

การประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพคลาสและแผนภาพซีคอนซ์



นายศุภวัชร รังสิยวัฒน์

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2549

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

ESTIMATING SOFTWARE LOGICAL STABILITY FROM CLASS DIAGRAM AND
SEQUENCE DIAGRAM

Mr. Supawath Rangsiyawath

สถาบันวิทยบริการ

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2006

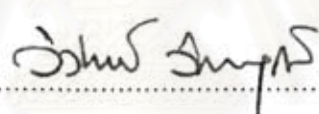
490431

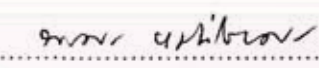
หัวข้อวิทยานิพนธ์	การประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพคลาสและแผนภาพ ซีควเอนซ์
โดย	นายศุภวัชร รังสียวัฒน์
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.พรศิริ หมั่นไชยศรี

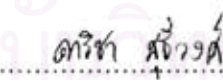
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้แนบวิทยานิพนธ์ฉบับนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต


..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวันยศิริ)

คณะกรรมการสอบวิทยานิพนธ์


..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวงศ์)


..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.พรศิริ หมั่นไชยศรี)


..... กรรมการ
(อาจารย์ ดร.ดาริชา สุธีวงศ์)


..... กรรมการ
(อาจารย์ นครทิพย์ พร้อมพูล)

นายคุภวัชร รังสิยวัฒน์ : การประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพคลาสและแผนภาพซีควเอนซ์. (ESTIMATING SOFTWARE LOGICAL STABILITY FROM CLASS DIAGRAM AND SEQUENCE DIAGRAM) อ. ที่ปรึกษา: รศ. ดร.พรศิริ นมื่นไวยศรี, 114 หน้า.

งานวิจัยนี้นำเสนอโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพคลาสและแผนภาพซีควเอนซ์ โดยหาความสัมพันธ์ระหว่างความเสถียรเชิงตรรกะที่วัดจากซอร์สโคดและค่าตัววัดจากแผนภาพคลาสและแผนภาพซีควเอนซ์

ความเสถียรเชิงตรรกะคำนวณจากผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงตามรูปแบบที่กำหนด โดยงานวิจัยนี้แบ่งรูปแบบการเปลี่ยนแปลงออกเป็น 2 ระดับด้วยกัน คือการเปลี่ยนแปลงระดับระบบ ซึ่งเป็นการเปลี่ยนแปลงที่เกี่ยวข้องกับความสัมพันธ์ระหว่างคลาส และการเปลี่ยนแปลงระดับคลาสซึ่งเป็นการเปลี่ยนแปลงภายในองค์ประกอบของคลาส การวัดค่าตัววัดแผนภาพในงานวิจัยนี้จะทำการวัดค่าตัววัดแผนภาพจากแผนภาพคลาสและแผนภาพซีควเอนซ์ที่อยู่ในรูปแบบของเอกสารเอกซ์เอ็มแอล

งานวิจัยนี้ได้แบ่งกลุ่มของซอฟต์แวร์ออกเป็น 3 กลุ่มได้แก่ ซอฟต์แวร์ด้านการคำนวณ ซอฟต์แวร์ด้านข้อความ และซอฟต์แวร์ด้านรูปภาพ เนื่องจากซอฟต์แวร์แต่ละกลุ่มมีพฤติกรรมเฉพาะ และบางพฤติกรรมขัดแย้งกัน และได้ใช้การวิเคราะห์ความถดถอยเชิงซ้อนเพื่อสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะจากตัววัดแผนภาพของซอฟต์แวร์แต่ละกลุ่ม และทำการพัฒนาเครื่องมือเพื่อประมาณค่าความเสถียรเชิงตรรกะตามโมเดลที่ได้

ผลลัพธ์ของงานวิจัยนี้ได้โมเดลการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ทั้ง 3 กลุ่ม ซึ่งโมเดลการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์การจัดการข้อความสามารถอธิบายค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ในกลุ่มของตนได้มากที่สุดคือ 84.6% รองลงมาคือ โมเดลการประมาณค่าของซอฟต์แวร์การคำนวณที่ 73.9% และโมเดลการประมาณค่าของซอฟต์แวร์การจัดการรูปภาพที่ 73.8%

ภาควิชา....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....*คุภวัชร รังสิยวัฒน์*
สาขาวิชา....วิศวกรรมซอฟต์แวร์.....ลายมือชื่ออาจารย์ที่ปรึกษา.....*ดร. นมื่นไวยศรี*
ปีการศึกษา2549.....

4670530421 : MAJOR SOFTWARE ENGINEERING

KEY WORD: METRICS / ESTIMATE / STABILITY / CLASS STABILITY / LOGICAL STABILITY / SYNTACTIC IMPACT

SUPAWATH RANGSIYAWATH : ESTIMATING SOFTWARE LOGICAL STABILITY FROM CLASS DIAGRAM AND SEQUENCE DIAGRAM. THESIS ADVISOR: ASSOC. PROF. PORNSIRI MUENCHAISRI, Ph.D., 114 pp.

This thesis proposes a model for estimating software logical stability from class diagram and sequence diagrams. The model is established by finding a relation between logical stability measured from source code and diagram metrics measured from class diagram and sequence diagrams.

The logical stability is measured by the effect of applying changes. Changes can be divided into 2 levels: system level which is changes related to relation between classes and class level which are changes that apply to class's elements. The diagram metrics measured from class diagram and sequence diagram are in XML format.

This research divides software into 3 categories: calculating software, text editing software and image processing software because each category has different behavior and may conflict with another category. This research uses Multiple Linear Regression technique to establish a logical stability estimation model for each category and implements a tool to exercise the established models.

The results of this research is logical stability estimation model for each software category, which the estimation model of text editing software can describe its logical stability the most at 84.6%, follow by the estimation model of calculating software at 73.9% and the estimation model of image processing software at 73.8%.

Department.... Computer Engineering.... Student's signature... *Supawath R.*.....
Field of study.... Software Engineering... Advisor's signature... *P. Muenchaisri*.....
Academic year ...2006.....

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณรองศาสตราจารย์ ดร.พรศิริ หมั่นไชยศรี อาจารย์ที่ปรึกษาวิทยานิพนธ์ของข้าพเจ้า ที่กรุณาให้คำปรึกษาและข้อเสนอแนะ รวมถึงคอยดูแลและช่วยเหลือข้าพเจ้ามาโดยตลอด จนกระทั่งวิทยานิพนธ์ฉบับนี้เสร็จสมบูรณ์

ข้าพเจ้าขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ ซึ่งเป็นประธานกรรมการสอบวิทยานิพนธ์ อาจารย์ ดร.ดาวิชา สุธีวงศ์ และอาจารย์ นครทิพย์ พร้อมพูล ซึ่งเป็นกรรมการสอบวิทยานิพนธ์ ที่ได้สละเวลามาเป็นกรรมการสอบให้ข้าพเจ้า และให้คำแนะนำต่างๆ

ขอขอบคุณเพื่อนๆ พี่ๆ ทุกคนในภาควิชาที่ให้คำปรึกษาและความคิดเห็นในการทำวิทยานิพนธ์ฉบับนี้ และขอขอบพระคุณ คุณพ่อ คุณแม่ รวมไปถึงครอบครัว และคนรอบตัวข้าพเจ้าที่คอยเป็นกำลังใจให้



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ	ช
สารบัญภาพ.....	ฅ
สารบัญตาราง.....	ญ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขตงานวิจัย	3
1.4 ขั้นตอนและวิธีดำเนินงานวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	5
2.1 ทฤษฎีที่เกี่ยวข้อง	5
2.1.1 ผลกระทบวงคลื่น	5
2.1.2 ความเสถียร	6
2.1.3 แผนภาพคลาสและแผนภาพซีควเन्ซ์.....	8
2.1.4 การวิเคราะห์ความถดถอยเชิงซ้อน	10
2.1.5 การทำนายที่ระดับแอล	14
2.2 งานวิจัยที่เกี่ยวข้อง	15
2.2.1 Investigation of Metrics for Object-Oriented Design Logical Stability.....	15
2.2.2 Ripple effect analysis for Software Maintenance.....	16
2.2.3 Some stability measures for software maintenance	17
บทที่ 3 การสร้างโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะ	20
3.1 คำนวณความเสถียรเชิงตรรกะของคลาสของชุดโปรแกรมพัฒนา	21
3.2 การแปลงซอร์สโคดเป็นแผนภาพคลาสและแผนภาพซีควเन्ซ์.....	30
3.3 การวัดค่าตัววัดแผนภาพที่ได้จากการแปลงกลับ	31
บทที่ 4 การสร้างและทดสอบโมเดลประมาณค่าความเสถียรเชิงตรรกะ	34

4.1 การสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะ.....	34
4.1.1 การสร้างโมเดลของกลุ่มโปรแกรมด้านการคำนวณ	38
4.1.2 การสร้างโมเดลของกลุ่มโปรแกรมด้านการจัดการข้อความ	44
4.1.3 การสร้างโมเดลของกลุ่มโปรแกรมการจัดการรูปภาพ	47
4.2 การประเมินผลโมเดลประมาณค่าความเสถียรเชิงตรรกะ	52
บทที่ 5 การออกแบบและพัฒนาเครื่องมือคำนวณความเสถียรเชิงตรรกะและเครื่องมือ	
ประมาณค่าความเสถียรเชิงตรรกะ	57
5.1 การออกแบบและพัฒนาเครื่องมือคำนวณความเสถียรเชิงตรรกะ	57
5.1.1 โครงสร้างของเครื่องมือคำนวณเสถียรภาพ	57
5.1.2 การพัฒนาเครื่องมือคำนวณความเสถียรเชิงตรรกะ	60
5.2 การออกแบบและพัฒนาเครื่องมือประมาณค่าความเสถียรเชิงตรรกะ.....	64
5.2.1 โครงสร้างการทำงานของโปรแกรม SDMetrics.....	64
5.2.2 การอ่านค่าข้อมูลแผนภาพและคำนวณตัววัดแผนภาพ	65
5.2.3 การแก้ไขและพัฒนาความสามารถในการประมาณค่าความเสถียรเชิงตรรกะ	
ให้แก่โปรแกรม SDMetrics	71
บทที่ 6 สรุปผลการวิจัย.....	81
6.1 สรุปผลการวิจัย.....	81
6.2 ข้อเสนอแนะ.....	83
6.3 แนวทางการวิจัยในอนาคต	83
6.4 ผลงานตีพิมพ์	84
รายการอ้างอิง.....	85
ภาคผนวก.....	87
ประวัติผู้เขียนวิทยานิพนธ์.....	114

สารบัญตาราง

	หน้า
ตารางที่ 3.1 รูปแบบการเปลี่ยนแปลง	22
ตารางที่ 3.2 ตัวอย่างซอร์สโค้ด	28
ตารางที่ 3.3 ตัวอย่างการคำนวณความเสถียรเชิงตรรกะ	29
ตารางที่ 4.1 รายชื่อโปรแกรมที่อยู่ในชุดพัฒนา	34
ตารางที่ 4.2 ตัวอย่างค่าความเสถียรเชิงตรรกะของคลาสและค่าตัววัดแผนภาพที่ใช้ในการ สร้างโมเดล	37
ตารางที่ 4.3 ผลสรุปของโมเดลสำหรับโปรแกรมด้านการคำนวณ	39
ตารางที่ 4.4 ตัววัดแผนภาพที่ถูกนำเข้าและออกจากโมเดลสำหรับโปรแกรมด้านการคำนวณ	40
ตารางที่ 4.5 ค่าสัมประสิทธิ์ถดถอยเชิงส่วนของตัววัดแผนภาพสำหรับโปรแกรม ด้านการคำนวณ	41
ตารางที่ 4.6 ตัววัดแผนภาพที่ไม่ได้ถูกนำมาใช้ สำหรับโปรแกรมด้านการคำนวณ	42
ตารางที่ 4.7 ผลสรุปของโมเดลสำหรับโปรแกรมด้านการจัดการข้อความ	45
ตารางที่ 4.8 ตัววัดแผนภาพที่ถูกนำเข้าและออกจากโมเดลสำหรับโปรแกรมด้าน การจัดการข้อความ	45
ตารางที่ 4.9 ค่าสัมประสิทธิ์ถดถอยเชิงส่วนของตัววัดแผนภาพสำหรับโปรแกรมด้าน การจัดการข้อความ	45
ตารางที่ 4.10 ตัววัดแผนภาพที่ไม่ได้ถูกนำมาใช้สำหรับโปรแกรมด้านการจัดการข้อความ	46
ตารางที่ 4.11 ผลสรุปของโมเดลสำหรับโปรแกรมด้านการจัดการรูปภาพ	48
ตารางที่ 4.12 ตัววัดแผนภาพที่ถูกนำเข้าและออกจากโมเดลสำหรับโปรแกรมด้าน การจัดการรูปภาพ	48
ตารางที่ 4.13 ค่าสัมประสิทธิ์ถดถอยเชิงส่วนของตัววัดแผนภาพสำหรับโปรแกรมด้านการจัดการ รูปภาพ	49
ตารางที่ 4.14 ตัววัดแผนภาพที่ไม่ได้ถูกนำมาใช้สำหรับโปรแกรมด้านการจัดการรูปภาพ	50
ตารางที่ 4.14 ตัววัดแผนภาพที่ไม่ได้ถูกนำมาใช้สำหรับโปรแกรมด้านการจัดการรูปภาพ (ต่อ) ...	51
ตารางที่ 4.15 รายชื่อโปรแกรมในชุดโปรแกรมสำหรับประเมินผล	52
ตารางที่ 4.16 ผลการประเมินโมเดลประมาณค่าความเสถียรเชิงตรรกะของชุดโปรแกรม การคำนวณ	54

ตารางที่ 4.17 ผลการประเมินโมเดลประมาณค่าความเสถียรเชิงตรรกะของชุดโปรแกรม การจัดการข้อความ	54
ตารางที่ 4.18 ผลการประเมินโมเดลประมาณค่าความเสถียรเชิงตรรกะของชุดโปรแกรม การจัดการภาพ	55
ตารางที่ 5.1 โครงสร้างของไฟล์ Metamodel Definition.....	65
ตารางที่ 5.2 การประกาศนิยามของโอเปอเรชัน	66
ตารางที่ 5.3 โครงสร้างของไฟล์ XMI Transformation.....	67
ตารางที่ 5.4 ตัวอย่างข้อกำหนดการเปลี่ยนรูปขององค์ประกอบโอเปอเรชัน	68
ตารางที่ 5.5 โครงสร้างรายละเอียดตัววัดแผนภาพ	69
ตารางที่ 5.6 ส่วนของเอกสารเอกซ์เอ็มแอลที่แสดงความสัมพันธ์แอตทริบิวต์	73
ตารางที่ 5.7 รายละเอียดองค์ประกอบจุดสิ้นสุดแอตทริบิวต์	73
ตารางที่ 5.8 รายละเอียดการเก็บค่าองค์ประกอบจุดสิ้นสุดแอตทริบิวต์	74
ตารางที่ 5.9 รายละเอียดตัววัดแผนภาพ	74
ตารางที่ 5.10 ส่วนของเอกสารเอกซ์เอ็มแอลที่แสดงความสัมพันธ์เจเนอรัลไลเซชัน	75
ตารางที่ 5.11 รายละเอียดองค์ประกอบความสัมพันธ์เจเนอรัลไลเซชัน	75
ตารางที่ 5.12 รายละเอียดการเก็บค่าองค์ประกอบความสัมพันธ์เจเนอรัลไลเซชัน.....	75
ตารางที่ 5.13 รายละเอียดตัววัดแผนภาพจำนวนคลาสแม่	76
ตารางที่ 5.14 รายละเอียดการคำนวณโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของ คลาสสำหรับโปรแกรมด้านการคำนวณ	76
ตารางที่ 5.15 รายละเอียดการคำนวณโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของ คลาสสำหรับโปรแกรมด้านการจัดการข้อความ	77
ตารางที่ 5.16 รายละเอียดการคำนวณโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของ คลาสสำหรับโปรแกรมด้านการจัดการรูปภาพ	77
ตารางที่ 5.17 รายละเอียดองค์ประกอบซอฟต์แวร์	77
ตารางที่ 5.18 รายละเอียดการเก็บค่าองค์ประกอบซอฟต์แวร์	77
ตารางที่ 5.19 รายละเอียดตัววัดแผนภาพจำนวนคลาสทั้งหมดในแพ็คเกจ	78
ตารางที่ 5.20 รายละเอียดตัววัดแผนภาพผลรวมค่าประมาณความเสถียรเชิงตรรกะของ ทุกคลาสในแพ็คเกจ	78
ตารางที่ 5.21 รายละเอียดตัววัดค่าประมาณความเสถียรเชิงตรรกะของแพ็คเกจ	78
ตารางที่ 5.22 รายละเอียดตัววัดแผนภาพจำนวนแพ็คเกจทั้งหมดในซอฟต์แวร์	79

ตารางที่ 5.23 รายละเอียดตัววัดแผนภาพผลรวมค่าประมาณความเสถียรเชิงตรรกะของทุก แพคเกจในซอฟต์แวร์.....	79
ตารางที่ 5.24 รายละเอียดตัววัดแผนภาพจำนวนคลาสทั้งหมดในซอฟต์แวร์.....	79
ตารางที่ 5.25 รายละเอียดตัววัดแผนภาพผลรวมค่าประมาณความเสถียรเชิงตรรกะของ ทุกคลาส.....	80
ตารางที่ 5.26 รายละเอียดตัววัดค่าประมาณความเสถียรเชิงตรรกะของซอฟต์แวร์.....	80
ตารางที่ ก-1 อัลกอริทึมหาค่าความเสถียรเชิงตรรกะของคลาส.....	88
ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอร์กับการเปลี่ยนแปลง.....	91
ในส่วนนี้จะแสดงข้อมูลที่ใช้ในการประเมินผลโมเดลการประมาณค่าความเสถียรเชิงตรรกะ ของทั้ง 3 กลุ่มซอฟต์แวร์.....	96
ตารางที่ ค-1 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการคำนวณ.....	96
ตารางที่ ค-2 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการจัดการข้อความ.....	98
ตารางที่ ค-3 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการจัดการรูปภาพ.....	99

สารบัญภาพ

หน้า

รูปที่ 2.1 ตัวอย่างความสัมพันธ์ระหว่างคลาส.....	7
รูปที่ 2.2 ตัวอย่างแผนภาพคลาส.....	9
รูปที่ 2.3 ตัวอย่างแผนภาพซีเคว้นซ์.....	10
รูปที่ 2.4 การส่งผลกระทบต่อกันของการเกิดผลกระทบวงคลื่นเชิงประสิทธิภาพ	17
รูปที่ 3.1 ขั้นตอนการดำเนินงานวิจัย.....	20
รูปที่ 3.2 ขั้นตอนการคำนวณความเสถียรเชิงตรรกะ.....	27
รูปที่ 3.3 ตัวอย่างแผนภาพคลาสที่ได้จากการแปลงกลับ.....	30
รูปที่ 3.4 ตัวอย่างแผนภาพซีเคว้นซ์ที่ได้จากการแปลงกลับ.....	31
รูปที่ 5.1 การทำงานของเครื่องมือคำนวณเสถียรภาพ	58
รูปที่ 5.2 แผนภาพคลาสของเครื่องมือคำนวณเสถียรภาพเฉพาะส่วนที่เกี่ยวข้อง.....	59
รูปที่ 5.3 แผนภาพซีเคว้นซ์การทำงานของเครื่องมือคำนวณเสถียรภาพ.....	59
รูปที่ 5.4 ขั้นตอนการคำนวณความเสถียรเชิงตรรกะของเครื่องมือคำนวณความเสถียร เชิงตรรกะ	60
รูปที่ 5.5 ขั้นตอนการจำลองการเปลี่ยนแปลงในระดับของระบบ	61
รูปที่ 5.6 ขั้นตอนการจำลองการเปลี่ยนแปลงระดับของคลาส.....	62
รูปที่ 5.7 ส่วนต่อประสานผู้ใช้ของเครื่องมือคำนวณความเสถียรเชิงตรรกะ	63
รูปที่ 5.8 ขั้นตอนการค้นหาไฟล์ทั้งหมดในไดเรกทอรีที่ต้องการ	63
รูปที่ 5.9 โครงสร้างการทำงานของโปรแกรม SDMetrics	64
รูปที่ 5.10 ตัวอย่างคลาสที่มีความสัมพันธ์แอสโซซิเอชัน.....	72
รูปที่ ๖-1 หน้าต่างการทำงานหลักของโปรแกรม MagicDraw UML.....	100
รูปที่ ๖-2 การสร้าง code engineering set.....	101
รูปที่ ๖-3 code engineering set ที่สร้างขึ้น.....	101
รูปที่ ๖-4 หน้าจอสำหรับเลือกไฟล์ซอร์สโค้ด.....	102
รูปที่ ๖-5 หน้าจอสำหรับเลือกไฟล์ซอร์สโค้ดเมื่อได้ทำการเลือกไฟล์ที่ต้องการแล้ว	102
รูปที่ ๖-6 รายการของไฟล์ซอร์สโค้ด	103
รูปที่ ๖-7 การเลือกคำสั่งเพื่อทำการแปลงกลับข้อมูล	103
รูปที่ ๖-8 ตัวเลือกการแปลงกลับ	104
รูปที่ ๖-9 แผนภาพคลาสที่ได้จากการแปลงกลับ.....	105

รูปที่ ง-10 รายการคลาสภายในไฟล์ซอร์สโค้ด	105
รูปที่ ง-11 การแปลงกลับแผนภาพชีเควนซ์	106
รูปที่ ง-12 แผนภาพชีเควนซ์ที่ได้จากการแปลงกลับ	106
รูปที่ จ-1 การตั้งค่าไอดีบีซี	107
รูปที่ จ-2 หน้าจอการทำงานหลักของโปรแกรมคำนวณความเสถียรเชิงตรรกะ	107
รูปที่ จ-3 การเลือกไฟล์ซอร์สโค้ด	108
รูปที่ จ-4 รายการของไฟล์ซอร์สโค้ดที่เลือก	108
รูปที่ จ-5 หน้าจอการทำงานเมื่อโปรแกรมคำนวณเสร็จ	109
รูปที่ จ-6 ตัวอย่างข้อมูลในฐานข้อมูล	109
รูปที่ ฉ-1 หน้าจอการทำงานหลักของโปรแกรม SDMetrics	110
รูปที่ ฉ-2 การตั้งค่าโปรเจคต์	111
รูปที่ ฉ-3 หน้าจอการแสดงผลข้อมูลสถิติตัววัด	112
รูปที่ ฉ-4 หน้าจอการแสดงผลข้อมูลแผนภาพตัววัดขององค์ประกอบ	112
รูปที่ ฉ-5 หน้าจอการแสดงผลแบบตาราง	113
รูปที่ ฉ-6 การเลือกระดับของการแสดงผล	113

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การเปลี่ยนแปลงซอฟต์แวร์ (Software Change) คือ การเปลี่ยนแปลงองค์ประกอบต่างๆ ของซอฟต์แวร์ เช่น ซอร์สโค้ด (Source Code) อินเตอร์เฟซ (Interface) หรือ แผนภาพการออกแบบ (Design Diagram) เป็นต้น โดยมีวัตถุประสงค์เพื่อแก้ไขข้อผิดพลาดของซอฟต์แวร์ เพิ่มความสามารถของซอฟต์แวร์ หรือปรับปรุงประสิทธิภาพของซอฟต์แวร์ โดยเมื่อมีการเปลี่ยนแปลงย่อมเกิดผลกระทบตามมา เรียกว่าผลกระทบจากการเปลี่ยนแปลง (Change Impact) โดยผลกระทบที่เกิดขึ้นสามารถกระจาย (ripple) ไปยังส่วนอื่นๆ ของซอฟต์แวร์ได้ ไม่จำกัดอยู่เพียงส่วนที่เกิดการเปลี่ยนแปลง การกระจายของผลกระทบไปยังส่วนอื่นๆ ของซอฟต์แวร์ เรียกว่าการเกิดผลกระทบวงคลื่น (Ripple Effect) การเกิดผลกระทบวงคลื่นไปยังส่วนต่างๆ ของซอฟต์แวร์ทำให้การเปลี่ยนแปลงซอฟต์แวร์มีความยุ่งยากยิ่งขึ้น ทั้งนี้ผลกระทบที่เกิดขึ้นอาจทำให้ซอฟต์แวร์ทำงานผิดพลาด หรือไม่สามารถทำงานได้เลย ดังนั้นในการเปลี่ยนแปลงซอฟต์แวร์จึงจำเป็นต้องทำการวิเคราะห์ผลกระทบของการเปลี่ยนแปลง (Change Impact Analysis) เพื่อให้ทราบถึงผลกระทบที่อาจเกิดขึ้นหากทำการเปลี่ยนแปลง ทั้งนี้เพื่อวางแผนและจัดการกับผลกระทบที่อาจเกิดขึ้นไม่ให้กลายเป็นปัญหาหรือข้อผิดพลาดใหม่ขึ้นมา

ในการวิเคราะห์ผลกระทบของการเปลี่ยนแปลงมีวิธีการวิเคราะห์และประเมินระดับของผลกระทบหลายวิธี ตามแต่การนิยามความหมายของการวิเคราะห์ผลกระทบการเปลี่ยนแปลงของแต่ละผู้วิจัย เช่น ได้มีการให้คำนิยามไว้ใน [1] กล่าวว่า การวิเคราะห์ผลกระทบการเปลี่ยนแปลงคือการประเมินความเสี่ยงที่เกิดจากการเปลี่ยนแปลงรวมไปถึงการประเมินผลกระทบต่อทรัพยากรแรงงานและเวลา และได้มีการให้คำนิยามไว้ใน [2] ว่าการวิเคราะห์ผลกระทบการเปลี่ยนแปลงคือการหาผลลัพธ์ที่เกิดจากการเปลี่ยนแปลง และประเมินความต้องการในการแก้ไขให้สอดคล้องกับการเปลี่ยนแปลง เป็นต้น และในการวิเคราะห์ผลกระทบการเปลี่ยนแปลงมีตัวชี้วัดหนึ่งสามารถบอกถึงระดับของผลกระทบที่จะเกิดขึ้นเมื่อทำการแก้ไขซอฟต์แวร์ คือ ความเสถียรของซอฟต์แวร์ (Software Stability) โดยความเสถียรของซอฟต์แวร์คือ โอกาสในการไม่ได้รับผลกระทบจากการเปลี่ยนแปลงของซอฟต์แวร์หรืออาจจะเป็นส่วนกลับของผลกระทบวงคลื่น โดยซอฟต์แวร์ที่มี

ความเสถียรสูงคือ ซอฟต์แวร์ที่มีโอกาสได้รับผลกระทบจากการเปลี่ยนแปลงต่ำทำให้สามารถทำการเปลี่ยนแปลงแก้ไขได้อย่างสะดวกและรวดเร็ว

ความเสถียรของซอฟต์แวร์สามารถแบ่งออกได้หลายแบบด้วยกันตามผลกระทบที่เกิดขึ้น แต่โดยส่วนใหญ่มักแบ่งออกเป็น ความเสถียรเชิงตรรกะ (Logical Stability)[3] และความเสถียรเชิงประสิทธิภาพ (Performance Stability)[3] โดยความเสถียรเชิงตรรกะคือ โอกาสในการไม่ได้รับผลกระทบที่ส่งผลให้การทำงานผิดพลาด และความเสถียรเชิงประสิทธิภาพคือ โอกาสในการไม่ได้รับผลกระทบที่ส่งผลให้ประสิทธิภาพในการทำงานเปลี่ยนไป โดยรายละเอียดของความเสถียรทั้ง 2 แบบจะอธิบายในบทที่ 2 ปัจจุบันมีผู้ทำการวิจัยเกี่ยวกับความเสถียรเชิงตรรกะอยู่มาก แต่โดยส่วนใหญ่เป็นการคำนวณความเสถียรเชิงตรรกะจากซอร์สโคดทั้งสิ้น ทำให้ไม่สามารถทำการวิเคราะห์เพื่อปรับปรุงซอฟต์แวร์ให้มีความเสถียรสูงได้ตั้งแต่ขั้นตอนการออกแบบ จำเป็นต้องรอจนซอฟต์แวร์พัฒนาเสร็จจึงสามารถทำการวัดค่าความเสถียรเชิงตรรกะได้ นอกจากนี้จากทฤษฎีค่าใช้จ่ายของการเปลี่ยนแปลงของ Boehm [4] ระบุว่า การเปลี่ยนแปลงซอฟต์แวร์ในช่วงหลังของการพัฒนาจะเสียค่าใช้จ่ายมากกว่าในช่วงแรกมาก ดังนั้นหากสามารถทำการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ได้ตั้งแต่ช่วงต้นของการพัฒนาจะช่วยในการตัดสินใจของผู้พัฒนาเพื่อทำการแก้ไขปรับปรุงการออกแบบซอฟต์แวร์ต่อไป

วิทยานิพนธ์นี้จึงมีวัตถุประสงค์เพื่อหาโมเดลในการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์จากแบบจำลองที่ใช้ในการออกแบบซอฟต์แวร์ โดยในวิทยานิพนธ์นี้ผู้วิจัยได้นิยามความเสถียรเชิงตรรกะเป็นส่วนกลับผลกระทบวงคลื่นเชิงตรรกะ และผู้วิจัยได้เลือกทำการวิจัยแผนภาพคลาสและแผนภาพซีควเन्ซ์ ทั้งนี้เพราะปัจจุบันการพัฒนาซอฟต์แวร์อาศัยแนวความคิดเชิงวัตถุและแผนภาพยูเอ็มแอลเป็นหลัก และแผนภาพทั้ง 2 เป็นแผนภาพที่ช่วยให้เห็นโครงสร้างและขั้นตอนการทำงานของซอฟต์แวร์ที่ออกแบบได้อย่างชัดเจน

ผลลัพธ์ของวิทยานิพนธ์นี้คือ โมเดลในการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์จากแผนภาพคลาสและแผนภาพซีควเन्ซ์ และซอฟต์แวร์ที่ใช้ในการประมาณค่าโดยรับข้อมูลเป็นแผนภาพคลาสและแผนภาพซีควเन्ซ์ที่ทำการแปลงเป็นรหัสข้อมูลเอกซ์เอ็มไอแล้ว จากนั้นทำการประมวลผลและประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ เพื่อทำการสรุปผลแจ้งแก่ผู้ใช้ถึงความเสถียรเชิงตรรกะของซอฟต์แวร์ที่ออกแบบ

1.2 วัตถุประสงค์

1. สร้างโมเดลในการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์จากแผนภาพคลาสและแผนภาพซีควเन्ซ์

2. พัฒนาเครื่องมือประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์เมื่อพัฒนาเสร็จจากแผนภาพคลาสและแผนภาพซีควเอนซ์

1.3 ขอบเขตงานวิจัย

1. โมเดลในการประมาณค่าความเสถียรเชิงตรรกะในวิทยานิพนธ์นี้สำหรับซอฟต์แวร์ที่พัฒนาตามโมเดลการออกแบบซอฟต์แวร์เชิงวัตถุด้วย แผนภาพคลาสและแผนภาพซีควเอนซ์ของยูเอ็มแอลเวอร์ชัน 1.4 และพัฒนาด้วยภาษาจาวาเท่านั้น
2. การวัดความเสถียรเชิงตรรกะจากซอร์สโคดทำการวิเคราะห์ผลกระทบจากการเปลี่ยนแปลงในตารางที่ 3.1 เท่านั้น
3. วิทยานิพนธ์นี้ทำการหาโมเดลประมาณค่าความเสถียรเชิงตรรกะเท่านั้นไม่ได้เสนอแนะแนวทางการปรับปรุงความเสถียรเชิงตรรกะ
4. พัฒนาซอฟต์แวร์สำหรับประมาณค่าความเสถียรเชิงตรรกะตามโมเดลที่ได้จากการทดลอง โดยมีข้อจำกัดดังนี้
 - ข้อมูลเข้าเป็นแผนภาพที่แปลงอยู่ในรูปแบบเอกซ์เอ็มไอเท่านั้น
 - ผลลัพธ์ที่ได้เป็นตัวเลขแสดงผลในรูปแบบตาราง โดยแสดงค่าความเสถียรเชิงตรรกะที่ประมาณได้ของแต่ละคลาส ค่าตัววัดแผนภาพของแต่ละคลาส และค่าความเสถียรเชิงตรรกะของซอฟต์แวร์
 - พัฒนาโดยภาษาจาวา
 - สามารถทำงานได้บนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ และระบบปฏิบัติการยูนิกซ์ที่รองรับภาษาจาวา

1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1. ศึกษาและค้นหาโมเดลวัดค่าความเสถียรเชิงตรรกะที่เหมาะสมต่อการทดลองจากบทความวิชาการต่างๆ
2. สร้างและเลือกซอฟต์แวร์สำหรับการทดลองตามคุณสมบัติที่กำหนด
3. วัดค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ที่เลือกมา
4. แปลงซอฟต์แวร์ที่เลือกมาเป็นแผนภาพคลาสและแผนภาพซีควเอนซ์ด้วยโปรแกรม MagicDraw UML 9.5
5. แปลงแผนภาพที่ได้ให้อยู่ในรูปแบบภาษาเอกซ์เอ็มไอด้วยโปรแกรม MagicDraw UML 9.5

6. วัดค่าตัววัดแผนภาพที่ได้ทำการแปลงให้อยู่ในรูปแบบภาษาเอกซ์เอ็มแอลแล้วด้วยโปรแกรม SDMetrics 2.0
7. หาสมการในการประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพด้วยวิธีการวิเคราะห์ถดถอยเชิงเส้นแบบหลายตัวแปร
8. ตรวจสอบความถูกต้องของโมเดลที่ได้จากการทดลอง
9. สร้างซอฟต์แวร์เพื่อทำการประมาณค่าความเสถียรเชิงตรรกะ
10. ตรวจสอบความถูกต้องของซอฟต์แวร์ที่พัฒนาขึ้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้โมเดลในการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ที่พัฒนาเสร็จแล้วจากแผนภาพคลาสและแผนภาพซีควเन्ซ์
2. ได้ซอฟต์แวร์ในการวิเคราะห์แผนภาพคลาสและซีควเन्ซ์เพื่อประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์เมื่อพัฒนาเสร็จสิ้น และสรุปข้อมูลเพื่อให้ผู้ใช้ได้ตัดสินใจเพื่อปรับปรุงการออกแบบซอฟต์แวร์ต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ผลกระทบวงคลื่น

ผลกระทบวงคลื่น คือปรากฏการณ์หนึ่งที่เกิดขึ้นเมื่อมีการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของโปรแกรมซึ่งจะส่งผลกระทบต่อส่วนอื่นๆ ของโปรแกรม [3] ตัวอย่างเช่น สมมติให้โปรแกรม P ประกอบไปด้วยส่วนย่อย 6 ส่วน คือ A B C D E และ F เมื่อมีการเปลี่ยนแปลงที่ส่วนย่อย A แล้วส่งผลให้ส่วนย่อย B และ C เกิดการเปลี่ยนแปลงตาม เช่น ไม่สามารถทำงานได้ หรือประสิทธิภาพของส่วนย่อยของโปรแกรมมีค่าลดลง การเกิดผลกระทบที่กระจายออกจากการเปลี่ยนแปลง ณ จุดใดจุดหนึ่งนี้ เรียกว่าผลกระทบวงคลื่น

ผลกระทบวงคลื่นสามารถแบ่งออกได้หลายแบบด้วยกัน เช่น งานวิจัยของ Yau และ Collofello [5] ได้แบ่งออกเป็น 2 แบบด้วยกัน คือ

- 1) ผลกระทบวงคลื่นเชิงตรรกะ (Logical Ripple Effect)
- 2) ผลกระทบวงคลื่นเชิงประสิทธิภาพ (Performance Ripple Effect)

นอกจากนี้ยังมีนักวิจัยทำการแบ่งผลกระทบวงคลื่นออกเป็นประเภทต่างๆ อีกดังนี้ [6]

- 1) ผลกระทบวงคลื่นเชิงการเขียนโปรแกรม (Coding Ripple Effect)
- 2) ผลกระทบวงคลื่นเชิงข้อมูล (Data Ripple Effect)
- 3) ผลกระทบวงคลื่นเชิงเอกสาร (Documentation Ripple Effect)
- 4) ผลกระทบวงคลื่นเชิงความต้องการ (Requirement Ripple Effect)
- 5) ผลกระทบวงคลื่นเชิงส่วนเชื่อมต่อ (Interface Ripple Effect)
- 6) ผลกระทบวงคลื่นเชิงสภาวะแวดล้อม (Environment Ripple Effect)
- 7) ผลกระทบวงคลื่นเชิงการจัดการ (Management Ripple Effect)

ในวิทยานิพนธ์นี้จะศึกษาโดยเน้นผลกระทบวงคลื่นเชิงตรรกะเป็นหลัก และมีนิยามคือผลกระทบจากการเปลี่ยนแปลงคลาสใดคลาสหนึ่งของโปรแกรมต่อการทำงานของคลาสอื่นๆ

2.1.2 ความเสถียร

ความเสถียรของซอฟต์แวร์ คือค่าความต้านทานของซอฟต์แวร์ต่อผลกระทบจากการเปลี่ยนแปลงที่เกิดขึ้นแก่ซอฟต์แวร์ [3] หรืออาจนิยามเป็นค่าความต้านทานของซอฟต์แวร์ต่อผลกระทบวงคลื่น ความเสถียรของซอฟต์แวร์สามารถแบ่งออกได้เป็น 2 แบบ ดังนี้

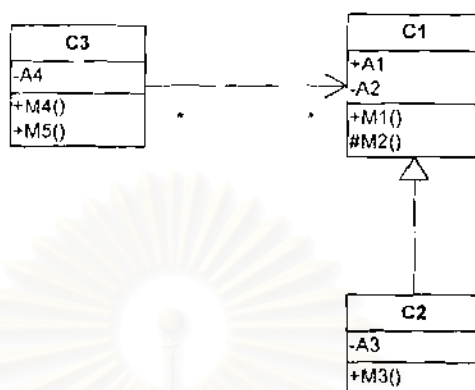
- 1) ความเสถียรเชิงตรรกะ
- 2) ความเสถียรเชิงประสิทธิภาพ

ความเสถียรเชิงตรรกะคือ โอกาสในการไม่ได้รับผลกระทบที่ส่งผลให้การทำงานผิดพลาด หรือไม่สามารถทำงานได้ ตัวอย่างของผลกระทบจากการเปลี่ยนแปลงที่ส่งผลเชิงตรรกะ เช่น จากรูปที่ 2.1 ให้เมทอด M4() ของคลาส C3 มีการอ้างถึง เมทอด M1() ของคลาส C1 และเมื่อมีการเปลี่ยนแปลงซิกเนเจอร์ของเมทอด M1() แล้วเมทอด M4() ที่ไม่ได้ทำการเปลี่ยนแปลงตามย่อมไม่สามารถเรียกใช้งานเมทอด M1() ได้ เป็นต้น

ค่าความเสถียรเชิงตรรกะของคลาสใดๆ หาได้จาก จำนวนครั้งที่คลาสจะไม่ได้รับผลกระทบเชิงตรรกะ หรือผลกระทบที่ทำให้การทำงานผิดพลาดจากจำนวนการเปลี่ยนแปลงทั้งหมดของซอฟต์แวร์ [7] โดยความเสถียรเชิงตรรกะของคลาสใดๆ สามารถใช้อธิบายโอกาสในการไม่ได้รับผลกระทบเชิงตรรกะที่เกิดจากการเปลี่ยนแปลงซอฟต์แวร์ของคลาสได้ เช่น หากมีคลาสหนึ่งมีความเสถียรเชิงตรรกะเท่ากับ 0.68 หมายความว่าหากเกิดการเปลี่ยนแปลงขึ้นในซอฟต์แวร์ โอกาสที่คลาสนี้จะไม่ได้รับผลกระทบเชิงตรรกะมีค่าเท่ากับ 68% หรือในการเปลี่ยนแปลง 100 ครั้ง คลาสนี้จะไม่ได้รับผลกระทบเชิงตรรกะและยังคงสามารถทำงานได้เหมือนเดิม 68 ครั้ง เป็นต้น

ความเสถียรเชิงประสิทธิภาพคือ โอกาสในการไม่ได้รับผลกระทบที่ส่งผลให้ประสิทธิภาพในการทำงานเปลี่ยนไป ตัวอย่างของผลกระทบจากการเปลี่ยนแปลงที่ส่งผลต่อประสิทธิภาพ เช่น จากรูปที่ 2.1 ให้เดิมคลาส C3 ใช้เวลาในการประมวลผล 0.3 วินาที แต่เมื่อทำการเปลี่ยนแปลงเมทอด M1() ซึ่งเป็นเมทอดที่เมทอด M4() ของคลาส C3 อ้างถึง แล้วส่งผลให้เวลาที่คลาส C3 ใช้ในการประมวลผลเพิ่มขึ้นเป็น 0.6 วินาที เป็นต้น เช่นเดียวกับค่าความเสถียรเชิงตรรกะ ค่าความเสถียรเชิงประสิทธิภาพของคลาสใดๆ ใช้อธิบายโอกาสในการไม่ได้รับผลกระทบเชิงประสิทธิภาพที่เกิดจากการเปลี่ยนแปลงซอฟต์แวร์ซึ่งส่งผลให้ประสิทธิภาพในการทำงานของคลาสนั้นๆ เปลี่ยนแปลงไป อย่างไรก็ตามการหาค่าความเสถียรเชิงประสิทธิภาพมีความซับซ้อนสูง ทั้งนี้เนื่องจากการหาผลกระทบเชิงประสิทธิภาพทำได้ยาก เพราะต้องกำหนดถึงประสิทธิภาพที่สนใจ และวิเคราะห์ถึงการเปลี่ยนแปลงว่าอาจเกิดผลกระทบกับประสิทธิภาพใดบ้าง นอกจากนี้

การหาค่าความเสถียรเชิงประสิทธิภาพต้องอาศัยการวิเคราะห์แบบพลวัต (Dynamic Analysis) ซึ่งเป็นการวิเคราะห์ระหว่างการทำงานจริงของซอฟต์แวร์อีกด้วย



รูปที่ 2.1 ตัวอย่างความสัมพันธ์ระหว่างคลาส

ค่าความเสถียรเชิงตรรกะของคลาสใดๆ คือ ความต้านทานต่อผลกระทบวงคลื่นเชิงตรรกะ โดยค่าผลกระทบวงคลื่นเชิงตรรกะคือ จำนวนครั้งที่คลาสที่สนใจจะได้รับผลกระทบจากการเปลี่ยนแปลงทุกประเภทของทุกๆ คลาสในโปรแกรมหารด้วยจำนวนการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมด ดังสมการ [7]

$$\text{ผลกระทบวงคลื่นเชิงตรรกะของคลาส} = \frac{\text{จำนวนครั้งที่คลาสได้รับผลกระทบจากการเปลี่ยนแปลง}}{\text{จำนวนการเปลี่ยนแปลงทั้งหมด}} \quad (1)$$

ดังนั้นความเสถียรเชิงตรรกะคือ จำนวนครั้งที่คลาสที่สนใจไม่ได้รับผลกระทบจากการเปลี่ยนแปลงทุกประเภทของทุกๆ คลาสในโปรแกรมหารด้วยจำนวนการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมด สามารถคำนวณได้ดังสมการ [7]

$$\text{ความเสถียรเชิงตรรกะของคลาส} = 1 - \text{ผลกระทบวงคลื่นเชิงตรรกะของคลาส} \quad (2)$$

หรือ

$$\text{ความเสถียรเชิงตรรกะของคลาส} = 1 - \frac{\text{จำนวนครั้งที่คลาสได้รับผลกระทบจากการเปลี่ยนแปลง}}{\text{จำนวนการเปลี่ยนแปลงทั้งหมด}} \quad (3)$$

ตัวอย่างในการคำนวณความเสถียรเชิงตรรกะของคลาสใดๆ เป็นดังนี้ สมมติให้ซอฟต์แวร์หนึ่งประกอบด้วยคลาส 5 คลาส C_1 , C_2 , C_3 , C_4 และ C_5 ในการหาค่าความเสถียรเชิงตรรกะของคลาส C_1 ทำได้โดย

- 1) เลือกคลาสที่ละคลาสจากคลาสทั้งหมด
- 2) จำลองการเปลี่ยนแปลงกับคลาสที่เลือกมาถ้าสามารถทำการเปลี่ยนแปลงได้ให้เพิ่มค่าจำนวนการเปลี่ยนแปลงทั้งหมด
- 3) ตรวจสอบว่าคลาส C_1 ได้รับผลกระทบจากการเปลี่ยนแปลงหรือไม่ ถ้าได้รับผลกระทบให้เพิ่มค่าจำนวนครั้งที่ C_1 ได้รับผลกระทบ
- 4) ทำการตรวจสอบจนครบทุกรูปแบบการเปลี่ยนแปลงของคลาสที่เลือกมา โดยสำหรับการเปลี่ยนแปลงในระดับคลาสให้ทำการตรวจสอบที่ละแอททริบิวต์และเมธอด
- 5) ทำการเลือกคลาสที่เหลือขึ้นมาจนครบทุกคลาส

สมมติให้ผลที่ได้จากการตรวจสอบเป็นดังนี้

- จำนวนการเปลี่ยนแปลงทั้งหมดเป็น 257
- จำนวนครั้งที่คลาส C_1 ได้รับผลกระทบเป็น 154

ค่าผลกระทบวงคลื่นเชิงตรรกะของคลาส C_1 มีค่าเท่ากับ 154หารด้วย 257 เท่ากับ 0.5992

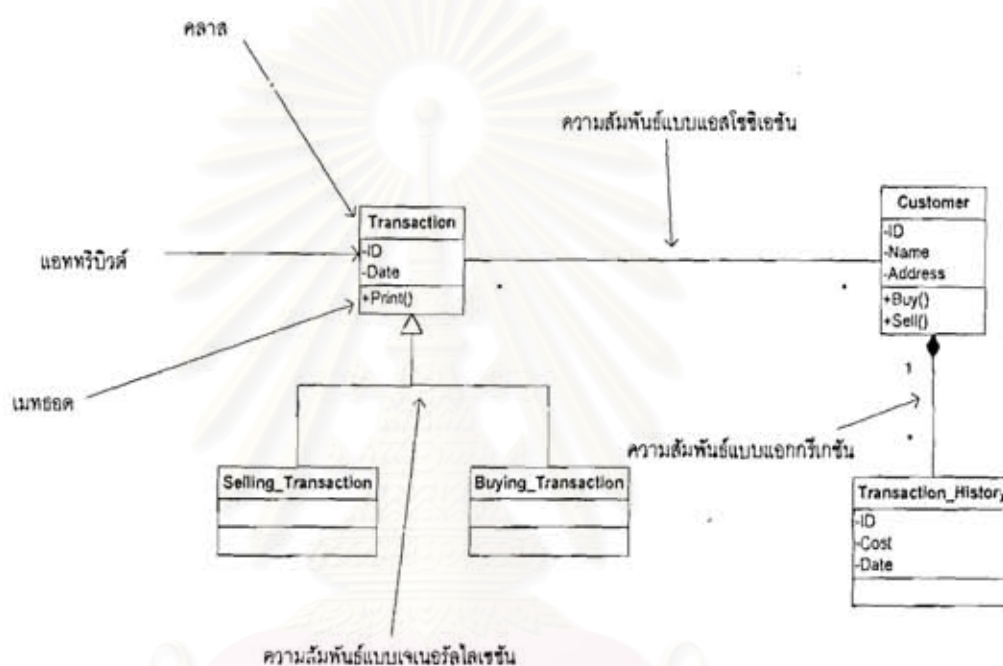
ดังนั้นค่าความเสถียรเชิงตรรกะของคลาส C_1 เท่ากับ $1 - 0.5992$ เท่ากับ 0.4008 ซึ่งหมายความว่า เมื่อมีการเปลี่ยนแปลงขึ้นภายในซอฟต์แวร์คลาส C_1 จะมีโอกาสการไม่ได้รับผลกระทบเท่ากับ 40.08% หรืออาจกล่าวได้ว่าหากมีการเปลี่ยนแปลงกับซอฟต์แวร์เป็นจำนวน 100 ครั้ง คลาส C_1 จะไม่ได้รับผลกระทบ 40 ครั้ง

ในการคำนวณค่าความเสถียรเชิงตรรกะต้องทำการคำนวณค่าผลกระทบวงคลื่นเชิงตรรกะก่อนแล้วจึงทำการคำนวณความเสถียรเชิงตรรกะ ทั้งนี้เนื่องจากการคำนวณหาจำนวนครั้งที่คลาสได้รับผลกระทบสามารถกระทำได้ง่ายกว่า สำหรับอัลกอริทึมในการหาค่าความเสถียรเชิงตรรกะของคลาสมีรายละเอียดในภาคผนวก ก

2.1.3 แผนภาพคลาสและแผนภาพซีควเอนซ์

แผนภาพคลาส [8]ใช้แสดงรายละเอียดของคลาสและความสัมพันธ์ระหว่างคลาสทั้งหมดของซอฟต์แวร์ซึ่งช่วยในการทำความเข้าใจระบบและการออกแบบระบบโดยรวม โดยคลาส

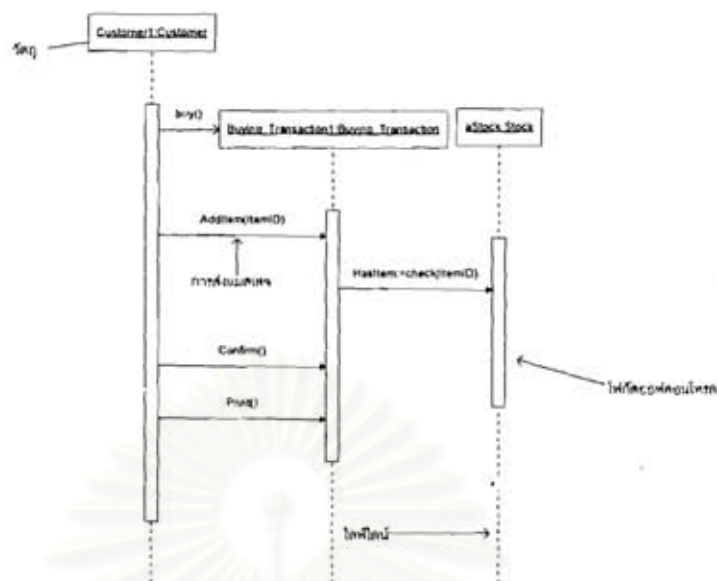
ประกอบด้วย แอททริบิวต์ (attribute) และเมทอด (method) หรือโอเปอเรชัน (operation) และคลาสจะเชื่อมโยงกันด้วยความสัมพันธ์แบบต่างๆ เช่น แอสโซซิเอชัน (association) เป็นการเชื่อมโยงแบบปกติในกรณีที่คลาสมีการติดต่อเรียกใช้ซึ่งกันและกัน แอ็กกรีเกชัน (aggregation) เป็นการเชื่อมโยงของคลาสที่เป็นส่วนหนึ่งของอีกคลาส และเจเนอรัไลเซชัน (generalization) เป็นความสัมพันธ์ระหว่างคลาสซึ่งคลาสหนึ่งเป็นคลาสย่อยของอีกคลาสหนึ่ง หรือเป็นความสัมพันธ์ระหว่างคลาสแม่กับคลาสลูก ตัวอย่างแผนภาพคลาสดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างแผนภาพคลาส [8]

แผนภาพซีควเन्ซ์ [8] แสดงการโต้ตอบกันระหว่างวัตถุ ซึ่งจะแสดงให้เห็นถึงลำดับของการส่งคำร้องขอ (Message) ถึงกันระหว่างวัตถุต่างๆ โดยปกติแผนภาพซีควเन्ซ์ใช้แสดงเค้าโครงเรื่อง (Scenario) หนึ่งๆ ของซอฟต์แวร์ และในการใช้งานจริงแผนภาพซีควเन्ซ์ใช้อธิบายเฉพาะสถานการณ์สำคัญเท่านั้น ทั้งนี้เพราะสำหรับซอฟต์แวร์หนึ่งๆ สามารถมีสถานการณ์ได้มากมาย

แผนภาพซีควเन्ซ์ประกอบไปด้วยองค์ประกอบหลักๆ ดังนี้ วัตถุ เส้นไลฟ์ไลน์ (lifeline) หรือช่วงที่วัตถุมีชีวิตอยู่ระหว่างการทำงาน และ โฟกัสออฟคอนโทรล (focus of control) หรือช่วงของเวลาที่วัตถุทำการรับส่งคำร้องขอตลอดเส้นไลฟ์ไลน์ ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างแผนภาพซีควเอนซ์ [8]

วิทยานิพนธ์นี้ต้องการทำการประมาณค่าความเสถียรเชิงตรรกะโดยรวมของซอฟต์แวร์ จึงต้องทราบข้อมูลโดยภาพรวมของระบบในขณะที่แผนภาพซีควเอนซ์ให้ข้อมูลได้เพียงบางส่วน เท่านั้น ดังนั้นงานวิจัยนี้จึงเน้นการศึกษาแผนภาพคลาสมากกว่าแผนภาพซีควเอนซ์ อย่างไรก็ตามถึงแม้แผนภาพคลาสจะแสดงภาพรวมและรายละเอียดโดยรวมของซอฟต์แวร์ ทว่าหากพิจารณาถึงความละเอียดของข้อมูลจะพบว่ามีความละเอียดน้อย โดยจากแผนภาพคลาสสามารถทราบถึงความสัมพันธ์ระหว่างคลาสแต่ไม่ทราบถึงการเรียกใช้เมธอดหรือแอททริบิวต์ต่างๆ จึงจำเป็นต้องนำแผนภาพซีควเอนซ์มาประกอบเพื่อเพิ่มรายละเอียดและความถูกต้อง

2.1.4 การวิเคราะห์ความถดถอยเชิงซ้อน (Multiple Regression analysis)

การวิเคราะห์ความถดถอยเชิงซ้อน [9] เป็นการศึกษาความสัมพันธ์ระหว่างตัวแปรมากกว่า 2 ตัวขึ้นไป (หากเป็นการศึกษาความสัมพันธ์ระหว่างตัวแปร 2 ตัวจะเรียกว่า การวิเคราะห์ความถดถอยอย่างง่าย (Simple Regression Analysis)) ในการวิเคราะห์ความถดถอยจะมีตัวแปร 2 ประเภทด้วยกัน คือ ตัวแปรอิสระ และตัวแปรตาม

ตัวแปรอิสระคือ ตัวแปรที่ทราบค่าและไม่ขึ้นกับตัวแปรใดๆ

ตัวแปรตามคือ ตัวแปรที่ค่าของตัวแปรเปลี่ยนแปลงตามค่าของตัวแปรอิสระอื่นๆ

วัตถุประสงค์ของการวิเคราะห์ความถดถอยมีดังนี้

- 1) เพื่อศึกษาความสัมพันธ์ระหว่างตัวแปรว่ามีความสัมพันธ์กันมากน้อยเพียงใด
- 2) ใช้ความสัมพันธ์ที่ได้ในการประมาณค่าตัวแปรตาม เมื่อกำหนดค่าตัวแปรอิสระ

รูปแบบสมการความถดถอยเชิงซ้อนเมื่อมีตัวแปรอิสระ k ตัว ($X_1, X_2, \dots, X_k$) ที่มีความสัมพันธ์กับตัวแปรตาม Y โดยมีความสัมพันธ์แบบเชิงเส้นจะมีรูปแบบดังนี้

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k$$

โดย Y คือตัวแปรตาม

X_i คือตัวแปรอิสระ เมื่อ $i = 1, 2, \dots, k$

β_0 คือส่วนตัดแกน Y เมื่อกำหนดให้ $X_1 = X_2 = \dots = X_k = 0$

β_i คือสัมประสิทธิ์ความถดถอยเชิงส่วน (Partial regression coefficient) เมื่อ $i = 1, 2, \dots, k$

และค่าประมาณของ Y คือ

$$\hat{Y} = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_k x_k + e$$

โดย

$\hat{Y}$ คือค่าประมาณของ Y

x_i คือตัวแปรอิสระของสมการความถดถอย เมื่อ $i = 1, 2, \dots, k$

b_0 คือค่าคงที่

b_i คือค่าประมาณของสัมประสิทธิ์ความถดถอยเชิงส่วน เมื่อ $i = 1, 2, \dots, k$

e คือความคลาดเคลื่อนอย่างสุ่ม (random error)

ความหมายของสัมประสิทธิ์ความถดถอยเชิงส่วน คือ ค่าสัมประสิทธิ์ที่แสดงถึงระดับความสัมพันธ์ของตัวแปรอิสระที่มีต่อตัวแปรตาม เช่น b , เป็นค่าซึ่งแสดงความสัมพันธ์ระหว่าง Y และ x , หมายถึงถ้าตัวแปร x , เพิ่มขึ้น 1 หน่วยจะทำให้ Y เปลี่ยนแปลง b , หน่วย

การวิเคราะห์ความถดถอยมีขั้นตอนดังนี้

1. ทำการประมาณค่าสัมประสิทธิ์ความถดถอยเชิงส่วนโดยใช้วิธีกำลังสองน้อยที่สุด เมื่อได้ค่าประมาณสัมประสิทธิ์ความถดถอยเชิงส่วนจะสามารถสร้างสมการถดถอยได้
2. ทดสอบสมการความถดถอยโดยใช้การวิเคราะห์ความแปรปรวนจำแนกแบบทางเดียว เพื่อทดสอบว่าตัวแปรอิสระ X มีความสัมพันธ์กับตัวแปรตาม Y จริงหรือไม่ โดยจะทดสอบว่า $\beta_1 = \beta_2 = \dots = \beta_n = 0$ หรือไม่ ถ้าไม่เท่ากับศูนย์หมายความว่า X_i อย่างน้อยหนึ่งตัวมีความสัมพันธ์กับ Y ในรูปแบบเชิงเส้นจริง
3. เมื่อทดสอบสมการความถดถอยแล้วพบว่า X_i อย่างน้อยหนึ่งตัวมีความสัมพันธ์กับ Y ในรูปแบบเชิงเส้นจริง จะต้องทดสอบเพื่อหาความสัมพันธ์ระหว่าง X_i แต่ละตัวกับ Y ว่ามีความสัมพันธ์กันจริงหรือไม่ โดยทดสอบเพื่อดูว่ามีค่า β_i ไบ้างที่เป็น 0 และถ้า β_i ไบ้ที่เป็น 0 แสดงว่าตัวแปร X_i นั้นไม่ได้มีความสัมพันธ์กับ Y สามารถตัดออกจากสมการได้
4. เมื่อได้สมการประมาณค่า Y แล้วจึงทำการหาค่าสัมประสิทธิ์การตัดสินใจเชิงซ้อน (Multiple Coefficient of Determination: R^2) ซึ่งเป็นค่าสัดส่วนที่ตัวแปรอิสระ ($X_1, X_2, \dots, X_n$) สามารถอธิบายการเปลี่ยนแปลง Y ได้ หรือเป็นเปอร์เซ็นต์ความผันแปร Y ที่มีสาเหตุเนื่องจากการผันแปรของ $X_1, X_2, \dots$ และ X_n และจากค่าสัมประสิทธิ์การตัดสินใจเชิงซ้อนจะสามารถคำนวณหาค่าสัมประสิทธิ์สหสัมพันธ์เชิงซ้อน (Multiple Coefficient of Correlation: R) ได้โดยการถอดรากที่ 2 สัมประสิทธิ์การตัดสินใจเชิงซ้อน สัมประสิทธิ์สหสัมพันธ์เชิงซ้อนแสดงถึงความสัมพันธ์ระหว่าง Y กับ $X_1, X_2, \dots$ และ X_n โดยถ้า R มีค่าเข้าใกล้ 0 แสดงว่า Y มีความสัมพันธ์กับ $X_1, X_2, \dots, X_n$ น้อยมาก แต่ถ้า R เข้าใกล้ 1 แสดงว่า Y มีความสัมพันธ์กับ $X_1, X_2, \dots, X_n$ มาก

การวิเคราะห์ความถดถอยเชิงซ้อนเป็นการศึกษาความสัมพันธ์ระหว่างตัวแปรตาม 1 ตัว และตัวแปรอิสระหลายตัวนั้น มีข้อกำหนดว่าตัวแปรอิสระเหล่านั้นต้องไม่มีความสัมพันธ์กัน เนื่องจากถ้าตัวแปรอิสระมีความสัมพันธ์ซึ่งกันและกันจะทำให้เกิดปัญหา Multicollinearity ซึ่งจะทำให้เกิดปัญหาดังนี้

1. ทำให้ค่าเบี่ยงเบนมาตรฐานของสัมประสิทธิ์ความถดถอยมีค่าสูงมาก
2. ทำให้เครื่องหมายของสัมประสิทธิ์ความถดถอยตรงกันข้ามกับที่ควรจะเป็น
3. ทำให้ค่าสัมประสิทธิ์ความถดถอยไม่คงที่เมื่อมีตัวแปรอิสระเพิ่มขึ้น

การเกิดปัญหาเหล่านี้ในท้ายที่สุดแล้วจะทำให้โมเดลที่ได้ไม่ถูกต้อง ดังนั้นจึงต้องแก้ไขด้วยการใช้วิธีการเลือกตัวแปรอิสระแบบสเตปไวส์ซึ่งเป็นการเลือกตัวแปรอิสระเข้าสู่สมการความถดถอยครั้งละ 1 ตัว ถ้าตัวแปรอิสระที่นำเข้ามามีความสัมพันธ์กับตัวแปรอิสระที่มีอยู่แล้ว จะทำการตัดตัวแปรอิสระตัวใดตัวหนึ่งทิ้งไป ซึ่งขั้นตอนของวิธีการเลือกตัวแปรอิสระแบบสเตปไวส์เป็นดังนี้

1. ทำการเลือกตัวแปรอิสระ 1 ตัวที่มีความสัมพันธ์กับตัวแปรตามอย่างมีนัยสำคัญที่สุดเข้าสู่สมการ โดยดูจากค่าสัมประสิทธิ์สหสัมพันธ์เชิงส่วน (Partial correlation) ซึ่งแสดงระดับความสัมพันธ์ระหว่างตัวแปรตาม และตัวแปรอิสระแต่ละตัว
2. จากตัวแปรอิสระที่เหลือ $k-1$ เลือกตัวแปรอิสระที่มีความสัมพันธ์กับตัวแปรตามอย่างมีนัยสำคัญที่สุดเข้าสู่สมการ จากนั้นทำการทดสอบความสัมพันธ์ระหว่างตัวแปรตามกับตัวแปรอิสระใหม่ และอาจจะทำการตัดตัวแปรอิสระตัวใดตัวหนึ่งออกไปถ้าตัวแปรอิสระนั้นมีความสัมพันธ์กับตัวแปรอิสระอื่นๆ
3. ทำซ้ำขั้นตอนที่ 2 จนกระทั่งไม่สามารถนำตัวแปรอิสระเข้าสู่สมการ หรือไม่มีตัวแปรอิสระที่สามารถตัดออกได้อีก

2.1.5 การทำนายที่ระดับแอล (Prediction at Level L)

การทำนายที่ระดับแอล [10] เป็นวิธีที่ใช้ในการประเมินผลโมเดลการประมาณค่า เพื่อประเมินความคลาดเคลื่อนในการประมาณค่าของโมเดลเมื่อเทียบกับค่าจริง โดยจะทำการคำนวณหาร้อยละความคลาดเคลื่อนในการประมาณค่าของโมเดลเทียบกับค่าจริง การทำนายที่ระดับแอลจะอาศัยขนาดความผิดพลาดสัมพัทธ์ซึ่งมีสมการดังนี้

$$MRE = |RE| = \left| \frac{E - \hat{E}}{E} \right|$$

โดย

MRE คือ ขนาดความผิดพลาดสัมพัทธ์ (Magnitude relative error)

RE คือ ความผิดพลาดสัมพัทธ์ (Relative error)

E คือ ค่าความเสถียรเชิงตรรกะของคลาส

$\hat{E}$ คือ ค่าประมาณค่าความเสถียรเชิงตรรกะของคลาสที่ได้จากโมเดล

การทำนายที่ระดับแอลมีสมการดังนี้

$$PRED(l) = \frac{k}{n}$$

โดย

$PRED(l)$ คือ การทำนายที่ระดับแอล

l คือ ช่วงระดับของความคลาดเคลื่อน มีค่าตั้งแต่ 0 -1

k คือ จำนวนหน่วยตัวอย่างที่มีค่า $MRE \leq l$

n คือ จำนวนหน่วยตัวอย่างทั้งหมด

โดยผลการทำนายที่ระดับแอลสามารถอธิบายได้ดังนี้ ถ้า $PRED(0.2)=0.85$ หมายความว่า 85% ของค่าที่ประมาณได้จะมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 20% และการนำผลการทำนายระดับแอลมาใช้ในการประเมินผลโมเดลสำหรับประมาณค่านั้น โดยทั่วไปจะยอมรับโมเดลสำหรับประมาณค่าที่ผลการทำนายระดับแอลที่ระดับ 0.25 มากกว่าหรือเท่ากับ 0.75 [10] หรือค่าประมาณที่ได้จะต้องคลาดเคลื่อนจากค่าจริงไม่เกิน 25% เป็นจำนวน 75% ขึ้นไป

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Investigation of Metrics for Object-Oriented Design Logical Stability [7] โดย Mahmoud O. Elish , David Rine

งานวิจัยนี้ได้ทำการศึกษาถึงความสัมพันธ์ระหว่างค่าความเสถียรเชิงตรรกะของคลาสกับ ตัววัดของซอฟต์แวร์เชิงวัตถุของ Chidamber และ Kemerer ทั้ง 6 ตัว ได้แก่

- Weighted Methods per Class (WMC)
- Depth of Inheritance Tree (DIT)
- Coupling Between Object classes (CBO)
- Response For a Class (RFC)
- Lack Of Cohesion in Methods (LCOM)
- Number Of Children (NOC)

โดยค่าตัววัดทั้ง 6 มาจากซอร์สโค้ด (Source Code) ของซอฟต์แวร์เช่นเดียวกับค่าความเสถียรเชิงตรรกะ

งานวิจัยนี้ได้ได้นิยามของความเสถียรว่าเป็นค่าความต้านทานต่อการกระจายระหว่างคลาสเมื่อเกิดการเปลี่ยนแปลง โดยแบ่งความเสถียรออกเป็น 2 แบบ คือ ความเสถียรเชิงตรรกะและความเสถียรเชิงประสิทธิภาพ ซึ่งงานวิจัยนี้มุ่งเน้นที่ความเสถียรเชิงตรรกะโดยอาศัยการศึกษาถึงการเปลี่ยนแปลงในระดับคลาสที่เป็นไปได้ว่ามีผลกระทบต่อคลาสอื่นในรูปแบบใด และทำการคำนวณค่าความเสถียรเชิงตรรกะจากจำนวนครั้งที่คลาสที่สนใจได้รับผลกระทบกับจำนวนครั้งที่เกิดการเปลี่ยนแปลงทั้งหมด

งานวิจัยนี้ได้ทำการเลือกโปรแกรมเพื่อทำการศึกษามา 3 โปรแกรมด้วยกัน ได้แก่ Jxplorer เวอร์ชัน 2.1.001 JE(Just an Editor) เวอร์ชัน 1.65 และ Phex เวอร์ชัน 0.7.3 ซึ่งทั้ง 3 โปรแกรมเป็นโปรแกรมโอเพนซอร์ส (Open Source) ที่พัฒนาโดยภาษาจาวา งานวิจัยนี้คำนวณค่าความเสถียรเชิงตรรกะตามวิธีการที่ได้สร้างขึ้นและหาค่าตัววัดทั้ง 6 ของทั้ง 3 โปรแกรม แล้วนำมาคำนวณค่าความสัมพันธ์แล้วทำการสรุป

ผลสรุปของงานวิจัยนี้พบว่าตัววัด 5 ตัวมีความสัมพันธ์เชิงลบกับค่าความเสถียรเชิงตรรกะของคลาส ได้แก่ WMC , DIT , CBO , RFC , LCOM โดย CBO และ RFC มีความสัมพันธ์สูงสามารถนำไปใช้เป็นตัวบ่งชี้ได้ แต่ไม่พบความสัมพันธ์ระหว่างตัววัด NOC กับค่าความเสถียรเชิงตรรกะของคลาส อย่างไรก็ตามงานวิจัยนี้ทำการศึกษาความสัมพันธ์จากซอร์สโค้ดซึ่งจำเป็นต้องรอจนโปรแกรมพัฒนาเสร็จ

2.2.2 Ripple effect analysis for Software Maintenance [5] โดย S. S. Yau, J. S. Collofello, T. MacGregor

งานวิจัยนี้ทำการวิเคราะห์และเสนอแนะแนวทางในการวิเคราะห์ผลกระทบวงคลื่นที่อาจเกิดขึ้นในกระบวนการบำรุงรักษา โดยแบ่งผลกระทบวงคลื่นออกเป็น 2 แบบด้วยกันดังนี้

1. ผลกระทบวงคลื่นเชิงตรรกะ (Logical Ripple Effect) คือ ผลกระทบที่เกิดจากการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของซอฟต์แวร์แล้วทำให้เกิดความขัดแย้งในการทำงานของส่วนอื่นๆ เช่นการเปลี่ยนชนิดของตัวแปร

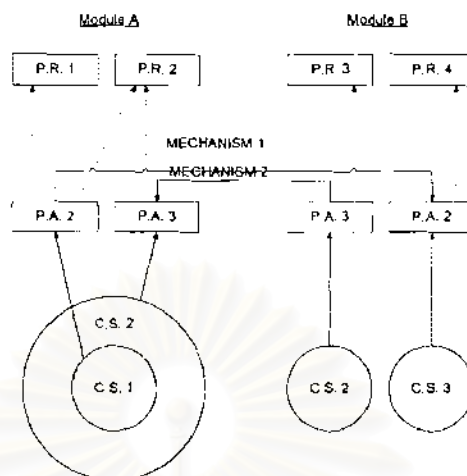
2. ผลกระทบวงคลื่นเชิงประสิทธิภาพ (Performance Ripple Effect) คือ ผลกระทบที่เกิดจากการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของซอฟต์แวร์แล้วส่งผลกระทบต่อประสิทธิภาพของส่วนอื่นๆ ลดลง

ผลกระทบวงคลื่นเชิงตรรกะคือ การเกิดผลกระทบวงคลื่นต่อโปรแกรมในด้านการทำงานของโปรแกรม กล่าวคือเมื่อมีการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของโปรแกรมแล้วทำให้ส่วนอื่นทำงานผิดพลาดไปจากเดิม หรือไม่สามารถทำงานได้ เช่น สมมติให้ส่วนย่อย A มีการประกาศตัวแปร X และส่วนย่อย B มีการเรียกใช้ตัวแปร X จาก A เมื่อมีการแก้ไขโดยเปลี่ยนประเภทของตัวแปร X แล้วทำให้ส่วนย่อย B ที่เรียกใช้ตัวแปร X ไม่สามารถทำงานได้ เป็นต้น

ผลกระทบวงคลื่นเชิงประสิทธิภาพ คือการเกิดผลกระทบวงคลื่นต่อโปรแกรมโดยมีผลกระทบต่อประสิทธิภาพ กล่าวคือเมื่อมีการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของโปรแกรมแล้วส่งผลกระทบต่อประสิทธิภาพของส่วนอื่นของโปรแกรมลดลง เช่น สมมติให้ส่วนย่อย A มีการเปลี่ยนแปลงแล้วส่วนย่อย B จากเดิมสามารถประมวลผลเสร็จสิ้นในเวลา 1 วินาที ใช้เวลา 2 วินาทีในการประมวลผล เป็นต้น

ในงานวิจัยนี้ได้ทำการเสนอแนวทางในการวิเคราะห์ผลกระทบวงคลื่น สำหรับผลกระทบวงคลื่นเชิงตรรกะทำการวิเคราะห์โปรแกรมแล้วกำหนดแหล่งกำเนิดข้อผิดพลาดหลัก (Primary Error Source) ซึ่งคือจุดที่ทำการแก้ไขแล้วมีโอกาสทำให้เกิดความไม่สอดคล้องกันต่อส่วนอื่นๆ จากนั้นทำการหาแหล่งกำเนิดข้อผิดพลาดที่สอง (Secondary Error Source) ซึ่งคือจุดที่ได้รับผลกระทบจากการเปลี่ยนแปลงของแหล่งกำเนิดข้อผิดพลาดหลัก สำหรับการวิเคราะห์ผลกระทบวงคลื่นเชิงประสิทธิภาพทำได้โดยทำการวิเคราะห์ความต้องการเชิงประสิทธิภาพ (Performance Requirement หรือ P.R.) เพื่อสร้างเป็นคุณสมบัติทางประสิทธิภาพ (Performance Attribute หรือ P.A.) ซึ่งคุณสมบัติทางประสิทธิภาพเหล่านี้จะได้รับผลกระทบเมื่อมีการเปลี่ยนแปลงส่วนของโปรแกรม ซึ่งจะส่งผลกระทบต่อความต้องการเชิงประสิทธิภาพต่อไป ส่วนของโปรแกรมที่เปลี่ยนแปลงแล้วส่งผลกระทบต่อคุณสมบัติทางประสิทธิภาพเรียกว่า ส่วนวิกฤต (Critical section หรือ C.S.)

นอกจากนี้ผลกระทบที่เกิดขึ้นต่อคุณสมบัติทางประสิทธิภาพหนึ่งยังส่งผลต่อคุณสมบัติทางประสิทธิภาพอื่นผ่านทางกลไกการทำงาน (Mechanism) ดังรูปที่ 2.4



รูปที่ 2.4 การส่งผลกระทบต่อกันของการเกิดผลกระทบวงคลื่นเชิงประสิทธิภาพ [5]

งานวิจัยนี้ได้เสนอแนะวิธีการในการทำนายความซับซ้อนของการเปลี่ยนแปลงซึ่งจะช่วยให้การวิเคราะห์ถึงภาระงานของโปรแกรมเมอร์ที่ต้องการ อย่างไรก็ตามวิธีการทำนายที่สร้างขึ้นไม่ได้บอกวิธีการวัดอย่างชัดเจน

โดยสรุปงานวิจัยนี้ได้ทำการวิเคราะห์และเสนอแนะวิธีการสำหรับวิเคราะห์ผลกระทบวงคลื่นที่อาจเกิดขึ้นโดยผลลัพธ์ที่ได้เป็นเพียงกลุ่มของส่วนโปรแกรมที่จะได้รับผลกระทบเท่านั้นไม่ได้เป็นค่าตัวเลขที่สามารถนำไปคำนวณได้ นอกจากนี้วิธีการวิเคราะห์ของงานวิจัยนี้เป็นวิธีการสำหรับโปรแกรมเชิงกระบวนการคำสั่ง (Procedural Programming) เท่านั้น แต่ในวิทยานิพนธ์นี้จะมุ่งเน้นที่โปรแกรมเชิงวัตถุซึ่งเป็นที่แพร่หลายในปัจจุบัน

2.2.3 Some stability measures for software maintenance [3] โดย S. Yau , J. Collofello , T. MacGregor

งานวิจัยนี้ทำการศึกษาและเสนอแนะวิธีการในการวัดความเสถียรเชิงตรรกะของโปรแกรมเชิงกระบวนการคำสั่ง โดยให้นิยามความเสถียรเป็นค่าความต้านทานของผลกระทบวงคลื่นที่อาจเกิดขึ้นเมื่อโปรแกรมมีการเปลี่ยนแปลง โดยทำการวิเคราะห์และเสนอแนะสมการและอัลกอริทึมสำหรับคำนวณความเสถียรเชิงตรรกะของโปรแกรม

งานวิจัยนี้ได้มองความเสถียรของโปรแกรมเป็นคุณสมบัติหนึ่งของโปรแกรมที่ส่งผลกระทบต่อกระบวนการบำรุงรักษา โดยโปรแกรมที่มีค่าความเสถียรสูงจะช่วยให้กระบวนการบำรุงรักษาทำ

ได้อย่างง่ายและลดค่าใช้จ่ายที่เกิดขึ้น ทั้งนี้เนื่องจากถ้าความเสถียรมีค่าต่ำผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงจะมีความซับซ้อนสูงทำให้แก้ไขได้ยาก งานวิจัยนี้ได้แบ่งความเสถียรออกเป็น 2 แบบ คือ ความเสถียรเชิงตรรกะและความเสถียรเชิงประสิทธิภาพ โดยมุ่งเน้นที่การคำนวณความเสถียรเชิงตรรกะเท่านั้น

ในงานวิจัยนี้ได้แบ่งการคำนวณค่าความเสถียรเชิงตรรกะออกเป็น 2 ส่วนคือ การหาความเสถียรเชิงตรรกะของโมดูล และการหาความเสถียรเชิงตรรกะของโปรแกรม โดยความเสถียรเชิงตรรกะของโมดูลคือค่าความต้านทานต่อผลกระทบการเปลี่ยนแปลงที่อาจเกิดขึ้นเมื่อทำการเปลี่ยนแปลงโมดูลหนึ่งต่อโมดูลอื่นๆ ในโปรแกรมในแง่ของการทำงานทางตรรกะของโปรแกรม โดยอาศัยการเปลี่ยนแปลงอย่างง่ายและเล็กที่สุดคือการเปลี่ยนแปลงของตัวแปรในการคำนวณ

เมื่อมีการเปลี่ยนแปลงตัวแปรในโมดูลหนึ่งจะส่งผลกระทบต่อตัวแปรอื่นในโมดูลเดียวกัน ปรากฏการณ์นี้เรียกว่า การไหลของข้อผิดพลาดภายในโมดูล (Intra module error flow) และเช่นเดียวกันการเปลี่ยนแปลงนั้นจะส่งผลกระทบต่อตัวแปรของโมดูลอื่นๆ ด้วยเรียกว่าการไหลของข้อผิดพลาดระหว่างโมดูล (Inter module error flow)

วิธีการหาค่าความเสถียรเชิงตรรกะของโมดูลของงานวิจัยนี้ทำได้โดย

- 1) หาตัวแปรส่วนติดต่อภายนอกที่ได้รับผลกระทบจากการเปลี่ยนแปลงตัวแปรภายในโมดูลหนึ่งๆ ซึ่งคือการวิเคราะห์การไหลของข้อผิดพลาดภายในโมดูลที่เกิดจากการเปลี่ยนแปลงตัวแปรใดๆ
- 2) หาตัวแปรของโมดูลอื่นๆ ที่ได้รับผลกระทบที่เกิดขึ้นต่อตัวแปรส่วนติดต่อภายนอก ซึ่งคือการวิเคราะห์การไหลของข้อผิดพลาดระหว่างโมดูล
- 3) หาโมดูลทั้งหมดที่เกี่ยวข้องต่อการไหลของข้อผิดพลาดระหว่างโมดูล
- 4) คำนวณความซับซ้อนของการแก้ไขโมดูลนั้นๆ จากการเปลี่ยนแปลงตัวแปรใดๆ โดยใช้วิธีการคำนวณความซับซ้อนของแมคเคบ (McCabe's complexity) เป็นตัววัดความซับซ้อนในการแก้ไข
- 5) คำนวณค่าผลกระทบวงคลื่นเชิงตรรกะโดยอาศัยผลรวมของที่น่าจะเป็นของการเกิดการเปลี่ยนแปลงตัวแปรใดๆ คูณกับความซับซ้อนของการแก้ไขโมดูลจากการเปลี่ยนแปลงตัวแปรนั้นๆ
- 6) คำนวณค่าความเสถียรเชิงตรรกะของโมดูลโดยให้เป็นส่วนกลับของค่าผลกระทบวงคลื่น

และจากค่าความเสถียรเชิงตรรกะของแต่ละโมดูลสามารถนำไปคำนวณค่าความเสถียรเชิงตรรกะของโปรแกรมได้ดังนี้

- 1) หาค่าผลกระทบวงคลื่นเชิงตรรกะของโปรแกรมโดยอาศัยผลรวมของผลคูณระหว่างความน่าจะเป็นที่จะเกิดการแก้ไขโมดูลใดๆ กับค่าผลกระทบวงคลื่นของโมดูลนั้นๆ
- 2) คำนวณค่าความเสถียรเชิงตรรกะของโปรแกรมโดยให้เป็นส่วนกลับของค่าผลกระทบวงคลื่นของโปรแกรม

ผลสรุปของงานวิจัยนี้คือ วิธีการและกระบวนการในการวัดค่าความเสถียรเชิงตรรกะของโปรแกรม ที่สามารถวัดค่าเป็นตัวเลขเพื่อนำมาใช้งานได้ อย่างไรก็ตามวิธีการของงานวิจัยนี้เป็นวิธีการสำหรับโปรแกรมเชิงกระบวนการคำสั่งและมีความซับซ้อนค่อนข้างสูงในการวัด ทำให้เมื่อนำไปใช้กับระบบที่มีขนาดใหญ่จะทำให้เสียเวลามากและอาจเกิดข้อผิดพลาดได้



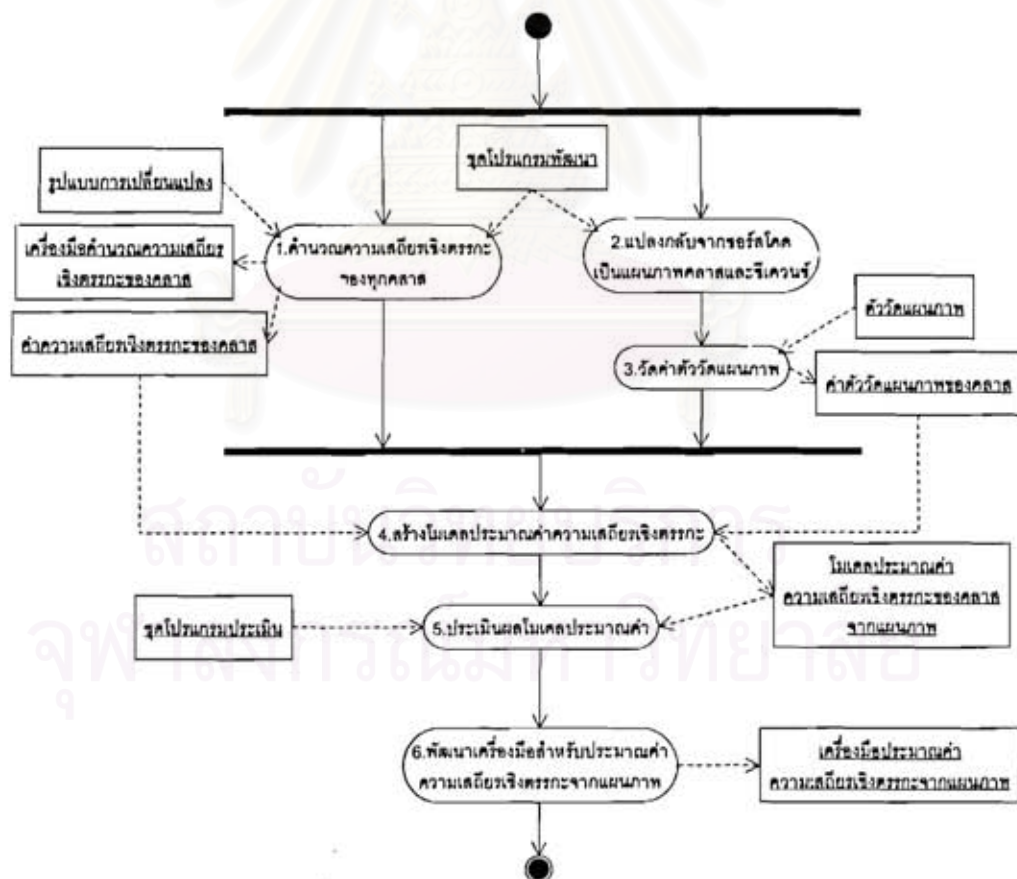
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การสร้างโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะ

จากการศึกษางานวิจัยต่างๆ พบว่า การคำนวณความเสถียรเชิงตรรกะในงานวิจัยส่วนใหญ่จะทำการคำนวณจากซอร์สโคด ซึ่งทำให้การแก้ไขจะต้องใช้เวลาและกำลังงานมากกว่าการแก้ไขการออกแบบ ดังนั้นผู้วิจัยจึงมีแนวคิดที่จะทำการสร้างโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของคลาสจากแผนภาพคลาสและแผนภาพซีควেনซ์ ซึ่งจะช่วยให้สามารถทำการประมาณค่าความเสถียรเชิงตรรกะได้จากแบบจำลองที่ใช้ในการออกแบบซอฟต์แวร์ และจะช่วยลดเวลาที่ใช้ในการแก้ไขเมื่อเทียบกับการแก้ไขที่ซอร์สโคดอีกด้วย

ในบทนี้จะกล่าวถึงการสร้างโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของคลาสจากแผนภาพคลาสและแผนภาพซีควেনซ์ แผนภาพกิจกรรมในรูปที่ 3.1 แสดงขั้นตอนการดำเนินงานเพื่อสร้างโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะ



รูปที่ 3.1 ขั้นตอนการดำเนินงานวิจัย

ขั้นตอนในการดำเนินงานเพื่อสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพ มีด้วยกัน 6 ขั้นตอน โดยเริ่มจากการคำนวณความเสถียรเชิงตรรกะของคลาสจากซอร์สโคดของชุดโปรแกรมทดสอบซึ่งมีรายละเอียดในหัวข้อ 3.1 ในขณะเดียวกันก็นำซอร์สโคดของชุดโปรแกรมทดสอบแปลงกลับเป็นแผนภาพคลาสและแผนภาพซีเคว็นซ์ดังรายละเอียดในหัวข้อ 3.2 แล้วจึงทำการวัดค่าตัววัดแผนภาพจากแผนภาพที่ได้จากการแปลงกลับ ในหัวข้อที่ 3.3 เมื่อได้ค่าความเสถียรเชิงตรรกะของคลาสจากซอร์สโคด และค่าตัววัดแผนภาพแล้วจึงทำการสร้างโมเดลเพื่อประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพ โดยอาศัยข้อมูลทั้ง 2 โดยรายละเอียดอยู่ในบทที่ 4 จากนั้นทำการประเมินผลโมเดลที่ได้เพื่อหาค่าความคลาดเคลื่อนของโมเดลในการประมาณค่า ซึ่งมีรายละเอียดในบทที่ 4 ขั้นตอนสุดท้ายทำการพัฒนาเครื่องมือเพื่อประมาณค่าความเสถียรเชิงตรรกะโดยมีรายละเอียดในบทที่ 5

3.1 คำนวณความเสถียรเชิงตรรกะของคลาสของชุดโปรแกรมพัฒนา

ความเสถียรเชิงตรรกะของแต่ละคลาส คือค่าความต้านทานการได้รับผลกระทบจากการเปลี่ยนแปลงที่คลาส ใดๆ ของโปรแกรม โดยค่าความเสถียรเชิงตรรกะของแต่ละคลาสสามารถคำนวณได้จากการนำรูปแบบการเปลี่ยนแปลงที่กำหนดไว้มาจำลองการเปลี่ยนแปลงกับโปรแกรม โดยดำเนินการจำลองที่ละคลาสและที่ละการเปลี่ยนแปลง จากนั้นทำการเก็บข้อมูลจำนวนครั้งการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมด พร้อมทั้งจำนวนครั้งที่คลาสหนึ่งๆ จะได้รับผลกระทบตามเงื่อนไขของรูปแบบการเปลี่ยนแปลง

รูปแบบการเปลี่ยนแปลงที่ใช้ในวิทยานิพนธ์นี้ ได้ทำการรวบรวมและดัดแปลงมาจากงานวิจัยอื่นๆ [2][7][11][12] โดยรวบรวมมาเฉพาะรูปแบบการเปลี่ยนแปลงที่ส่งผลกระทบเชิงตรรกะซึ่งจะทำให้โปรแกรมเกิดการผิดพลาด ไม่สามารถทำงานได้หรือทำงานผิดไปจากเดิม โดยรูปแบบการเปลี่ยนแปลงที่รวบรวมมาสามารถแบ่งออกเป็น 2 ระดับด้วยกัน คือการเปลี่ยนแปลงในระดับของระบบ (System-level Change) และการเปลี่ยนแปลงในระดับของคลาส (Class-level Change) การเปลี่ยนแปลงในระดับของระบบได้แก่ การเพิ่ม ลบ คลาส และการเพิ่ม ลบ ความสัมพันธ์ต่างๆ ส่วนการเปลี่ยนแปลงในระดับคลาสแบ่งออกเป็น การเปลี่ยนแปลงที่เกิดแก่แอตทริบิวต์และ เมธอดของคลาส เช่น การเพิ่ม ลบ แอตทริบิวต์ หรือ เมธอด การแก้ไขประเภทแอตทริบิวต์ และการแก้ไขขอบเขต เป็นต้น โดยรายละเอียดของรูปแบบการเปลี่ยนแปลงทั้งหมด รวมทั้งคลาสที่จะได้รับผลกระทบจากแต่ละการเปลี่ยนแปลงแสดงอยู่ในตารางที่ 3.1

การเปลี่ยนแปลงในระดับของระบบเป็นการเปลี่ยนแปลงที่เกี่ยวข้องกับความสัมพันธ์ของคลาส และภาพรวมของระบบ ซึ่งเป็นการเปลี่ยนแปลงที่สามารถกระทำได้ที่ซอร์สโคด และแบบจำลองในการออกแบบซอฟต์แวร์ ทั้งนี้สาเหตุในการเลือกรูปแบบการเปลี่ยนแปลงในระดับของระบบมาใช้ในวิทยานิพนธ์นี้ เนื่องจากวิทยานิพนธ์นี้มีวัตถุประสงค์ในการสร้างโมเดลเพื่อประมาณค่าความเสถียรเชิงตรรกะจากแบบจำลองการออกแบบซอฟต์แวร์ ซึ่งแสดงถึงภาพรวมและรายละเอียดของซอฟต์แวร์ ทว่ารูปแบบการเปลี่ยนแปลงในระดับของคลาสไม่เพียงพอที่จะสะท้อนภาพรวมของระบบ จึงต้องนำรูปแบบการเปลี่ยนแปลงในระดับของระบบมาใช้ในการคำนวณความเสถียรเชิงตรรกะด้วย

ทั้งนี้โอกาสในการเกิดการเปลี่ยนแปลงแต่ละรูปแบบมีค่าไม่เท่ากัน อย่างไรก็ตามเนื่องจากวิทยานิพนธ์นี้มีวัตถุประสงค์ในการสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะเพื่อนำไปใช้ในช่วงการออกแบบซอฟต์แวร์ซึ่งไม่สามารถทราบได้ว่าจะเกิดการเปลี่ยนแปลงใดได้บ้าง จึงให้โอกาสในการเกิดการเปลี่ยนแปลงแต่ละรูปแบบเท่ากันหมด

ตารางที่ 3.1 รูปแบบการเปลี่ยนแปลง

หมายเลข	รูปแบบการเปลี่ยนแปลง	คลาสที่อาจได้รับผลกระทบ
การเปลี่ยนแปลงระดับระบบ		
1	เพิ่มความสัมพันธ์คลาสแม่ของ C	C,S
2	ลบความสัมพันธ์คลาสแม่ของ C	C,S,R
3	เพิ่มความสัมพันธ์คลาสลูกของ C	
4	ลบความสัมพันธ์คลาสลูกของ C	
5	เพิ่มความสัมพันธ์แอกรีกเกตคลาสของ C	C,S
6	ลบความสัมพันธ์แอกรีกเกตคลาสของ C	C,S,R
7	เพิ่มความสัมพันธ์คลาส C เป็นแอกรีกเกตคลาส	
8	ลบความสัมพันธ์คลาส C ที่เป็นแอกรีกเกตชั้นคลาส	
9	เพิ่มความสัมพันธ์แอสโซซิเอชันคลาสของ C	C,S
10	ลบความสัมพันธ์แอสโซซิเอชันคลาสของ C	C,S,R

ตารางที่ 3.1 รูปแบบการเปลี่ยนแปลง (ต่อ)

หมายเลข	รูปแบบการเปลี่ยนแปลง	คลาสที่อาจได้รับผลกระทบ
12	ลบความสัมพันธ์คลาส C เป็นแอตทริบิวต์ คลาส	
13	เพิ่มคลาส C	
14	ลบคลาส C	C,S,R
การเปลี่ยนแปลงระดับคลาส		
	แอตทริบิวต์ A ของ C	
15	เพิ่มแอตทริบิวต์	S (เมื่อขอบเขตเป็นพบบลิต เท่านั้น)
16	ลบแอตทริบิวต์	
	ขอบเขตเป็นพบบลิต	C,X,S
	ขอบเขตเป็นโพเทคต์	C,S
	ขอบเขตเป็นโพเวด	C
17	เปลี่ยนค่าของแอตทริบิวต์	
	ขอบเขตเป็นพบบลิต	C,X,S
	ขอบเขตเป็นโพเทคต์	C,S
	ขอบเขตเป็นโพเวด	C
18	เปลี่ยนขอบเขตการเข้าถึง	
	พบบลิตเป็นโพเวด	S,X
	พบบลิตเป็นโพเทคต์	X
	โพเทคต์เป็นพบบลิต	
	โพเทคต์เป็นโพเวด	S
	โพเวดเป็นโพเทคต์	

ตารางที่ 3.1 รูปแบบการเปลี่ยนแปลง (ต่อ)

หมายเลข	รูปแบบการเปลี่ยนแปลง	คลาสที่อาจได้รับผลกระทบ
19	เปลี่ยนประเภทแอททริบิวต์	
	ขอบเขตเป็นพบบิลด์	C,X,S
	ขอบเขตเป็นโพรเทคต์	C,S
	ขอบเขตเป็นไพรเวต	C
20	เปลี่ยนชื่อแอททริบิวต์	
	ขอบเขตเป็นพบบิลด์	C,X,S
	ขอบเขตเป็นโพรเทคต์	C,S
	ขอบเขตเป็นไพรเวต	C
	เมธอด M ของ C	
21	เพิ่มเมธอด	S (เมื่อขอบเขตเป็นพบบิลด์เท่านั้น)
22	ลบเมธอด	
	ขอบเขตเป็นพบบิลด์	C,Y,S
	ขอบเขตเป็นโพรเทคต์	C,S
	ขอบเขตเป็นไพรเวต	C
23	เปลี่ยนชื่อเมธอด	
	ขอบเขตเป็นพบบิลด์	C,Y,S
	ขอบเขตเป็นโพรเทคต์	C,S
	ขอบเขตเป็นไพรเวต	C
24	เปลี่ยนซิกเนเจอร์	
	ขอบเขตเป็นพบบิลด์	C,Y,S
	ขอบเขตเป็นโพรเทคต์	C,S

ตารางที่ 3.1 รูปแบบการเปลี่ยนแปลง (ต่อ)

หมายเลข	รูปแบบการเปลี่ยนแปลง	คลาสที่อาจได้รับผลกระทบ
	ขอบเขตเป็นไพรเวต	C
25	เปลี่ยนชนิดข้อมูลคืนกลับ	
	ขอบเขตเป็นพัबลิค	C,Y,S
	ขอบเขตเป็นโพรเทคต์	C,S
	ขอบเขตเป็นไพรเวต	C
26	เปลี่ยนรายละเอียดการทำงาน	
	ขอบเขตเป็นพับลิค	C,Y,S
	ขอบเขตเป็นโพรเทคต์	C,S
	ขอบเขตเป็นไพรเวต	C
27	เปลี่ยนขอบเขตการเข้าถึง	
	พับลิคเป็นไพรเวต	S,Y
	พับลิคเป็นโพรเทคต์	Y
	โพรเทคต์เป็นพับลิค	
	โพรเทคต์เป็นไพรเวต	S
	ไพรเวตเป็นพับลิค	
	ไพรเวตเป็นโพรเทคต์	
C คือคลาสที่สนใจ		
S คือคลาสลูกของ C		
R คือคลาสที่มีความสัมพันธ์กับ C ในฐานะผู้กระทำ เช่น เป็นแอสโซซิเอชันคลาสของ C		
X คือคลาสที่เรียกใช้แอททริบิวต์ A ของ C		
Y คือคลาสที่เรียกใช้เมธอด M ของ C		

ตารางที่ 3.1 แสดงรูปแบบการเปลี่ยนแปลงที่เป็นไปได้ และอาจส่งผลกระทบต่อเชิงตรรกะแก่ซอฟต์แวร์ พร้อมทั้งกลุ่มของคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงนั้นๆ เพื่อใช้ในการวัดค่าความเสถียรเชิงตรรกะจากซอร์สโคดของซอฟต์แวร์ โดยแต่ละรูปแบบของการเปลี่ยนแปลงจะส่งผลกระทบต่อกลุ่มของคลาสต่างกันไปขึ้นอยู่กับรูปแบบการเปลี่ยนแปลงและขอบเขตของแอททริบิวต์และเมธอด ตัวอย่างเช่น การเปลี่ยนแปลงที่ 16 การลบแอททริบิวต์ หากแอททริบิวต์มีขอบเขตเป็นพับลิก คลาสที่จะได้รับผลกระทบจากการเปลี่ยนแปลงคือ คลาสที่มีแอททริบิวต์นั้นอยู่ คลาสลูก และคลาสที่มีการอ้างอิงแอททริบิวต์นั้นๆ เพราะเมื่อขอบเขตของแอททริบิวต์เป็นพับลิกทุกๆ คลาสในซอฟต์แวร์จะสามารถเข้าถึงได้โดยตรง และเมื่อขอบเขตของแอททริบิวต์เป็นไพรเวตคลาสที่ได้รับผลกระทบคือ คลาสที่มีแอททริบิวต์อยู่ และคลาสลูก ส่วนคลาสอื่นๆ ไม่ได้ได้รับผลกระทบเพราะไม่สามารถเข้าถึงแอททริบิวต์ได้ เป็นต้น

การเปลี่ยนแปลงในระดับของระบบที่แสดงในตารางที่ 3.1 ที่เกี่ยวข้องกับความสัมพันธ์ระหว่างคลาส จะแยกพิจารณาเป็นความสัมพันธ์แบบทางเดียวทั้งสิ้น และในการจำลองการเปลี่ยนแปลงเพื่อวิเคราะห์และคำนวณความเสถียรเชิงตรรกะ จะทำการจำลองที่ละคลาสและพิจารณาผลกระทบที่เกิดกับคลาสที่สนใจเป็นหลัก ทำให้การเปลี่ยนแปลงบางรูปแบบถือว่าไม่เกิดผลกระทบขึ้น เช่นก รเพิ่มความสัมพันธ์ให้คลาสที่สนใจเป็นแอสโซซิเอชันคลาส หรือมีคลาสอื่นมาเรียกใช้ ในกรณีนี้ถือว่าไม่มีผลกระทบกับคลาสที่สนใจ ในขณะที่เพิ่มความสัมพันธ์แอสโซซิเอชันคลาสของคลาสที่สนใจ หรือเรียกใช้คลาสอื่น กรณีนี้จะเกิดผลกระทบกับคลาสที่สนใจและคลาสลูก เป็นต้น อย่างไรก็ตามในการคำนวณความเสถียรเชิงตรรกะจะต้องทำการจำลองการเปลี่ยนแปลงทุกรูปแบบให้แก่ทุกๆ คลาส ดังนั้นทุกคลาสจะได้รับการวิเคราะห์เหมือนกันหมดทั้งสิ้น

รูปแบบการเปลี่ยนแปลงและคลาสที่ได้รับผลกระทบเชิงตรรกะในตารางที่ 3.1 ใช้เป็นพื้นฐานในการวิเคราะห์เพื่อวัดค่าความเสถียรเชิงตรรกะของคลาสใดๆ โดยเมื่อทำการตรวจสอบการเปลี่ยนแปลงหนึ่งๆ ของคลาสใดๆ (C) แล้วทำการเปรียบเทียบกับตารางที่ 3.1 ว่าคลาสที่อาจได้รับผลกระทบจากการเปลี่ยนแปลงนั้นๆ มีคลาสใดบ้าง

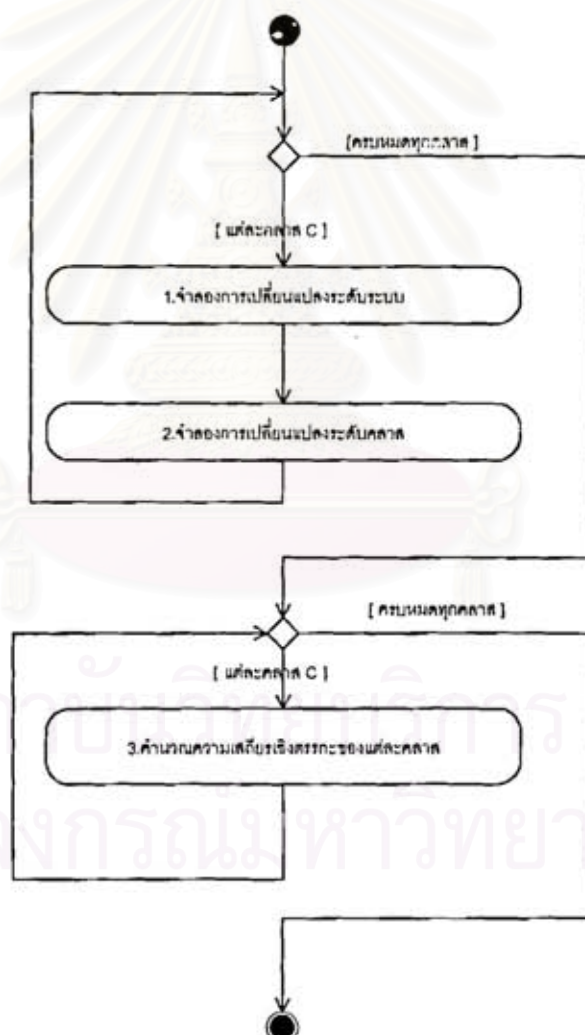
การเปลี่ยนแปลงในตารางที่ 3.1 เป็นการเปลี่ยนแปลงพื้นฐานทั่วไปและเป็นองค์ประกอบของการเปลี่ยนแปลงระดับสูงต่างๆ เช่นการทำรีแฟคเตอร์ เป็นต้น ตัวอย่างของความสัมพันธ์ระหว่างการทำรีแฟคเตอร์ [13] และการเปลี่ยนแปลงแสดงดังตารางที่ 3.1 เช่น

- การย้ายเมธอด (Move Method): คือการลบเมธอด และเพิ่มเมธอดตามลำดับ

- การขยายคลาส (Extract Class) : คือการลบเมทอดด ลบแอทธิบิวต์ เพิ่มคลาส และเพิ่มความสัมพันธ์ระหว่างคลาสตามลำดับ

โดยตารางแสดงความสัมพันธ์ระหว่างการทำรีแฟคเตอร์กับการเปลี่ยนแปลงพื้นฐานแสดงในภาคผนวก ข

ขั้นตอนการคำนวณความเสถียรเชิงตรรกะของคลาสสามารถแสดงเป็นแผนภาพกิจกรรมได้ดังรูปที่ 3.2 โดยวิธีการในการคำนวณความเสถียรเชิงตรรกะของงานวิจัยนี้จะทำการปรับปรุงวิธีการหาค่าความเสถียรเชิงตรรกะจากงานวิจัยของ Mahmoud O. Elish และ David Rine [7] ซึ่งเป็นวิธีการหาค่าความเสถียรเชิงตรรกะที่สอดคล้องกับแนวทางของงานวิจัยนี้เนื่องจากเป็นวิธีการในการหาค่าความเสถียรเชิงตรรกะของโปรแกรมเชิงวัตถุจากซอร์สโคด และมีวิธีการคำนวณความเสถียรเชิงตรรกะใกล้เคียงกับที่ต้องการ



รูปที่ 3.2 ขั้นตอนการคำนวณความเสถียรเชิงตรรกะ

จากรูปที่ 3.2 การคำนวณความเสถียรเชิงตรรกะของโปรแกรมหนึ่งประกอบไปด้วย 4 ขั้นตอนด้วยกัน เริ่มจากการจำลองการเปลี่ยนแปลงระดับของระบบแก่คลาสในขั้นตอนที่ 1 แล้วจึงทำการจำลองการเปลี่ยนแปลงแก่ทุกๆ เมทอดของคลาสในขั้นตอนที่ 2 และทำการจำลองการเปลี่ยนแปลงแก่ทุกๆ แอททริบิวต์ของคลาสในขั้นตอนที่ 3 และกลับไปทำขั้นตอนที่ 1 ใหม่จนกว่าจะครบทุกคลาส และในระหว่างขั้นตอนที่ 1 ถึง 3 จะมีการเก็บค่าจำนวนการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมดของโปรแกรม และจำนวนครั้งที่แต่ละคลาสจะได้รับผลกระทบจากการเปลี่ยนแปลงนั้นๆ เพื่อนำไปคำนวณค่าความเสถียรเชิงตรรกะของแต่ละคลาสในขั้นตอนที่ 4 ต่อไป

การคำนวณความเสถียรเชิงตรรกะเริ่มด้วยการจำลองการเปลี่ยนแปลงตามตารางที่ 3.1 ให้กับทุกคลาสในโปรแกรมทีละคลาส โดยเริ่มจากการเปลี่ยนแปลงในระดับของระบบก่อน จากนั้นจึงจำลองการเปลี่ยนแปลงระดับของคลาสให้แก่ทุกเมทอดและแอททริบิวต์ในคลาส และในการจำลองการเปลี่ยนแปลงแต่ละครั้งจะทำการตรวจสอบและบันทึกข้อมูลดังนี้

1. เพิ่มค่าจำนวนการเปลี่ยนแปลงทั้งหมดไปอีกหนึ่งเมื่อการเปลี่ยนแปลงนั้นสามารถทำการจำลองได้
2. ตรวจสอบคลาสที่อาจได้รับผลกระทบและเพิ่มค่าจำนวนครั้งที่ได้รับผลกระทบของคลาสนั้นๆ โดยจะนับเพียงแค่ 1 ครั้งต่อ 1 การเปลี่ยนแปลงเท่านั้น

เมื่อเสร็จสิ้นการจำลองการเปลี่ยนแปลงและเก็บข้อมูลทุกคลาสเสร็จสิ้น จึงทำการคำนวณค่าความเสถียรเชิงตรรกะของแต่ละคลาสตามสมการที่ 3 และตัวอย่างการคำนวณความเสถียรเชิงตรรกะแสดงได้ดังตารางที่ 3.2 และ 3.3

ตารางที่ 3.2 ตัวอย่างซอร์สโค้ด

<pre>public class C1{ public int a1; private C2 a2; public int mC1(){ C3 c = new C3(); a1= c.mC3(); return a1; } }</pre>	<pre>public class C2{ public int b1; Public void mC2(){ C3 c = new C3(); b1 = c.mC3(); } }</pre>	<pre>public class C3 extends C1{ public int c1; public int mC3(){ c1 = 10; return c1; } }</pre>
--	--	---

ตารางที่ 3.3 ตัวอย่างการคำนวณความเสถียรเชิงตรรกะ

การเปลี่ยนแปลง	คลาส C1	คลาส C2	คลาส C3
ลบคลาส C1	x		x
ลบคลาส C2	x	x	
ลบคลาส C3	x	x	x
เพิ่มแอตทริบิวต์ของ C1	x		x
ลบแอตทริบิวต์ a1	x		x
เปลี่ยนขอบเขต a1 เป็น private			x
เปลี่ยนขอบเขต a1 เป็น protected			
เปลี่ยนชื่อ method mC3()	x	x	x
เปลี่ยนขอบเขต mC3() เป็น private	x	x	
เปลี่ยนขอบเขต mC3() เป็น protected	x	x	
รวม	8	5	6
ความเสถียรเชิงตรรกะ	$1-(8/10)=0.2$	$1-(5/10)=0.5$	$1-(6/10)=0.4$

ตารางที่ 3.2 คือตัวอย่างซอร์สโค้ด ซึ่งประกอบไปด้วยคลาส 3 คลาส คือ C1, C2 และ C3 โดย C2 เป็นคลาสลูกของ C1 และเมธอด mC3() ถูกเรียกใช้โดย C1 และ C2 และตารางที่ 3.3 คือส่วนหนึ่งของการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมด และคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงนั้นๆ ที่นำมาเพื่อแสดงให้เห็นวิธีการคำนวณความเสถียรเชิงตรรกะเท่านั้น โดยให้ การเปลี่ยนแปลงที่เป็นไปได้ทั้งหมดเท่ากับ 10 ครั้ง และคลาส C1, C2 และ C3 ได้รับผลกระทบเป็นจำนวน 8, 5 และ 6 ครั้งตามลำดับ ดังนั้นสามารถคำนวณความเสถียรเชิงตรรกะของแต่ละ คลาสตามสมการที่ 3 ได้ดังนี้

$$\text{ความเสถียรเชิงตรรกะของ C1} = 1 - (8/10) = 0.2$$

$$\text{ความเสถียรเชิงตรรกะของ C2} = 1 - (5/10) = 0.5$$

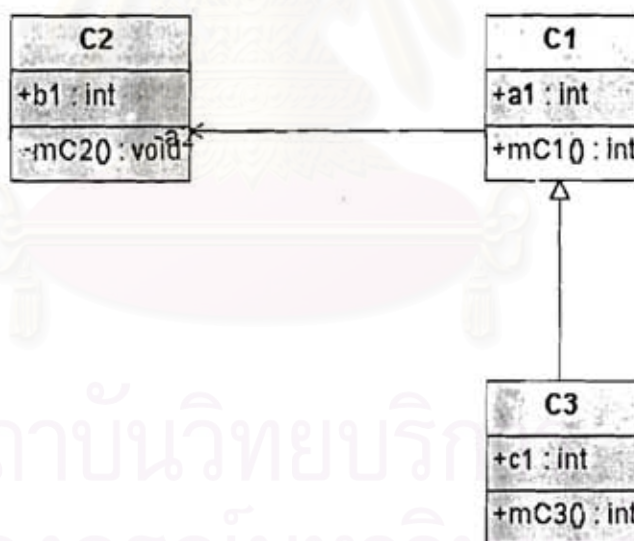
$$\text{ความเสถียรเชิงตรรกะของ C3} = 1 - (6/10) = 0.4$$

ดังนั้นจะได้ว่าคลาส C1 C2 และ C3 มีความเสถียรเชิงตรรกะเป็น 0.2 0.5 และ 0.4 ตามลำดับ

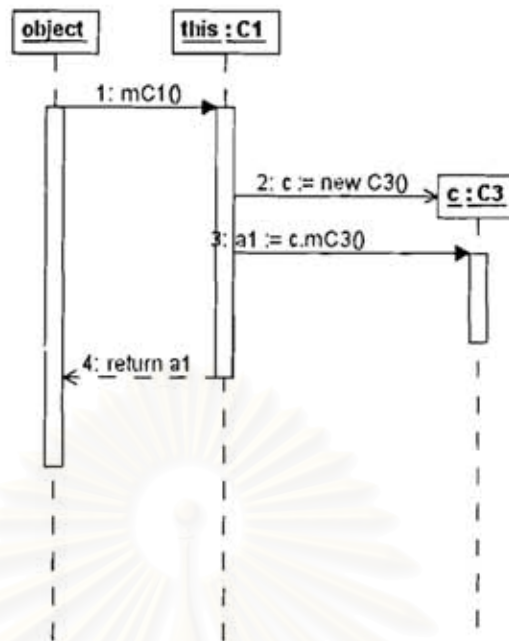
3.2 การแปลงซอร์สโค้ดเป็นแผนภาพคลาสและแผนภาพซีเควนซ์

การแปลงกลับแผนภาพในขั้นตอนนี้จะอาศัยโปรแกรม MagicDraw UML เวอร์ชัน 9.5 [14] ซึ่งมีคุณสมบัติในการแปลงกลับซอร์สโค้ดภาษาต่างๆ เป็นแผนภาพยูเอ็มแอลรวมไปถึงภาษาจาวาด้วย โดยในการแปลงกลับนี้โปรแกรมจะทำการค้นหาไฟล์ซอร์สโค้ดภาษาจาวาทั้งหมดที่อยู่ภายใต้โฟลเดอร์ที่ต้องการแล้วทำการแปลงกลับเป็นข้อมูลเข้าสู่โปรแกรม และผู้ใช้สามารถเลือกให้โปรแกรมทำการสร้างแผนภาพคลาสของโปรแกรมที่ถูกแปลงกลับได้โดยอัตโนมัติ จากนั้นผู้ใช้ต้องทำการเลือกและสั่งให้โปรแกรม MagicDraw ทำการสร้างแผนภาพซีเควนซ์จากเมธอดของโปรแกรมที่ถูกแปลงกลับทีละเมธอด โดยจะได้ 1 แผนภาพซีเควนซ์ต่อ 1 เมธอด เมื่อสร้างแผนภาพซีเควนซ์ของทุกเมธอดจนครบแล้วสามารถทำการบันทึกข้อมูลแผนภาพให้อยู่ในรูปแบบของภาษาเอ็กซ์เอ็มแอลได้เพื่อนำไปใช้ต่อไป ตัวอย่างวิธีการแปลงกลับแสดงอยู่ในภาคผนวก ค

รูปที่ 3.3 แสดงตัวอย่างแผนภาพคลาสที่ได้จากการแปลงกลับซอร์สโค้ดในตารางที่ 3.2 และรูปที่ 3.4 แสดงตัวอย่างแผนภาพซีเควนซ์ที่ได้จากการแปลงกลับเมธอด `mC1()` ของคลาส `C1` ในตารางที่ 3.2



รูปที่ 3.3 ตัวอย่างแผนภาพคลาสที่ได้จากการแปลงกลับ



รูปที่ 3.4 ตัวอย่างแผนภาพซีควเอนซ์ที่ได้จากการแปลงกลับ

3.3 การวัดค่าตัววัดแผนภาพที่ได้จากการแปลงกลับ

หลังจากแปลงกลับแผนภาพคลาสและแผนภาพซีควเอนซ์ของชุดโปรแกรมพัฒนาทุกโปรแกรม และบันทึกอยู่ในรูปแบบของเอกซ์เอ็มแอลเป็นที่เรียบร้อยแล้ว จะทำการวัดตัววัดแผนภาพที่ได้กำหนดไว้ด้วยโปรแกรม SDMetrics เวอร์ชัน 2.0 [15] ซึ่งสามารถอ่านข้อมูลแผนภาพจากไฟล์เอกซ์เอ็มแอลและทำการวัดค่าตัววัดแผนภาพต่างๆ ได้ โดยได้มีการเพิ่มเติมตัววัดแผนภาพเข้าไป และรายละเอียดของตัววัดแผนภาพที่ทำการเพิ่มเติมอยู่ในบทที่ 5 เมื่อทำการคำนวณตัววัดแผนภาพแล้วสามารถส่งออกข้อมูลให้อยู่ในรูปแบบไฟล์ซีเอสวี (CSV) เพื่อนำไปใช้ในการหาความสัมพันธ์กับความเสถียรเชิงตรรกะของคลาส เพื่อสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพในบทที่ 4

ตัววัดแผนภาพที่ใช้ในงานวิจัยนี้สามารถแบ่งออกได้เป็น 2 กลุ่มคือ ตัววัดแผนภาพคลาสและตัววัดแผนภาพซีควเอนซ์ โดยตัววัดแผนภาพคลาสประกอบไปด้วย

1. ตัววัดจำนวนแอตทริบิวต์ (NumAttr) คือ จำนวนของแอตทริบิวต์ของคลาส
2. ตัววัดจำนวนโอเปอเรเตอร์ (NumOps) คือ จำนวนโอเปอเรเตอร์หรือเมทอดของคลาส
3. ตัววัดจำนวนโอเปอเรเตอร์ที่มีขอบเขตการเข้าถึงเป็นแบบพับบลิค (NumPubOps) จำนวนโอเปอเรเตอร์หรือเมทอดของคลาสที่มีขอบเขตการเข้าถึงเป็นพับบลิค

4. ตัววัดระดับความลึกคลาสภายใน (Nesting) คือ ระดับความลึกที่คลาสอยู่ในคลาสอื่นๆ
5. ตัววัดจำนวนอินเตอร์เฟซที่ใช้สร้างคลาส (IFImpI) คือ จำนวนของอินเตอร์เฟซที่คลาสที่สนใจนำมาใช้ในการสร้างคลาส
6. ตัววัดจำนวนคลาสลูก (NOC) คือ จำนวนของคลาสลูกของคลาสที่สนใจ
7. ตัววัดจำนวนคลาสแม่ (NOP) คือจำนวนคลาสแม่ของคลาสที่สนใจ
8. ตัววัดจำนวนคลาสสืบทอด (NumDesc) คือ จำนวนของคลาสลูกรวมไปถึงคลาสที่สืบทอดจากคลาสลูกและต่อไป
9. ตัววัดจำนวนคลาสบรรพบุรุษ (NumAnc) คือ จำนวนของคลาสแม่และคลาสต้นที่คลาสแม่สืบทอดมา
10. ตัววัดระดับความลึกของการสืบทอด (DIT) คือ ระดับของความลึกของคลาสที่สนใจในแผนผังต้นไม้การสืบทอด
11. ตัววัดความลึกจากคลาสสู่ปลายสุดของการสืบทอด (CLD) คือ ระดับของความลึกจากคลาสที่สนใจจนถึงระดับที่ลึกที่สุดในแผนผังต้นไม้การสืบทอด
12. ตัววัดจำนวนโอเปอเรเตอร์ที่รับสืบทอดมา (OpsInh) คือ จำนวนของโอเปอเรเตอร์หรือเมทอดที่รับสืบทอดมาจากคลาสอื่น
13. ตัววัดจำนวนแอททริบิวต์ที่รับสืบทอดมา (AttrInh) คือจำนวนของแอททริบิวต์ที่รับสืบทอดมาจากคลาสอื่น
14. จำนวนครั้งที่คลาสมีความสัมพันธ์แอสโซซิเอชันในฐานะผู้เรียกใช้ (NumAss_User) คือ จำนวนความสัมพันธ์แอสโซซิเอชันของคลาสที่สนใจในแผนภาพคลาส เมื่อคลาสที่สนใจเป็นผู้ใช้งานคลาสอื่น
15. จำนวนครั้งที่คลาสมีความสัมพันธ์แอสโซซิเอชันในฐานะผู้ถูกเรียกใช้ (NumAss_Provider) คือจำนวนความสัมพันธ์แอสโซซิเอชันของคลาสที่สนใจในแผนภาพคลาส เมื่อคลาสที่สนใจเป็นผู้ถูกใช้งานจากคลาสอื่น
16. จำนวนครั้งที่คลาสถูกใช้เป็นพารามิเตอร์ (EC_Par) คือ จำนวนครั้งที่คลาสที่สนใจถูกนำไปใช้เป็นพารามิเตอร์ในคลาสอื่นและคลาสตัวเอง
17. จำนวนพารามิเตอร์ในคลาสที่มีชนิดเป็นคลาสอื่นๆ (IC_Par) คือ จำนวนของพารามิเตอร์ในคลาสที่สนใจที่มีชนิดเป็นคลาสอื่นๆ

สำหรับตัววัดแผนภาพที่เคอร์เนลประกอบไปด้วย

1. จำนวนคลาสสัฟายเออร์ที่มีคลาสนี้เป็นพื้นฐาน (ClassifInst) คือ จำนวนครั้งที่คลาสที่สนใจปรากฏในแผนภาพที่เคอร์เนล หรือคือจำนวนขอบเขตบนแผนภาพที่เคอร์เนลของคลาสที่สนใจ
2. จำนวนครั้งการส่งแมสเสจไปยังคลาสอื่น (MsgSent) คือ จำนวนครั้งที่คลาสที่สนใจส่งแมสเสจไปยังคลาสอื่นๆ
3. จำนวนครั้งที่ได้รับแมสเสจจากคลาสอื่น (MsgRecv) คือ จำนวนครั้งที่คลาสที่สนใจได้รับแมสเสจจากคลาสอื่นๆ
4. จำนวนครั้งที่ส่งแมสเสจหาตัวเอง (MsgSelf) คือ จำนวนครั้งที่คลาสที่สนใจส่งแมสเสจไปยังตนเอง



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การสร้างและทดสอบโมเดลประมาณค่าความเสถียรเชิงตรรกะ

4.1 การสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะ

ในการสร้างโมเดลเพื่อประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพจะใช้ข้อมูลค่าความเสถียรเชิงตรรกะของคลาสที่คำนวณได้จากซอร์สโคด และค่าตัววัดแผนภาพที่ได้จากการแปลงกลับซอร์สโคด โดยใช้วิธีการวิเคราะห์ความถดถอยเชิงซ้อนแบบเสถปไวส์ และอาศัยโปรแกรม SPSS เวอร์ชัน 13.0 ช่วยในการคำนวณ

จากนั้นเมื่อได้โมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาสจากแผนภาพแล้ว จึงทำการประเมินผลความคลาดเคลื่อนของโมเดลต่อไป

ในการสร้างโมเดลเพื่อประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพคลาสและแผนภาพซีควেনซ์ จะอาศัยชุดโปรแกรมสำหรับพัฒนาในการวิเคราะห์ความถดถอยเชิงซ้อนหาความสัมพันธ์ระหว่างค่าความเสถียรเชิงตรรกะที่ได้จากการคำนวณจากซอร์สโคด กับค่าตัววัดแผนภาพต่างๆ ที่ได้จากการแปลงกลับแผนภาพตามขั้นตอนการดำเนินงานในบทที่ 3

ชุดโปรแกรมพัฒนาจะประกอบไปด้วยโปรแกรมที่เขียนขึ้นโดยภาษาจาวา และจะประกอบไปด้วยโปรแกรม 3 ประเภทด้วยกันดังนี้

1. โปรแกรมด้านการคำนวณ ได้แก่ โปรแกรมเครื่องคิดเลข
2. โปรแกรมด้านการจัดการข้อความ ได้แก่ โปรแกรมเทกซ์อีดิเตอร์
3. โปรแกรมด้านการจัดการภาพ ได้แก่ โปรแกรมวาดและตกแต่งภาพ

สำหรับรายชื่อและขนาดของโปรแกรมในชุดพัฒนาแสดงในตารางที่ 4.1

ตารางที่ 4.1 รายชื่อโปรแกรมที่อยู่ในชุดพัฒนา

ชื่อโปรแกรม	จำนวนคลาส	รายละเอียด
โปรแกรมด้านการคำนวณ		
Java Scientific Calculator	123	เป็นเครื่องคิดเลขทางวิทยาศาสตร์อย่างง่ายใช้งานด้วยเมาส์และคีย์บอร์ด

ตารางที่ 4.1 รายชื่อโปรแกรมที่อยู่ในชุดพัฒนา (ต่อ)

ชื่อโปรแกรม	จำนวนคลาส	รายละเอียด
InfixCalCulator	4	เป็นเครื่องคิดเลขทางวิทยาศาสตร์สำหรับ Java Phone โดยใช้การคำนวณแบบ Infix Notation
CIAC Calculator	47	เป็นโปรแกรมคำนวณที่สร้างขึ้นให้คล้าย Matlab สามารถคำนวณเมทริกซ์ เวกเตอร์ และการคำนวณทั่วไป
โปรแกรมด้านการจัดการข้อความ		
C.E.L.Pad	9	เป็นโปรแกรมจัดการข้อความขนาดเล็กที่สามารถปรับแต่งรูปร่างลักษณะของข้อความได้
SWedit	79	เป็นโปรแกรมจัดการข้อความที่สร้างขึ้นโดยต้องการให้เป็นโปรแกรมสำหรับใช้ในการเขียนโปรแกรม
EdasTexter2	19	เป็นโปรแกรมจัดการข้อความขนาดเล็กที่สามารถปรับแต่งค่าต่างๆ ได้
โปรแกรมด้านการจัดการภาพ		
JImageMosaic	52	เป็นโปรแกรมสำหรับการสร้างภาพโมเสค
Image Processing	7	เป็นโปรแกรมตกแต่งรูปภาพคล้ายกับโปรแกรมโฟโตชอป
RadImage	25	เป็นโปรแกรมตกแต่งรูปภาพมีความสามารถในการแก้ไขรูปภาพทั่วไป และบันทึกรูปด้วยรูปแบบเอกซ์เอ็มแอล

ทั้งนี้ในการสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพคลาสและแผนภาพซีควเอนซ์ด้วยชุดโปรแกรมข้างต้น จะแบ่งทำการทดลองเพื่อหาโมเดลประมาณค่าความเสถียรเชิงตรรกะของโปรแกรมแต่ละประเภทแยกจากกัน เนื่องจากโปรแกรมแต่ละประเภทจะมีรูปแบบการทำงานหลักที่ต่างกันออกไปและคล้ายคลึงกันเองในกลุ่มโปรแกรมประเภทเดียวกัน โดยโปรแกรมในด้านการคำนวณจะเน้นการทำงานที่เกี่ยวกับการคำนวณและตัวเลข ในขณะที่โปรแกรมการจัดการข้อความจะทำงานกับข้อความและตัวอักษรเป็นหลัก และโปรแกรมการ

จัดการภาพจะเน้นการแสดงผลภาพและการแก้ไขรูปภาพเป็นหลัก ทั้งนี้เพื่อให้ได้โมเดลประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพคลาสและแผนภาพซีเควน์ซ์ที่ถูกต้องมากที่สุด

การเลือกโปรแกรมเพื่อนำมาใช้ในวิทยานิพนธ์นี้ ได้ทำการรวบรวมโปรแกรมทั้ง 3 ประเภทจาก [16] และ [17] โดยเป็นการเลือกแบบสุ่ม อย่างไรก็ตามการเลือกโปรแกรมมาใช้มีข้อจำกัดในเรื่องของขนาดของโปรแกรม เนื่องจากโปรแกรม SDMetrics ไม่รองรับแผนภาพที่มีองค์ประกอบมากกว่า 60000 องค์ประกอบ ดังนั้นหากโปรแกรมใดเมื่อทำการแปลงเป็นแผนภาพคลาสและแผนภาพซีเควน์ซ์แล้ว มีจำนวนขององค์ประกอบมากกว่าที่กำหนดจะต้องทำการตัดออก และการเลือกโปรแกรมเข้าสู่ชุดพัฒนาจะทำการเลือกโปรแกรมที่มีจำนวนคลาสมาก่อน จากนั้นเมื่อโปรแกรมที่เหลือมีจำนวนคลาสน้อยใกล้เคียงกันจะทำการสุ่มเลือกแทนให้ได้จำนวนประเภทละ 3 โปรแกรม ทั้งนี้เพื่อให้ได้จำนวนของข้อมูลที่ใช้ในการสร้างโมเดลที่มากเพียงพอ เพื่อให้โมเดลที่ได้มีความถูกต้องและน่าเชื่อถือสามารถนำไปใช้งานได้จริง

หลังจากทำการคำนวณค่าความเสถียรเชิงตรรกะของทุกคลาส พร้อมทั้งค่าตัววัดแผนภาพเสร็จสิ้นแล้ว นำข้อมูลที่ได้มาทำการวิเคราะห์ความถดถอยเชิงซ้อนแบบสเตปไวส์ โดยอาศัยโปรแกรมเอสพีเอสเอสสำหรับวินโดวส์ช่วยในการคำนวณ โดยตัวอย่างข้อมูลที่นำมาใช้แสดงอยู่ในตารางที่ 4.2 และจากตารางที่ 4.2 จะพบว่าความเสถียรเชิงตรรกะของคลาสจะมีค่าแตกต่างกันไปแต่โดยส่วนใหญ่จะมีค่าประมาณ 0.88 ถึง 0.98 ทั้งนี้เนื่องจากการคำนวณความเสถียรเชิงตรรกะคำนวณจากจำนวนการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมด ซึ่งมีค่ามากกว่าจำนวนครั้งที่แต่ละคลาสจะได้รับผลกระทบมาก ดังนั้นในการพิจารณาความเสถียรเชิงตรรกะจะต้องพิจารณาที่ระดับทศนิยมที่มีความละเอียดสูง

เนื่องจากการวิจัยนี้มีวัตถุประสงค์เพื่อสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะจากตัววัดแผนภาพคลาสและแผนภาพซีเควน์ซ์ ดังนั้นในการวิเคราะห์ความถดถอยเชิงซ้อนจึงกำหนดให้ความเสถียรเชิงตรรกะของคลาสเป็นตัวแปรตาม และตัววัดแผนภาพต่างๆ เป็นตัวแปรอิสระ และทำการวิเคราะห์ที่ระดับความเชื่อมั่น 95% ซึ่งหมายถึงโอกาสที่ความเสถียรเชิงตรรกะของคลาส และตัววัดแผนภาพมีความสัมพันธ์กันจริง

ขั้นตอนในการวิเคราะห์ความถดถอยเชิงซ้อนแบบสเตปไวส์จะดำเนินการโดยเลือกตัวแปรอิสระที่มีระดับนัยสำคัญสูงสุดเข้าสมการทีละตัว และเมื่อเลือกตัวแปรตัวใหม่เข้ามาจะต้องทำการคำนวณระดับนัยสำคัญใหม่ทั้งตัวแปรที่เลือกเข้ามาแล้วและตัวแปรที่ยังไม่ได้เลือกเข้ามา และทำการตัดตัวแปรที่ถูกเลือกเข้ามาในโมเดลแล้วทิ้งไป หากตัวแปรนั้นมีระดับนัยสำคัญที่คำนวณได้ใหม่น้อยกว่าที่กำหนดไว้ จากนั้นพิจารณาตัวแปรที่เหลือเพื่อนำตัวแปรที่มีนัยสำคัญสูงสุดเข้ามาเป็นลำดับต่อไป จนกระทั่งไม่สามารถเลือกได้อีกต่อไป

ตารางที่ 4.2 ตัวอย่างค่าความเสถียรเชิงตรรกะของคลาสและค่าตัววัดแผนภาพที่ใช้ในการสร้างโมเดล

ClassName	ClassStability	NumAss _Provider	NumAss _User	NOP	NumAttr	NumOps	Num PubOps	IFimpl	NOC	Num Desc	Num Anc	DiT	CLD	Ops Inh	Attr Inh	EC_ Attr	IC_ Attr	EC_ Par	IC_ Par	Msg Sent	Msg Recv	Msg Self
calc	0.931258	1	0	0	4	9	2	0	0	0	0	0	0	0	0	0	0	0	2	49	19	22
Float	0.523995	0	1	0	10	43	42	0	0	0	0	0	0	0	0	0	0	3	0	12	12	894
InfCalc	0.875486	0	0	0	3	8	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46
input	0.885863	1	1	0	3	11	11	0	0	0	0	0	0	0	0	0	0	1	2	9	35	22
CELPad	0.418398	0	0	0	3	13	12	0	0	0	0	0	0	0	0	0	0	0	0	3	3	103
FileMenuActionListener	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	37
EditMenuActionListener	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	14
LookAndFeelActionListener	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	11
CFilter	0.949555	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CplusplusFilter	0.949555	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CSharpFilter	0.949555	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
JavaFilter	0.949555	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TextFilter	0.949555	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Test	0.98613	0	0	0	0	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23
GridFrame	0.817002	5	0	1	0	10	4	1	0	0	1	1	0	17	1	0	0	0	0	22	7	95
GridLines	0.975839	0	0	0	3	7	7	0	0	0	0	0	0	0	0	0	0	0	0	1	3	39
ImageComponent	0.915884	0	1	1	0	18	15	0	0	0	1	1	0	12	4	0	0	3	0	2	18	77
ImageComponentFactory	0.983445	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	1	8	4	20
ImageSurface	0.823714	1	1	1	0	32	19	1	0	0	1	1	0	13	0	0	0	0	1	11	3	152
AbstractComponentIOFactory	0.981208	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	1	5	5	4
ComponentIO	0.961074	0	0	0	0	13	13	0	2	2	0	0	1	0	0	0	0	3	4	0	3	10
ComponentIORaster	0.924385	0	0	1	0	9	8	0	0	0	1	1	0	13	0	0	0	0	4	1	3	18
ComponentIOXML	0.932439	0	0	1	0	5	5	0	0	0	1	1	0	13	0	0	0	0	4	4	3	24
RasterIOFactory	0.980313	0	0	0	0	2	2	1	0	0	0	0	0	0	0	0	0	0	1	4	2	2

4.1.1 การสร้างโมเดลของกลุ่มโปรแกรมด้านการคำนวณ

การวิเคราะห์ความถดถอยเชิงซ้อนแบบสเตปไวส์ของกลุ่มโปรแกรมด้านการคำนวณ จะเลือกตัววัด ClassifInst ที่มีระดับความสัมพันธ์เชิงส่วนสูงที่สุด หรือมีความสัมพันธ์กับความเสถียรเชิงตรรกะของคลาสมากที่สุด คือ -0.752 เข้ามาเป็นอันดับแรกดังจะดูได้จากตารางที่ 4.5 จากนั้นทำการคำนวณระดับความสัมพันธ์ใหม่จะพบว่าตัววัดที่มีความสัมพันธ์เชิงส่วนสูงที่สุดที่เหลืออยู่ และมีความสัมพันธ์กับความเสถียรเชิงตรรกะ ($\text{Sig } t < 0.05$) คือ NumOps โดยมีความสัมพันธ์เชิงส่วนเท่ากับ 0.548 ดังตารางที่ 4.6 และเมื่อนำตัววัด NumOps เข้ามาในโมเดลแล้วต้องทำการคำนวณระดับความสัมพันธ์ใหม่อีกครั้ง เพื่อตัดตัววัดที่มีระดับความสัมพันธ์ใหม่น้อยกว่าที่กำหนด ($\text{Sig } t > 0.05$) ทว่าจากตารางที่ 4.5 จะพบว่าตัววัดทั้งสองต่างมีระดับนัยสำคัญในระดับที่กำหนดไว้จึงไม่ต้องทำการตัดตัววัดใดทิ้งไป จากนั้นทำการเลือกตัววัดที่เหลืออยู่ต่อไปจนกว่าจะไม่มีตัววัดใดที่มีความสัมพันธ์กับความเสถียรเชิงตรรกะ ($\text{Sig } t > 0.05$) อีกต่อไป

เมื่อทำการวิเคราะห์ความถดถอยเสร็จสิ้นแล้วจะได้ลำดับของการนำเข้าและนำออกตัววัด แผนภาพดังแสดงในตารางที่ 4.4 และมีผลสรุปของโมเดลดังในตารางที่ 4.3 จากนั้นทำการเลือกโมเดลที่มีค่าสัมประสิทธิ์การตัดสินใจเชิงซ้อนมากที่สุด มาใช้เป็นโมเดลในการประมาณค่าโดยดูจากตารางที่ 4.3 คือ โมเดลที่ 8 โดยมีค่าสัมประสิทธิ์การตัดสินใจเชิงซ้อนเท่ากับ 0.739 และประกอบไปด้วยตัววัดแผนภาพ 6 ตัว ได้แก่ ClassifInst NumOps NumAnc AttrInh MsgSent และ NumDesc และจะมีสมการดังนี้

$$\begin{aligned} \text{ความเสถียรเชิงตรรกะของคลาส} = & 0.9805680980 - 0.0032740467 \text{ClassifInst} \\ & + 0.0048858207 \text{NumOps} - 0.0064901626 \text{NumAnc} + 0.0080432844 \text{AttrInh} \\ & - 0.0010171931 \text{MsgSent} + 0.0005569513 \text{NumDesc} \end{aligned}$$

จากโมเดลที่ได้จะพบว่าความเสถียรเชิงตรรกะของคลาสใดๆ ของโปรแกรมด้านการคำนวณจะมีค่าเพิ่มขึ้นเมื่อ มีจำนวนโอเปอเรเตอร์เพิ่มขึ้น หรือมีจำนวนแอททริบิวต์ที่สืบทอดมามากขึ้น หรือมีจำนวนคลาสลูกมากขึ้น ในขณะที่ความเสถียรเชิงตรรกะของคลาสใดๆ จะมีค่าลดลงเมื่อ มีจำนวนออบเจกต์ของคลาสในแผนภาพซีเควนซ์มาก มีจำนวนคลาสบรรพบุรุษมาก หรือมีการส่งแมสเสจไปคลาสอื่นมาก

ทั้งนี้ผลที่ได้มาจากพฤติกรรมการทำงานของโปรแกรมด้านการคำนวณเช่น คลาสที่มีโอเปอเรเตอร์เป็นจำนวนมากเป็นคลาสที่มีโอเปอเรเตอร์หรือเมทอดสำหรับการคำนวณให้

คลาสอื่นเรียกใช้ โดยที่ตัวคลาสเองไม่ได้เรียกใช้คลาสอื่นทำให้เมื่อมีการเปลี่ยนแปลงคลาสนี้จึงได้รับผลกระทบน้อย เป็นต้น นอกจากนี้ผลที่ได้ในบางส่วนยังสอดคล้องกับทฤษฎีของความเสถียรเชิงตรรกะโดยตรงเช่น คลาสที่มีการส่งแมสเสจมากจะมีความเสถียรเชิงตรรกะต่ำ เพราะการส่งแมสเสจคือการเรียกใช้งานคลาสอื่น ดังนั้นยิ่งเรียกใช้งานคลาสอื่นมาก โอกาสการได้รับผลกระทบเมื่อมีการเปลี่ยนแปลงจะยิ่งมากด้วย เป็นต้น

โดยค่าความเสถียรเชิงตรรกะที่ได้จากโมเดลประมาณค่า อาจมีค่าน้อยกว่า 0 หรือมากกว่า 1 ได้ ซึ่งค่าความเสถียรเชิงตรรกะโดยปกติจะมีค่าระหว่าง 0 ถึง 1 ดังนั้นจึงต้องมีการปรับค่าก่อนนำไปใช้งานต่อไป โดยถ้าค่าประมาณความเสถียรเชิงตรรกะที่ได้มีค่าน้อยกว่า 0 ให้ถือว่ามีความเท่ากับ 0 และถ้าค่าประมาณความเสถียรเชิงตรรกะที่ได้มีค่ามากกว่า 1 ให้ถือว่ามีความเท่ากับ 1

ตารางที่ 4.3 ผลสรุปของโมเดลสำหรับโปรแกรมด้านการคำนวณ

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.752 ^a	.565	.563	.026810020
2	.834 ^b	.696	.692	.022499574
3	.843 ^c	.711	.706	.021973561
4	.848 ^d	.719	.713	.021736006
5	.858 ^e	.736	.728	.021149262
6	.863 ^f	.744	.735	.020869070
7	.863 ^g	.740	.733	.020969465
8	.865 ^h	.748	.739	.020705145

a. Predictors: (Constant), CLASSIFI

b. Predictors: (Constant), CLASSIFI, NUMOPS

c. Predictors: (Constant), CLASSIFI, NUMOPS, MSGSELF

d. Predictors: (Constant), CLASSIFI, NUMOPS, MSGSELF, NUMANC

e. Predictors: (Constant), CLASSIFI, NUMOPS, MSGSELF, NUMANC, ATTRINH

f. Predictors: (Constant), CLASSIFI, NUMOPS, MSGSELF, NUMANC, ATTRINH, MSGSENT

g. Predictors: (Constant), CLASSIFI, NUMOPS, NUMANC, ATTRINH, MSGSENT

h. Predictors: (Constant), CLASSIFI, NUMOPS, NUMANC, ATTRINH, MSGSENT, NUMDESC

ตารางที่ 4.4 ตัววัดแผนภาพที่ถูกนำเข้าและออกจากโมเดลสำหรับโปรแกรมด้านการคำนวณ

Variables Entered/Removed ^a			
Model	Variables Entered	Variables Removed	Method
1	CLASSIFI	.	Stepwise (Criteria: Probability-of-F-to-enter $\leq$.050, Probability-of-F-to-remove $\geq$.100).
2	NUMOPS	.	Stepwise (Criteria: Probability-of-F-to-enter $\leq$.050, Probability-of-F-to-remove $\geq$.100).
3	MSGSELF	.	Stepwise (Criteria: Probability-of-F-to-enter $\leq$.050, Probability-of-F-to-remove $\geq$.100).
4	NUMANC	.	Stepwise (Criteria: Probability-of-F-to-enter $\leq$.050, Probability-of-F-to-remove $\geq$.100).
5	ATTRINH	.	Stepwise (Criteria: Probability-of-F-to-enter $\leq$.050, Probability-of-F-to-remove $\geq$.100).
6	MSGSENT	.	Stepwise (Criteria: Probability-of-F-to-enter $\leq$.050, Probability-of-F-to-remove $\geq$.100).
7	.	MSGSELF	Stepwise (Criteria: Probability-of-F-to-enter $\leq$.050, Probability-of-F-to-remove $\geq$.100).
8	NUMDESC	.	Stepwise (Criteria: Probability-of-F-to-enter $\leq$.050, Probability-of-F-to-remove $\geq$.100).

a. Dependent Variable: STAB

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.5 ค่าสัมประสิทธิ์ถดถอยเชิงส่วนของตัววัดแผนภาพสำหรับโปรแกรมด้านการคำนวณ

Coefficients ^a									
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Correlations		
		B	Std. Error	Beta			Zero-order	Partial	Part
1	(Constant)	.985423896	.002		445.713	.000			
	CLASSIFI	-.0017992746	.000	-.752	-14.870	.000	-.752	-.752	-.752
2	(Constant)	.9782425145	.002		479.907	.000			
	CLASSIFI	-.0030540219	.000	-1.276	-17.055	.000	-.752	-.795	-.724
	NUMOPS	.0036887373	.000	.637	8.507	.000	-.415	.548	.361
3	(Constant)	.9779010944	.002		490.439	.000			
	CLASSIFI	-.0024802791	.000	-1.036	-9.625	.000	-.752	-.596	-.399
	NUMOPS	.0035784253	.000	.618	8.420	.000	-.415	.545	.349
	MSGSELF	-.0001245029	.000	-.257	-3.031	.003	-.732	-.228	-.126
4	(Constant)	.9836578625	.003		297.219	.000			
	CLASSIFI	-.0024050305	.000	-1.005	-9.348	.000	-.752	-.586	-.383
	NUMOPS	.0034226689	.000	.591	8.025	.000	-.415	.528	.329
	MSGSELF	-.0001414558	.000	-.292	-3.419	.001	-.732	-.256	-.140
	NUMANC	-.0037724296	.002	-.094	-2.166	.032	.056	-.165	-.089
5	(Constant)	.9806250170	.003		292.306	.000			
	CLASSIFI	-.0024862532	.000	-1.039	-9.882	.000	-.752	-.609	-.394
	NUMOPS	.0034996922	.000	.604	8.419	.000	-.415	.547	.336
	MSGSELF	-.0001239585	.000	-.256	-3.051	.003	-.732	-.230	-.122
	NUMANC	-.0059349366	.002	-.148	-3.257	.001	.056	-.245	-.130
	ATTRINH	.0071093785	.002	.143	3.224	.002	.181	.243	.129
6	(Constant)	.9824474174	.003		288.910	.000			
	CLASSIFI	-.0028729804	.000	-1.201	-9.636	.000	-.752	-.600	-.379
	NUMOPS	.0044231429	.001	.763	7.775	.000	-.415	.518	.306
	MSGSELF	-.0000734182	.000	-.152	-1.613	.109	-.732	-.125	-.063
	NUMANC	-.0069102941	.002	-.172	-3.744	.000	.056	-.280	-.147
	ATTRINH	.0071768746	.002	.145	3.298	.001	.181	.249	.130
	MSGSENT	-.0007272716	.000	-.156	-2.343	.020	-.196	-.179	-.092
7	(Constant)	.9823345505	.003		287.555	.000			
	CLASSIFI	-.0032611976	.000	-1.363	-18.451	.000	-.752	-.820	-.730
	NUMOPS	.0047950737	.001	.828	9.176	.000	-.415	.580	.363
	NUMANC	-.0069090017	.002	-.172	-3.725	.000	.056	-.278	-.147
	ATTRINH	.0076116992	.002	.153	3.508	.001	.181	.263	.139
8	(Constant)	.9805660980	.003		283.415	.000			
	CLASSIFI	-.0032740467	.000	-1.368	-18.750	.000	-.752	-.825	-.732
	NUMOPS	.0048858207	.001	.843	9.441	.000	-.415	.592	.369
	NUMANC	-.0064901626	.002	-.162	-3.526	.001	.056	-.265	-.138
	ATTRINH	.0080432844	.002	.162	3.740	.000	.181	.280	.146
	MSGSENT	-.0010171931	.000	-.218	-3.737	.000	-.196	-.279	-.146
	NUMDESC	.0005569513	.000	.092	2.295	.023	.067	.176	.030

a. Dependent Variable: STAB

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.6 ตัววัดแผนภาพที่ไม่ได้ถูกนำมาใช้ สำหรับโปรแกรมด้านการคำนวณ

Excluded Variables ^a							
Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance	
1	NUMASS_P	.214 ^a	4.038	.000	.297	.837	
	NUMASS_U	.060 ^a	1.076	.284	.082	.822	
	NOP	-.071 ^a	-1.326	.187	-.101	.899	
	NUMATT	.225 ^a	3.259	.001	.243	.508	
	NUMOPS	.637 ^a	8.507	.000	.548	.322	
	NUMRUBOP	.675 ^a	8.155	.000	.531	.264	
	NESTING	.060 ^a	1.358	.176	.104	1.000	
	FINPL	.254 ^a	4.972	.000	.357	.861	
	NOC	.075 ^a	1.489	.138	.114	.999	
	NUMDESC	.074 ^a	1.462	.146	.112	1.000	
	NUMANC	-.120 ^a	-2.346	.020	-.178	.949	
	DIT	-.120 ^a	-2.346	.020	-.178	.949	
	OLD	.047 ^a	.938	.350	.072	.999	
	OPSNH	-.112 ^a	-2.171	.031	-.165	.945	
	ATTRNH	.074 ^a	1.463	.145	.112	.975	
	EC_PAR	.264 ^a	5.226	.000	.373	.865	
	IC_PAR	.226 ^a	4.519	.000	.328	.914	
	MSGSENT	.140 ^a	2.704	.008	.204	.822	
	MSGRECV	.316 ^a	5.847	.000	.410	.732	
	MSGSELF	-.318 ^a	-3.168	.002	-.237	.240	
2	NUMASS_P	-.125 ^b	-1.928	.056	-.147	.425	
	NUMASS_U	-.042 ^b	-.864	.389	-.067	.771	
	NOP	.020 ^b	.434	.665	.033	.849	
	NUMATT	-.064 ^b	-.914	.362	-.070	.362	
	NUMRUBOP	.301 ^b	1.944	.054	.148	7.396E-02	
	NESTING	.045 ^b	1.070	.286	.082	.996	
	FINPL	.060 ^b	1.092	.276	.084	.606	
	NOC	.066 ^b	1.555	.122	.119	.999	
	NUMDESC	.074 ^b	1.750	.082	.134	1.000	
	NUMANC	-.066 ^b	-1.500	.135	-.115	.927	
	DIT	-.066 ^b	-1.500	.135	-.115	.927	
	OLD	.046 ^b	1.077	.283	.083	.999	
	OPSNH	-.038 ^b	-.848	.398	-.065	.906	
	ATTRNH	.111 ^b	2.630	.009	.199	.976	
	EC_PAR	.077 ^b	1.434	.153	.110	.613	
	IC_PAR	.017 ^b	-.293	.770	-.023	.555	
	MSGSENT	-.172 ^b	-2.913	.004	-.219	.494	
	MSGRECV	-.134 ^b	-1.465	.145	-.112	.213	
	MSGSELF	-.257 ^b	-3.031	.003	-.228	.229	
	3	NUMASS_P	-.116 ^c	-1.833	.069	-.140	.424
NUMASS_U		-.034 ^c	-.728	.467	-.056	.769	
NOP		-.014 ^c	-.290	.766	-.023	.799	
NUMATT		-.053 ^c	-.774	.440	-.060	.361	
NUMRUBOP		.100 ^c	.572	.568	.044	5.648E-02	
NESTING		.059 ^c	1.419	.158	.109	.985	
FINPL		-.009 ^c	-.144	.885	-.011	.503	
NOC		.060 ^c	1.449	.149	.111	.996	
NUMDESC		.068 ^c	1.652	.100	.127	.998	
NUMANC		-.094 ^c	-2.166	.032	-.165	.894	
DIT		-.094 ^c	-2.166	.032	-.165	.894	
OLD		.058 ^c	1.394	.165	.107	.990	
OPSNH		-.049 ^c	-1.115	.266	-.086	.900	
ATTRNH		.090 ^c	2.118	.036	.162	.933	
EC_PAR		.035 ^c	.637	.525	.049	.565	
IC_PAR		.006 ^c	.111	.911	.009	.545	
MSGSENT		-.110 ^c	-1.622	.107	-.125	.369	
MSGRECV		-.135 ^c	-1.512	.132	-.116	.213	
4		NUMASS_P	-.142 ^d	-2.249	.026	-.172	.413
		NUMASS_U	-.088 ^d	-1.736	.084	-.134	.650
	NOP	.097 ^d	1.567	.119	.121	.438	
	NUMATT	-.078 ^d	-1.131	.260	-.087	.353	
	NUMRUBOP	.122 ^d	.706	.481	.055	5.525E-02	
	NESTING	.048 ^d	1.163	.247	.090	.969	
	FINPL	-.018 ^d	-.301	.764	-.023	.500	
	NOC	.048 ^d	1.155	.250	.089	.975	
	NUMDESC	.054 ^d	1.287	.200	.099	.964	
	DIT	.054 ^d	1.287	.200	.099	.964	
	OLD	.045 ^d	1.076	.283	.083	.965	
	OPSNH	-.024 ^d	-.521	.603	-.040	.825	
	ATTRNH	.143 ^d	3.224	.002	.243	.907	
	EC_PAR	.030 ^d	.548	.585	.042	.563	
	IC_PAR	.005 ^d	.093	.926	.007	.545	
	MSGSENT	-.153 ^d	-2.234	.027	-.171	.349	
	MSGRECV	-.169 ^d	-1.898	.057	-.146	.208	
	5	NUMASS_P	-.112 ^e	-1.757	.074	-.139	.422
		NUMASS_U	-.059 ^e	-1.180	.240	-.092	.627
		NOP	.036 ^e	.554	.580	.043	.369
NUMATT		-.054 ^e	-.885	.422	-.063	.348	
NUMRUBOP		.083 ^e	.491	.624	.038	5.595E-02	
	NESTING	.054 ^e	1.335	.184	.103	.967	

ตารางที่ 4.6 ตัววัดแผนภาพที่ไม่ได้ถูกนำมาใช้ สำหรับโปรแกรมด้านการคำนวณ (ต่อ)

	IFMPL	-.004 ^a	-.062	.951	-.005	.497
	NOC	.060 ^a	1.488	.130	.115	.967
	NUMDESC	.068 ^a	1.673	.096	.129	.954
	DIT	.	.	.	.	.000
	CLD	.060 ^a	1.477	.142	.114	.953
	OPSHH	-.064 ^a	-1.407	.161	-.109	.771
	EC_PAR	.045 ^a	.844	.400	.066	.559
	IC_PAR	.019 ^a	.399	.720	.028	.541
	MSGSENT	-.156 ^a	-2.343	.020	-.179	.349
	MSGRECV	-.159 ^a	-1.827	.070	-.141	.208
6	NUMASS_P	-.077 ^f	-1.199	.232	-.093	.370
	NUMASS_U	-.065 ^f	-1.301	.195	-.101	.625
	NOP	.046 ^f	.727	.468	.057	.387
	NUMATT	-.075 ^f	-1.118	.265	-.087	.343
	NUMPUBOP	.048 ^f	.289	.773	.023	5.554E-02
	NESTING	.042 ^f	1.034	.303	.080	.949
	IFMPL	.027 ^f	.399	.691	.031	.478
	NOC	.080 ^f	1.980	.049	.153	.935
	NUMDESC	.084 ^f	2.074	.040	.160	.933
	DIT	.	.	.	.	.000
	CLD	.067 ^f	1.665	.098	.129	.949
	OPSHH	-.030 ^f	-.635	.527	-.049	.675
	EC_PAR	.060 ^f	1.132	.259	.088	.552
	IC_PAR	.055 ^f	.986	.325	.077	.507
	MSGRECV	-.060 ^f	-.553	.581	-.043	.134
7	NUMASS_P	-.061 ^g	-.947	.345	-.074	.378
	NUMASS_U	-.063 ^g	-1.255	.211	-.097	.626
	NOP	.059 ^g	.939	.249	.073	.394
	NUMATT	-.082 ^g	-1.219	.225	-.094	.345
	NUMPUBOP	.137 ^g	.886	.377	.069	6.531E-02
	NESTING	.033 ^g	.812	.418	.063	.965
	IFMPL	.055 ^g	1.087	.279	.084	.604
	NOC	.088 ^g	2.215	.028	.170	.960
	NUMDESC	.092 ^g	2.295	.023	.176	.955
	DIT	.	.	.	.	.000
	CLD	.066 ^g	1.641	.103	.127	.949
	OPSHH	-.017 ^g	-.354	.724	-.028	.695
	EC_PAR	.079 ^g	1.574	.117	.122	.611
	IC_PAR	.056 ^g	1.003	.317	.078	.507
	MSGRECV	.007 ^g	.072	.943	.006	.155
	MSGSELF	-.152 ^g	-1.613	.109	-.125	.175
8	NUMASS_P	-.061 ^h	-.956	.341	-.074	.378
	NUMASS_U	-.061 ^h	-1.632	.105	-.126	.612
	NOP	.070 ^h	1.130	.260	.088	.392
	NUMATT	-.079 ^h	-1.191	.235	-.093	.344
	NUMPUBOP	.122 ^h	.794	.428	.062	6.510E-02
	NESTING	.037 ^h	.931	.353	.072	.963
	IFMPL	.059 ^h	1.168	.245	.091	.604
	NOC	.034 ^h	.384	.701	.030	.200
	DIT	.	.	.	.	.000
	CLD	-.011 ^h	-.181	.856	-.014	.385
	OPSHH	.002 ^h	.034	.973	.003	.674
	EC_PAR	.037 ^h	.662	.509	.052	.497
	IC_PAR	.054 ^h	.991	.323	.077	.507
	MSGRECV	-.001 ^h	-.012	.990	-.001	.155
	MSGSELF	-.122 ^h	-1.291	.198	-.100	.171

a. Predictors in the Model: (Constant), CLASSIFI

b. Predictors in the Model: (Constant), CLASSIFI, NUMOPS

c. Predictors in the Model: (Constant), CLASSIFI, NUMOPS, MSGSELF

d. Predictors in the Model: (Constant), CLASSIFI, NUMOPS, MSGSELF, NUMANC

e. Predictors in the Model: (Constant), CLASSIFI, NUMOPS, MSGSELF, NUMANC, ATTRINH

f. Predictors in the Model: (Constant), CLASSIFI, NUMOPS, MSGSELF, NUMANC, ATTRINH, MSGSENT

g. Predictors in the Model: (Constant), CLASSIFI, NUMOPS, NUMANC, ATTRINH, MSGSENT

h. Predictors in the Model: (Constant), CLASSIFI, NUMOPS, NUMANC, ATTRINH, MSGSENT, NUMDESC

i. Dependent Variable: STAB

4.1.2 การสร้างโมเดลของกลุ่มโปรแกรมด้านการจัดการข้อความ

สำหรับการวิเคราะห์ความถดถอยเพื่อสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะของกลุ่มโปรแกรมจัดการข้อความมีรายละเอียดเช่นเดียวกับการวิเคราะห์ความถดถอยของกลุ่มโปรแกรมด้านการคำนวณ โดยตัววัดแผนภาพที่ถูกเลือกเข้ามาเป็นอันดับแรกคือ NumPubOps ซึ่งมีสัมประสิทธิ์ความสัมพันธ์เชิงส่วนเท่ากับ -0.886 จากตารางที่ 4.9 และตัววัดแผนภาพถัดไปที่ถูกนำเข้ามาในโมเดลคือ NumOps ซึ่งมีสัมประสิทธิ์ความสัมพันธ์เชิงส่วนเท่ากับ -0.545 จากตารางที่ 4.10 และเมื่อทำการคำนวณระดับความสัมพันธ์ใหม่แล้ว ตัววัดแผนภาพทั้ง 2 ยังคงมีระดับนัยสำคัญในระดับที่กำหนดไว้จึงไม่ต้องการตัดตัววัดใดทิ้งไป จากนั้นเมื่อทำการพิจารณาเพื่อเลือกตัววัดแผนภาพตัวถัดไป พบว่าตัววัดแผนภาพที่เหลืออยู่ต่างมีระดับนัยสำคัญต่ำกว่าที่กำหนดทั้งสิ้น ($\text{sig } t > .05$) ดังนั้นจึงไม่สามารถเลือกตัววัดแผนภาพใดเข้ามาได้อีก ทำให้ได้โมเดลประมาณค่าความเสถียรเชิงตรรกะของกลุ่มโปรแกรมจัดการข้อความที่ประกอบไปด้วยตัววัดแผนภาพ 2 ตัว คือ NumPubOps และ NumOps และมีสัมประสิทธิ์การตัดสินใจเชิงซ้อนเท่ากับ 0.846 จากตารางที่ 4.7 และมีสมการของโมเดลดังนี้

$$\begin{aligned} \text{ความเสถียรเชิงตรรกะของคลาส} = & 1.0426495664 - 0.0239384916\text{NumPubOps} \\ & - 0.0049619233\text{NumOps} \end{aligned}$$

จากโมเดลที่ได้จะพบว่าความเสถียรเชิงตรรกะของคลาสใดๆ ของโปรแกรมด้านการจัดการข้อความจะมีค่าน้อยเมื่อ มีจำนวนโอเปอเรเตอร์และฟับบลิกโอเปอเรเตอร์สูง

ทั้งนี้ผลที่ได้มาจากการพฤติกรรมการทำงานของโปรแกรมด้านการจัดการข้อความที่คลาสที่มีโอเปอเรเตอร์เป็นจำนวนมาก โอเปอเรเตอร์เหล่านั้นทำงานโดยการเรียกใช้คลาสอื่น ทั้งนี้จะสามารถพิจารณาได้จากลักษณะของโปรแกรมจัดการข้อความที่จะมีหน้าตาหลักสำหรับพิมพ์และแก้ไขข้อความ และคำสั่งอื่นๆ ซึ่งคำสั่งเหล่านี้จะเรียกใช้คลาสอื่นๆ อีกทีหนึ่ง ทำให้คลาสที่มีโอเปอเรเตอร์เป็นจำนวนมากมีความเสถียรต่ำ

ตารางที่ 4.7 ผลสรุปของโมเดลสำหรับโปรแกรมด้านการจัดการข้อความ

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.886 ^a	.785	.783	.048771254
2	.921 ^b	.849	.846	.041075097

a. Predictors: (Constant), NUMPUBOP

b. Predictors: (Constant), NUMPUBOP, NUMOPS

ตารางที่ 4.8 ตัววัดแผนภาพที่ถูกนำเข้าและออกจากโมเดลสำหรับโปรแกรมด้านการจัดการข้อความ

Variables Entered/Removed^a

Model	Variables Entered	Variables Removed	Method
1	NUMPUBOP	.	Stepwise (Criteria: Probability-of-F-to-enter ≤ .050, Probability-of-F-to-remove ≥ .100).
2	NUMOPS	.	Stepwise (Criteria: Probability-of-F-to-enter ≤ .050, Probability-of-F-to-remove ≥ .100).

a. Dependent Variable: STAB

ตารางที่ 4.9 ค่าสัมประสิทธิ์ถดถอยเชิงส่วนของตัววัดแผนภาพสำหรับโปรแกรมด้านการจัดการข้อความ

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Correlations		
		B	Std. Error				Zero-order	Partial	Part
1	(Constant)	1.0392770149	.006		173.265	.000			
	NUMPUBOP	-.0300788440	.002	-.886	-19.560	.000	-.886	-.886	-.886
2	(Constant)	1.0426495664	.005		205.360	.000			
	NUMPUBOP	-.0239384916	.002	-.705	-15.040	.000	-.886	-.828	-.574
	NUMOPS	-.0049619233	.001	-.311	-6.636	.000	-.721	-.545	-.253

a. Dependent Variable: STAB

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.10 ตัววัดแผนภาพที่ไม่ได้ถูกนำมาใช้สำหรับโปรแกรมด้านการจัดการข้อความ

Excluded Variables ^a						
Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics Tolerance
1	NUMASS_P	-.243 ^a	-5.481	.000	-.473	.816
	NUMASS_U	.011 ^a	.249	.804	.024	.999
	NOP	.104 ^a	2.315	.023	.221	.970
	NUMATT	-.322 ^a	-5.489	.000	-.474	.468
	NUMOPS	-.311 ^a	-6.636	.000	-.545	.662
	NESTING	.114 ^a	2.458	.016	.234	.914
	NOC	-.009 ^a	-.189	.851	-.019	.998
	NUMDESC	-.009 ^a	-.189	.851	-.019	.998
	NUMANC	.104 ^a	2.300	.023	.220	.971
	DIT	.104 ^a	2.300	.023	.220	.971
	CLD	-.009 ^a	-.189	.851	-.019	.998
	OPSINH	.104 ^a	2.300	.023	.220	.971
	EC_PAR	-.042 ^a	-.914	.363	-.089	.994
	IC_PAR	.140 ^a	3.213	.002	.300	.998
	CLASSIFI	-.172 ^a	-3.816	.000	-.350	.899
	MSGSENT	-.171 ^a	-3.994	.000	-.365	.984
	MSGRECV	-.053 ^a	-1.165	.246	-.114	.989
	MSGSELF	-.355 ^a	-6.464	.000	-.535	.489
2	NUMASS_P	.005 ^b	.062	.951	.006	.195
	NUMASS_U	.062 ^b	1.602	.112	.156	.963
	NOP	.035 ^b	.856	.394	.084	.893
	NUMATT	-.078 ^b	-.850	.397	-.083	.176
	NESTING	.046 ^b	1.114	.268	.109	.849
	NOC	-.013 ^b	-.335	.738	-.033	.998
	NUMDESC	-.013 ^b	-.335	.738	-.033	.998
	NUMANC	.032 ^b	.795	.429	.078	.890
	DIT	.032 ^b	.795	.429	.078	.890
	CLD	-.013 ^b	-.335	.738	-.033	.998
	OPSINH	.032 ^b	.795	.429	.078	.890
	EC_PAR	-.035 ^b	-.907	.367	-.089	.993
	IC_PAR	.065 ^b	1.625	.107	.158	.896
	CLASSIFI	-.020 ^b	-.389	.698	-.038	.580
	MSGSENT	-.038 ^b	-.806	.422	-.079	.666
	MSGRECV	.028 ^b	.703	.484	.069	.895
	MSGSELF	-.159 ^b	-1.315	.191	-.128	9.940E-02

a. Predictors in the Model: (Constant), NUMPUBOP

b. Predictors in the Model: (Constant), NUMPUBOP, NUMOPS

c. Dependent Variable: STAB

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.1.3 การสร้างโมเดลของกลุ่มโปรแกรมการจัดการรูปภาพ

สำหรับการวิเคราะห์ความถดถอยเพื่อสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะของกลุ่มโปรแกรมจัดการรูปภาพ ตัววัดแผนภาพที่ถูกเลือกเข้ามาเป็นอันดับแรกคือ NumOps ซึ่งมีสัมประสิทธิ์ความสัมพันธ์เชิงส่วนเท่ากับ -0.690 จากตารางที่ 4.13 และตัววัดแผนภาพถัดไปที่ถูกนำเข้ามาในโมเดลคือ OpsInh ซึ่งมีสัมประสิทธิ์ความสัมพันธ์เชิงส่วนจากตารางที่ 4.14 เท่ากับ -0.514 และเมื่อทำการคำนวณระดับความสัมพันธ์ใหม่แล้ว ตัววัดแผนภาพทั้ง 2 ยังคงมีระดับนัยสำคัญในระดับที่กำหนดไว้จึงไม่ต้องทำการตัดตัววัดใดทิ้งไป จากนั้นเมื่อทำการพิจารณาเพื่อเลือกตัววัดแผนภาพจนไม่สามารถเลือกตัววัดแผนภาพใดเข้ามาได้อีกต่อไปจะได้โมเดลประมาณค่าความเสถียรเชิงตรรกะของกลุ่มโปรแกรมจัดการรูปภาพที่ประกอบไปด้วยตัววัดแผนภาพ 6 ตัว คือ NumOps OpsInh NumAnc IflImpl NumAtt และ CLD และมีสัมประสิทธิ์การตัดสินใจเชิงซ้อนจากตารางที่ 4.11 เท่ากับ 0.738 และมีสมการของโมเดลดังนี้

$$\begin{aligned} \text{ความเสถียรเชิงตรรกะของคลาส} = & 0.9957807364048 - 0.002715558649803 * \text{NumOps} \\ & - 0.01111321561789 * \text{OpsInh} + 0.06851665842496 * \text{NumAnc} \\ & - 0.07517081092635 * \text{IfIImpl} - 0.006663411472335 * \text{NumAtt} \\ & - 0.02806078126307 * \text{CLD} \end{aligned}$$

จากโมเดลที่ได้จะพบว่าความเสถียรเชิงตรรกะของคลาสใดๆ ของโปรแกรมด้านการจัดการรูปภาพจะมีค่าสูงเมื่อ มีจำนวนคลาสบรรพบุรุษมาก ในขณะที่ความเสถียรเชิงตรรกะของคลาสใดๆ จะมีค่าต่ำเมื่อ มีจำนวนโอเปอเรเตอร์มาก หรือมีจำนวนโอเปอเรเตอร์ที่รับสืบทอดมา มาก มีจำนวนคลาสอินเตอร์เฟสที่ใช้มาก มีจำนวนแอททริบิวต์มาก หรือ มีความลึกจากคลาสสู่ปลายสุดของลำดับการสืบทอดสูง โดยตัววัดที่มีผลต่อความเสถียรเชิงตรรกะมากที่สุดคือ จำนวนคลาสอินเตอร์เฟสที่ใช้

ทั้งนี้ผลที่ได้มาจากพฤติกรรมการทำงานของโปรแกรมด้านการจัดการรูปภาพเช่น คลาสที่มีจำนวนบรรพบุรุษมากเป็นคลาสที่มีความเสถียรสูง เนื่องจากคลาสเหล่านั้นเป็นคลาสเฉพาะทางที่สืบทอดคุณสมบัติทั่วไปมาจากคลาสบรรพบุรุษ และมีความสามารถที่เฉพาะเจาะจงมากขึ้น ใ้ให้คลาสอื่นเรียกใช้ โดยไม่ได้เรียกใช้คลาสอื่น เป็นต้น นอกจากนี้ผลที่ได้ในบางส่วนยังสอดคล้องกับทฤษฎีของความเสถียรเชิงตรรกะโดยตรงเช่น คลาสที่มีจำนวนคลาสอินเตอร์เฟสที่ใช้มากจะมีความเสถียรเชิงตรรกะต่ำ เพราะการมีคลาสอินเตอร์เฟสมากทำให้โอกาสในการได้รับผลกระทบเมื่อเกิดการเปลี่ยนแปลงที่อินเตอร์เฟสสูงตามด้วย เป็นต้น

ตารางที่ 4.11 ผลสรุปของโมเดลสำหรับโปรแกรมด้านการจัดการรูปภาพ

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.690 ^a	.476	.470	6.373E-02
2	.784 ^b	.615	.605	5.499E-02
3	.819 ^c	.670	.658	5.119E-02
4	.839 ^d	.704	.689	4.881E-02
5	.860 ^e	.739	.722	4.614E-02
6	.870 ^f	.757	.738	4.478E-02

a. Predictors: (Constant), NUMOPS

b. Predictors: (Constant), NUMOPS, OPSINH

c. Predictors: (Constant), NUMOPS, OPSINH, NUMANC

d. Predictors: (Constant), NUMOPS, OPSINH, NUMANC, IFIMPL

e. Predictors: (Constant), NUMOPS, OPSINH, NUMANC, IFIMPL, NUMATT

f. Predictors: (Constant), NUMOPS, OPSINH, NUMANC, IFIMPL, NUMATT, CLD

ตารางที่ 4.12 ตัววัดแผนภาพที่ถูกนำเข้าและออกจากโมเดลสำหรับโปรแกรมด้านการจัดการรูปภาพ

Variables Entered/Removed^a

Model	Variables Entered	Variables Removed	Method
1	NUMOPS	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	OPSINH	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	NUMANC	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
4	IFIMPL	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
5	NUMATT	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
6	CLD	.	Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: STAB

ตารางที่ 4.13 ค่าสัมประสิทธิ์ถดถอยเชิงส่วนของตัววัดแผนภาพสำหรับโปรแกรมด้านการจัดการ
รูปภาพ

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Correlations		
		B	Std. Error	Beta			Zero-order	Partial	Part
1	(Constant)	.9901530711	.009		108.686	.000			
	NUMOPS	-.0076824350	.001	-.690	-8.631	.000	-.690	-.690	-.690
2	(Constant)	1.0037313726	.008		121.828	.000			
	NUMOPS	-.0065689803	.001	-.590	-8.261	.000	-.690	-.676	-.570
	OPSINH	-.0050468901	.001	-.386	-5.400	.000	-.539	-.514	-.372
3	(Constant)	.9933171736	.008		121.466	.000			
	NUMOPS	-.0053758242	.001	-.483	-6.649	.000	-.690	-.597	-.427
	OPSINH	-.0114096361	.002	-.872	-5.882	.000	-.539	-.549	-.378
	NUMANC	.0674781379	.018	.526	3.670	.000	-.252	.380	.236
4	(Constant)	.9972917329	.008		126.079	.000			
	NUMOPS	-.0051616642	.001	-.464	-6.667	.000	-.690	-.600	-.408
	OPSINH	-.0110198897	.002	-.842	-5.943	.000	-.539	-.556	-.364
	NUMANC	.0625374992	.018	.487	3.551	.001	-.252	.371	.217
	IFIMPL	-.0585803113	.020	-.186	-2.997	.004	-.286	-.319	-.183
5	(Constant)	.9936906900	.008		131.451	.000			
	NUMOPS	-.0031761431	.001	-.285	-3.323	.001	-.690	-.352	-.192
	OPSINH	-.0104998840	.002	-.802	-5.966	.000	-.539	-.560	-.345
	NUMANC	.0622622888	.017	.485	3.741	.000	-.252	.390	.216
	IFIMPL	-.0772680897	.019	-.246	-3.991	.000	-.286	-.412	-.231
	NUMATT	-.0051907076	.002	-.269	-3.229	.002	-.601	-.343	-.187
6	(Constant)	.57807364	.007		134.780	.000			
	NUMOPS	-.0027155586	.001	-.244	-2.867	.005	-.690	-.311	-.161
	OPSINH	-.0111132156	.002	-.849	-6.434	.000	-.539	-.591	-.361
	NUMANC	.0685166584	.016	.534	4.187	.000	-.252	.431	.235
	IFIMPL	-.0751708109	.019	-.239	-3.996	.000	-.286	-.414	-.224
	NUMATT	-.0066634115	.002	-.290	-3.561	.001	-.601	-.376	-.200
	Q.D	-.0280607813	.012	-.139	-2.408	.018	-.168	-.265	-.135

a. Dependent Variable: STAB

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.14 ตัววัดแผนภาพที่ไม่ได้ถูกนำมาใช้สำหรับโปรแกรมด้านการจัดการรูปภาพ

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics
						Tolerance
1	NUMASS_P	.029 ^a	.349	.728	.039	.912
	NUMASS_U	.144 ^a	1.015	.073	.198	.987
	NOP	-.272 ^a	-3.603	.001	-.372	.976
	NUMATT	-.252 ^a	-2.409	.018	-.259	.550
	NUMPUBOP	-.166 ^a	-1.162	.248	-.128	.312
	NESTING	.108 ^a	1.304	.196	.143	.924
	IFIMPL	-.195 ^a	-2.494	.015	-.267	.981
	NOC	-.045 ^a	-.551	.583	-.061	.984
	NUMDESC	-.026 ^a	-.321	.749	-.036	.995
	NUMANC	-.227 ^a	-2.974	.004	-.314	.999
	DIT	-.227 ^a	-2.974	.004	-.314	.999
	CLD	-.080 ^a	-.989	.325	-.109	.983
	OPSDINH	-.386 ^a	-5.400	.000	-.514	.933
	ATTRINH	-.392 ^a	-4.565	.000	-.452	.697
	EC_ATT	.016 ^a	.194	.847	.022	.997
	IC_ATT	.004 ^a	.049	.961	.005	.953
	EC_PAR	.016 ^a	.194	.847	.021	.992
	IC_PAR	.034 ^a	.423	.673	.047	.995
	CLASSIFI	.601 ^a	2.686	.009	.286	.119
	MSGSENT	.040 ^a	.421	.675	.047	.727
	MSGRECV	.250 ^a	3.127	.002	.328	.906
	MSGSELF	-.236 ^a	-1.657	.101	-.181	.308
2	NUMASS_P	.114 ^b	1.559	.123	.172	.874
	NUMASS_U	.159 ^b	2.356	.021	.255	.985
	NOP	.394 ^b	2.445	.017	.264	.173
	NUMATT	-.162 ^b	-1.731	.087	-.190	.530
	NUMPUBOP	.003 ^b	.023	.982	.003	.292
	NESTING	.016 ^b	.221	.825	.025	.871
	IFIMPL	-.207 ^b	-3.125	.002	-.330	.980
	NOC	-.075 ^b	-1.070	.288	-.119	.977
	NUMDESC	-.061 ^b	-.873	.385	-.097	.986
	NUMANC	.526 ^b	3.670	.000	.380	.201
	DIT	.523 ^b	3.670	.000	.380	.201
	CLD	-.093 ^b	-1.344	.183	-.149	.982
	ATTRINH	-.217 ^b	-2.296	.024	-.249	.504
	EC_ATT	-.006 ^b	-.083	.934	-.009	.994
	IC_ATT	-.047 ^b	-.660	.511	-.074	.936
	EC_PAR	.023 ^b	.337	.737	.038	.992
	IC_PAR	.130 ^b	1.859	.067	.204	.939
	CLASSIFI	.426 ^b	2.140	.035	.233	.115
	MSGSENT	.046 ^b	.565	.574	.063	.727
	MSGRECV	.193 ^b	2.739	.008	.293	.884
	MSGSELF	-.243 ^b	-1.991	.050	-.217	.308
3	NUMASS_P	.089 ^c	1.300	.198	.145	.865
	NUMASS_U	.138 ^c	2.174	.033	.238	.977
	NOP	.080 ^c	.410	.683	.046	.109
	NUMATT	-.170 ^c	-1.957	.054	-.215	.530
	NUMPUBOP	.040 ^c	.330	.742	.037	.290
	NESTING	.069 ^c	.982	.329	.110	.837
	IFIMPL	-.186 ^c	-2.997	.004	-.319	.972
	NOC	-.120 ^c	-1.850	.068	-.204	.947
	NUMDESC	-.082 ^c	-1.267	.209	-.141	.979
	DIT	. ^c	.	.	.	.000
	CLD	-.132 ^c	-2.055	.043	-.225	.959
	ATTRINH	-.123 ^c	-1.288	.201	-.143	.451
	EC_ATT	.003 ^c	.044	.965	.005	.993
	IC_ATT	-.069 ^c	-1.038	.302	-.116	.929
	EC_PAR	.038 ^c	.563	.562	.065	.986
	IC_PAR	.115 ^c	1.755	.083	.194	.935
	CLASSIFI	.371 ^c	1.987	.050	.218	.114
	MSGSENT	.084 ^c	1.105	.273	.123	.714
	MSGRECV	.182 ^c	2.768	.007	.297	.882
	MSGSELF	-.153 ^c	-1.295	.199	-.144	.292

ตารางที่ 4.14 ตัววัดแผนภาพที่ไม่ได้ถูกนำมาใช้สำหรับโปรแกรมด้านการจัดการรูปภาพ (ต่อ)

4	NUMASS_P	.113 ^d	1.731	.087	.192	.853
	NUMASS_U	.127 ^d	2.084	.040	.230	.973
	NOP	.050 ^d	.486	.628	.055	.109
	NUMATT	-.269 ^d	-3.229	.002	-.343	.482
	NUMPUBOP	.043 ^d	.376	.708	.043	.290
	NESTING	.041 ^d	.609	.544	.069	.821
	NOC	-.099 ^d	-1.573	.120	-.175	.933
	NUMDESC	-.064 ^d	-1.025	.308	-.115	.969
	DIT	. ^d	.	.	.	.000
	CLD	-.118 ^d	-1.908	.060	-.211	.953
	ATTRINH	-.184 ^d	-2.018	.047	-.223	.433
	EC_ATT	-.003 ^d	-.048	.962	-.005	.992
	IC_ATT	-.081 ^d	-1.281	.204	-.144	.926
	EC_PAR	.019 ^d	.310	.757	.035	.978
	IC_PAR	.115 ^d	1.846	.069	.205	.935
	CLASSIFI	.311 ^d	1.727	.088	.192	.113
	MSGSENT	.100 ^d	1.379	.172	.154	.711
	MSGRECV	.159 ^d	2.493	.015	.272	.867
	MSGSELF	-.222 ^d	-1.961	.053	-.217	.282
5	NUMASS_P	.107 ^a	1.727	.088	.193	.853
	NUMASS_U	.106 ^a	1.811	.074	.202	.959
	NOP	-.226 ^a	-1.141	.257	-.129	8.511E-02
	NUMPUBOP	.023 ^a	.211	.834	.024	.289
	NESTING	.058 ^a	.912	.365	.103	.815
	NOC	-.124 ^a	-2.104	.039	-.233	.919
	NUMDESC	-.082 ^a	-1.397	.167	-.157	.961
	UIT	. ^a	.	.	.	.000
	CLD	-.139 ^a	-2.408	.018	-.265	.943
	ATTRINH	-.047 ^a	-.455	.650	-.052	.312
	EC_ATT	.002 ^a	.032	.974	.004	.991
	IC_ATT	-.085 ^a	-1.426	.158	-.160	.925
	EC_PAR	-.032 ^a	-.524	.602	-.060	.912
	IC_PAR	.090 ^a	1.494	.139	.168	.916
	CLASSIFI	.235 ^a	1.365	.176	.154	.110
	MSGSENT	.087 ^a	1.275	.206	.144	.708
	MSGRECV	.121 ^a	1.941	.056	.216	.827
	MSGSELF	-.074 ^a	-.598	.552	-.068	.221
6	NUMASS_P	.079 ^f	1.279	.205	.145	.814
	NUMASS_U	.095 ^f	1.667	.100	.188	.952
	NOP	-.063 ^f	-.303	.763	-.035	7.349E-02
	NUMPUBOP	.059 ^f	.554	.581	.063	.283
	NESTING	.041 ^f	.645	.521	.074	.803
	NOC	.064 ^f	.378	.706	.043	.113
	NUMDESC	.103 ^f	1.008	.317	.115	.301
	DIT	. ^f	.	.	.	.000
	ATTRINH	-.051 ^f	-.501	.617	-.057	.312
	EC_ATT	-.002 ^f	-.034	.973	-.004	.990
	IC_ATT	-.099 ^f	-1.707	.092	-.192	.918
	EC_PAR	-.015 ^f	-.256	.799	-.029	.899
	IC_PAR	.085 ^f	1.465	.147	.166	.915
	CLASSIFI	.153 ^f	.881	.381	.101	.105
	MSGSENT	.043 ^f	.612	.543	.070	.645
	MSGRECV	.085 ^f	1.330	.187	.151	.757
	MSGSELF	-.134 ^f	-1.102	.274	-.125	.213

a. Predictors in the Model: (Constant), NUMOPS

b. Predictors in the Model: (Constant), NUMOPS, OPSINH

c. Predictors in the Model: (Constant), NUMOPS, OPSINH, NUMANC

d. Predictors in the Model: (Constant), NUMOPS, OPSINH, NUMANC, IFIMPL

e. Predictors in the Model: (Constant), NUMOPS, OPSINH, NUMANC, IFIMPL, NUMATT

f. Predictors in the Model: (Constant), NUMOPS, OPSINH, NUMANC, IFIMPL, NUMATT, CLD

g. Dependent Variable: STAB

4.2 การประเมินผลโมเดลประมาณค่าความเสถียรเชิงตรรกะ

หลังจากได้โมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของคลาสสำหรับโปรแกรมทั้ง 3 ประเภทแล้ว จากนั้นทำการประเมินผลโมเดลที่ได้ในแต่ละประเภท เพื่อวัดระดับความคลาดเคลื่อนของโมเดลที่เกิดขึ้น ซึ่งวิธีการประเมินผลโมเดลจะใช้วิธีการทำนายที่ระดับแอล ซึ่งมีรายละเอียดในบทที่ 2

สำหรับข้อมูลที่จะนำมาใช้ในการประเมินผลโมเดลนั้น จะอาศัยชุดโปรแกรมสำหรับประเมินผลโมเดล ซึ่งเป็นชุดโปรแกรมภาษาจาวาที่ไม่ได้อยู่ในชุดโปรแกรมสำหรับการพัฒนา และเช่นเดียวกับชุดโปรแกรมสำหรับพัฒนา ชุดโปรแกรมสำหรับประเมินผลจะแบ่งออกเป็น 3 ประเภทเช่นกัน โดยรายชื่อโปรแกรมและจำนวนคลาสของแต่ละโปรแกรมแสดงอยู่ในตารางที่ 4.15

ตารางที่ 4.15 รายชื่อโปรแกรมในชุดโปรแกรมสำหรับประเมินผล

ชื่อโปรแกรม	จำนวนคลาส	รายละเอียด
โปรแกรมด้านการคำนวณ		
JCalC	5	เป็นโปรแกรมเครื่องคิดเลขธรรมดาและทางวิทยาศาสตร์
Java Calculator Suite	19	เป็นชุดโปรแกรมรวบรวมการคำนวณแบบต่างๆ
โปรแกรมด้านการจัดการข้อความ		
TWEdit	4	เป็นโปรแกรมจัดการข้อความขนาดเล็กมีความสามารถเหมือน notepad
JPad	5	เป็นโปรแกรมที่ลอกแบบมาจาก notepad
โปรแกรมด้านการจัดการภาพ		
JPhotoTweek	5	เป็นโปรแกรมตกแต่งรูปภาพอย่างง่าย สามารถทำการแก้ไขแสง สี และอื่นๆ ได้
Imgen	10	เป็นโปรแกรมที่สามารถสร้างรูปภาพผ่านทางคำสั่งของผู้ใช้

โดยขั้นตอนในการประเมินผลโมเดลจะมีขั้นตอนคล้ายกับการสร้างโมเดลดังนี้

1. วัดค่าความเสถียรเชิงตรรกะของแต่ละคลาสจากซอร์สโคด
2. แปลงกลับแผนภาพคลาสและซีเควนซ์จากซอร์สโคด
3. วัดค่าตัววัดแผนภาพจากแผนภาพที่แปลงกลับ
4. ประมาณค่าความเสถียรเชิงตรรกะของคลาสตามโมเดลที่ได้จากหัวข้อ 4.1
5. ประเมินผลด้วยวิธีการทำนายที่ระดับแอล

สำหรับวิธีการประเมินผลด้วยการทำนายที่ระดับแอลจะเริ่มจากการหาขนาดความผิดพลาดสัมพัทธ์ (MRE) ของแต่ละคลาสตามสมการในหัวข้อที่ 3.5 โดยมีตัวอย่างดังนี้

คลาส `filterObject` ของโปรแกรม `JPhotoTweek` มีค่าความเสถียรเชิงตรรกะที่ได้จากการคำนวณเท่ากับ 0.9017094 และมีค่าความเสถียรเชิงตรรกะที่ได้จากการประมาณค่าเท่ากับ 0.882595 ดังนั้นขนาดความผิดพลาดสัมพัทธ์ของคลาส `filterObject` มีค่าเป็น

$$= \left| \frac{\text{ค่าความเสถียรเชิงตรรกะจากซอร์สโคด} - \text{ค่าประมาณความเสถียรเชิงตรรกะจากแผนภาพ}}{\text{ค่าคำนวณความเสถียรเชิงตรรกะจากซอร์ส}} \right|$$

$$= \left| \frac{0.9017094 - 0.882595}{0.901709} \right|$$

$$= 0.019114$$

หลังจากได้ทำการคำนวณขนาดความผิดพลาดสัมพัทธ์ของทุกคลาสในแต่ละกลุ่มของโปรแกรมที่ใช้ในการประเมินผลโมเดลประมาณค่าความเสถียรเชิงตรรกะแล้ว จึงทำการคำนวณค่าการทำนายที่ระดับแอลตามระดับที่สนใจ โดยทำการนับจำนวนคลาสที่มีค่าขนาดความผิดพลาดสัมพัทธ์น้อยกว่าหรือเท่ากับค่าของระดับแอลที่ต้องการเช่นผลการทำนายที่ระดับ 0.25 ของโปรแกรมด้านการจัดการข้อความ จะทำการนับจำนวนคลาสที่มีค่าขนาดความผิดพลาดสัมพัทธ์

น้อยกว่าหรือเท่ากับค่าของระดับแอลที่ต้องการเช่น ผลการทำนายที่ระดับ 0.25 ของโปรแกรมด้านการจัดการข้อความ จะทำการนับจำนวนคลาสที่มีค่าขนาดความผิดพลาดสัมพัทธ์น้อยกว่าเท่ากับ 0.25 ซึ่งมีจำนวนเท่ากับ 9 คลาสและจำนวนคลาสทั้งหมดที่นำมาประเมินมีจำนวนเท่ากับ 9 คลาส ดังนั้นผลการทำนายที่ระดับ 0.25 ของโปรแกรมด้านการจัดการข้อความมีค่าดังนี้

$$\begin{aligned}
 &= \frac{\text{จำนวนคลาสที่มีขนาดความผิดพลาดสัมพัทธ์น้อยกว่าเท่ากับ 0.25}}{\text{จำนวนคลาสทั้งหมด}} \\
 &= \frac{9}{9} \\
 &= 1
 \end{aligned}$$

โดยได้ผลลัพธ์ของโปรแกรมทั้ง 3 ประเภทดังตารางที่ 4.16 ตารางที่ 4.17 และตารางที่ 4.18 ตามลำดับ และข้อมูลที่ใช้ในการประเมินผลโมเดลแสดงอยู่ในภาคผนวก ค

ตารางที่ 4.16 ผลการประเมินโมเดลประมาณค่าความเสถียรเชิงตรรกะของชุดโปรแกรมการคำนวณ

ระดับแอล (l)	จำนวนคลาสที่ค่า MRE น้อยกว่า แอล (k)	จำนวนคลาส ทั้งหมด(n)	ค่าทำนาย (k/n)
0.10	19	24	0.7916
0.20	21	24	0.875
0.25	22	24	0.9167

ตารางที่ 4.17 ผลการประเมินโมเดลประมาณค่าความเสถียรเชิงตรรกะของชุดโปรแกรมการจัดการข้อความ

ระดับแอล (l)	จำนวนคลาสที่ค่า MRE น้อยกว่า แอล (k)	จำนวนคลาส ทั้งหมด(n)	ค่าทำนาย (k/n)
0.10	6	9	0.6667
0.20	8	9	0.8889
0.25	9	9	1.0

ตารางที่ 4.18 ผลการประเมินโมเดลประมาณค่าความเสถียรเชิงตรรกะของชุดโปรแกรมการจัดการภาพ

ระดับแอล (l)	จำนวนคลาสที่ค่า MRE น้อยกว่า แอล (k)	จำนวนคลาส ทั้งหมด(n)	ค่าทำนาย (k/n)
0.10	7	15	0.4667
0.20	13	15	0.8667
0.25	14	15	0.9333

จากตารางที่ 4.16 ผลการประเมินโมเดลสำหรับโปรแกรมด้านการคำนวณสามารถอธิบายได้ว่าการทำนายในระดับ 0.25 จะมีความคลาดเคลื่อนในการประมาณค่าที่น้อยกว่าหรือเท่ากับ 25 เปอร์เซ็นต์อยู่ 91.67 เปอร์เซ็นต์ หรือหมายความว่าใน 100 คลาสที่นำมาประมาณค่าด้วยโมเดลการประมาณค่าความเสถียรเชิงตรรกะสำหรับโปรแกรมด้านการคำนวณ จะมีคลาสที่มีค่าประมาณคลาดเคลื่อนไปจากค่าจริงน้อยกว่าหรือเท่ากับ 25 เปอร์เซ็นต์อยู่ 92 คลาสด้วยกัน

จากตารางที่ 4.17 ผลการประเมินโมเดลสำหรับโปรแกรมด้านการจัดการข้อความสามารถอธิบายได้ว่าการทำนายในระดับ 0.25 จะมีความคลาดเคลื่อนในการประมาณค่าที่น้อยกว่าหรือเท่ากับ 25 เปอร์เซ็นต์อยู่ 100 เปอร์เซ็นต์ นั่นคือการประมาณค่าความเสถียรเชิงตรรกะของคลาสของโปรแกรมด้านการจัดการข้อความ จะมีความคลาดเคลื่อนไม่เกิน 25 เปอร์เซ็นต์

จากตารางที่ 4.18 ผลการประเมินโมเดลสำหรับโปรแกรมด้านการจัดการรูปภาพสามารถอธิบายได้ว่าการทำนายในระดับ 0.25 จะมีความคลาดเคลื่อนในการประมาณค่าที่น้อยกว่าหรือเท่ากับ 25 เปอร์เซ็นต์อยู่ 93.33 เปอร์เซ็นต์ หรือหมายความว่าใน 100 คลาสที่นำมาประมาณค่าด้วยโมเดลการประมาณค่าความเสถียรเชิงตรรกะสำหรับโปรแกรมด้านการคำนวณ จะมีคลาสที่มีค่าประมาณคลาดเคลื่อนไปจากค่าจริงน้อยกว่าหรือเท่ากับ 25 เปอร์เซ็นต์อยู่ 93 คลาสด้วยกัน

จากผลการประเมินโมเดลการประมาณค่าของโปรแกรมทั้ง 3 กลุ่มพบว่า ค่าการทำนายที่ระดับแอลเท่ากับ 0.25 มีค่ามากกว่า 0.75 ทั้งสิ้น นั่นคือโมเดลการประมาณค่าของโปรแกรมทั้ง 3 กลุ่ม มีความคลาดเคลื่อนในการประมาณค่าอยู่ในระดับที่ยอมรับได้และสามารถนำไปใช้งานได้จริง และจากผลการทำนายระดับแอลสามารถนำไปใช้ประกอบการประมาณค่าความเสถียรเชิงตรรกะของคลาสได้ โดยสามารถนำผลการทำนายระดับแอลไปใช้สร้างช่วงของค่าความเสถียรเชิงตรรกะที่แท้จริงได้ เช่นจากผลการทำนายที่ระดับแอลเท่ากับ 0.25 ของโมเดลประมาณค่าความ

เสถียรเชิงตรรกะของกลุ่มโปรแกรมการจัดการข้อความ ทำให้ทราบว่าหากผลการประมาณค่าด้วยโมเดลที่ได้มีค่าความเสถียรเชิงตรรกะเท่ากับ 0.80 แล้ว ค่าความเสถียรเชิงตรรกะที่แท้จริงจะมีค่าอยู่ระหว่าง 0.64 ถึง 1 โดยหาได้จากการกำหนดให้ค่าประมาณเป็นค่าคลาดเคลื่อนมากที่สุดกับน้อยสุด

นอกจากการหาโมเดลการประมาณค่าความเสถียรเชิงตรรกะของโปรแกรมทั้ง 3 ประเภทแล้ว ผู้เขียนวิทยานิพนธ์ได้ทำการทดลองนำโปรแกรมทั้ง 3 ประเภทมารวมกันแล้วทำการหาโมเดลการประมาณค่าความเสถียรเชิงตรรกะของโปรแกรมทั้ง 3 ประเภท พบว่าโมเดลที่ได้ไม่สามารถใช้อธิบายความเสถียรเชิงตรรกะของคลาสได้ เนื่องจากค่าสัมประสิทธิ์การตัดสินใจเชิงซ้อนของโมเดลที่ได้มีค่าประมาณ 0.63 ซึ่งหมายความว่าความเสถียรเชิงตรรกะของคลาสจะขึ้นอยู่กับโมเดลที่ได้เพียง 63% ซึ่งถือว่าไม่เพียงพอ ทั้งนี้อาจมีสาเหตุมาจากโปรแกรมทั้ง 3 ประเภทต่างมีลักษณะ หรือพฤติกรรมเฉพาะกลุ่มที่เมื่อนำมารวมกันแล้วเกิดการขัดแย้งขึ้นตัวอย่างเช่น ในโมเดลของโปรแกรมด้านการคำนวณ ค่าตัววัดแผนภาพจำนวนโอเปอเรเตอร์มีผลทางบวกแก่ความเสถียรเชิงตรรกะของคลาส ทว่าในโมเดลของโปรแกรมด้านข้อความค่าตัววัดแผนภาพเดียวกันกลับมีผลทางลบ เป็นต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การออกแบบและพัฒนาเครื่องมือคำนวณความเสถียรเชิงตรรกะและเครื่องมือ ประมาณค่าความเสถียรเชิงตรรกะ

เครื่องมือคำนวณความเสถียรเชิงตรรกะของงานวิจัยนี้ได้ทำการพัฒนาขึ้น โดยการแก้ไขเครื่องมือคำนวณเสถียรภาพ Changelmpact[18] โดยแก้ไขวิธีการคำนวณ และเพิ่มส่วนต่อประสานผู้ใช้ให้ง่ายต่อการใช้งาน โดยรายละเอียดของการแก้ไขและรายละเอียดของโปรแกรมจะอยู่ในหัวข้อที่ 5.1.2

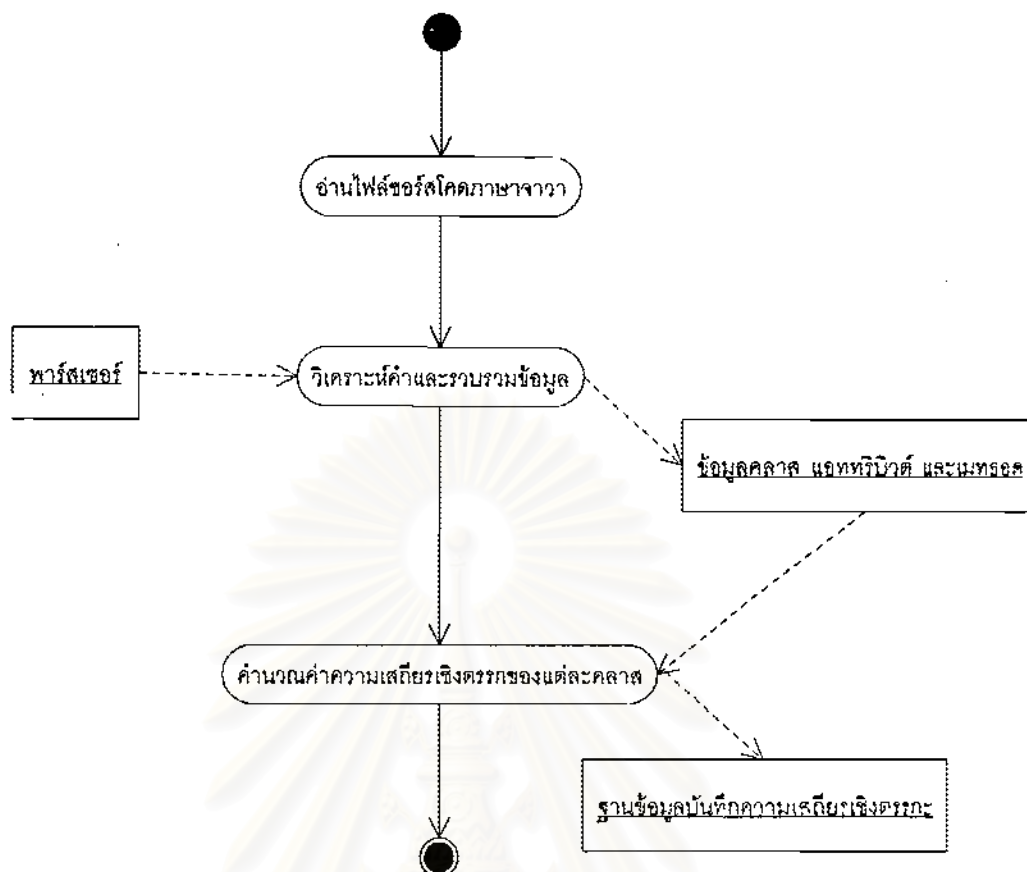
สำหรับเครื่องมือการประมาณค่าความเสถียรเชิงตรรกะจากตัววัดแผนภาพตามโมเดลที่ได้ของงานวิจัยนี้จะอาศัยโปรแกรม SDMetrics[14] ที่ใช้ในการวัดค่าตัววัดแผนภาพในขั้นตอนที่ 3.3 โดยโปรแกรม SDMetrics มีคุณสมบัติในการอ่านข้อมูลแผนภาพที่ได้รับการแปลงให้อยู่ในรูปแบบเอกซ์เอ็มแอล แล้วทำการคำนวณค่าข้อมูลต่างๆ ของแผนภาพเพื่อนำไปใช้ในการวัดค่าตัววัดแผนภาพต่อไป และอนุญาตให้ผู้ใช้สามารถทำการแก้ไขเพิ่มเติมวิธีการอ่านค่า และการคำนวณต่างๆ ได้ รายละเอียดของโปรแกรมและรายละเอียดการแก้ไขโปรแกรมอยู่ในหัวข้อที่ 5.2

5.1 การออกแบบและพัฒนาเครื่องมือคำนวณความเสถียรเชิงตรรกะ

เครื่องมือคำนวณความเสถียรเชิงตรรกะที่พัฒนาขึ้นโดยการแก้ไข และปรับปรุงเครื่องมือคำนวณเสถียรภาพ Changelmpact[18] โดยทำการแก้ไขขั้นตอนและวิธีการคำนวณให้สอดคล้องกับ รูปแบบการเปลี่ยนแปลงในตารางที่ 3.1

5.1.1 โครงสร้างของเครื่องมือคำนวณเสถียรภาพ

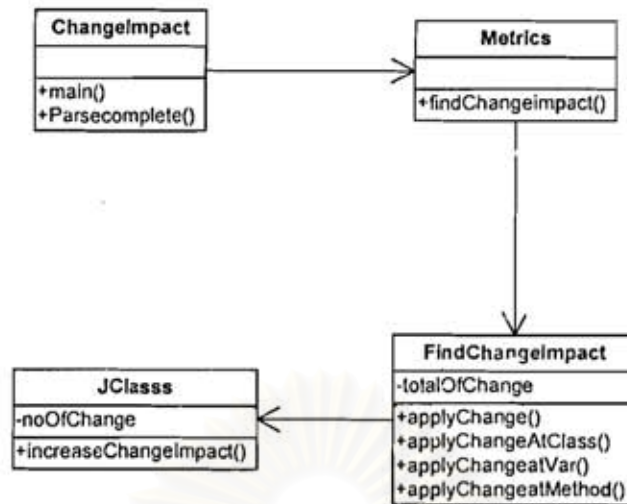
เครื่องมือคำนวณเสถียรภาพที่นำมาพัฒนา มีการทำงานโดยจะทำการอ่านไฟล์ซอร์สโค้ดภาษาจาวาที่ระบุไว้ แล้วทำการวิเคราะห์และเก็บรวบรวมข้อมูลของแต่ละคลาส โดยอาศัยพาร์เซอร์ภาษาจาวาในการวิเคราะห์แล้วจึงแยกเก็บข้อมูลของแต่ละคลาส แอพริวิวด์ และเมทอด จากนั้นจึงทำการวัดค่าความเสถียรเชิงตรรกะจากชุดข้อมูลที่รวบรวมได้ แล้วบันทึกผลลัพธ์ที่ได้ลงฐานข้อมูลที่ได้กำหนดไว้ แผนภาพกิจกรรมการทำงานของเครื่องมือคำนวณเสถียรภาพแสดงได้ดังรูปที่ 5.1



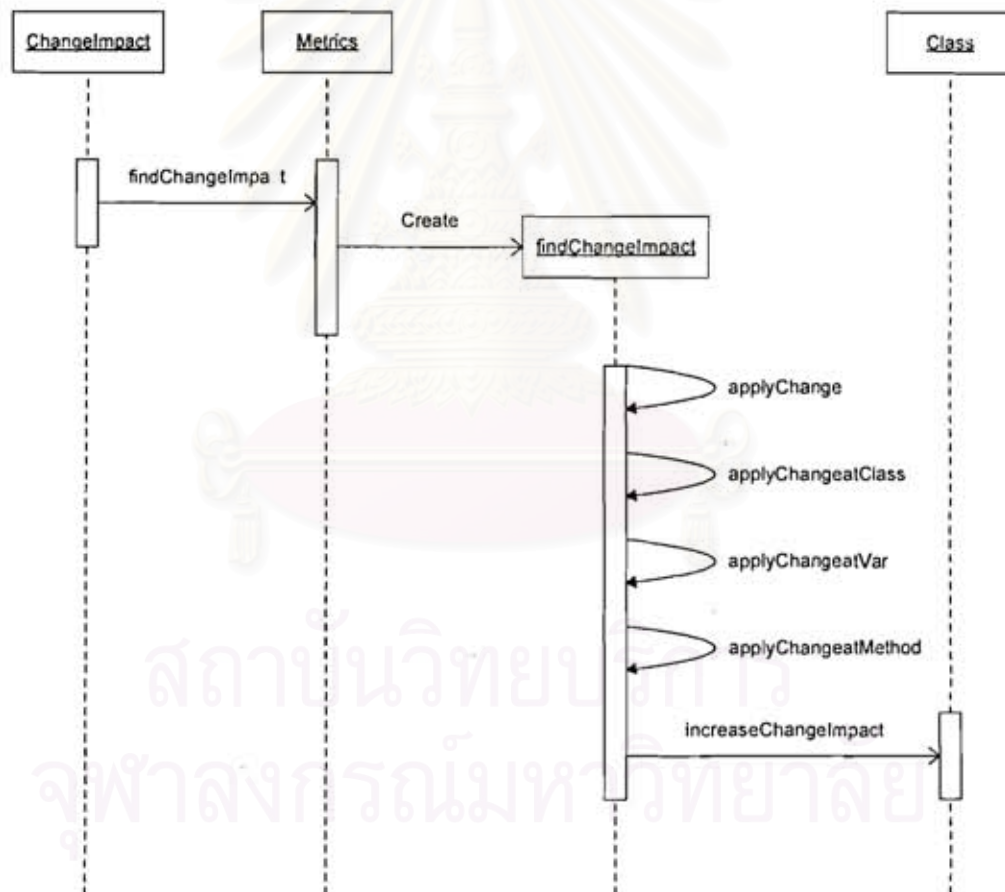
รูปที่ 5.1 การทำงานของเครื่องมือคำนวณเสถียรภาพ

งานวิจัยนี้ได้ทำการแก้ไขวิธีการในการคำนวณความเสถียรเชิงตรรกะของเครื่องมือคำนวณเสถียรภาพเดิม โดยมีรายละเอียดในหัวข้อที่ 5.1.2

รูปที่ 5.2 แสดงแผนภาพคลาสของเครื่องมือคำนวณเสถียรภาพเฉพาะในส่วนของการคำนวณค่าความเสถียร โดยมีคลาส ChangelImpact เป็นคลาสหลักเมื่อมีการเรียกใช้งานจะทำการอ่านรายชื่อไฟล์ซอร์สโค้ดจากไฟล์ที่กำหนดแล้วทำการวิเคราะห์ค่า เมื่อทำการวิเคราะห์เสร็จสิ้นจะทำการเก็บข้อมูลของคลาสทั้งหมดไว้ในคลาส metrics จากนั้นทำการตรวจสอบผลกระทบผ่านทางคลาส metrics ซึ่งจะทำการสร้างคลาส FindChangelImpact ขึ้นและทำการตรวจสอบผลกระทบพร้อมทั้งเพิ่มค่าจำนวนครั้งที่แต่ละคลาสได้รับผลกระทบ แล้วจึงทำการคำนวณความเสถียรเชิงตรรกะและทำการบันทึกลงฐานข้อมูล ขั้นตอนการทำงานเพื่อคำนวณความเสถียรเชิงตรรกะของเครื่องมือคำนวณเสถียรภาพแสดงได้โดยแผนภาพซีควเอนซ์ในรูปที่ 5.3



รูปที่ 5.2 แผนภาพคลาสของเครื่องมือคำนวณเสถียรภาพเฉพาะส่วนที่เกี่ยวข้อง



รูปที่ 5.3 แผนภาพซีควเอนซ์การทำงานของเครื่องมือคำนวณเสถียรภาพ

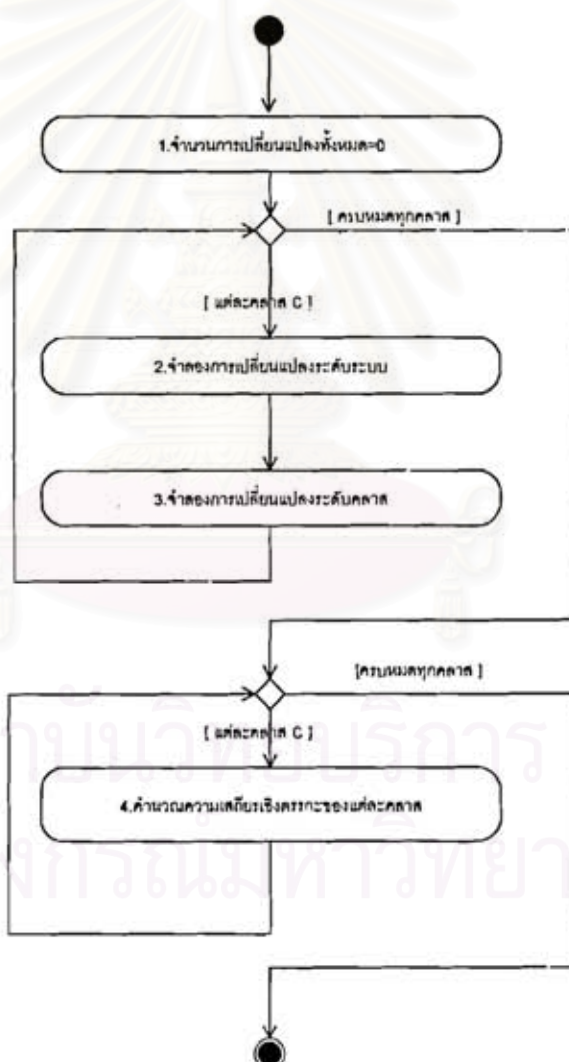
5.1.2 การพัฒนาเครื่องมือคำนวณความเสถียรเชิงตรรกะ

การพัฒนาเครื่องมือคำนวณความเสถียรเชิงตรรกะในงานวิจัยนี้แบ่งเป็น 2 ส่วนดังนี้

1. การพัฒนากระบวนการคำนวณความเสถียรเชิงตรรกะ
2. การพัฒนาส่วนต่อประสานผู้ใช้และการเลือกไฟล์ซอร์สโค้ด

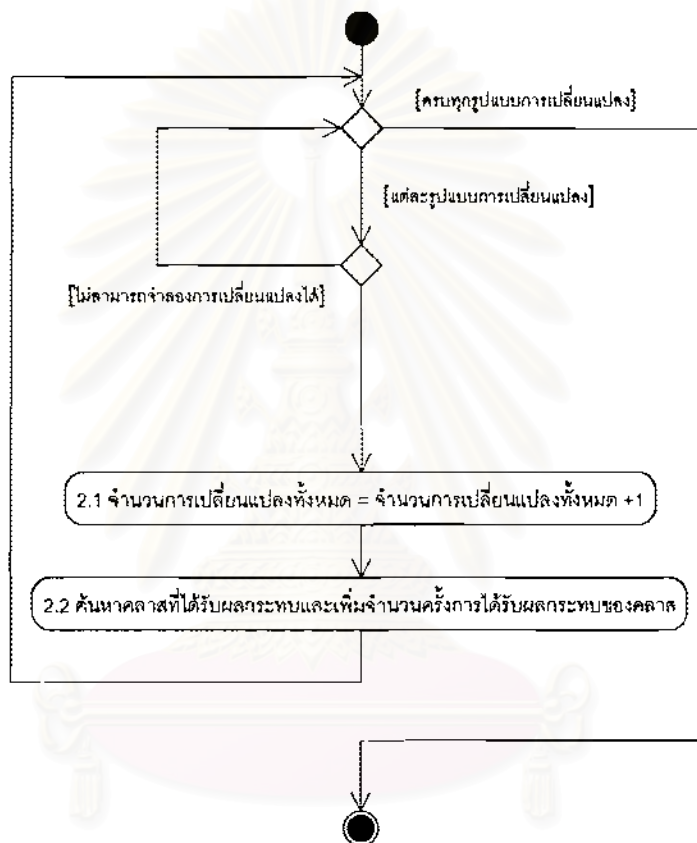
5.1.2.1 การพัฒนากระบวนการคำนวณความเสถียรเชิงตรรกะ

การพัฒนากระบวนการคำนวณความเสถียรเชิงตรรกะกระทำโดยทำการแก้ไขคลาส FindChangelImpact ของเครื่องมือคำนวณเสถียรภาพ ให้มีขั้นตอนการทำงานตามแผนภาพกิจกรรมการคำนวณความเสถียรเชิงตรรกะในรูปที่ 5.4 ในส่วนของการจำลองรูปแบบการเปลี่ยนแปลง



รูปที่ 5.4 ขั้นตอนการคำนวณความเสถียรเชิงตรรกะของเครื่องมือคำนวณความเสถียรเชิงตรรกะ

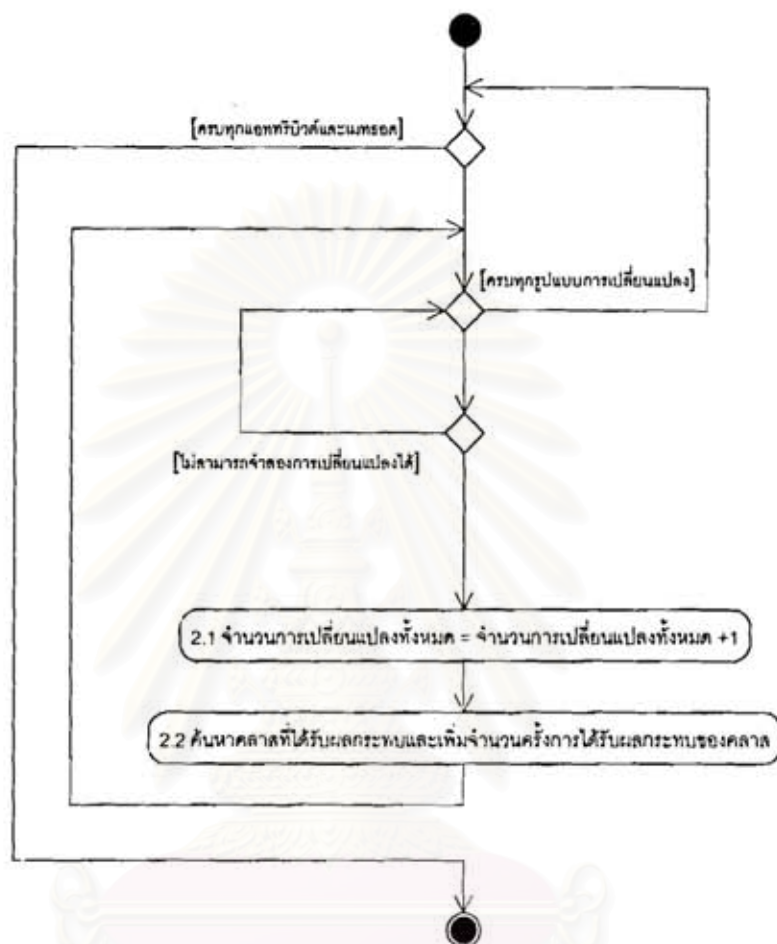
จากรูปที่ 5.4 ขั้นตอนการคำนวณความเสถียรเชิงตรรกะเริ่มโดยทำการเลือกคลาสขึ้นมาหนึ่งคลาสจากทั้งหมด จากนั้นทำการจำลองการเปลี่ยนแปลงระดับของระบบให้แก่คลาสตามการเปลี่ยนแปลงที่ 1-14 ในตารางที่ 3.1 แล้วจึงทำการจำลองการเปลี่ยนแปลงในระดับคลาสตามการเปลี่ยนแปลงที่ 15-27 ในตารางที่ 3.1 จากนั้นเลือกคลาสถัดไปมาทำการจำลองทั้งระดับของระบบและระดับคลาสเช่นเดียวกันและทำซ้ำจนครบทุกคลาส เมื่อทำการจำลองการเปลี่ยนแปลงครบทุกคลาสแล้วจึงทำการคำนวณความเสถียรเชิงตรรกะของแต่ละคลาสตามสมการในหัวข้อ 2.1.2 รูปที่ 5.5 แสดงขั้นตอนการจำลองการเปลี่ยนแปลงในระดับระบบ



รูปที่ 5.5 ขั้นตอนการจำลองการเปลี่ยนแปลงในระดับของระบบ

การเปลี่ยนแปลงระดับคลาสในรูปที่ 5.4 จะประกอบไปด้วยการเปลี่ยนแปลงที่แอททริบิวต์และการเปลี่ยนแปลงที่เมทอด ซึ่งมีลำดับการทำงานดังรูปที่ 5.6 โดยการเปลี่ยนแปลงที่แอททริบิวต์คือการเปลี่ยนแปลงที่ 15-20 และการเปลี่ยนแปลงที่เมทอดคือการเปลี่ยนแปลงที่ 21-27 ในตารางที่ 3.1 และทุกครั้งที่ทำกรจำลองการเปลี่ยนแปลงจะต้องทำการค้นหาคลาสที่ได้รับผลกระทบจากการเปลี่ยนแปลงนั้นๆ เพื่อเพิ่มค่าจำนวนครั้งที่คลาสใดๆ ได้รับ

ผลกระทบจากการเปลี่ยนแปลง นอกจากนี้ยังต้องการเพิ่มค่าจำนวนการเปลี่ยนแปลงทั้งหมด
 เปรียบเทียบจะเห็นได้จากรูปที่ 5.5 และรูปที่ 5.6



รูปที่ 5.6 ขั้นตอนการจำลองการเปลี่ยนแปลงระดับของคลาส

5.1.2.2 การพัฒนาส่วนต่อประสานผู้ใช้และการเลือกไฟล์ซอร์สโคด

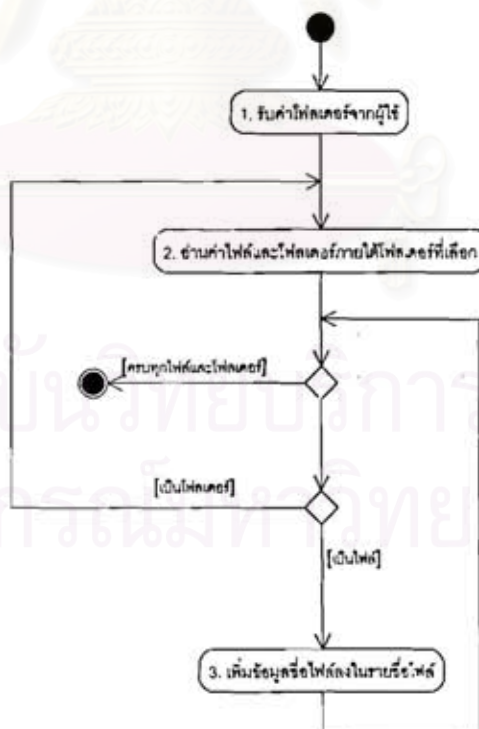
การพัฒนาส่วนต่อประสานผู้ใช้และการเลือกไฟล์ซอร์สโคดทำขึ้นเพื่ออำนวยความสะดวก
 ในการเลือกไฟล์ของซอฟต์แวร์ที่ต้องการคำนวณความเสถียรเชิงตรรกะของแต่ละคลาสจากเดิมที่
 ต้องทำการแก้ไขข้อมูลในไฟล์ config.dat โดยให้มีส่วนต่อประสานผู้ใช้และให้ผู้ใช้เลือกเพียงไดเรก
 ทอรีที่มีไฟล์ซอร์สโคดทั้งหมดอยู่ และโปรแกรมจะทำการอ่านไฟล์ในไดเรกทอรีนั้นและไดเรกทอรี
 ย่อยด้วย

ส่วนต่อประสานผู้ใช้จะมีลักษณะดังรูปที่ 5.7 โดยรายละเอียดของส่วนต่อประสานและ
 การใช้งานโปรแกรมคำนวณความเสถียรเชิงตรรกะอยู่ในภาคผนวก ง



รูปที่ 5.7 ส่วนต่อประสานผู้ใช้ของเครื่องมือคำนวณความเสถียรเชิงตรรกะ

การเลือกไฟล์ซอร์สโคดจะอาศัยการทำงานแบบวนกลับ (recursive) ดังแสดงในรูปที่ 5.8 โดยเมื่อผู้ใช้เลือกไดเรกทอรีที่ต้องการโปรแกรมจะดึงข้อมูลไฟล์และไดเรกทอรีที่อยู่ภายในไดเรกทอรีออกมาแล้วทำการตรวจสอบทีละไฟล์หรือไดเรกทอรี ถ้าเป็นไฟล์ให้ทำการเพิ่มข้อมูลในรายชื่อไฟล์ ถ้าเป็นไดเรกทอรีให้ทำการวนกลับตรวจสอบไดเรกทอรีต่อไปจนกว่าจะครบทั้งหมด



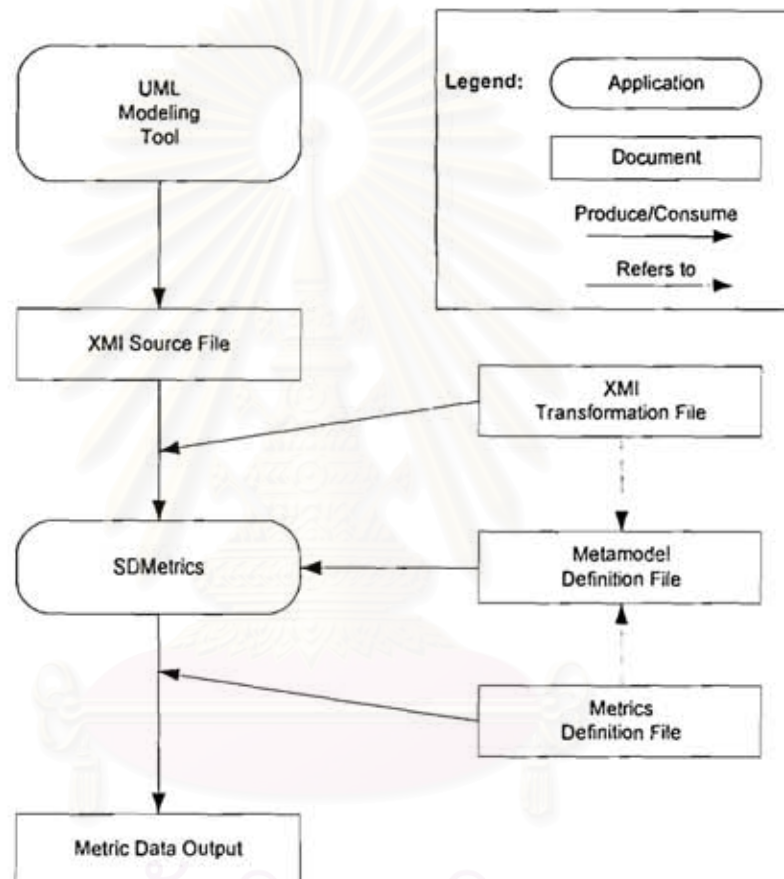
รูปที่ 5.8 ขั้นตอนการค้นหาไฟล์ทั้งหมดในไดเรกทอรีที่ต้องการ

5.2 การออกแบบและพัฒนาเครื่องมือประมาณค่าความเสถียรเชิงตรรกะ

การพัฒนาเครื่องมือคำนวณความเสถียรเชิงตรรกะของงานวิจัยนี้จะอาศัยโปรแกรม SDMetrics ซึ่งเป็นโปรแกรมที่มีความสามารถในการคำนวณค่าตัววัดแผนภาพยูเอ็มแอลที่อยู่ในรูปแบบของเอกซ์เอ็มแอลหรือเอกซ์เอ็มไอได้ และรองรับการแก้ไขเพิ่มเติมตัววัดได้

5.2.1 โครงสร้างการทำงานของโปรแกรม SDMetrics

โครงสร้างการทำงานของโปรแกรม SDMetrics แสดงได้ดังรูปที่ 5.9



รูปที่ 5.9 โครงสร้างการทำงานของโปรแกรม SDMetrics

จากรูปที่ 5.9 โปรแกรม SDMetrics จะสามารถทำการวัดค่าตัววัดแผนภาพได้โดยอาศัยข้อมูลจากไฟล์นิยาม 3 ไฟล์ได้แก่ XMI Transformation file Metamodel Definition file และ Metrics Definition file โดยโปรแกรม SDMetrics จะทำการวัดตัววัดแผนภาพโดยอ่านข้อมูลจากไฟล์เอกซ์เอ็มไอ แล้วทำการดึงข้อมูลที่รู้จักซึ่งระบุไว้ใน Metamodel Definition file โดยใช้วิธีการดึงข้อมูลตามที่ระบุไว้ใน XMI Transformation file จากนั้นทำการวัดตัววัดแผนภาพตามที่ระบุไว้

ใน Metrics Definition File จากนั้นทำการแสดงผลตัววัดแผนภาพที่วัดได้ โดยไฟล์นิยามทั้ง 3 มีรายละเอียดและโครงสร้างตามหลักภาษาเอกซ์เอ็มแอล

5.2.2 การอ่านค่าข้อมูลแผนภาพและคำนวณตัววัดแผนภาพ

การอ่านข้อมูลแผนภาพจากไฟล์เอกซ์เอ็มไอ และเอกซ์เอ็มแอลเพื่อนำมาใช้ในการคำนวณตัววัดแผนภาพนั้นสามารถอธิบายเป็นขั้นตอนได้ดังนี้

ขั้นตอนที่ 1 กำหนดข้อมูลที่ต้องการอ่าน

ขั้นตอนที่ 2 กำหนดวิธีการอ่านข้อมูลนั้นๆ

ขั้นตอนที่ 3 กำหนดวิธีการนำข้อมูลที่อ่านได้ไปใช้คำนวณ

การกำหนดข้อมูลที่ต้องการอ่านค่าในขั้นตอนที่ 1 นั้นทำได้โดยการกำหนดค่าลงในไฟล์ Metamodel Definition โดยไฟล์ Metamodel Definition คือไฟล์ที่เป็นตัวกำหนดชนิดและรายละเอียดของข้อมูลแผนภาพที่จำเป็นในการคำนวณตัววัดแผนภาพและแสดงผล ซึ่งมีรูปแบบโครงสร้างดังตารางที่ 5.1

ตารางที่ 5.1 โครงสร้างของไฟล์ Metamodel Definition

```
<sdmetricsmetamodel version="2.0" >
  <modelelement name="element1">
    <attribute name="attr1" type="data" multiplicity="one" />
    <attribute name="attr2" type="ref" multiplicity="many" />
    ..
  </modelelement>
  <modelelement name="element2" parent="element1">
    ...
  </modelelement>
  ...
</sdmetricsmetamodel>
```

จากตารางที่ 5.1 โครงสร้างของไฟล์ Metamodel definition เป็นไปตามข้อกำหนดของภาษาเอกซ์เอ็มแอล โดยต้องมีการประกาศประเภทของข้อมูลในไฟล์ และเวอร์ชันของไฟล์เสียก่อนดังในบรรทัดแรก ซึ่งเป็นการประกาศประเภทของเอกสารเป็น เอกสาร sdmetricsmetamodel เวอร์ชัน 2 บรรทัดถัดมาเป็นการประกาศองค์ประกอบของโมเดล (Model Element) หรือข้อมูล

แผนภาพที่ต้องการโดยรายละเอียดข้อมูลที่ต้องการต้องประกาศภายใต้แท็ก <modelelement> และต้องทำการกำหนดชื่อให้แก่ทุกองค์ประกอบ

องค์ประกอบโมเดลทุกองค์ประกอบจะมีข้อมูลแอททริบิวต์ขององค์ประกอบ ซึ่งจะเก็บข้อมูลขององค์ประกอบ และข้อมูลอ้างอิงกับองค์ประกอบอื่นๆ การประกาศแอททริบิวต์ขององค์ประกอบต้องอยู่ภายใต้แท็ก <attribute> ซึ่งมีแอททริบิวต์เอกซ์เอ็มแอล 3 แอททริบิวต์ดังนี้

- name คือชื่อของแอททริบิวต์ที่ประกาศ
- type คือชนิดของแอททริบิวต์มี 2 ชนิดคือ ข้อมูล (data) และข้อมูลอ้างอิง (ref)
- multiplicity คือจำนวนการเก็บข้อมูลในแต่ละแอททริบิวต์ ได้แก่ เก็บข้อมูลค่าเดียว (one) หรือหลายค่า (many)

ตัวอย่างการประกาศนิยามขององค์ประกอบแสดงในตารางที่ 5.2 ซึ่งเป็นการประกาศนิยามของโอเปอร์เรชัน

ตารางที่ 5.2 การประกาศนิยามของโอเปอร์เรชัน

```
<modelelement name="operation">
  <attribute name="id" />
  <attribute name="name" />
  <attribute name="context" type="ref" />
  <attribute name="visibility" />
</modelelement>
```

จากตารางที่ 5.2 โอเปอร์เรชันจะประกอบไปด้วยแอททริบิวต์แบบค่าเดียว 4 แอททริบิวต์ด้วยกันดังนี้

- id คือหมายเลขเอกซ์เอ็มไอของโอเปอร์เรชันแต่ละตัว
- name คือชื่อของโอเปอร์เรชัน
- context คือค่าอ้างอิงถึงองค์ประกอบที่เป็นเจ้าของโอเปอร์เรชัน
- visibility คือค่าขอบเขตของโอเปอร์เรชัน

ขั้นตอนที่ 2 ของการอ่านข้อมูลแผนภาพคือการกำหนดวิธีในการอ่าน โดยการกำหนดไฟล์ XMI Transformation ซึ่งเป็นไฟล์ที่ระบุข้อกำหนดการเปลี่ยนรูปเพื่ออ่านค่าข้อมูลที่ต้องการตามที่

ได้กำหนดไว้ในไฟล์ Metamodel Definition โดยจะทำการอ่านข้อมูลจากไฟล์แผนภาพที่แปลงให้อยู่ในรูปแบบของเอกซ์เอ็มแอล โครงสร้างของไฟล์ XMI Transformation แสดงในตารางที่ 5.3 ทั้งนี้ข้อกำหนดการเปลี่ยนรูปจะแตกต่างกันไปตามแต่ผู้กำหนด

ตารางที่ 5.3 โครงสร้างของไฟล์ XMI Transformation

```
<xmittransformations version="2.0" >
  <xmittransformation ...xmittransformation attributes... />
    <trigger ...trigger attributes... />
    <trigger ...trigger attributes... />
    ...
  </xmittransformation>
  <xmittransformation ...xmittransformation attributes... />
    <trigger ...trigger attributes... />
    ...
  </xmittransformation>
  ...
</xmittransformations>
```

จากตารางที่ 5.3 การกำหนดการเปลี่ยนรูปข้อมูลหรือการอ่านค่าข้อมูลสามารถกระทำได้โดยแท็ก <xmittransformation> ซึ่งจะเป็นการกำหนดการเปลี่ยนรูปข้อมูลขององค์ประกอบ 1 องค์ประกอบเท่านั้น ภายในแท็ก xmittransformation สามารถประกาศแอททริบิวต์ได้ดังนี้

- modelement คือชื่อขององค์ประกอบที่ต้องการเปลี่ยนรูป
- xmapattern คือรูปแบบเอกซ์เอ็มแอลขององค์ประกอบในไฟล์ข้อมูลแผนภาพที่ต้องการเปลี่ยนรูป
- recurse คือการกำหนดว่าองค์ประกอบที่ต้องการเปลี่ยนรูปสามารถเป็นเจ้าของขององค์ประกอบอื่นได้หรือไม่
- condition คือการกำหนดเงื่อนไขให้ดำเนินการเปลี่ยนรูปเมื่อเงื่อนไขถูกต้องเท่านั้น

การเปลี่ยนรูปข้อมูลขององค์ประกอบสามารถกระทำโดยอาศัยแท็ก trigger ซึ่งมีแอททริบิวต์ที่จำเป็นดังนี้

- type คือการกำหนดประเภทการเปลี่ยนรูปข้อมูล เช่น รหัสเอกซ์เอ็มไอขององค์ประกอบ หรือชื่อขององค์ประกอบ เป็นต้น
- name คือชื่อขององค์ประกอบที่ต้องการเปลี่ยนรูปตามที่จะพบได้ใน Metamodel Definition

ตัวอย่างการสร้างข้อกำหนดการเปลี่ยนรูปขององค์ประกอบโอเปอร์เรชันแสดงในตารางที่

5.4

ตารางที่ 5.4 ตัวอย่างข้อกำหนดการเปลี่ยนรูปขององค์ประกอบโอเปอร์เรชัน

```
<xmitransformation modelelement="operation"
  xmitpattern="Foundation.Core.Operation" recurse="true">
  <trigger name="id" type="attrval" attr="xmi.id" />
  <trigger name="name" type="cstext" src="Foundation.Core.ModelElementName" />
  <trigger name="visibility" type="cattrval" src="Foundation.Core.ModelElement.visibility"
    attr="xmi.value"/>
  <trigger name="context" type="gcattrval" src="Foundation.Core.Feature.owner" attr="xmi.idref"/>
</xmitransformation>
```

จากตารางที่ 5.4 องค์ประกอบโอเปอร์เรชันจะมีรูปแบบในภาษาเอกซ์เอ็มไอเป็น Foundation.Core.Operation และมีแอททริบิวต์ดังนี้

- id คือรหัสขององค์ประกอบเอกซ์เอ็มไอประจำแต่ละองค์ประกอบในข้อมูลแผนภาพโดยรับค่าจากการตั้งรหัสขององค์ประกอบโอเปอร์เรชันมาโดยตรง
- name คือชื่อของโอเปอร์เรชันโดยรับค่าจากการตั้งข้อความจากองค์ประกอบลูกของ โอเปอร์เรชัน
- visibility คือขอบเขตการเข้าถึงของโอเปอร์เรชันโดยรับค่าจากการตั้งข้อมูลแอททริบิวต์ขององค์ประกอบลูกของโอเปอร์เรชัน
- context คือรหัสขององค์ประกอบที่เป็นเจ้าของโอเปอร์เรชันโดยรับค่าจากการตั้งข้อมูล แอททริบิวต์ขององค์ประกอบหลานของโอเปอร์เรชัน

หลังจากทำการเปลี่ยนรูปข้อมูลแผนภาพให้อยู่ในรูปแบบข้อมูลที่สามารถนำไปใช้ในการคำนวณค่าตัววัดแผนภาพได้แล้ว ขั้นตอนต่อไปคือทำการกำหนดวิธีการนำข้อมูลที่ได้นำไปใช้คำนวณ

ตัววัดแผนภาพที่ต้องการ ซึ่งรายละเอียดการคำนวณตัววัดแผนภาพจะถูกกำหนดไว้ในไฟล์ metrics definition ซึ่งมีโครงสร้างดังตารางที่ 5.5

ตารางที่ 5.5 โครงสร้างรายละเอียดตัววัดแผนภาพ

```
<metric name="metricname" domain="metricdomain" category="metriccategory" internal="true/false">
  <description>Description of the metric.</description>
  <'metric definition' ...>
</metric>
```

จากตารางที่ 5.5 การกำหนดรายละเอียดของตัววัดแผนภาพจะต้องกำหนดภายใต้แท็ก <metric> ซึ่งจะมีแอททริบิวต์ดังนี้

- name คือชื่อของตัววัดแผนภาพ
- domain คือโดเมนของตัววัดแผนภาพว่าเป็นตัววัดแผนภาพในระดับใด เช่น คลาส หรือแพ็คเกจ เป็นต้น
- category คือกลุ่มประเภทของตัววัดแผนภาพที่ผู้ใช้สามารถกำหนดได้เอง เช่น ขนาด ความซับซ้อน หรือการขึ้นต่อกัน เป็นต้น
- internal คือตัวเลือกในการเลือกให้แสดงผลตัววัดแผนภาพแก่ผู้ใช้หรือไม่

แอททริบิวต์ทั้ง 4 เป็นตัวบอกคุณสมบัติของตัววัดแผนภาพ โดยมีข้อจำกัดบางประการคือ ตัววัดแผนภาพที่อยู่ในโดเมนเดียวกันไม่สามารถมีชื่อเหมือนกันได้ แต่ตัววัดที่อยู่คนละโดเมนสามารถมีชื่อเหมือนกันได้ นอกจากคุณสมบัติของตัววัดแล้วผู้ใช้อย่างยังสามารถกำหนดรายละเอียดหรือคำอธิบายของตัววัดแผนภาพได้ด้วยแท็ก <description>

รายละเอียดการคำนวณตัววัดแผนภาพจะถูกกำหนดไว้ใน <metric definition> ซึ่งจะมีแท็กที่ใช้แตกต่างกันออกไปตามประเภทของตัววัดแผนภาพ โดยประเภทของตัววัดแผนภาพสามารถจำแนกได้เป็น 2 ประเภทหลักๆ คือตัววัดที่ได้จากข้อมูลโดยตรง และตัววัดที่ได้จากค่าของตัววัดอื่น

การประกาศรายละเอียดการคำนวณของตัววัดที่ได้จากข้อมูลโดยตรงจะใช้แท็ก <projection> ซึ่งมีการทำงานหลักโดยการนับจำนวนขององค์ประกอบที่ต้องการ โดยสามารถกำหนดขอบเขตและเงื่อนไขของการนับได้โดยอาศัยแอททริบิวต์เพิ่มเติมของแท็กดังนี้

- relation คือการจำกัดองค์ประกอบที่ใช้ในการคำนวณเฉพาะขององค์ประกอบที่มีความสัมพันธ์ตามที่กำหนดกับองค์ประกอบที่ต้องการคำนวณตัววัด เช่น เป็นองค์ประกอบย่อย (context) เป็นต้น
- relset คือการจำกัดองค์ประกอบที่ใช้ในการคำนวณเฉพาะที่อยู่ในกลุ่มขององค์ประกอบที่ได้ระบุไว้ก่อน
- target คือการจำกัดชนิดขององค์ประกอบที่ใช้ในการคำนวณ
- element คือการจำกัดองค์ประกอบที่มีความสัมพันธ์ทางอ้อมกับองค์ประกอบที่ต้องการคำนวณตัววัด
- eltype คือการจำกัดชนิดขององค์ประกอบที่มีความสัมพันธ์ทางอ้อมกับองค์ประกอบที่ต้องการคำนวณตัววัด
- condition และ targetcondition คือการจำกัดองค์ประกอบที่มีคุณสมบัติตรงตามที่ระบุ เช่น มีขอบเขตการเข้าถึงเป็นพบบลิต เป็นต้น
- scope คือการจำกัดองค์ประกอบที่ใช้ในการคำนวณเฉพาะขององค์ประกอบที่มีขอบเขตตามที่กำหนด เช่น อยู่ในแพ็คเกจเดียวกับองค์ประกอบที่ต้องการคำนวณตัววัด เป็นต้น
- sum คือการนับผลรวมตัววัดขององค์ประกอบที่ใช้ในการคำนวณ เช่น จำนวนโอเปอเรเตอร์ (NumOps) ทั้งหมดของคลาสที่อยู่ในแพ็คเกจที่สนใจ เป็นต้น
- stat คือการนับค่าตัววัดที่มากที่สุดหรือน้อยที่สุดขององค์ประกอบที่ใช้ในการคำนวณ เช่น จำนวนโอเปอเรเตอร์ (NumOps) ที่มากที่สุดของคลาสที่อยู่ในแพ็คเกจที่สนใจ เป็นต้น
- recurse คือการกำหนดให้การคำนวณค่าตัววัดทำการทำซ้ำกับองค์ประกอบประเภทเดียวกันกับองค์ประกอบที่ต้องการคำนวณตัววัดและมีความสัมพันธ์ตามที่ระบุ เช่น นับจำนวนคลาสในแพ็คเกจที่ต้องการรวมไปถึงแพ็คเกจลูก และต่อไป เป็นต้น
- nesting คือการกำหนดให้การคำนวณค่าตัววัดทำการนับระดับความลึกขององค์ประกอบ

การคำนวณค่าตัววัดแผนภาพจากตัววัดแผนภาพอื่นต้องทำการประกาศรายละเอียดภายใต้แท็ก <compoundmetric> โดยจะทำการคำนวณตามรายละเอียดที่กำหนด แท็ก <compoundmetric> ประกอบไปด้วยแอททริบิวต์ดังนี้

- term คือแอททริบิวต์สำหรับกำหนดรายละเอียดการคำนวณ
- fallback คือแอททริบิวต์สำหรับส่งกลับค่าที่ระบุในกรณีที่การคำนวณตัววัดได้ผลลัพธ์ที่ไม่ใช่ตัวเลข
- condition คือแอททริบิวต์สำหรับกำหนดเงื่อนไขในการคำนวณ ถ้าเงื่อนไขถูกต้องจะทำการคำนวณตามที่กำหนด แต่ถ้าเงื่อนไขไม่ถูกต้องจะทำงานตามการทำงานสำรอง (alt)
- alt คือแอททริบิวต์สำหรับกำหนดการทำงานสำรองในกรณีที่เงื่อนไขการทำงานไม่ถูกต้อง

5.2.3 การแก้ไขและ ัฒนาความสามารถในการประมาณค่าความเสถียรเชิงตรรกะให้แก่โปรแกรม SDMetrics

การแก้ไขและพัฒนาความสามารถในการประมาณค่าความเสถียรเชิงตรรกะให้แก่โปรแกรม SDMetrics สามารถกระทำได้โดยทำการแก้ไขไฟล์นิยามทั้ง 3 โดยในงานวิจัยนี้ได้ทำการเพิ่มเติมตัววัดแผนภาพ ดังนี้

1. ตัววัดจำนวนความสัมพันธ์แอตทริบิวต์เมื่อคลาสที่สนใจเป็นผู้ใช้งาน (NumAss_User)
2. ตัววัดจำนวนความสัมพันธ์แอตทริบิวต์เมื่อคลาสที่สนใจเป็นผู้ถูกใช้งาน (NumAss_Provider)
3. ตัววัดจำนวนความสัมพันธ์คลาสแม่ของคลาสที่สนใจ (NOP)

นอกจากนี้ได้ทำการแก้ไขเพื่อให้โปรแกรมสามารถทำการประมาณค่าความเสถียรเชิงตรรกะของแต่ละคลาสจากตัววัดแผนภาพ และแสดงผลการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ได้

5.2.2.1 การเพิ่มตัววัดจำนวนความสัมพันธ์แอตทริบิวต์เมื่อคลาสที่สนใจเป็นผู้ใช้งาน และตัววัดจำนวนความสัมพันธ์แอตทริบิวต์เมื่อคลาสที่สนใจเป็นผู้ถูกใช้งาน

ความสัมพันธ์แอสโซซิเอชันเป็นความสัมพันธ์ระหว่างคลาส 2 คลาสซึ่งแสดงโดยเส้นเชื่อมระหว่างคลาสในแผนภาพคลาส โดยปลายทั้งสองของเส้นจะเชื่อมกับคลาสที่มีความสัมพันธ์แอสโซซิเอชันกัน ความสัมพันธ์แอสโซซิเอชันสามารถแบ่งเป็น 2 แบบ คือความสัมพันธ์แบบทางเดียวและแบบสองทาง

ความสัมพันธ์แบบทางเดียว คือกรณีที่คลาสหนึ่งเป็นผู้ใช้งานอีกคลาสหนึ่งเท่านั้น ในขณะที่ความสัมพันธ์แบบสองทางจะเป็นการใช้งานซึ่งกันและกัน โดยเมื่อทำการแปลงแผนภาพคลาสให้อยู่ในรูปแบบของเอกซ์เอ็มแอลแล้วปลายเส้นความสัมพันธ์แอสโซซิเอชันจะถูกแปลงให้เป็นองค์ประกอบของแผนภาพเช่นเดียวกับคลาส หรือแอททริบิวต์ เรียกว่าจุดสิ้นสุดแอสโซซิเอชัน (AssociationEnd) และมีคุณสมบัติต่างๆ เช่นเดียวกับองค์ประกอบอื่นๆ

คุณสมบัติประการหนึ่งของจุดสิ้นสุดแอสโซซิเอชันที่สามารถใช้ในการแบ่งแยกว่าคลาสใดเป็นผู้เรียกใช้และคลาสใดเป็นผู้ถูกเรียกใช้คือคุณสมบัติ isNavigable ซึ่งจุดสิ้นสุดแอสโซซิเอชันที่เชื่อมต่อกับคลาสที่ถูกเรียกใช้จะมีค่าคุณสมบัตินี้เป็นจริง ดังนั้นในการสร้างตัววัดแผนภาพจำนวนความสัมพันธ์แอสโซซิเอชันเมื่อคลาสที่สนใจเป็นผู้ใช้งานนั้นจะนับจำนวนแอสโซซิเอชันของคลาสที่สนใจเมื่อจุดสิ้นสุดแอสโซซิเอชันที่เชื่อมต่อกับคลาสที่สนใจมีค่าคุณสมบัติ isNavigable เป็นเท็จ



รูปที่ 5.10 ตัวอย่างคลาสที่มีความสัมพันธ์แอสโซซิเอชัน

รูปที่ 5.10 แสดงตัวอย่างคลาสที่มีความสัมพันธ์แอสโซซิเอชัน โดยคลาส C3 เป็นผู้เรียกใช้และคลาส C2 เป็นผู้ถูกเรียกใช้ซึ่งเมื่อทำการแปลงแผนภาพให้อยู่ในรูปแบบไฟล์เอกซ์เอ็มแอลแล้วจะได้ส่วนของเอกสารเอกซ์เอ็มแอลที่แสดงความสัมพันธ์แอสโซซิเอชันระหว่างทั้ง 2 คลาสดังในตารางที่ 5.6

ตารางที่ 5.6 ส่วนของเอกสารเอกซ์เอ็มแอลที่แสดงความสัมพันธ์แอสโซซิเอชัน

```
<UML:Association xmi.id='_9_5_1_24400562_1172516764739_988991_168'>

  <UML:Association.connection>

    <UML:AssociationEnd xmi.id='_9_5_1_24400562_1172516764739_786514_169' name='c'
isNavigable='true' participant='_9_5_1_24400562_1172516764729_273078_153'>

      </UML:AssociationEnd>

    <UML:AssociationEnd xmi.id='_9_5_1_24400562_1172516764739_201817_171'
participant='_9_5_1_24400562_1172516764729_480783_155'>

      </UML:AssociationEnd>
    </UML:Association.connection>
  </UML:Association>
```

จากตารางที่ 5.6 จะพบว่าภายใต้องค์ประกอบแอสโซซิเอชันจะประกอบไปด้วย 2 จุดสิ้นสุดแอสโซซิเอชัน โดยจะมีจุดสิ้นสุดแอสโซซิเอชันหนึ่งที่มีคุณสมบัติ isNavigable เป็นค่าจริง ซึ่งจุดสิ้นสุดแอสโซซิเอชันนั้นเชื่อมต่อกับคลาส C2 ซึ่งเป็นคลาสที่ถูกเรียกใช้ ทั้งนี้สามารถตรวจสอบได้ว่าจุดสิ้นสุดแอสโซซิเอชันเชื่อมต่อกับคลาสใดโดยดูจากคุณสมบัติ participant ซึ่งจะระบุหมายเลขรหัสเอกซ์เอ็มแอลขององค์ประกอบที่เชื่อมต่อ ในที่นี้เป็นหมายเลขรหัสเอกซ์เอ็มแอลของคลาส C2 ในแผนภาพคลาส

การเก็บค่าคุณสมบัติ isNavigable สามารถทำได้โดยการระบุค่าแอททริบิวต์ navigable เป็นหนึ่งในข้อมูลองค์ประกอบที่ต้องการไว้ในไฟล์ Metamodel Definition และกำหนดวิธีการเก็บค่าไว้ในไฟล์ XMI Transformation ดังตารางที่ 5.7 และ 5.8

ตารางที่ 5.7 รายละเอียดองค์ประกอบจุดสิ้นสุดแอสโซซิเอชัน

```
<modelelement name="associationend">

  <attribute name="navigable" type="data">Boolean indicating if the association end is navigable.</attribute>

  <attribute name="associationendtype" type="ref">A link to the element attached to the association end.</attribute>

</modelelement>
```

ตารางที่ 5.8 รายละเอียดการเก็บค่าองค์ประกอบจุดสิ้นสุดแอสโซซิเอชัน

```
<xmitransformation modelement="associationend" xmitpattern="UML:AssociationEnd" >

    <trigger name="navigable" type="attrval" attr="isNavigable" />

    <trigger name="associationendtype" type="attrval" attr="participant"/>

    <trigger name="context" type="attrval" attr="association"/>

</xmitransformation>
```

หลังจากเก็บค่าคุณสมบัติที่จำเป็นได้แล้วทำการสร้างตัววัดโดยอาศัยคุณสมบัติที่เก็บได้ การวัดจำนวนความสัมพันธ์แอสโซซิเอชันเมื่อคลาสที่สนใจเป็นผู้ใช้งาน สามารถทำได้โดยการนับจำนวนจุดสิ้นสุดแอสโซซิเอชันที่เชื่อมต่อกับคลาสที่สนใจและมีค่าคุณสมบัติ isNavigable เป็นเท็จ ในขณะที่การวัดจำนวนความสัมพันธ์แอสโซซิเอชันเมื่อคลาสที่สนใจเป็นผู้ถูกใช้งานสามารถทำได้โดยการนับจำนวนจุดสิ้นสุดแอสโซซิเอชันที่เชื่อมต่อกับคลาสที่สนใจ และมีค่าคุณสมบัติ isNavigable เป็นจริง รายละเอียดของตัววัดแผนภาพทั้งสองที่สร้างขึ้นแสดงดังตารางที่ 5.9

ตารางที่ 5.9 รายละเอียดตัววัดแผนภาพ

```
<metric name="NumAss_user" domain="class">

<description>Number of association relation when the class is a user.</description>

<projection relation="associationendtype" target="associationend" condition="navigable!='true' />

</metric>

<metric name="NumAss_provider" domain="class">

<description> Number of association relation when the class is a provider or being used.</description>

<projection relation="associationendtype" target="associationend" condition="navigable='true' />

</metric>
```

5.2.2.2 การเพิ่มตัววัดจำนวนความสัมพันธ์คลาสแม่ของคลาสที่สนใจ

จำนวนความสัมพันธ์คลาสแม่ หรือจำนวนความสัมพันธ์เจเนอรัลไลเซชันเมื่อคลาสที่สนใจเป็นผู้เรียกใช้ สามารถคำนวณได้โดยการนับจำนวนความสัมพันธ์แบบเจเนอรัลไลเซชันของคลาสที่สนใจเมื่อคลาสที่สนใจเป็นผู้เรียกใช้ที่ปรากฏอยู่ในแผนภาพคลาส โดยเช่นเดียวกับ

ความสัมพันธ์แอซซิซิเซชัน ความสัมพันธ์แบบเจเนอรัลไลเซชันก็เป็นองค์ประกอบหนึ่งของแผนภาพเช่นเดียวกัน และมีรูปแบบข้อมูลเมื่อแปลงเป็นเอกสารเอกซ์เอ็มแอลดังในตารางที่ 5.10

ตารางที่ 5.10 ส่วนของเอกสารเอกซ์เอ็มแอลที่แสดงความสัมพันธ์เจเนอรัลไลเซชัน

```
<UML:Generalization xmi.id='_9_5_1_24400562_1172516764739_898503_167'
child='_9_5_1_24400562_1172516764729_273078_153' parent='_9_5_1_24400562_1172516764729_645070_151' />
```

คุณสมบัติที่สำคัญขององค์ประกอบความสัมพันธ์เจเนอรัลไลเซชัน ซึ่งเป็นคุณสมบัติที่ใช้ในการระบุความสัมพันธ์ระหว่างองค์ประกอบคลาสคือ คุณสมบัติ child ซึ่งระบุรหัสขององค์ประกอบคลาสที่เป็นคลาสลูกผ่านทางความสัมพันธ์นี้ และคุณสมบัติ parent ซึ่งระบุรหัสขององค์ประกอบคลาสที่เป็นคลาสแม่ผ่านทางความสัมพันธ์นี้ ดังนั้นเพื่อนำค่าคุณสมบัติทั้งสองไปใช้ในการสร้างตัววัดแผนภาพจำนวนความสัมพันธ์คลาสแม่ จึงต้องทำการระบุค่าแอททริบิวต์ genchild และ genparent เป็นข้อมูลองค์ประกอบที่ต้องการไว้ในไฟล์ Metamodel Definition และกำหนดวิธีการเก็บค่าไว้ในไฟล์ XMI Transformation ดังตารางที่ 5.11 และ 5.12

ตารางที่ 5.11 รายละเอียดขององค์ประกอบความสัมพันธ์เจเนอรัลไลเซชัน

```
<modelelement name="generalization">
    <attribute name="genchild" type="ref">Link to the child in the generalization.</attribute>
    <attribute name="genparent" type="ref">Link to the parent in the generalization.</attribute>
</modelelement>
```

ตารางที่ 5.12 รายละเอียดการเก็บค่าองค์ประกอบความสัมพันธ์เจเนอรัลไลเซชัน

```
<xmitransformation modelelement="generalization" xmi:pattern="UML:Generalization">
    <trigger name="genchild" type="attrval" attr="child" />
    <trigger name="genparent" type="attrval" attr="parent" />
</xmitransformation>
```

การคำนวณค่าตัววัดแผนภาพจำนวนคลาสแม่ของคลาสที่สนใจ สามารถคำนวณได้โดยการนับจำนวนคลาสที่มีความสัมพันธ์แบบเจเนอรัลไลเซชันในฐานะคลาสแม่กับคลาสที่สนใจ เมื่อคลาสที่สนใจอยู่ในสถานะคลาสลูก ตารางที่ 5.13 แสดงรายละเอียดของตัววัดแผนภาพที่สร้างขึ้น

ตารางที่ 5.13 รายละเอียดตัววัดแผนภาพจำนวนคลาสแม่

```
<metric name="NOP" domain="class">
    <description> Number of parent class</description>
    <projection relation="genchild" target="generalization" element="genparent" eltype="class" />
</metric>
```

5.2.2.3 การเพิ่มความสามารถในการประมาณค่าความเสถียรเชิงตรรกะตามโมเดลที่พัฒนาขึ้น

โมเดลประมาณค่าความเสถียรเชิงตรรกะที่พัฒนาขึ้นจะอาศัยตัววัดแผนภาพต่างๆ ในการคำนวณ ดังนั้นในการเพิ่มความสามารถให้โปรแกรม SDMetrics สามารถประมาณค่าความเสถียรเชิงตรรกะของแต่ละคลาสสามารถทำได้โดย สร้างเป็นตัววัดแผนภาพของคลาสซึ่งอาศัยค่าจากตัววัดแผนภาพอื่นในการคำนวณโดยอาศัยแท็ก <compoundmetric>

เนื่องจากงานวิจัยนี้ได้ทำการสร้างโมเดลประมาณค่าความเสถียรเชิงตรรกะของคลาสจากแผนภาพคลาสและแผนภาพซีเคอร์รี่สำหรับโปรแกรม 3 ประเภท ดังนั้นจึงต้องทำการสร้างตัววัดประมาณค่าความเสถียรเชิงตรรกะ 3 ตัววัดด้วยกันตามโมเดลของโปรแกรมแต่ละประเภท โดยมีรายละเอียดดังตารางที่ 5.14 ถึง 5.16 และทำการสร้างตัววัดประมาณค่าความเสถียรเชิงตรรกะทั้ง 3 แยกจากกันอยู่ในไฟล์ข้อกำหนดตัววัดแผนภาพ 3 ไฟล์

ตารางที่ 5.14 รายละเอียดการคำนวณโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของคลาสสำหรับโปรแกรมด้านการคำนวณ

```
<metric name="EstLSta" domain="class">
    <description>Estimated Logical Stability.</description>
    <compoundmetric term="0.9805680980-0.0032740467*Classifirst +0.0048858207*NumOps-
    0.0064901626*NumAnc+0.0080432844*Attrinh -0.0010171931*MsgSent +0.0005569513*NumDesc"
    fallback="0"/>
</metric>
```

ตารางที่ 5.15 รายละเอียดการคำนวณโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของคลาส
สำหรับโปรแกรมด้านการจัดการข้อความ

```
<metric name="EstLSta" domain="class">
  <description>Estimated Logical Stability.</description>
  <compoundmetric term="1.0426495664 - 0.0239384916*NumPubOps - 0.0049619233*NumOps"
    fallback="0"/>
</metric>
```

ตารางที่ 5.16 รายละเอียดการคำนวณโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของคลาส
สำหรับโปรแกรมด้านการจัดการรูปภาพ

```
<metric name="EstLSta" domain="class">
  <description>Estimated Logical Stability.</description>
  <compoundmetric term="0.9908283241 - 0.0829356623*IfImpl - 0.0541333205*NOP
    - 0.0004169792*MsgSelf - 0.0212684802*CLD " fallback="0"/>
</metric>
```

การเลือกใช้โมเดลให้เหมาะสมกับโปรแกรมที่ต้องการประมาณค่า ผู้ใช้ต้องทำการเลือก
เองโดยทำการเลือกไฟล์ข้อกำหนดตัววัดแผนภาพที่ต้องการ

จากการเพิ่มความสามารถในการประมาณค่าความเสถียรเชิงตรรกะของคลาสตามโมเดล
ที่พัฒนาขึ้น ทำให้สามารถเพิ่มความสามารถในการประมาณค่าความเสถียรเชิงตรรกะของ
ซอฟต์แวร์ได้อีกด้วย โดยความเสถียรเชิงตรรกะของซอฟต์แวร์จะเท่ากับค่าเฉลี่ยของความเสถียร
เชิงตรรกะของทุกคลาสในซอฟต์แวร์ โดยในการเพิ่มความสามารถในการแสดงค่าประมาณของ
ความเสถียรเชิงตรรกะของซอฟต์แวร์ จะต้องทำการเพิ่มข้อกำหนดองค์ประกอบซอฟต์แวร์ลงใน
ไฟล์ MetaModel Definition และ XML Transformation เสียก่อน เพื่อให้โปรแกรม SDMetrics
รู้จักองค์ประกอบโปรแกรมและแสดงผลในระดับซอฟต์แวร์ได้ รายละเอียดของข้อกำหนด
องค์ประกอบซอฟต์แวร์แสดงในตารางที่ 5.17 และ 5.18

ตารางที่ 5.17 รายละเอียดองค์ประกอบซอฟต์แวร์

```
<modelelement name="software" />
```

ตารางที่ 5.18 รายละเอียดการเก็บค่าองค์ประกอบซอฟต์แวร์

```
<xmitransformation modelelement="software" xmitpattern="UML:Model" recurse="true" />
```

การหาค่าเฉลี่ยของค่าประมาณความเสถียรเชิงตรรกะของทุกคลาสในโปรแกรมแบ่งออกเป็น 2 กรณีด้วยกันคือ กรณีที่มีการแบ่งแพ็คเกจ และกรณีที่ไม่มีการแบ่งแพ็คเกจ

สำหรับกรณีที่มีการแบ่งแพ็คเกจต้องการหาค่าเฉลี่ยของค่าประมาณความเสถียรเชิงตรรกะของทุกคลาสในแพ็คเกจเสียก่อน โดยในการหาค่าเฉลี่ยของค่าประมาณความเสถียรเชิงตรรกะของทุกคลาสในแพ็คเกจ จะอาศัยตัววัดแผนภาพ NumCls_tc ซึ่งเป็นจำนวนคลาสทั้งหมดในแพ็คเกจ และตัววัดแผนภาพ SumStab_p ซึ่งเป็นผลรวมของค่าประมาณความเสถียรเชิงตรรกะของทุกคลาสในแพ็คเกจ และค่าประมาณความเสถียรเชิงตรรกะของแพ็คเกจจะเท่ากับ ค่าประมาณความเสถียรเชิงตรรกะของทุกคลาสในแพ็คเกจหารด้วยจำนวนคลาสทั้งหมดในแพ็คเกจ รายละเอียดของตัววัดแผนภาพ NumCls_tc SumStab_p และค่าประมาณความเสถียรเชิงตรรกะของแพ็คเกจแสดงในตารางที่ 5.19 ถึง 5.21 ตามลำดับ

ตารางที่ 5.19 รายละเอียดตัววัดแผนภาพจำนวนคลาสทั้งหมดในแพ็คเกจ

```
<metric name="NumCls_tc" domain="package" category="Size">
    <description>The number of classes in the package, its subpackages, and so on</description>
    <projection relation="context" target="class" recurse="true"/>
</metric>
```

ตารางที่ 5.20 รายละเอียดตัววัดแผนภาพผลรวมค่าประมาณความเสถียรเชิงตรรกะของทุกคลาสในแพ็คเกจ

```
<metric name="SumStab_p" domain="package" internal="true">
    <description>The Summation of class logical stability in this package, sub package and so on.
</description>
    <projection relation="context" target="class" sum="EstLSta" recurse="true"/>
</metric>
```

ตารางที่ 5.21 รายละเอียดตัววัดค่าประมาณความเสถียรเชิงตรรกะของแพ็คเกจ

```
<metric name="EstStab" domain="package">
    <description>The estimate logical stability of this package.</description>
    <compoundmetric conditions="NumCls_tc>0" term="SumStab/NumCls_tc" fallback="0">
</compoundmetric>
</metric>
```

หลังจากสามารถคำนวณค่าประมาณความเสถียรเชิงตรรกะของแพ็คเกจได้แล้ว จะสามารถคำนวณค่าประมาณความเสถียรเชิงตรรกะของซอฟต์แวร์ได้โดย หาผลรวมของค่าประมาณความเสถียรเชิงตรรกะของทุกแพ็คเกจแล้วหารด้วยจำนวนแพ็คเกจทั้งหมด ดังนั้นต้องทำการสร้างตัววัดแผนภาพเพิ่มเติมอีก 2 ตัววัด คือ ตัววัดจำนวนแพ็คเกจ(NumPck) และผลรวมของค่าประมาณความเสถียรเชิงตรรกะของทุกแพ็คเกจ(SumStab) ซึ่งมีรายละเอียดในตารางที่ 5.22 และ 5.23 ตามลำดับ

ตารางที่ 5.22 รายละเอียดตัววัดแผนภาพจำนวนแพ็คเกจทั้งหมดในซอฟต์แวร์

```
<metric name="Numpck" domain="software">
    <description>Number of All the Package in Software</description>
    <projection relation="context" target="package" condition="NumCls_tc>0" recurse="true"/>
</metric>
```

ตารางที่ 5.23 รายละเอียดตัววัดแผนภาพผลรวมค่าประมาณความเสถียรเชิงตรรกะของทุกแพ็คเกจในซอฟต์แวร์

```
<metric name="SumStab" domain="software" internal="true" >
    <description>Sum of the estimate logical stability of package</description>
    <projection relation="context" target="package" sum="EstStab" condition="NumCls_tc>0" recurse="true"/>
</metric>
```

สำหรับกรณีที่ซอฟต์แวร์ไม่มีการแบ่งแพ็คเกจ หรือไม่มีแพ็คเกจ ต้องทำการหาจำนวนคลาสทั้งหมดของซอฟต์แวร์ และผลรวมของค่าประมาณความเสถียรเชิงตรรกะของทุกคลาส โดยทำการสร้างตัววัดแผนภาพอีก 2 ตัวคือ NumCls และ SumStab_c ซึ่งมีรายละเอียดในตารางที่ 5.24 และตารางที่ 5.25 ตามลำดับ

ตารางที่ 5.24 รายละเอียดตัววัดแผนภาพจำนวนคลาสทั้งหมดในซอฟต์แวร์

```
<metric name="NumCls" domain="software">
    <description>Number of All the class in Software</description>
    <projection relation="context" target="class" recurse="true"/>
</metric>
```

ตารางที่ 5.25 รายละเอียดตัววัดแผนภาพผลรวมค่าประมาณความเสถียรเชิงตรรกะของทุกคลาส

```
<metric name="SumStab_c" domain="software" internal="true">
  <description>Sum of the estimate logical stability of package</description>
  <projection relation="context" target="class" sum="EstLSta" recurse="true"/>
</metric>
```

ความเสถียรเชิงตรรกะของซอฟต์แวร์จะมีค่าเท่ากับผลรวมค่าประมาณความเสถียรเชิงตรรกะของทุกแพ็คเกจในซอฟต์แวร์หารด้วยจำนวนแพ็คเกจ หรือเท่ากับผลรวมค่าประมาณความเสถียรเชิงตรรกะของทุกคลาสในซอฟต์แวร์หารด้วยจำนวนคลาสทั้งหมด โดยรายละเอียดของตัววัดค่าประมาณความเสถียรเชิงตรรกะของซอฟต์แวร์อยู่ในตารางที่ 5.26

ตารางที่ 5.26 รายละเอียดตัววัดค่าประมาณความเสถียรเชิงตรรกะของซอฟต์แวร์

```
<metric name="EstLSta" domain="software">
  <description>The Estimate Logical Stability of program calculated by sum of all class estimate logical
  stability divide by number of classes</description>
  <compoundmetric condition="Numpck>0" term="SumStab_p/Numpck" alt="SumStab_c/NumCls"
  fallback="0"/>
</metric>
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

สรุปผลการวิจัย

6.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะจากแผนภาพคลาสและแผนภาพซีเควนซ์ เพื่อให้สามารถทำการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ได้ตั้งแต่ช่วงของการออกแบบ ซึ่งหากต้องการแก้ไขจะสามารถแก้ไขได้ง่ายและประหยัดกว่าการแก้ไขที่ซอร์สโคด การสร้างโมเดลในงานวิจัยนี้ใช้การวิเคราะห์ความถดถอยเชิงซ้อน เพื่อหาความสัมพันธ์ระหว่างความเสถียรเชิงตรรกะที่วัดจากซอร์สโคด และค่าตัววัดแผนภาพที่วัดจากแผนภาพคลาสและแผนภาพซีเควนซ์ โดยตัววัดแผนภาพที่นำมาใช้ในงานวิจัยนี้มีจำนวนทั้งสิ้น 21 ตัววัด และมีตัววัดจำนวน 11 ตัววัดที่ส่งผลต่อความเสถียรเชิงตรรกะของโปรแกรมทั้ง 3 กลุ่ม

การวัดค่าความเสถียรเชิงตรรกะของคลาสจากซอร์สโคดในงานวิจัยนี้ อาศัยเครื่องมือคำนวณความเสถียรเชิงตรรกะที่พัฒนาจากเครื่องมือคำนวณเสถียรภาพ โดยได้ทำการแก้ไขขั้นตอนการคำนวณ และผลกระทบที่เกิดจากการเปลี่ยนแปลงตามรูปแบบการเปลี่ยนแปลงที่ได้กำหนดขึ้นและครอบคลุมรูปแบบการเปลี่ยนแปลงที่ใช้ในวิทยานิพนธ์นี้ทั้งหมด รูปแบบการเปลี่ยนแปลงที่กำหนดขึ้นเพื่อใช้ในงานวิจัยนี้แบ่งออกเป็น 2 ระดับด้วยกัน คือการเปลี่ยนแปลงระดับของระบบ ซึ่งเป็นการเปลี่ยนแปลงที่เกี่ยวข้องกับความสัมพันธ์ระหว่างคลาส รวมไปถึงการเพิ่ม หรือลบคลาส และการเปลี่ยนแปลงระดับของคลาสซึ่งเป็นการเปลี่ยนแปลงแก่องค์ประกอบของคลาส คือแอตทริบิวต์ และเมธอด

การวัดค่าตัววัดแผนภาพในงานวิจัยนี้จะทำการวัดค่าจากแผนภาพคลาสและแผนภาพซีเควนซ์ที่อยู่ในรูปแบบของเอกสารเอกซ์เอ็มแอล โดยอาศัยโปรแกรม SDMetrics ซึ่งสามารถวัดค่าตัววัดแผนภาพต่างๆ ได้ และได้ทำการเพิ่มเติมตัววัดแผนภาพที่ไม่มีอยู่ในโปรแกรมเพื่อให้โปรแกรมสามารถวัดค่าตัววัดได้อย่างครบถ้วน

งานวิจัยนี้ได้แบ่งกลุ่มของซอฟต์แวร์ออกเป็น 3 กลุ่มได้แก่ซอฟต์แวร์ด้านการคำนวณ ซอฟต์แวร์ด้านข้อความ และซอฟต์แวร์ด้านรูปภาพ เนื่องจากผู้เขียนวิทยานิพนธ์เห็นว่าโปรแกรมแต่ละประเภทจะมีพฤติกรรมการทำงานที่ต่างกันออกไป และคล้ายคลึงกันเองในกลุ่มโปรแกรมประเภทเดียวกัน และจะมีพฤติกรรมบางประการที่ขัดแย้งกัน ดังนั้นจึงควรแยกพิจารณาเพื่อให้โมเดลที่ได้สามารถอธิบายความเสถียรเชิงตรรกะได้จริง และจากผลของการวิจัยที่ได้พบว่า

โปรแกรมแต่ละกลุ่มมีพฤติกรรมบางประการที่ขัดแย้งกันจริง เช่นในกลุ่มโปรแกรมด้านการคำนวณ คลาสที่มีจำนวนโอเปอเรเตอร์สูง จะมีความเสถียรเชิงตรรกะสูงในขณะที่โปรแกรมด้านการจัดการข้อความ คลาสที่มีจำนวนโอเปอเรเตอร์สูง จะมีความเสถียรเชิงตรรกะต่ำ

ผลของการวิจัยพบว่าโมเดลการประมาณค่าความเสถียรเชิงตรรกะของกลุ่มซอฟต์แวร์ด้านข้อความมีสัมประสิทธิ์การตัดสินใจเชิงซ้อนสูงที่สุด โดยมีค่าเท่ากับ 84.6% และประกอบไปด้วยตัววัดแผนภาพ 2 ตัวคือ NumOps และ NumPubOps หรืออาจสรุปได้ว่า 84.6% ของค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ด้านข้อความสามารถอธิบายได้ด้วยโมเดลการประมาณซึ่งประกอบไปด้วยตัววัดแผนภาพทั้งสอง และเมื่อทำการประเมินโมเดลด้วยชุดโปรแกรมสำหรับทดสอบพบว่า ผลการประมาณค่าจากโมเดลการประมาณค่าความเสถียรเชิงตรรกะสำหรับกลุ่มซอฟต์แวร์ด้านข้อความจะมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์เสมอ

โมเดลการประมาณค่าความเสถียรเชิงตรรกะของกลุ่มซอฟต์แวร์ที่มีสัมประสิทธิ์การตัดสินใจเชิงซ้อนสูงรองลงมา คือโมเดลการประมาณค่าความเสถียรเชิงตรรกะของกลุ่มซอฟต์แวร์ด้านการคำนวณ โดยมีค่าเท่ากับ 73.9% และประกอบไปด้วยตัววัดแผนภาพ 6 ตัว ได้แก่ ClassifInst NumOps NumAnc AttrInh MsgSent และ NumDesc หรืออาจสรุปได้ว่า 73.9% ของค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ด้านการคำนวณสามารถอธิบายได้ด้วยโมเดลการประมาณค่าซึ่งประกอบไปด้วยตัววัดแผนภาพทั้งหมด และจากผลการประเมินโมเดลการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ด้านการคำนวณพบว่า 91.7 เปอร์เซ็นต์ของค่าที่ประมาณได้จะคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์

โมเดลการประมาณค่าความเสถียรเชิงตรรกะของกลุ่มซอฟต์แวร์ด้านรูปภาพมีสัมประสิทธิ์การตัดสินใจเชิงซ้อนเท่ากับ 73.8% และประกอบไปด้วยตัววัดแผนภาพ 6 ตัว คือ NumOps OpsInh NumAnc IfImpl NumAtt และ CLD หรืออาจสรุปได้ว่า 73.8% ของค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ด้านการจัดการรูปภาพสามารถอธิบายได้ด้วยโมเดลการประมาณซึ่งประกอบไปด้วยตัววัดแผนภาพทั้งหมด และจากผลการประเมินโมเดลการประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ด้านการจัดการรูปภาพพบว่า 93.3 เปอร์เซ็นต์ของค่าที่ประมาณได้จะคลาดเคลื่อนจากค่าจริงไม่เกิน 25 เปอร์เซ็นต์

อย่างไรก็ตามเมื่อทำการทดลองเพื่อหาโมเดลประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ทั้ง 3 กลุ่มรวมกัน ผลที่ได้ไม่น่าพึงพอใจเนื่องจากได้โมเดลที่สัมประสิทธิ์การตัดสินใจเชิงซ้อนต่ำ หรือได้โมเดลที่ไม่สามารถอธิบายความเสถียรเชิงตรรกะของซอฟต์แวร์ได้อย่างแท้จริง ทั้งนี้เนื่องจากซอฟต์แวร์แต่ละกลุ่มต่างมีลักษณะเฉพาะของตน ซึ่งเมื่อนำมารวมกันแล้วก่อให้เกิดความขัดแย้งขึ้น ส่งผลให้โมเดลที่ได้ไม่เพียงพอ

หลังจากทำการสร้างโมเดลสำหรับประมาณค่าความเสถียรเชิงตรรกะของซอฟต์แวร์ทั้ง 3 กลุ่ม รวมทั้งทำการประเมินผลซอฟต์แวร์เสร็จสิ้นแล้ว จึงทำการพัฒนาโปรแกรม SDMetrics ให้สามารถทำการคำนวณตามโมเดลการประมาณค่าทั้ง 3 เพื่อประมาณค่าความเสถียรเชิงตรรกะของ คลาส แพคเกจ และซอฟต์แวร์ได้โดยอัตโนมัติ

6.2 ข้อสังเกต

1. ขนาดของซอฟต์แวร์ เนื่องจากข้อจำกัดเรื่องขนาดของซอฟต์แวร์ของโปรแกรมที่ใช้ในการทดลองทำให้ไม่สามารถทำการทดลองกับซอฟต์แวร์ที่มีขนาดใหญ่ได้ ทำให้หากนำโมเดลที่ได้ไปใช้กับซอฟต์แวร์ที่มีขนาดใหญ่ อาจเกิดการคลาดเคลื่อนได้
2. คลาสที่มีรายละเอียดเหมือนกันเมื่อนำประมาณค่าด้วยโมเดลประมาณค่าความเสถียรเชิงตรรกะของโปรแกรมทั้ง 3 กลุ่ม จะได้ค่าความเสถียรเชิงตรรกะที่ไม่เท่ากัน ทั้งนี้เนื่องจากงานวิจัยศึกษาโปรแกรมแต่ละกลุ่มแยกจากกัน ทำให้โมเดลที่ได้จากวิทยานิพนธ์นี้สามารถใช้ประมาณค่าความเสถียรเชิงตรรกะได้เฉพาะโปรแกรมที่อยู่ในกลุ่มเดียวกันเท่านั้น

6.3 แนวทางการวิจัยในอนาคต

1. เพิ่มตัววัดแผนภาพของแผนภาพอื่นๆ เช่น แผนภาพกิจกรรม เป็นต้น เพื่อให้โมเดลการประมาณค่ามีความถูกต้องยิ่งขึ้น
2. ทำการศึกษาซอฟต์แวร์ทั้ง 3 กลุ่ม เพื่อศึกษาความสัมพันธ์ระหว่างความเสถียรเชิงตรรกะกับพฤติกรรมของซอฟต์แวร์ในแต่ละประเภท เพื่อสร้างโมเดลที่สามารถประมาณค่าความเสถียรเชิงตรรกะของโปรแกรมประเภทใดๆ ก็ได้
3. เพิ่มจำนวนของโปรแกรมที่ใช้ในการทดลองเพื่อให้โมเดลมีความถูกต้องมากขึ้น
4. นำโปรแกรมที่มีขนาดใหญ่มาทำการทดลองเพิ่มเพื่อสร้างโมเดลที่สามารถประมาณค่าความเสถียรเชิงตรรกะของโปรแกรมที่มีขนาดใหญ่ได้
5. ใช้ผลของการวิจัย เพื่อสร้างวิธีการวัดความเสถียรของแบบจำลองการออกแบบซอฟต์แวร์โดยตรง

6.4 ผลงานตีพิมพ์

วิทยานิพนธ์ฉบับนี้มีผลงานตีพิมพ์ หรือเข้าร่วมการประชุมวิชาการดังนี้

1. งานสัมมนาวิชาการนานาชาติ "4" Joint Conference on Computer Science and Software Engineer (JCSSE 2007)" ระหว่างวันที่ 2 -4 พฤษภาคม พ.ศ.2550
บทความชื่อ "Estimating Software Logical Stability from Class Diagram and Sequence Diagram"



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

- [1] Shari Lawrence Pfleeger and Shawn A. Bohner. A Framework for Software Maintenance Metrics. IEEE Transactions on Software Engineering, 1990.
- [2] R. S. Arnold and S. A. Bohner. Impact Analysis – Towards A Framework for Comparison. Proceedings of the Conference on Software Maintenance, 1993.
- [3] S. Yau and J. Collofello. Some stability measures for software maintenance. IEEE Transactions on Software Engineering, 1980.
- [4] Boehm, Barry W. Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall, 1981.
- [5] S. Yau, J. Collofello and T. MacGregor. Ripple effect analysis of software maintenance. Computer Software and Applications Conference, 1978.
- [6] Michelle L. Lee. Change Impact Analysis of Object-Oriented Software. Technical Report, George Mason University, 1998.
- [7] Mahmoud O. Elish and David Rine. Investigation of Metrics for Object-Oriented Design Logical Stability. Proceedings of the Seventh European Conference On Software Maintenance And Reengineering, 2003.
- [8] James Rumbaugh, Ivar Jacobson and Grady Booch. The Unified Modeling Language Reference Manual. Massachusetts: Addison-Wesley, 1999.
- [9] กัลยา วาณิชยบัญชา. การวิเคราะห์สถิติ : สถิติสำหรับการบริหารและวิจัย. พิมพ์ครั้งที่ ๑ : กรุงเทพมหานคร : ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย, 2539.
- [10] Conte, S.D., Dunsmore, H.E., and Shen, V.Y. Software engineering metrics and models. Benjamin/Cummings Publishing Company, 1986.
- [11] L. Li and J. Offutt. Algorithmic Analysis of the Impact of Changes to Object-Oriented Software. Proceedings of the International Conference on Software Maintenance, 1996.
- [12] D. Kung, J. Gao, P. Hsia, F. Wen, Y. Toyoshima, and C. Chen. Change Impact Identification in Object Oriented Software Maintenance. Proceedings of the International Conference on Software Maintenance, 1994.
- [13] รายการรีแฟคเตอริง แหล่งที่มา: <http://www.refactoring.com> [2004, January 6]

- [14] SDMetrics Home Page. <http://www.sdmetrics.com>
- [15] MagicDraw UML Home Page. <http://www.magicdraw.com>
- [16] Java source code, Available from : <http://sourceforge.net> [2007 ,January 16]
- [17] Java source code, Available from : <http://www.planet-source-code.com>
[2007, January 16]
- [18] ธีรเดช แซ่ตัน. การใช้มาตรฐานวัดเชิงวัดคุณภาพของเมทอด. วิทยานิพนธ์ปริญญา
มหาบัณฑิต, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์
มหาวิทยาลัย, 2547.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

อัลกอริทึมหาค่าความเสถียรเชิงตรรกะของคลาส

ในส่วนนี้จะแสดงถึงอัลกอริทึมที่ใช้ในการหาค่าความเสถียรเชิงตรรกะของคลาสจากซอร์สโคด โดยมีรายละเอียดดังแสดงในตารางที่ ก-1

ตารางที่ ก-1 อัลกอริทึมหาค่าความเสถียรเชิงตรรกะของคลาส

```
BEGIN  
  
    TNC = 0  
  
    FOR each class C DO  
  
        FOR each type of System level change I DO  
  
            IF I can be applied to C THEN  
  
                IC = {set of classes impacted by applying I to C}  
  
                FOR each class I in IC DO  
  
                     $NTE_i = NTE_i + 1$   
  
                ENDIF  
  
                TNC = TNC + 1  
  
            ENDIF  
  
        ENDFOR  
  
    ENDFOR  
  
    FOR each attribute A in C DO  
  
        FOR each type of attribute change J DO  
  
            IF J can be applied to A THEN  
  
                IC = {set of classes impacted by applying J to A}  
  
                FOR each class I in IC DO  
  
                     $NTE_i = NTE_i + 1$   
  
                ENDIF  
  
                TNC = TNC + 1  
  
            ENDIF  
  
        ENDFOR  
  
    ENDFOR
```

ตารางที่ ก-1 อัลกอริทึมหาค่าความเสถียรเชิงตรรกะของคลาส (ต่อ)

```

        ENDFOR
    ENDFOR
    FOR each method M in C DO
        FOR each type of method change K DO
            IF K can be applied to M THEN
                IC = {set of classes impacted by applying K to M}
                FOR each class I in IC DO
                     $NTE_i = NTE_i + 1$ 
                ENDIF
                 $TNC = TNC + 1$ 
            ENDIF
        ENDFOR
    ENDFOR
    ENDFOR
    FOR each class I DO
         $CLS_i = 1 - (NTE_i / TNC)$ 
    ENDFOR
END

```

สำหรับความหมายของตัวแปรในอัลกอริทึมมีดังนี้

- C คือ คลาสที่สนใจ
- A คือ แอททริบิวต์ของคลาสที่สนใจ
- M คือ เมธอดของคลาสที่สนใจ
- I คือ รูปแบบการเปลี่ยนแปลงระดับระบบ
- J คือ รูปแบบการเปลี่ยนแปลงระดับคลาสในส่วนของแอททริบิวต์
- K คือ รูปแบบการเปลี่ยนแปลงระดับในส่วนของเมธอด

- TNC คือ จำนวนครั้งของการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมด (Total Number of Changes)
- NTE, คือ จำนวนครั้งที่แต่ละคลาสได้รับผลกระทบ (Number of Time class recieved Effects)



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง

ในส่วนนี้จะแสดงความสัมพันธ์ระหว่างรูปแบบการเปลี่ยนแปลงในตารางที่ 3.1 กับรูปแบบการทำรีแฟคเตอริงต่างๆ

การทำรีแฟคเตอริง คือการจัดรูปแบบใหม่เพื่อให้รูปแบบโครงสร้างมีความถูกต้องเหมาะสม การจัดรูปแบบใหม่ทำได้โดยการเปลี่ยนแปลงแก้ไขส่วนหนึ่งของโปรแกรม ดังนั้นกระบวนการในการทำรีแฟคเตอริงจึงเกิดจากการประกอบกันของรูปแบบการเปลี่ยนแปลงในตารางที่ 3.1 เช่น การย้ายเมทอด (Move Method) คือการลบเมทอดจากคลาสหนึ่ง และเพิ่มเมทอดในอีกคลาสหนึ่ง หรือการย้ายเมทอดประกอบไปด้วยรูปแบบการเปลี่ยนแปลง 2 รูปแบบ คือ รูปแบบที่ 22 ลบเมทอด และรูปแบบที่ 21 เพิ่มเมทอด รายละเอียดแสดงความสัมพันธ์ทั้งหมดระหว่างรูปแบบการทำรีแฟคเตอริงกับรูปแบบการเปลี่ยนแปลงแสดงดังตารางที่ ข-1 โดยด้านซ้ายคือรูปแบบการทำรีแฟคเตอริง และด้านขวาคือรูปแบบการเปลี่ยนแปลงที่ใช้ในการทำรีแฟคเตอริงนั้นๆ

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง

รีแฟคเตอริง	การเปลี่ยนแปลง
Add Parameter	24
Change Bidirectional Association to Unidirectional	10
Change Reference to Value	10,9
Change Unidirectional Association to Bidirectional	9
Change Value to Reference	9,10
Collapse Hierarchy	4,14
Consolidate Conditional Expression	26
Consolidate Duplicate Conditional Fragments	26
Convert Dynamic to Static Construction by Gerard M. Davison	26
Convert Static to Dynamic Construction by Gerard M. Davison	26

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง (ต่อ)

รีแฟคเตอริง	การเปลี่ยนแปลง
Decompose Conditional	26,21
Duplicate Observed Data	22,13,9
Eliminate Inter-Entity Bean Communication	1
Encapsulate Collection	26,21
Encapsulate Downcast	26
Encapsulate Field	18,21
Extract Class	16,22,13,9
Extract Interface	1
Extract Method	26,21
Extract Package by Gerard M. Davison	*
Extract Subclass	16,22,3
Extract Superclass	16,22,1
Form Template Method	26,21,22
Hide Delegate	22,21,10,9
Hide Method	27
Hide presentation tier-specific details from the business tier	*
Inline Class	16,22,14,15,21
Inline Method	26,22
Inline Temp	16
Introduce A Controller (Link only)	26,21,13
Introduce Assertion	26
Introduce Business Delegate (Link only)	3

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง (ต่อ)

รีแฟคเตอริง	การเปลี่ยนแปลง
Introduce Explaining Variable	26,15
Introduce Foreign Method	21
Introduce Local Extension	13 หรือ 3
Introduce Null Object	3
Introduce Parameter Object	13,24,26
Introduce Synchronizer Token (Link only)	11
Localize Disparate Logic (Link only)	13
Merge Session Beans (Link only)	14,15,21
Move Business Logic to Session (Link only)	
Move Class by Gerar M. Davison	*
Move Field	16,15
Move Method	22,21
Parameterize Method	22,21
Preserve Whole Object	26
Pull Up Constructor Body	21,26
Pull Up Field	16,15
Pull Up Method	22,21
Push Down Field	16,15
Push Down Method	22,21
Reduce Scope of Variable by Mats Henricson	26
Refactor Architecture by Tiers	*
Remove Assignments to Parameters	26

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง (ต่อ)

รีแฟคเตอริง	การเปลี่ยนแปลง
Remove Control Flag	26
Remove Double Negative by Ashley Frieze and Martin Fowler	26,21
Remove Middle Man	22,21,10,9
Remove Parameter	24
Remove Setting Method	22
Rename Method	23
Replace Array with Object	26 หรือ 13
Replace Assignment with Initialization by Mats Henricson	26
Replace Conditional with Polymorphism	26
Replace Conditional with Visitor by Ivan Mitrovic	26,13
Replace Constructor with Factory Method	26,22,21
Replace Data Value with Object	16,5
Replace Delegation with Inheritance	10,3
Replace Error Code with Exception	26
Replace Exception with Test	26
Replace Inheritance with Delegation	4,9
Replace Iteration with Recursion by Dave Whipp	26
Replace Magic Number with Symbolic Constant	26,15
Replace Nested Conditional with Guard Clauses	26
Replace Parameter with Explicit Methods	26,21
Replace Parameter with Method	26
Replace Record with Data Class	13,26

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอร์กับการเปลี่ยนแปลง (ต่อ)

รีแฟคเตอร์	การเปลี่ยนแปลง
Replace Recursion with Iteration by Ivan Mitrovic	26
Replace Static Variable with Parameter by Marian Vittek	24,26
Replace Subclass with Fields	22,15,26
Replace Temp with Query	26,21
Replace Type Code with State/Strategy	16,9
Replace Type Code with Subclasses	16,3
Reverse Conditional by Bill Murphy and Martin Fowler	26
Self Encapsulate Field	26,21
Separate Data Access Code (Link only)	26,9
Separate Query from Modifier	22,21
Split Loop by Martin Fowler	26
Split Temporary Variable	26
Substitute Algorithm	26
Use a Connection Pool (Link only)	•
Wrap entities with session (Link only)	•

• ไม่สามารถระบุได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

ข้อมูลที่ใช้ในการประเมินผลโมเดล

ในส่วนนี้จะแสดงข้อมูลที่ใช้ในการประเมินผลโมเดลการประมาณค่าความเสถียรเชิงตรรกะของทั้ง 3 กลุ่มซอฟต์แวร์

ตารางที่ ค-1 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการคำนวณ

Class Name	NumOps	NumDesc	NumAnc	AttrInh	ClassInst	MsgSent	Class Stability	Estimated Class Stability	MRE
JCalc	1	0	0	0	1	3	0.954386	0.979128	0.024742
JCalcBuffer	20	1	0	0	47	8	0.8	0.916824	0.116824
JCalcList	12	0	1	1	21	16	0.629825	0.955721	0.325896
JCalcMenuBar	6	0	0	0	8	1	0.738597	0.982673	0.244077
JCalcStandardFrame	23	0	0	0	26	34	0.498246	0.973232	0.474987
CalcMachineNumber	0	0	0	0	20	4	0.919964	0.911018	0.008946
func	3	0	0	0	18	56	0.967626	0.87933	0.088296
CalculatorException	0	0	0	0	34	29	0.968525	0.839752	0.128773
CalculatorTester	0	0	0	0	12	14	0.938849	0.927039	0.01181

ตารางที่ ค-1 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการคำนวณ (ต่อ)

Class Name	NumOps	NumDesc	NumAnc	AttrInh	ClassInst	MsgSent	Class Stability	Estimated Class Stability	MRE
CalculatorTest	0	0	0	0	6	4	1	0.956855	0.043145
E	0	0	0	0	5	1	0.952338	0.963181	0.010843
Entries	0	0	0	0	16	9	0.948741	0.919029	0.029712
Entry	0	0	0	0	2	1	1	0.973003	0.026997
GuiCommandLine	1	0	0	0	9	6	0.940648	0.949884	0.009237
jcalc	1	0	0	0	7	7	0.958633	0.955415	0.003218
jcalc_applet	0	0	0	0	1	2	0.980216	0.97526	0.004956
jcalc_math	0	0	0	0	13	4	0.936151	0.933937	0.002214
jcalc_trig	2	0	0	0	17	5	0.869604	0.929595	0.059991
operatorChecker	0	0	0	0	1	0	0.94964	0.977294	0.027654
OperatorControlCenter	3	0	0	0	19	17	0.94964	0.915726	0.033914
PI	0	0	0	0	13	2	0.928957	0.935971	0.007014
ResultsList	1	0	0	0	13	9	0.923561	0.933737	0.010175

ตารางที่ ค-1 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการคำนวณ (ต่อ)

ResultList_MouseListener	1	0	0	0	6	0	0.958633	0.96581	0.007177
VariableTable	0	0	0	0	6	6	0.954137	0.954821	0.000684

ตารางที่ ค-2 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการจัดการข้อความ

Class Name	NumOps	NumPubOps	Class Stability	Estimated Class Stability	MRE
TWedit	2	2	0.9502165	0.984849	0.034632
TWedit_Frame1	25	15	0.32467532	0.559524	0.234849
TWedit_Frame1_AboutBox	5	2	0.8095238	0.969963	0.160439
MyTextArea	9	9	0.8773424	0.782546	0.094797
AboutDialog	5	1	0.9199319	0.993901	0.07397
FindReplaceDialog	5	1	0.8994889	0.993901	0.094413
MainWindow	28	9	0.49403745	0.688269	0.194232
TextDocumentListener	6	5	0.9131175	0.893186	0.019932
WindowSizer	6	5	0.91141397	0.893186	0.018228

ตารางที่ ค-3 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการจัดการรูปภาพ

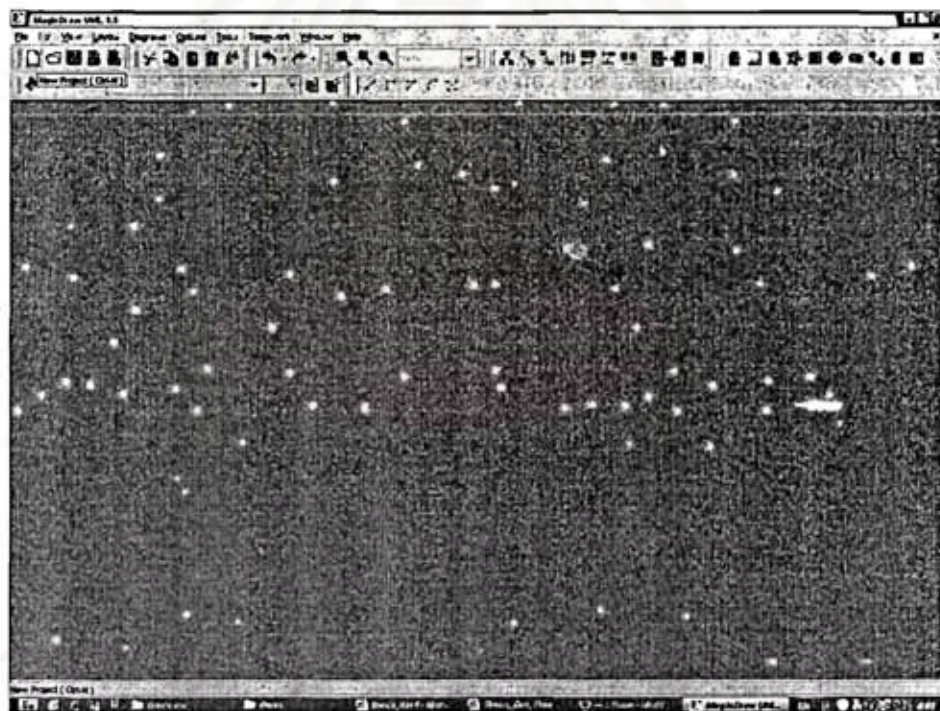
Class Name	NumAttr	NumOps	Impl	NumAnc	CLD	Opsinh	Class Stability	Estimated Class Stability	MRE
filterObject	0	8	0	0	0	0	0.901709	0.974056	0.072347
graphicsfile	0	2	0	0	0	0	0.965812	0.99035	0.024538
graphicswindow	9	5	0	0	0	0	0.489316	0.922232	0.432916
JDrawerPane	0	2	0	0	0	0	0.963675	0.99035	0.026674
jgraphicsframe	0	16	0	0	0	0	0.75	0.952332	0.202332
IMGCanvas	0	0	0	1	0	13	0.791339	0.919826	0.128487
IMGCircle	0	0	0	2	0	14	0.937008	0.977229	0.040221
IMGEllipse	0	0	0	1	1	13	0.811024	0.891765	0.080741
IMGen	0	0	0	0	0	0	0.852362	0.995781	0.143419
IMGImage	0	0	1	0	0	0	0.830709	0.92061	0.089901
IMGLine	0	0	0	1	0	13	0.78937	0.919826	0.130456
IMGRectangle	0	0	0	1	1	13	0.78937	0.891765	0.102395
IMGShape	0	0	1	0	2	0	0.751969	0.864488	0.11252
IMGSquare	0	0	0	2	0	16	0.913386	0.955003	0.041617
IMGText	0	0	1	0	0	0	0.814961	0.92061	0.105649

ภาคผนวก ง

ตัวอย่างวิธีการแปลงกลับ

การแปลงกลับแผนภาพจากซอร์สโคด เพื่อสร้างแผนภาพคลาสและแผนภาพซีควเन्ซ์เพื่อใช้ในการวิจัยนี้ ได้อาศัยโปรแกรม MagicDraw UML เวอร์ชัน 9.5 ซึ่งเป็นโปรแกรมสำหรับการวาดแผนภาพยูเอ็มแอล และมีระบบสนับสนุนการทำงานเพื่อช่วยในการออกแบบซอฟต์แวร์ และมีคุณสมบัติในการแปลงกลับแผนภาพจากซอร์สโคดอีกด้วย สำหรับรายละเอียดทั้งหมดของโปรแกรม MagicDraw UML สามารถหาได้ที่ www.magicdraw.com โดยในที่นี้จะแสดงการทำงานในส่วนของการแปลงกลับแผนภาพคลาสและแผนภาพซีควเन्ซ์จากซอร์สโคดเท่านั้น ซึ่งมีการทำงานดังนี้

เมื่อเปิดโปรแกรม MagicDraw UML ขึ้นจะปรากฏหน้าต่างการทำงานหลักมีลักษณะดังรูปที่ ง-1 โดยจะแบ่งออกเป็น 2 ส่วนคือ ส่วนของเมนูการทำงานด้านบน และส่วนของพื้นที่การทำงาน



รูปที่ ง-1 หน้าต่างการทำงานหลักของโปรแกรม MagicDraw UML

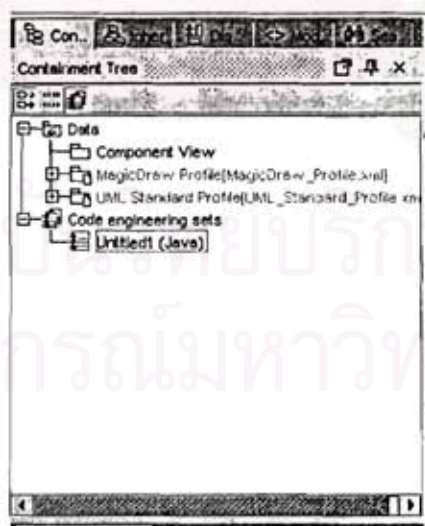
เริ่มการทำงานโดยเลือกคำสั่ง File->New Project เพื่อสร้างโปรเจกต์ใหม่ โดยเมื่อทำการสร้างโปรเจกต์ใหม่จะปรากฏหน้าต่างย่อยทางด้านซ้ายมือของพื้นที่การทำงานหลัก ซึ่งแสดงรายละเอียดโครงสร้างและองค์ประกอบของโปรเจกต์ที่สร้างขึ้น องค์ประกอบที่สำคัญหนึ่งของทุก

โปรเจกต์คือ Code Engineering Set ซึ่งเป็นองค์ประกอบสำหรับสร้าง ระบุและทำงานต่างๆ ที่เกี่ยวข้องกับซอร์สโค้ด รวมไปถึงการแปลงกลับแผนภาพด้วย ในการแปลงกลับแผนภาพต้องทำการสร้าง Code Engineering Set เสียก่อน โดยคลิกขวาที่ Code Engineering Set แล้วเลือกคำสั่ง new ซึ่งจะปรากฏรายการของประเภทไฟล์ซอร์สโค้ดของ code engineering set ในที่นี้เลือกรายการเป็นจาวา แสดงดังรูปที่ ง-2



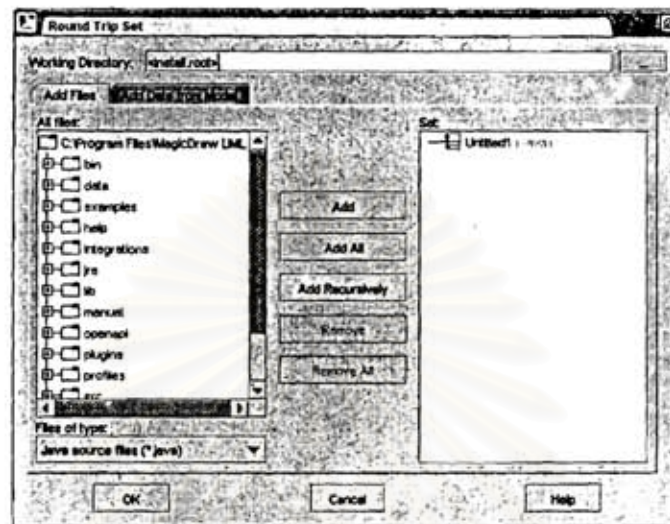
รูปที่ ง-2 การสร้าง code engineering set

เมื่อทำการเลือกประเภทของไฟล์ซอร์สโค้ดที่ต้องการนำเข้าแล้ว จะปรากฏ code engineering set ที่สร้างขึ้นในหน้าต่างแสดงองค์ประกอบดังรูปที่ ง-3



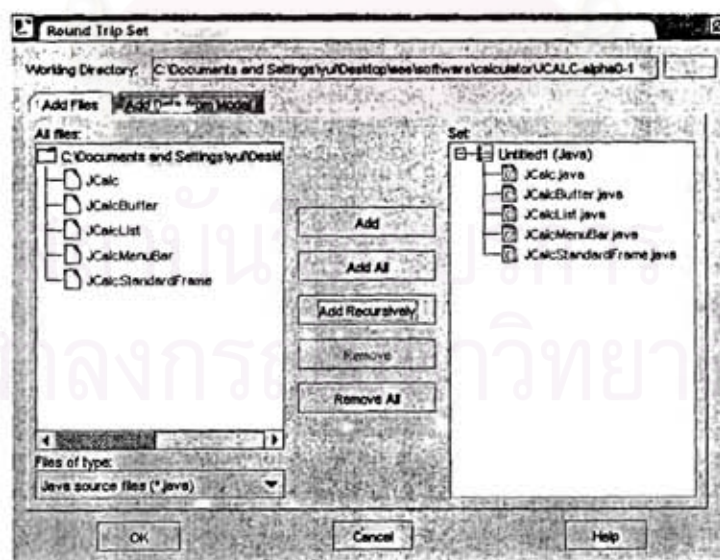
รูปที่ ง-3 code engineering set ที่สร้างขึ้น

เมื่อสร้าง code engineering set ที่ต้องการแล้วต้องทำการนำเข้าไฟล์ซอร์สโค้ดของซอฟต์แวร์ที่ต้องการแปลงกลับ โดยคลิกขวาที่ code engineering set ที่สร้างขึ้นแล้วเลือกคำสั่ง edit จากนั้นโปรแกรมจะแสดงหน้าต่างสำหรับเลือกรายการของไฟล์ซอร์สโค้ดที่ต้องการ ดังรูปที่ ๓-4



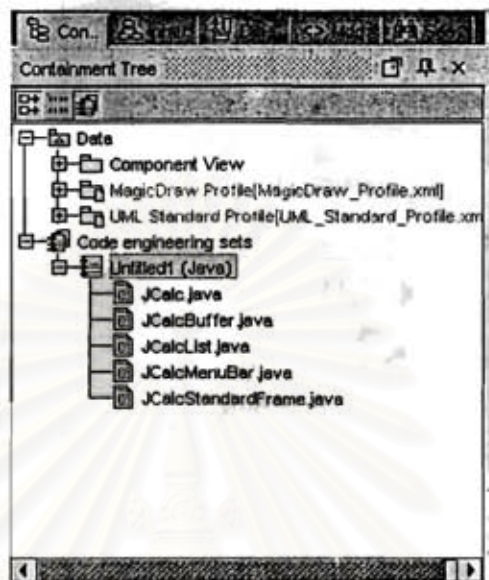
รูปที่ ๓-4 หน้าจอสำหรับเลือกไฟล์ซอร์สโค้ด

ในการเลือกไฟล์สามารถเลือกได้ 3 แบบด้วยกันคือ เลือกไฟล์เดียว (Add) เลือกทั้งหมด (Add All) หรือเลือกแบบวนกลับ (Add Recursively) โดยการเลือกแบบวนกลับ คือการเลือกไฟล์ซอร์สโค้ดตามประเภทที่กำหนดข้างต้นจากไดเรกทอรีปัจจุบัน และไดเรกทอรีย่อยทั้งหมดโดยอัตโนมัติ เมื่อทำการเลือกไฟล์ที่ต้องการนำเข้าเสร็จสิ้นแล้วจะปรากฏหน้าต่างดังรูปที่ ๓-5



รูปที่ ๓-5 หน้าจอสำหรับเลือกไฟล์ซอร์สโค้ดเมื่อได้ทำการเลือกไฟล์ที่ต้องการแล้ว

หลังจากทำการเลือกไฟล์ที่ต้องการเข้าสู่โปรเจกต์แล้ว องค์ประกอบของ code engineering set จะเปลี่ยนแปลงไปโดยจะปรากฏรายการของไฟล์ซอร์สโค้ดที่ได้ทำการนำเข้าอยู่ภายใน code engineering set ดังรูปที่ ง-6

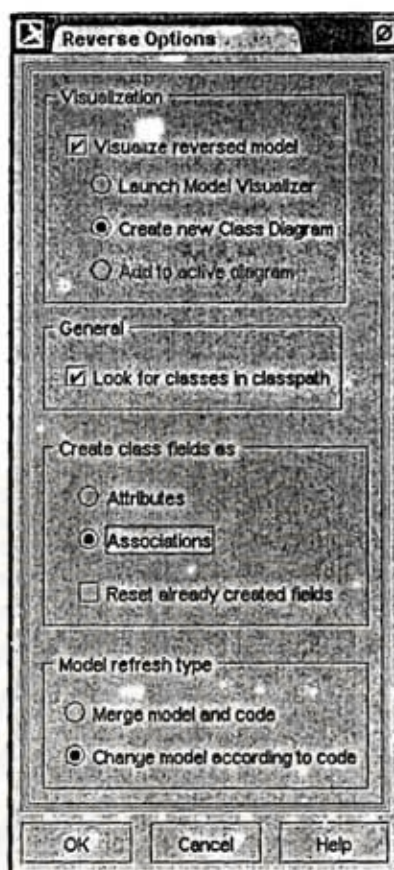


รูปที่ ง-6 รายการของไฟล์ซอร์สโค้ด

หลังจากทำการนำเข้าไฟล์ซอร์สโค้ดแล้วต้องทำการแปลงกลับซอร์สโค้ดที่เลือกเข้ามาให้อยู่ในรูปแบบข้อมูลที่สามารถนำไปใช้ในการสร้างแผนภาพเสียก่อน โดยการคลิกขวาที่ code engineering set ที่สร้างขึ้นแล้วเลือกคำสั่ง Reverse ดังรูปที่ ง-7 จากนั้นจะปรากฏหน้าต่างสำหรับการเลือกตัวเลือกในการแปลงกลับดังรูปที่ ง-8



รูปที่ ง-7 การเลือกคำสั่งเพื่อทำการแปลงกลับข้อมูล

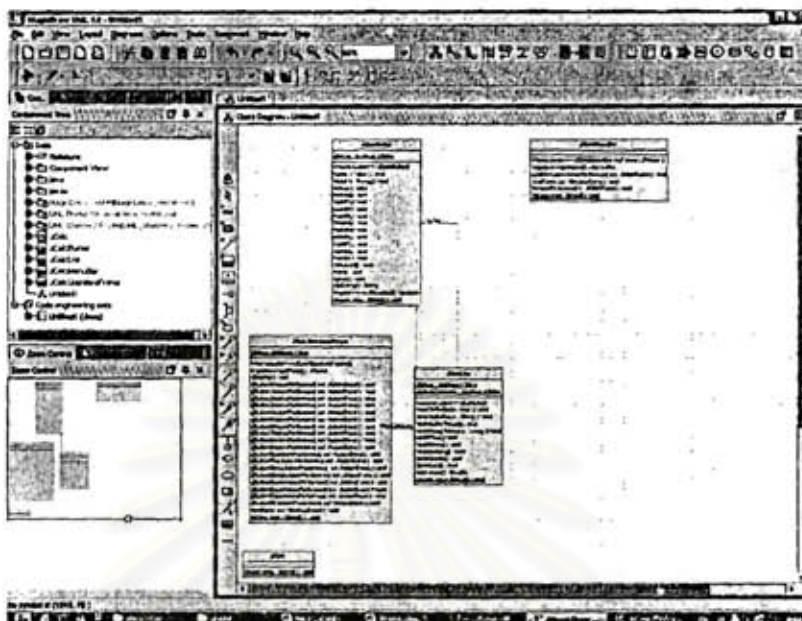


รูปที่ ง-8 ตัวเลือกการแปลงกลับ

ตัวเลือกการแปลงกลับมี 4 ตัวเลือกด้วยกัน ดังนี้

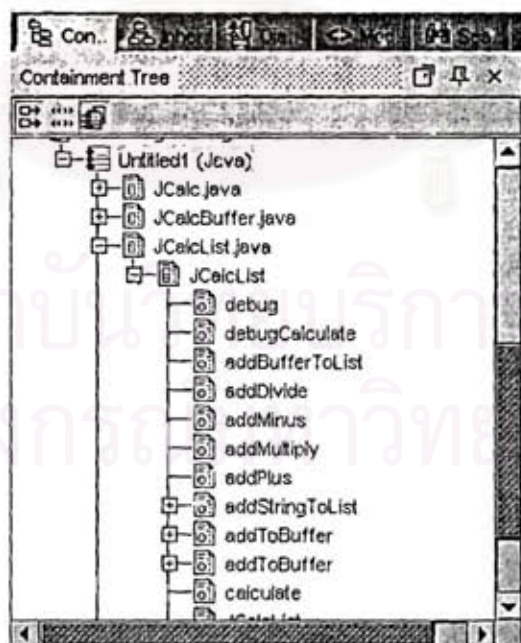
- ตัวเลือก Visualization คือตัวเลือกในการสร้างแผนภาพที่ได้จากการแปลงกลับ โดยสามารถเลือกให้แสดงตัวช่วยสร้างแผนภาพ หรือสร้างแผนภาพคลาสโดยอัตโนมัติ
- ตัวเลือก General คือตัวเลือกให้โปรแกรมทำการมองหาคลาสเพิ่มเติมในคลาสพาธหรือไม่
- ตัวเลือก Create class fields as คือตัวเลือกการแสดงผลความสัมพันธ์แอสโซซิเอชันระหว่างคลาส ให้อยู่ในรูปแบบของแอททริบิวต์ของอีกคลาส หรือแสดงเป็นความสัมพันธ์แอสโซซิเอชัน
- ตัวเลือก Model refresh type คือตัวเลือกเงื่อนไขการทำให้โมเดลและซอร์สโค้ดสอดคล้องกัน โดยมีตัวเลือกคือ ให้ทั้งโมเดลและซอร์สโค้ดทำการแก้ไขให้สอดคล้องกันเมื่อมีการแก้ไขฝ่ายหนึ่งฝ่ายใด หรือให้โมเดลทำการแก้ไขให้สอดคล้องกับซอร์สโค้ดเพียงฝ่ายเดียว

เมื่อทำการแปลงกลับข้อมูลเสร็จสิ้นแล้วหากมีการเลือกให้โปรแกรมทำการสร้างแผนภาพคลาสโดยอัตโนมัติ จะปรากฏแผนภาพคลาสที่สร้างขึ้นอยู่ในหน้าต่างการทำงานดังรูปที่ ง-9



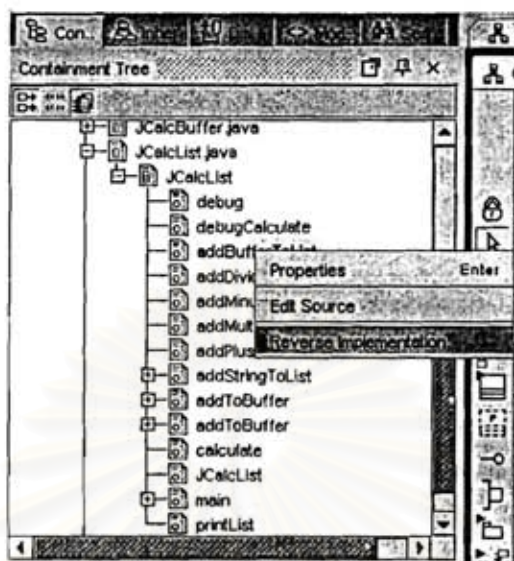
รูปที่ ง-9 แผนภาพคลาสที่ได้จากการแปลงกลับ

เมื่อทำการแปลงกลับข้อมูลจากซอร์สโค้ดพร้อมทั้งแผนภาพคลาสเสร็จสิ้นแล้ว จะปรากฏการเปลี่ยนแปลงที่รายการของไฟล์ซอร์สโค้ด โดยจะมีรายการของคลาส แอพริวิวด์ และเมทอดที่อยู่ในแต่ละไฟล์ปรากฏขึ้นมา ดังรูปที่ ง-10



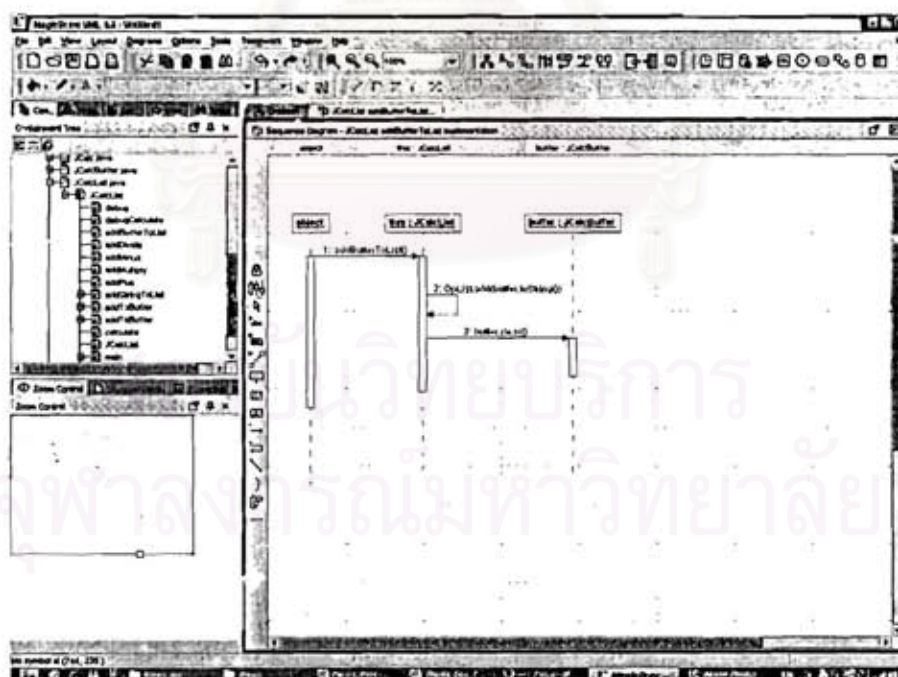
รูปที่ ง-10 รายการคลาสภายในไฟล์ซอร์สโค้ด

การแปลงกลับแผนภาพซีคอนซ์สามารถทำได้โดย ทำการเลือกเมทอดที่ต้องการจากนั้นคลิกขวาที่เมทอดแล้วเลือกคำสั่ง Reverse Implementation ดังรูปที่ ง-11



รูปที่ ง-11 การแปลงกลับแผนภาพซีคอนซ์

เมื่อทำการแปลงกลับแผนภาพซีคอนซ์เสร็จสิ้นแล้วจะปรากฏแผนภาพซีคอนซ์ขึ้นบริเวณหน้าต่างการทำงานดังรูปที่ ง-12 ทั้งนี้ในการทำงานวิจัยนี้ต้องทำการแปลงกลับแผนภาพซีคอนซ์ของทุกๆ เมทอด ในทุกๆ คลาสเพื่อให้ได้รายละเอียดมากที่สุด

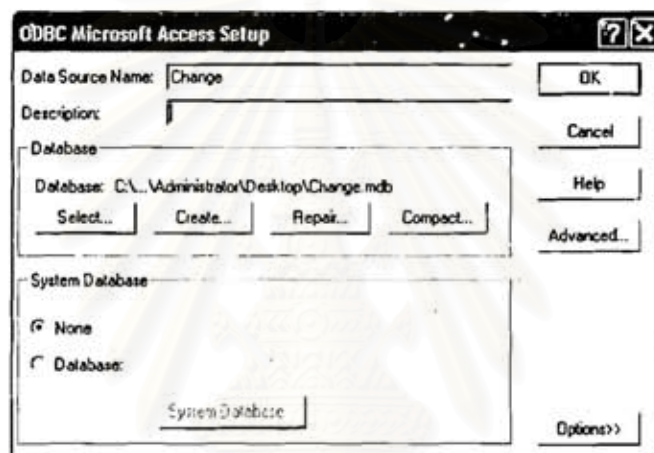


รูปที่ ง-12 แผนภาพซีคอนซ์ที่ได้จากการแปลงกลับ

ภาคผนวก จ

การใช้งานโปรแกรมคำนวณความเสถียรเชิงตรรกะ

โปรแกรมคำนวณความเสถียรเชิงตรรกะที่พัฒนาขึ้นจะทำการคำนวณความเสถียรเชิงตรรกะจากไฟล์ซอร์สโค้ดที่ผู้ใช้เลือก แล้วบันทึกค่าลงในฐานข้อมูลที่กำหนดไว้ ดังนั้นก่อนการใช้งานโปรแกรมคำนวณความเสถียรเชิงตรรกะ ต้องทำการกำหนดช่องทางติดต่อฐานข้อมูลผ่านโอดีบีซี โดยใช้ ไดรเวอร์ของไมโครซอฟต์แอคเซสและกำหนดดาต้าซอร์สเนม ชื่อ "Change" และไฟล์ดาต้าเบสชื่อ "change.mdb" เสียก่อนดังแสดงในรูปที่ จ-1



รูปที่ จ-1 การตั้งค่าโอดีบีซี

โปรแกรมคำนวณความเสถียรเชิงตรรกะมีการทำงานผ่านทางส่วนต่อประสานผู้ใช้ โดยเมื่อทำการเปิดโปรแกรมจะปรากฏหน้าจอการทำงานหลักดังรูปที่ จ-2

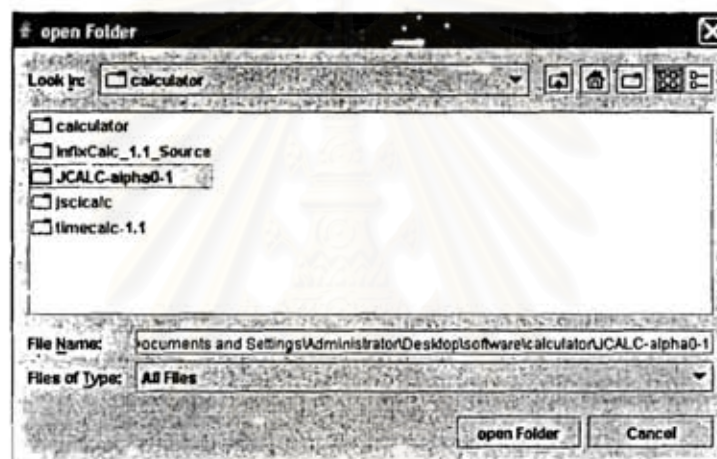


รูปที่ จ-2 หน้าจอการทำงานหลักของโปรแกรมคำนวณความเสถียรเชิงตรรกะ

โดยในหน้าจอการทำงานหลักจะมีปุ่มคำสั่ง 3 ปุ่มบริเวณด้านล่างดังนี้

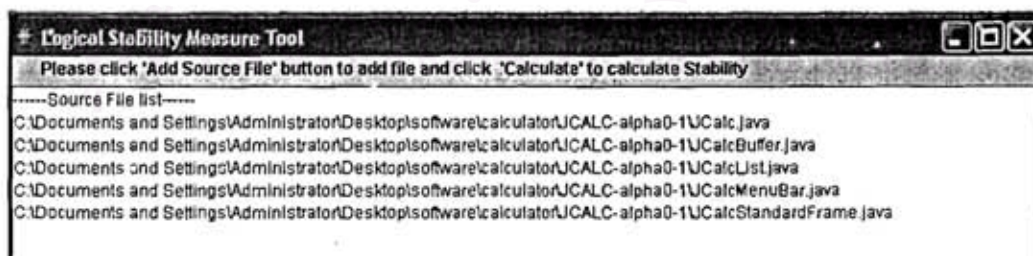
- ปุ่มคำสั่ง Add file สำหรับทำการเลือกไฟล์ซอร์สโค้ด
- ปุ่มคำสั่ง Calculate สำหรับสั่งให้โปรแกรมทำการคำนวณ
- ปุ่มคำสั่ง Reset สำหรับสั่งให้ทำการล้างข้อมูลทั้งหมด

การเลือกไฟล์ซอร์สโค้ดเข้าสู่โปรแกรมสามารถทำได้โดยการคลิกปุ่มคำสั่ง Add file และเมื่อทำการคลิกปุ่มคำสั่งแล้ว โปรแกรมจะแสดงหน้าจอสำหรับเลือกไฟล์ซอร์สโค้ดที่ต้องการ ทั้งนี้ การเลือกไฟล์ซอร์สโค้ดของโปรแกรมคำนวณความเสถียรเชิงตรรกะ ผู้ใช้ต้องทำการเลือกไฟล์เดอร์ที่เก็บไฟล์ซอร์สโค้ดที่ต้องการไว้ จากนั้นโปรแกรมจะทำการอ่านค่าไฟล์ซอร์สโค้ดทั้งหมดในไฟล์เดอร์นั้น และไฟล์เดอร์ย่อย หน้าจอการเลือกไฟล์เดอร์แสดงดังรูปที่ ๑-3

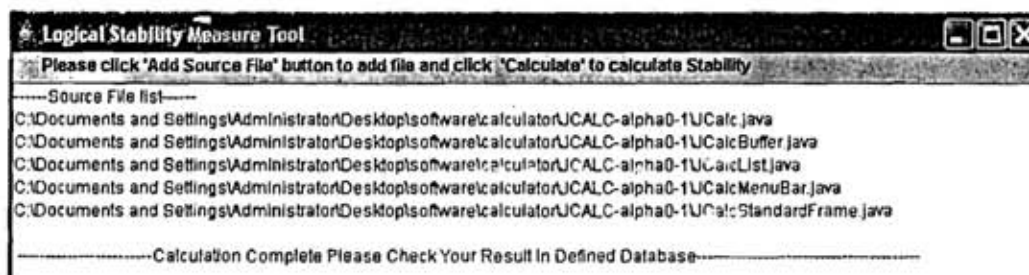


รูปที่ ๑-3 การเลือกไฟล์ซอร์สโค้ด

เมื่อผู้ใช้ทำการเลือกไฟล์ซอร์สโค้ดเสร็จสิ้น โปรแกรมจะแสดงรายการของไฟล์ที่เลือกทั้งหมดให้ผู้ได้รับทราบ ดังแสดงในรูปที่ ๑-4 จากนั้นผู้ใช้สามารถทำการคำนวณค่าความเสถียรเชิงตรรกะได้โดยคลิกที่ปุ่ม Calculate และโปรแกรมจะทำการคำนวณความเสถียรเชิงตรรกะพร้อมทั้งบันทึกผลลงในฐานข้อมูลที่ระบุไว้โดยอัตโนมัติ แล้วจึงทำการแจ้งให้ผู้ได้รับทราบดังรูปที่ ๑-5 ตัวอย่างข้อมูลที่บันทึกลงในฐานข้อมูลแสดงในรูปที่ ๑-6



รูปที่ ๑-4 รายการของไฟล์ซอร์สโค้ดที่เลือก



รูปที่ จ-5 หน้าจอการทำงานเมื่อโปรแกรมคำนวณเสร็จ

ClassTable : ตาราง					
	ID	ProjName	PackageName	ClassName	ClassStability
	1618	null	None	C1	26086956
	1619	null	None	JCalc	9514019
	1620	null	null	WindowAdapter	1
	1621	null	None	JCalcBuffer	7905542
	1622	null	None	JCalcList	6130841
	1623	null	None	JCalcMenuBar	7738318

รูปที่ จ-6 ตัวอย่างข้อมูลในฐานข้อมูล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

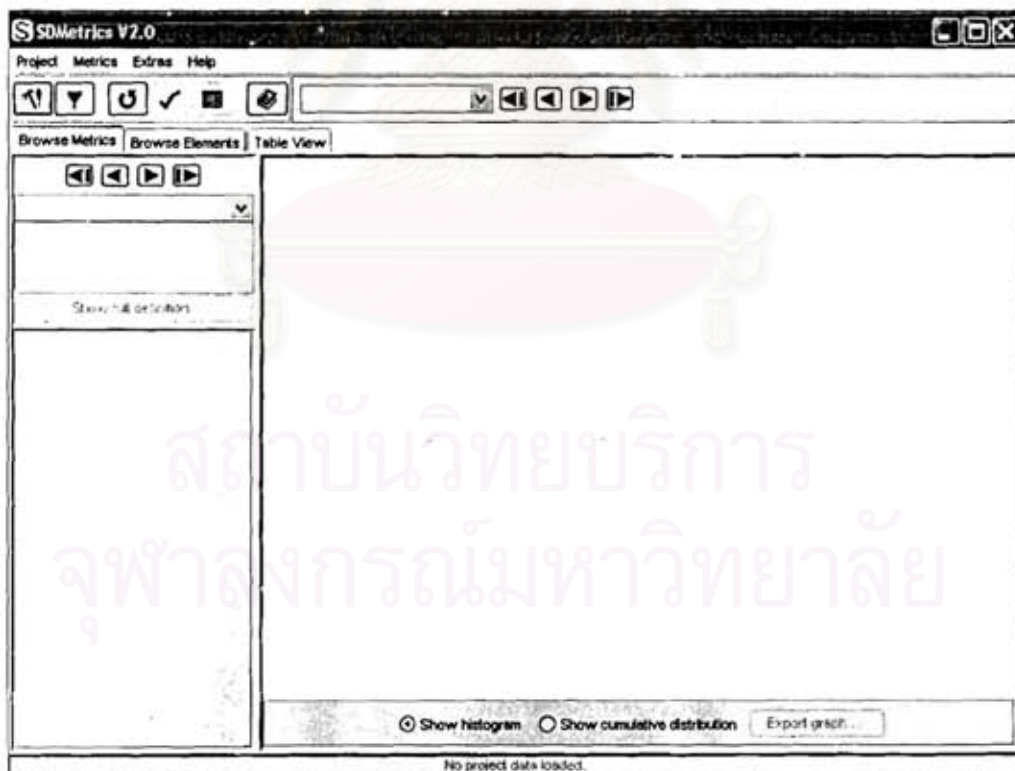
ภาคผนวก จ

การใช้งานโปรแกรม SDMetrics

ในที่นี้จะกล่าวถึงการใช้งานโปรแกรม SDMetrics ในส่วนของการใช้งานหลักเพื่อทำการคำนวณค่าตัววัดแผนภาพเท่านั้น สำหรับรายละเอียดของโปรแกรมโดยละเอียดสามารถศึกษาเพิ่มเติมได้ในเว็บไซต์ของโปรแกรมในรายการอ้างอิง

โปรแกรม SDMetric มีความสามารถในการคำนวณค่าตัววัดแผนภาพ โดยรับค่าแผนภาพที่อยู่ในรูปแบบไฟล์เอกซ์เอ็มแอล หรือเอกซ์เอ็มไอ แล้วทำการอ่านค่าแผนภาพที่ต้องการตามที่มีการระบุไว้ในไฟล์ Metamodel definition โดยวิธีการในการอ่านค่าแผนภาพจากไฟล์ข้อมูลแผนภาพต้องระบุไว้ในไฟล์ XMI Transformation จากนั้นเมื่อได้ข้อมูลแผนภาพที่ต้องการแล้วจึงทำการคำนวณค่าตัววัดแผนภาพต่างๆ ตามที่มีการระบุไว้ในไฟล์ Metrics definition

หน้าจอการทำงานหลักของโปรแกรม SDMetrics แสดงดังรูปที่ จ-1 โดยจะมีเมนูคำสั่งและปุ่มคำสั่งต่างๆ อยู่ด้านบน และมีการแสดงผลการคำนวณตัววัดต่างๆ อยู่ด้านล่าง

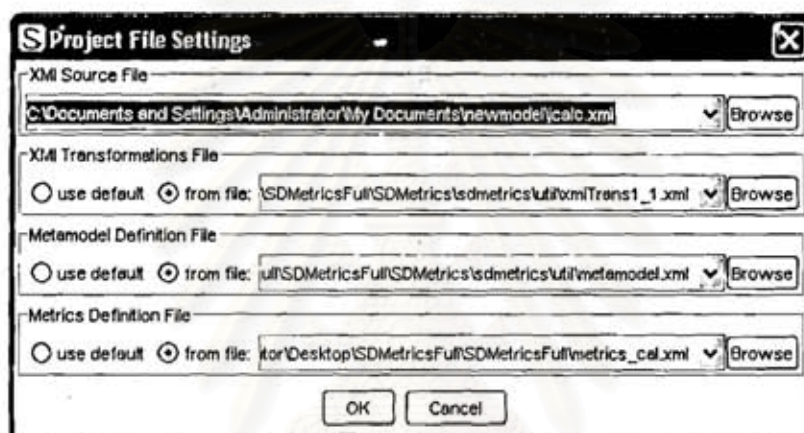


รูปที่ จ-1 หน้าจอการทำงานหลักของโปรแกรม SDMetrics

ในการคำนวณตัววัดแผนภาพจะต้องทำการตั้งค่าโปรเจกต์เพื่อระบุข้อมูลที่จำเป็นดังนี้

1. ไฟล์ของข้อมูลแผนภาพ
2. ไฟล์ Metamodel definition ที่ต้องการใช้
3. ไฟล์ XMI transformation ที่ต้องการใช้
4. ไฟล์ Metrics definition ที่ต้องการใช้

การตั้งค่าโปรเจกต์สามารถทำได้โดยการเลือกคำสั่ง Project->Project File Setting บนแถบเมนูด้านบน หรือคลิกปุ่ม  บนแถบเครื่องมือด้านบน จากนั้นโปรแกรมจะแสดงหน้าจอสำหรับการตั้งค่าโปรเจกต์ดังรูปที่ ข-2

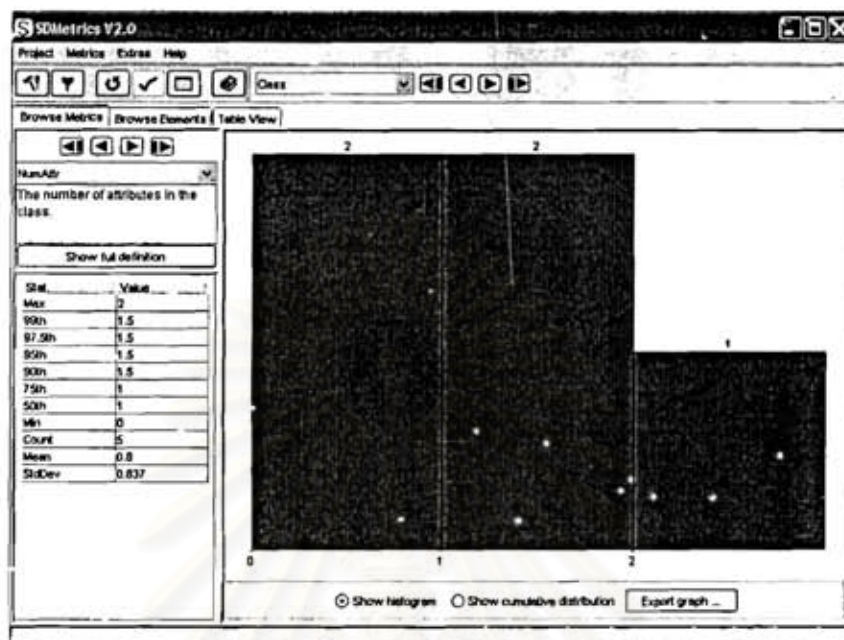


รูปที่ ข-2 การตั้งค่าโปรเจกต์

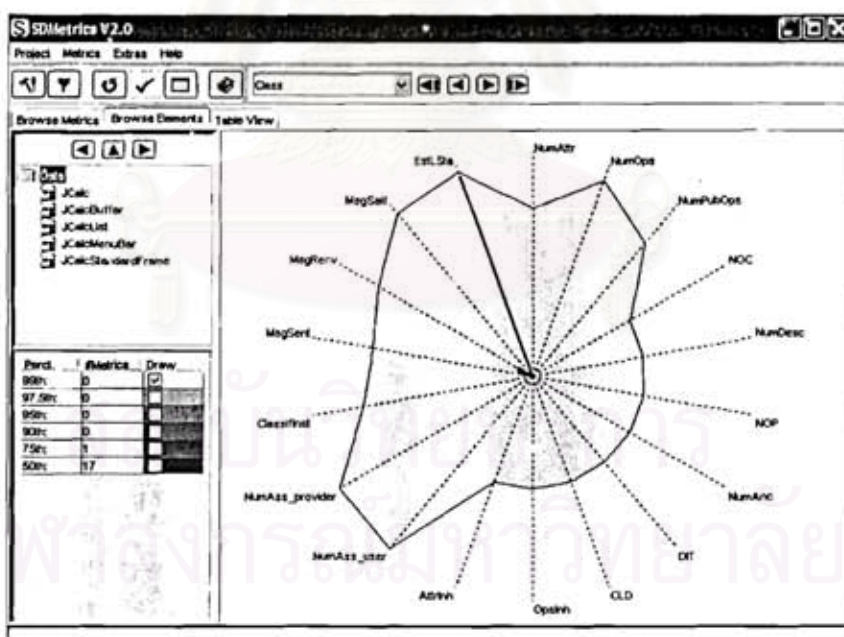
จากรูปที่ ข-2 ผู้ใช้ต้องทำการเลือกไฟล์ข้อมูลแผนภาพที่ต้องการ โดยการคลิกปุ่ม Browse เพื่อเลือกไฟล์ที่ต้องการ ในขณะที่ไฟล์ Metamodel definition ไฟล์ XMI Transformation และ ไฟล์ Metrics definition ผู้ใช้สามารถเลือกไฟล์ที่ต้องการโดยการคลิกปุ่ม Browse หรือเลือกใช้ค่ามาตรฐานของโปรแกรมก็ได้ ทั้งนี้ในงานวิจัยนี้ผู้ใช้ต้องทำการเลือก ไฟล์ Metamodel definition และไฟล์ XMI Transformation ที่กำหนดไว้สำหรับงานวิจัยนี้โดยเฉพาะ เช่นเดียวกับไฟล์ Metrics definition ซึ่งต้องเลือกใช้ไฟล์ที่กำหนดไว้เช่นกัน ยกเว้นเพียงแต่ไฟล์ Metrics definition ที่ใช้ในงานวิจัยนี้จะมี 3 ไฟล์ด้วยกันตามประเภทของซอฟต์แวร์ที่ต้องการ โดยผู้ใช้ต้องทำการเลือกให้ถูกต้อง

หลังจากทำการตั้งค่าโปรเจกต์เสร็จสิ้นแล้วสามารถทำการคำนวณตัววัดแผนภาพได้โดยเลือกคำสั่ง Metrics->Calculate Metrics บนแถบเมนู หรือคลิกปุ่ม  บนแถบเครื่องมือ จากนั้นโปรแกรมจะทำการคำนวณค่าตัววัดแผนภาพโดยอัตโนมัติแล้วจึงแสดงผลเมื่อทำการคำนวณเสร็จสิ้น โดยการแสดงผลจะแบ่งออกเป็น 3 แบบ คือ การแสดงผลข้อมูลสถิติตัววัด การแสดงผลข้อมูล

แผนภาพตัววัดขององค์ประกอบ และการแสดงผลแบบตาราง โดยผู้ใช้สามารถเลือกรูปแบบการแสดงผลที่ต้องการได้จากแท็บแบบทั้ง 3 คือ Browse Metrics Browse Element และ Table View ภาพหน้าจอกการแสดงผลทั้ง 3 แบบแสดงในรูปที่ จ-3 รูปที่ จ-4 และรูปที่ จ-5 ตามลำดับ



รูปที่ จ-3 หน้าจอการแสดงผลข้อมูลสถิติตัววัด



รูปที่ จ-4 หน้าจอการแสดงผลข้อมูลแผนภาพตัววัดขององค์ประกอบ

Name	s_user	NumAss_provider	EC_Abr	IC_Abr	EC_Par	IC_Par	ObjSet	ClassInst	MapSent	MapRecy	MapSelf	EstSta
Data_Kick	0	0	0	0	0	0	0	0	0	0	0	0.9791283
Data_KickButler	1	0	0	0	0	0	47	8	31	76	0	0.91682374
Data_KickList	1	0	0	0	0	0	21	16	19	99	0	0.95572102
Data_KickMenuBar	0	0	0	0	0	0	8	1	1	39	0	0.98267347
Data_KickStandardFrame	0	0	0	0	0	0	26	34	8	110	0	0.97323221

รูปที่ จ-5 หน้าจอการแสดงผลแบบตาราง

นอกจากนี้ในการแสดงผลแต่ละรูปแบบยังแบ่งระดับของข้อมูลแผนภาพที่นำมาแสดงผลได้อีกด้วยตามแต่ที่มีการกำหนดไว้ในไฟล์ Metamodel definition โดยในงานวิจัยนี้จะมี 3 ระดับด้วยกัน คือ ระดับคลาส ระดับแพคเกจ และระดับซอฟต์แวร์ โดยสามารถเลือกดูข้อมูลในแต่ละระดับได้จากตัวเลือกบนแถบเครื่องมือดังรูปที่ จ-6

Name	NumAss	NumOps	NumPubOps	Netting	First	HOC	NumDesc	NOP	NumAnc	DT	CLD	OpStr	AbStr	NumAss_user
Data_Kick	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Data_KickButler	1	20	19	0	0	1	1	0	0	0	1	0	0	0
Data_KickList	2	12	11	0	0	0	0	1	1	1	0	20	1	1
Data_KickMenuBar	0	8	0	0	0	0	0	0	0	0	0	0	0	0
Data_KickStandardFrame	1	23	0	0	0	0	0	0	0	0	0	0	0	1

รูปที่ จ-6 การเลือกระดับของการแสดงผล

ประวัติผู้เขียนวิทยานิพนธ์

นาย ศุภวัชร รัชสิยวัฒน์ เกิดเมื่อวันที่ 3 กันยายน พ.ศ. 2524 จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนสวนกุหลาบวิทยาลัย จากนั้นได้เข้าศึกษาต่อในระดับปริญญาตรี สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี และสำเร็จการศึกษาในปี พ.ศ. 2546 และเข้าศึกษาต่อในระดับปริญญาโท สาขาบริหารธุรกิจที่จุฬาลงกรณ์มหาวิทยาลัย ในหลักสูตรวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ในปีเดียวกัน



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย