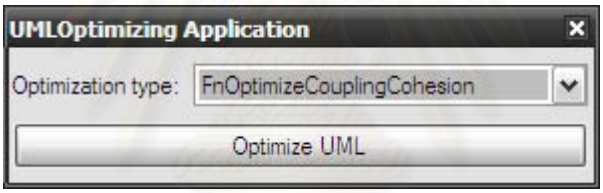
ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

การใช้งานเครื่องมือที่ใช้ทดสอบโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

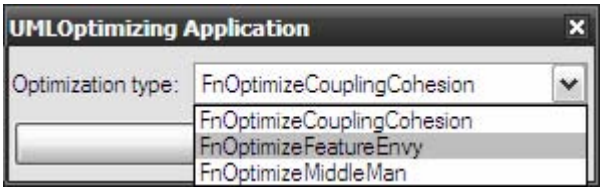
ในภาคผนวกนี้จะแสดงรายละเอียดเกี่ยวกับวิธีการใช้เครื่องมือทดสอบโครงร่างสำหรับงานวิจัยนี้ โปรแกรมประยุกต์อย่างง่ายได้ถูกจัดทำขึ้นเพื่อเป็นเครื่องมือในการทดสอบการทำงานของโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุที่ได้พัฒนาขึ้นมา โดยที่โปรแกรมประยุกต์นี้พัฒนาขึ้นมาเพื่อให้ทำงานภายใต้สภาพแวดล้อมของ.NET Framework 2.0 ซึ่งปัจจุบันถูกใช้อย่างแพร่หลายบนระบบปฏิบัติการวินโดว์ โดยใช้ภาษา C# เป็นภาษาที่ใช้เขียนรหัสต้นทาง โปรแกรมประยุกต์ดังกล่าวมีลักษณะดังที่ได้แสดงไว้ในรูปที่ ก.1



รูปที่ ก.1 โปรแกรมประยุกต์ที่ใช้ทดสอบโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

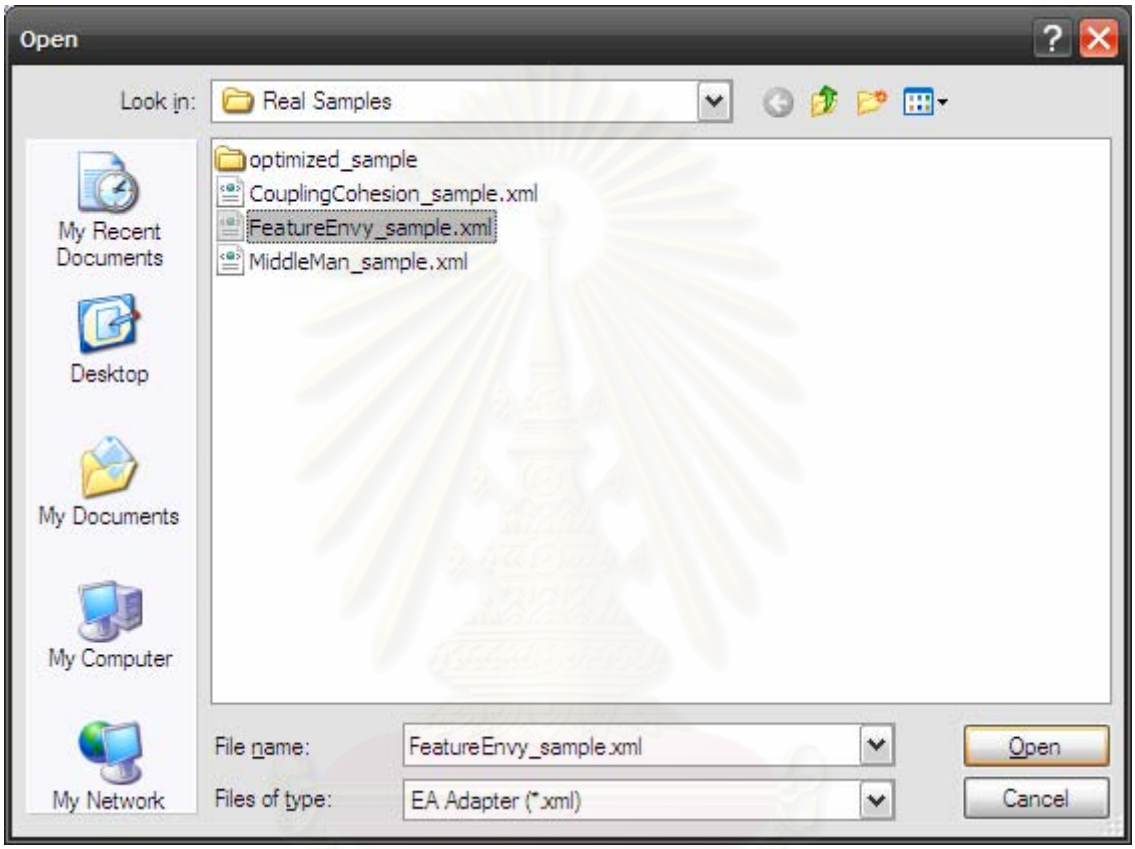
โปรแกรมประยุกต์นี้จะทำหน้าที่ปรับปรุงคุณภาพของโมเดลการออกแบบใดๆ ที่ถูกเลือกโดยความช่วยเหลือของโครงร่าง โดยจะมีวิธีการใช้งานเป็นลำดับดังนี้

1. หลังจากสั่งให้โปรแกรมนี้เริ่มทำงานแล้ว ผู้ใช้สามารถเลือกได้ว่าจะใช้ปลั๊กอินที่ใช้ในการประเมินคุณภาพของโมเดลการออกแบบ ตัวใด ดังแสดงในรูปที่ ก.2



รูปที่ ก.2 หน้าจอสำหรับเลือกปลั๊กอินที่ใช้ประเมินคุณภาพ

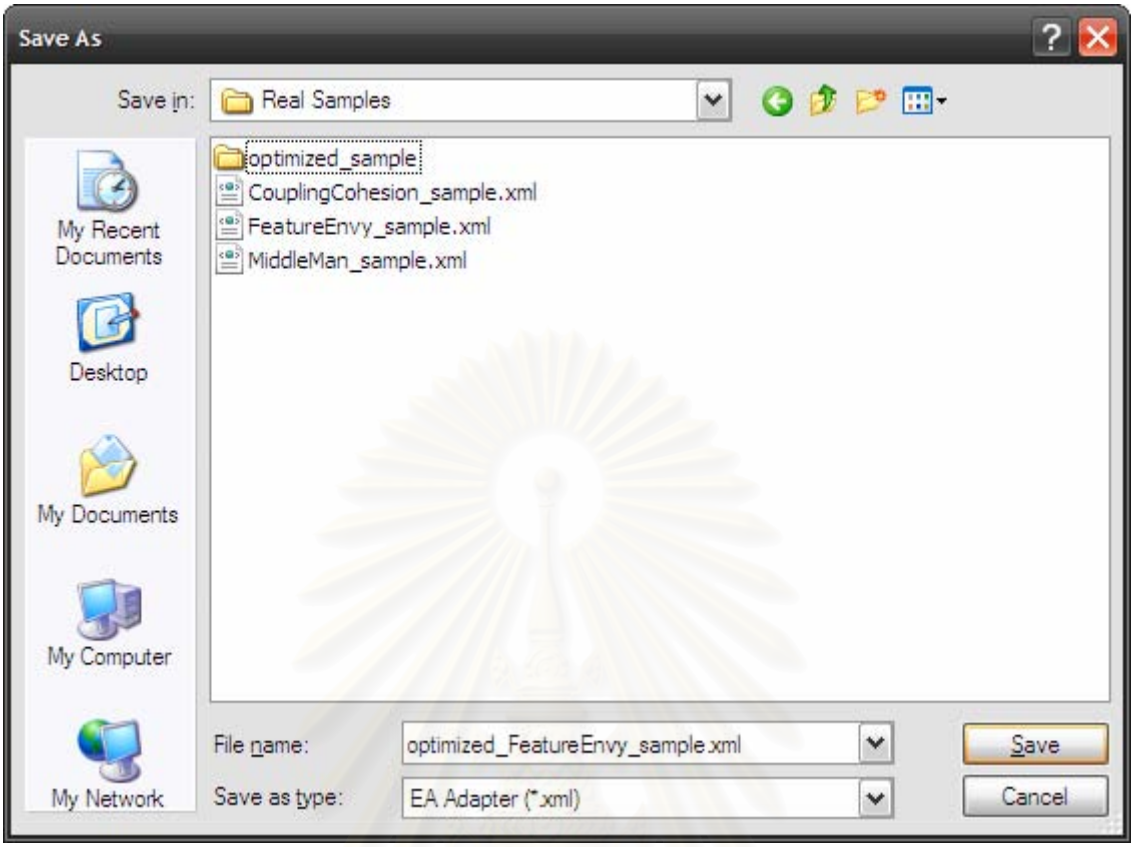
2. เมื่อผู้ใช้ทำการเลือกปลั๊กอินที่ใช้ในการประเมินคุณภาพของโมเดลการออกแบบแล้ว โปรแกมจะให้ผู้ใช้เลือกไฟล์ของโมเดลการออกแบบต้นทาง ดังรูปที่ ก.3 ซึ่งในกรณีของตัวอย่างนี้จะเป็นไฟล์ที่ถูกผลิตโดยเครื่องมือ Sparx Enterprise Architect



รูปที่ ก.3 หน้าจอสำหรับเลือกไฟล์ของโมเดลการออกแบบต้นทาง

หลังจากเลือกไฟล์ของโมเดลการออกแบบต้นทางแล้ว โปรแกรมจะสั่งให้โครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุทำการเปลี่ยนแปลงโครงสร้างโมเดลการออกแบบที่ได้นำเข้าเพื่อปรับปรุงคุณภาพของโมเดลการออกแบบนั้น ให้สอดคล้องกับปลั๊กอินที่ใช้ในการประเมินคุณภาพที่ได้เลือกไว้

3. เมื่อทำการปรับปรุงคุณภาพของโมเดลการออกแบบเสร็จแล้ว โปรแกรมจะให้ผู้ใช้เลือกว่าจะบันทึกโมเดลการออกแบบผลลัพธ์ที่ได้ลงไปที่ใดดังรูปที่ ก.4 ซึ่งถือเป็นการสิ้นสุดกระบวนการทำงาน



รูปที่ ก.4 หน้าจอสำหรับเลือกตำแหน่งสำหรับบันทึกบันทึกโมเดลการออกแบบผลลัพธ์ที่ได้

โปรแกรมประยุกต์นี้จะบันทึกรายละเอียดของโมเดลการออกแบบต้นทางและของโมเดลการออกแบบผลลัพธ์ในการทำการปรับปรุงคุณภาพโมเดลการออแบบครั้งหนึ่งๆ ไว้ในแฟ้มข้อมูลภายนอก โดยเนื้อหาภายในแฟ้มข้อมูลจะมีลักษณะดังแสดงในรูปที่ ก.5

UMLOptimizerMng Initial Model Details ----------
COPA_D Class: A COH: 1
COPA_D Class: B COH: 1
COPA_D Class: C COPA: 0
COPA_D Class: A COPA: 1
COPA_D Class: B COPA: 0
FnOptimizeCouplingCohesion COH_D: 2 COPA_D: 1 Quality: 0.7
UMLOptimizerMng Initial Model Cost = 0.7
UMLOptimizerMng Result Model Details ----------
COPA_D Class: 1 COH: 1
COPA_D Class: 1 COPA: 0
FnOptimizeCouplingCohesion COH_D: 1 COPA_D: 0 Quality: 0.6
UMLOptimizerMng Result Model Cost = 0.6

รูปที่ ก.5 รายละเอียดของโมเดลการออกแบบต้นทางและของโมเดลการออกแบบผลลัพธ์ที่บันทึกในแฟ้มข้อมูลภายนอก

ภาคผนวก ข

อินเทอร์เฟซโครงร่างสำหรับหารูปแบบที่เหมาะสมของโมเดลการออกแบบเชิงวัตถุ

class UMLOptimizingMng

คลาสสำหรับจัดการการหารูปแบบที่เหมาะสมของโมเดลยูเอ็มแอล

UML.UMLModel optimize(string strActiveCostFnName, UML.UMLModel theUMLModel)
เมทธอดสำหรับหารูปแบบที่เหมาะสมของโมเดลยูเอ็มแอล **พารามิเตอร์** • ***strActiveCostFnName***: ชื่อของปลั๊กอินฟังก์ชันการประเมินเพื่อปรับปรุงคุณภาพโมเดลการออกแบบ • ***theUMLModel***: โมเดลยูเอ็มแอลที่ต้องการจะปรับปรุง **ค่ารีเทิร์น** • โมเดลยูเอ็มแอลที่ถูกปรับปรุงแล้ว

interface IUMLCostFnEngine

อินเทอร์เฟซสำหรับปลั๊กอินฟังก์ชันการประเมินเพื่อปรับปรุงคุณภาพโมเดลการออกแบบ

double calculateUMLCost(UML.UMLModel theUMLModel)
เมทธอดสำหรับหาค่าประเมิน (ค่าคุณภาพ) ของโมเดลยูเอ็มแอล **พารามิเตอร์** • ***theUMLModel***: โมเดลยูเอ็มแอลที่ต้องการจะหาค่าประเมิน **ค่ารีเทิร์น** • ค่าประเมินของโมเดลยูเอ็มแอล

class OptimizationMng

คลาสสำหรับจัดการการโหลดปลั๊กอินการหารูปแบบที่เหมาะสมเข้าสู่หน่วยความจำ

static Optimizer.IOptimizerEngine loadOptimizerEngine(string strEngineName)
เมทธอดสำหรับโหลดปลั๊กอินการหารูปแบบที่เหมาะสมเข้าสู่หน่วยความจำ **พารามิเตอร์** • ***strEngineName***: ชื่อของปลั๊กอินการหารูปแบบที่เหมาะสมที่ต้องการโหลด **ค่ารีเทิร์น** • ปลั๊กอินการหารูปแบบที่เหมาะสม

interface IOptimizerEngine

อินเทอร์เฟซของปลั๊กอินการหารูปแบบที่เหมาะสมของอาเรย์ตัวเลขใดๆ

void registerCostFunctionCalculator(Optimizer.ICostFunctionCalculator theCostFunctionCalculator)
เมทธอดสำหรับลงทะเบียนออบเจ็กต์คำนวณค่าประเมิน **พารามิเตอร์** • ***theCostFunctionCalculator***: ออบเจ็กต์คำนวณค่าประเมิน

System.Collections.Generic.List<double> run(System.Collections.Generic.List<double> theInitialState)
เมทธอดสำหรับการหารูปแบบที่เหมาะสมของอาเรย์ตัวเลข **พารามิเตอร์** • ***theInitialState***: อาเรย์ของตัวเลขตั้งต้น **ค่ารีเทิร์น** • อาเรย์ของตัวเลขที่อยู่ในรูปแบบที่เหมาะสมแล้ว

interface ICostFunctionCalculator

อินเทอร์เฟซของออบเจ็กต์คำนวณค่าประเมิน

double calculateCostFunction(System.Collections.Generic.List<double> theState)
เมทธอดสำหรับการหาค่าประเมินของอาเรย์ตัวเลข **พารามิเตอร์** • ***theState***: อาเรย์ของตัวเลขที่ต้องหาค่าประเมิน **ค่ารีเทิร์น** • ค่าประเมิน

class UMLMetricMng

คลาสสำหรับจัดการการหาค่ามาตรวัดของชุดของส่วนประกอบย่อยของโมเดลยูเอ็มแอลใดๆ

static double compute(string strMetricName, UML.UMLEntity[] theUMLEntityArray)
เมทธอดสำหรับใช้บริการปลั๊กอินมาตรวัดในการคำนวณค่ามาตรวัดของชุดของส่วนประกอบย่อยของโมเดลยูเอ็มแอล **พารามิเตอร์** • ***strMetricName***: ชื่อของปลั๊กอินมาตรวัดที่ต้องการใช้ • ***theUMLEntityArray***: ชุดของส่วนประกอบย่อยของโมเดลยูเอ็มแอลที่ต้องการหาค่ามาตรวัด **ค่ารีเทิร์น** • ค่ามาตรวัดที่คำนวณได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

interface IUMLMetricEngine

อินเทอร์เฟซของปลั๊กอินมาตรวัด

double compute(UML.UMLEntity[] theUMLEntityArray)
เมทธอดสำหรับคำนวณค่ามาตรวัดของชุดของส่วนประกอบย่อยของโมเดลยูเอ็มแอล **พารามิเตอร์** • ***theUMLEntityArray***: ชุดของส่วนประกอบย่อยของโมเดลยูเอ็มแอลที่ต้องการหาค่ามาตรวัด **ค่ารีเทิร์น** • ค่ามาตรวัดที่คำนวณได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

class IOMng

คลาสสำหรับจัดการการโหลดและบันทึกโมเดลยูเอ็มแอล

static UML.UMLModel load(string strEngineName, string strUMLStorageName)

เมทธอดสำหรับใช้บริการของปลั๊กอินการโหลดและบันทึกโมเดลยูเอ็มแอลในการโหลดโมเดลยูเอ็มแอล

พารามิเตอร์

- ***strEngineName***: ชื่อของปลั๊กอินการโหลดและบันทึกโมเดลยูเอ็มแอลที่ต้องการเรียกใช้
- ***strUMLStorageName***: ชื่อไฟล์ต้นทางของโมเดลยูเอ็มแอลที่จัดเก็บอยู่บนฮาร์ดดิสก์

ค่ารีเทิร์น

- โมเดลยูเอ็มแอลที่โหลดมาได้

static bool save(UML.UMLModel theUMLModel, string strEngineName, string strUMLStorageName)

เมทธอดสำหรับใช้บริการของปลั๊กอินการโหลดและบันทึกโมเดลยูเอ็มแอลในการบันทึกโมเดลยูเอ็มแอล

พารามิเตอร์

- ***theUMLMode***: โมเดลยูเอ็มแอลที่ต้องการบันทึก
- ***strEngineName***: ชื่อของปลั๊กอินการโหลดและบันทึกโมเดลยูเอ็มแอลที่ต้องการเรียกใช้
- ***strUMLStorageName***: ชื่อไฟล์ปลายทางของโมเดลยูเอ็มแอลที่จะทำการบันทึกบนฮาร์ดดิสก์

ค่ารีเทิร์น

- ค่าบูลีนบ่งชี้ความสำเร็จในการบันทึกไฟล์

interface IUMLIOEngine

อินเทอร์เฟซของปลั๊กอินการโหลดและบันทึกโมเดลยูเอ็มแอล

UML.UMLModel loadUMLModel(string strUMLStorageName)
เมทธอดสำหรับโหลดโมเดลยูเอ็มแอล **พารามิเตอร์** • ***strUMLStorageName***: ชื่อไฟล์ต้นทางของโมเดลยูเอ็มแอลที่จัดเก็บอยู่บนฮาร์ดดิสก์ **ค่ารีเทิร์น** • โมเดลยูเอ็มแอลที่โหลดมาได้

bool saveUMLModel(UML.UMLModel theUMLModel, string strUMLStorageName)
เมทธอดสำหรับบันทึกโมเดลยูเอ็มแอล **พารามิเตอร์** • ***theUMLMode***: โมเดลยูเอ็มแอลที่ต้องการบันทึก • ***strUMLStorageName***: ชื่อไฟล์ปลายทางของโมเดลยูเอ็มแอลที่จะทำการบันทึกบนฮาร์ดดิสก์ **ค่ารีเทิร์น** • ค่าบูลีนบ่งชี้ความสำเร็จในการบันทึกไฟล์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายอติโรจน์ สืบวงศ์คล้าย เกิดเมื่อวันที่ 1 พฤษภาคม พ.ศ. 2520 สำเร็จการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต (วศ.บ.) สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อปีการศึกษา 2542 และเข้าศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2547

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย