การระบุโอกาสสำหรับการรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจ

นางสาวศศิพัชร บุญขวัญ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2557
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

IDENTIFICATION OF OPPORTUNITIES FOR MOVE METHOD REFACTORING USING DECISION THEORY

Miss Sasipat Boonkwan

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Software Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2014

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การระบุโอกาสสำหรับการรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจ
โดย	นางสาวศศิพัชร บุญขวัญ
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี)

.....กรรมการ

(ดร.พีรพล เวทีกูล)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร. มฑุปายาส ทองมาก)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ศศิพัชร บุญขวัญ : การระบุโอกาสสำหรับการรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจ (IDENTIFICATION OF OPPORTUNITIES FOR MOVE METHOD REFACTORING USING DECISION THEORY) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.พรศิริ หมื่นไชยศรี, 109 หน้า.

วิทยานิพนธ์นี้นำเสนอวิธีการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงโดยใช้ทฤษฎีการตัดสินใจ ซึ่งวิธีการนี้จะค้นหา**เมท็อดเป้าหมายเพื่อย้ายไปยังคลาสที่เหมาะสม โดยใช้วิธีลาปลาสเป็นเกณฑ์ในการตัดสินใจ** วิธีการที่นำเสนอสามารถช่วยให้นักพัฒนาเลือกเมท็อดเป้าหมายที่จะถูกย้ายไปยังคลาสที่เหมาะสม

งานวิจัยนี้ได้มีการออกแบบวิธีการปรับปรุงคุณภาพของรหัสต้นทาง โดยการมูฟเมท็อดรีแฟคทอริงเพื่อปรับปรุงคุณภาพของซอฟต์แวร์เชิงวัตถุ การค้นหาเมท็อดเป้าหมายจะถูกแบ่งออกเป็น 2 ส่วน โดยส่วนแรกคือ การหาคลาสใดมีการเรียกใช้เมท็อดเมท็อดเป้าหมายบ้าง ส่วนที่สองคือ การพิจารณาว่าจะย้ายเมท็อดเป้าหมายไปที่คลาสใดที่เหมาะสม โดยวัด CIM, RMC, ICBC และ IMC ในแต่ละคลาส และเลือกคลาสที่เหมาะสมสำหรับให้เมท็อดเป้าหมายย้ายไป ด้วยเกณฑ์ลาปลาส

งานวิจัยนี้ได้พัฒนาเครื่องมือสำหรับการระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อดด้วยภาษาซีชาร์ป และได้ทำการทดสอบเครื่องมือกับระบบที่พัฒนาด้วยภาษาซีชาร์ป 6 ระบบ โดยระบบที่นำมาทดสอบจะวัดด้วยมาตรวัด LCOM, TCC, RFC, CBO ผลการประเมินพบว่า การประยุกต์ใช้วิธีการระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อดด้วยทฤษฎีการตัดสินใจ สามารถเพิ่มคุณภาพของรหัสต้นทางได้ โดยพิจาณาจากมาตรวัด เมื่อคิดเป็นร้อยละของมาตรวัดจากทั้งระบบโดยนำค่า Mean ของมาตรวัดการเกาะกลุ่มกันและการเข้าคู่กัน ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดยมาตรวัด LCOM ลดลงร้อยละ 34.82 TCC เพิ่มขึ้นร้อยละ 8.46 RFC ลดลงร้อยละ 33.07 CBO ลดลงร้อยละ 13.43

5570997021 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: MOVE METHOD REFACTORING / DECISION THEORY / OBJECT-ORIENTED SOFTWARE METRICS

SASIPAT BOONKWAN: IDENTIFICATION OF OPPORTUNITIES FOR MOVE METHOD REFACTORING USING DECISION THEORY. ADVISOR: ASSOC. PROF. PORNSIRI MUENCHAISRI, Ph.D., 109 pp.

This thesis proposes an approach to identity opportunity for move method using decision theory. The approach searches target methods of class to be moved to an appropriate class which is used as decision criteria of Laplace method. The proposed method can help developers to select a target method that will be moved to an appropriate class.

This thesis has improved the design of source code by Move Method refactoring in order to improve quality of object oriented software. Finding the target method which will be moved to appropriate class is divided into two principal parts. The first part is to find candidate classes which have at least one call to a target method. The second part is to choose a target class from candidate classes using CIM, RMC, ICBC and IMC measures as Laplace criterion.

This thesis has created a tool to identify opportunities of Move Method Refactoring with C# language. The tool measures LCOM, TCC, RFC and CBO metrics of six object-oriented software systems. Results of the evaluation show that the proposed method can improve the design quality of the source code by considering from the cohesion and coupling metrics. LCOM value decreases 34.82%, TCC value increases 8.46%, RFC decreases 33.07%, and CBO value decreases 13.43%.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความช่วยเหลือจาก รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี อาจารย์ที่ปรึกษาวิทยานิพนธ์ของข้าพเจ้า กราบขอบพระคุณอาจารย์ที่ให้คำแนะนำและข้อเสนอแนะต่างๆ ตลอดระยะเวลาของการทำวิทยานิพนธ์ของข้าพเจ้า จนสำเร็จลุล่วงไปได้ด้วยดี

กราบขอบพระคุณ รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา ประธานกรรมการสอบวิทยานิพนธ์ของข้าพเจ้า และผู้ช่วยศาสตราจารย์ ดร. มฑุปายาส ทองมาก และ อาจารย์ ดร. พีรพล เวทีกูล กรรมการสอบวิทยานิพนธ์ของข้าพเจ้า ซึ่งได้เสียสละเวลาให้คำแนะนำและข้อคิดดีๆ ในการสอบวิทยานิพนธ์ของข้าพเจ้าอย่างดียิ่ง จนสำเร็จลุล่วงไปได้ด้วยดี

ท้ายสุดนี้ กราบขอบพระคุณบิดา มารดา พี่สาว และเพื่อนๆ ทุกคนที่คอยเป็นกำลังใจและให้การสนับสนุนมาโดยตลอด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฎ

สารบัญภาพ ..... ณ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 3

1.3 ขอบเขตงานวิจัย 3

1.4 ขั้นตอนการวิจัย 3

1.5 ประโยชน์ที่จะได้รับ 4

1.6 คำจำกัดความที่ใช้ในงานวิจัยนี้ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 6

2.1 ทฤษฎีที่เกี่ยวข้อง 6

2.1.1 การรีแฟคทอริง (Refactoring) 6

2.1.2 ทฤษฎีการตัดสินใจ (Decision Theory) 7

2.1.3 มาตรวัดซอฟต์แวร์เชิงวัตถุ (Object-Oriented Software Metrics) 11

2.1.3.1 มาตรวัดการติดต่อปฏิสัมพันธ์ระหว่างเมท็อด (Call-based Interaction between Methods- CIM)[8] 11

2.1.3.2 มาตรวัดความสัมพันธ์การเข้าคู่กันของเมท็อด (Relative Method Coupling - RMC)[9] 11

2.1.3.3 มาตรวัดการตอบสนองต่อคลาส (Response for a class - RFC)[10] 12

2.1.3.4 มาตรวัดการเข้าคู่กันระหว่างวัตถุ (Coupling between objects - CBO) [10] 13

2.1.3.5 มาตรวัดระดับของการขาดการเกาะกลุ่มกันของเมท็อดภายในคลาส (Lack of cohesion of methods - LCOM) [11],[12],[13] 13

2.1.3.6 มาตรวัดการเกาะกลุ่มกันหนาแน่น (Tight Class Cohesion -TCC)[14] 13

ตัวอย่างการคำนวณมาตรวัด 14

2.1.4 การแปลงข้อมูล (Data Transformation)[15] 17

2.2 งานวิจัยที่เกี่ยวข้อง 17

2.2.1 Suggesting Extract Class Refactoring Opportunities by Measuring Strength of Method Interactions [16] 17

2.2.2 Move Code Refactoring with Dynamic Analysis [17] 18

2.2.3 Playing with Refactoring Identifying Extract Class Opportunities through Game Theory [18] 18

บทที่ 3 การระบุโอกาสสำหรับการรีแฟคทอริงแบบมูฟเมท็อดด้วยวิธีทฤษฎีการตัดสินใจ 20

3.1 มาตรวัดการเข้าคู่กัน 20

3.1.1 มาตรวัดความสัมพันธ์การเข้าคู่กันระหว่างคลาส (Interaction coupling between classes-ICBC) 20

3.1.2 มาตรวัดความสัมพันธ์การเข้าคู่กันระหว่างเมท็อด (Interaction method coupling - IMC) 20

3.2 ขั้นตอนการปรับปรุงคุณภาพของรหัสต้นทาง 22

ขั้นตอนที่ 1 การคำนวณมาตรวัดการเกาะกลุ่มกันการเข้าคู่กันก่อนรีแฟคทอริง 25

ขั้นตอนที่ 2 การระบุโอกาสการย้ายเมท็อดด้วยทฤษฎีการตัดสินใจ 25

ขั้นตอนที่ 2.1 การกำหนดเมท็อดเป้าหมาย 25

ขั้นตอนที่ 2.2 การค้นหาคลาสที่มีการเรียกใช้เมท็อดเป้าหมาย 25

หน้า

ขั้นตอนที่ 2.3 การเลือกคลาสที่เหมาะสมสำหรับเมท็อดเป้าหมาย..... 27
ขั้นตอนที่ 2.4 การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม..... 28
ขั้นตอนที่ 3 การมูฟเมท็อดรีแฟคทอริง..... 28
ขั้นตอนที่ 4 การคำนวณมาตรวัดการเกาะกลุ่มกันการเข้าคู่กันหลังรีแฟคทอริง..... 28
ขั้นตอนที่ 5 การประเมินค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กัน..... 28
การประเมินความสามารถในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง..... 29
3.3 ตัวอย่างการปรับปรุงคุณภาพของรหัสต้นทาง..... 29
บทที่ 4 การออกแบบเครื่องมือ IMRD สำหรับการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง..... 36
4.1 แผนภาพยูสเคสแสดงการปฏิสัมพันธ์ระหว่างผู้ใช้งานกับระบบ..... 36
4.1.1 ยูสเคส Add C# Source code..... 37
4.1.2 ยูสเคส Parser Generator..... 38
4.1.3 ยูสเคส Finding Alternative from Method Target..... 38
4.1.4 ยูสเคส Calculate Metrics..... 39
4.1.5 ยูสเคส View Identification Move Method Refactoring..... 39
4.1.6 ยูสเคส Identification of Move Method Refactoring..... 40
4.1.7 ยูสเคส View Metrics Cohesion & Coupling..... 40
4.2 แผนภาพคลาสแสดงความสัมพันธ์ของการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง..... 41
4.3 แผนภาพลำดับการทำงาน..... 44
4.3.1 แผนแสดงภาพการเพิ่มโปรแกรมภาษาซีชาร์ปเข้าสู่ระบบ..... 44
4.3.2 แผนภาพแสดงการทำพาร์เซอร์..... 44
4.3.3 แผนภาพแสดงการคำนวณค่ามาตรวัด..... 45
4.3.4 แผนภาพแสดงผลการวัดจากการเลือกเมนู View เมนูย่อย Metric..... 46

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

4.3.5 แผนภาพแสดงการระบุโอกาสการมูฟเมท็อด 47
4.4 การพัฒนาเครื่องมือ IMRD Tool 48
บทที่ 5 การประเมินการระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อดด้วยวิธีของทฤษฎีการตัดสินใจ 51
5.1 ระบบซอฟต์แวร์ที่นำมาทดสอบ 51
5.2 การระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อด 53
5.3 การประเมินความสามารถในการทำรีแฟคทอริงหลังจากใช้ทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อด 60
5.3.1 ระบบ Repairs Management System 60
5.3.2 ระบบ Restaurant Billing System 62
5.3.3 ระบบ Book Store 64
5.3.4 ระบบ Ordering Administration System 67
5.3.5 ระบบ Graph of Function 69
5.3.6 ระบบ Image Fan 70
5.3.7 การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ทุกระบบ 71
บทที่ 6 บทสรุปการวิจัย 73
6.1 บทสรุป 73
6.2 ข้อจำกัด 74
6.3 ข้อเสนอแนะ 74
รายการอ้างอิง 75
ภาคผนวก ก คู่มือการใช้งาน IMRD Tool 78
ภาคผนวก ข การระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อดด้วยวิธีของทฤษฎีการตัดสินใจกับระบบที่พัฒนาด้วยภาษาซีชาร์ป 84
ข-1 ระบบ Restaurant Billing System 84
ข-2 ระบบ Book Store 89

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

ข-3 ระบบ Ordering Administration System 98
ข-4 ระบบ Graph of Function 101
ข-5 ระบบ Image Fan 105
ประวัติผู้เขียนวิทยานิพนธ์ 109

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

หน้า

ตารางที่ 2.1 ตัวอย่างตารางตัดสินใจ 8
ตารางที่ 2.2 การตัดสินใจการจะเปิดสาขาใหม่ของบริษัทเครื่องเสียง 10
ตารางที่ 2.3 ผลตอบแทนเมทริกซ์สำหรับเกมความลำบากใจของนักโทษ 19
ตารางที่ 3.1 การเรียกใช้กันระหว่างคลาสต้นทางและคลาสปลายทาง 32
ตารางที่ 3.2 การเรียกใช้ Method 3 จากคลาสภายนอกหลายคลาส 32
ตารางที่ 3.3 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับ Method 3 33
ตารางที่ 3.4 การเรียกใช้ Method 6 จากภายนอกคลาสและภายในคลาส 33
ตารางที่ 3.5 การวัดค่า IMC ของการเรียกใช้ Method 6 34
ตารางที่ 3.6 การตัดสินใจการเรียกใช้ Method 6 จากคลาสภายนอก 34
ตารางที่ 3.7 เมท็อดที่แนะนำในการย้ายไปยังคลาสที่เหมาะสม 34
ตารางที่ 3.8 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริง 35
ตารางที่ 4.1 แสดงขั้นตอนการทำงานของยูสเคส Add C# Source code 37
ตารางที่ 4.2 แสดงขั้นตอนการทำงานของยูสเคส Parser Generator 38
ตารางที่ 4.3 แสดงขั้นตอนการทำงานของยูสเคส Finding Alternative from Method Target 38
ตารางที่ 4.4 แสดงขั้นตอนการทำงานของยูสเคส Calculate Metrics 39
ตารางที่ 4.5 แสดงขั้นตอนการทำงานของยูสเคส View Identification Move Method Refactoring 39
ตารางที่ 4.6 แสดงขั้นตอนการทำงานของยูสเคส Identification of Move Method Refactoring 40
ตารางที่ 4.7 แสดงขั้นตอนการทำงานของยูสเคส View Metrics Cohesion & Coupling 40
ตารางที่ 5.1 รายชื่อและรายละเอียดของระบบที่นำมาทดสอบ 51

ตารางที่ 5.2 การเรียกใช้เมท็อดภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Repairs Management 54
ตารางที่ 5.3 การเรียกใช้เมท็อดภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของเมท็อด subtol 54
ตารางที่ 5.4 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับเมท็อด subtol 55
ตารางที่ 5.5 การเรียกใช้เมท็อดภายนอกคลาสและการเรียกใช้เมท็อดภายในคลาสของระบบ Repairs Management 55
ตารางที่ 5.6 การคำนวณด้วยมาตรวัด IMC ของระบบ Repairs Management 57
ตารางที่ 5.7 การตัดสินใจเลือกคลาสที่เหมาะสม สำหรับเมท็อด Fill 58
ตารางที่ 5.8 การตัดสินใจเลือกคลาสที่เหมาะสม สำหรับเมท็อด GetData 58
ตารางที่ 5.9 เมท็อดที่แนะนำในการย้ายไปยังคลาสที่เหมาะสม ของระบบ Repairs Management 59
ตารางที่ 5.10 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Repairs Management 60
ตารางที่ 5.11 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Restaurant Billing System 62
ตารางที่ 5.12 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Book Store 64
ตารางที่ 5.13 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Ordering Administration System 67
ตารางที่ 5.14 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Graph of Function 69
ตารางที่ 5.15 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Image Fan 70
ตารางที่ 5.16 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Graph of Function 71
ตารางที่ ข.1 การเรียกใช้เมท็อดภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Restaurant Billing System 84
ตารางที่ ข.2 การเรียกใช้เมท็อดภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Restaurant Billing System 85
ตารางที่ ข.3 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับเมท็อด cmbCustomerName_Format.. 85

ตารางที่ ข.4 การเรียกใช้เมท็อดภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Restaurant Billing System 86
ตารางที่ ข.5 การคำนวณด้วยมาตรวัด IMC ของระบบ Restaurant Billing System..... 87
ตารางที่ ข.6 การตัดสินใจเลือกคลาสที่เหมาะสม สำหรับเมท็อดเป้าหมายของระบบ Restaurant Billing System..... 88
ตารางที่ ข.7 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Restaurant Billing System 89
ตารางที่ ข.8 การเรียกใช้เมท็อดภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Book Store 89
ตารางที่ ข.9 การเรียกใช้ภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Book Store..... 90
ตารางที่ ข.10 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับเมท็อดเป้าหมายของระบบ Book Store..... 90
ตารางที่ ข.11 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Book Store..... 91
ตารางที่ ข.12 การคำนวณด้วยมาตรวัด IMC ของระบบ Book Store 93
ตารางที่ ข.13 การคำนวณมาตรวัดในกรณีมีการเรียกใช้ภายในคลาสของระบบ Book Store..... 96
ตารางที่ ข.14 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Book Store..... 97
ตารางที่ ข.15 การเรียกใช้ภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Ordering Administration System..... 98
ตารางที่ ข.16 การเรียกใช้ภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Ordering Administration System..... 98
ตารางที่ ข.17 การตัดสินใจเลือกคลาสที่เหมาะสม สำหรับเมท็อดเป้าหมายของระบบ Ordering Administration System 98
ตารางที่ ข.18 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Ordering Administration System 99
ตารางที่ ข.19 การคำนวณด้วยมาตรวัด IMC ของระบบ Ordering Administration System..... 99

ตารางที่ ข.20 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Ordering Administration System 100
ตารางที่ ข.21 การเรียกใช้ภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Graph of Function 101
ตารางที่ ข.22 การเรียกใช้ภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Graph of Function 101
ตารางที่ ข.23 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับเมท็อดที่จะย้ายระบบ Graph of Function 102
ตารางที่ ข.24 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Graph of Function 102
ตารางที่ ข.25 การคำนวณด้วยมาตรวัด IMC ของระบบ Graph of Function 103
ตารางที่ ข.26 การคำนวณมาตรวัดในกรณีมีการเรียกใช้ภายในคลาสของระบบ Graph of Function 104
ตารางที่ ข.27 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Graph of Function 105
ตารางที่ ข.28 การเรียกใช้ภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Image Fan 105
ตารางที่ ข.29 การเรียกใช้ภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Image Fan 106
ตารางที่ ข.30 การคำนวณมาตรวัดในกรณีไม่มีการเรียกใช้ภายในคลาสของระบบ Image Fan 106
ตารางที่ ข.31 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Image Fan 107
ตารางที่ ข.32 การคำนวณด้วยมาตรวัด IMC ของระบบ Image Fan 107
ตารางที่ ข.33 การคำนวณมาตรวัดในกรณีมีการเรียกใช้ภายในคลาสของระบบ Image Fan 108
ตารางที่ ข.34 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Image Fan 108

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญภาพ

หน้า

ภาพที่ 2.1 ตัวอย่างการ Move Method ระหว่างคลาส A และ คลาส B..... 7
ภาพที่ 2.2 ตัวอย่างของรหัสต้นทาง..... 14
ภาพที่ 2.3 ตัวอย่างแผนภาพคลาสของรหัสต้นทาง 15
ภาพที่ 2.4 ตัวอย่างการเรียกใช้ตัวแปรของเมท็อดในคลาส..... 16
ภาพที่ 3.1 แผนภาพเมท็อดเป้าหมายที่มีการเรียกใช้จากคลาสภายนอก 22
ภาพที่ 3.2 แผนภาพขั้นตอนการปรับปรุงคุณภาพของรหัสต้นทางด้วยการมูฟเมท็อดรีแฟคทอริง ... 24
ภาพที่ 3.3 การระบุโอกาสการย้ายเมท็อดด้วยทฤษฎีการตัดสินใจ..... 26
ภาพที่ 3.4 ตัวอย่างของรหัสต้นทาง..... 31
ภาพที่ 3.5 ตัวอย่างแผนภาพคลาสของรหัสต้นทาง 31
ภาพที่ 4.1 แผนภาพยูสเคสของระบบการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง..... 37
ภาพที่ 4.2 แผนภาพคลาสแสดงหน้าจอการติดต่อกับเครื่องมือ IMRD..... 42
ภาพที่ 4.3 แผนภาพคลาสการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงของเครื่องมือ IMRD..... 43
ภาพที่ 4.4 แผนภาพลำดับการทำงานแสดงการเพิ่มรหัสโปรแกรมภาษาซีชาร์ปเข้าสู่ระบบ 44
ภาพที่ 4.5 แผนภาพลำดับการทำงานแสดงการทำพาร์เซอร์ 45
ภาพที่ 4.6 แผนภาพลำดับการทำงานแสดงการคำนวณค่ามาตรวัด..... 45
ภาพที่ 4.7 แผนภาพลำดับการทำงานแสดงผลการคำนวณค่ามาตรวัด 46
ภาพที่ 4.8 แผนภาพลำดับการระบุโอกาสการมูฟเมท็อด..... 47
ภาพที่ 4.9 องค์ประกอบของเครื่องมือ IMRD..... 48
ภาพที่ 4.10 องค์ประกอบของ IMRD.Studio.exe 49
ภาพที่ 4.11 องค์ประกอบของ IMRD.CodeDom.dll 50
ภาพที่ ก-1 หน้าจอเมื่อผู้ใช้งานเข้าสู่ระบบ 78

ภาพที่ ก-2 หน้าจอเมื่อผู้ใช้งานเลือกรหัสต้นทาง 79
ภาพที่ ก-3 หน้าจอแสดงรหัสต้นทางที่ผู้ใช้งานเลือก 79
ภาพที่ ก-4 หน้าจอแสดงการระบุโอกาสการมูฟเมท็อด 80
ภาพที่ ก-5 หน้าจอแสดงค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กัน 81
ภาพที่ ก-6 หน้าจอแสดงหน้าต่างค้นหาชื่อตัวแปร เมท็อด และ คลาส 81
ภาพที่ ก-7 หน้าจอแสดงเมื่อผู้ใช้งานยกเลิกรหัสต้นทางชุดเดิม 82
ภาพที่ ก-8 หน้าจอแสดงเมื่อผู้ใช้งานต้องการออกจากโปรแกรม IMRD 82
ภาพที่ ก-9 หน้าจอแสดงรายละเอียดของการจัดทำเครื่องมือ IMRD 83

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

กระบวนการบำรุงรักษาซอฟต์แวร์เป็นกระบวนการหนึ่งซึ่งมีความสำคัญมากในการพัฒนาซอฟต์แวร์เพราะเป็นกระบวนการที่จะต้องมีการปรับปรุง เปลี่ยนแปลง แก้ไข เพิ่มเติม ให้ซอฟต์แวร์มีคุณภาพตรงตามความต้องการของผู้ใช้ โดยการบำรุงรักษาซอฟต์แวร์จะประกอบไปด้วยการเพิ่มเติมความสามารถของซอฟต์แวร์ การแก้ไขข้อผิดพลาด การทำให้ซอฟต์แวร์มีประสิทธิภาพดีขึ้น การปรับปรุงให้เข้ากับสภาพแวดล้อมใหม่ การป้องกันข้อผิดพลาดที่อาจจะเกิดขึ้นในอนาคต ซึ่งวิธีการที่สามารถช่วยให้กระบวนการเปลี่ยนแปลง แก้ไข หรือเพิ่มเติม มีประสิทธิภาพเพิ่มขึ้น คือ การรีแฟคทอริง (Refactoring)

การรีแฟคทอริงเป็นการปรับปรุงเปลี่ยนแปลงโครงสร้างภายในของซอฟต์แวร์โดยไม่ส่งผลกระทบต่อพฤติกรรมภายนอกของซอฟต์แวร์[1] ซึ่งการประยุกต์ใช้รีแฟคทอริงให้เหมาะสมนั้นสามารถทำให้ซอฟต์แวร์มีประสิทธิภาพเพิ่มขึ้นและสามารถลดความซ้ำซ้อนของซอฟต์แวร์ได้ และสามารถทำความเข้าใจซอฟต์แวร์และการปรับปรุงเปลี่ยนแปลงได้ง่ายขึ้น[2],[3]

การปรับปรุงเปลี่ยนแปลงโครงสร้างภายในของซอฟต์แวร์ เพื่อต้องการจะลดความซับซ้อนของซอฟต์แวร์ และต้องการให้ง่ายต่อการเปลี่ยนแปลงแก้ไข ง่ายต่อการบำรุงรักษา ง่ายต่อความเข้าใจ ง่ายในการนำไปพัฒนาต่อ สิ่งที่ต้องพิจารณา คือ การเกาะกลุ่มกัน(Cohesion) และการเข้าคู่กัน (Coupling) ซึ่งเป็นสาเหตุหนึ่งที่ทำให้ซอฟต์แวร์มีความซับซ้อน เปลี่ยนแปลงแก้ไขยาก เพราะถ้าแต่ละคลาสของซอฟต์แวร์มีการเข้าคู่กันมาก เช่น มีการใช้ตัวแปรต่างๆ ในซอฟต์แวร์ร่วมกันมาก เมื่อมีการแก้ไขคลาสหนึ่งก็จะส่งผลกับคลาสอื่นๆ ที่เกี่ยวข้องด้วย หรือในคลาสหนึ่งๆ มีพฤติกรรม (Behavior) หลายพฤติกรรมอยู่ในคลาสเดียวกัน ก็จะทำให้คลาสมีการเกาะกลุ่มกันน้อยลง ดังนั้นถ้าต้องการที่จะลดความซับซ้อนของซอฟต์แวร์ และให้ซอฟต์แวร์มีการเปลี่ยนแปลงแก้ไขและการบำรุงรักษาง่ายขึ้น ก็ต้องพิจารณาให้อยู่ภายใต้กฎที่ว่าการเกาะกลุ่มกันสูง (High cohesion) การเข้าคู่กันต่ำ (Loosely coupling)

การมูฟเมท็อดรีแฟคทอริง (Move Method Refactoring) สามารถช่วยเพิ่มการเกาะกลุ่มกันและลดการเข้าคู่กันได้ โดยพิจารณาจากแรงจูงใจ(Motivation) ของการมูฟเมท็อด คือ เมื่อคลาสมีหลายพฤติกรรมในการทำงาน (การเกาะกลุ่มกันต่ำ) หรือ คลาสมีการทำงานร่วมกันมากเกินไป (การเข้าคู่กันสูง) [1] โดยการใช้มาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันมาช่วยพิจารณาว่าจะย้ายเมท็อดใดไปยังคลาสที่เหมาะสม ถ้ามีเมท็อดที่สามารถจะย้ายไปได้หลายคลาส เป็นสิ่งที่นักพัฒนาซอฟต์แวร์ต้องตัดสินใจว่าจะย้ายเมท็อดนั้นไปที่คลาสใดที่เหมาะสม ที่จะทำให้ค่าระหว่างการเกาะกลุ่มกันและการเข้าคู่กันเป็นค่าที่ดีขึ้นในการย้ายเมท็อดในแต่ละครั้ง ดังนั้นการจะตัดสินใจเลือกค่าที่เหมาะสมที่ได้จากมาตรวัดจึงเป็นเรื่องที่ยาก ซึ่งทฤษฎีการตัดสินใจสามารถช่วยให้นักพัฒนาซอฟต์แวร์ตัดสินใจในการเลือกค่าที่เหมาะสมได้ง่ายขึ้น

ทฤษฎีการตัดสินใจ เป็นทฤษฎีที่มีกระบวนการตัดสินใจ ในการเลือกทางเลือกที่เหมาะสมจากเหตุการณ์ต่างๆ ที่เกิดขึ้นในแต่ละทางเลือก ทฤษฎีการตัดสินใจประสบความสำเร็จ ในด้านอื่น ๆ โดยเฉพาะในสาขาสถิติ เศรษฐศาสตร์ คณิตศาสตร์ และการที่จะประสบความสำเร็จในการเลือก (การเลือกค่าที่ดีที่สุด) ขึ้นอยู่กับการเลือกผลตอบแทนที่มากที่สุดและต้นทุนน้อยที่สุด และนั่นเป็นการใช้ทฤษฎีการตัดสินใจในการตัดสินใจเพื่อเลือกคำตอบที่ต้องการ กระบวนการตัดสินใจ มีรูปแบบของการตัดสินใจภายใต้สภาวการณ์ต่างๆ และในแต่ละสภาวการณ์มีวิธีการตัดสินใจอยู่หลายวิธีด้วยกัน ซึ่งมีอยู่วิธีหนึ่ง ในสภาวการณ์การตัดสินใจภายใต้ความไม่แน่นอน คือวิธีลาปลาส(Laplace Criterion) ที่มองทุกทางเลือกทุกเหตุการณ์ที่เกิดขึ้น มีโอกาสที่จะเป็นทางเลือกที่ดีที่สุดเท่ากันหมด

ดังนั้นงานวิจัยนี้จึงนำทฤษฎีการตัดสินใจ มาใช้พิจารณาในการเลือกคลาสที่เหมาะสมที่ เมท็อดจะย้ายไป โดยการเลือกค่าที่ดีที่เหมาะสมระหว่างการเกาะกลุ่มกันและการเข้าคู่กัน ของการปรับปรุงเปลี่ยนแปลงรหัสต้นทาง เพื่อหาความสมดุล ระหว่างการเกาะกลุ่มกันและการเข้าคู่กันในการมูฟเมท็อดรีแฟคทอริง เพื่อเพิ่มคุณภาพและลดความซับซ้อนของซอฟต์แวร์ โดยใช้วิธีลาปลาส ของทฤษฎีการตัดสินใจ ในสภาวการณ์การตัดสินใจภายใต้ความไม่แน่นอน งานวิจัยนี้มีการพัฒนาเครื่องมือสำหรับการระบุโอกาสของเมท็อดที่จะย้ายไปยังคลาสที่เหมาะสม และทดสอบกับระบบที่พัฒนาด้วยภาษาซีชาร์ป 6 ระบบ ซึ่งสามารถปรับปรุงรหัสต้นทางให้มีคุณภาพที่ดีขึ้นได้ โดยประเมินจากค่าของมาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันก่อนรีแฟคทอริงและหลังรีแฟคทอริงที่ได้จากทุกระบบที่นำมาทดสอบ

1.2 วัตถุประสงค์

1.2.1 เพื่อออกแบบวิธีการระบุโอกาสการปรับปรุงรหัสต้นทางด้วยการย้ายเมท็อดด้วยการใช้วิธีทฤษฎีการตัดสินใจ

1.2.2 เพื่อออกแบบและพัฒนาเครื่องมือวิธีการระบุโอกาสการปรับปรุงรหัสต้นทางด้วยการย้ายเมท็อดด้วยการใช้วิธีทฤษฎีการตัดสินใจ

1.3 ขอบเขตงานวิจัย

1.3.1 งานวิจัยนี้ทำการระบุโอกาสการปรับปรุงคุณภาพของรหัสต้นทาง โดยใช้วิธีมูฟเมท็อดรีแฟคทอริงและใช้ทฤษฎีการตัดสินใจ เป็นการตัดสินใจภายใต้ความไม่แน่นอนของการหาค่าความสมดุลในการปรับปรุงคุณภาพของรหัสต้นทาง

1.3.2 งานวิจัยนี้ใช้ค่ามาตรวัดที่ระบุถึงปริมาณของการเกาะกลุ่มกันและการเข้าคู่กัน ในการปรับปรุงคุณภาพของรหัสต้นทาง

1.3.3 งานวิจัยนี้พัฒนาเครื่องมือที่ใช้ในการระบุโอกาสในการปรับปรุงคุณภาพของรหัสต้นทาง โดยใช้ร่วมกับมาตรวัดที่ระบุถึงปริมาณของการเกาะกลุ่มกันและการเข้าคู่กัน

1.3.4 งานวิจัยนี้จะทดสอบการประยุกต์ใช้ทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงกับระบบซอฟต์แวร์ที่เป็นโอเพนซอร์ส (Open Source) 5 ระบบ โดยแต่ละระบบจะต้องมีจำนวนคลาสตั้งแต่ 3 คลาส ถึง 20 คลาส

1.3.5 ระบบซอฟต์แวร์ที่นำมาใช้ในงานวิจัยนี้จะพิจารณาเฉพาะรหัสต้นทาง ไม่รวมถึงไลบรารี่ที่ใช้ในระบบ

1.4 ขั้นตอนการวิจัย

1.4.1 ศึกษาทฤษฎีการตัดสินใจในสภาวการณ์การตัดสินใจภายใต้ความไม่แน่นอน

1.4.2 ศึกษาวิธีการมูฟเมท็อดรีแฟคทอริง

1.4.3 ศึกษามาตรวัดการเกาะกลุ่มกัน และการเข้าคู่กัน ที่มีใช้อยู่แล้วในปัจจุบัน

1.4.4 ศึกษาวิธีประยุกต์ใช้ทฤษฎีการตัดสินใจในสภาวการณ์การตัดสินใจภายใต้ความไม่แน่นอนกับการมูฟเมท็อดรีแฟคทอริง

1.4.5 ออกแบบเครื่องมือที่ใช้ในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง ในการปรับปรุงคุณภาพของรหัสต้นทาง

1.4.6 จัดทำเครื่องมือ ที่ใช้ในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงในการปรับปรุงคุณภาพของรหัสต้นทาง โดยการมูฟเมท็อดรีแฟคทอริงและมาตรวัดของการเกาะกลุ่มกัน และการเข้าคู่กัน

1.4.7 ประเมินคุณภาพของวิธีการที่นำเสนอในการใช้วิธีของทฤษฎีการตัดสินใจ

1.4.8 สรุปผลของการวิจัยและข้อเสนอแนะ

1.4.9 จัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่จะได้รับ

1.5.1 ได้เครื่องมือที่จัดทำขึ้นเพื่อใช้ในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงในการปรับปรุงคุณภาพของรหัสต้นทาง

1.5.2 ได้ทราบถึงความสัมพันธ์ระหว่างการประยุกต์ใช้ทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง

1.5.3 ได้ทราบถึงข้อดีและข้อเสียในการประยุกต์ใช้ทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง

1.6 คำจำกัดความที่ใช้ในงานวิจัยนี้

1.6.1 เมท็อดเป้าหมาย (Target Method) หมายถึง เมท็อดที่กำลังพิจารณาว่าจะย้ายเมท็อดไปยังคลาสที่มีการเรียกใช้เมท็อดนี้หรือไม่

1.6.2 คลาสเป้าหมาย (Target Class) หมายถึง คลาสที่ได้รับเลือกจากการใช้วิธีที่นำเสนอ ในการย้ายเมท็อดเป้าหมายมายังคลาสนี้

1.6.3 ทางเลือก หมายถึง คลาสที่มีการเรียกใช้เมท็อดเป้าหมาย โดยในคลาสนั้นจะมีการเรียกใช้กี่ครั้งก็ได้นับเป็นหนึ่งทางเลือก (หนึ่งคลาส เท่ากับ หนึ่งทางเลือก)

1.6.4 เหตุการณ์ หมายถึง มาตรวัดที่ใช้วัดในแต่ละคลาส โดยหนึ่งมาตรวัดนับเป็นหนึ่งเหตุการณ์

1.6.5 สถานการณ์ความไม่แน่นอน หมายถึง สถานการณ์ที่ได้จากเหตุการณ์ มีความไม่แน่นอน (ค่าที่ได้จากมาตรวัดในแต่ละคลาส)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.6.6 ย้าย หมายถึง เมท็อดที่จะย้ายจากคลาสที่เมท็อดนั้นอยู่ ไปยังคลาสที่เหมาะสม

1.6.7 ต้นทาง หมายถึง จำนวนการเรียกใช้เมท็อดจากคลาสต้นทางหรือภายในคลาส

1.6.8 ปลายทาง หมายถึง จำนวนการเรียกใช้เมท็อดจากคลาสปลายทางหรือคลาสภายนอก

1.6.9 $U(a_i)$ หมายถึง ค่าเฉลี่ยในของทางเลือกในลำดับที่ i

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องที่ใช้ในงานวิจัยการปรับปรุงคุณภาพของรหัสต้นทาง ได้แก่ รีแฟคทอริง, ทฤษฎีการตัดสินใจ, การวัดซอฟต์แวร์ ซึ่งมีรายละเอียดดังนี้

2.1.1 การรีแฟคทอริง (Refactoring)

การรีแฟคทอริงเป็นการปรับปรุงโครงสร้างภายในของซอฟต์แวร์โดยไม่กระทบต่อพฤติกรรมภายนอก[1] ซึ่งการปรับปรุงจะต้องกระทำกับรหัสต้นทาง (Source Code) ที่มีอยู่แล้ว เพื่อให้รหัสต้นทางนั้นง่ายต่อความเข้า ง่ายในการนำไปพัฒนาต่อ ง่ายในการค้นพบข้อผิดพลาด ช่วยให้โปรแกรมสามารถทำงานได้ดีขึ้น และเพิ่มความสามารถในการบำรุงรักษาซอฟต์แวร์ ซึ่งวิธีการในการทำรีแฟคทอริง มี 72 วิธี แต่ในหัวข้อนี้ข้ออธิบาย มูฟเมท็อดรีแฟคทอริงซึ่งใช้ในงานวิจัยนี้

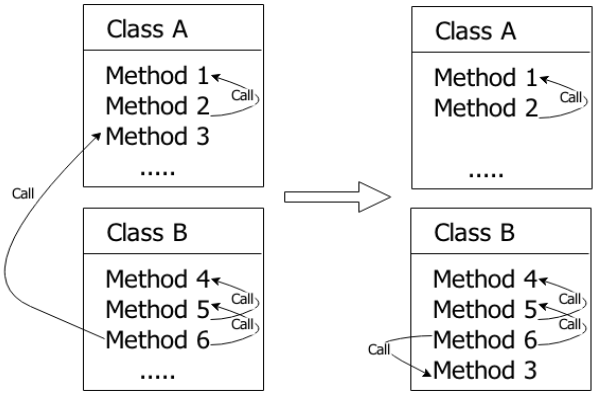
มูฟเมท็อดรีแฟคทอริง

เป็นการย้ายเมท็อดจากคลาสเดิมที่เมท็อดนั้นอยู่ ไปยังคลาสอื่น เมื่อคลาสอื่นมีการเรียกใช้เมท็อดนั้น มากกว่าคลาสเดิมที่เมท็อดนั้นอยู่ โดยเมื่อย้ายเมท็อดแล้วก็แก้ไขรหัสต้นทางให้มีการเรียกใช้เมท็อดนั้นใหม่ หลังจากนั้นจึงลบเมท็อดนั้นที่อยู่ในคลาสเดิมออก

แรงจูงใจการมูฟเมท็อดรีแฟคทอริง

แรงจูงใจในการมูฟเมท็อด คือ เมื่อคลาสมีหลายพฤติกรรมในการทำงาน (การเกาะกลุ่มกันต่ำ) หรือ หลายคลาสมีการทำงานร่วมกันมากเกินไป (การเข้าคู่กันสูง) [1]

ภาพที่ 2.1 แสดงให้เห็นว่าคลาส B มีการเรียกใช้ Method3 ของคลาส A ซึ่งคลาส A มีการเรียกใช้ Method3 ภายในคลาส A น้อยดังนั้นจึงสร้าง Method3ในคลาส B ใหม่ โดยการสร้างใหม่นี้สามารถเปลี่ยนชื่อเมท็อดได้ หลังจากนั้นคัดลอกรหัสต้นทางของ Method3 ในคลาส A ไปยัง Method3 ในคลาส B แล้วแก้ไขการอ้างอิงของ Method3 ในคลาส B และสุดท้ายทำการลบ Method3 ในคลาส A



ภาพที่ 2.1 ตัวอย่างการ Move Method ระหว่างคลาส A และ คลาส B

2.1.2 ทฤษฎีการตัดสินใจ (Decision Theory)

ทฤษฎีการตัดสินใจมีกระบวนการตัดสินใจ ในการเลือกทางเลือกการปฏิบัติ เพื่อให้บรรลุวัตถุประสงค์ที่วางไว้ จึงจำเป็นต้องทำการตัดสินใจเลือกทางเลือก ในการแก้ปัญหาที่เหมาะสม หรือให้เป็นไปตามวัตถุประสงค์ที่วางไว้มากที่สุด

โครงสร้างของการตัดสินใจประกอบด้วย[4],[5]

- ทางเลือก (Action) ทางเลือกในการตัดสินใจที่มีมากกว่า 1 ทางเลือก ระบุทางเลือกที่เป็นไปได้ทั้งหมด
- สภาวการณ์นอกบังคับ (State of nature or Events) ในแต่ละทางเลือกมีสภาวการณ์ที่เกี่ยวข้องที่ไม่สามารถควบคุมได้มากกว่า 1 สภาวการณ์
- ผลลัพธ์หรือผลตอบแทน (Outcomes or Value of Payoff)
- โอกาสความเป็นไปได้มากที่สุดในแต่ละสถานการณ์ (Likelihood for Each of Environmental State)
- เกณฑ์ในการตัดสินใจ (Decision Criterion)

จากโครงสร้างของการตัดสินใจเบื้องต้น นำมาพิจารณาประกอบกันภายใต้ แต่ละทางเลือก แต่ละสภาวการณ์ โดยกำหนดผลลัพธ์(Outcome or Value of Payoff) ของแต่ละทางเลือก ผลลัพธ์ดังกล่าวอยู่ในรูปของตาราง เรียกว่าตารางการตัดสินใจ (Decision Table or Decision Matrix)

ตารางที่ 2.1 ตัวอย่างตารางตัดสินใจ

ทางเลือก (Actions)	เหตุการณ์ (Events)				
1	1	2	3	...	n
2	C_{11}	C_{12}	C_{13}		C_{1n}
3	C_{21}	C_{22}	C_{23}		C_{2n}
...					
m	C_{m1}	C_{m2}	C_{m3}		C_{mn}

จากตารางที่ 2.1 ค่า C_{ij} ในตารางตัดสินใจอาจจะแสดงได้ในรูปของกำไร (Profit) หรือต้นทุน (Cost) [5]

รูปแบบของการตัดสินใจภายใต้สภาวการณ์ต่างๆ

เมื่อรวบรวมปัญหาที่ต้องการตัดสินใจให้อยู่ในรูปของตารางตัดสินใจ โดยตารางดังกล่าวแสดงถึง ทางเลือกเหตุการณ์ ที่จะเกิดขึ้นและผลตอบแทนของทางเลือกต่างๆ โดยขั้นตอนต่อไปเราจะเลือกทางเลือกที่เหมาะสมในการแก้ปัญหาที่เกิดขึ้น ผู้ตัดสินใจจะต้องทราบว่าปัญหาที่กำลังเกิดขึ้น อยู่ภายใต้สภาวการณ์ใด โดยทั่วไปสามารถแบ่งตัวแบบการตัดสินใจภายใต้สภาวการณ์ต่างๆดังนี้[6],[7]

- การตัดสินใจภายใต้ความแน่นอน (Decision making under certainty)

การตัดสินใจซึ่งผู้ที่ทำการตัดสินใจมีความสามารถในการพยากรณ์เหตุการณ์ในอนาคตได้อย่างแน่นอน เพราะข้อมูลของปัญหาประเภทนี้จะมีทางเลือกอยู่หลายทาง และแต่ละทางเลือกจะเกิดขึ้นในสภาวการณ์ที่แน่นอนเพียงอย่างเดียว

- การตัดสินใจภายใต้ความเสี่ยง (Decision making under risk)

การตัดสินใจซึ่งผู้ที่ทำการตัดสินใจไม่ทราบแน่ชัดว่าเหตุการณ์ใดจะเกิดขึ้นอย่างแน่นอน แต่ผู้ตัดสินใจทราบโอกาสหรือความน่าจะเป็น (Probability) ที่จะเกิดเหตุการณ์ต่างๆ ได้โดยอาศัยข้อมูลในอดีต ประสบการณ์ของผู้ตัดสินใจ

ดังนั้นการตัดสินใจภายใต้ความเสี่ยง ผู้ที่ทำการตัดสินใจจะต้องทราบทางเลือกและเหตุการณ์ที่จะเกิดขึ้นในอนาคตแล้ว โดยผู้ตัดสินใจจะต้องหาความน่าจะเป็นของเหตุการณ์ต่างๆ เพื่อใช้ประกอบการตัดสินใจ ในตัวแบบการตัดสินใจหรือปัญหาการตัดสินใจการหาความน่าจะเป็นของเหตุการณ์ต่างๆ แบ่งเป็น 2 ประเภท คือ

a) ความน่าจะเป็นที่ได้มาอย่างมีหลักเกณฑ์ (Objective probability)

เป็นความน่าจะเป็นโดยใช้ข้อมูลในอดีตที่เคยเกิดขึ้น นั่นคือใช้ความถี่ของการเกิดในแต่ละเหตุการณ์มาคำนวณหาความน่าจะเป็น

b) ความน่าจะเป็นแบบกำหนดเอง (Subjective probability)

เป็นความน่าจะเป็นโดยผู้ตัดสินใจเป็นผู้กำหนดขึ้นมาเอง เพราะบางปัญหาเป็นปัญหาใหม่ที่ไม่เคยเกิดขึ้นมาก่อน ผู้ตัดสินใจจึงอาจต้องใช้ประสบการณ์ตัดสิน

- การตัดสินใจภายใต้ความไม่แน่นอน (Decision making under uncertainty)

การตัดสินใจซึ่งผู้ที่ทำการตัดสินใจเลือกทางเลือกโดยไม่ทราบความน่าจะเป็นของแต่ละเหตุการณ์ที่เกิดขึ้น ปัญหาการตัดสินใจแบบนี้มักเป็นปัญหาของธุรกิจใหม่ที่ไม่มีข้อมูลเดิมอยู่เลยหรือมีน้อยมาก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เกณฑ์การตัดสินใจด้วยวิธีลาปลาส (Laplace Criterion or Equally likely)

วิธีนี้มีแนวคิดว่าถ้าไม่ทราบความน่าจะเป็น หรือโอกาสของเหตุการณ์ที่เกิดขึ้น ก็ควรกำหนดให้เหตุการณ์เหล่านั้นมีความน่าจะเป็นที่จะเกิดขึ้นเท่ากัน (equally likely) ทั้งหมด เป็นการหาทางเลือกที่มีผลตอบแทนเฉลี่ยสูงที่สุดในแต่ละทางเลือก แล้วเลือกที่ให้ค่ามากที่สุด โดยมีขั้นตอนการวิเคราะห์ดังนี้

1) กำหนดความน่าจะเป็นของแต่ละเหตุการณ์เท่าๆกัน

2) คำนวณค่าคาดหมายของแต่ละทางเลือกด้วยการรวมผลคูณระหว่างผลตอบแทนกับค่าความน่าจะเป็นในแต่ละเหตุการณ์

3) เลือกทางเลือกที่ให้ค่าคาดหมายสูงที่สุด

$$U(a_i) = \frac{1}{m}(u_{i1} + u_{i2} + ... + u_{im}) \qquad (1)$$
$$\max\{U(a_i)\}$$

ให้ a คือ ทางเลือก

m คือ จำนวนเหตุการณ์ทั้งหมด

U คือ ฟังก์ชันผลตอบแทน

$U(a_i)$ คือ ค่าเฉลี่ยของทางเลือกในลำดับที่ i

$\max\{U(a_i)\}$ คือ การเลือกค่าเฉลี่ยที่มากที่สุดจากทุกทางเลือกเป็นคำตอบของวิธีลาปลาส

ตัวอย่าง บริษัทผลิตและจำหน่ายเครื่องเสียง ต้องการจะเปิดสาขาใหม่ โดยบริษัทกำลังตัดสินใจจะเช่าสถานที่ตั้ง ซึ่งมีทางเลือกอยู่ 3 ทางคือ ทางเลือกที่ 1 สถานที่ตั้ง 1,600 ตร.ฟุต ทางเลือกที่ 2 สถานที่ตั้ง 8,000 ตร.ฟุต ทางเลือกที่ 3 ไม่เปิดสาขา โดยบริษัทได้คาดคะเนกำไร-ขาดทุน ซึ่งตั้งไว้ 2 เหตุการณ์ คือ ยอดขายสูงและยอดขายต่ำ[5]

ตารางที่ 2.2 การตัดสินใจการจะเปิดสาขาใหม่ของบริษัทเครื่องเสียง

ทางเลือก (Actions)	เหตุการณ์ (Events)		$U(a_i)$
	ยอดขายสูง(กำไร)	ยอดขายต่ำ (กำไร)	
1,600 ตร.ฟุต	50,000	10,000	30,000
8,000 ตร.ฟุต	150,000	-15,000	67,000
ไม่เปิดสาขา	0	0	0

ด้วยวิธีของลาปลาส บริษัทจะตัดสินใจเลือกสถานที่ 8,000 ตร.ฟุต

2.1.3 มาตรวัดซอฟต์แวร์เชิงวัตถุ (Object-Oriented Software Metrics)

มาตรวัดซอฟต์แวร์เชิงวัตถุ เป็นเงื่อนไขหรือข้อกำหนดที่จะใช้ในการตรวจสอบซอฟต์แวร์ที่สร้างขึ้น โดยมาตรวัด(Metric) นี้เป็นส่วนหนึ่งของการวัด (Measurement) เป็นเสมือนตัวชี้วัดคุณภาพของระบบซอฟต์แวร์ เพื่อเป็นการสะท้อนให้เห็นถึงคุณภาพของซอฟต์แวร์และการออกแบบ ทั้งการตรวจสอบและติดตามคุณภาพของซอฟต์แวร์ เพื่อที่จะสามารถควบคุมคุณภาพของซอฟต์แวร์ได้อย่างมีประสิทธิภาพ โดยมาตรวัดที่นำมาใช้ในงานวิจัยนี้เป็นมาตรวัดเกี่ยวกับการเข้าคู่กันและการเกาะกลุ่มกันของรหัสต้นทาง ระหว่างเมท็อด คลาส ตัวแปร ซึ่งแสดงรายละเอียดดังนี้

2.1.3.1 มาตรวัดการติดต่อปฏิสัมพันธ์ระหว่างเมท็อด (Call-based Interaction between Methods- CIM)[8]

CIM คือ มาตรวัดการเข้าคู่กันระหว่างเมท็อดของคลาส 2 คลาส โดยพิจารณาการ เรียกใช้ระหว่างกัน โดย CIM มีค่าอยู่ในช่วง [0, 1]

- ถ้า CIM มีค่าเท่ากับ 0 แสดงว่า m_i ไม่มีการเรียก m_j
- ถ้า CIM มีค่าเท่ากับ 1 แสดงว่า m_i มีการเรียก m_j (มีการเข้าคู่กันสูง)

$$CIM_{i \to j} = \frac{call(m_i, m_j)}{call_{in}(m_j)} \quad (2)$$

เมื่อ m คือ เมท็อด

$call(m_i, m_j)$ คือ จำนวนครั้งที่เมท็อด m_i มีการเรียกใช้เมท็อด m_j

$call_{in}(m_j)$ คือ จำนวนการเรียกใช้เมท็อด m_j ทั้งหมด

2.1.3.2 มาตรวัดความสัมพันธ์การเข้าคู่กันของเมท็อด (Relative Method Coupling - RMC)[9]

RMC คือ การวัดการเข้าคู่กันระหว่างคลาส โดยวัดความสัมพันธ์ของเมท็อดที่มีการเรียกใช้ตัวแปรและเมท็อดจากคลาสอื่น และการเรียกใช้ตัวแปรและเมท็อดภายในคลาส

$$RMC_m^{C_i} \rightarrow C_j = \frac{A_{C_j \leftarrow m} + M_{C_j \leftarrow m}}{\max(1, A_{C_i \leftarrow m} + M_{C_i \leftarrow m})} \quad (3)$$

เมื่อ m คือ เมท็อดของคลาส C_i

$RMC_m^{C_i} \rightarrow C_j$ คือ จำนวนตัวแปรและเมท็อดของคลาส C_j ที่มีการเรียกใช้ด้วยเมท็อด m ของคลาส C_i

$A_{C_j \leftarrow m}$ คือ จำนวนตัวแปรในคลาส C_j ที่เรียกใช้ทางตรงและทางอ้อมด้วยเมท็อด m ของคลาส C_i

$A_{C_i \leftarrow m}$ คือ จำนวนตัวแปรภายในคลาส C_i ที่เรียกใช้ทางตรงและทางอ้อมด้วยเมท็อด m ในคลาส C_i (เรียกใช้ภายในคลาสตัวเอง)

$M_{C_j \leftarrow m}$ คือ จำนวนเมท็อดในคลาส C_j ที่เรียกใช้ทางตรงและทางอ้อมด้วยเมท็อด m ของคลาส C_i

$M_{C_i \leftarrow m}$ คือ จำนวนเมท็อดภายในคลาส C_i ที่เรียกใช้ทางตรงและทางอ้อมด้วยเมท็อด m ในคลาส C_i (เรียกใช้ภายในคลาสตัวเอง)

2.1.3.3 มาตรวัดการตอบสนองต่อคลาส (Response for a class - RFC)[10]

RFC คือ จำนวนเมท็อดที่สามารถตอบสนองกับการเรียกใช้งาน รวมถึงเมท็อดภายนอกคลาสซึ่งถูกเรียกใช้ด้วย

โดย Response Set (RS) คือ เซตของเมท็อดที่ถูกเรียกใช้งาน เมื่อเมสเสจของออบเจกต์ของคลาสที่พิจารณาถูกเรียกใช้งาน (นับจำนวนเมท็อดที่คลาสมีการเรียกใช้จาก ภายในคลาสและภายนอกคลาส)

$$RS = \{M\} \cup all\ i\{R_i\} \quad (4)$$

เมื่อ $\{M\}$ คือ เซตของเมท็อดในคลาสที่พิจารณา

$\{R_i\}$ คือ เซตของเมท็อดที่ถูกเรียกโดยเมท็อด i

แล้ว $RFC = |RS|$

2.1.3.4 มาตรวัดการเข้าคู่กันระหว่างวัตถุ (Coupling between objects - CBO) [10]

CBO คือ การแสดงความสัมพันธ์ระหว่างวัตถุ เมื่อมีการเรียกใช้ตัวแปรหรือเมท็อดระหว่างกัน (นับจำนวนคลาส)

2.1.3.5 มาตรวัดระดับของการขาดการเกาะกลุ่มกันของเมท็อดภายในคลาส (Lack of cohesion of methods - LCOM) [11],[12],[13]

LCOM คือ เมท็อดนั้นมีความสัมพันธ์กับ เมท็อดอื่นโดยมีการอ้างอิงตัวแปรร่วมกัน การที่มีค่าของ LCOM ต่ำ (การขาดการเกาะกลุ่มกันต่ำ (Low lack of cohesion)) แสดงว่าคลาสนั้นมีการออกแบบที่ดี

$$LCOM = \begin{cases} |P| - |Q| & if\ |P| > |Q|; \\ 0 & otherwise. \end{cases}$$
$$P = \{(I_i, I_j) \mid I_i \cap I_j = \phi\} \quad (5)$$
$$Q = \{(I_i, I_j) \mid I_i \cap I_j \neq \phi\}$$

เมื่อ P คือ ชุดของคู่ของเมท็อดที่ไม่ใช้ตัวแปรร่วมกัน

Q คือ ชุดของคู่ของเมท็อดที่ใช้ตัวแปรร่วมกันอย่างน้อยหนึ่งตัวแปร

I_i คือ ชุดของตัวแปรที่ถูกเรียกใช้ในเมท็อด m_i

I_j คือ ชุดของตัวแปรที่ถูกเรียกใช้ในเมท็อด m_j

2.1.3.6 มาตรวัดการเกาะกลุ่มกันหนาแน่น (Tight Class Cohesion -TCC)[14]

TCC คือ จำนวนความสัมพันธ์ของเมท็อดที่สัมพันธ์กันทางตรง โดยค่าที่ได้จะอยู่ในช่วง [0, 1] ถ้าเข้าใกล้ 1 แสดงว่าเมท็อดมีการเกาะกลุ่มกันสูง ซึ่งคำนวณได้จาก

$$TCC = \frac{NDC(C)}{NP(C)} \quad (6)$$

เมื่อ Number of direct connection (NDC(C)) คือ จำนวนของความสัมพันธ์ทางตรงระหว่างเมท็อดภายในคลาส C นั่นคือ เมท็อด 2 เมท็อดที่จะมีความสัมพันธ์ทางตรง

เมื่อทั้ง 2 เมท็อดมีการใช้งานคุณลักษณะตัวเดียวกันของคลาส แต่ไม่รวมถึงความสัมพันธ์ระหว่างเมท็อดกับคอนสตรัคเตอร์

Number of pairs of abstracted methods (NP) คือ ค่าที่เป็นไปได้มากที่สุดของความสัมพันธ์ทั้งทางตรง และทางอ้อมในคลาส คำนวณได้จาก

$$\frac{N(N-1)}{2} \quad (7)$$

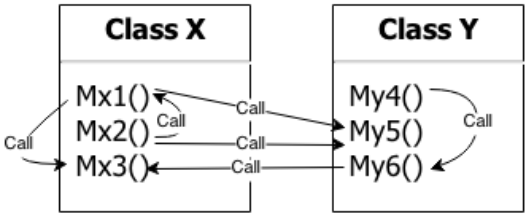
โดยที่ N คือจำนวนเมท็อดทั้งหมดภายในคลาส ยกเว้นคอนสตรัคเตอร์

ตัวอย่างการคำนวณมาตรวัด

```
public class X
{
    int a, b, c, d, e f;
    Y callY = new Y ();
    private void Mx1 ()
    {
        Mx3();
        callY.My5();
    }
    private void Mx2()
    {
        ... uses d, e ...
        Mx1();
        callY.My5();
    }
    private void Mx3()  { ... uses e, f ... }
}
```

```
public class Y
{
    int g, u, k;
    private void My4 ()
    {
        ... uses k, u;
        My6();
    }
    public void My5() { ... uses k, u..}
    public void My6()
    {
        ... uses u, g..
        X callX = new X ();
        callX. Mx3();
    }
}
```

ภาพที่ 2.2 ตัวอย่างของรหัสต้นทาง



ภาพที่ 2.3 ตัวอย่างแผนภาพคลาสของรหัสต้นทาง

ตัวอย่างการคำนวณมาตรวัด *CIM*

$call(Mx1, My5)$ = 1

$call_{in}(My5)$ = 2

$CIM_{1\to5} = \dfrac{call(Mx1, My5)}{call_{in}(My5)}$ จากสูตรจะได้ $CIM_{1\to5}$ = 0.5

ตัวอย่างการคำนวณมาตรวัด *RMC*

$A_{Y\leftarrow Mx1} + M_{Y\leftarrow Mx1}$ = 0+1 = 1

$\max(1, A_{X\leftarrow Mx1} + M_{X\leftarrow Mx1})$ = 1

$RMC^{X}_{Mx1} \to Y = \dfrac{A_{Y\leftarrow Mx1} + M_{Y\leftarrow Mx1}}{\max(1, A_{X\leftarrow Mx1} + M_{X\leftarrow Mx1})}$ จากสูตรจะได้ $RMC^{X}_{M1} \to Y$ = 1

ตัวอย่างการคำนวณมาตรวัด *CBO*

จาก class X จะได้ CBO = 1

ตัวอย่างการคำนวณมาตรวัด *LCOM*

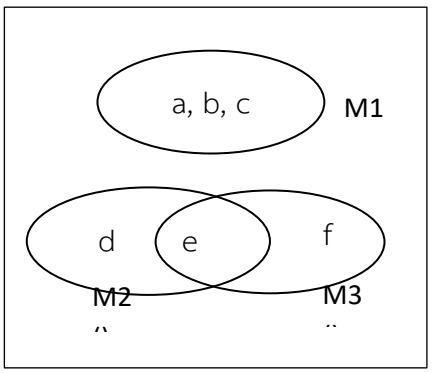
จาก class X จะได้

<M1, M2> = P (ไม่มีการใช้ตัวแปรร่วมกัน)

<M1, M3> = P (ไม่มีการใช้ตัวแปรร่วมกัน)

<M2, M3> = Q (มีการใช้ตัวแปร e ร่วมกัน)

จากสูตรจะได้ LCOM = 2-1 = 1 (คลาส X มีค่า LCOM = 1)



ภาพที่ 2.4 ตัวอย่างการเรียกใช้ตัวแปรของเมท็อดในคลาส

ตัวอย่างการคำนวณมาตรวัด *TCC*

จาก class X จะได้

<M1, M2> = 0 (ไม่มีการใช้ตัวแปรร่วมกัน)

<M1, M3> = 0 (ไม่มีการใช้ตัวแปรร่วมกัน)

<M2, M3> = 1 (มีการใช้ตัวแปร e ร่วมกัน)

NDC(X) = 1

NP = 3(3-1) / 2 = 3

จากสูตรจะได้ TCC = 1 / 3 = 0.33 (คลาส X มีค่า TCC = 0.33)

ตัวอย่างการคำนวณมาตรวัด *RFC*

จาก class X, $\{M\}$ = 3

$\{R_i\}$ = 2+2+0 = 4, $|RS|$=3+4 จะได้ RFC = 7

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.4 การแปลงข้อมูล (Data Transformation)[15]

การแปลงข้อมูล คือ การทำนอร์มอลไลซ์ (Normalization) โดยการแปลงค่าข้อมูลให้อยู่ในช่วงสั้นๆ เช่น ช่วง 0..1 เพื่อให้กลุ่มข้อมูลที่นำมาพิจารณาอยู่ในช่วงเดียวกัน และสามารถนำไปใช้ประมวลผลได้ วิธีการทำนอร์มอลไลซ์ข้อมูล ได้แก่ Min-Max Normalization

Min-Max Normalization

$$V' = \frac{V - \min_A}{\max_A - \min_A}(new_\max_A - new_\min_A) + new_\min_A \qquad (10)$$

เมื่อ V คือค่าคุณลักษณะเดิม

V' คือค่าคุณลักษณะใหม่

$\max_A$, $\min_A$ คือค่าต่ำสุดและสูงสุดเดิมของคุณลักษณะ A

$new_\max_A$, $new_\min_A$ คือค่าต่ำสุดและสูงสุดใหม่ของคุณลักษณะ A

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Suggesting Extract Class Refactoring Opportunities by Measuring Strength of Method Interactions [16]

งานวิจัยนี้ได้เสนอมาตรวัดใหม่ ที่ช่วยแนะนำโอกาสในการ เอ็กซ์แทร็กท์คลาสรีแฟคทอริง (Extract Class Refactoring) โดยใช้มาตรวัดที่ผู้วิจัยเสนอใช้ในการวัดการปฏิสัมพันธ์อย่างเหนียวแน่นของเมท็อดในคลาส บนพื้นฐานการเรียกใช้พารามิเตอร์ และ คุณลักษณะระหว่างกัน โดยมาตรวัด จะวัดต้นทุน โดยนับจากจำนวนขนาดของวัตถุที่มีการอ้างอิงถึงกัน ผู้วิจัยได้แบ่งเมท็อดเป็น 2 ชุด คือ เมท็อดที่มีการปฏิสัมพันธ์กันต่ำ และเมท็อดที่มีการปฏิสัมพันธ์กันสูง หลังจากได้เมท็อดที่มีการปฏิสัมพันธ์กันสูงผู้วิจัยได้ทำการแยกจากคลาสเดิมไปคลาสใหม่

ผู้เขียนงานวิจัยนี้สรุปว่ามาตรวัดที่ผู้วิจัยนำเสนอสามารถจับการปฏิสัมพันธ์กันระหว่างเมท็อดในคลาส โดยทำการทดสอบกับระบบ JHOTDRAW เพื่อที่จะประเมินความสามารถของการบำรุงรักษาและความสามารถของมาตรวัดที่ได้นำเสนอในการแนะนำโอกาสการรีแฟคทอริง โดยใช้มาตรวัดความซับซ้อน คือ WMC และมาตรวัดการเกาะกลุ่มกัน คือ LCOM โดยทำการเปรียบเทียบก่อนและหลังการใช้กระบวนการรีแฟคทอริง พบว่าหลังจากผ่านกระบวนการรีแฟคทอริงด้วยวิธีที่

นำเสนอสามารถทำให้ความซับซ้อน ลดลงและสามารถเพิ่มการเกาะกลุ่มกัน วิทยานิพนธ์นี้นำแนวคิดการเปรียบเทียบการใช้มาตรวัดการเกาะกลุ่มกัน ก่อนและหลังการรีแฟคทอริงมาใช้

2.2.2 Move Code Refactoring with Dynamic Analysis [17]

งานวิจัยนี้นำเสนอเทคนิคการวิเคราะห์แบบไดนามิก (Dynamic analysis) ในการตรวจสอบรีแฟคทอริงที่เป็นไปได้ โดยการวิเคราะห์ร่องรอยในระหว่างที่โปรแกรมประมวลผล เพื่อรวบรวมและวิเคราะห์ข้อมูลที่เกิดขึ้นจากการปฏิบัติงานจริงที่ได้ระหว่างการประมวลผล ในการที่จะระบุโอกาสในการรีแฟคทอริงซึ่งเน้นเฉพาะการมูฟ รหัสต้นทาง งานวิจัยนี้ได้พัฒนาเครื่องมือต้นแบบ ที่สนับสนุนเทคนิคที่ผู้วิจัยได้นำเสนอ โดยผู้วิจัยได้แบ่งขั้นตอนของเทคนิคที่นำเสนอออกเป็น 5 ขั้นตอน คือ วิธีการตรวจสอบ การแบ่งวิธีการติดตาม การนับเมท็อดที่มีการเรียกใช้ แสดงข้อมูลของการนับในแต่ละช่วง ตรวจหารีแฟคทอริงที่เป็นไปได้ในช่วงที่นับได้ โดยผู้วิจัยได้แบ่งเมท็อดในคลาสที่มีการเรียกใช้ระหว่างกัน และทำการนับการเรียกใช้จากเมท็อดหนึ่งไปยังอีกเมท็อดหนึ่ง จากนั้นผู้วิจัยกำหนดสีตามการเรียกใช้ในแต่ละแบบ แล้วระบบจะเติมสีตามการเรียกใช้ที่ผู้วิจัยกำหนด ถ้ามีการเรียกใช้ที่ผิดปกติผิดรูปแบบที่ผู้วิจัยกำหนดไว้ระบบจะเติมสีที่ผิดปกติที่ผู้วิจัยกำหนดไว้ เพื่อบอกว่าเมท็อดนั้นเป็นเมท็อดที่ต้องมูฟเมท็อดรีแฟคทอริง ผู้วิจัยสรุปว่าจากการทดสอบกับซอฟต์แวร์จริง 2 ระบบที่ใช้ในงานวิจัยนี้ เทคนิคการวิเคราะห์แบบไดนามิกที่ผู้วิจัยนำเสนอสามารถตรวจสอบรีแฟคทอริงที่เป็นไปได้ในการมูฟ วิทยานิพนธ์นี้นำแนวคิดการนับเมท็อดที่มีการเรียกใช้จากคลาสอื่นมาใช้

2.2.3 Playing with Refactoring Identifying Extract Class Opportunities through Game Theory [18]

งานวิจัยนี้ใช้ทฤษฎีเกมที่จะแนะนำโอกาสในการ เอ็กซ์แทร็กท์คลาสรีแฟคทอริง ในการหาสมดุลระหว่างการเข้าคู่กันและการเกาะกลุ่มกันเมื่อมีการแยกคลาสลำดับการดำเนินการรีแฟคทอริงจะคำนวณโดยใช้ทฤษฎีเกม ซึ่งอยู่ในสมดุลของแนช (Nash Equilibrium)

ผู้ทำการวิจัยประยุกต์ใช้กับเกมความลำบากใจของนักโทษ (Prisoner's dilemma) โดยเป็นเกมที่มีผู้เล่น 2 คนที่มีทางเลือก 2 ทาง โดยเกมมีลักษณะที่ผู้เล่นทั้งสองฝ่ายพยายามเลือกทางเลือกที่ได้ผลตอบแทนมากที่สุด แต่กลับทำให้ผลตอบแทนรวมที่ได้ต่ำลง มีสถานการณ์ดังนี้ มีโบรกเกอร์ 2 คน คือ แซลลี่และทอม ที่ถูกกล่าวหาว่ามีการซื้อขายฉ้อโกง ทั้ง 2 คน ถูกจับและถูกสอบปากคำแยกทีละคน และไม่รู้ สิ่งที่อีกคนให้ปากคำ แต่โบรกเกอร์ทั้ง 2 ต้องการลดจำนวนเวลาที่ต้องอยู่ในเรือนจำ

โดยแซลลี่และทอมมี 2 ทางเลือก คือ สารภาพและไม่สารภาพ ซึ่งถ้าทอมและแซลลี่สารภาพทั้งคู่จะถูกจำคุก 6 ปีเท่ากัน ถ้าทอมสารภาพและแซลลี่ไม่สารภาพ ทอมจะถูกจำคุกคนเดียว 7 ปี ถ้าทอมไม่

สารภาพและแซลลี่สารภาพ แซลลี่จะถูกจำคุกคนเดียว 7 ปี ถ้าทอมและแซลลี่ไม่สารภาพทั้งคู่จะถูกจำคุก 4 ปีเท่ากัน กลยุทธ์ที่พวกเขาสามารถเลือกได้ดังตารางที่ 2.3

ตารางที่ 2.3 ผลตอบแทนเมทริกซ์สำหรับเกมความลำบากใจของนักโทษ

		ทอม	
		สารภาพ	ไม่สารภาพ
แซลลี่	สารภาพ	(6,6)	(0,7)
	ไม่สารภาพ	(7,0)	(4,4)

(6, 6) คือ สมดุลของแนช (Nash Equilibrium)

การสร้างเกณฑ์ในการเลือกทางออกที่ดีที่สุดสำหรับเกมความลำบากใจของนักโทษ ในความเป็นจริงถ้าเลือกแก้ปัญหาด้วยยอดรวมที่ต่ำสุดหรือผลตอบแทนรวมสูงสุดควรเลือก(สารภาพ,ไม่สารภาพ) และ (ไม่สารภาพ,สารภาพ) ซึ่งหนึ่งในโบรกเกอร์ไม่ติดคุก แต่เมื่อโบรกเกอร์ทั้งสองคนเลือกทางเลือกนี้ กลับให้ผลที่ไม่น่าพอใจ ถึงแม้ผู้เล่นจะทราบว่าผลตอบแทนที่ดีจะเกิดขึ้นเมื่อทั้งสองฝ่ายไม่รับสารภาพ แต่ทั้งคู่อาจจะไม่กล้าทำ เพราะไม่ไว้ใจอีกฝ่ายหนึ่งว่าจะรับสารภาพหรือไม่ ในสถานการณ์เช่นนี้สมดุลของแนชจะช่วยแก้ปัญหาในการประนีประนอมและยุติธรรมทั้ง 2 ฝ่าย

ผู้เขียนงานวิจัยนี้สรุปว่าการประยุกต์ใช้ทฤษฎีเกมเพื่อสนับสนุน เอ็กซ์แทร็กท์คลาสรีแฟคทอริงด้วยวิธีเกมความลำบากใจของนักโทษซึ่งอยู่ในสมดุลของแนช ประสบความสำเร็จในการประเมินเบื้องต้นเมื่อเปรียบเทียบกับวิธีจุดที่เหมาะสมที่สุดของปาเรโต (Pareto optimum) วิทยานิพนธ์นี้นำแนวคิดการใช้มาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันมาใช้ ในการตัดสินใจ สำหรับการย้ายเมท็อดมาใช้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 3
การระบุโอกาสสำหรับการรีแฟคทอริงแบบมูฟเมท็อดด้วยวิธีทฤษฎีการตัดสินใจ

งานวิจัยนี้มีการนำเสนอมาตรวัดใหม่ 2 มาตรวัดเพื่อใช้ในการระบุโอกาสการย้ายเมท็อดไปยังคลาสที่เหมาะสม โดยมาตรวัดที่นำเสนอคือ มาตรวัดความสัมพันธ์การเข้าคู่กันระหว่างคลาสและมาตรวัดความสัมพันธ์การเข้าคู่กันระหว่างเมท็อด

3.1 มาตรวัดการเข้าคู่กัน

มาตรวัดใหม่ที่นำเสนอในงานวิจัยนี้เป็นมาตรวัดเกี่ยวกับการเข้าคู่กันของรหัสต้นทาง ระหว่างเมท็อด คลาส ตัวแปร ซึ่งแสดงรายละเอียดดังนี้

3.1.1 มาตรวัดความสัมพันธ์การเข้าคู่กันระหว่างคลาส (Interaction coupling between classes-ICBC)

ICBC คือ มาตรวัดการเรียกใช้กันระหว่างคลาส 2 คลาส ถ้า ICBC เท่ากับ 0 แสดงว่าไม่มีการเรียกใช้งานจากคลาสอื่น

$$ICBC = \frac{Interaction(C_i, C_j) + Interaction(C_j, C_i)}{2} \quad (8)$$

เมื่อ $Interaction(C_i, C_j)$ คือ จำนวนการเรียกใช้ ของคลาส C_i ที่มีการเรียกใช้ตัวแปรและเมท็อดของคลาส C_j

$Interaction(C_j, C_i)$ คือ จำนวนการเรียกใช้ ของคลาส C_j ที่มีการเรียกใช้ตัวแปรและเมท็อดของคลาส C_i

3.1.2 มาตรวัดความสัมพันธ์การเข้าคู่กันระหว่างเมท็อด (Interaction method coupling - IMC)

IMC คือ มาตรวัดจำนวนการเรียกใช้เมท็อดภายในและการเรียกใช้เมท็อดจากภายนอก โดย IMC มีค่าอยู่ในช่วง [0...1]

- ถ้า IMC มีค่าเท่ากับ 0 แสดงว่าคลาส ไม่มีการเรียกใช้จากคลาสอื่น

- ถ้า IMC มีค่าเท่ากับ 1 แสดงว่าคลาส มีการเรียกใช้จากคลาสอื่นและไม่มีการเรียกใช้ภายในคลาสเดียวกัน

$$\text{IMC} = \frac{call(C_j, m)}{call(C_i, m) + call(A_{C_i}, m) + call(C_j, m)} \quad (9)$$

เมื่อ m คือ เมท็อดภายในคลาส C_i

$call(C_j, m)$ คือ จำนวนการเรียกใช้เมท็อด m จากคลาส C_j

$call(C_i, m)$ คือ จำนวนการเรียกใช้เมท็อด m ภายในคลาส C_i

$call(A_{C_i}, m)$ คือ จำนวนการเรียกใช้ตัวแปรของเมท็อด m ภายในคลาส C_i

ตัวอย่างการคำนวณมาตรวัด

การคำนวณมาตรวัด IMC และ ICBC ใช้รหัสต้นทาง จากภาพที่ 2.2 ตัวอย่างของรหัสต้นทาง ในบทที่2 หน้าที่ 14

การคำนวณมาตรวัด *IMC*

จาก class Y ในเมท็อด My5() จะได้

$call(X, My5) = 2$ $call(Y, My5) = 0$ $call(AX, My5) = 1$

$$\text{IMC} = \frac{call(X, My5)}{call(Y, My5) + call(X, My5) + call(A_X, My5)}$$ จากสูตรจะได้ $I = 0.67$

การคำนวณมาตรวัด *ICBC*

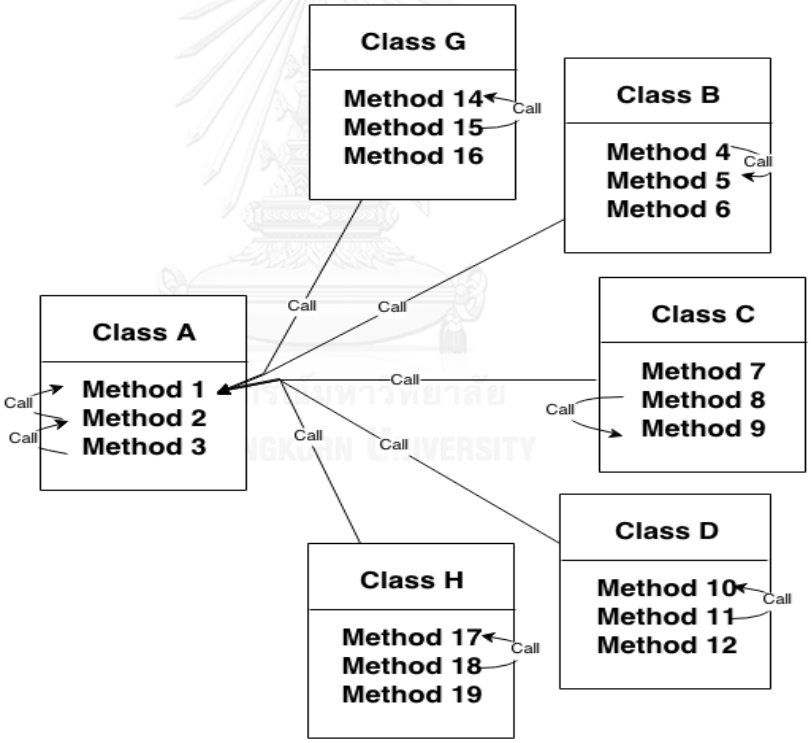
$Interaction(X, Y) = 2$

$Interaction(Y, X) = 1$

$ICBC = \frac{Interaction(X, Y) + Interaction(Y, X)}{2}$ จากสูตรจะได้ $ICBC = 1.5$

3.2 ขั้นตอนการปรับปรุงคุณภาพของรหัสต้นทาง

งานวิจัยนี้มีแนวคิดจากการมองปัญหาของ การปรับปรุงโครงสร้างภายในของซอฟต์แวร์ เพื่อต้องการให้ง่ายต่อเปลี่ยนแปลงแก้ไขและง่ายต่อการบำรุงรักษา ง่ายต่อความเข้าใจ ง่ายในการนำไปพัฒนาต่อ โดยการพิจารณาต้องให้อยู่ภายใต้กฎการเกาะกลุ่มกันสูง การเข้าคู่กันต่ำ เพราะถ้าแต่ละคลาสของซอฟต์แวร์มีการเข้าคู่กันมาก เช่น มีการใช้ตัวแปรต่างๆ ในซอฟต์แวร์ร่วมกันมาก เมื่อมีการแก้ไขคลาสหนึ่งก็จะส่งผลกับคลาสอื่นๆ ที่เกี่ยวข้องด้วย หรือในคลาสหนึ่งๆ มีพฤติกรรมหลายพฤติกรรมอยู่ในคลาสเดียวกัน ก็จะทำให้คลาสมีการเกาะกลุ่มกันน้อยลง โดยมาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันจะสามารถบอกปริมาณการเกาะกลุ่มกันและการเข้าคู่กันของซอฟต์แวร์ได้



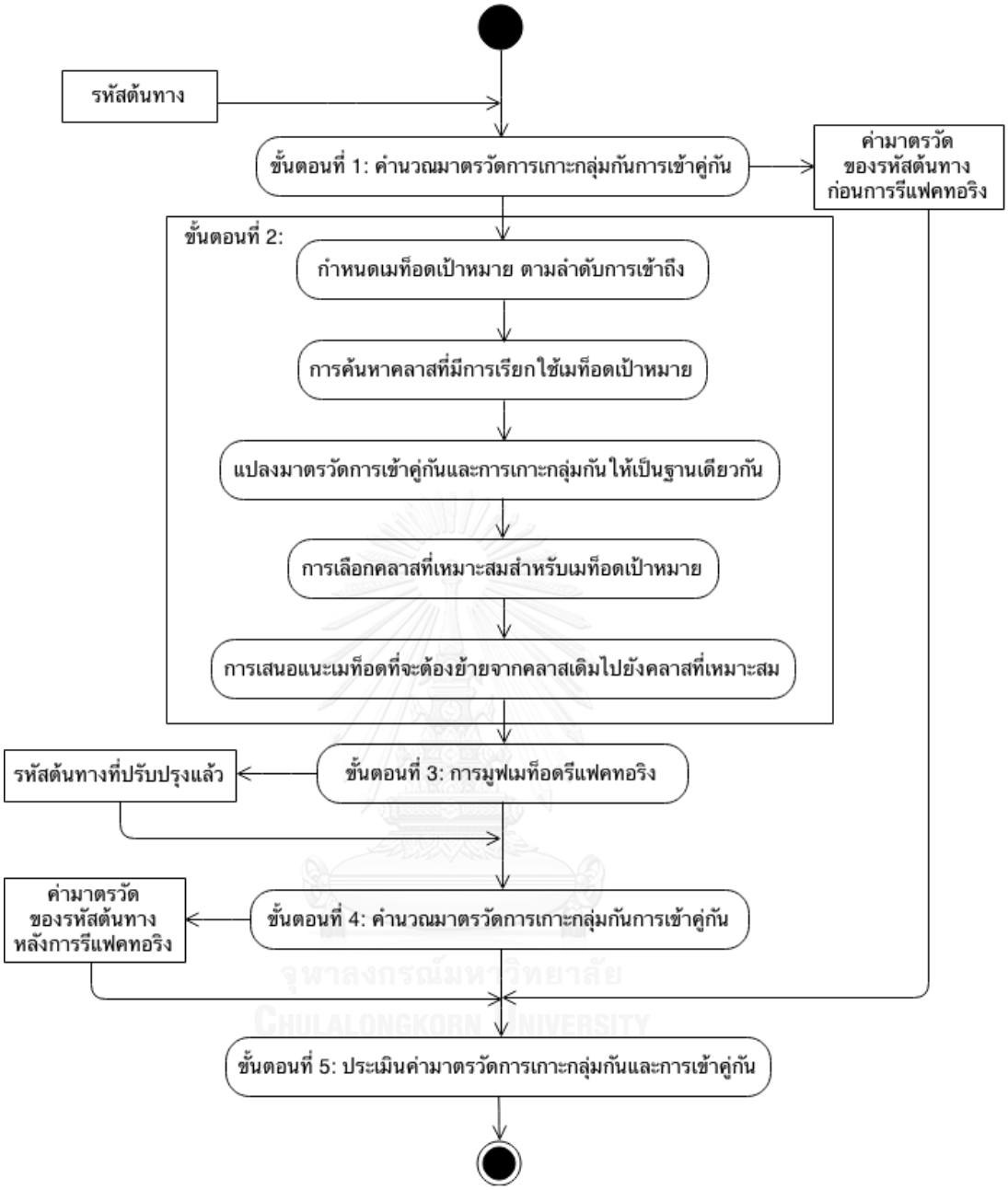
ภาพที่ 3.1 แผนภาพแสดงเมท็อดเป้าหมายที่มีการเรียกใช้จากคลาสภายนอก

จากภาพที่ 3.1. มีคลาสทั้งหมด 6 คลาส คือคลาสA,คลาสB, คลาสG, คลาสC, คลาสD และคลาสH ซึ่งทุกคลาสมีการเรียกใช้ Method1 ของคลาส A โดยในแต่ละคลาสอาจมีจำนวนการเรียกใช้ Method1 มากน้อยต่างกัน และภายในคลาส A อาจมีการเรียกใช้ Method1 น้อยกว่าที่คลาสภายนอก ซึ่งจะทำให้การเข้าคู่กันมีค่าที่สูงขึ้น ดังนั้นจึงจำเป็นต้องพิจารณาว่าเมท็อดเป้าหมาย ซึ่งก็

คือ Method1 จะอยู่ที่คลาสเดิมหรือจำเป็นต้องย้ายไปยังคลาสที่เหมาะสม เพื่อที่จะทำให้การเข้าคู่กันมีค่าน้อยลง ซึ่งจะมี 5 ทางเลือกในการพิจารณา จากคลาสทั้งหมด 5 คลาส ที่มีการเรียกใช้ Method1 ของคลาส A โดยต้องพิจารณาว่าจะย้ายเมท็อด Method1ไปที่คลาสใดที่เหมาะสมกว่าหรือ อยู่ที่เดิมเหมาะสมแล้ว

งานวิจัยนี้มีการประยุกต์ใช้ทฤษฎีการตัดสินใจมาใช้ในการมูฟเมท็อดรีแฟคทอริง เพื่อระบุโอกาสการย้ายเมท็อดเป้าหมายไปยังคลาสที่เหมาะสม โดยใช้เกณฑ์การตัดสินใจแบบลาปลาส ของทฤษฎีการตัดสินใจ เป็นเกณฑ์ในการระบุโอกาสว่าจะย้ายเมท็อดเป้าหมายไปที่คลาสใดที่เหมาะสม ที่มีการเรียกใช้เมท็อดนั้น ซึ่งลาปลาสจะตัดสินใจจากค่าที่ได้จากมาตรวัดที่ใช้วัดการเข้าคู่กัน ในแต่ละทางเลือก โดยหนึ่งคลาสจะเป็นหนึ่งทางเลือก หลังจากนั้นก็เฉลี่ยค่าที่ได้จากมาตรวัดในแต่ละทางเลือก แล้วเลือกทางเลือกที่ได้ค่าเฉลี่ยจากทุกมาตรวัดที่มากที่สุด เพื่อเสนอแนะเป็นโอกาสการย้ายเมท็อดเป้าหมายไปยังคลาสที่เหมาะสม

งานวิจัยนี้สามารถแบ่งการดำเนินงานออกเป็น 5 ขั้นตอน ดังแสดงด้วยแผนภาพกิจกรรม (Activity diagram) ของแผนภาพขั้นตอนการปรับปรุงคุณภาพของรหัสต้นทางด้วยการมูฟเมท็อดรีแฟคทอริง ได้แก่ ขั้นตอนแรกคือ การคำนวณมาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันก่อนรีแฟคทอริง ขั้นตอนที่ 2 คือ การระบุโอกาสการย้ายเมท็อดด้วยวิธีที่นำเสนอ โดยแบ่งออกเป็น 4 ขั้นตอนย่อย ได้แก่ การกำหนดเมท็อดเป้าหมายตามลำดับการเข้าถึง การค้นหาคลาสที่มีการเรียกใช้เมท็อดเป้าหมาย การเลือกคลาสที่เหมาะสมสำหรับเมท็อดเป้าหมาย การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม ขั้นตอนที่ 3 คือ การมูฟเมท็อดรีแฟคทอริง ตามการระบุโอกาสการย้ายเมท็อดที่ได้จากขั้นตอนที่สอง ขั้นตอนที่ 4 คือ การคำนวณมาตรวัดการเกาะกลุ่มกันและการเข้าคู่กัน หลังรีแฟคทอริง และขั้นตอนที่ 5 คือ การประเมินความสามารถของการปรับปรุงคุณภาพของรหัสต้นทางโดยประเมิน ค่าที่ได้จากมาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันก่อนและหลังทำรีแฟคทอริง ดังแสดงในแผนภาพที่ 3.2



ภาพที่ 3.2 แผนภาพขั้นตอนการปรับปรุงคุณภาพของรหัสต้นทางด้วยการมูฟเมท็อดรีแฟคทอริง

ขั้นตอนนี้จะแสดงกระบวนการปรับปรุงคุณภาพของรหัสต้นทางด้วยการมูฟเมท็อดรีแฟคทอริง ซึ่งจะใช้ทฤษฎีการตัดสินใจ ในการระบุโอกาสของเมท็อดในการย้ายไปยังคลาสที่เหมาะสม โดยสามารถแสดงรายละเอียดตามภาพที่ 3.2 ได้ดังต่อไปนี้

ขั้นตอนที่ 1 การคำนวณมาตรวัดการเกาะกลุ่มกันการเข้าคู่กันก่อนรีแฟคทอริง

ในขั้นตอนนี้ นำรหัสต้นทางมาคำนวณมาตรวัดการเกาะกลุ่มกัน และการเข้าคู่กัน ด้วยมาตรวัด LCOM, TCC, CBO และ RFC ก่อนที่จะรีแฟคทอริง

ขั้นตอนที่ 2 การระบุโอกาสการย้ายเมท็อดด้วยทฤษฎีการตัดสินใจ

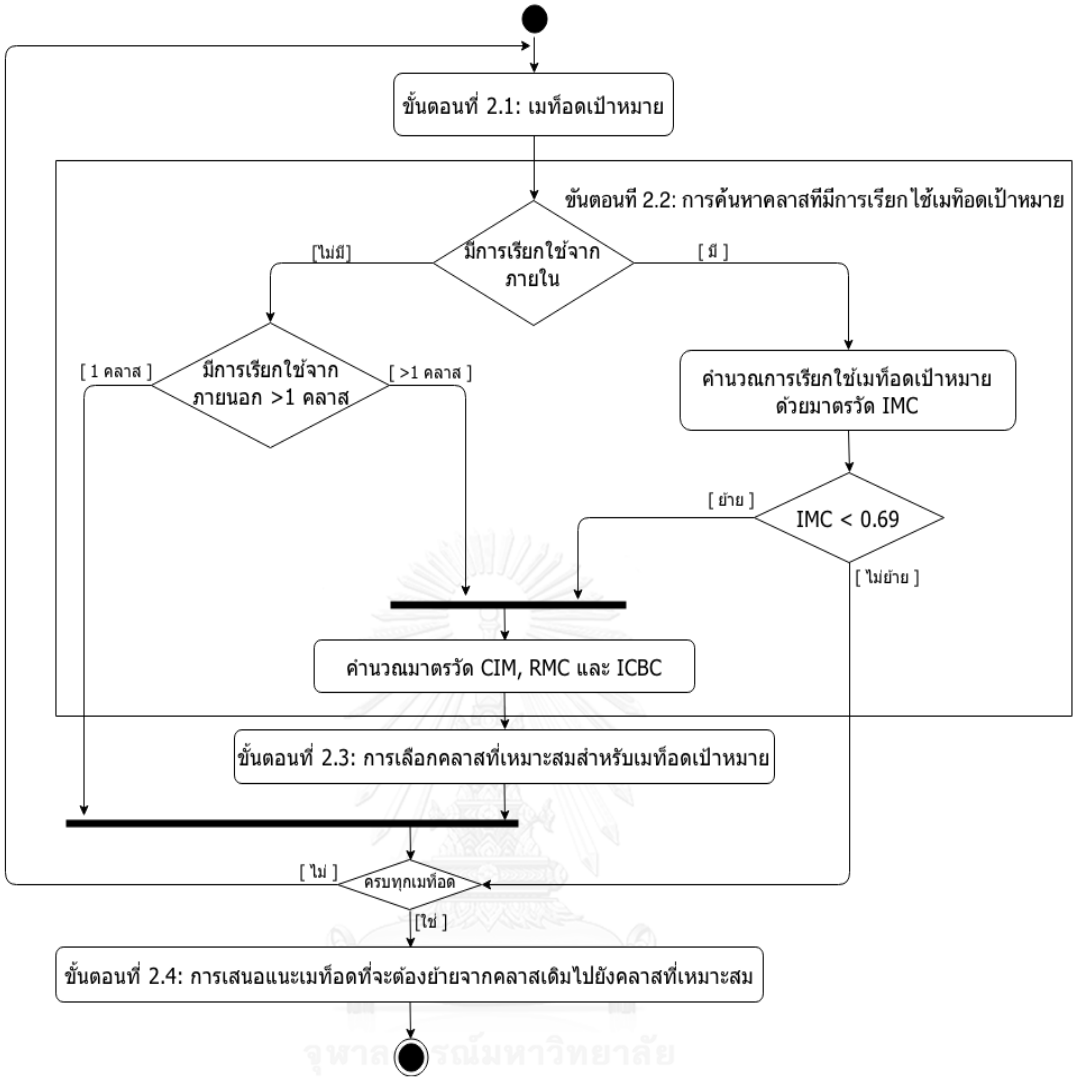
ในขั้นตอนการระบุโอกาสการย้ายเมท็อดด้วยทฤษฎีการตัดสินใจ สามารถแบ่งออกเป็น 4 ขั้นตอนย่อย ได้แก่ การกำหนดเมท็อดเป้าหมายตามลำดับการเข้าถึง การค้นหาคลาสที่มีการเรียกใช้เมท็อดเป้าหมาย การเลือกคลาสที่เหมาะสมสำหรับเมท็อดเป้าหมาย การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม โดยสามารถแสดงด้วยแผนภาพกิจกรรม ตามภาพที่ 3.3 สามารถอธิบายขั้นตอนการระบุโอกาสการย้ายเมท็อดด้วยทฤษฎีการตัดสินใจ ได้ดังนี้

ขั้นตอนที่ 2.1 การกำหนดเมท็อดเป้าหมาย

การกำหนดเมท็อดเป้าหมายตามลำดับการเข้าถึง ในแต่ละคลาสของระบบซอฟต์แวร์

ขั้นตอนที่ 2.2 การค้นหาคลาสที่มีการเรียกใช้เมท็อดเป้าหมาย

การค้นหาคลาสที่มีการเรียกใช้เมท็อดเป้าหมาย คือการค้นหาว่าคลาสใดมีการเรียกใช้ เมท็อดเป้าหมายบ้าง โดยคลาสนั้นจะเรียกใช้เมท็อดหนึ่งครั้งหรือมากกว่า ก็จะเป็นหนึ่งทางเลือก แสดงว่าหนึ่งคลาสจะเท่ากับหนึ่งทางเลือก โดยแบ่งเป็นการค้นหาคลาสที่มีการเรียกใช้เมท็อดเป้าหมาย และการคำนวณมาตรวัดในแต่ละคลาส



ภาพที่ 3.3 การระบุโอกาสการย้ายเมท็อดด้วยทฤษฎีการตัดสินใจ

การค้นหาคลาสที่มีการเรียกใช้เมท็อดเป้าหมายแบ่งเป็น 3 กรณี

1) **การค้นหาคลาส** จากเมท็อดเป้าหมายที่มีการเรียกใช้จากภายนอกคลาสเพียง 1 คลาส และไม่มีการเรียกใช้ภายในคลาส

   การคำนวณมาตรวัด ในกรณีนี้ ไม่จำเป็นต้องคำนวณมาตรวัด เพราะเมท็อดเป้าหมาย มีการเรียกใช้จากภายนอกคลาสเพียง 1 คลาส และไม่มีการเรียกใช้ภายในคลาส

2) **การค้นหาคลาส** จากเมท็อดเป้าหมายที่มีการเรียกใช้จากภายนอกคลาสมากกว่า 1 คลาส และไม่มีการเรียกใช้ภายในคลาส

การคำนวณมาตรวัด ในกรณีนี้ ต้องคำนวณมาตรวัดการเข้าคู่กัน เพราะเมท็อดเป้าหมายที่มีการเรียกใช้จากภายนอกคลาสมากกว่า 1 คลาสโดยคำนวณด้วยมาตรวัด CIM, RMC และ ICBC

3) **การค้นหาคลาส** จากเมท็อดเป้าหมายที่มีการเรียกใช้จากภายนอกคลาส และมีการเรียกใช้ภายในคลาส

การคำนวณมาตรวัด ในกรณีนี้ ต้องคำนวณมาตรวัดการเข้าคู่กัน เพราะมีการเรียกใช้เมท็อดเป้าหมายภายในคลาสและมีการเรียกใช้จากภายนอกคลาส โดยต้องคำนวณด้วยมาตรวัด IMC ก่อนเพื่อตัดสินใจว่าจะย้ายเมท็อดเป้าหมายจากคลาสเดิม ไปยังคลาสอื่นหรือไม่

โดยมาตรวัด IMC จะนับจำนวนการเรียกใช้เมท็อดเป้าหมายภายในคลาสเมท็อดเป้าหมายมีการเรียกใช้ตัวแปรภายในคลาส และการเรียกใช้เมท็อดเป้าหมายจากภายนอกคลาส โดยแบ่งการพิจารณาเป็น

ก. ถ้ามาตรวัด IMC มีค่ามากกว่าหรือเท่ากับ 0.70 ก็พิจารณาต่อว่าจะมูฟเมท็อดไปที่คลาสใดด้วยมาตรวัด CIM, RMC และ ICBC

ข. ถ้ามาตรวัด IMC มีค่าน้อยกว่า 0.70 ก็จะไม่พิจารณาคลาสนี้ เพราะการเรียกใช้เมท็อดเป้าหมายของคลาสนี้ มีค่าใกล้เคียงกันกับการเรียกใช้เมท็อดเป้าหมายภายในคลาส

การแปลงข้อมูล หลังจากวัดค่าด้วยมาตรวัดการเกาะกลุ่มแล้ว จะต้องทำการแปลงค่าของตัวเลขที่ได้ ให้อยู่ในช่วง 0...1 เหมือนกัน โดยมาตรวัดที่จะต้องแปลงคือ RMC, ICBC เพราะไม่ได้อยู่ในช่วง 0...1

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

<u>ขั้นตอนที่ 2.3</u> การเลือกคลาสที่เหมาะสมสำหรับเมท็อดเป้าหมาย

การเลือกคลาสที่เหมาะสมสำหรับเมท็อดเป้าหมาย ด้วยวิธีลาปลาส ของทฤษฎีการตัดสินใจ โดยการนำค่าของมาตรวัดการเกาะกลุ่มกันที่ได้ในแต่ละคลาส มาคำนวณดังนี้

1) การเฉลี่ยค่าที่ได้จากมาตรวัดการเกาะกลุ่มกัน ด้วยสูตรในการหาคือ

$$U(a_i) = \frac{1}{3}(CIM_{i1} + RMC_{i2} + ICBC_{i3})$$

ซึ่ง CIM คือ มาตรวัดการเข้าคู่กันระหว่างเมท็อดของคลาส 2 คลาส

RMC คือ การวัดความสัมพันธ์ของเมท็อดที่มีการเรียกใช้ตัวแปรและเมท็อดจากคลาสอื่น และการเรียกใช้ตัวแปรและเมท็อดภายในคลาส

ICBC คือ มาตรวัดการเรียกใช้กันระหว่างคลาส 2 คลาส

a_i คือ คลาสที่ i โดย i มีค่าตั้งแต่ 1,.., n ซึ่งได้จากขั้นตอนที่ 2.2

$U(a_i)$ หมายถึง ค่าเฉลี่ยในของทางเลือกในลำดับที่ i

2) เลือกค่าที่มากที่สุดจากคลาสที่มีทั้งหมด โดยสูตรในการเลือกคือ

$$\max\{U(a_i)\}$$

โดยจะได้คลาส a_i ที่มีค่าเฉลี่ยจากมาตรวัด CIM, RMC และ ICBC ที่มากที่สุด เป็นคลาสที่เหมาะสมที่เมท็อดเป้าหมายจะย้ายไป

<u>ขั้นตอนที่ 2.4</u> การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม

เมื่อค้นหาเมท็อดที่จะต้องย้ายจากขั้นตอนที่ 2.1 ถึงขั้นตอนที่ 2.3 ทุกเมท็อด ในขั้นตอนนี้จะแสดงเมท็อดทั้งหมดในระบบ ที่จะเสนอแนะให้ย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม

<u>ขั้นตอนที่ 3</u> การมูฟเมท็อดรีแฟคทอริง

ในขั้นตอนนี้ เป็นการนำทางเลือกที่ได้จากขั้นตอนที่ 2 ไปทำรีแฟคทอริงกับรหัสต้นทาง เพื่อทำการปรับปรุงรหัสต้นทางด้วยการมูฟเมท็อดรีแฟคทอริง

โดยแบ่งออกเป็น 3 ขั้นตอน

1. สร้างเมท็อดใหม่ภายในคลาสที่ต้องการ ซึ่งอาจจะมีการตั้งชื่อใหม่ตามการใช้งาน
2. หลังจากสร้างเมท็อดใหม่แล้ว ทำการคัดลอกรหัสต้นทางจากเมท็อดเดิมไปยังเมท็อดใหม่ และแก้ไขการอ้างอิงถึงเมท็อด เพื่อให้สามารถทำงานได้
3. ทำการแก้ไขรหัสต้นทางภายในเมท็อดเดิมให้ไปเรียกใช้เมท็อดใหม่แทน

<u>ขั้นตอนที่ 4</u> การคำนวณมาตรวัดการเกาะกลุ่มกันการเข้าคู่กันหลังรีแฟคทอริง

ในขั้นตอนนี้ นำรหัสต้นทางที่ปรับปรุงแล้วมาคำนวณค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันอีกครั้งหลังทำรีแฟคทอริง

<u>ขั้นตอนที่ 5</u> การประเมินค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กัน

ในขั้นตอนสุดท้าย นำค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กัน ก่อนและหลังทำรีแฟคทอริง มาคำนวณเพื่อเปรียบเทียบค่าของมาตรวัด และประเมินความสามารถในการทำรีแฟคทอริง

หลังจากใช้ทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อด เพื่อนำค่าที่ได้ไปประเมินความสามารถในการปรับปรุงรหัสต้นทาง

การประเมินความสามารถในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง

โดยจะพิจารณาจากการวัด การเรียกใช้ตัวแปรภายในคลาสร่วมกัน และการเรียกใช้เมท็อดทั้งภายในคลาสและภายนอกคลาส และการเรียกใช้จากคลาสอื่น โดยใช้มาตรวัด LCOM, TCC, RFC และ CBO

หลักเกณฑ์การพิจารณาค่ามาตรวัด

1) มาตรวัด LCOM ต้องให้ค่าต่ำลง เพราะการที่ค่าของ LCOM มีค่าต่ำลงแสดงว่ามีการเกาะกลุ่มกันสูง และคลาสนั้นมีการออกแบบที่ดี
2) มาตรวัด TCC ต้องให้ค่าที่สูงขึ้น เพราะมาตรวัด TCC เป็นมาตรวัดที่ใช้ในการนับจำนวนที่มีการเรียกใช้ เมท็อดกันโดยตรงภายในคลาส ถ้ามีค่าสูงขึ้นแสดงว่ามีการเกาะกลุ่มกันสูงและเป็นการออกแบบที่ดีขึ้น
3) มาตรวัด CBO ต้องให้ค่าต่ำลง เพราะเป็นการลดการเข้าคู่กันระหว่างคลาส
4) มาตรวัด RFC ต้องให้ค่าต่ำลง เพราะเป็นการลดการเรียกใช้ระหว่างเมท็อด ทำให้ลดการเข้าคู่กันและความซับซ้อนน้อยลง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.3 ตัวอย่างการปรับปรุงคุณภาพของรหัสต้นทาง

public class A
{
int a, c, d, e;
private void Method1 () { uses a, c, e }
private void Method 2()
{
... uses d, e ...
return Method 1();
}
	public class B
	{
	int a, b, c;
	private void Method 4 ()
	{
	... uses a, c;
	Method *5(); Method6();*
	}
	public void Method 5()
	{

public int Method 3() { … }
}
	A callA = new A ();
	if (a>b) a= ***callA***. **Method 3();**
	else b=***callA***. **Method 3();**
	}
	public int Method 6(){….}
	}

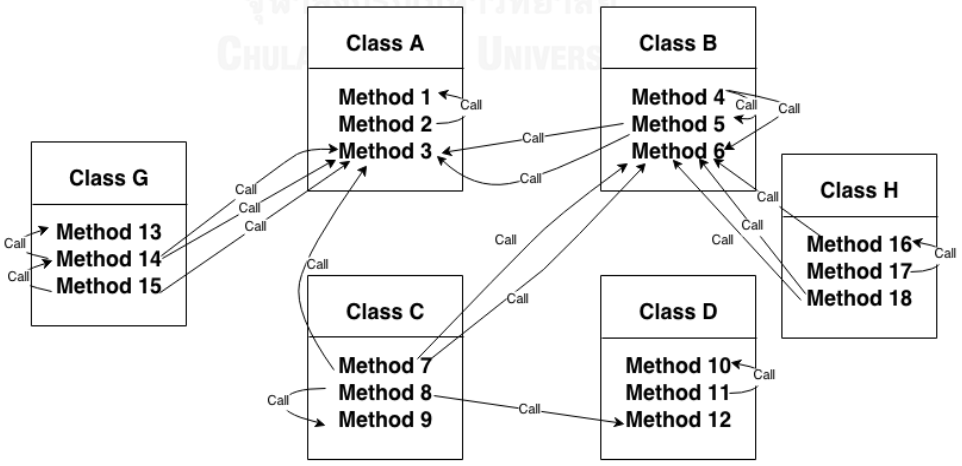
public class C
{
int a, c, d, e;
private void Method 7 ()
{
… uses a, d;
A callA = new A ();
e = ***callA***. **Method 3();**
B callB = new B ();
if (a>c) d = ***callB***. **Method 6();**
else a = ***callB***. **Method 6()**;
}
public void Method8()
{ … uses d, e …
Method 9();
D callD = new D ();
callD. **Method 12();**
}
public void Method 9(){… uses a …}
}
	public class D
	{
	int a, c, d, e;
	private void Method10 () { uses a, e }
	private void Method 11()
	{
	… uses d, e …
	Method 10();
	}
	public void Method 12(){… uses a …}
	}

public class G { int a, c, d, e; private void Method 13 () { *uses a,*	public class H { int a, c, d, e; private void Method 16()

```
    d;}
    public void Method14()
    {
        Method 13();
        A callA = new A ();
        if (a>c)  d= callA. Method 3();
        else a=callA. Method 3();
    }
    public void Method 15()
    {
        Method 14();
        A callA = new A ();
        d = callA. Method 3();
    }
}
```

```
    {
        ... uses a, d;
        B callB = new B ();
        a =  callB. Method 6();
    }
    public void Method17()
    {    ... uses d, e ...
        Method 16();
    }
    public void Method 18()
    {
        B callB = new B ();
        if (a>c)  d= callB. Method 6();
        else a=callB. Method 6();
    }
}
```

ภาพที่ 3.4 ตัวอย่างของรหัสต้นทาง



ภาพที่ 3.5 ตัวอย่างแผนภาพคลาสของรหัสต้นทาง

จากตัวอย่างในภาพที่ 3.5 นำมาสร้างเป็นตารางการเรียกใช้เมท็อดระหว่างคลาสและภายในคลาส ซึ่งจะแสดงการเรียกใช้จากคลาสปลายทางไปเรียกใช้เมท็อดจากคลาสต้นทาง(Source Class) ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 การเรียกใช้กันระหว่างคลาสต้นทางและคลาสปลายทาง

ลำดับ	เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
1	Method 3	Class A	Class B	0	2
2	Method 3	Class A	Class C	0	1
3	Method 3	Class A	Class G	0	3
4	Method 6	Class B	Class C	1	2
5	Method 6	Class B	Class H	1	3
6	Method 12	Class D	Class C	0	1

การค้นหาคลาสเพื่อระบุโอกาสการย้ายเมท็อดเป้าหมาย ไปยังคลาสที่เหมาะสม ตามการเรียกใช้เมท็อดเป้าหมายจากภายในคลาสและภายนอกคลาส

1) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหนึ่งคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้ มีหนึ่งเมท็อด คือ Method12 และสามารถค้นหาคลาสที่เรียกใช้ Method12 ได้หนึ่งคลาส ดังนั้น ทางเลือกของ Method12 คือ ย้ายจากคลาสD ไปยังคลาส C

2) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหลายคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้ มีหนึ่งเมท็อด คือ Method3 และสามารถค้นหาคลาสที่เรียกใช้ Method3ได้ 3 คลาสดังแสดงรายละเอียดในตารางที่ 3.2

ตารางที่ 3.2 การเรียกใช้ Method 3 จากคลาสภายนอกหลายคลาส

ลำดับ	เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
1	Method 3	Class A	Class B	0	2
2	Method 3	Class A	Class C	0	1
3	Method 3	Class A	Class G	0	3

อ้างอิงลำดับตามตารางที่ 3.1

การคำนวณค่ามาตรวัด ในกรณีนี้คำนวณด้วยมาตรวัด CIM, RMC และ ICBC โดยไม่ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ 3.3 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับ Method 3

ทางเลือกลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
1	0.330	0.020	0.010	0.120
2	0.170	0.010	0.005	0.062
3	0.500	0.030	0.015	**0.182**

การเลือกคลาสที่เหมาะสม โดยการเลือกด้วยวิธีลาปลาส จะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด คือทางเลือกที่ 3 ดังนั้น เลือกย้าย Method3 จากคลาสA ไปยังคลาสG

3) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอก และมีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีมีหนึ่งเมท็อด คือ Method 6 และสามารถค้นหาคลาสที่เรียกใช้ Method 6ได้ 2 คลาสดังแสดงรายละเอียดในตารางที่ 3.4

ตารางที่ 3.4 การเรียกใช้ Method 6 จากภายนอกคลาสและภายในคลาส

ลำดับ	เมท็อด	คลาสต้นทาง	คลาสปลายทาง	เรียกจากต้นทาง	เรียกจากปลายทาง
4	Method 6	Class B	Class C	1	2
5	Method 6	Class B	Class H	1	3

อ้างอิงลำดับตามตารางที่ 3.1

การคำนวณค่ามาตรวัด ในกรณีนี้ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะมีการเรียกใช้เมท็อดเป้าหมายภายในคลาส จากนั้นคำนวณด้วยมาตรวัด CIM, RMC และ ICBC

โดยแบ่งเป็น 2 ขั้นตอน

- **ขั้นตอนที่ 1** คำนวณด้วยมาตรวัด IMC เพื่อจะคำนวณการเรียกใช้ภายในคลาส และการเรียกใช้จากภายนอกคลาส ของเมท็อดเป้าหมาย

ตารางที่ 3.5 การวัดค่า IMC ของการเรียกใช้ Method 6

ลำดับ	คลาสต้นทาง	คลาสปลายทาง	IMC
4	Class B	Class C	0.67
5	Class B	Class H	**0.75**

อ้างอิงลำดับตามตารางที่ 3.1

- **ขั้นตอนที่ 2** เลือกทางเลือกที่มีค่า IMC >= 0.70 มาคำนวณด้วยมาตรวัด CIM, RMC และ ICBC ในแต่ละทางเลือก เพื่อพิจารณาจะย้ายเมท็อดไปที่คลาสใด

ตารางที่ 3.6 การตัดสินใจการเรียกใช้ Method 6 จากคลาสภายนอก

ทางเลือก ลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
5	0.330	0.030	0.015	0.125

อ้างอิงลำดับตามตารางที่ 3.1

การเลือกคลาสที่เหมาะสม โดยจะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด แต่ในตัวอย่างมีทางเลือกเดียวที่มีค่า IMC >= 0.70 ดังนั้น เลือกย้าย Method 6 จากคลาสB ไปยังคลาสH

การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม จากตัวอย่างสามารถเสนอแนะเมท็อดที่จะต้องย้าย ดังที่แสดงในตารางที่3.7 คือย้าย Method 3 จากคลาส A ไปยังคลาส G ย้าย Method 6 จากคลาส B ไปยังคลาส H ย้าย Method 12 จากคลาส D ไปยังคลาส C

ตารางที่ 3.7 เมท็อดที่แนะนำในการย้ายไปยังคลาสที่เหมาะสม

ลำดับ	เมท็อด	คลาสต้นทาง	คลาสปลายทาง
3	Method 3	Class A	Class G
5	Method 6	Class B	Class H
6	Method 12	Class D	Class C

อ้างอิงลำดับตามตารางที่ 3.1

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

การประเมินความสามารถในการทำรีแฟคทอริงหลังจากใช้ทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อด

การประเมินค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่ ประเมินได้จากการนำค่าของมาตรวัดการเกาะกลุ่มกันการเข้าคู่กัน ก่อนรีแฟคทอริง และหลังจากการทำรีแฟคทอริงโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อด ซึ่งการประเมินจะพิจารณาค่าที่ได้จากมาตรวัดจะมีค่าสูงขึ้นหรือต่ำลง ตามหลักเกณฑ์ของมาตรวัด

ตารางที่ 3.8 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริง

	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
Class A	1	0.33	3	0	0 ↓	0.33	2 ↓	0
Class B	1	0.33	5	1	0 ↓	0.33	5	2 ↑
Class C	0	0.67	7	2	2 ↑	0.67	6 ↓	2
Class D	0	0.67	3	0	0	0.33 ↓	2 ↓	0
Class G	0	1	6	1	0	1	4 ↓	0 ↓
Class H	0	1	6	1	0	1	4 ↓	0 ↓
Mean	0.33	0.67	5	0.83	0.33	0.61	3.83	0.67
S.D.	0.39	0.27	1.53	0.49	0.68	0.30	1.46	0.77
Max	1	1	7	2	2	1	6	2
Min	0	0.33	3	0	0	0.33	2	0

Mean คือ ค่าเฉลี่ยของแต่ละมาตรวัดทั้งระบบ

S.D. คือ ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation)

จากตัวอย่าง ค่าของมาตรวัดที่ประเมินได้ เมื่อนำระบบมาคำนวณด้วยมาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันก่อนรีแฟคทอริงและหลังรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสมาเปรียบเทียบกัน ดังที่แสดงในตารางที่ 3.8 สามารถปรับปรุงรหัสต้นทางให้มีคุณภาพที่ดีขึ้นได้โดยพิจาณาจากมาตรวัด เมื่อคิดเป็นร้อยละของมาตรวัดจากทั้งระบบโดยนำค่า Mean ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดยมาตรวัด LCOM ไม่เปลี่ยนแปลง TCC ลดลงร้อยละ 8.96 RFC ลดลงร้อยละ 23.40 CBO ลดลงร้อยละ 19.28

บทที่ 4
การออกแบบเครื่องมือ IMRD สำหรับการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง

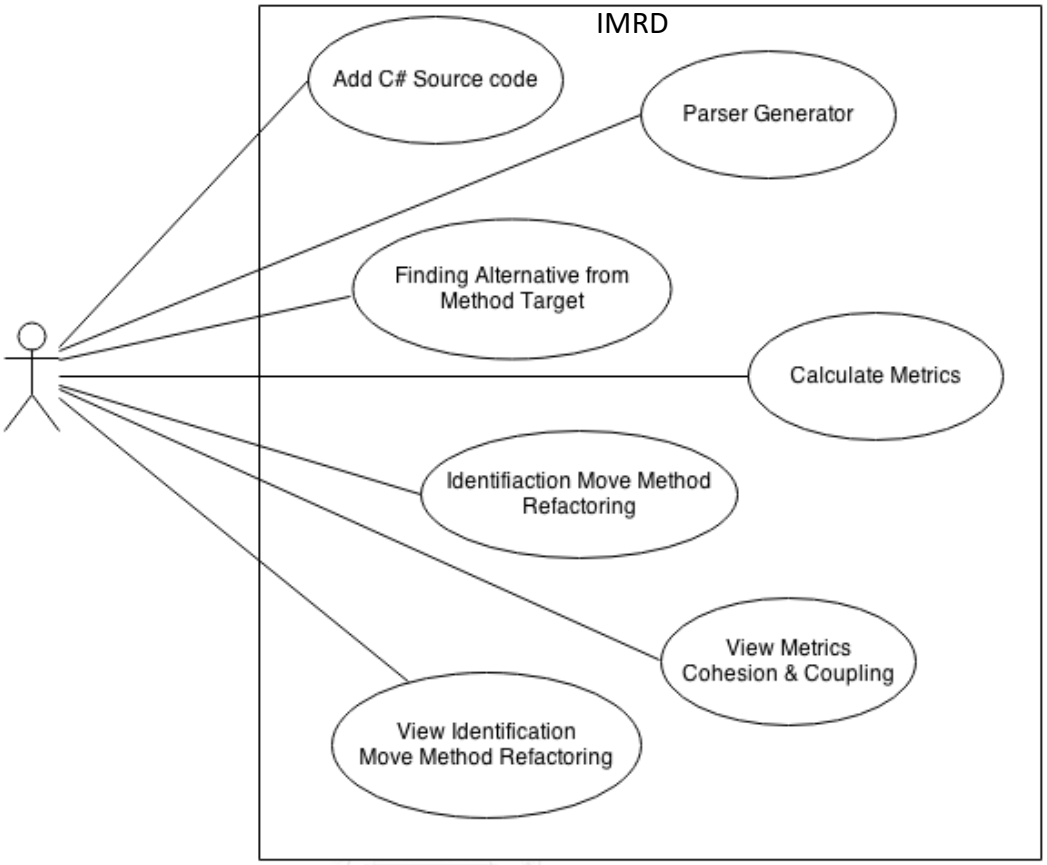
ในบทนี้จะกล่าวถึงการออกแบบและพัฒนาเครื่องมือที่ใช้ในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง ซึ่งจะเรียกเครื่องมือที่พัฒนาขึ้นว่า IMRD Tool ซึ่งย่อมาจาก **I**dentification of opportunities for **M**ove method **R**efactoring using **D**ecision theory tool. โดยเครื่องมือจะแบ่งออกเป็น การตรวจจับเมท็อดที่จะมูฟ การคำนวณมาตรวัด และการระบุโอกาสการย้ายเมท็อดด้วยทฤษฎีการตัดสินใจ ซึ่งจะอธิบายโดยใช้ภาษายูเอ็มแอล (Unified Modeling Language – UML) ด้วยแผนภาพยูสเคส(Use Case Diagram) ซึ่งเป็นการนำเสนอภาพรวมของระบบ แผนภาพคลาส (Class Diagram) แสดงถึงคลาสและความสัมพันธ์ระหว่างคลาสที่ประกอบด้วยพฤติกรรมและสถานะของคลาส แผนภาพลำดับการทำงาน (Sequence Diagram) แสดงถึงการติดต่อกันระหว่างอ็อบเจกต์หรือฟังก์ชันการทำงานในแผนภาพยูสเคส แผนภาพองค์ประกอบ (Component Diagram) แสดงถึงองค์ประกอบของเครื่องมือ IMRD ดังรายละเอียดต่อไปนี้

4.1 แผนภาพยูสเคสแสดงการปฏิสัมพันธ์ระหว่างผู้ใช้งานกับระบบ

จากการวิเคราะห์และการออกแบบเครื่องมือการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงสามารถพิจารณารูปแบบการใช้ระบบงานที่เกิดขึ้น โดยแสดงให้เห็นว่าผู้ใช้จะนำระบบไปใช้อะไรบ้าง และจะอธิบายการทำงานเป็นลำดับของเหตุการณ์ที่เกิดขึ้น จากการวิเคราะห์การเกิดปฏิสัมพันธ์ระหว่างผู้ใช้งานกับระบบ สามารถเขียนเป็นแผนภาพได้ดังภาพที่ 4.1 ซึ่งสามารถอธิบายความสัมพันธ์ของยูสเคสที่เกิดขึ้นภายในระบบงานได้ดังนี้

ผู้ใช้งานสามารถติดต่อและใช้เครื่องมือนี้ได้โดย

1) ผู้ใช้งานสามารถกำหนดชุดข้อมูลของรหัสต้นทางที่ต้องการนำมาใช้เพื่อหาโอกาสการมูฟเมท็อดรีแฟคทอริงได้
2) ผู้ใช้งานสามารถดูค่ามาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันของรหัสต้นทาง ก่อนการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง
3) ผู้ใช้งานสามารถดูการแนะนำการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงของรหัสต้นทางของเมท็อดที่มีการเรียกใช้จากคลาสอื่นได้
4) ผู้ใช้งานสามารถยกเลิกชุดข้อมูลของรหัสต้นทางเดิมเพื่อกำหนดชุดข้อมูลของรหัสต้นทางใหม่ได้



ภาพที่ 4.1 แผนภาพยูสเคสของระบบการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง

4.1.1 ยูสเคส Add C# Source code

จากตารางที่ 4.1 เป็นยูสเคสที่แสดงเหตุการณ์ที่เกิดขึ้นเมื่อผู้ใช้งานต้องการนำโปรแกรมต้นฉบับภาษาซีชาร์ป (C#) เข้าสู่เครื่องมือการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง (IMRD)

ตารางที่ 4.1 แสดงขั้นตอนการทำงานของยูสเคส Add C# Source code

Use Case Name	Add C# Source code
Entry condition	1. ผู้ใช้งานเลือกรหัสต้นทางเข้าสู่ระบบ
Flow of events	2. ผู้ใช้งานเลือกรหัสต้นทางภาษาซีชาร์ป จากแหล่งที่เก็บ 3. เครื่องมือ IMRD นำรหัสต้นทางที่ผู้ใช้งานกำหนดเข้าสู่ระบบงาน
Exit condition	4. ระบบทำการเพิ่มรหัสต้นทางตามที่ผู้ใช้งานกำหนด จนกว่าผู้ใช้งานจะทำการเลือกรายการอื่นๆ

4.1.2 ยูสเคส Parser Generator

จากตารางที่ 4.2 เป็นยูสเคสที่แสดงเหตุการณ์ที่เกิดขึ้นเมื่อเครื่องมือ IMRD เริ่มทำการสร้างโหนดของต้นไม้เอเอสที

ตารางที่ 4.2 แสดงขั้นตอนการทำงานของยูสเคส Parser Generator

Use Case Name	Parser Generator
Entry condition	1. รหัสต้นทางของโปรแกรมภาษาซีชาร์ปที่ต้องการวัดเพื่อระบุโอกาสการมูฟเมท็อดรีแฟคทอริง
Flow of events	2. ระบบนำรหัสต้นทางจากยูสเคส Add C# Source code มาทำการสร้างโหนดของต้นไม้เอเอสที ที่เก็บข้อมูลหรือคุณสมบัติ ต่างๆ เช่น ตัวแปร คลาส เมท็อด วิธีดำเนินการ เป็นต้น
Exit condition	3. รหัสต้นทางของโปรแกรมภาษาซีชาร์ป ถูกตรวจสอบหมดแล้ว

4.1.3 ยูสเคส Finding Alternative from Method Target

จากตารางที่ 4.3 เป็นยูสเคสที่แสดงเหตุการณ์ที่เกิดขึ้นเมื่อเครื่องมือ IMRD เริ่มทำการสร้างโหนดของต้นไม้เอเอสที

ตารางที่ 4.3 แสดงขั้นตอนการทำงานของยูสเคส Finding Alternative from Method Target

Use Case Name	Finding Alternative from Method Target
Entry condition	1. โหนดของต้นไม้เอเอสที ที่เก็บข้อมูลหรือคุณสมบัติต่างๆ เช่น ตัวแปร คลาส เมท็อด วิธีดำเนินการ
Flow of events	2. ระบบนำโหนดของต้นไม้เอเอสทีจากยูสเคส Parser Generator มาทำการหา เมท็อด Target เพื่อระบุทางเลือกในการมูฟเมท็อด
Exit condition	3. โหนดของต้นไม้เอเอสที ถูกนำไปทำการหาทางเลือกครบทุกโหนด

4.1.4 ยูสเคส Calculate Metrics

จากตารางที่ 4.4 เป็นยูสเคสที่แสดงเหตุการณ์ที่เกิดขึ้นเมื่อเครื่องมือ IMRD นำโหนดของต้นไม้เอเอสทีมาทำการคำนวณเพื่อหาค่าวัดสำหรับการระบุโอกาสในแต่ละทางเลือก

ตารางที่ 4.4 แสดงขั้นตอนการทำงานของยูสเคส Calculate Metrics

Use Case Name	Calculate Metrics
Entry condition	1. โหนดของต้นไม้เอเอสที ที่เก็บข้อมูลหรือคุณสมบัติต่างๆ เช่น ตัวแปร คลาส เมท็อด วิธีดำเนินการ
Flow of events	2. ระบบนำโหนดของต้นไม้เอเอสทีจากยูสเคส Parser Generator และยูสเคส Finding Alternative from Method Target มาทำการหาค่าของมาตรวัดในแต่ละทางเลือก สำหรับการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงตามที่ได้ กำหนดไว้ในเครื่องมือ IMRD ซึ่งค่ามาตรวัดที่คำนวณมีดังนี้ 2.1 Call-based Interaction between Methods (CIM) 2.2 Relative Method Coupling (RMC) 2.3 Interaction coupling between classes (ICBC) 2.4 Interaction method coupling (IMC)
Exit condition	3. โหนดของต้นไม้เอเอสที ถูกนำไปทำการหาค่ามาตรวัดครบทุกโหนด

4.1.5 ยูสเคส View Identification Move Method Refactoring

จากตารางที่ 4.5 เป็นยูสเคสที่แสดงเหตุการณ์ที่เกิดขึ้นเมื่อผู้ใช้งานเลือกเมนู View Identification Move Method Refactoring จากเครื่องมือ IMRD

ตารางที่ 4.5 แสดงขั้นตอนการทำงานของยูสเคส View Identification Move Method Refactoring

Use Case Name	View Identification Move Method Refactoring
Entry condition	1. ผู้ใช้งานเลือกเมนู View Identification Move Method Refactoring
Flow of events	2. ระบบนำค่าที่ได้จากยูสเคส Identification Move Method Refactoring มาแสดงผล
Exit condition	3. ผู้ใช้งานเลือกทำรายการในเมนูอื่น

4.1.6 ยูสเคส Identification of Move Method Refactoring

จากตารางที่ 4.6 เป็นยูสเคสที่แสดงเหตุการณ์ที่เกิดขึ้นเมื่อเครื่องมือ IMRD ทำการระบุโอกาสในแต่ละทางเลือกโดยใช้ทฤษฎีการตัดสินใจ

ตารางที่ 4.6 แสดงขั้นตอนการทำงานของยูสเคส Identification of Move Method Refactoring

Use Case Name	Identification Move Method Refactoring
Entry condition	1. โหนดของต้นไม้เอเอสที ที่เก็บข้อมูลหรือคุณสมบัติต่างๆ เช่น ตัวแปร คลาส เมท็อด วิธีดำเนินการ
Flow of events	2. ระบบนำค่าที่ได้จากยูสเคส Calculate Metrics มาทำการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงโดยใช้ทฤษฎีการตัดสินใจ ตามที่ได้ กำหนดไว้ในเครื่องมือ IMRD ซึ่งใช้ Laplace Criterion เป็นเกณฑ์การตัดสินใจ
Exit condition	3. โหนดของต้นไม้เอเอสที ถูกนำไปทำการหาค่ามาตรวัดครบทุกโหนด

4.1.7 ยูสเคส View Metrics Cohesion & Coupling

จากตารางที่ 4.7 เป็นยูสเคสที่แสดงเหตุการณ์ที่เกิดขึ้นเมื่อผู้ใช้งานเลือกเมนู View Metrics Cohesion & Coupling จากเครื่องมือ IMRD

ตารางที่ 4.7 แสดงขั้นตอนการทำงานของยูสเคส View Metrics Cohesion & Coupling

Use Case Name	View Metrics Cohesion & Coupling
Entry condition	1. ผู้ใช้งานเลือกเมนู View Metrics Cohesion & Coupling
Flow of events	2. ระบบนำโหนดของต้นไม้เอเอสทีจากยูสเคส Parser Generator มาทำการหาค่าของมาตรวัด ตามที่ได้กำหนดไว้ในเครื่องมือ IMRD ซึ่งค่ามาตรวัดที่คำนวณได้มีดังนี้ 2.1 Response for a class (RFC) 2.2 Coupling between objects (CBO) 2.3 Lack of cohesion of methods (LCOM) 2.4 Tight Class Cohesion (TCC) 3. ระบบนำค่าที่ได้จากมาตรวัดแต่ละตัว มาแสดงผล
Exit condition	4. ผู้ใช้งานเลือกทำรายการในเมนูอื่น

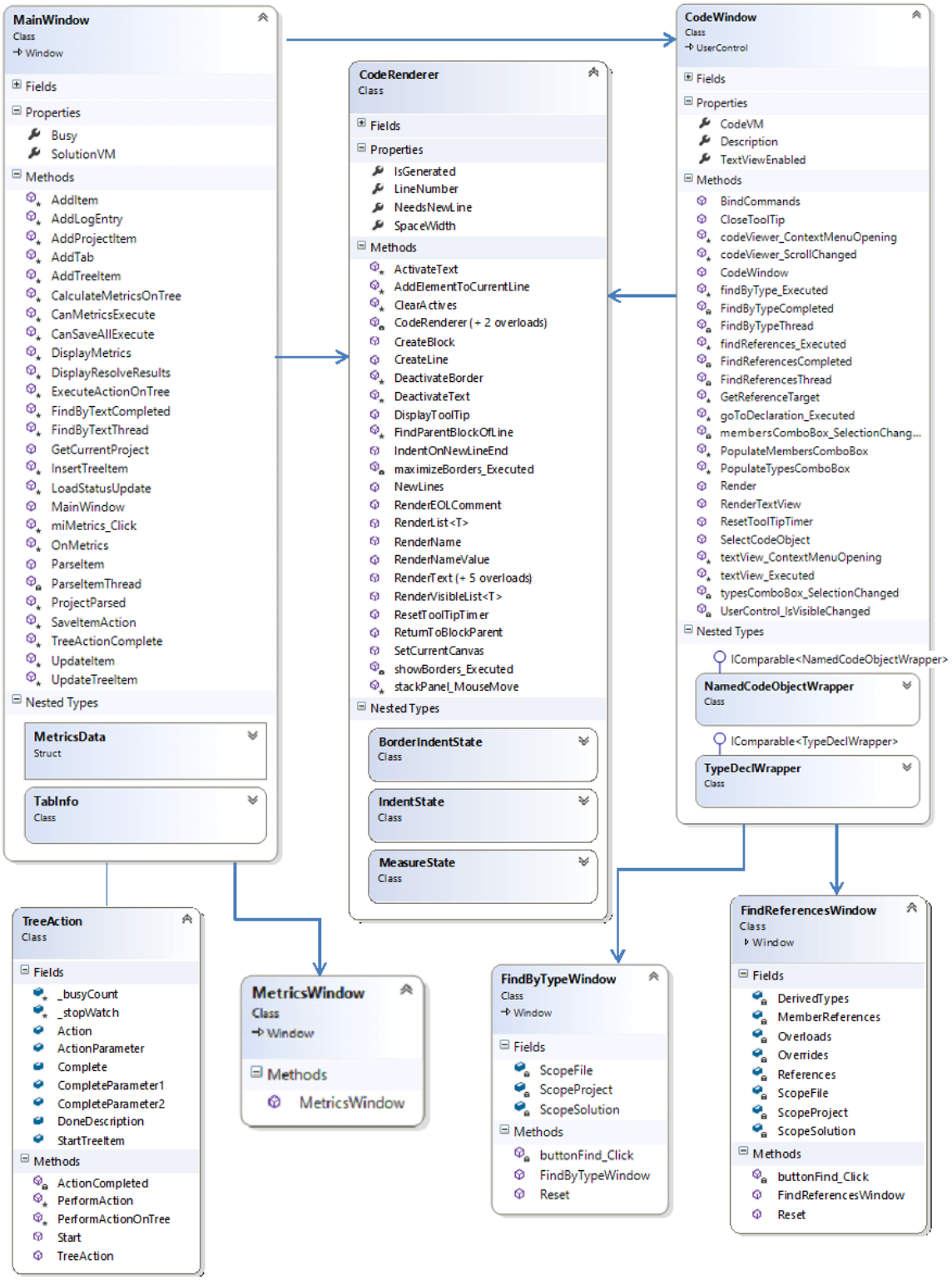
4.2 แผนภาพคลาสแสดงความสัมพันธ์ของการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง

แผนภาพคลาสจะแสดงถึงคลาสและความสัมพันธ์ระหว่างคลาสที่จะอธิบายถึงความสัมพันธ์ขององค์ประกอบของคลาสในการสร้างเครื่องมือ IMRD ได้ดังนี้

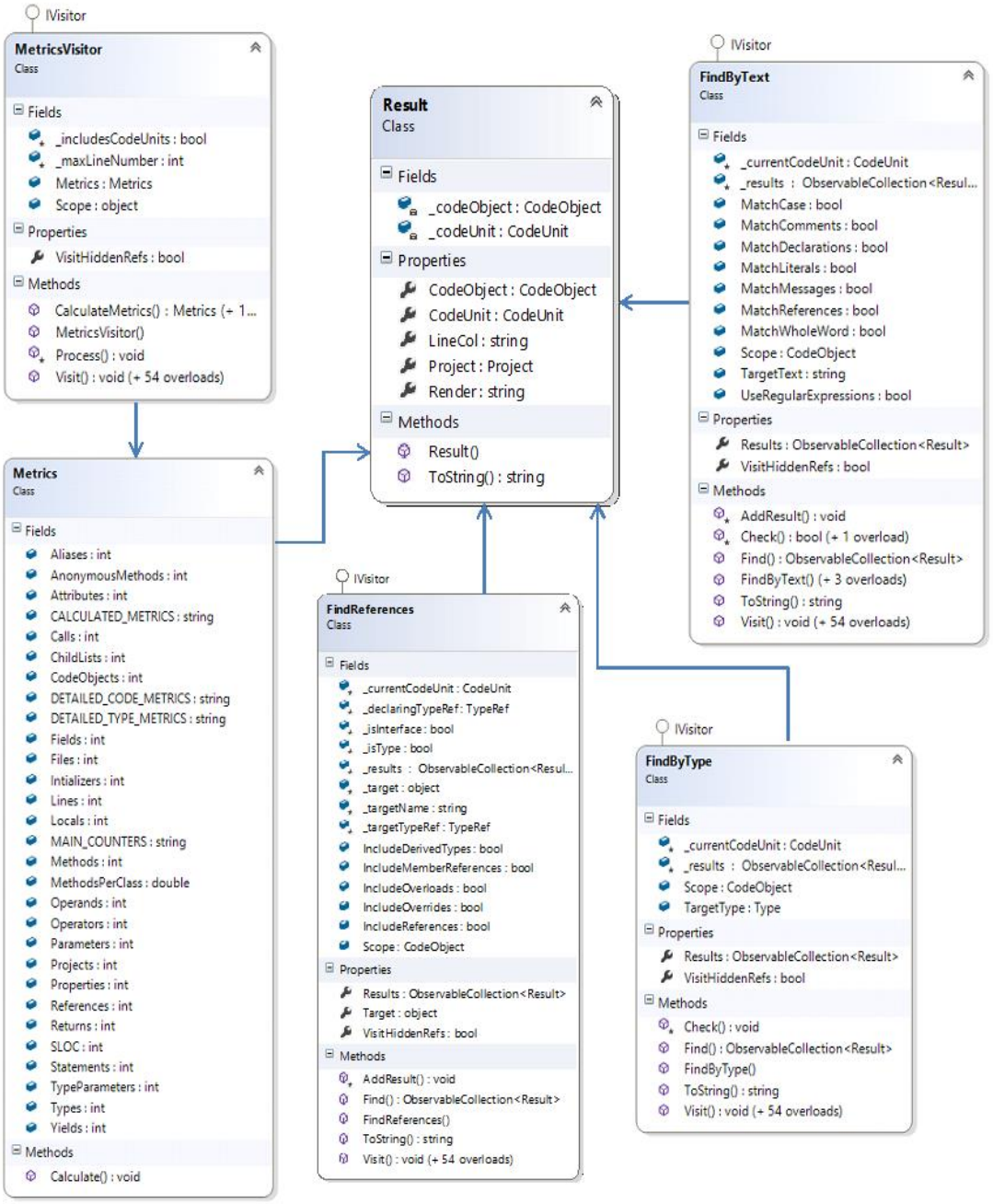
แผนภาพคลาสหน้าจอเป็นคลาสที่แสดงหน้าจอการติดต่อกับเครื่องมือ IMRD โดยมีเมนูหลัก 4 เมนู ได้แก่ เมนู File เมนู Run เมนู View และเมนู About โดยเมนู File มีเมนูย่อย 3 เมนูคือ เมนู Open เมนู Close Solution และเมนู Exit โดยเมนู Open ใช้เพื่อกำหนดชุดข้อมูลของรหัสต้นทางสำหรับใช้ในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง โดยมีการเรียกใช้งานไปยังคลาส MainWindow, CodeWindow, CodeRenderer, TreeAction, FindByTypeWindow และ FindReferencesWindow เมนู Close Solution สำหรับปิดชุดข้อมูลของรหัสต้นทาง เพื่อกำหนดชุดข้อมูลของรหัสต้นทางใหม่ และเมนู Exit เพื่อหยุดการใช้งานโปรแกรมและออกจากเครื่องมือ IMRD เมนู View มีเมนูย่อย 2 เมนูคือ เมนู Metric และเมนู Find โดยเมนู Metric ใช้สำหรับการประมาณค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันของชุดข้อมูลของรหัสต้นทาง และแสดงค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันของคลาสต่างๆ ในชุดข้อมูลของรหัสต้นทาง โดยมีการเรียกใช้งานไปยังคลาส MetricsVistor และ Metrics เมนู Find สำหรับค้นหาตัวแปรหรือเมท็อดในชุดข้อมูลของรหัสต้นทาง เมนู About แสดงรายละเอียดเกี่ยวกับเครื่องมือ IMRD

สำหรับเมนู Run ใช้ในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง โดยมีการเรียกใช้งานไปยังคลาส MetricsVistor, Metrics, FindByText, FindReference, FindByType, Result ซึ่งอยู่ในแผนภาพคลาสการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงโดยคลาส Result แสดงโอกาสการมูฟเมท็อดรีแฟคทอริง ในชุดข้อมูลของรหัสต้นทาง ซึ่งแสดงเมท็อดที่ต้องมูฟไปยังคลาสที่มีการเรียกใช้ โดยคลาส Result มีการเรียกใช้งานคลาส MetricsVistor และ Metrics เพื่อให้ส่งค่าประมาณที่ได้จากมาตรวัด มาพิจารณาประกอบการตัดสินใจกับทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



ภาพที่ 4.2 แผนภาพคลาสแสดงหน้าจอการติดต่อกับเครื่องมือ IMRD



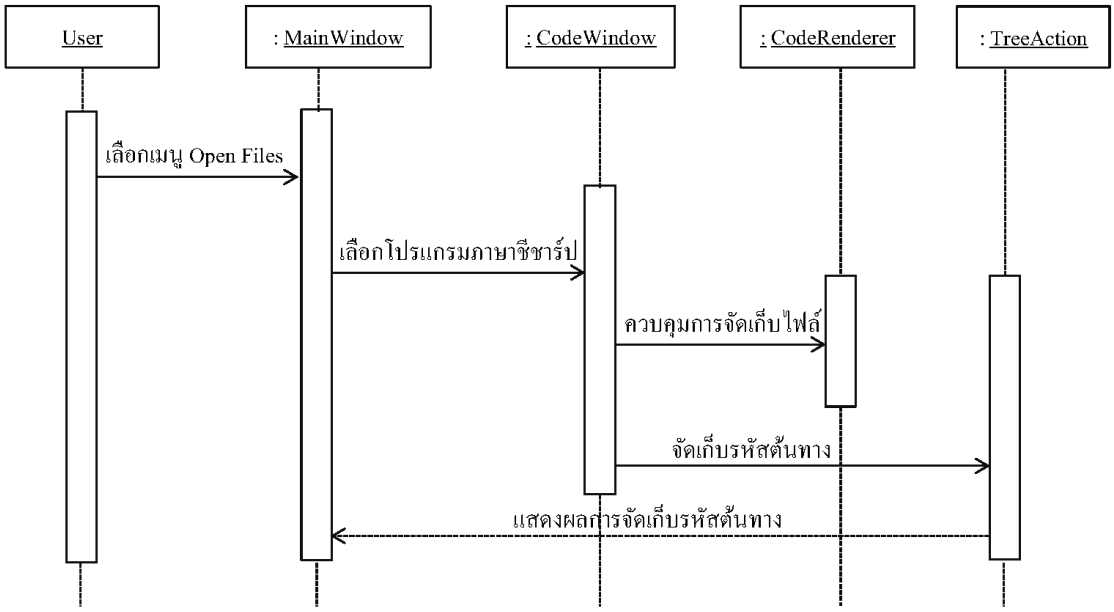
ภาพที่ 4.3 แผนภาพคลาสการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงของเครื่องมือ IMRD

4.3 แผนภาพลำดับการทำงาน

แผนภาพลำดับการทำงาน เป็นแผนภาพที่แสดงให้เห็นถึงการติดต่อกันระหว่างอ็อบเจกต์ รวมถึงการส่งผ่านข้อความระหว่างฟังก์ชันการทำงาน ซึ่งเป็นแผนภาพที่เน้นช่วงเวลาของการเกิดปฏิสัมพันธ์มาอธิบายการสร้างเครื่องมือ IMRD ได้ดังนี้

4.3.1 แผนแสดงภาพการเพิ่มโปรแกรมภาษาซีชาร์ปเข้าสู่ระบบ

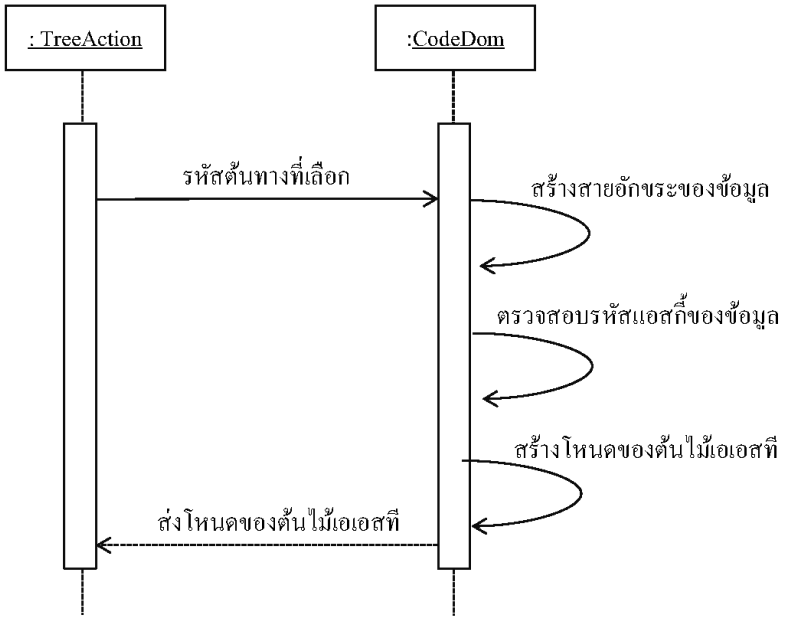
จากภาพที่ 4.4 เป็นแผนภาพที่แสดงการเพิ่มรหัสโปรแกรมภาษาซีชาร์ปเข้าสู่ระบบโดยเริ่มจากผู้ใช้งานทำการเลือกเมนู Add Files จากวัตถุ MainWindow หลังจากนั้นทำการเลือกรหัสโปรแกรมภาษาซีชาร์ปโดยใช้วัตถุ CodeWindow ในการนำรหัสโปรแกรมเข้าสู่ระบบโดยมีวัตถุ CodeRenderer ทำหน้าที่ในการควบคุมการจัดเก็บและจะใช้วัตถุ TreeAction ในการรวบรวมรหัสโปรแกรมเพื่อนำมาใช้ในการหาค่ามาตรวัด และผลของการจัดเก็บรหัสโปรแกรมจะถูกส่งกลับไปยังวัตถุ MainWindow



ภาพที่ 4.4 แผนภาพลำดับการทำงานแสดงการเพิ่มรหัสโปรแกรมภาษาซีชาร์ปเข้าสู่ระบบ

4.3.2 แผนภาพแสดงการทำพาร์เซอร์

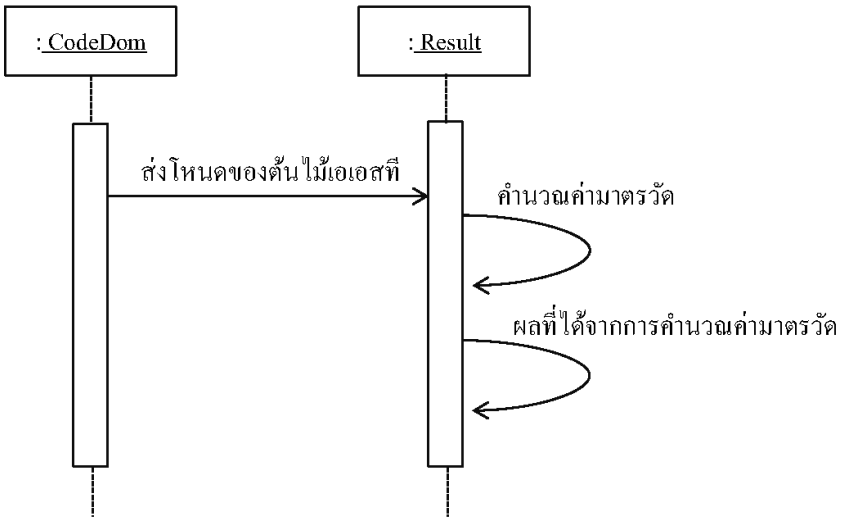
จากภาพที่ 4.5 เป็นแผนภาพที่แสดงการทำพาร์เซอร์โดยระบบจะนำรหัสโปรแกรมที่ได้จากการทำงานในวัตถุ TreeAction มาทำการสร้างสายอักขระของข้อมูลและทำการตรวจสอบความถูกต้องของข้อมูลเพื่อทำการสร้างต้นไม้เอเอสที่ โดยใช้วัตถุ CodeDom ในการสร้าง และผลที่ได้จากการสร้างต้นไม้เอเอสทีจะส่งคืนกลับไปยังวัตถุ TreeAction ต่อไป



ภาพที่ 4.5 แผนภาพลำดับการทำงานแสดงการทำพาร์เซอร์

4.3.3 แผนภาพแสดงการคำนวณค่ามาตรวัด

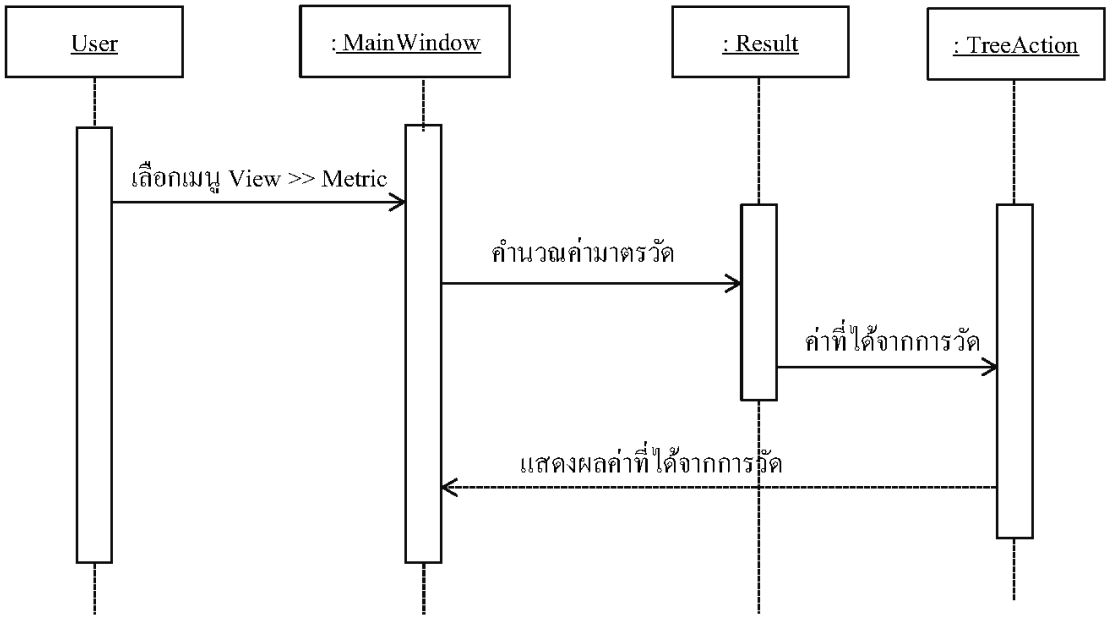
จากภาพที่ 4.6 เป็นแผนภาพที่แสดงการคำนวณค่ามาตรวัดโดยระบบจะนำโหนดของต้นไม้เอเอสทีจากวัตถุ CodeDom มาทำการหาค่ามาตรวัดต่างๆ โดยใช้วัตถุ Result ในการคำนวณค่ามาตรวัด และผลที่ได้จากการคำนวณค่ามาตรวัดจะส่งคืนกลับไปยังวัตถุ Result ต่อไป



ภาพที่ 4.6 แผนภาพลำดับการทำงานแสดงการคำนวณค่ามาตรวัด

4.3.4 แผนภาพแสดงผลการวัดจากการเลือกเมนู View เมนูย่อย Metric

จากภาพที่ 4.7 เป็นแผนภาพที่แสดงผลการวัด โดยเริ่มจากผู้ใช้งานทำการเลือกจากเมนู View เมนูย่อย Metric จากวัตถุ MainWindow หลังจากนั้นระบบจะทำการคำนวณจากวัตถุ Result ซึ่งค่าที่ได้มีความสัมพันธ์กับเมนูที่ผู้ใช้ทำการเลือกและจะถูกนำมาจัดรูปแบบในการแสดงผลโดยวัตถุ TreeAction และผลของรูปแบบที่ได้จะถูกส่งกลับไปยังวัตถุ MainWindow เพื่อนำไปแสดงผล

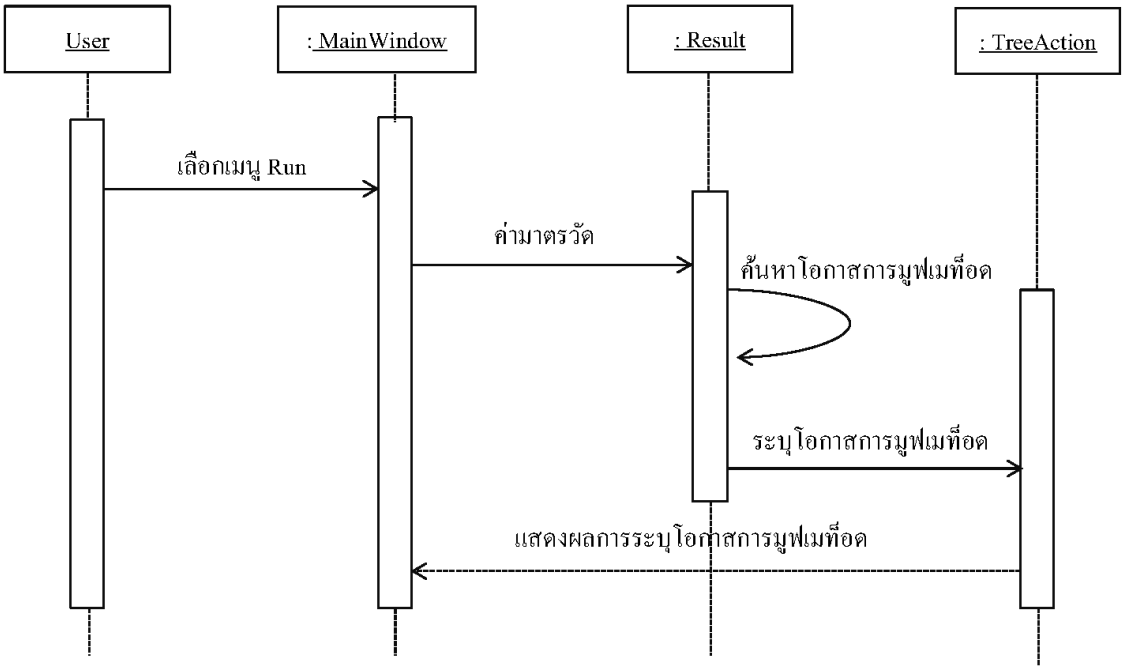


ภาพที่ 4.7 แผนภาพลำดับการทำงานแสดงผลการคำนวณค่ามาตรวัด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.3.5 แผนภาพแสดงการระบุโอกาสการมูฟเมท็อด

จากภาพที่ 4.8 เป็นแผนภาพที่แสดงผลการระบุโอกาสการมูฟเมท็อดการรีแฟคทอริง โดยเริ่มจากผู้ใช้งานทำการเลือกจากเมนู Run จากวัตถุ MainWindow หลังจากนั้นระบบจะทำการคำนวณจากวัตถุ Result ซึ่งค่าที่ได้มีความสัมพันธ์กับเมนูที่ผู้ใช้ทำการเลือกและจะถูกนำมาจัดรูปแบบในการแสดงผลโดยวัตถุ TreeAction และผลของรูปแบบที่ได้จะถูกส่งกลับไปยังวัตถุ MainWindow เพื่อนำไปแสดงผล



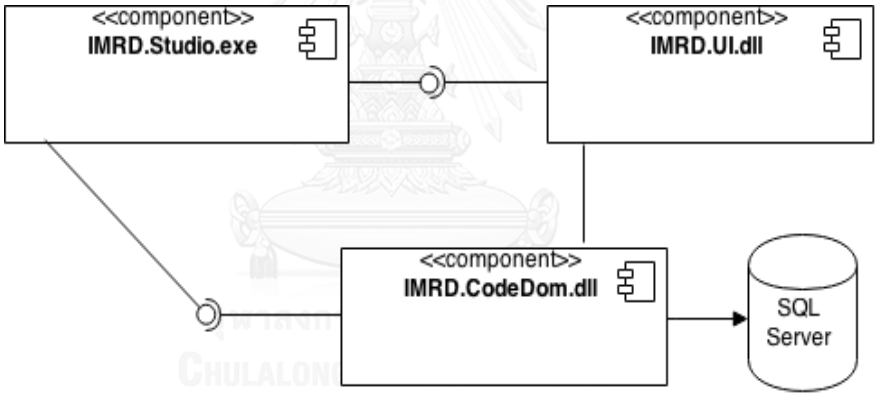
ภาพที่ 4.8 แผนภาพลำดับการระบุโอกาสการมูฟเมท็อด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.4 การพัฒนาเครื่องมือ IMRD Tool

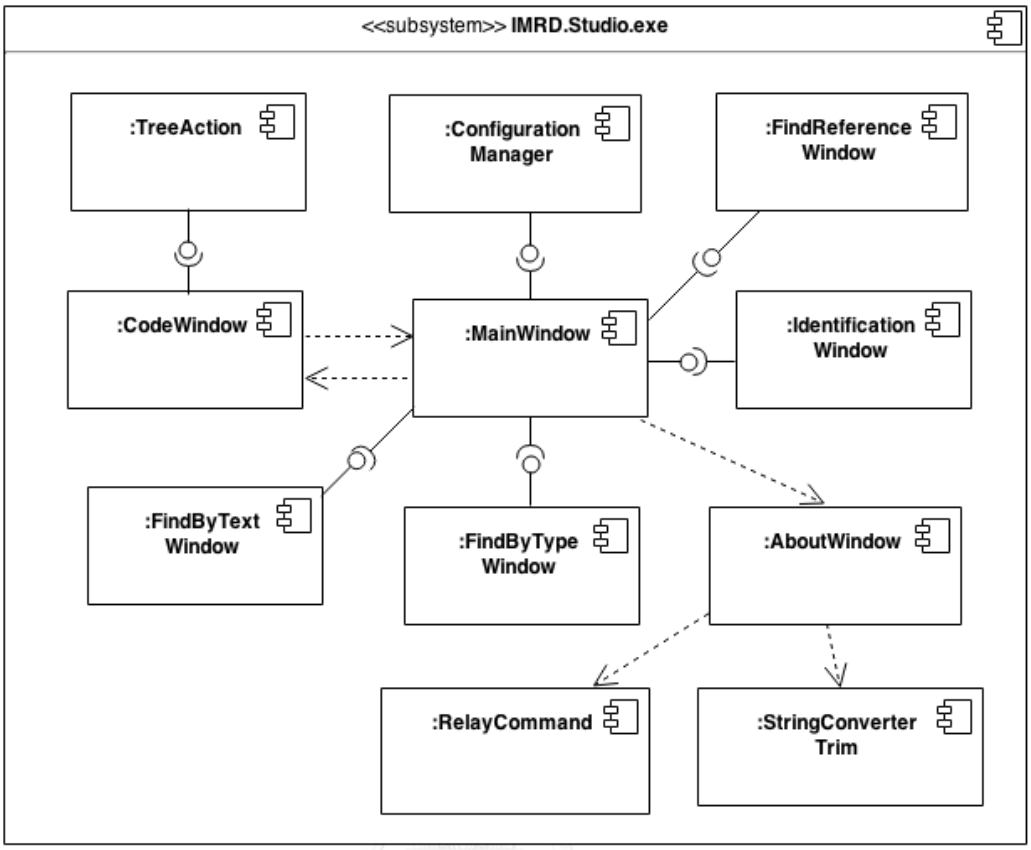
การพัฒนาเครื่องมือ IMRD Tool ที่ประยุกต์ใช้การระบุโอกาสการมูฟเมท็อดรีแฟคทอริงด้วยทฤษฎีการตัดสินใจนี้ ผู้วิจัยได้เลือกใช้ภาษาซีชาร์ป (C#) เป็นภาษาในการเขียนโปรแกรม และได้เลือกใช้เครื่องมือ Visual Studio 2013 ของบริษัทไมโครซอฟต์เป็นเครื่องมือในการเขียนโปรแกรมที่ใช้พัฒนา เครื่องมือประยุกต์ใช้การระบุโอกาสการมูฟเมท็อดรีแฟคทอริงด้วยทฤษฎีการตัดสินใจ และใช้ฐานข้อมูล SQL Server 2012 เป็นฐานข้อมูลในการเก็บข้อมูลรายละเอียดของระบบที่นำมาทดสอบ เช่น ค่ามาตรวัดที่ได้จากระบบต่างๆ ที่นำมาทดสอบ ผู้วิจัยได้ใช้เทคโนโลยี WPF ของบริษัทไมโครซอฟต์ที่นำมาใช้เป็นเฟรมเวิร์กในการเขียนโปรแกรม

การพัฒนาเครื่องมือ IMRD เริ่มจากการออกแบบหน้าจอ เพื่อเป็นส่วนติดต่อกับผู้ใช้งาน จากนั้นออกแบบฐานข้อมูลที่ใช้เก็บข้อมูลที่ได้จากการประมวลผลในการทดสอบระบบ แล้วจึงเขียนโปรแกรม สุดท้ายทำการทดสอบโปรแกรมเพื่อหาข้อบกพร่อง ข้อผิดพลาด เพื่อแก้ไขโปรแกรมให้ตรงความต้องการต่อไป



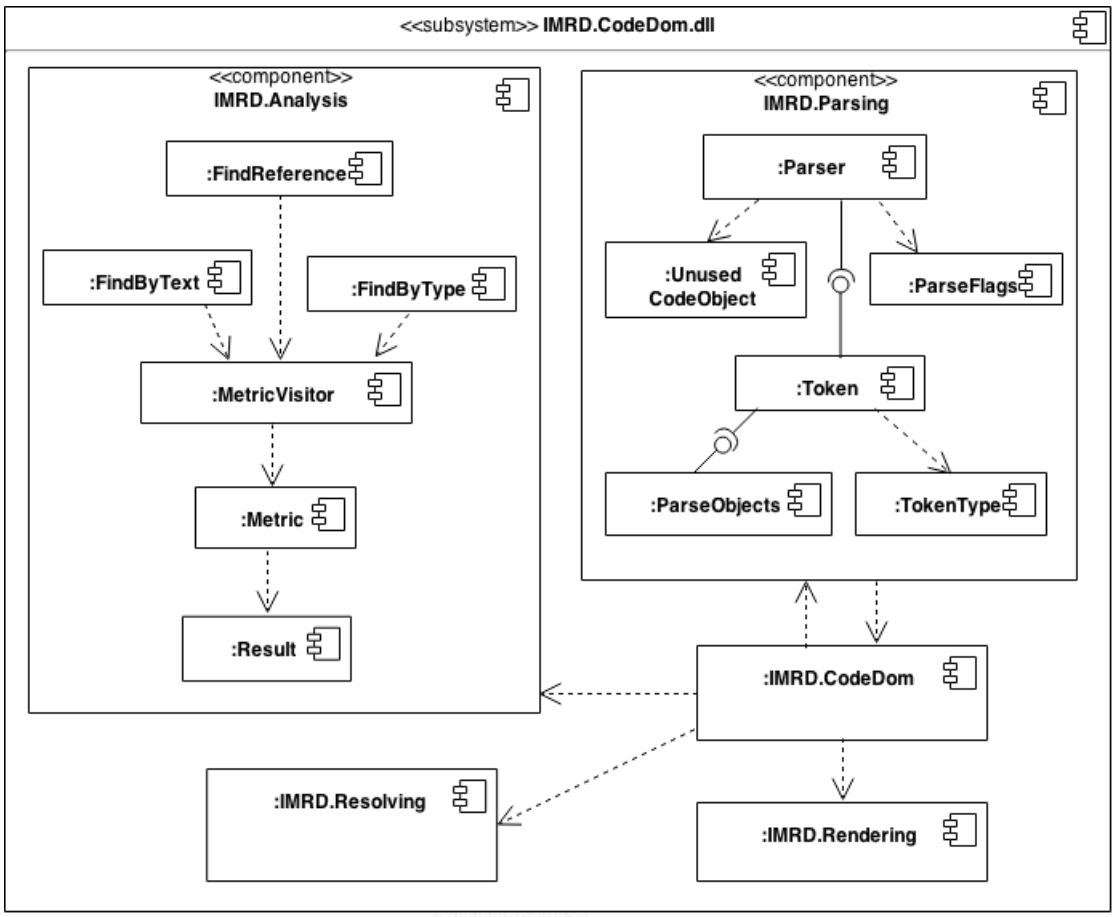
ภาพที่ 4.9 องค์ประกอบของเครื่องมือ IMRD

จากภาพที่ 4.9 อธิบายองค์ประกอบของเครื่องมือ IMRD สามารถแบ่งออกได้เป็น 4 องค์ประกอบ คือ IMRD.Studio.exe, IMRD.UI.dll, IMRD.CodeDom.dll และ SQL Server โดย IMRD.UI.dll และ IMRD.CodeDom.dll ร้องขอบริการไปยัง IMRD.Studio.exe และข้อมูลที่ได้จาก IMRD.CodeDom.dll จะถูกส่งไปบันทึกหรือปรับปรุงในฐานข้อมูล SQL Server ซึ่งในแต่ละองค์ประกอบจะมีองค์ประกอบย่อยซึ่งจะอธิบายรายละเอียดในส่วนต่อไป



ภาพที่ 4.10 องค์ประกอบของ IMRD.Studio.exe

จากภาพอธิบายองค์ประกอบของ IMRD.Studio.exe สามารถแบ่งออกได้เป็น 11 องค์ประกอบตามภาพที่ 4.10 ซึ่ง MainWindow ร้องขอบริการไปยัง ConfigurationManager, FindReferenceWindow, FindByTextWindow, FindByTypeWindow และ Identification-Window โดย MainWindow มีการแลกเปลี่ยนข้อมูลกับ CodeWindow ซึ่ง CodeWindow ก็ร้องขอบริการไปยัง TreeAction และ MainWindow ขอใช้ข้อมูลจาก AboutWindow ซึ่ง AboutWindow ก็ขอใช้ข้อมูลจากRelayCommand และ StringConverterTrim



ภาพที่ 4.11 องค์ประกอบของ IMRD.CodeDom.dll

จากภาพอธิบายองค์ประกอบของ IMRD.CodeDom.dll สามารถแบ่งออกได้เป็น 5 องค์ประกอบตามภาพที่ 4.11 ซึ่ง IMRD.CodeDom มีการแลกเปลี่ยนข้อมูลกับ IMRD.Parsing ซึ่งใน IMRD.Parsing มีองค์ประกอบย่อย 6 องค์ประกอบ โดย Parser ขอใช้ข้อมูลจาก ParseFlags และ UnusedCodeObject และ Parser ร้องขอบริการไปยัง Token ซึ่ง Token ขอใช้ข้อมูลจาก TokenType และร้องขอบริการไปยัง ParseObject

IMRD.CodeDom ขอใช้ข้อมูลจาก IMRD.Resolving, IMRD.Rendering และ IMRD.Analysis ซึ่งใน IMRD. Analysis มีองค์ประกอบย่อย 6 องค์ประกอบ โดย Result ขอใช้ข้อมูลจาก Metric และ Metric ขอใช้ข้อมูลจาก MetricVisitor ซึ่ง MetricVisitor ขอใช้ข้อมูลจาก FindReference, FindByText และ FindByType

บทที่ 5
การประเมินการระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อดด้วยวิธีของทฤษฎีการตัดสินใจ

ในบทนี้จะประเมินการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง โดยเริ่มจากการนำระบบที่จะทดสอบมาการคำนวณมาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันก่อนรีแฟคทอริง และค้นหาทางเลือกในการมูฟเมท็อด การเลือกทางเลือกในการมูฟเมท็อดรีแฟคทอริงโดยใช้วิธีของทฤษฎีการตัดสินใจ การทำรีแฟคทอริง การคำนวณมาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันหลังรีแฟคทอริง และประเมินความสามารถของการปรับปรุงคุณภาพของรหัสต้นทาง โดยประเมินจากค่าของมาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันก่อนและหลังทำรีแฟคทอริง

5.1 ระบบซอฟต์แวร์ที่นำมาทดสอบ

ระบบที่นำมาทดสอบในงานวิจัยนี้ เป็นระบบที่พัฒนาด้วยภาษาซีชาร์ป โดยในงานวิจัยนี้จะพิจารณาเฉพาะรหัสต้นทาง ไม่รวมถึงไลบรารี่ที่ใช้ในระบบ

ตารางที่ 5.1 รายชื่อและรายละเอียดของระบบที่นำมาทดสอบ

#	ชื่อระบบ	จำนวนคลาส	จำนวนเมท็อด	รายละเอียด
1	Repairs Management System[1]	36	586	ระบบรับซ่อมคอมพิวเตอร์และโทรศัพท์ ถูกเขียนโดยใช้ภาษา C# และฐานข้อมูล SQL Server 2012 คุณสมบัติหลัก คือ 1) ล็อกอินเข้าสู่ระบบ 2) การบริหารจัดการลูกค้า 3) ใบเสร็จรับเงินใบแจ้งหนี้
2	Restaurant Billing System[1]	41	736	ระบบการเรียกเก็บเงินร้านอาหาร ได้รับการพัฒนาโดยใช้ภาษา C# และฐานข้อมูล SQL Server 2012 คุณสมบัติหลัก คือ

[1] http://www.sourcecodester.com/

#	ชื่อระบบ	จำนวนคลาส	จำนวนเมท็อด	รายละเอียด
				1) การบริหารการจัดการ ลูกค้า 2) การบริหารการจัดการ การขาย 3) การบริหารการจัดการ ใบเสร็จ 4) การบริหารการจัดการ รายงาน
3	Book Store[2]	65	318	ระบบการจัดการร้านหนังสือได้รับการพัฒนาโดยใช้ภาษา C# และ . NET Framework 2.0 บนระบบปฏิบัติการวินโดวส์ คุณสมบัติหลัก คือ 1) การบริหารจัดการหนังสือ เช่น จัดเก็บ คัดลอก แสดง และลบ 2) ผู้ใช้สามารถค้นหาหนังสือตามชื่อเรื่อง ชื่อผู้แต่ง ปีที่พิมพ์ สำนักพิมพ์ เป็นต้น 3) ใช้ฐานข้อมูลกลางเป็นพื้นที่เก็บข้อมูล
4	Ordering Administration System[3]	12	23	ระบบการสั่งซื้อของผู้ดูแลระบบได้รับการพัฒนาโดยใช้ภาษา C# . NET Framework 4.5 และฐานข้อมูล SQL Server 2012 บนระบบปฏิบัติการวินโดวส์ คุณสมบัติหลัก คือ 1) การบริหารการจัดการ ลูกค้า 2) การบริหารการจัดการ การสั่งซื้อ 3) ค้นหาข้อมูลการสั่งซื้อ

[2] http://www.codeproject.com/Articles/63715/BookStore

[3] http://www.codeproject.com/Articles/171002/LightSwitch-Online-Ordering-System

#	ชื่อระบบ	จำนวนคลาส	จำนวนเมท็อด	รายละเอียด
5	Graph of Function[4]	31	142	ระบบการสร้างกราฟฟังก์ชันโดยผู้ใช้จะต้องกำหนดขอบเขตของกราฟ ได้รับการพัฒนาโดยใช้ภาษา C# คุณสมบัติหลัก คือ 1) สร้างจำนวนฟังก์ชันกราฟิก 2) การตั้งค่าการแสดงผล 3) ลบและเปลี่ยนฟังก์ชัน
6	Image Fan[5]	20	103	ระบบรีวิวภาพได้รับการพัฒนาโดยใช้ภาษา C# และ . NET Framework 2.0 คุณสมบัติหลัก คือ 1) เลือกไฟล์จากไดร์ฟ และเลือกรายการภาพ 2) เลื่อนภาพขึ้นลงโดยใช้ปุ่มลูกศรและแถบเลื่อน

5.2 การระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อด

ระบบ Repairs Management System

การค้นหาคลาสเพื่อระบุโอกาสการย้ายเมท็อดเป้าหมาย ไปยังคลาสที่เหมาะสม ตามการเรียกใช้เมท็อดเป้าหมายจากภายในคลาสและภายนอกคลาส

1) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหนึ่งคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้สามารถเสนอแนะเมท็อดที่จะต้องย้ายไปยังคลาสที่เหมาะสม โดยไม่ต้องคำนวณค่ามาตรวัดและไม่ต้องใช้เกณฑ์ลาปลาสในการตัดสินใจ โดยเมท็อด frmInvoice_FromClosing เมท็อด frmProductRepair_Load เมท็อด delete_records และเมท็อด cmbCustomerName_Format

[4] http://www.codeproject.com/

[5] http://www.codeproject.com/Articles/60975/ImageFan

มีการเรียกใช้จากคลาสปลายทางหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสต้นทาง ดังนั้นจึงเสนอแนะเมท็อดให้ย้ายเมท็อดไปยังคลาสปลายทางที่เรียกใช้เมท็อดนั้น ดังแสดงในตารางที่ 5.2

ตารางที่ 5.2 การเรียกใช้เมท็อดภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Repairs Management

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
frmInvoice_From Closing	frmBilling	frmBillingRecords	0	1
frmProductRepair _ Load	frmBilling	frmBillingRecords	0	1
delete_records	frmRegistration	frmRegistered UsersDetails	0	1
cmbCustomerNa me_Format	frmBillingRecords	CustomerTable Adapter	0	3

2) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหลายคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้ต้องคำนวณค่ามาตรวัดและต้องใช้เกณฑ์ลาปลาสในการตัดสินใจ เพราะมีการเรียกใช้เมท็อดจากคลาสปลายทางหลายคลาส โดยเมท็อด subtol มีการเรียกใช้จากคลาสปลายทาง 3 คลาส คือ คลาส frmRegisteredUsersDetail คลาส Invoice_InfoTableAdapter และคลาส frm-CustomersRecord1 ดังแสดงในตารางที่ 5.3

ตารางที่ 5.3 การเรียกใช้เมท็อดภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของเมท็อด subtol

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
subtol	frmBilling	frmRegisteredUsersDetails	0	7
	frmBilling	Invoice_InfoTableAdapter	0	3
	frmBilling	frmCustomersRecord1	0	2

การคำนวณค่ามาตรวัด ในกรณีนี้คำนวณด้วยมาตรวัด CIM, RMC และ ICBC โดยไม่ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาสดังแสดงในตารางที่ 5.4

ตารางที่ 5.4 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับเมท็อด subtol

ทางเลือก ลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
1	0.550	0.460	0.210	0.407
2	0.370	0.010	0.150	0.177
3	0.330	0.020	0.010	0.120

การเลือกคลาสที่เหมาะสม โดยการเลือกด้วยวิธีลาปลาส จะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด คือทางเลือกที่ 1 ดังนั้น เลือกย้ายเมท็อด subtol จากคลาส frmBilling ไปยังคลาส frm-RegisteredUsersDetails

3) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอก และมีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้ต้องคำนวณค่ามาตรวัดและต้องใช้เกณฑ์ลาปลาสในการตัดสินใจ เพราะมีการเรียก ใช้เมท็อดจากคลาสปลายทางและมีการเรียกใช้เมท็อดภายในคลาสต้นทาง ซึ่งในกรณีนี้มีเมท็อด Calculate เมท็อด FillCombo เมท็อด frmSalesRecord_FromClosing เมท็อด GetData เมท็อด Fill เมท็อด Reset เมท็อด Autocomplete และเมท็อด Email_Address_Validating ดังแสดงในตารางที่ 5.5

ตารางที่ 5.5 การเรียกใช้เมท็อดภายนอกคลาสและการเรียกใช้เมท็อดภายในคลาสของระบบ Repairs Management

เมท็อด	**คลาสต้นทาง**	**คลาสปลายทาง**	**ต้นทาง**	**ปลายทาง**
Calculate	frmBilling	frmBillingRecords	3	2
	frmBilling	CustomerTable Adapter	3	3
	frmBilling	frmCustomers Record1	3	4
FillCombo	frmBillingRecords	frmBilling	1	4

ตารางที่ 5.5 การเรียกใช้เมท็อดภายนอกคลาสและการเรียกใช้เมท็อดภายในคลาสของระบบ Repairs Management (ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
FillCombo	frmBillingRecords	RegistrationTable Adapter	1	2
	frmBillingRecords	frmCustomers	1	2
frmSalesRecord_ FromClosing	frmBillingRecords	frmCustomers	1	1
GetData	CustomerTable Adapter	frmCustomers Record1	2	8
	CustomerTable Adapter	frmRegistered UsersDetails	2	1
GetData	CustomerTable Adapter	Invoice_InfoTable Adapter	2	5
	CustomerTable Adapter	frmLoginDetails	2	6
	CustomerTable Adapter	RegistrationTable Adapter	2	4
	CustomerTable Adapter	ProductRepaired TableAdapter	2	6
Fill	RegistrationTable Adapter	Invoice_InfoTable Adapter	2	7
	RegistrationTable Adapter	ProductRepaired TableAdapter	2	4
Reset	frmCustomers	Invoice_InfoTable Adapter	2	1
Autocomplete	frmRegistration	frmRegistered UsersDetails	2	5
Email_Address_ Validating	frmRegistration	frmRegistered UsersDetails	1	2

การคำนวณค่ามาตรวัด ในกรณีนี้ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะมีการเรียกใช้เมท็อดเป้าหมายภายในคลาส จากนั้นคำนวณด้วยมาตรวัด CIM, RMC และ ICBC

โดยแบ่งเป็น 2 ขั้นตอน

ขั้นตอนที่ 1 คำนวณด้วยมาตรวัด IMC เพื่อจะคำนวณการเรียกใช้ภายในคลาส และการเรียกใช้จากภายนอกคลาส ของเมท็อดเป้าหมาย ดังแสดงในตารางที่ 5.6

ตารางที่ 5.6 การคำนวณด้วยมาตรวัด IMC ของระบบ Repairs Management

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
Calculate	frmBilling	frmBillingRecords	0.470
	frmBilling	CustomerTableAdapter	0.550
	frmBilling	frmCustomersRecord1	0.680
FillCombo	frmBillingRecords	RegistrationTableAdapter	0.670
	frmBillingRecords	frmBilling	0.850
	frmBillingRecords	frmCustomers	0.640
frmSalesRecord_FromClosing	frmBillingRecords	frmCustomers	0.570
GetData	CustomerTable Adapter	frmCustomers Record1	0.830
	CustomerTable Adapter	frmRegistered UsersDetails	0.330
	CustomerTable Adapter	Invoice_InfoTable Adapter	0.750
	CustomerTable Adapter	frmLoginDetails	0.330
	CustomerTable Adapter	RegistrationTable Adapter	0.780
	CustomerTable Adapter	ProductRepaired TableAdapter	0.700
Reset	frmCustomers	Invoice_InfoTable Adapter	0.330

ตารางที่ 5.6 การคำนวณด้วยมาตรวัด IMC ของระบบ Repairs Management (ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
Fill	RegistrationTable Adapter	Invoice_InfoTable Adapter	0.790
	RegistrationTable Adapter	ProductRepaired TableAdapter	0.670
Email_Address_ Validating	frmRegistration	frmRegistered UsersDetails	0.670
Autocomplete	frmRegistration	frmRegistered UsersDetails	0.500

ขั้นตอนที่ 2 เลือกทางเลือกที่มีค่า IMC >= 0.70 มาคำนวณด้วยมาตรวัด CIM, RMC, ICBC ในแต่ละทางเลือกเพื่อพิจารณาจะย้ายเมท็อดไปที่คลาสใด โดยเมท็อดที่เลือกมีเมท็อด Fill ดังแสดงในตารางที่ 5.7 และเมท็อด GetData ดังแสดงในตารางที่ 5.8

ตารางที่ 5.7 การตัดสินใจเลือกคลาสที่เหมาะสม สำหรับเมท็อด Fill

ทางเลือก (#Id)	เหตุการณ์			$(U(a_i))$
	CIM	RMC	ICBC	
1	0.650	0.150	0.180	0.327
2	0.370	0.075	0.151	0.199

ตารางที่ 5.8 การตัดสินใจเลือกคลาสที่เหมาะสม สำหรับเมท็อด GetData

ทางเลือก (#Id)	เหตุการณ์			$(U(a_i))$
	CIM	RMC	ICBC	
1	0.550	0.640	0.010	0.400
2	0.350	0.010	0.150	0.170
3	0.730	0.620	0.150	0.500
4	0.330	0.200	0.210	0.247

การเลือกคลาสที่เหมาะสม โดยจะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด ในกรณีของเมท็อด GetData คือ ทางเลือกที่ 3 มีค่าเท่ากับ 0.500 และในกรณีของเมท็อด Fill คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.327 ดังนั้นเลือกย้ายเมท็อด GetData จากคลาส CustomerTableAdapter ไปยังคลาส RegistrationTableAdapter และย้ายเมท็อด Fill จากคลาส RegistrationTableAdapter ไปยังคลาส Invoice_InfoTableAdapter

การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม จากระบบ Repairs Management สามารถเสนอแนะเมท็อดที่จะต้องย้าย ดังแสดงในตารางที่5.9

ตารางที่ 5.9 เมท็อดที่แนะนำในการย้ายไปยังคลาสที่เหมาะสม ของระบบ Repairs Management

เมท็อด	คลาสต้นทาง	คลาสปลายทาง
frmInvoice_FromClosing	frmBilling	frmBillingRecords
frmProductRepair_Load	frmBilling	frmBillingRecords
delete_records	frmRegistration	frmRegisteredUsersDetails
Subtol	frmBilling	frmRegisteredUsersDetails
FillCombo	frmBillingRecords	frmBilling
GetData	CustomerTableAdapter	RegistrationTableAdapter
Fill	RegistrationTableAdapter	Invoice_InfoTableAdapter
cmbCustomerName_Format	frmBillingRecords	CustomerTableAdapter

การแสดงรายละเอียดในระบบ Restaurant Billing System, Book Store, Graph of Function, Ordering Administration System และ Image Fan จะแสดงในภาคผนวก ข

5.3 การประเมินความสามารถในการทำรีแฟคทอริงหลังจากใช้ทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อด

การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ ก่อนและหลังการทำรีแฟคทอริงโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสการมูฟเมท็อด โดยการประเมินจะพิจารณาจากหลักเกณฑ์ของมาตรวัดที่ได้ ค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐาน

5.3.1 ระบบ Repairs Management System

การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ก่อนและหลังการทำรีแฟคทอริงของระบบ Repairs Management ดังแสดงในตารางที่5.10

ตารางที่ 5.10 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Repairs Management

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
frmBilling	7.00	0.33	5.00	3.00	3.00↓	0.67↑	3.00 ↓	2.00↓
frmBilling Records	3.00	0.33	1.00	5.00	2.00↓	0.53↑	2.00↑	5.00
frmRegistration	0	0.67	2.00	6.00	2.00↑	0.67	2.00	5.00↓
frmRegistered UsersDetails	1.00	0.67	3.00	3.00	0 ↓	0.33↓	1.00↓	2.00↓
Registration TableAdapter	0	1.00	1.00	5.00	0	1.00	0 ↓	4.00↓
Customer TableAdapter	1.00	1.00	1.00	5.00	0 ↓	1.00	0 ↓	3.00↓
Invoice_Info TableAdapter	1.00	0.33	2.00	4.00	1.00	0.67↑	1.00↓	3.00↓
frmCustomers Record1	2.00	0.67	2.00	3.00	1.00 ↓	1.00↑	2.00	3.00
frmCustomers	3.00	0.67	0	4.00	0 ↓	1.00↑	0	1.00↓

ตารางที่ 5.10 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Repairs Management (ต่อ)

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
frmLogin Details	2.00	0.67	1.00	3.00	1.00↓	0.33↓	0 ↓	2.00↓
ProductRepairedTableAdapter	1.00	0.67	1.00	5.00	1.00	0.67	1.00	5.00
frmRegistration	0	0.33	0	4.00	2.00↑	0.33	0	3.00↓
Mean	1.750	0.612	1.583	3.577	1.083↓	0.683↑	1.000↓	3.167↓
S.D.	1.960	0.241	1.379	1.337	0.996↓	0.270↑	1.045↓	1.030↓
Max	7.00	1.00	5.00	6.00	3.00	1.00	3.00	5.00
Min	0	0.33	0	3.00	0	0.33	0	1.00

จากระบบ Repairs Management System ค่าของมาตรวัดที่ประเมินได้ เมื่อนำระบบมาคำนวณด้วยมาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันก่อนรีแฟคทอริงและหลังรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสมาเปรียบเทียบกัน ดังที่แสดงในตารางที่ 5.10 สามารถปรับปรุงรหัสต้นทางให้มีคุณภาพที่ดีขึ้นได้โดยพิจาณาจากมาตรวัด เมื่อคิดเป็นร้อยละของมาตรวัดจากทั้งระบบโดยนำค่า Mean ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดยมาตรวัด LCOM ลดลงร้อยละ 38.10 TCC เพิ่มขึ้นร้อยละ 7.16 RFC ลดลงร้อยละ 58.33 CBO เพิ่มขึ้นร้อยละ 40.99

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

5.3.2 ระบบ Restaurant Billing System

การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ก่อนและหลังการทำรีแฟคทอริงของระบบ Restaurant Billing System ดังแสดงในตารางที่5.11

ตารางที่ 5.11 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Restaurant Billing System

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
frmChange Password	1.00	0.33	0	3.00	0 ↓	0.33	0	2.00↓
frmChange Password	1.00	0.33	0	3.00	0 ↓	0.33	0	2.00↓
Category TableAdapter	1.00	0.33	1.00	5.00	0 ↓	0.33	2.00	5.00
Registration TableAdapter	0	0.67	2.00	7.00	2.00↑	0.67	2.00↑	6.00↓
RSM_DBData Set	0	0.67	0	3.00	0	0.33↓	0	2.00↓
Invoice_Info TableAdapter	4.00	0.33	1.00	6.00	2.00↓	1.00↑	0 ↓	4.00↓
ProductTable Adapter	0	1.00	1.00	5.00	0	1.00	0 ↓	4.00↓
ProductSold TableAdapter	1.00	0.33	1.00	8.00	1.00	0.33	1.00 ↑	7.00↓
TableAdapter Manager	3.00	0.67	2.00	4.00	1.00↓	1.00↑	2.00	3.00↓
Customer TableAdapter	5.00	0.67	0	3.00	2.00↓	1.00↑	0	1.00
frmMainMenu	0	0.33	2.00	1.00	0	0.67↑	0 ↓	1.00↓
frmSplash	1.00	0.67	2.00	4.00	1.00	0.67	2.00	4.00
frmAbout	0	0.33	0	6.00	1.00↑	0.33	0	5.00↓

ตารางที่ 5.11 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Restaurant Billing System (ต่อ)

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
frmRecovery Password	1.00	0.33	0	3.00	1.00	0.33	0	2.00↓
frmLogin	1.00	0.33	1.00	5.00	0 ↓	0.33	2.00↑	5.00
frmProduct	0	0.67	2.00	7.00	1.00↑	0.67	2.00↑	6.00↓
frmProducts Record	0	0.67	0	3.00	0	0.33↓	0 ↓	2.00↓
frmCustomers Record	3.00	1.00	1.00	11.00	2.00↓	1.00	0 ↓	4.00↓
frmRegistered UsersDetails	5.00	1.00	1.00	5.00	3.00↓	1.00	0 ↓	4.00↓
frmCustomers	4.00	0.33	1.00	8.00	2.00↓	0.33	1.00	7.00↓
frmSales Record1	2.00	0.67	2.00	3.00	1.00↓	1.00↑	2.00↑	3.00
frmSalesRecord	3.00	0.67	0	4.00	2.00↓	1.00↑	0	1.00↓
frmLogin Details	1.00	0.67	2.00	11.00	1.00	0.33↓	0↓	8.00↓
frmRegistration	1.00	0.67	2.00	8.00	1.00	0.67	2.00	6.00↓
frmCategory Record	0	0.33	0	6.00	2.00↓	0.33	0	5.00↓
frmSales	1.00	0.33	2.00	3.00	0 ↓	0.33	0↓	2.00↓
frmCategory	1.00	0.33	1.00	5.00	0 ↓	0.33	2.00↓	5.00
Mean	1.500	0.551	1.080	4.269	1.000↓	0.602↑	0.769↓	4.000↓
S.D.	1.606	0.232	0.951	2.491	0.894↓	0.300↑	0.812	2.000↓
Max	5.00	1.00	2.00	11.00	3.00	1.00	2.00	8.00
Min	0	0.33	0	1.00	0	0.33	0	1.00

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากระบบ Restaurant Billing System ค่าของมาตรวัดที่ประเมินได้ เมื่อนำระบบมาคำนวณด้วยมาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันก่อนรีแฟคทอริงและหลังรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสมาเปรียบเทียบกัน ดังแสดงในตารางที่ 5.11 สามารถปรับปรุงรหัสต้นทางให้มีคุณภาพที่ดีขึ้นได้โดยพิจาณาจากมาตรวัด เมื่อคิดเป็นร้อยละของมาตรวัดจากทั้งระบบโดยนำค่า Mean ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดยมาตรวัด LCOM ลดลงร้อยละ 50.00 TCC เพิ่มขึ้นร้อยละ 5.03 RFC ลดลงร้อยละ 31.08 CBO ลดลงร้อยละ 26.92

5.3.3 ระบบ Book Store

การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ก่อนและหลังการทำรีแฟคทอริงของระบบ Book Store ดังแสดงในตารางที่ 5.12

ตารางที่ 5.12 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Book Store

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
ByteArray	0	0.33	0	2.00	0	0.33	0	2.00
BookEntities Exception	1.00	0.33	1.00	5.00	0 ↓	0.33	2.00↑	4.00↓
OptionsForm	1.00	0.67	2.00	7.00	2.00↑	0.67↓	2.00	6.00↓
RegistryAccess Exception	0	0.33	0	3.00	0	0.67↑	0	2.00↓
DiscAccess Exception	0	1.00	1.00	4.00	0 ↓	1.00	0 ↓	4.00
ANDQuery Type	2.00	0.33	1.00	5.00	1.00 ↓	1.00↑	0 ↓	4.00↓
Database Exception	1.00	0.33	1.00	8.00	1.00	0.33	1.00	7.00↓
ORQueryType	2.00	0.67	2.00	3.00	1.00↓	1.00↑	2.00	3.00
MultipleInstancesErrorForm	3.00	0.67	0	3.00	2.00↓	1.00↑	0	1.00↓

ตารางที่ 5.12 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Book Store (ต่อ)

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
ActivateAction	0	0.33	2.00	9.00	0	0.33	0 ↓	8.00↓
Deactivate Action	1.00	0.67	2.00	5.00	1.00	0.67	2.00	5.00
Book	19.00	0.43	5.00	6.00	7.00↓	0.67↑	1.00↓	3.00↓
BookDetails	5.00	0.33	0	3.00	2.00↓	0.33	0	2.00↓
BooksData	2.00	0.33	1.00	5.00	2.00	0.33	2.00↑	5.00
MainForm	3.00	0.37	0	3.00	1.00↓	0.73↑	0	2.00↓
AddNewItem Form	2.00	0.67	2.00	5.00	1.00↓	1.00↑	1.00↓	4.00 ↓
CustomBuffer	0	1.00	1.00	6.00	0	1.00	0 ↓	4.00↓
DeleteAction	1.00	0.33	1.00	8.00	1.00	0.33	1.00	7.00↓
NativeQuery Action	2.00	0.67	2.00	3.00	1.00↓	1.00↑	2.00	3.00
UpdateStore Action	3.00	0.67	0	4.00	1.00↓	1.00↓	0	1.00
DummyAction	0	0.67	0	2.00	0	0.33↓	0	2.00
QueryBy ExampleAction	1.00	0.67	2.00	9.00	1.00	0.67	2.00	8.00 ↓
WorkIn ProgressForm	0	0.33	0	6.00	1.00	0.33	0	5.00 ↓
Author	5.00	0.33	5.00	3.00	2.00↓	0.67↑	4.00↓	2.00 ↓
BookQuery Control	1.00	0.33	3.00	5.00	0 ↓	0.33	2.00↓	5.00
Database Manager	1.00	0.67	2.00	7.00	2.00↑	0.67	2.00	6.00↓
File	1.00	0.67	0	3.00	1.00	0.33↓	0	2.00↓

ตารางที่ 5.12 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Book Store (ต่อ)

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
RenameItem Form	3.00	1.00	1.00	6.00	2.00↓	1.00	2.00↑	3.00↓
YearOf Publishing	5.00	1.00	1.00	6.00	1.00↓	1.00	0 ↓	2.00↓
Tag	2.00	0.67	2.00	3.00	3.00↑	1.00↑	2.00	3.00
Publishing House	1.00	0.33	1.00	6.00	1.00	0.33	1.00	5.00↓
YearsOfPublishingQueryDataCategory	3.00	0.67	2.00	3.00	2.00↓	1.00 ↑	1.00↓	2.00↓
AuthorsQuery DataCategory	0	0.67	2.00	7.00	0	0.33 ↓	0 ↓	5.00↓
PublishingHousesQueryData Category	1.00	0.67	2.00	8.00	1.00	0.67	2.00	8.00
TagsQuery DataCategory	0	0	0	6.00	2.00↑	0.33 ↑	0	5.00↓
SetOperations	1.00	0.33	0	3.00	0 ↓	0.33	1.00↑	2.00↓
BookSelect Form	1.00	0.33	1.00	4.00	0 ↓	0.73 ↑	2.00↑	4.00
BooksList Operations	0	0.33	2.00	7.00	1.00↑	0.67 ↑	2.00	6.00↓
Folder	0	0.67	0	3.00	0	1.00↓	0	2.00↓
Background Task	1.00	0.33	0	5.00	0 ↓	0.67 ↑	1.00↑	4.00↓

ตารางที่ 5.12 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Book Store (ต่อ)

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
Mean	1.902	0.545	1.268	4.902	1.122↓	0.653↑	1.024↓	3.951↓
S.D.	3.105	0.222	1.225	1.972	1.249↓	0.280↑	1.012↓	1.940↓
Max	19.00	1.00	5.00	9.00	7.00	1.00	4.00	8.00
Min	0	0	0	2.00	0	0.33	0	1.00

จากระบบ Book Store ค่าของมาตรวัดที่ประเมินได้ เมื่อนำระบบมาคำนวณด้วยมาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันก่อนรีแฟคทอริงและหลังรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสมาเปรียบเทียบกัน ดังแสดงในตารางที่ 5.12 สามารถปรับปรุงรหัสต้นทางให้มีคุณภาพที่ดีขึ้นได้โดยพิจาณาจากมาตรวัด เมื่อคิดเป็นร้อยละของมาตรวัดจากทั้งระบบโดยนำค่า Mean ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดยมาตรวัด LCOM ลดลงร้อยละ 78.04 TCC เพิ่มขึ้นร้อยละ 10.82 RFC ลดลงร้อยละ 24.39 CBO ลดลงร้อยละ 95.12

5.3.4 ระบบ Ordering Administration System

การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ก่อนและหลังการทำรีแฟคทอริงของระบบ Ordering Administration System ดังแสดงในตารางที่ 5.13

ตารางที่ 5.13 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Ordering Administration System

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
Administrator	1.00	0.33	0	3.00	0 ↓	0.33	0	2.00 ↓
SearchFacade	1.00	0.33	1.00	5.00	0 ↓	0.33	2.00↑	5.00
UserManager	0	0.67	2.00	7.00	2.00↑	0.67	2.00	6.00↓
Customer	0	0.67	0	3.00	0	0.33↓	0	2.00↓

ตารางที่ 5.13 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Ordering Administration System (ต่อ)

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
Log	0	1.00	1.00	6.00	0	1.00	0 ↓	4.00↓
KeywordSet	0	1.00	1.00	6.00	0	1.00	0 ↓	4.00↓
Genre	1.00	0.33	1.00	8.00	1.00	0.33	1.00	7.00↓
ShippingInfo	2.00	0.67	2.00	3.00	1.00↓	1.00↑	2.00	3.00
Order	3.00	0.67	0	4.00	0 ↓	1.00↑	0	1.00↓
OrderDetail	0	0.67	2.00	11.00	0	0.33↓	0 ↓	8.00↓
Product	1.00	0.67	2.00	8.00	1.00	0.67	2.00	8.00
ShoppingCart	0	0.33	0	6.00	2.00↑	0.33	0	5.00↓
Mean	0.750	0.610	1.000	5.583	0.583↓	0.612↑	0.750↓	4.833↓
Std.dev.	0.965	0.241	0.965	2.443	0.793↓	0.314↑	0.853↓	2.353↓
Max	3.00	1.00	2.00	11.00	2.00	1.00	2.00	8.00
Min	0	0.33	0	3.00	0	0.33	0	1.00

จากระบบ Ordering Administration System ค่าของมาตรวัดที่ประเมินได้ เมื่อนำระบบมาคำนวณด้วยมาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันก่อนรีแฟคทอริงและหลังรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสมาเปรียบเทียบกัน ดังแสดงในตารางที่ 5.13 สามารถปรับปรุงรหัสต้นทางให้มีคุณภาพที่ดีขึ้นได้โดยพิจาณาจากมาตรวัด เมื่อคิดเป็นร้อยละของมาตรวัดจากทั้งระบบโดยนำค่า Mean ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดยมาตรวัด LCOM ลดลงร้อยละ 16.67 TCC ลดลงร้อยละ 0.17 RFC ลดลงร้อยละ 25.00 CBO ลดลงร้อยละ 75.00

5.3.5 ระบบ Graph of Function

การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ก่อนและหลังการทำรีแฟคทอริงของระบบ Graph of Function ดังแสดงในตารางที่ 5.14

ตารางที่ 5.14 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Graph of Function

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
FormFunction	2.00	0.33	3.00	3.00	0 ↓	0.33	1.00↓	2.00↓
Neg	1.00	0.67	1.00	5.00	1.00	0.33↓	2.00↑	5.00
MyChar	8.00	0.57	5.00	7.00	2.00↓	0.67↑	2.00↓	6.00↓
FromMy	0	0.67	0	3.00	0	0.33↓	0	2.00↓
FunctionColor	2.00	1.00	1.00	6.00	2.00	1.00	0 ↓	4.00↓
SyntaxTree	0	1.00	7.00	6.00	0	1.00	3.00↓	4.00↓
Ctg	1.00	0.33	1.00	8.00	1.00	0.33	1.00	7.00↓
Log	2.00	0.67	2.00	3.00	1.00↓	1.00↑	2.00	3.00
Bracket Opening	3.00	0.33	0	2.00	0 ↓	1.00↑	0	1.00↓
BracketClosing	0	0.67	2.00	4.00	0	0.33↓	0 ↓	3.00↓
Mean	1.400	0.624	2.000	4.280	0.700↓	0.632↑	1.100↓	3.700↓
S.D.	2.378	0.248	2.251	2.003	0.823↓	0.333↑	1.101↓	1.889↓
Max	8.00	1.00	7.00	8.00	2.00	1.00	3.00	7.00
Min	0	0.33	0	2.00	0	0.33	0	1.00

จากระบบ Graph of Function ค่าของมาตรวัดที่ประเมินได้ เมื่อนำระบบมาคำนวณด้วยมาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันก่อนรีแฟคทอริงและหลังรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสมาเปรียบเทียบกัน ดังแสดงในตารางที่ 5.14 สามารถปรับปรุงรหัสต้นทางให้มีคุณภาพที่ดีขึ้นได้โดยพิจาณาจากมาตรวัด เมื่อคิดเป็นร้อยละของมาตรวัดจากทั้งระบบโดยนำค่า Mean ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดยมาตรวัด LCOM ลดลงร้อยละ 70.00 TCC เพิ่มขึ้นร้อยละ 0.80 RFC ลดลงร้อยละ 90.00 CBO ลดลงร้อยละ 58.00

5.3.6 ระบบ Image Fan

การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ก่อนและหลังการทำรีแฟคทอริงของระบบ Image Fan ดังแสดงในตารางที่ 5.15

ตารางที่ 5.15 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Image Fan

ชื่อคลาส	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
Thumbnails Sequence	12.00	0.33	3.00	7.00	5.00↓	0.67↑	1.00↓	3.00↓
ThumbnailBox	5.00	0	1.00	5.00	2.00↓	0.33↓	2.00↑	4.00↓
ImageFolder	1.00	0.67	2.00	7.00	1.00	0.67	2.00	6.00↓
FolderTree View	1.00	0.67	1.00	3.00	0 ↓	0.33↓	1.00	2.00↓
ImageForm	0	1.00	1.00	6.00	0	1.00	0↓	4.00↓
FullScreen Image	0	1.00	2.00	5.00	0	1.00	1.00↓	5.00
ImageFile	1.00	0.33	1.00	8.00	1.00	0.33	1.00	6.00↓
Tasks Dispatcher	2.00	0.67	2.00	3.00	1.00↓	1.00↑	2.00	3.00
MainForm	3.00	0.67	0	4.00	0 ↓	1.00↑	0	1.00↓
ImageResizer	0	0.67	2.00	9.00	0	0.33↓	0↓	7.00↓
Mean	2.500	0.601	1.500	5.400	1.900↓	0.666↑	1.000↓	4.600↓
S.D.	3.689	0.307	0.850	2.058	1.564↓	0.316↑	0.817↓	1.912↓
Max	12.00	1.00	3.00	9.00	5.00	1.00	2.00	7.00
Min	0	0	1.00	3.00	0	0.33	0	1.00

จากระบบ Image Fan ค่าของมาตรวัดที่ประเมินได้ เมื่อนำระบบมาคำนวณด้วยมาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันก่อนรีแฟคทอริงและหลังรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสมาเปรียบเทียบกัน ดังแสดงในตารางที่ 5.15 สามารถปรับปรุงรหัสต้นทางให้มีคุณภาพที่ดีขึ้นได้โดยพิจาณาจากมาตรวัด เมื่อคิดเป็นร้อยละของมาตรวัดจากทั้งระบบโดยนำค่า Mean ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดย

มาตรวัด LCOM ลดลงร้อยละ 24.00 TCC เพิ่มขึ้นร้อยละ 10.82 RFC ลดลงร้อยละ 50.00 CBO ลดลงร้อยละ 80.00

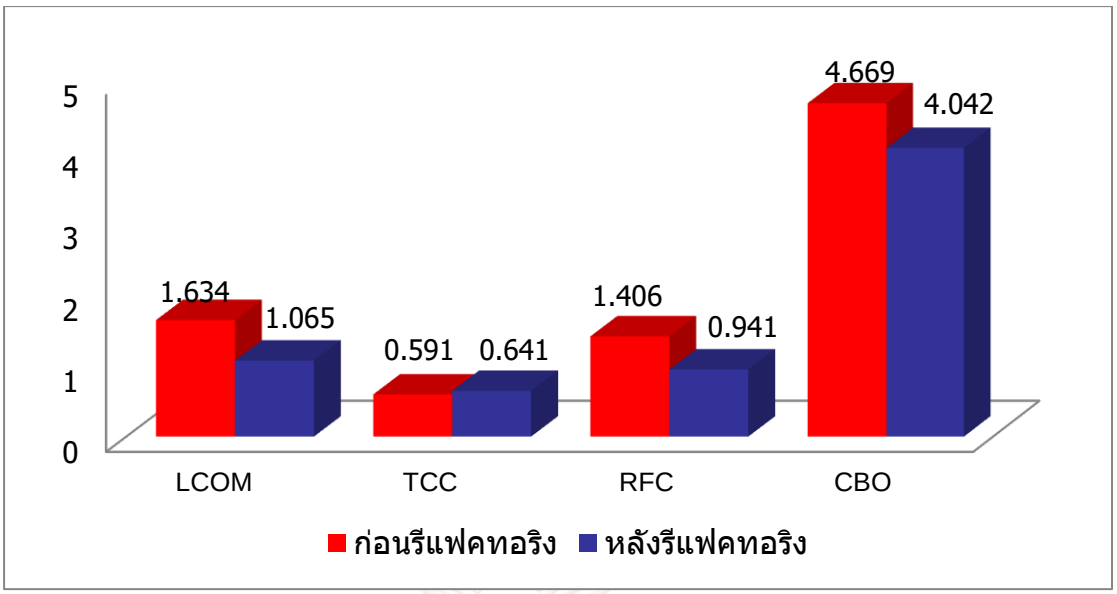
5.3.7 การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ทุกระบบ

การประเมินค่ามาตรวัดการเกาะกลุ่มกันการเข้าคู่ ก่อนและหลังการทำรีแฟคทอริงจากทุกระบบ โดยนำค่า Mean ในแต่ละระบบมาคำนวณ ดังแสดงในตารางที่ 5.16

ตารางที่ 5.16 ค่ามาตรวัดก่อนและหลังทำการรีแฟคทอริงของระบบ Graph of Function

ชื่อระบบ	ก่อนรีแฟคทอริง				หลังรีแฟคทอริง			
	LCOM	TCC	RFC	CBO	LCOM	TCC	RFC	CBO
Repairs Management	1.750	0.612	1.583	3.577	1.083↓	0.683↑	1.000↓	3.167↓
Restaurant Billing	1.500	0.551	1.080	4.269	1.000↓	0.601↑	0.769↓	4.000↓
Book Store	1.902	0.545	1.268	4.902	1.122↓	0.653↑	1.024↓	3.951↓
Ordering Administration	0.750	0.610	1.000	5.583	0.583↓	0.612↑	0.750↓	4.833↓
Graph of Function	1.400	0.624	2.000	4.280	0.700↓	0.632↑	1.100↓	3.700↓
Image Fan	2.500	0.601	1.500	5.400	1.900↓	0.666↑	1.000↓	4.600↓
Mean	1.634	0.591	1.406	4.669	1.065↓	0.641↑	0.941↓	4.042↓
S.D.	0.581	0.034	0.370	0.765	0.463↓	0.032↑	0.145↓	0.605↓

ค่าของมาตรวัดที่ประเมินได้จากทุกระบบที่นำมาทดสอบ เมื่อนำมาคำนวณด้วยมาตรวัดการเข้าคู่กันและการเกาะกลุ่มกันก่อนรีแฟคทอริงและหลังรีแฟคทอริงแบบมูฟเมท็อดโดยใช้ทฤษฎีการตัดสินใจในการระบุโอกาสมาเปรียบเทียบกัน ดังที่แสดงในตารางที่ 5.16 สามารถปรับปรุงรหัสต้นทางให้มีคุณภาพที่ดีขึ้นได้โดยพิจาณาจากมาตรวัด เมื่อคิดเป็นร้อยละของมาตรวัดจากทั้งระบบโดยนำค่า Mean ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดยมาตรวัด LCOM ลดลงร้อยละ 34.82 TCC เพิ่มขึ้นร้อยละ 8.46 RFC ลดลงร้อยละ 33.07 CBO ลดลงร้อยละ 13.43 ดังแสดงในภาพที่ 5.1



ภาพที่ 5.1 แผนภาพกราฟค่า Mean ที่ได้จากค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กันก่อนและหลังรีแฟคทอริง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 6
บทสรุปการวิจัย

ในบทนี้จะกล่าวถึงบทสรุป ข้อจำกัดและข้อเสนอแนะต่างๆ ที่ได้จากงานวิจัย การออกแบบวิธีการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงโดยใช้ทฤษฎีการตัดสินใจ ซึ่งมีดังรายละเอียดต่อไปนี้

6.1 บทสรุป

ในงานวิจัยนี้ได้มีการออกแบบวิธีการระบุโอกาสสำหรับการมูฟเมท็อดรีแฟคทอริงโดยใช้วิธีลาปลาสของทฤษฎีการตัดสินใจ เป็นเกณฑ์ในการตัดสินใจสำหรับการระบุโอกาสการย้ายเมท็อดเป้าหมายไปยังคลาสที่เหมาะสม งานวิจัยนี้ได้ประเมินกับระบบที่พัฒนาด้วยภาษาซีชาร์ป โดยงานวิจัยนี้สามารถระบุโอกาสการมูฟเมท็อด ซึ่งทำให้ผู้ใช้ทราบว่าจะย้ายเมท็อดเป้าหมายไปที่คลาสใดที่เหมาะสม ที่จะทำให้การเข้าคู่กันระหว่างคลาสน้อยลงและเพิ่มการเกาะกลุ่มกัน โดยการพิจารณาการเรียกใช้เมท็อดเป้าหมาย เพื่อช่วยทำให้นักออกแบบและนักพัฒนาซอฟต์แวร์ลดเวลาในการพิจารณาเมท็อดที่จะย้าย

งานวิจัยนี้พัฒนาเครื่องมือสำหรับการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงโดยใช้ทฤษฎีการตัดสินใจ ซึ่งขั้นตอนการระบุโอกาสในเครื่องมือนี้ คือ การเลือกระบบที่ต้องการปรับปรุง จากนั้นกำหนดเมท็อดเป้าหมาย แล้วหาว่าคลาสใดมีการเรียกใช้เมท็อดเมท็อดเป้าหมายบ้าง จากนั้นพิจารณาว่าจะย้ายเมท็อดเป้าหมายไปที่คลาสใดที่เหมาะสม โดยจะวัดการเข้าคู่กันระหว่างคลาสที่เมท็อดเป้าหมายอยู่และคลาสที่เรียกใช้เมท็อดนั้น ด้วยมาตรวัด CIM, RMC, ICBC, IMC แล้วจึงตัดสินใจเลือกคลาสที่เหมาะสมที่จะแนะนำให้เมท็อดนั้นย้ายไป

งานวิจัยนี้ได้ทำการประเมินกับระบบที่พัฒนาด้วยภาษาซีชาร์ป 6 ระบบ โดยระบบที่นำมาทดสอบจะวัดด้วยมาตรวัด LCOM, TCC, RFC, CBO ก่อนและหลังรีแฟคทอริงในทุกระบบ โดยนำค่า Mean ก่อนและหลังรีแฟคทอริงมาคำนวณร้อยละของการเพิ่มขึ้นหรือลดลงของมาตรวัด โดยมาตรวัด LCOM ลดลงร้อยละ 34.82 TCC เพิ่มขึ้นร้อยละ 8.46 RFC ลดลงร้อยละ 33.07 CBO ลดลงร้อยละ 13.43

6.2 ข้อจำกัด

6.2.1 ในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงด้วยวิธีที่นำเสนอได้มีการทดสอบกับระบบขนาดเล็กที่มีจำนวนคลาสสูงสุด 65 คลาส หากนำไปใช้ในการระบุโอกาสการย้ายเมท็อดกับระบบซอฟต์แวร์ขนาดใหญ่อาจทำให้มีผลกระทบจากการปรับปรุงโครงสร้างภายใน ที่ได้รับจากการเสนอแนะเมท็อดในการย้ายไปยังคลาสที่ระบุได้

6.2.2 เครื่องมือที่พัฒนาขึ้นยังมีข้อจำกัดและใช้ทดสอบกับระบบที่กำหนดเท่านั้น หากนำไปใช้ทดสอบกับระบบอื่นอาจมีผลกระทบที่ได้รับจากการปรับปรุงโครงสร้างภายใน ที่ได้รับจากการเสนอแนะเมท็อดในการย้ายไปยังคลาสที่ระบุได้

6.3 ข้อเสนอแนะ

6.3.1 เพิ่มมาตรวัดที่ใช้ในการตัดสินใจในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง เพื่อศึกษาว่ามาตรวัดมีผลให้การตัดสินใจในการระบุโอกาสเปลี่ยนแปลงไปหรือไม่

6.3.2 เพิ่มเกณฑ์การตัดสินใจในสถานการณ์อื่น สำหรับการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง เพื่อศึกษาถึงผลลัพธ์จากการระบุโอกาสในเกณฑ์ที่แตกต่างกัน

6.3.3 สร้างรูปแบบการตัดสินใจการรีแฟคทอริงวิธีอื่น เช่น Extract Class, Push Up, Push Down เพื่อศึกษาวิธีการระบุโอกาสการรีแฟคทอริง

6.3.4 เพิ่มการเปรียบเทียบค่ามาตรวัด IMC จากค่า 0.70 เป็นค่าอื่น เช่น 0.5, 0.6, 0.8, 0.9 เพื่อศึกษาถึงผลลัพธ์จากการระบุโอกาสที่แตกต่างกัน ในกรณีเมท็อดเป้าหมายมีการเรียกใช้ภายในคลาสและเมท็อดเป้าหมายมีการเรียกใช้ภายนอกคลาส

6.3.5 เพิ่มการจัดลำดับการย้ายเมท็อด เพื่อศึกษาถึงผลลัพธ์จากการจัดลำดับมีผลทำให้ค่าการเกาะกลุ่มกันเพิ่มขึ้นและค่าการเข้าคู่กันลดลงหรือไม่

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รายการอ้างอิง

1. Fowler, M., *Refactoring: improving the design of existing code.* 1999: Addison-Wesley.
2. Mika Mäntylä, Jari Vanhanen, and Casper Lassenius, *A Taxonomy and an Initial Empirical Study of Bad Smells in Code.* Software Maintenance, 2003. ICSM 2003. Proceedings. International Conference on, 2003: p. 381 - 384.
3. Alan Dennis, Barbara Haley Wixom, and David Tegarden, *Systems Analysis and Design: An Object-Oriented Approach with UML*. 2002: John Wiley & Sons, Inc.,.
4. DeGroot, M.H., *Optimal Statistical Decisions*. WCL Edition ed. 2004, New York: McGraw-Hill Book Co.
5. วันชัย ริจิวนิช และสุทัศน์ รัตนเกื้อกังวาน, วิธีการเชิงปริมาณเพื่อการตัดสินใจทางธุรกิจ: กรณีศึกษา. 2540: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย. 1-57.
6. R.G., C., *Decision Analysis* 1972, London: Yhomas and Sons Ltd., .
7. รศ.ทวี รื่นจินดา, ทฤษฎีตัดสินใจทางสถิติ. พิมพ์ครั้งที่ 6. 2547: สำนักพิมพ์มหาวิทยาลัยรามคำแหง.
8. Gabriele Bavota, et al., *A Two-Step Technique for Extract Class Refactoring.* ASE '10 Proceedings of the IEEE/ACM international conference on Automated software engineering, 2010: p. 151-154
9. Joshi, P., *Microscopic Relative Coupling Metrics.* 2006.
10. Amandeep Kaur, et al., *Empirical Analysis of CK & MOOD Metric Suit.* International Journal of Innovation, Management and Technology, 2010. 1(5).
11. S. R. Chidamber and C. F. Kemerer, *A metrics suite for object oriented design.* IEEE Transactions on Software Engineering, 1994. 20(6): p. 476–493.
12. Lionel C. Briand, John W. Daly, and Jürgen Wüst, *A Unified Framework for Cohesion Measurement in Object-Oriented Systems.* Empirical Software Engineering, 1998. 3(1): p. 65-117

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

13. Martin Hitz and Behzad Montazeri, *Measuring Coupling and Cohesion In Object-Oriented Systems.* Proceedings of International Symposium on Applied Corporate Computing, 1995: p. 1-10.
14. James M. Bieman and Byung-Kyoo Kang, *Cohesion and Reuse in Object-Oriented Systems.* SSR '95 Proceedings of the 1995 Symposium on Software reusability, 1995. 20: p. 259-262.
15. รศ.ดร.ญาใจ ลิ่มปิยะกรณ์, เอกสารคำสอน : การทำเหมืองข้อมูล. 2556, ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย. 41-43.
16. Giuseppe Pappalardo and Emiliano Tramontana, *Suggesting Extract Class Refactoring Opportunities by Measuring Strength of Method Interactions.* Software Engineering Conference (APSEC, 2013 20th Asia-Pacific), 2013. 2: p. 105-110.
17. Shuhei Kimura, et al., *Move Code Refactoring with Dynamic Analysis.* Software Maintenance (ICSM), 2012 28th IEEE International Conference on, 2012: p. 575 - 578.
18. Gabriele Bavota, et al., *Playing with Refactoring Identifying Extract Class Opportunities through Game Theory.* Software Maintenance (ICSM), 2010 IEEE International Conference on, 2010: p. 1-5.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

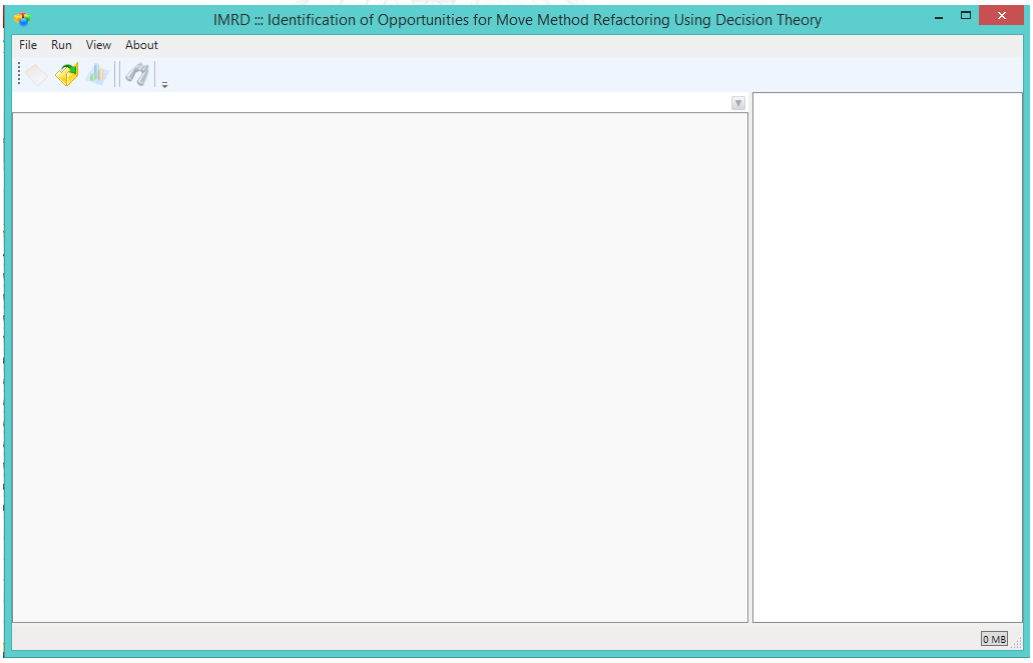
ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

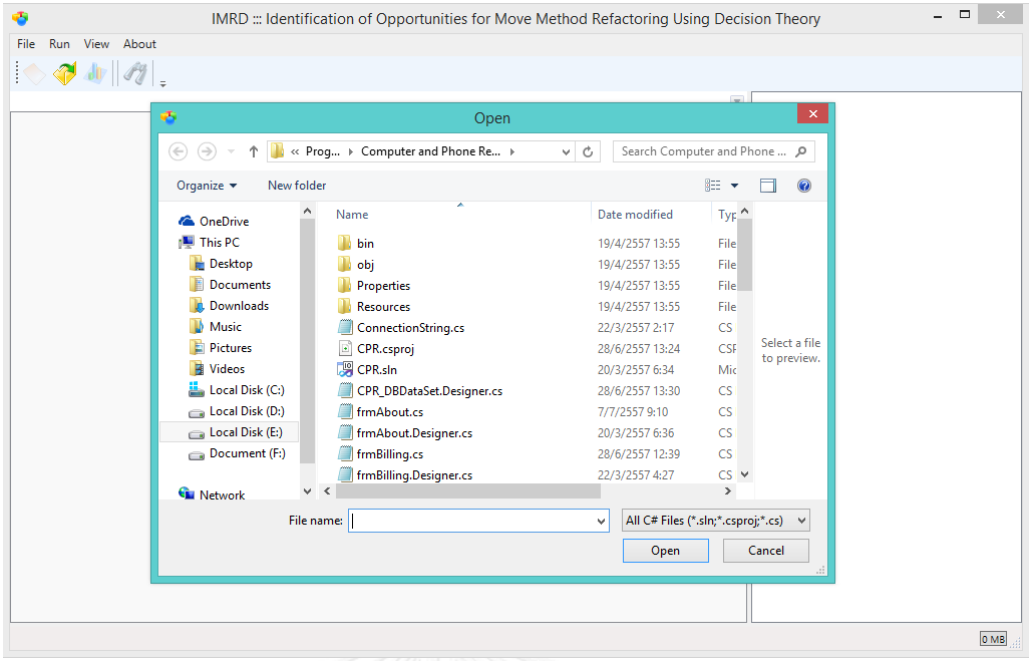
ภาคผนวก ก
คู่มือการใช้งาน IMRD Tool

ในบทนี้จะอธิบายการใช้งานเครื่องมือ IMRD Tool เป็นเครื่องมือที่ใช้สำหรับการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงด้วยทฤษฎีการตัดสินใจ โดยในการระบุโอกาสจะต้องมีการนำเข้าของรหัสต้นทางที่ต้องการระบุโอกาสการมูฟเมท็อดรีแฟคทอริง ซึ่งจะมีรายละเอียดดังนี้

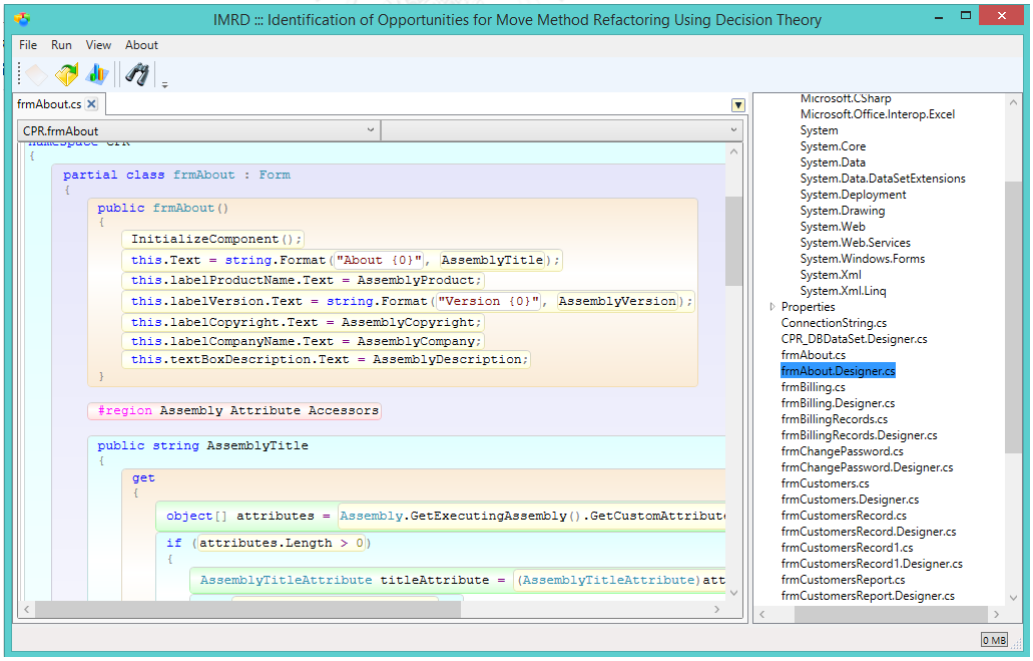
1) ผู้ใช้งานกำหนดรหัสต้นทางที่ต้องการนำมาใช้ในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงโดยใช้เมนู File>Open หรือคลิกปุ่ม บนแถบเครื่องมือ ดังแสดงในภาพที่ ก-1 จากนั้นเครื่องมือ IMRD จะแสดงหน้าต่างให้เลือกรหัสต้นทางที่ต้องการ ดังแสดงในภาพที่ ก-2 เมื่อเลือกรหัสต้นทางที่ต้องการแล้ว คลิกปุ่ม Open จากนั้นไฟล์ของรหัสต้นทางที่เลือกจะแสดงบนหน้าจอ ดังแสดงในภาพที่ ก-3



ภาพที่ ก-1 หน้าจอเมื่อผู้ใช้งานเข้าสู่ระบบ

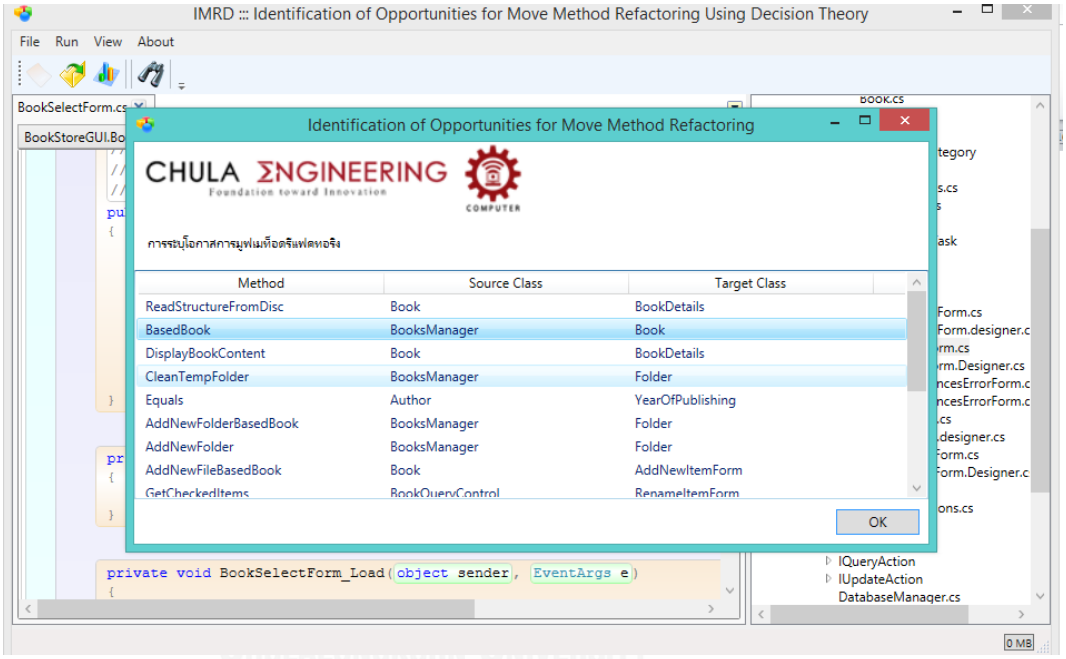


ภาพที่ ก-2 หน้าจอเมื่อผู้ใช้งานเลือกรหัสต้นทาง



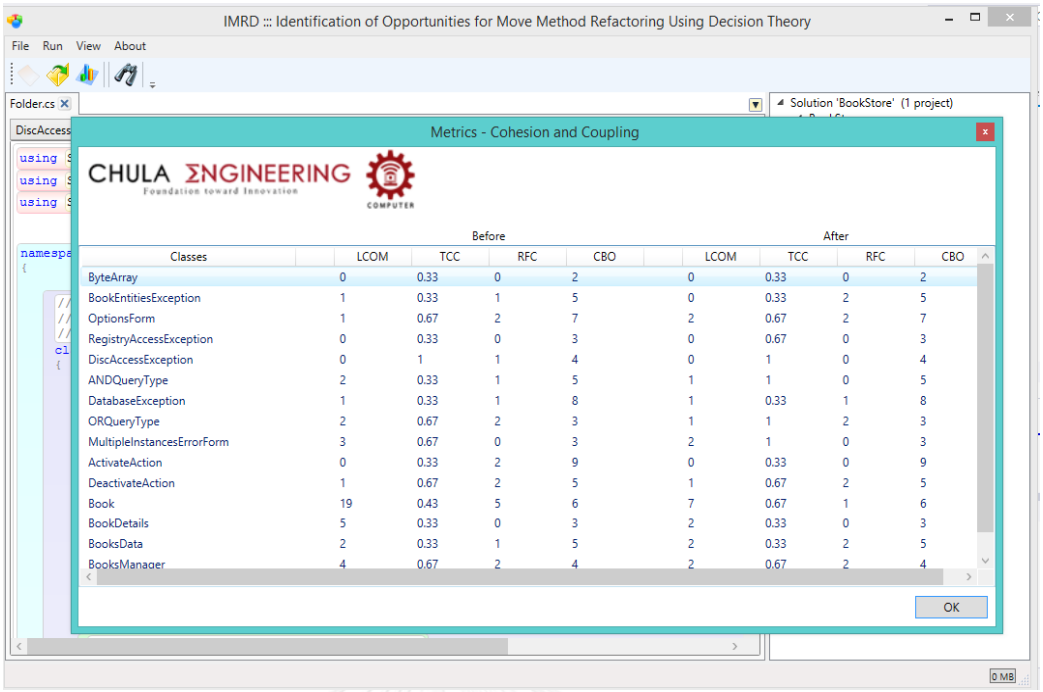
ภาพที่ ก-3 หน้าจอแสดงรหัสต้นทางที่ผู้ใช้งานเลือก

2) ผู้ใช้งานทำการระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อด ได้โดยเลือกเมนู Run จากนั้นโปรแกรมจะทำการตรวจจับเมท็อดเป้าหมายที่มีการเรียกใช้จากภายนอกคลาส และตรวจจับว่ามีคลาสใดที่เรียกใช้เมท็อดเป้าหมายบ้าง เพื่อใช้เป็นทางเลือกของเมท็อดเป้าหมายว่าจะมูฟไปที่คลาสใด จากนั้นวัดค่าการเข้าคู่กันในแต่ละทางเลือก และใช้ทฤษฎีการตัดสินใจในการเลือกคำตอบว่าจะมูฟเมท็อดเป้าหมายไปที่คลาสใด ซึ่งจะเป็นคลาสเป้าหมาย เมื่อทำงานเสร็จเรียบร้อยแล้วโปรแกรมจะแสดงการระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อด โดยจะระบุเมท็อดเป้าหมาย คลาสต้นทาง คลาสเป้าหมาย ดังแสดงในภาพที่ ก-4



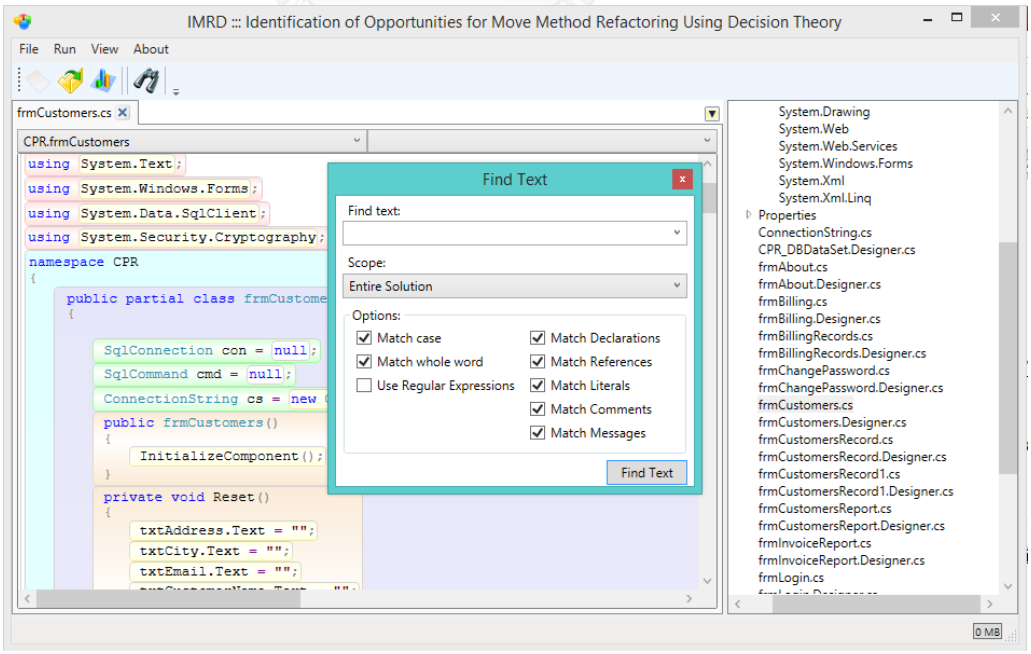
ภาพที่ ก-4 หน้าจอแสดงการระบุโอกาสการมูฟเมท็อด

3) ผู้ใช้งานสามารถดูค่ามาตรวัด LCOM, TCC, RFC, CBO ซึ่งเป็นมาตรวัดการเกาะกลุ่มกันและการเข้าคู่กัน โดยจะเป็นมาตรวัดที่ใช้วัดรหัสต้นทาง ในแต่ละคลาสของระบบที่นำมาทดสอบ โดยเลือกเมนู View>Metrics หรือคลิกปุ่ม บนแถบเครื่องมือ ดังแสดงในภาพที่ ก-5 จากนั้นโปรแกรมจะคำนวณและแสดงค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กัน



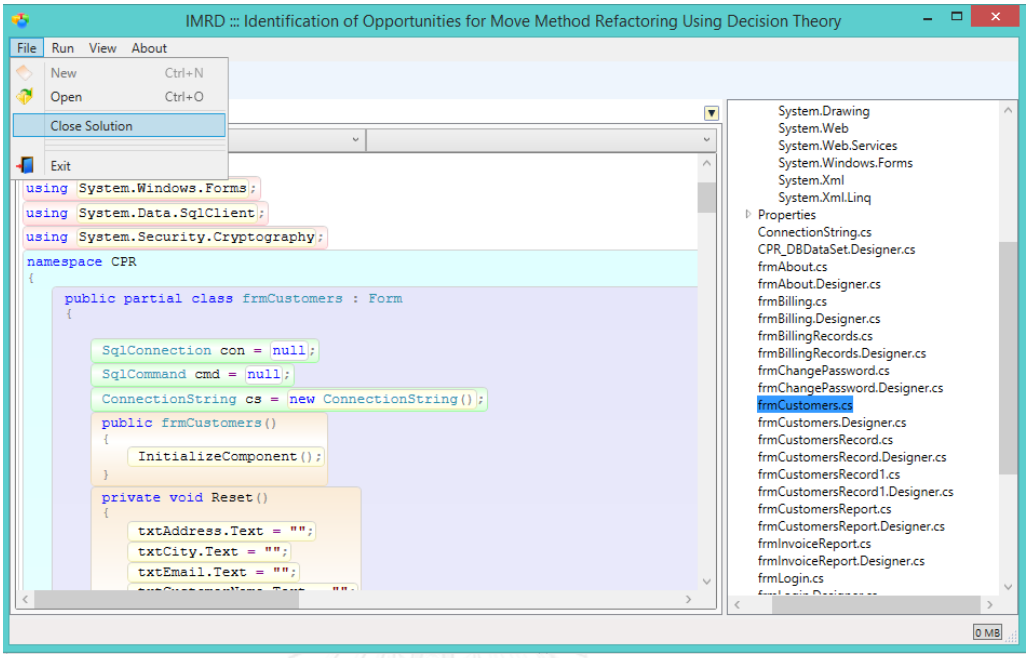
ภาพที่ ก-5 หน้าจอแสดงค่ามาตรวัดการเกาะกลุ่มกันและการเข้าคู่กัน

4) ผู้ใช้งานสามารถค้นหาชื่อตัวแปร เมท็อด และ คลาส ในรหัสต้นทางที่ผู้ใช้งานต้องการนำมาใช้ในการระบุโอกาสการมูฟเมท็อดรีแฟคทอริงโดยใช้เมนู View>Find หรือคลิกปุ่ม บนแถบเครื่องมือ ดังแสดงในภาพที่ ก-6



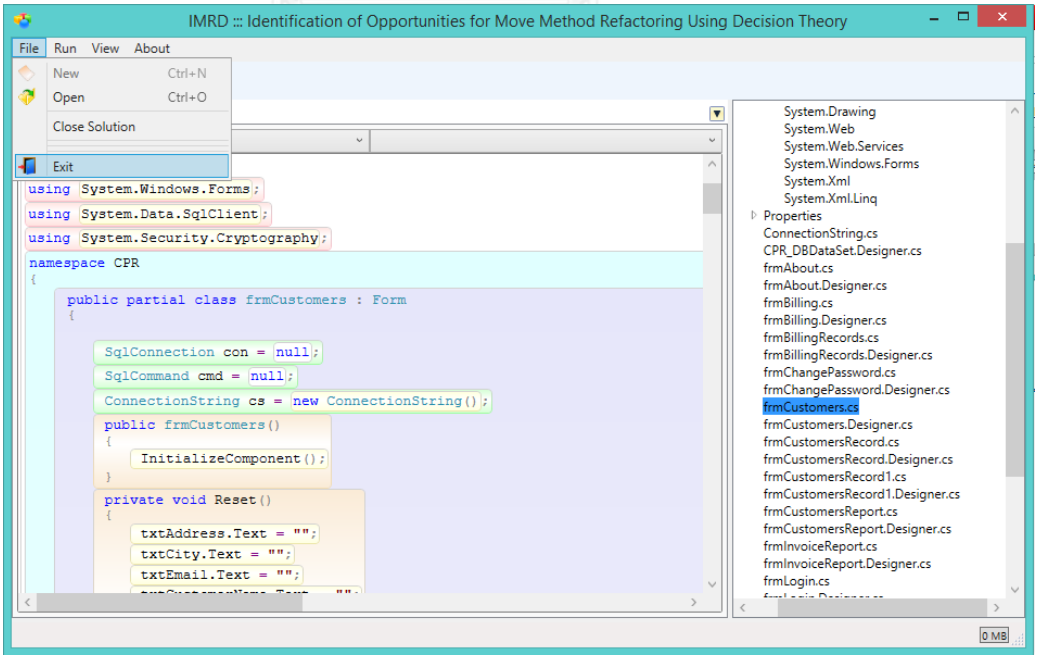
ภาพที่ ก-6 หน้าจอแสดงหน้าต่างค้นหาชื่อตัวแปร เมท็อด และ คลาส

5) ผู้ใช้งานสามารถยกเลิกรหัสต้นทางชุดเดิมเพื่อกำหนดรหัสต้นทางชุดใหม่ได้โดยเลือกเมนู File>Close Solution ดังแสดงในภาพที่ ก-7



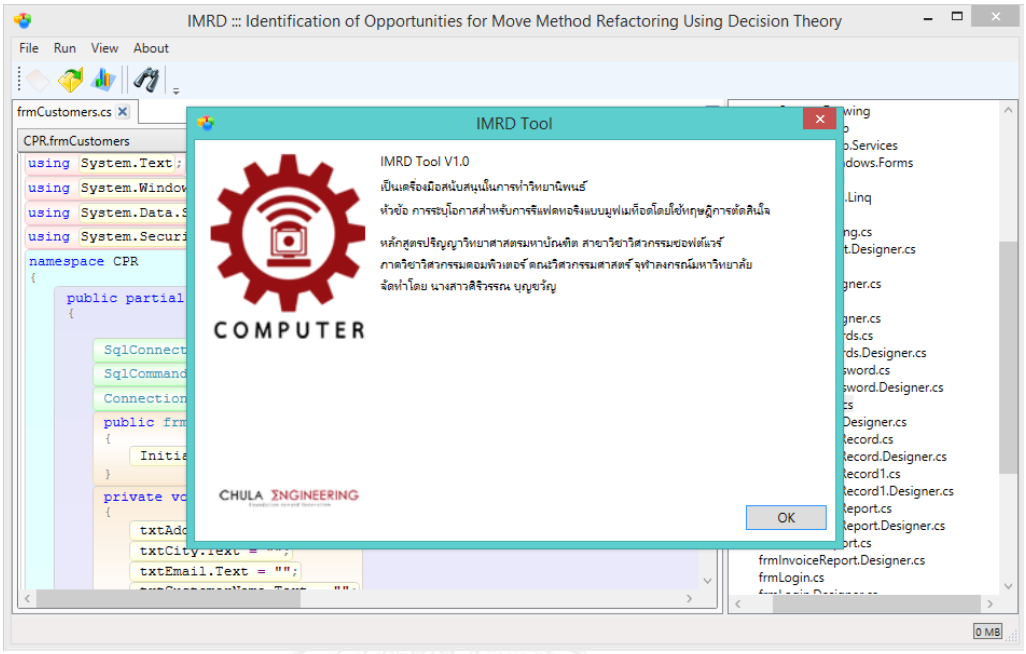
ภาพที่ ก-7 หน้าจอแสดงเมื่อผู้ใช้งานยกเลิกรหัสต้นทางชุดเดิม

6) ผู้ใช้งานสามารถออกจากโปรแกรม IMRD ได้โดยเลือกเมนู File>Exit บนแถบเครื่องมือหรือคลิกปุ่ม ดังแสดงในภาพที่ ก-8



ภาพที่ ก-8 หน้าจอแสดงเมื่อผู้ใช้งานต้องการออกจากโปรแกรม IMRD

7) ผู้ใช้งานสามารถดูรายละเอียดของการจัดทำเครื่องมือ IMRD ได้โดยใช้เมนู About บนแถบเครื่องมือ ดังแสดงในภาพที่ ก-9



ภาพที่ ก-9 หน้าจอแสดงรายละเอียดของการจัดทำเครื่องมือ IMRD

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ข
การระบุโอกาสการรีแฟคทอริงแบบมูฟเมท็อดด้วยวิธีของทฤษฎีการตัดสินใจกับระบบที่พัฒนาด้วยภาษาซีชาร์ป

ในภาคผนวก ข จะอธิบายการระบุโอกาสในแต่ละระบบ ซึ่งจะอธิบายการระบุโอกาสของระบบที่ไม่ได้อธิบายไว้ในบทที่ 4 คือ ระบบ Restaurant Billing System, ระบบ Book Store, ระบบ Ordering Administration System, ระบบ Graph of Function และระบบ Image Fan

ข-1 ระบบ Restaurant Billing System

การค้นหาคลาสเพื่อระบุโอกาสการย้ายเมท็อดเป้าหมาย ไปยังคลาสที่เหมาะสม ตามการเรียกใช้เมท็อดเป้าหมายจากภายในคลาสและภายนอกคลาส

1) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหนึ่งคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้สามารถเสนอแนะเป็นทางเลือกในการย้ายเมท็อดโดยไม่ต้องคำนวณค่ามาตรวัดและไม่ต้องใช้เกณฑ์ลาปลาสในการตัดสินใจ

ตารางที่ ข.1 การเรียกใช้เมท็อดภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Restaurant Billing System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
txtSaleQty_KeyPress	frmCategory	frmSales	0	2
Username_Validating	frmLoginDetails	frmRegistration	0	1
CheckAvailability	frmRegistration	frmLogin	0	1

2) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหลายคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ ข.2 การเรียกใช้เมท็อดภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Restaurant Billing System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
cmbCustomer Name_Format	frmSalesRecord	frmCustomers	0	2
cmbCustomer Name_Format	frmSalesRecord	frmLoginDetails	0	1
	frmSalesRecord	frmCustomersRecord	0	3
	frmSalesRecord	frmRegisteredUsersDetails	0	1

การคำนวณค่ามาตรวัด ในกรณีนี้คำนวณด้วยมาตรวัด CIM, RMC และ ICBC โดยไม่ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ ข.3 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับเมท็อด cmbCustomerName_Format

ทางเลือก ลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
1	0.650	0.460	0.210	0.440
2	0.370	0.010	0.150	0.177
3	0.430	0.030	0.017	0.159
4	0.270	0.210	0.150	0.210
5	0.330	0.020	0.020	0.370

การเลือกคลาสที่เหมาะสม โดยการเลือกด้วยวิธีลาปลาส จะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด คือทางเลือกที่ 1 มีค่าเท่ากับ 0.407 ดังนั้น เลือกย้ายเมท็ cmbCustomerName_Format จากคลาส frmSalesRecord ไปยังคลาส frmRegistration

3) การค้นหาทางเลือกจาก เมท็อดเป้าหมาย ที่มีการเรียกใช้ภายในคลาสและมีการเรียกใช้จากภายนอกคลาส

ตารางที่ ข.4 การเรียกใช้เมท็อดภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Restaurant Billing System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
GetData	frmRegisteredUsers Details	frmCustomers	1	2
GetData	frmRegisteredUsers Details	frmProductsRecord	1	1
	frmRegisteredUsers Details	frmCustomersRecord	1	5
	frmRegisteredUsers Details	frmLoginDetails	1	1
	frmRegisteredUsers Details	frmRegistration	1	3
	frmRegisteredUsers Details	frmProduct	1	2
	frmRegisteredUsers Details	frmCategoryRecord	1	2
FillCombo	frmProduct	frmRegistered UsersDetails	3	1
	frmProduct	frmLoginDetails	3	2
	frmProduct	frmRegistration	3	7
	frmProduct	frmSales	3	4
Calculate	frmSales	frmSalesRecord	4	3
	frmSales	frmCategory	4	1
	frmSales	frmCategoryRecordv	4	2
GetUniqueKey	frmCustomers	frmLoginDetails	2	1
	frmCustomers	frmRegistration	2	3
	frmCustomers	frmCategoryRecord	2	1
	frmCustomers	frmSalesRecord	2	1
	frmCustomers	frmSales	2	4

ตารางที่ ข.4 การเรียกใช้เมท็อดภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Restaurant Billing System (ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
Reset	frmCustomers	frmCustomersRecord	1	3
	frmCustomers	frmSalesRecord	1	1

การคำนวณค่ามาตรวัด ในกรณีนี้ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะมีการเรียกใช้เมท็อดเป้าหมายภายในคลาส จากนั้นคำนวณด้วยมาตรวัด CIM, RMC และ ICBC

โดยแบ่งเป็น 2 ขั้นตอน

ขั้นตอนที่ 1 คำนวณด้วยมาตรวัด IMC เพื่อจะคำนวณการเรียกใช้ภายในคลาส และการเรียกใช้จากภายนอกคลาส ของเมท็อดเป้าหมาย

ตารางที่ ข.5 การคำนวณด้วยมาตรวัด IMC ของระบบ Restaurant Billing System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
GetData	frmRegisteredUsersDetails	frmCustomers	0.670
	frmRegisteredUsersDetails	frmProductsRecord	0.500
	frmRegisteredUsersDetails	frmCustomersRecord	0.830
	frmRegisteredUsersDetails	frmLoginDetails	0.500
	frmRegisteredUsersDetails	frmRegistration	0.750
	frmRegisteredUsersDetails	frmProduct	0.670
	frmRegisteredUsersDetails	frmCategoryRecord	0.670
FillCombo	frmProduct	frmRegistered UsersDetails	0.250
	frmProduct	frmLoginDetails	0.400
	frmProduct	frmRegistration	0.700
	frmProduct	frmSales	0.570
Calculate	frmSales	frmSalesRecord	0.430
	frmSales	frmCategory	0.200
	frmSales	frmCategoryRecordv	0.330

ตารางที่ ข.5 การคำนวณด้วยมาตรวัด IMC ของระบบ Restaurant Billing System(ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
GetUniqueKey	frmCustomers	frmLoginDetails	0.330
	frmCustomers	frmRegistration	0.600
	frmCustomers	frmCategoryRecord	0.330
	frmCustomers	frmSalesRecord	0.330
	frmCustomers	frmSales	0.670
Reset	frmCustomers	frmCustomersRecord	0.750
	frmCustomers	frmSalesRecord	0.700

ขั้นตอนที่ 2 เลือกทางเลือกที่มีค่า IMC >= 0.70 มาคำนวณด้วยมาตรวัด CIM, RMC, ICBC ในแต่ละทางเลือกเพื่อพิจารณาจะย้ายเมท็อดไปที่คลาสใด

ตารางที่ ข.6 การตัดสินใจเลือกคลาสที่เหมาะสม สำหรับเมท็อดเป้าหมายของระบบ Restaurant Billing System

ทางเลือก ลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
เมท็อด GetData				
1	0.780	0.270	0.250	0.433
2	0.520	0.350	0.310	0.393
เมท็อด Reset				
1	0.670	0.280	0.150	0.367
2	0.480	0.310	0.200	0.330

การเลือกคลาสที่เหมาะสม โดยจะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด ในกรณีของเมท็อด GetData คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.433 และในกรณีของเมท็อด Reset คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.367 ดังนั้นเลือกย้ายเมท็อด GetData จากคลาส frmRegisteredUsersDetails ไปยังคลาส frmCustomersRecord และย้ายเมท็อด Reset จากคลาส frmCustomers ไปยังคลาส frmCustomers Record

การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม จากระบบ Restaurant Billing System สามารถเสนอแนะเมท็อดที่จะต้องย้าย ดังที่แสดงในตารางที่ ข.7

ตารางที่ ข.7 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Restaurant Billing System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง
txtSaleQty_KeyPress	frmCategory	frmSales
Username_Validating	frmLoginDetails	frmRegistration
CheckAvailability	frmRegistration	frmLogin
cmbCustomerName_Format	frmSalesRecord	frmRegistration
FillCombo	frmProduct	frmRegistration
GetData	frmRegisteredUsersDetails	frmCustomersRecord
Reset	frmCustomers	frmCustomersRecord

ข-2 ระบบ Book Store

การค้นหาคลาสเพื่อระบุโอกาสการย้ายเมท็อดเป้าหมาย ไปยังคลาสที่เหมาะสม ตามการเรียกใช้เมท็อดเป้าหมายจากภายในคลาสและภายนอกคลาส

1) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหนึ่งคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้สามารถเสนอแนะเป็นทางเลือกในการย้ายเมท็อดโดยไม่ต้องคำนวณค่ามาตรวัดและไม่ต้องใช้เกณฑ์ลาปลาสในการตัดสินใจ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ ข.8 การเรียกใช้เมท็อดภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Book Store

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
ReadStructureFromDisc	Book	BookDetails	0	1
BasedBook	BooksManager	Book	0	1
DisplayBookContent	Book	BooksData	0	1
CleanTempFolder	BooksManager	Folder	0	2

2) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหลายคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ ข.9 การเรียกใช้ภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Book Store

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
AddNewFolder BasedBook	BooksManager	BookQueryControl	0	1
	BooksManager	MainForm	0	2
	BooksManager	Folder	0	5
AddNewFolder	BooksManager	MainForm	0	1
	BooksManager	Folder	0	3
AddNewFileBased Book	BooksManager	Book	0	1
	BooksManager	AddNewItemForm	0	3
	BooksManager	MainForm	0	2
GetCheckedItems	BookQueryControl	AddNewItemForm	0	2
	BookQueryControl	RenameItemForm	0	5

การคำนวณค่ามาตรวัด ในกรณีนี้คำนวณด้วยมาตรวัด CIM, RMC และ ICBC โดยไม่ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ ข.10 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับเมท็อดเป้าหมายของระบบ Book Store

ทางเลือก ลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
เมท็อด AddNewFolderBasedBook				
1	0.580	0.020	0.010	0.203
2	0.430	0.010	0.150	0.197
3	0.750	0.460	0.210	0.473
เมท็อด AddNewFolder				
1	0.320	0.220	0.020	0.187
2	0.470	0.250	0.150	0.290
เมท็อด AddNewFileBasedBook				
1	0.330	0.021	0.010	0.120
2	0.720	0.350	0.200	0.423
3	0.370	0.110	0.150	0.210

ตารางที่ ข.10 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับเมท็อดเป้าหมายของระบบ Book Store (ต่อ)

ทางเลือกลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
เมท็อด GetCheckedItems				
1	0.370	0.010	0.150	0.177
2	0.550	0.460	0.210	0.407

การเลือกคลาสที่เหมาะสม โดยจะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด ในกรณีของเมท็อด AddNewFolderBasedBook คือ ทางเลือกที่ 3 มีค่าเท่ากับ 0.473 ในกรณีของเมท็อด AddNewFolder คือ ทางเลือกที่ 2 มีค่าเท่ากับ 0.290ในกรณีของเมท็อด AddNewFileBased-Book คือ ทางเลือกที่ 2 มีค่าเท่ากับ 0.423 และในกรณีของเมท็อด GetCheckedItems คือ ทางเลือกที่ 2 มีค่าเท่ากับ 0.407 ดังนั้นเลือกย้ายAddNewFolderBasedBook จากคลาส Books-Manager ไปยังคลาส Folder ย้าย เมท็อด AddNewFolder จากคลาส BooksManager ไปยังคลาส Folder ย้ายเมท็อด AddNewFileBasedBook จากคลาส BooksManager ไปยังคลาส Main-Form และย้ายเมท็อด GetCheckedItems จากคลาส BookQueryControl ไปยังคลาส Rename-ItemForm

3) การค้นหาทางเลือกจาก เมท็อดเป้าหมาย ที่มีการเรียกใช้ภายในคลาสและมีการเรียกใช้จากภายนอกคลาสของระบบ Book Store

ตารางที่ ข.11 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Book Store

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
PerformUpdate	ActivateAction	DeactivateAction	2	3
	ActivateAction	DeleteAction	2	2
	ActivateAction	UpdateStoreAction	2	6
	ActivateAction	DummyAction	2	1
	ActivateAction	QueryByExampleAction	2	1
CompareTo	Author	YearOfPublishing	3	1
	Author	PublishingHouse	3	4
	Author	Tag	3	2
	Author	Book	3	1

ตารางที่ ข.11 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Book Store (ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	#access Source	#access Target
Equals	Author	YearOfPublishing	2	5
	Author	PublishingHouse	2	2
	Author	Tag	2	3
ANDInternalQuery	YearsOfPublishing QueryDataCategory	AuthorsQueryData Category	5	10
ANDInternalQuery	YearsOfPublishing QueryDataCategory	PublishingHouses QueryDataCategory	5	9
	YearsOfPublishing QueryDataCategory	TagsQueryData Category	5	3
ChangeBookTitle	BooksListOperations	Book	2	2
	BooksListOperations	BookDetails	2	1
	BooksListOperations	BooksManager	2	5
	BooksListOperations	BooksData	2	3
GetCategorized Books	BooksData	BookDetails	1	1
	BooksData	BooksManager	1	4
	BooksData	MainForm	1	2
GetAllBooks	BooksData	BookDetails	3	1
	BooksData	Book	3	1
	BooksData	BooksManager	3	4
	BooksData	MainForm	3	12
	BooksData	BookSelectForm	3	7
	BooksData	BooksListOperations	3	9
	BooksData	BookQueryControl	3	8
ORInternalQuery	YearsOfPublishing QueryDataCategory	AuthorsQueryData Category	5	8
	YearsOfPublishing QueryDataCategory	PublishingHouses QueryDataCategory	5	11

ตารางที่ ข.11 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Book Store (ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	#access Source	#access Target
ORInternalQuery	YearsOfPublishing QueryDataCategory	TagsQueryData Category	5	6
	YearsOfPublishing QueryDataCategory	AuthorsQueryData Category	5	8
GetSoughtBooks	MainForm	BookSelectForm	3	8
	MainForm	SetOperations	3	1
	MainForm	BooksManager	3	4
	MainForm	BooksData	3	2
	MainForm	BooksListOperations	3	11
ResetBooksList	MainForm	BooksManager	1	4
	MainForm	RenameItemForm	1	2
	MainForm	SetOperations	1	1
	MainForm	BookSelectForm	1	1

การคำนวณค่ามาตรวัด ในกรณีนี้ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะมีการเรียกใช้เมท็อดเป้าหมายภายในคลาส จากนั้นคำนวณด้วยมาตรวัด CIM, RMC และ ICBC

โดยแบ่งเป็น 2 ขั้นตอน

ขั้นตอนที่ 1 คำนวณด้วยมาตรวัด IMC เพื่อจะคำนวณการเรียกใช้ภายในคลาส และการเรียกใช้จากภายนอกคลาส ของเมท็อดเป้าหมาย

ตารางที่ ข.12 การคำนวณด้วยมาตรวัด IMC ของระบบ Book Store

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
PerformUpdate	ActivateAction	DeactivateAction	0.760
	ActivateAction	DeleteAction	0.500
	ActivateAction	UpdateStoreAction	0.810
	ActivateAction	DummyAction	0.330
	ActivateAction	QueryByExampleAction	0.330

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ ข.12 การคำนวณด้วยมาตรวัด IMC ของระบบ Book Store (ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
ANDInternalQuery	YearsOfPublishing QueryDataCategory	AuthorsQueryData Category	0.670
	YearsOfPublishing QueryDataCategory	PublishingHouses QueryDataCategory	0.640
	YearsOfPublishing QueryDataCategory	TagsQueryData Category	0.380
GetCategorizedBooks	BooksData	BookDetails	0.500
	BooksData	BooksManager	0.800
	BooksData	MainForm	0.700
CompareTo	Author	YearOfPublishing	0.250
	Author	PublishingHouse	0.570
	Author	Tag	0.400
	Author	Book	0.250
Equals	Author	YearOfPublishing	0.710
	Author	PublishingHouse	0.500
	Author	Tag	0.600
ANDInternalQuery	YearsOfPublishing QueryDataCategory	AuthorsQueryData Category	0.670
	YearsOfPublishing QueryDataCategory	PublishingHouses QueryDataCategory	0.640
	YearsOfPublishing QueryDataCategory	TagsQueryData Category	0.380
GetCategorizedBooks	BooksData	BookDetails	0.500
	BooksData	BooksManager	0.800
	BooksData	MainForm	0.700

ตารางที่ ข.12 การคำนวณด้วยมาตรวัด IMC ของระบบ Book Store (ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
ChangeBookTitle	BooksListOperations	Book	0.500
	BooksListOperations	BookDetails	0.330
	BooksListOperations	BooksManager	0.790
	BooksListOperations	BooksData	0.700
GetAllBooks	BooksData	BookDetails	0.250
	BooksData	Book	0.250
	BooksData	BooksManager	0.570
	BooksData	MainForm	0.890
	BooksData	BookSelectForm	0.700
	BooksData	BooksListOperations	0.750
	BooksData	BookQueryControl	0.330
ORInternalQuery	YearsOfPublishing QueryDataCategory	AuthorsQueryData Category	0.620
	YearsOfPublishing QueryDataCategory	PublishingHouses QueryDataCategory	0.690
	YearsOfPublishing QueryDataCategory	TagsQueryData Category	0.550
	YearsOfPublishing QueryDataCategory	AuthorsQueryData Category	0.620
GetSoughtBooks	MainForm	BookSelectForm	0.730
	MainForm	SetOperations	0.250
	MainForm	BooksManager	0.570
	MainForm	BooksData	0.400
	MainForm	BooksListOperations	0.790
ResetBooksList	MainForm	BooksManager	0.800
	MainForm	RenameItemForm	0.670
	MainForm	SetOperations	0.750
	MainForm	BookSelectForm	0.530

ขั้นตอนที่ 2 เลือกทางเลือกที่มีค่า IMC >= 0.70 มาคำนวณด้วยมาตรวัด CIM, RMC, ICBC ในแต่ละทางเลือกเพื่อพิจารณาจะย้ายเมท็อดไปที่คลาสใด

ตารางที่ ข.13 การคำนวณมาตรวัดในกรณีมีการเรียกใช้ภายในคลาสของระบบ Book Store

ทางเลือกลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
เมท็อด PerformUpdate				
1	0.540	0.370	0.150	0.353
2	0.780	0.250	0.310	0.447
เมท็อด ChangeBookTitle				
1	0.671	0.380	0.240	0.430
2	0.380	0.420	0.250	0.350
เมท็อด GetCategorizedBooks				
1	0.650	0.420	0.240	0.437
2	0.350	0.510	0.150	0.337
3	0.470	0.250	0.210	0.310
เมท็อด GetAllBooks				
1	0.810	0.380	0.200	0.463
2	0.580	0.610	0.250	0.480
เมท็อด GetSoughtBooks				
1	0.480	0.250	0.310	0.347
2	0.510	0.470	0.150	0.377
เมท็อด ResetBooksList				
1	0.780	0.370	0.210	0.453
2	0.550	0.510	0.250	0.437

การเลือกคลาสที่เหมาะสม โดยจะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด ในกรณีของเมท็อด PerformUpdate คือ ทางเลือกที่ 2 มีค่าเท่ากับ 0.447 ในกรณีของเมท็อด ChangeBookTitle คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.430 ในกรณีของเมท็อด GetAllBooks คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.437 ในกรณีของเมท็อด GetCategorizedBooks คือ ทางเลือกที่ 2 มีค่าเท่ากับ 0.480 ในกรณีของเมท็อด GetSoughtBooks คือ ทางเลือกที่ 2 มีค่าเท่ากับ 0.377 และในกรณีของเมท็อด

ResetBooksList คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.453 ดังนั้นเลือกย้ายเมท็อด PerformUpdate จากคลาส ActivateAction ไปยังคลาส UpdateStoreAction ย้ายเมท็อด ChangeBookTitle จากคลาส BooksListOperations ไปยังคลาส BooksManager ย้ายเมท็อด GetAllBooks จากคลาส BooksData ไปยังคลาส BooksList- Operations ย้ายเมท็อด GetCategorizedBooks จากคลาส BooksData ไปยังคลาส BookDetails ย้ายเมท็อด GetSoughtBooks จากคลาส MainForm ไปยังคลาส BooksListOperations และย้ายเมท็อด ResetBooksList จากคลาส MainForm ไปยังคลาส BooksManager

การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม จากระบบ Book Store สามารถเสนอแนะเมท็อดที่จะต้องย้าย ดังที่แสดงในตารางที่ ข.14

ตารางที่ ข.14 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Book Store

เมท็อด	คลาสต้นทาง	คลาสปลายทาง
ReadStructureFromDisc	Book	BookDetails
BasedBook	BooksManager	Book
DisplayBookContent	Book	BooksData
CleanTempFolder	BooksManager	Folder
Equals	Author	YearOfPublishing
AddNewFolderBasedBook	BooksManager	Folder
AddNewFolder	BooksManager	Folder
AddNewFileBasedBook	BooksManager	AddNewItemForm
GetCheckedItems	BookQueryControl	RenameItemForm
PerformUpdate	ActivateAction	DeleteAction
ChangeBookTitle	BooksListOperations	Book
GetAllBooks	BooksData	BookDetails
GetCategorizedBooks	BooksData	BookDetails
GetSoughtBooks	MainForm	SetOperations
ResetBooksList	MainForm	BooksManager

ข-3 ระบบ Ordering Administration System

การค้นหาคลาสเพื่อระบุโอกาสการย้ายเมท็อดเป้าหมาย ไปยังคลาสที่เหมาะสม ตามการเรียกใช้เมท็อดเป้าหมายจากภายในคลาสและภายนอกคลาส

1) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหนึ่งคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้สามารถเสนอแนะเป็นทางเลือกในการย้ายเมท็อดโดยไม่ต้องคำนวณค่ามาตรวัดและไม่ต้องใช้เกณฑ์ลาปลาสในการตัดสินใจ

ตารางที่ ข.15 การเรียกใช้ภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Ordering Administration System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
CreditCardInfo	Customer	UserManager	0	1

2) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหลายคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ ข.16 การเรียกใช้ภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Ordering Administration System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
LogUser	Administrator	SearchFacade	0	2
	Administrator	UserManager	0	1
	Administrator	Log	0	3

การคำนวณค่ามาตรวัด ในกรณีนี้คำนวณด้วยมาตรวัด CIM, RMC และ ICBC โดยไม่ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ ข.17 การตัดสินใจเลือกคลาสที่เหมาะสม สำหรับเมท็อดเป้าหมายของระบบ Ordering Administration System

ทางเลือกลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
1	0.330	0.020	0.010	0.120
2	0.170	0.010	0.005	0.062
3	0.500	0.030	0.015	0.182

การเลือกคลาสที่เหมาะสม โดยการเลือกด้วยวิธีลาปลาส จะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด คือทางเลือกที่ 3 มีค่าเท่ากับ 0.182 ดังนั้น เลือกย้ายเมท็อด LogUser จากคลาส Administrator ไปยังคลาส Log

3) การค้นหาทางเลือกจาก เมท็อดเป้าหมาย ที่มีการเรียกใช้ภายในคลาสและมีการเรียกใช้จากภายนอกคลาสของระบบ Ordering Administration System

ตารางที่ ข.18 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Ordering Administration System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
ByKeyword	SearchFacade	UserManager	1	2
	SearchFacade	KeywordSet	1	3
CusShipping	Order	OrderDetail	1	4
	Order	Genre	1	2
	Order	ShippingInfo	1	1
DisplayProduct	Product	OrderDetail	2	3
	Product	ShippingCart	2	4
GetGenreCus	Genre	ShippingCart	2	1
GetGenreProduct	Genre	ShippingInfo	2	2
	Genre	Customer	2	1

การคำนวณค่ามาตรวัด ในกรณีนี้ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะมีการเรียกใช้เมท็อดเป้าหมายภายในคลาส จากนั้นคำนวณด้วยมาตรวัด CIM, RMC และ ICBC
โดยแบ่งเป็น 2 ขั้นตอน

ขั้นตอนที่ 1 คำนวณด้วยมาตรวัด IMC เพื่อจะคำนวณการเรียกใช้ภายในคลาส และการเรียกใช้จากภายนอกคลาส ของเมท็อดเป้าหมาย

ตารางที่ ข.19 การคำนวณด้วยมาตรวัด IMC ของระบบ Ordering Administration System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
ByKeyword	SearchFacade	UserManager	0.670
ByKeyword	SearchFacade	KeywordSet	0.750
CusShipping	Order	OrderDetail	0.800

ตารางที่ ข.19 การคำนวณด้วยมาตรวัด IMC ของระบบ Ordering Administration System(ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
CusShipping	Order	Genre	0.670
CusShipping	Order	ShippingInfo	0.500
DisplayProduct	Product	OrderDetail	0.600
DisplayProduct	Product	ShippingCart	0.770
GetGenreCus	Genre	ShippingCart	0.330
GetGenreProduct	Genre	ShippingInfo	0.670
GetGenreProduct	Genre	Customer	0.750

ขั้นตอนที่ 2 เลือกทางเลือกที่มีค่า IMC >= 0.70 มาคำนวณด้วยมาตรวัด CIM, RMC, ICBC ในแต่ละทางเลือกเพื่อพิจารณาจะย้ายเมท็อดไปที่คลาสใด

การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม จากระบบ Repairs Management สามารถเสนอแนะเมท็อดที่จะต้องย้าย ดังที่แสดงในตารางที่ ข.20

ตารางที่ ข.20 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Ordering Administration System

เมท็อด	คลาสต้นทาง	คลาสปลายทาง
CreditCardInfo	Customer	UserManager
LogUser	Administrator	Log
ByKeyword	SearchFacade	KeywordSet
CusShipping	Order	OrderDetail
DisplayProduct	Product	ShippingCart
GetGenreProduct	Genre	Customer

ข-4 ระบบ Graph of Function

การค้นหาคลาสเพื่อระบุโอกาสการย้ายเมท็อดเป้าหมาย ไปยังคลาสที่เหมาะสม ตามการเรียกใช้เมท็อดเป้าหมายจากภายในคลาสและภายนอกคลาส

1) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหนึ่งคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้สามารถเสนอแนะเป็นทางเลือกในการย้ายเมท็อดโดยไม่ต้องคำนวณค่ามาตรวัดและไม่ต้องใช้เกณฑ์ลาปลาสในการตัดสินใจ

ตารางที่ ข.21 การเรียกใช้ภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Graph of Function

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
IsBinaryOperation	MyChar	Neg	0	2
BuildListSyntaxNode	SyntaxTree	FormFunction	0	1
DeleteExcessiveBrackets	SyntaxTree	FormFunction	0	1
DrawBackGround	FormMy	FunctionColor	0	1
DrawFunction	FormMy	FunctionColor	0	1

2) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหลายคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ ข.22 การเรียกใช้ภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Graph of Function

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
ReadFunction	SyntaxTree	Ctg	0	4
	SyntaxTree	FormFunction	0	5
	SyntaxTree	SyntaxNode	0	2
ReadOperation	SyntaxTree	FormFunction	0	4
	SyntaxTree	Ctg	0	2
	SyntaxTree	Log	0	1

การคำนวณค่ามาตรวัด ในกรณีนี้คำนวณด้วยมาตรวัด CIM, RMC และ ICBC โดยไม่ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ ข.23 การตัดสินใจเลือกคลาสที่เหมาะสมสำหรับเมท็อดที่จะย้ายระบบ Graph of Function

ทางเลือก ลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
เมท็อด ReadFunction				
1	0.330	0.240	0.190	0.253
2	0.570	0.310	0.005	0.295
3	0.270	0.030	0.150	0.150
เมท็อด ReadOperation				
1	0.500	0.020	0.010	0.177
2	0.170	0.015	0.015	0.067
3	0.150	0.032	0.024	0.069

การเลือกคลาสที่เหมาะสม โดยการเลือกด้วยวิธีลาปลาส จะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด ในกรณีของเมท็อด ReadFunction คือ ทางเลือกที่ 3 มีค่าเท่ากับ 0.253 และในกรณีของเมท็อด ReadOperation คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.177 เลือกย้ายเมท็อด ReadFunction จากคลาส SyntaxTree ไปยังคลาส FormFunction และย้ายเมท็อด ReadOperation จากคลาส SyntaxTree ไปยังคลาส FormFunction

3) การค้นหาทางเลือกจาก เมท็อดเป้าหมาย ที่มีการเรียกใช้ภายในคลาสและมีการเรียกใช้จากภายนอกคลาสของระบบ Graph of Function

ตารางที่ ข.24 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Graph of Function

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
IsDomainOfFunction	BracketClosing	MyChar	3	3
	BracketClosing	BracketOpening	3	2
	BracketClosing	FormFunction	3	7
	BracketClosing	Var	3	5
BuildSyntaxTree	SyntaxTree	BracketClosing	2	1

ตารางที่ ข.24 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Graph of Function(ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
BuildSyntaxTree	SyntaxTree	BracketOpening	2	1
	SyntaxTree	SyntaxNode	2	4
Calculate	SyntaxTree	FormFunction	3	8
	SyntaxTree	SyntaxNode	3	2
	SyntaxTree	Ctg	3	5
	SyntaxTree	Log	3	1
CompareOperations	SyntaxTree	SyntaxNode	1	2
FindMinPrioritiOperation	SyntaxTree	FormFunction	3	1
ReadVar	SyntaxTree	FormFunction	4	10
	SyntaxTree	SyntaxNode	4	5
	SyntaxTree	Neg	4	3

การคำนวณค่ามาตรวัด ในกรณีนี้ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะมีการเรียกใช้เมท็อดเป้าหมายภายในคลาส จากนั้นคำนวณด้วยมาตรวัด CIM, RMC และ ICBC

โดยแบ่งเป็น 2 ขั้นตอน

ขั้นตอนที่ 1 คำนวณด้วยมาตรวัด IMC เพื่อจะคำนวณการเรียกใช้ภายในคลาส และการเรียกใช้จากภายนอกคลาส ของเมท็อดเป้าหมาย

ตารางที่ ข.25 การคำนวณด้วยมาตรวัด IMC ของระบบ Graph of Function

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
IsDomainOfFunction	BracketClosing	MyChar	0.500
	BracketClosing	BracketOpening	0.400
	BracketClosing	FormFunction	0.700
	BracketClosing	Var	0.630
BuildSyntaxTree	SyntaxTree	BracketClosing	0.330
	SyntaxTree	BracketOpening	0.330
	SyntaxTree	SyntaxNode	0.670

ตารางที่ ข.25 การคำนวณด้วยมาตรวัด IMC ของระบบ Graph of Function(ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
Calculate	SyntaxTree	FormFunction	0.790
	SyntaxTree	SyntaxNode	0.400
	SyntaxTree	Ctg	0.710
	SyntaxTree	Log	0.250
CompareOperations	SyntaxTree	SyntaxNode	0.670
FindMinPrioritiOperation	SyntaxTree	FormFunction	0.250
ReadVar	SyntaxTree	SyntaxNode	0.810
	SyntaxTree	FormFunction	0.700
	SyntaxTree	Neg	0.430

ขั้นตอนที่ 2 เลือกทางเลือกที่มีค่า IMC >= 0.70 มาคำนวณด้วยมาตรวัด CIM, RMC, ICBC ในแต่ละทางเลือกเพื่อพิจารณาจะย้ายเมท็อดไปที่คลาสใด

ตารางที่ ข.26 การคำนวณมาตรวัดในกรณีมีการเรียกใช้ภายในคลาสของระบบ Graph of Function

ทางเลือก ลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
เมท็อด Calculate				
1	0.750	0.370	0.150	0.423
2	0.480	0.250	0.310	0.347
เมท็อด ReadVar				
1	0.671	0.380	0.240	0.430
2	0.580	0.510	0.150	0.413

การเลือกคลาสที่เหมาะสม โดยจะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด ในกรณีของเมท็อด Calculate คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.423 และในกรณีของเมท็อด ReadVar คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.430 ดังนั้นเลือกย้ายเมท็อด Calculate จากคลาส SyntaxTree ไปยังคลาส FormFunction และย้ายเมท็อด ReadVar จากคลาส SyntaxTree ไปยังคลาส SyntaxNode

การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม จากระบบ Graph of Function สามารถเสนอแนะเมท็อดที่จะต้องย้าย ดังที่แสดงในตารางที่ ข.27

ตารางที่ ข.27 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Graph of Function

เมท็อด	คลาสต้นทาง	คลาสปลายทาง
IsBinaryOperation	MyChar	Neg
BuildListSyntaxNode	SyntaxTree	FormFunction
DeleteExcessiveBrackets	SyntaxTree	FormFunction
DrawBackGround	FormMy	FunctionColor
DrawFunction	FormMy	FunctionColor
ReadFunction	SyntaxTree	FormFunction
ReadOperation	SyntaxTree	FormFunction
Calculate	SyntaxTree	FormFunction
ReadVar	SyntaxTree	SyntaxNode

ข-5 ระบบ Image Fan

การค้นหาคลาสเพื่อระบุโอกาสการย้ายเมท็อดเป้าหมาย ไปยังคลาสที่เหมาะสม ตามการเรียกใช้เมท็อดเป้าหมายจากภายในคลาสและภายนอกคลาส

1) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหนึ่งคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ในกรณีนี้สามารถเสนอแนะเป็นทางเลือกในการย้ายเมท็อดโดยไม่ต้องคำนวณค่ามาตรวัดและไม่ต้องใช้เกณฑ์ลาปลาสในการตัดสินใจ

ตารางที่ ข.28 การเรียกใช้ภายนอกหนึ่งคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Image Fan

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
AddNewThumbnails	ThumbnailsSequence	ThumbnailBox	0	1
GenerateThumbnail	ThumbnailsSequence	ThumbnailBox	0	1
OnPaint	ThumbnailsSequence	ThumbnailBox	0	1
GenerateThumbnail	ThumbnailsSequence	ImageFolder	0	4
SetBorder	ThumbnailBox	ImageFolder	0	3

2) มีการเรียกใช้เมท็อดเป้าหมายจากคลาสภายนอกหลายคลาส และไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ ข.29 การเรียกใช้ภายนอกหลายคลาสและไม่มีการเรียกใช้ภายในคลาสของระบบ Image Fan

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
OnAfterCollapse	FolderTreeView	ImageFolder	0	2
	FolderTreeView	ImageForm	0	3
OnAfterExpand	FolderTreeView	ImageFolder	0	5
	FolderTreeView	FolderTreeView	0	2

การคำนวณค่ามาตรวัด ในกรณีนี้คำนวณด้วยมาตรวัด CIM, RMC และ ICBC โดยไม่ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะไม่มีการเรียกใช้เมท็อดเป้าหมายภายในคลาส

ตารางที่ ข.30 การคำนวณมาตรวัดในกรณีไม่มีการเรียกใช้ภายในคลาสของระบบ Image Fan

ทางเลือกลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
เมท็อด OnAfterCollapse				
1	0.250	0.270	0.351	0.290
2	0.470	0.015	0.250	0.245
เมท็อด OnAfterExpand				
1	0.650	0.150	0.180	0.327
2	0.350	0.250	0.080	0.227

การเลือกคลาสที่เหมาะสม โดยการเลือกด้วยวิธีลาปลาส จะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด ในกรณีของเมท็อด OnAfterCollapse คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.290 และในกรณีของเมท็อด OnAfterExpand คือ ทางเลือกที่ 1 มีค่าเท่ากับ 0.327 ดังนั้น เลือกย้ายเมท็อด OnAfterCollapse จากคลาส FolderTreeView ไปยังคลาส ImageFolder และย้ายเมท็อด OnAfterExpand จากคลาส FolderTreeView ไปยังคลาส ImageFolder

3) การค้นหาทางเลือกจาก เมท็อดเป้าหมาย ที่มีการเรียกใช้ภายในคลาสและมีการเรียกใช้จากภายนอกคลาสของระบบ Image Fan

ตารางที่ ข.31 การเรียกใช้ภายนอกคลาสและการเรียกใช้ภายในคลาสของระบบ Image Fan

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	ต้นทาง	ปลายทาง
DisposeOfPrevious Thumbnails	ThumbnailsSequence	FolderTreeView	2	11
	ThumbnailsSequence	ImageFolder	2	5
GetImageFromFile	ImageFile	Thumbnails Sequence	3	2
CreateImagesList	ImageFolder	ThumbnailBox	1	1
ImageSetUp	ImageFolder	FullScreenImage	4	1
	ImageFolder	FolderTreeView	4	5
	ImageFolder	ImageForm	4	2
BuildSubfoldersList	FolderTreeView	Thumbnails Sequence	2	1
SetPendingAction	ThumbnailsSequence	ImageForm	2	3
	ThumbnailsSequence	FolderTreeView	2	1
ReadImageFrom DiscThread	ThumbnailBox	FolderTreeView	1	1

การคำนวณค่ามาตรวัด ในกรณีนี้ต้องคำนวณด้วยมาตรวัด IMC ก่อน เพราะมีการเรียกใช้เมท็อดเป้าหมายภายในคลาส จากนั้นคำนวณด้วยมาตรวัด CIM, RMC และ ICBC

โดยแบ่งเป็น 2 ขั้นตอน

ขั้นตอนที่ 1 คำนวณด้วยมาตรวัด IMC เพื่อจะคำนวณการเรียกใช้ภายในคลาส และการเรียกใช้จากภายนอกคลาส ของเมท็อดเป้าหมาย

ตารางที่ ข.32 การคำนวณด้วยมาตรวัด IMC ของระบบ Image Fan

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
DisposeOfPrevious Thumbnails	ThumbnailsSequence	FolderTreeView	0.890
	ThumbnailsSequence	ImageFolder	0.670
SetPendingAction	ThumbnailsSequence	ImageForm	0.650
	ThumbnailsSequence	FolderTreeView	0.330

ตารางที่ ข.32 การคำนวณด้วยมาตรวัด IMC ของระบบ Image Fan(ต่อ)

เมท็อด	คลาสต้นทาง	คลาสปลายทาง	IMC
ImageSetUp	ImageFolder	FullScreenImage	0.250
	ImageFolder	FolderTreeView	0.560
	ImageFolder	ImageForm	0.350

ขั้นตอนที่ 2 เลือกทางเลือกที่มีค่า IMC >= 0.70 มาคำนวณด้วยมาตรวัด CIM, RMC, ICBC ในแต่ละทางเลือกเพื่อพิจารณาจะย้ายเมท็อดไปที่คลาสใด

ตารางที่ ข.33 การคำนวณมาตรวัดในกรณีมีการเรียกใช้ภายในคลาสของระบบ Image Fan

ทางเลือก ลำดับที่	เหตุการณ์			$U(a_i)$
	CIM	RMC	ICBC	
เมท็อด DisposeOfPreviousThumbnails				
1	0.670	0.480	0.240	0.463
2	0.480	0.530	0.150	0.390

การเลือกคลาสที่เหมาะสม โดยจะเลือกจากค่าเฉลี่ย($U(a_i)$) ที่มากที่สุด ในกรณีของเมท็อด DisposeOfPreviousThumbnails คือ ทางเลือกที่ 1 ดังนั้นเลือกย้ายเมท็อด DisposeOfPrevious-Thumbnails จากคลาส ThumbnailsSequence ไปยังคลาส FolderTree View

การเสนอแนะเมท็อดที่จะต้องย้ายจากคลาสเดิมไปยังคลาสที่เหมาะสม จากระบบ Image Fan สามารถเสนอแนะเมท็อดที่จะต้องย้าย ดังที่แสดงในตารางที่ ข.34

ตารางที่ ข.34 ทางเลือกที่แนะนำในการมูฟเมท็อดของระบบ Image Fan

เมท็อด	คลาสต้นทาง	คลาสปลายทาง
AddNewThumbnails	ThumbnailsSequence	ThumbnailBox
GenerateThumbnail	ThumbnailsSequence	ThumbnailBox
OnPaint	ThumbnailsSequence	ThumbnailBox
GenerateThumbnail	ThumbnailsSequence	ImageFolder
SetBorder	ThumbnailBox	ImageFolder
OnAfterCollapse	FolderTreeView	ImageFolder
DisposeOfPreviousThumbnails	ThumbnailsSequence	FolderTreeView
GenerateThumbnail	ThumbnailsSequence	ImageFolder
SetBorder	ThumbnailBox	ImageFolder

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวศศิพัชร บุญขวัญ เกิดวันที่ 31 สิงหาคม พ.ศ. 2526 ที่จังหวัดนครศรีธรรมราช สำเร็จการศึกษาหลักสูตรวิทยาศาสตร์บัณฑิต (วท.บ) สาขาวิทยาศาสตร์คอมพิวเตอร์ จากมหาวิทยาลัยรามคำแหงเมื่อปีการศึกษา 2552 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต(วท.ม) สาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2555

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University