

การตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์
โดยการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด

นางสาวปัทมาภรณ์ สายลิม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2555

ลิขสิทธิ์ของไฟล์ฉบับนี้สงวนลิขสิทธิ์โดยมหาวิทยาลัย
บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

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

CHECKING COMPLETENESS OF SOFTWARE CHANGE REQUIREMENTS
BY COMPARISON OF CONCEPTUAL CLASS DIAGRAMS

Miss Pattamaporn Saisim

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

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การตรวจสอบความครบถ้วนของความถี่ในการเปลี่ยนแปลงซอฟต์แวร์โดยการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด
โดย	นางสาวปัทมาภรณ์ สายสิม
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ทวีชัย เสนีวงศ์ ณ อยุธยา

คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสมเลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.พรศิริ หมั่นไชยศิริ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ทวีชัย เสนีวงศ์ ณ อยุธยา)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.มณฑุปายาส ทองมาก)

บททบทวน สยาลิม : การตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์โดยการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด. (CHECKING COMPLETENESS OF SOFTWARE CHANGE REQUIREMENTS BY COMPARISON OF CONCEPTUAL CLASS DIAGRAMS) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.ทวีติย์ เสนิงค์ ณ อยุธยา, 99 หน้า.

ปัญหาที่พบในการเปลี่ยนแปลงระบบซอฟต์แวร์โดยพัฒนาระบบใหม่บนพื้นฐานของข้อกำหนดความต้องการของระบบเดิม คือ ฟังก์ชันงานหรือข้อมูลที่เคยมีในระบบเดิมและยังต้องการคงไว้ เกิดการตกหล่นไปในการออกแบบระบบใหม่ จึงอาจส่งผลให้การส่งมอบงานล่าช้ากว่ากำหนด เนื่องจากต้องแก้ไขให้ระบบใหม่มีข้อมูลหรือฟังก์ชันงานที่ครบถ้วนถูกต้องก่อน งานวิจัยนี้นำเสนอวิธีการตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์ตั้งแต่ในระยะเริ่มต้นของโครงการซอฟต์แวร์ใหม่ โดยการนำแผนภาพคลาสเชิงแนวคิดของระบบเดิมที่มีอยู่แล้วและของระบบที่ออกแบบใหม่มาเปรียบเทียบกัน อัลกอริทึมที่ใช้จะพิจารณาทั้งความคล้ายกันของโครงสร้างและความคล้ายกันในเชิงความหมายของการตั้งชื่อในแผนภาพ ผลลัพธ์ของการเปรียบเทียบจะทำให้เห็นถึงความคล้ายคลึงและความแตกต่างของแผนภาพทั้งสอง อันจะช่วยให้นักวิเคราะห์ระบบและผู้ใช้สามารถตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลง สำหรับการออกแบบเชิงแนวคิดของระบบใหม่ที่จะพัฒนาต่อไปได้

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....
 สาขาวิชา.....วิศวกรรมซอฟต์แวร์.....ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....
 ปีการศึกษา..2555.....

5470966521 : MAJOR SOFTWARE ENGINEERING

KEYWORDS : REQUIREMENT CHANGE / CONCEPTUAL CLASS DIAGRAM /
WORDNET

PATTAMAPORN SAISIM : CHECKING COMPLETENESS OF SOFTWARE
CHANGE REQUIREMENTS BY COMPARISON OF CONCEPTUAL CLASS
DIAGRAMS. ADVISOR : ASSOC. PROF. TWITTIE SENIVONGSE, Ph.D., 99
pp.

An important problem in development of a new software system based on the requirements specification of the original system is that certain functions or data that are present in the original system and should be retained are missing from the new system. This problem can delay product delivery as the new system will need to be fixed to fulfill all functional and data requirements. This paper presents a method to check for completeness of software change requirements at the early stage of the new software project by comparing the conceptual class diagram of the original system with that of the new system to be developed. The algorithm considers structural similarity and semantic similarity of names in both diagrams. The comparison result can identify similarities and differences between the two diagrams, and hence can support system analysts and users in checking for completeness of change requirements for the conceptual design of the new system.

Department : Computer Engineering Student's Signature

Field of Study : Software Engineering Advisor's Signature

Academic Year : 2012

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณ รองศาสตราจารย์ ดร.ทวีชัย เสนีวงศ์ ณ อยุธยา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่เสียสละเวลาให้ความช่วยเหลือและคำปรึกษา ตลอดจนให้คำแนะนำที่มีประโยชน์ ทำให้การจัดทำวิทยานิพนธ์สำเร็จลุล่วงไปด้วยดี ขอขอบพระคุณคณาจารย์ทุกท่านที่ได้ให้คำแนะนำสั่งสอน และให้ความรู้แก่ข้าพเจ้าตลอดระยะเวลาการศึกษา

ขอขอบพระคุณ รองศาสตราจารย์ ดร.พรศิริ หมั่นไชยศรี ประธานกรรมการสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร.มธุปายาส ทองมาก กรรมการสอบวิทยานิพนธ์ ที่กรุณาให้คำแนะนำต่างๆ ทำให้วิทยานิพนธ์นี้มีความถูกต้องและสมบูรณ์มากขึ้น

ขอขอบพระคุณ คุณพ่อ คุณแม่ และพี่สาวทุกคนที่คอยห่วงใย ให้กำลังใจ พร้อมทั้งให้ความสนับสนุนตลอดการศึกษาจนสามารถสำเร็จลุล่วงได้

ข้าพเจ้าหวังเป็นอย่างยิ่งว่างานวิทยานิพนธ์นี้จะเป็นประโยชน์ต่อผู้ศึกษา ค้นคว้า และสนใจ หากมีความผิดพลาดประการใด ข้าพเจ้าขอน้อมรับไว้เพื่อนำไปใช้ในการปรับปรุงให้วิทยานิพนธ์นี้มีความสมบูรณ์ยิ่งขึ้น อันความดีและประโยชน์จากวิทยานิพนธ์นี้ข้าพเจ้าขอมอบให้แก่ผู้มีพระคุณทุกท่าน

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฅ
สารบัญภาพ.....	ญ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 ขั้นตอนการวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 ผลงานตีพิมพ์.....	4
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	5
2.1 แนวคิดและทฤษฎี.....	5
2.1.1 ยูเอ็มแอล.....	5
2.1.2 เอกซ์เอ็มไอ.....	6
2.1.3 การวัดค่าความคล้ายคลึงกันของสายอักขระ.....	7
2.1.4 เวิร์ดเน็ต.....	8
2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	9
2.2.1 งานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบแผนภาพคลาส.....	9
2.2.2 งานวิจัยที่เกี่ยวข้องกับการใช้เวิร์ดเน็ต.....	12
3 การออกแบบอัลกอริทึม S-UMLDiff.....	13
3.1 ขั้นตอนการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด.....	13
3.2 การทำงานของอัลกอริทึม S-UMLDiff.....	18

บทที่	หน้า
3.3 ค่าขีดแบ่งของความคล้ายคลึง (Similarity Threshold).....	20
3.4 การพิจารณาความคล้ายกันของชื่อเอนทิตี.....	20
3.5 การพิจารณาความคล้ายกันของโครงสร้าง.....	22
3.6 การพิจารณาความคล้ายกันโดยรวม.....	23
3.7 ตัวอย่างการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดด้วยอัลกอริทึม S-UMLDiff.....	27
4 การพัฒนาเครื่องมือ.....	30
4.1 การออกแบบฟังก์ชันการทำงานของเครื่องมือ.....	30
4.2 การออกแบบแผนภาพคลาสของเครื่องมือ.....	32
4.3 การออกแบบส่วนต่อประสานผู้ใช้.....	32
4.4 สภาพแวดล้อมในการพัฒนาเครื่องมือ.....	36
4.4.1 ฮาร์ดแวร์.....	36
4.4.2 ซอฟต์แวร์.....	36
5 ผลการทดสอบ.....	37
5.1 ผลการทดสอบเปรียบเทียบแผนภาพคลาสเชิงแนวคิดจำนวน10ชุด.....	37
5.2 การทดสอบเปรียบเทียบแผนภาพคลาสเชิงแนวคิดของระบบงานจริง.....	38
5.3 อภิปรายผล.....	41
6 สรุปผลการวิจัย ข้อจำกัด และข้อเสนอแนะ.....	43
6.1 สรุปผลการวิจัย.....	43
6.3 อุปสรรคและข้อจำกัด.....	43
6.4 ข้อเสนอแนะ.....	44
รายการอ้างอิง.....	45
ภาคผนวก.....	48
ภาคผนวก ก ตัวอย่างของผลลัพธ์ที่ได้จากการเปรียบเทียบด้วย S-UMLDiff.....	49
ภาคผนวก ข ตัวอย่างของแผนภาพคลาสเชิงแนวคิดที่ใช้ในการทดสอบ.....	52
ภาคผนวก ค ผลลัพธ์ที่ได้จากการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดของระบบงานจริง.....	73
ประวัติผู้เขียนวิทยานิพนธ์.....	99

สารบัญตาราง

ตารางที่		หน้า
3.1	ตัวอย่างเอนทิตีที่ได้จากแผนภาพระบบเดิมภาพที่ 3.2.....	16
3.2	การกำหนดความสัมพันธ์ระหว่างคลาส.....	17
3.3	ตัวอย่างความสัมพันธ์ที่ได้จากแผนภาพระบบใหม่ภาพที่ 3.3.....	17
3.4	ประเภทของความเปลี่ยนแปลงที่อัลกอริทึม S-UMLDiff รายงานผล.....	17
3.5	การเปรียบเทียบเชิงโครงสร้าง.....	22
4.1	คำอธิบายฟังก์ชันการทำงาน.....	31
5.1	ผลการทดสอบการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดด้วย S-UMLDiff.....	37
5.2	คุณสมบัติของผู้เชี่ยวชาญที่ร่วมทำการทดสอบ.....	38
ก.1	ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 3.2 และ 3.3...	49
ข.1	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 1.....	53
ข.2	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 2.....	54
ข.3	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 3.....	56
ข.4	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 4.....	57
ข.5	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 5.....	60
ข.6	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 6.....	62
ข.7	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 7.....	64
ข.8	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 8.....	67
ข.9	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 9.....	69
ข.10	การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 10.....	71
ค.1	ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ S-UMLDiff.....	73
ค.2	ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff.....	81
ค.3	ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3.....	90

สารบัญภาพ

ภาพที่		หน้า
2.1	แผนภาพคลาสเชิงแนวคิด.....	6
2.2	การเปรียบเทียบแผนภาพคลาสที่ Xing นำเสนอ.....	10
2.3	ส่วนประกอบของอัลกอริทึม UMLDiff.....	11
3.1	ขั้นตอนการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด.....	14
3.2	แผนภาพคลาสเชิงแนวคิดของระบบเดิม.....	15
3.3	แผนภาพคลาสเชิงแนวคิดของระบบใหม่.....	15
3.4	ขั้นตอนการทำงานของอัลกอริทึม S-UMLDiff	19
3.5	การตรวจสอบการเปลี่ยนชื่อของอัลกอริทึม S-UMLDiff.....	26
4.1	แผนภาพยูสเคสแสดงฟังก์ชันการทำงานของเครื่องมือ.....	30
4.2	แผนภาพคลาสเชิงแนวคิดของเครื่องมือที่พัฒนา.....	32
4.3	หน้าจอ Log in เพื่อเข้าสู่ระบบ.....	33
4.4	หน้าจอที่ใช้ในการนำเข้าเอกสารเอกซ์เอ็มไอ.....	33
4.5	ตัวอย่างการระบุรหัสในแผนภาพคลาสเชิงแนวคิด.....	33
4.6	ตัวอย่างการระบุรหัสและเอกสารเอกซ์เอ็มไอ.....	34
4.7	หน้าจอแสดงผลลัพธ์ที่ได้จากการอ่านข้อมูลจากเอกสารเอกซ์เอ็มไอ.....	34
4.8	หน้าจอสำหรับระบุค่าขีดแบ่ง.....	35
4.9	หน้าจอแสดงผลลัพธ์ของการเปรียบเทียบ.....	35
5.1	แผนภาพคลาสเชิงแนวคิดของระบบงานเดิมที่ออกแบบโดยผู้เชี่ยวชาญคนที่ 1..	39
5.2	แผนภาพคลาสเชิงแนวคิดของระบบงานใหม่ที่ออกแบบโดยผู้เชี่ยวชาญคนที่ 2.	40
ข.1	แผนภาพคลาสเชิงแนวคิดชุดที่ 1 ระบบงานเดิม.....	52
ข.2	แผนภาพคลาสเชิงแนวคิดชุดที่ 1 ระบบงานใหม่.....	52
ข.3	แผนภาพคลาสเชิงแนวคิดชุดที่ 2 ระบบงานเดิม.....	53
ข.4	แผนภาพคลาสเชิงแนวคิดชุดที่ 2 ระบบงานใหม่.....	54
ข.5	แผนภาพคลาสเชิงแนวคิดชุดที่ 3 ระบบงานเดิม.....	55
ข.6	แผนภาพคลาสเชิงแนวคิดชุดที่ 3 ระบบงานใหม่.....	55
ข.7	แผนภาพคลาสเชิงแนวคิดชุดที่ 4 ระบบงานเดิม.....	56
ข.8	แผนภาพคลาสเชิงแนวคิดชุดที่ 4 ระบบงานใหม่.....	57

ภาพที่		หน้า
ข.9	แผนภาพคลาสเชิงแนวคิดชุดที่ 5 ระบบงานเดิม.....	59
ข.10	แผนภาพคลาสเชิงแนวคิดชุดที่ 5 ระบบงานใหม่.....	59
ข.11	แผนภาพคลาสเชิงแนวคิดชุดที่ 6 ระบบงานเดิม.....	61
ข.12	แผนภาพคลาสเชิงแนวคิดชุดที่ 6 ระบบงานใหม่.....	61
ข.13	แผนภาพคลาสเชิงแนวคิดชุดที่ 7 ระบบงานเดิม.....	63
ข.14	แผนภาพคลาสเชิงแนวคิดชุดที่ 7 ระบบงานใหม่.....	64
ข.15	แผนภาพคลาสเชิงแนวคิดชุดที่ 8 ระบบงานเดิม.....	66
ข.16	แผนภาพคลาสเชิงแนวคิดชุดที่ 8 ระบบงานใหม่.....	66
ข.17	แผนภาพคลาสเชิงแนวคิดชุดที่ 9 ระบบงานเดิม.....	68
ข.18	แผนภาพคลาสเชิงแนวคิดชุดที่ 9 ระบบงานใหม่.....	68
ข.19	แผนภาพคลาสเชิงแนวคิดชุดที่ 10 ระบบงานเดิม.....	70
ข.20	แผนภาพคลาสเชิงแนวคิดชุดที่ 10 ระบบงานใหม่.....	71

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การพัฒนาระบบงานซอฟต์แวร์ประกอบด้วยกระบวนการที่สำคัญ ได้แก่ การพัฒนา (Development) การนำไปใช้งาน (Deployment) และการบำรุงรักษาระบบ (Maintenance) สำหรับกระบวนการบำรุงรักษาระบบนั้น หมายถึง กระบวนการเปลี่ยนแปลงซอฟต์แวร์หลังจากนำไปใช้งานแล้ว [1] ซึ่งนับว่าเป็นกระบวนการที่มีความสำคัญและใช้ต้นทุนถึง 70% ของต้นทุนที่ใช้ในการพัฒนาระบบทั้งหมด [2] การเปลี่ยนแปลงซอฟต์แวร์มีทั้งแบบที่เป็นการเพิ่มฟังก์ชันใหม่ในระบบเดิมและการพัฒนาระบบใหม่ทั้งหมดบนพื้นฐานของข้อกำหนดของระบบเดิม โดยอาจมีการเพิ่ม ลด แก้ไขโปรแกรมหรือปรับขั้นตอนการทำงานบางอย่าง สาเหตุสำคัญที่ทำให้ต้องทำการพัฒนาระบบใหม่ทั้งหมด ได้แก่ 1) เพื่อแก้ไขข้อบกพร่องของการออกแบบเดิม 2) ต้องการเพิ่มฟังก์ชันการทำงานโดยการปรับปรุงโครงสร้างให้สอดคล้องกัน 3) ลดต้นทุนในการบำรุงรักษาระบบ 4) เพิ่มประสิทธิภาพของซอฟต์แวร์ 5) ปรับปรุงขั้นตอนการดำเนินงานให้มีประสิทธิภาพ

ในปัจจุบันโครงการพัฒนาซอฟต์แวร์ที่เป็นการพัฒนาระบบใหม่บนพื้นฐานของข้อกำหนดของระบบเดิมนั้นยังมีปัญหาในเรื่องต่างๆ ตัวอย่างเช่น ระบบใหม่ที่พัฒนาข้อมูลหรือฟังก์ชันไม่ครบตามที่เคยมีในระบบเดิม ทั้งๆ ที่เป็นข้อมูลหรือฟังก์ชันที่ต้องการคงไว้ สาเหตุเกิดจากผู้ให้ความต้องการของระบบใหม่อาจหลงลืมและไม่ได้แจ้งให้ผู้พัฒนาระบบทราบ หรือผู้ให้ความต้องการไม่ทราบว่าข้อมูลหรือฟังก์ชันนี้ในระบบ ดังนั้นจึงส่งผลให้มีการใช้เวลาและทรัพยากรในการพัฒนาระบบมากเกินไป เนื่องจากต้องมีการเพิ่มหรือแก้ไขระบบภายหลังซึ่งอาจกระทบโครงสร้างของระบบที่ได้ออกแบบไว้

จากปัญหาดังกล่าวผู้วิจัยจึงมีแนวคิดที่จะเสนอวิธีการที่ใช้ในการตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์ พร้อมทั้งพัฒนาเครื่องมือสนับสนุน เนื่องจากผู้วิจัยเห็นว่าหากสามารถทราบถึงความต้องการที่ขาดหายไปตั้งแต่ช่วงของการวิเคราะห์ความต้องการจะทำให้นักวิเคราะห์ระบบสามารถแก้ไขได้ทัน่วงทีก่อนที่จะมีการออกแบบระบบในขั้นตอนต่อไป วิธีการที่ผู้วิจัยเห็นว่าเป็นวิธีการที่เหมาะสมสำหรับการตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์คือการนำแผนภาพคลาสเชิงแนวคิด (Conceptual Class Diagram) ของระบบเดิมที่มีอยู่แล้วและแผนภาพคลาสเชิงแนวคิดของระบบใหม่ที่ได้จากการวิเคราะห์ความต้องการมาเปรียบเทียบกัน เนื่องจากในการพัฒนาซอฟต์แวร์ในลักษณะนี้เป็นการพัฒนาบนพื้นฐานของข้อกำหนดเดิม แต่อาจมีการเพิ่ม ลด หรือแก้ไขการทำงานบางอย่าง

ดังนั้นหากนำแผนภาพคลาสเชิงแนวคิดของระบบเดิมและระบบใหม่มาเปรียบเทียบกัน ผลลัพธ์ที่ได้จะทำให้เห็นถึงความแตกต่างของแผนภาพคลาสทั้งสอง อันจะนำไปสู่การตรวจสอบความครบถ้วนของความต้องการของระบบใหม่ กล่าวคือในกรณีของข้อมูลหรือฟังก์ชันที่มีในระบบเดิมแต่ไม่มีในระบบใหม่นั้นเป็นส่วนที่ผู้ใช้งานต้องการลบออกจากระบบใหม่หรือไม่ และในกรณีของข้อมูลหรือฟังก์ชันที่ไม่มีในระบบเดิมแต่ถูกเพิ่มเข้ามาในระบบใหม่นั้นเป็นส่วนที่ผู้ใช้งานต้องการเพิ่มเข้ามาในระบบหรือไม่

วิธีการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในงานวิจัยนี้เริ่มด้วยการแปลงแผนภาพคลาสเชิงแนวคิดเป็นเอกสารเอกซ์เอ็มไอ (XMI : XML Metadata Interchange) [3] จากนั้นทำการสกัด (Extract) ส่วนย่อย (Element) ของแผนภาพคลาสออกมาจากเอกสารเอกซ์เอ็มไอเพื่อนำมาสร้างกราฟโดยใช้วิธีการของ Zhenchang Xing [4] แล้วทำการเปรียบเทียบกราฟด้วยอัลกอริทึม S-UMLDiff ซึ่งปรับปรุงจากอัลกอริทึม UMLDiff ที่ Zhenchang Xing ได้นำเสนอ อัลกอริทึม UMLDiff ได้ใช้วิธี LCS (Longest Common Subsequence) ในการเปรียบเทียบความคล้ายคลึงกันของสายอักขระ ได้แก่ ชื่อคลาส ลักษณะประจำ (Attribute) การดำเนินการ (Operation) เป็นต้น ผู้วิจัยจึงได้นำเวิร์ดเน็ต (WordNet) [5] มาประยุกต์ใช้เพื่อเพิ่มประสิทธิภาพของอัลกอริทึม UMLDiff ให้สามารถเปรียบเทียบคำเชิงความหมายและตรวจสอบการเปลี่ยนแปลงประเภทของส่วนย่อยได้อีกด้วย

1.2 วัตถุประสงค์ของการวิจัย

เสนอวิธีการและพัฒนาเครื่องมือที่ใช้ในการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด โดยมีการพิจารณาความคล้ายกันเชิงความหมายของคำในแผนภาพ เพื่อใช้ในการตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์

1.3 ขอบเขตของการวิจัย

- 1.3.1 ทำการแปลงแผนภาพคลาสเชิงแนวคิดเป็นเอกซ์เอ็มไอโดยใช้เครื่องมือสนับสนุน ได้แก่ โปรแกรมอาร์โกยูเอ็มแอล (ArgoUML) [6]
- 1.3.2 ออกแบบอัลกอริทึม S-UMLDiff โดยการปรับปรุงจากอัลกอริทึม UMLDiff ร่วมกับการประยุกต์ใช้เวิร์ดเน็ต
- 1.3.3 พัฒนาเครื่องมือเพื่อประยุกต์ใช้อัลกอริทึม S-UMLDiff ด้วยภาษาจาวา โดยใช้โปรแกรมอีคลิป์ส (Eclipse)

- 1.3.4 ทดสอบและประเมินผลเครื่องมือโดยการนำมาใช้เปรียบเทียบแผนภาพคลาสเชิงแนวคิดไม่น้อยกว่า 10 ชุด ร่วมกับการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดของระบบงานจริง 1 ระบบ ซึ่งเป็นการพัฒนาระบบใหม่บนพื้นฐานของข้อกำหนดของระบบเดิม แผนภาพคลาสเชิงแนวคิดของระบบงานนี้ประกอบด้วยคลาสจำนวน 20-30 คลาส

1.4 ขั้นตอนการวิจัย

- 1.4.1 ศึกษาอัลกอริทึม UMLDiff
- 1.4.2 ศึกษาเกี่ยวกับเอกซ์เอ็มไอ
- 1.4.3 ศึกษาเกี่ยวกับข้อกำหนดของยูเอ็มแอล
- 1.4.4 ศึกษาเกี่ยวกับแผนภาพคลาสเชิงแนวคิด
- 1.4.5 ศึกษาเครื่องมือที่ใช้ในการแปลงแผนภาพคลาสเป็นเอกซ์เอ็มไอ
- 1.4.6 ศึกษาเกี่ยวกับเวิร์ดเน็ต
- 1.4.7 ออกแบบอัลกอริทึม S-UMLDiff โดยการปรับปรุงและแก้ไขอัลกอริทึม UMLDiff รวมทั้งนำเวิร์ดเน็ตมาประยุกต์ใช้
- 1.4.8 พัฒนาเครื่องมือเพื่อประยุกต์ใช้อัลกอริทึม S-UMLDiff
- 1.4.9 ประเมินผลและปรับปรุงเครื่องมือ
- 1.4.10 วิเคราะห์และสรุปผล พร้อมข้อเสนอแนะ
- 1.4.11 จัดทำเอกสารผลงานทางวิชาการและตีพิมพ์
- 1.4.12 จัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ผลลัพธ์ที่ได้จากวิธีการและเครื่องมือที่นำเสนอจะทำให้เห็นถึงความแตกต่างของข้อมูลและฟังก์ชันที่มีในระบบเดิมและระบบใหม่ ทำให้สามารถตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์ได้ นักวิเคราะห์ระบบและผู้ให้ความต้องการสามารถนำผลลัพธ์นี้ไปใช้ในการพิจารณาว่าสิ่งที่ไม่ใช่ในระบบเดิมแต่ถูกเพิ่มเข้ามาในระบบใหม่นั้นเป็นสิ่งที่ต้องการเพิ่มหรือไม่ ในทางกลับกันสิ่งที่มีในระบบเดิมแต่ไม่มีในระบบใหม่นั้นเป็นสิ่งที่ไม่ต้องการให้มีในระบบใหม่หรือไม่

1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้นำเสนอในการประชุมวิชาการ ดังนี้

1. The 9thNational Conference on Computing and Information Technology (NCCIT 2013), 9 – 10 May 2013, Faculty of Information Technology, King Mongkut's University of Technology North Bangkok, Thailand ในบทความเรื่อง การตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์ โดยผู้แต่ง คือ ปัทมาภรณ์ สายสีม และ ทวีติย์ เสนีวงศ์ ณ อยุธยา

บทที่ 2

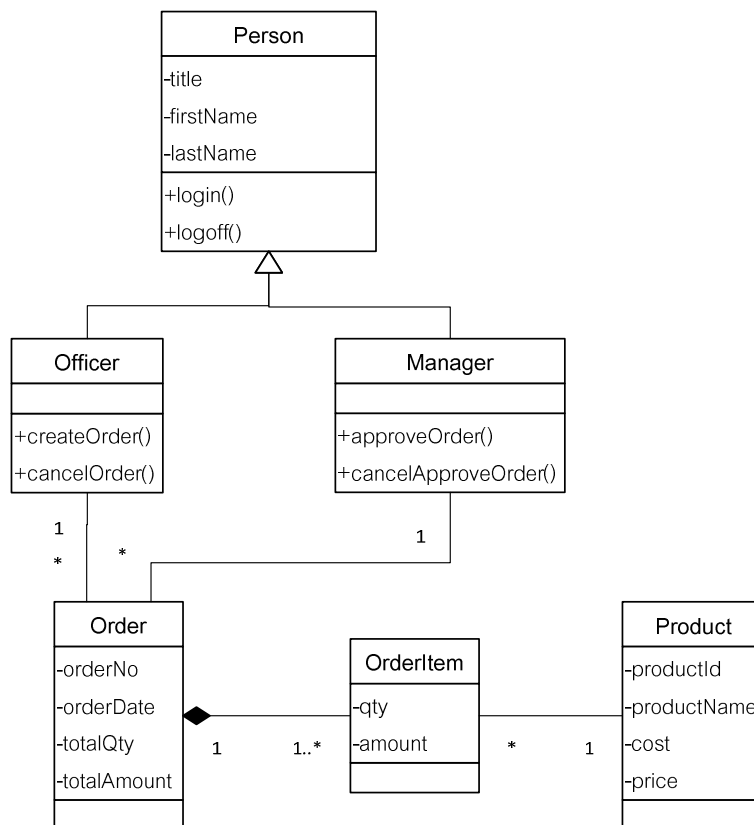
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎี

2.1.1 ยูเอ็มแอล

ยูเอ็มแอล (UML : Unified Modeling Language) [7] เป็นภาษาเชิงรูปภาพซึ่งเป็นมาตรฐานที่กำหนดขึ้นโดยโอเอ็มจี (OMG : Object Management Group) ยูเอ็มแอลถูกนำมาใช้เป็นเครื่องมือสำหรับการออกแบบระบบงานซอฟต์แวร์ เพื่อแสดงถึงแนวคิด ภาพรวมของระบบ ส่วนประกอบของซอฟต์แวร์ รวมถึงอธิบายการทำงานของโปรแกรม ยูเอ็มแอลได้มีการนิยามแผนภาพต่างๆ เพื่อใช้ในการอธิบายระบบงานในรูปแบบที่แตกต่างกัน ได้แก่ แผนภาพยูสเคส (Use Case Diagram) แผนภาพกิจกรรม (Activity Diagram) แผนภาพคลาส (Class Diagram) แผนภาพสถานะการทำงาน (Statechart Diagram) แผนภาพลำดับการทำงาน (Sequence Diagram) และแผนภาพส่วนประกอบ (Component Diagram) เป็นต้น

ในกระบวนการวิเคราะห์ความต้องการซอฟต์แวร์ได้มีการใช้แผนภาพคลาสเพื่อแสดงถึงฟังก์ชันการทำงานและข้อมูลต่างๆ ที่เกี่ยวข้องในระบบ ตลอดจนความสัมพันธ์ระหว่างข้อมูลเหล่านั้น การออกแบบแผนภาพคลาสในกระบวนการนี้เป็นการออกแบบในระดับแนวคิดหรือเรียกว่าแผนภาพคลาสเชิงแนวคิด ซึ่งไม่ได้แสดงรายละเอียดในเชิงตรรกะการเขียนโปรแกรม ตัวอย่างของแผนภาพคลาสเชิงแนวคิดแสดงดังภาพที่ 2.1



ภาพที่ 2.1 แผนภาพคลาสเชิงแนวคิด

จากภาพที่ 2.1 แผนภาพคลาสแสดงให้เห็นว่าระบบประกอบด้วยข้อมูลต่างๆ ได้แก่ Officer, Manager, Order, OrderItem และ Product โดยที่ Officer และ Manager เป็นประเภทหนึ่งของ Person ความสัมพันธ์ของคลาสแสดงความสัมพันธ์ของข้อมูล กล่าวคือ Officer เป็นผู้สร้างและยกเลิกรายการ Order ในขณะที่ Manager เป็นผู้อนุมัติและยกเลิกอนุมัติ รายการ Order ประกอบด้วย OrderItem ซึ่งเป็นการระบุ Product ต่างๆ รวมถึงจำนวน qty และ amount ของแต่ละ Product

2.1.2 เอกซ์เอ็มไอ

เอกซ์เอ็มไอ [3] เป็นรูปแบบของการแลกเปลี่ยนข้อมูลแบบจำลองยูเอ็มแอล (UML Model) โดยการจับเก็บแบบจำลองยูเอ็มแอลให้อยู่ในรูปแบบของเอกสารเอกซ์เอ็มแอล (XML : Extensible Markup Language) ปัจจุบันมีเครื่องมือสนับสนุนที่ใช้สำหรับแปลงแบบจำลองยูเอ็มแอลให้อยู่ในรูปแบบของเอกซ์เอ็มไอ เช่น โปรแกรมวิซวลพาราไดม์ (Visual Paradigm) โปรแกรมอีคลิปส์ยูเอ็มแอล (EclipseUML) และโปรแกรมอาร์โกยูเอ็มแอลเมื่อแบบจำลองยูเอ็มแอลถูกแปลง

ให้อยู่ในรูปแบบของเอกซ์เอ็มไอแล้วจะสามารถนำไปสกัดเพื่อให้ได้ข้อมูลของแบบจำลองยูเอ็มแอล จากนั้นจะสามารถนำข้อมูลที่ได้ไปทำการประมวลผลต่างๆ ตามต้องการ

2.1.3 การวัดค่าความคล้ายคลึงกันของสายอักขระ

ปัจจุบันมีวิธีที่ใช้ในการวัดค่าความคล้ายคลึงกันของสายอักขระ (String Similarity) หลายวิธี วิธีที่มักถูกนำไปใช้ในงานวิจัยต่างๆ คือ Longest Common Subsequence หรือ LCS ซึ่งสามารถอธิบายวิธีการด้วยการยกตัวอย่างประกอบ [8] ดังนี้

Subsequence ของสายอักขระหนึ่งๆ คือ ชุดของตัวอักขระที่อยู่ในสายอักขระนั้น เรียงลำดับจากซ้ายไปขวา

ตัวอย่าง

Subsequence ของ ACTTGCG เช่น ACT , ATTC , T , ACTTGC

แต่ TTA ไม่ใช่ Subsequence ของ ACTTGCG

Common Subsequence ของสายอักขระสองชุด คือ Subsequence ที่ปรากฏในทั้งสองสายอักขระ

Longest Common Subsequence คือ Common Subsequence ที่มีความยาวมากที่สุด

ตัวอย่าง

S1 = AAACCGTGAGTTATTCGTTCTAGAA

S2 = CACCCCTAAGGTACCTTTGGTTC

Longest Common Subsequence คือ ACCTAGTACTTTG

2.1.4 เวิร์ดเน็ต

เวิร์ดเน็ต [5] เป็นฐานข้อมูลคำศัพท์ภาษาอังกฤษที่ถูกพัฒนาโดย Princeton University เป็นการจัดกลุ่มคำศัพท์แยกตามชนิดของคำ คือ คำนาม (Noun) คำกริยา (Verb) คำคุณศัพท์ (Adjective) และคำวิเศษณ์ (Adverb) คำที่มีความหมายคล้ายกันเรียกว่า Synonym Set หรือ Synset โดยแต่ละ Synset ถูกเชื่อมโยงไปยัง Synset อื่นๆ ด้วยความสัมพันธ์เชิงความหมายและมีโครงสร้างแบบลำดับชั้น (Hierarchical Structure) ความสัมพันธ์ระหว่าง Synset มี 2 แบบ คือ ความสัมพันธ์แบบ Is-A หรือเรียกว่า Hypernym และ Hyponym เป็นความสัมพันธ์แบบ Super-Subordinate ตัวอย่างของความสัมพันธ์แบบนี้ ได้แก่ car กับ railcar และความสัมพันธ์แบบ Part-Of หรือเรียกว่า Holonym และ Meronym เป็นความสัมพันธ์แบบ Part-Whole ตัวอย่างของความสัมพันธ์แบบนี้ เช่น wheel กับ car

จากการจัดเก็บข้อมูลแบบมีลำดับชั้นและเชื่อมโยงกันด้วยความหมายจึงได้มีการพัฒนา WordNet::Similarity [9] ซึ่งเป็นการวัดความคล้ายคลึงกันเชิงความหมาย (Semantic Similarity) และความสัมพันธ์กัน (Relatedness) ระหว่างสอง Synset ออกมาเป็นตัวเลข WordNet::Similarity มีวิธีการวัดให้เลือกใช้หลายวิธี เช่น วิธี Wu & Palmer วิธี Hirst & St-Onge และวิธี Lin เป็นต้น แต่ละวิธีมีอัลกอริทึม (Algorithm) ที่ใช้ในการวัดที่แตกต่างกัน ตัวอย่างของการวัดความสัมพันธ์กันของ Synset ด้วยวิธี Wu & Palmer วิธี Hirst & St-Onge และวิธี Lin มีดังนี้

วิธี Wu & Palmer ซึ่งมีค่าความสัมพันธ์อยู่ระหว่าง 0 - 1

วัดค่าความสัมพันธ์กันของ customer และ employee ได้เท่ากับ 0.6667

วัดค่าความสัมพันธ์กันของ customer และ client ได้เท่ากับ 1

วิธี Hirst & St-Onge ซึ่งมีค่าความสัมพันธ์อยู่ระหว่าง 0 - 16

วัดค่าความสัมพันธ์กันของ customer และ employee ได้เท่ากับ 2

วัดค่าความสัมพันธ์กันของ customer และ client ได้เท่ากับ 16

วิธี Lin ซึ่งมีค่าความสัมพันธ์อยู่ระหว่าง 0 - 1

วัดค่าความสัมพันธ์กันของ customer และ employee ได้เท่ากับ 0.2613

วัดค่าความสัมพันธ์กันของ customer และ client ได้เท่ากับ 1

จากตัวอย่างนี้แสดงให้เห็นว่าวิธี Wu & Palmer วัดค่าความสัมพันธ์ของ customer และ employee ออกมาได้มากกว่าวิธี Hirst & St-Onge และวิธี Lin ส่วน customer และ client ทั้งสามวิธีเห็นว่าทั้งสองคำนี้เป็นคำที่มีความหมายเหมือนกัน

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยที่เกี่ยวข้องกับการเปรียบเทียบแผนภาพคลาส

1) งานวิจัยของ Udo Kelte และคณะ [10] ได้นำเสนออัลกอริทึมในการคำนวณหาค่าความแตกต่างของแบบจำลองยูเอ็มแอล เพื่อหาความแตกต่างระหว่างแบบจำลองยูเอ็มแอลในแต่ละเวอร์ชัน โดยได้ทำการทดสอบอัลกอริทึมด้วยแผนภาพคลาส ซึ่งแผนภาพคลาสที่นำมาเปรียบเทียบกันนั้นเป็นแผนภาพคลาสเดียวกันแต่ต่างเวอร์ชัน ในงานวิจัยนี้ได้ทำการแปลงแผนภาพคลาสให้อยู่ในรูปของเอกสารเอกซ์เอ็มไอ จากนั้นจึงทำการคำนวณหาความแตกต่างของเอกสารเอกซ์เอ็มไอด้วยอัลกอริทึมที่นำเสนอ และมีการใช้วิธี LCS ในการเปรียบเทียบความคล้ายคลึงกันของชื่อส่วนย่อยต่างๆ เช่น ชื่อคลาส ชื่อลักษณะประจำ ชื่อการดำเนินการ เป็นต้น

งานวิจัยนี้สามารถหาค่าความแตกต่างของแผนภาพคลาสออกมาเป็นตัวเลขได้ แต่ผลลัพธ์ที่ได้ไม่สามารถบอกได้ว่าแผนภาพคลาสทั้งสองมีข้อมูลหรือฟังก์ชันใดบ้างที่แตกต่างกัน ดังนั้นอัลกอริทึมที่นำเสนอในงานวิจัยนี้จึงไม่สามารถนำมาใช้ในการเปรียบเทียบแผนภาพคลาสเพื่อใช้ในการตรวจสอบความครบถ้วนของความต้องการได้

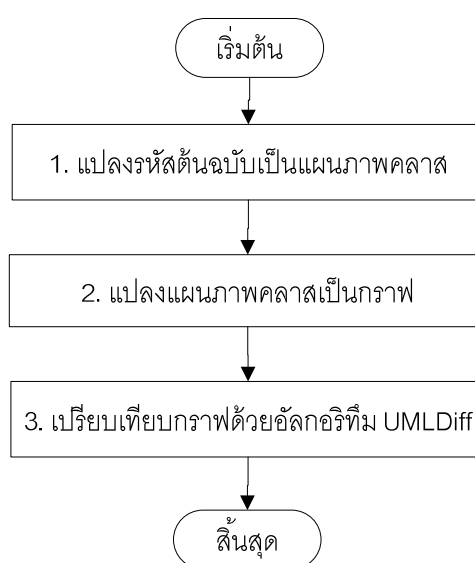
2) งานวิจัยของ Martin Girschick [11] ได้นำเสนอวิธีการในการหาความแตกต่างของแผนภาพคลาสและแสดงผลเป็นภาพ (Visual) เพื่อใช้ในการจัดเก็บเวอร์ชันของการสร้างแผนภาพคลาส วิธีการที่นำเสนอ คือ ทำการแปลงแผนภาพคลาสเป็นเอกสารเอกซ์เอ็มแอล จากนั้นทำการเปรียบเทียบเอกสารเอกซ์เอ็มแอลโดยใช้อัลกอริทึมที่นำเสนอ ซึ่งอัลกอริทึมนี้ได้แนวคิดมาจากวิธีที่ใช้ในการเปรียบเทียบโครงสร้างของข้อมูลโดยทั่วไป เช่น วิธีของงานวิจัย [12] แต่เพิ่มเติมกลวิธีบางอย่างเพื่อให้เหมาะสมกับการเปรียบเทียบโครงสร้างข้อมูลของแผนภาพคลาส

งานวิจัยดังกล่าวสามารถแสดงให้เห็นถึงความเปลี่ยนแปลงที่เกิดขึ้นกับแผนภาพคลาสหนึ่งจนทำให้ได้เป็นอีกแผนภาพคลาสหนึ่ง ซึ่งเป็นความแตกต่างระหว่างแผนภาพคลาสสองเวอร์ชันนั่นเอง ความเปลี่ยนแปลงที่สามารถตรวจหา (Detect) ได้ด้วยวิธีการที่นำเสนอในงานวิจัยนี้ได้แก่ การเพิ่ม ลบ เปลี่ยนชื่อ ย้ายส่วนย่อยต่างๆ อันประกอบด้วย คลาส ลักษณะประจำ การดำเนินการ นอกจากนี้ยังสามารถตรวจหาการสำเนาการดำเนินการจากคลาสแม่ไปยังคลาสลูก หรือเรียกว่า Method Overriding และสามารถตรวจหาความเปลี่ยนแปลงที่เกิดขึ้นกับลักษณะประจำ ได้แก่ การเปลี่ยนประเภทของข้อมูล (Data Type) และการเปลี่ยนสภาพการมองเห็น (Visibility) เป็นต้น แต่เนื่องจากในแผนภาพคลาสเชิงแนวคิดไม่มีการทำ Method Overriding ดังนั้นวิธีการที่ Martin Girschick นำเสนอในงานวิจัยนี้จึงมีความซับซ้อนเกินไปสำหรับการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด

3) งานวิจัยของ Ludovic Auxepeales และคณะ [13] ได้นำเสนอวิธีการเปรียบเทียบแผนภาพคลาสสำหรับพัฒนาเป็นเครื่องมือที่ใช้ในการเรียนการสอน เครื่องมือดังกล่าวสามารถวิเคราะห์ความแตกต่างระหว่างแผนภาพคลาสของนักเรียนกับแผนภาพคลาสของผู้เชี่ยวชาญได้ วิธีการที่ใช้ในการเปรียบเทียบโครงสร้างของแผนภาพคลาส คือ การจับคู่กราฟ (Graph Matching) ซึ่ง Sorlin และคณะได้นำเสนอไว้ใน [14] ส่วนการเปรียบเทียบชื่อ เช่น ชื่อคลาส ชื่อลักษณะประจำ ชื่อการดำเนินการ ได้ใช้วิธีการของ Fausto Giunchiglia และคณะ [15] ร่วมกับวิธีการของ Hong-Hai Do และคณะ [16] โดยวิธีการของ Giunchiglia และคณะนั้นได้มีการใช้เวิร์ดเน็ตสำหรับการเปรียบเทียบคำเชิงความหมายด้วย

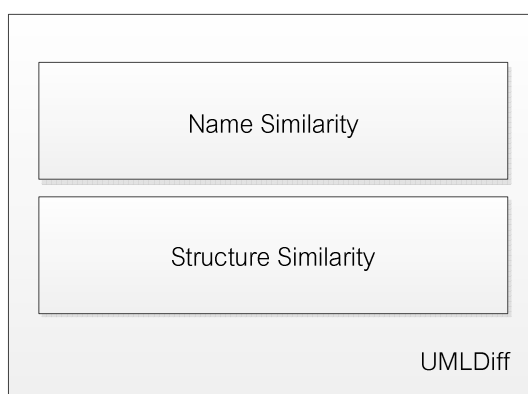
งานวิจัยดังกล่าวสามารถหาความแตกต่างระหว่างแผนภาพคลาสได้ อีกทั้งยังสามารถเปรียบเทียบคำเชิงความหมายได้อีกด้วย แต่จากการทดลองในงานวิจัยนี้ Ludovic Auxepeales และคณะ ได้สรุปไว้ว่าวิธีการที่นำเสนอเหมาะสมสำหรับการเปรียบเทียบแผนภาพคลาสที่ไม่ซับซ้อน ดังนั้นวิธีการนี้จึงไม่เหมาะนักที่จะนำมาใช้ในการทำงานจริง เนื่องจากแผนภาพคลาสที่ได้จากการวิเคราะห์ความต้องการของระบบงานซอฟต์แวร์นั้นมีความซับซ้อนมากกว่าแผนภาพคลาสที่สร้างจากความต้องการของโจทย์ที่ใช้ในการเรียนการสอน

4) งานวิจัยของ Zhenchang Xing [4] ได้นำเสนออัลกอริทึมชื่อว่า UMLDiff สำหรับการเปรียบเทียบแผนภาพคลาส โดยเป็นการหาความแตกต่างของแผนภาพคลาสที่ถูกแปลงจากรหัสต้นฉบับ (Source Code) ของระบบงานเดียวกันแต่ต่างเวอร์ชัน เพื่อให้เห็นวิวัฒนาการในการพัฒนาซอฟต์แวร์ ขั้นตอนการเปรียบเทียบแผนภาพคลาสที่ Xing นำเสนอแสดงดังภาพที่ 2.2



ภาพที่ 2.2 ขั้นตอนการเปรียบเทียบแผนภาพคลาสที่ Xing นำเสนอ

จากภาพที่ 2.2 วิธีการเปรียบเทียบแผนภาพคลาสเริ่มจากการสร้างแผนภาพคลาสจากรหัสต้นฉบับของโปรแกรมสองเวอร์ชัน จากนั้นจึงนำแผนภาพคลาสที่ได้มาเปรียบเทียบกันโดยแปลงเป็นกราฟ ก่อนที่จะทำการเปรียบเทียบกราฟด้วยอัลกอริทึม UMLDiff ในส่วนของอัลกอริทึม UMLDiff จะมีการเปรียบเทียบสองส่วน ได้แก่ ความคล้ายคลึงกันของชื่อคลาส ชื่อลักษณะประจำชื่อการดำเนินการ หรือเรียกว่า Name Similarity และความคล้ายคลึงกันของโครงสร้าง หรือ Structure Similarity ส่วนประกอบของอัลกอริทึม UMLDiff แสดงดังภาพที่ 2.3



ภาพที่ 2.3 ส่วนประกอบของอัลกอริทึม UMLDiff

ในส่วนของการเปรียบเทียบความคล้ายคลึงกันของชื่อนั้น UMLDiff ใช้วิธี LCS โดยได้เพิ่มเติมอัลกอริทึมบางอย่างเพื่อปรับปรุงให้ LCS มีประสิทธิภาพมากขึ้น

ผลลัพธ์ที่ได้จากการเปรียบเทียบกราฟโดยใช้อัลกอริทึม UMLDiff ทำให้ทราบความแตกต่างประเภทต่างๆ ของแผนภาพคลาส ได้แก่ การเพิ่ม ลบ เปลี่ยนชื่อ ย้ายส่วนย่อย เป็นต้น อีกทั้งยังเป็นอัลกอริทึมที่พัฒนาขึ้นมาเพื่อใช้ในการตรวจหาความเปลี่ยนแปลงที่เกิดขึ้นในแผนภาพคลาสโดยเฉพาะ

2.2.2 งานวิจัยที่เกี่ยวข้องกับการใช้เวิร์ดเน็ต

1) Yasser Kotb [17] ได้นำเสนอวิธีการในการตรวจสอบความสอดคล้องกันของการออกแบบซอฟต์แวร์ที่ใช้แผนภาพยูเอ็มแอลในการออกแบบ โดยการตรวจสอบความสอดคล้องกันของยูสเคส แผนภาพคลาส แผนภาพลำดับการทำงาน ที่เป็นการออกแบบระบบงานเดียวกัน วิธีการที่นำเสนอเริ่มจากการแปลงแผนภาพยูเอ็มแอลทั้งสามประเภทเป็นเอกสารเอกซ์เอ็มไอ จากนั้นจึงสกัดคำศัพท์ออกมาเพื่อทำการเปรียบเทียบกันว่ามีความคล้ายคลึงกันหรือไม่ โดยวัดความคล้ายคลึงกันทั้งเชิงวากยสัมพันธ์ (Syntactic) และเชิงความหมาย (Semantic) ในงานวิจัยนี้ได้ยกตัวอย่างการหาความสอดคล้องกันของแผนภาพคลาสและแผนภาพลำดับการทำงาน โดยการแปลงแผนภาพเป็นเอกสารเอกซ์เอ็มไอ จากนั้นจึงสกัดชื่อคลาสเพื่อนำมาเปรียบเทียบกันโดยใช้ TSS (Text Semantic Similarity) [18] ซึ่ง TSS ได้วัดความคล้ายคลึงกันของคำเชิงความหมายโดยใช้เวิร์ดเน็ต

จากการนำเวิร์ดเน็ตมาใช้ในการเปรียบเทียบชื่อคลาสทำให้สามารถวัดความคล้ายกันเชิงความหมายของคำได้ ดังนั้นจึงสามารถตรวจสอบได้ว่าคลาสต่างๆ ที่มีในแผนภาพยูเอ็มแอลหนึ่งสอดคล้องกับคลาสที่มีในแผนภาพยูเอ็มแอลอื่นๆ หรือไม่

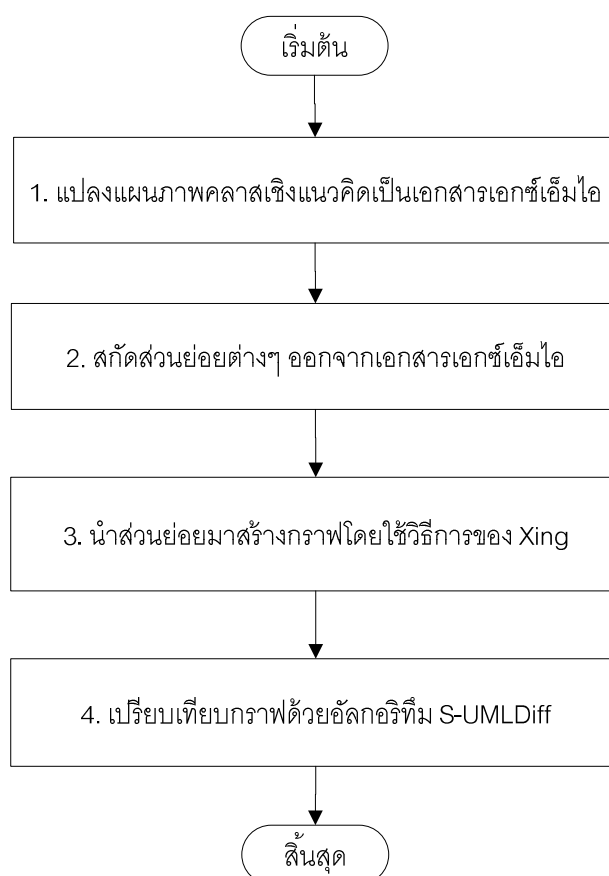
บทที่ 3

การออกแบบอัลกอริทึม S-UMLDiff

งานวิจัยนี้นำเสนอวิธีการในการตรวจสอบความครบถ้วนของความต้องการสำหรับการพัฒนาซอฟต์แวร์ที่เป็นการพัฒนาระบบใหม่บนพื้นฐานของข้อกำหนดของระบบเดิม เพื่อให้ นักวิเคราะห์ระบบและผู้ให้ความต้องการซอฟต์แวร์ได้ทำการตรวจสอบความครบถ้วนของความต้องการของระบบใหม่ว่ามีความครบถ้วนและถูกต้องหรือไม่ วิธีการที่นำเสนอ คือ การนำ แผนภาพคลาสเชิงแนวคิดของระบบเดิมที่มีอยู่แล้วและแผนภาพคลาสเชิงแนวคิดของระบบใหม่ที่ได้จากการวิเคราะห์ความต้องการมาเปรียบเทียบกัน เพื่อหาความแตกต่างระหว่างแผนภาพทั้งสอง ทั้งนี้ผู้วิจัยได้ทำการพัฒนาอัลกอริทึม S-UMLDiff หรือ Semantic-UMLDiff เพื่อใช้ในการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดโดยพิจารณาถึงความหมายของคำในแผนภาพ อัลกอริทึม S-UMLDiff เป็นอัลกอริทึมที่ปรับมาจากรหัสต้นฉบับของอัลกอริทึม UMLDiff ซึ่ง Xing ได้พัฒนาไว้ด้วยภาษาจาวา [19] แต่เนื่องจาก UMLDiff เป็นอัลกอริทึมที่ใช้ในการเปรียบเทียบแผนภาพคลาสที่เป็นแบบจำลองเชิงตรรกะ (Logical Model) เมื่อนำมาปรับใช้ในการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดจึงจำเป็นต้องปรับวิธีการบางอย่าง นอกจากนี้ผู้วิจัยยังทำการเพิ่มประสิทธิภาพของอัลกอริทึมเพื่อให้เหมาะสมสำหรับการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด ทั้งในส่วนของ การเปรียบเทียบคำเชิงความหมายและการตรวจสอบการเปลี่ยนประเภทของส่วนย่อยอีกด้วย

3.1 ขั้นตอนการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด

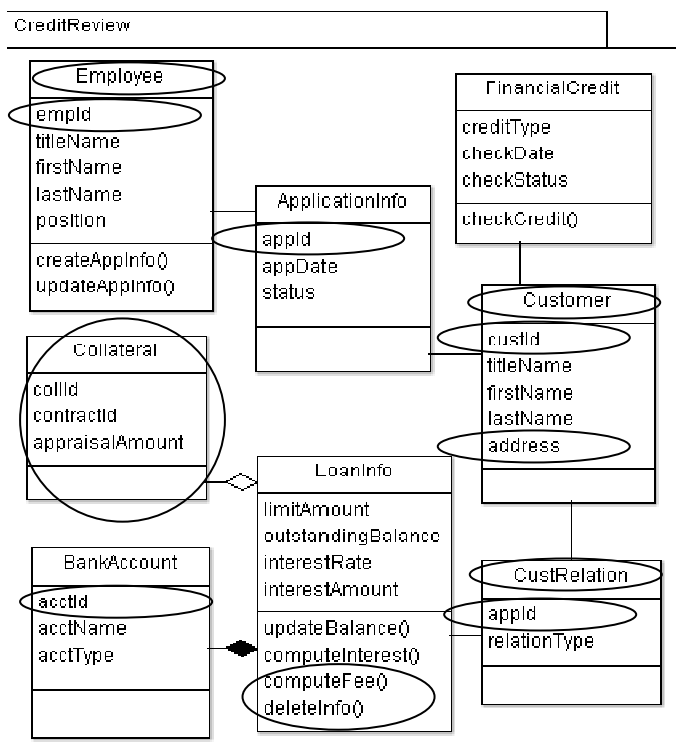
ขั้นตอนการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดเพื่อตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์ที่นำเสนอในงานวิจัยนี้แสดงดังภาพที่ 3.1



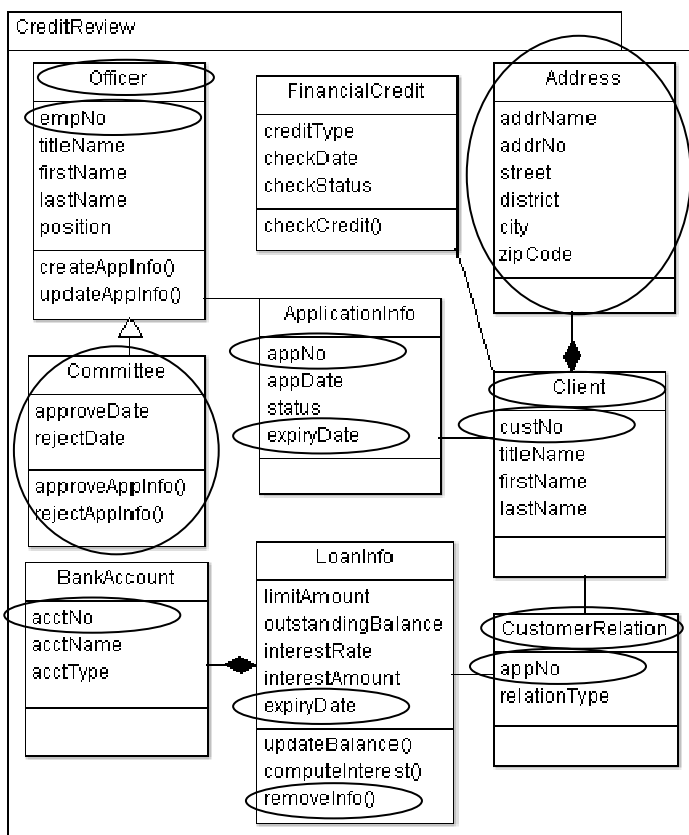
ภาพที่ 3.1 ขั้นตอนการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด

ผู้วิจัยขออธิบายการทำงานแต่ละขั้นตอนโดยยกตัวอย่างบางส่วนของแผนภาพคลาสเชิงแนวคิดของระบบงานจริงของธนาคารแห่งหนึ่งในประเทศไทย มาใช้ในการนำเสนอ ภาพที่ 3.2 เป็นแผนภาพคลาสเชิงแนวคิดของระบบซอฟต์แวร์เดิม และภาพที่ 3.3 เป็นแผนภาพคลาสเชิงแนวคิดของระบบซอฟต์แวร์ที่จะพัฒนาใหม่ ขั้นตอนการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดสามารถอธิบายได้ดังนี้

- 1) นำแผนภาพคลาสเชิงแนวคิดของระบบเดิมและระบบใหม่มาแปลงเป็นเอกสารเอกซ์เอ็มไอโดยใช้เครื่องมือสนับสนุน คือ โปรแกรมอาร์โกยูเอ็มแอล
- 2) สกัดส่วนย่อยต่าง ๆ ของแผนภาพคลาสเชิงแนวคิดจากเอกสารเอกซ์เอ็มไอ ได้แก่ ชื่อแพ็คเกจ ชื่อคลาส ชื่อลักษณะประจำ ชื่อการดำเนินการ เป็นต้น



ภาพที่ 3.2 แผนภาพคลาสเชิงแนวคิดของระบบเดิม



ภาพที่ 3.3 แผนภาพคลาสเชิงแนวคิดของระบบใหม่

3) นำส่วนย่อยต่าง ๆ ที่สกัดได้มาสร้างข้อมูลกราฟด้วยวิธีการที่นำเสนอใน [4] คือ กำหนดให้ชื่อส่วนย่อยต่าง ๆ เป็นจุดต่อ (Node) หรือเอนทิตี (Entity) และกำหนดความสัมพันธ์ระหว่างส่วนย่อยเป็นเส้นเชื่อม (Edge) หรือความสัมพันธ์ (Relation) ตัวอย่างของข้อมูลเอนทิตีที่ได้จากแผนภาพคลาสเชิงแนวคิดในภาพที่ 3.2 แสดงดังตารางที่ 3.1 (ซึ่ง Virtual Root เป็นเอนทิตีที่จะถูกกำหนดให้กับทุกแผนภาพคลาสเชิงแนวคิดเสมอ) และเนื่องจาก UMLDiff เป็นอัลกอริทึมที่ใช้ในการเปรียบเทียบแผนภาพคลาสเชิงตรรกะ ดังนั้นเมื่อนำมาประยุกต์ใช้ในการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดจึงจำเป็นต้องกำหนดประเภทความสัมพันธ์ของคลาสตามประเภทความสัมพันธ์ของเอนทิตีที่มีกำหนดไว้ใน UMLDiff ดังตารางที่ 3.2 นอกจากนี้ยังมีความสัมพันธ์อื่น ๆ ที่ใช้แสดงถึงความสัมพันธ์ระหว่างเอนทิตี ได้แก่ Contain, Field Data Type และ Method Return Type ซึ่งมีตัวอย่างของข้อมูลความสัมพันธ์ที่ได้จากแผนภาพคลาสเชิงแนวคิดในภาพที่ 3.3 แสดงไว้ดังตารางที่ 3.3 แต่เนื่องจาก UMLDiff จำเป็นต้องระบุประเภทข้อมูลของลักษณะประจำ และประเภทข้อมูลที่ถูกส่งกลับจากการดำเนินการ (Method Return Type) แต่ในแผนภาพคลาสเชิงแนวคิดไม่มีข้อมูลเหล่านี้ ดังนั้นในงานวิจัยนี้จึงกำหนดให้ทุกลักษณะประจำมีประเภทข้อมูลเป็น String และทุกการดำเนินการมีประเภทข้อมูลที่ถูกส่งกลับเป็น void

4) นำข้อมูลกราฟที่ได้มาทำการเปรียบเทียบกันโดยใช้อัลกอริทึม S-UMLDiff ซึ่งปรับปรุงจากอัลกอริทึม UMLDiff ทั้งในส่วนของการวัดความคล้ายกันของชื่อ (Name Similarity) และการวัดความคล้ายคลึงกันของโครงสร้าง (Structural Similarity) ผลลัพธ์ที่ได้จากการเปรียบเทียบจะทำให้ทราบว่าเอนทิตีมีการเปลี่ยนแปลงอย่างไรบ้าง ประเภทของความเปลี่ยนแปลงที่อัลกอริทึม S-UMLDiff รายงานผลแสดงดังตารางที่ 3.4

ตารางที่ 3.1 ตัวอย่างเอนทิตีที่ได้จากแผนภาพระบบเดิมภาพที่ 3.2

ประเภท	ชื่อเอนทิตี
Virtual Root	Virtual Root
Package	CreditReview
Class	FinancialCredit
Attribute	creditType, checkDate, checkStatus
Method	checkCredit()

ตารางที่ 3.2 การกำหนดความสัมพันธ์ระหว่างคลาส

ความสัมพันธ์ระหว่างคลาส ในแผนภาพคลาสเชิงแนวคิด	ความสัมพันธ์ ที่ใช้ในอัลกอริทึม UMLDiff
Extend	Extend
Association	Use
Aggregation	Implement
Composition	Implement

ตารางที่ 3.3 ตัวอย่างความสัมพันธ์ที่ได้จากแผนภาพระบบใหม่ภาพที่ 3.3

ประเภท	เอนทิตีที่เกี่ยวข้อง	
Contain	Virtual Root	CreditReview
	CreditReview	ApplicationInfo
	ApplicationInfo	appNo
	ApplicationInfo	create()
Extend	Committee	Officer
Implement	LoanInfo	BankAccount
Use	LoanInfo	CustomerRelation
Field Data Type	appNo	String
Method Return Type	create()	void

ตารางที่ 3.4 ประเภทของความเปลี่ยนแปลงที่อัลกอริทึม S-UMLDiff รายงานผล

ประเภท	คำอธิบาย
match	เอนทิตีของระบบเดิมมีชื่อตรงกับเอนทิตีของระบบใหม่
rename	เอนทิตีของระบบเดิมมีชื่อคล้ายกับเอนทิตีของระบบใหม่
smatch	เอนทิตีของระบบเดิมมีชื่อที่มีความหมายคล้ายกับเอนทิตีของระบบใหม่
move	เอนทิตีของระบบเดิมมีชื่อคล้ายกับเอนทิตีของระบบใหม่ แต่มี parent ต่างกัน

ตารางที่ 3.4 ประเภทของความเปลี่ยนแปลงที่อัลกอริทึม S-UMLDiff รายงานผล (ต่อ)

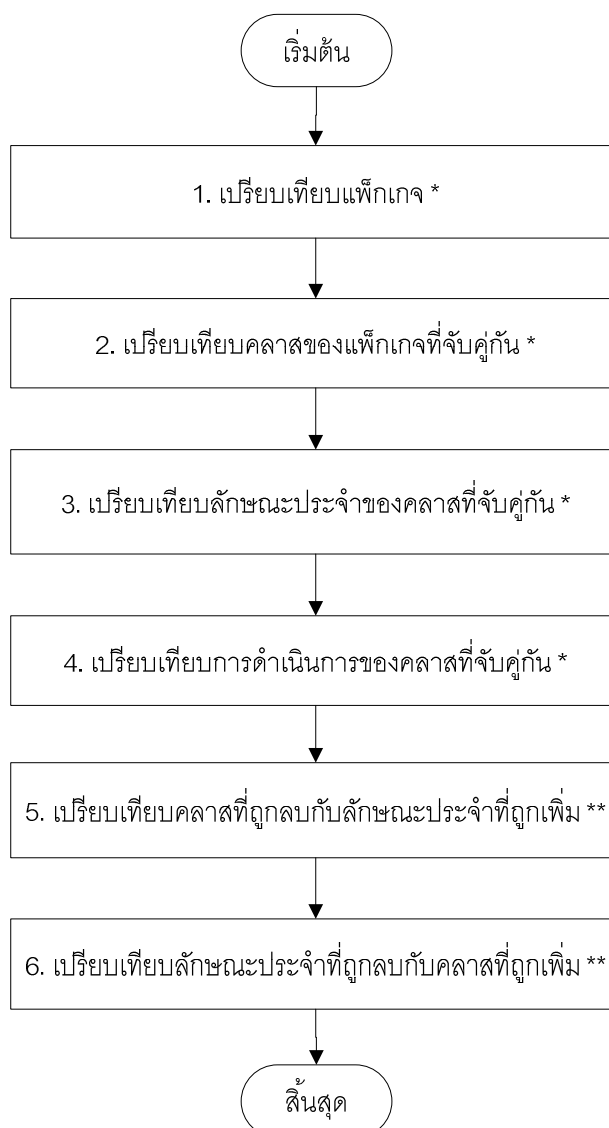
ประเภท	คำอธิบาย
add	มีเอนทิตีนี้ในระบบใหม่แต่ไม่มีในระบบเดิม
remove	มีเอนทิตีนี้ในระบบเดิมแต่ไม่มีในระบบใหม่
changetype	เอนทิตีของระบบเดิมมีชื่อเหมือนกันหรือคล้ายกันกับเอนทิตีของระบบใหม่ แต่มีประเภทของส่วนย่อยต่างกัน

3.2 การทำงานของอัลกอริทึม S-UMLDiff

หลังจากที่นำส่วนย่อยต่างๆ ของแผนภาพคลาสเชิงแนวคิด ได้แก่ แพ็กเกจ คลาส ลักษณะประจำ และการดำเนินการมาสร้างกราฟตามวิธีการของ Xing แล้ว จากนั้นจะทำการเปรียบเทียบกราฟด้วยอัลกอริทึม S-UMLDiff ซึ่งปรับปรุงมาจากอัลกอริทึม UMLDiff การทำงานของอัลกอริทึม S-UMLDiff แสดงดังภาพที่ 3.4

* คือ ขั้นตอนที่มีอยู่แล้วใน UMLDiff แต่ผู้วิจัยทำการปรับปรุงในส่วนของการเปรียบเทียบชื่อเอนทิตี ให้สามารถเปรียบเทียบความคล้ายกันเชิงความหมายได้โดยใช้เวรด์เน็ต

** คือ ขั้นตอนที่ผู้วิจัยได้เพิ่มเติมเข้ามาใน S-UMLDiff เพื่อเป็นการตรวจสอบการเปลี่ยนแปลงประเภทของส่วนย่อย



ภาพที่ 3.4 ขั้นตอนการทำงานของอัลกอริทึม S-UMLDiff

จากภาพที่ 3.4 การเปรียบเทียบเอนทิตีในขั้นตอนที่ 1-4 อัลกอริทึม S-UMLDiff จะทำการตรวจสอบเอนทิตี 3 แบบ เช่นเดียวกันกับอัลกอริทึม UMLDiff ได้แก่

- 1) Identify Match คือ การตรวจสอบเพื่อหาเอนทิตีที่ชื่อตรงกัน
- 2) Identify Rename คือ การตรวจสอบเพื่อหาว่าเอนทิตีถูกเปลี่ยนชื่อหรือไม่
- 3) Identify Move คือ การตรวจสอบเพื่อหาว่าเอนทิตีถูกย้ายหรือไม่

การเปรียบเทียบเอนทิตีในแต่ละขั้นตอนจะเริ่มจากการตรวจสอบว่าเอนทิตีมีชื่อตรงกันหรือไม่ (Identify Match) จากนั้นนำเอนทิตีที่เหลือมาทำการตรวจสอบว่าเป็นการเปลี่ยนชื่อหรือไม่ (Identify Rename) โดยพิจารณาความคล้ายกันโดยรวม (Overall Similarity) ของเอนทิตี ได้แก่ การเปรียบเทียบความคล้ายกันของรูปคำโดยใช้วิธี LCS และการเปรียบเทียบความคล้ายกันของ

โครงสร้างตามวิธีการของ UMLDiff นอกจากนี้ผู้วิจัยได้เพิ่มการพิจารณาความคล้ายกันเชิงความหมายของชื่อโดยใช้เวรด์เน็ตด้วย หลังจากตรวจสอบการเปลี่ยนชื่อแล้วจะนำเอนทิตีที่เหลือมาทำการตรวจสอบว่าถูกย้ายหรือไม่ (Identify Move) โดยตรวจสอบว่าเอนทิตีนั้นมีการเปลี่ยน parent หรือไม่

3.3 ค่าขีดแบ่งของความคล้ายคลึง (Similarity Threshold)

ค่าขีดแบ่งของความคล้ายคลึง คือ ค่าที่อัลกอริทึม S-UMLDiff นำมาใช้ในการพิจารณาความคล้ายคลึงกันของเอนทิตี ยกตัวอย่างเช่น ถ้าในขั้นตอนของการพิจารณาความคล้ายคลึงเชิงความหมายของชื่อเอนทิตีมีการกำหนดค่าขีดแบ่งเท่ากับ 0.7 กรณีที่ค่าความคล้ายคลึงกันของชื่อเอนทิตีที่ได้จากเวรด์เน็ตมีค่าน้อยกว่า 0.7 จะทำให้ S-UMLDiff ตัดสินว่าเอนทิตีทั้งสองนี้ไม่ใช่เอนทิตีเดียวกันในเชิงความหมาย ในทางกลับกันถ้าค่าความคล้ายคลึงกันของชื่อเอนทิตีที่เวรด์เน็ตคำนวณได้มีค่ามากกว่าหรือเท่ากับ 0.7 จะทำให้ S-UMLDiff ตัดสินว่าชื่อเอนทิตีทั้งสองเป็นชื่อเดียวกันในเชิงความหมาย

ในงานวิจัยนี้ได้มีการกำหนดค่าขีดแบ่ง 3 ค่า ได้แก่ ค่าขีดแบ่งของการเปลี่ยนชื่อ (Rename Threshold) ค่าขีดแบ่งของการวัดความคล้ายคลึงเชิงความหมายของคำ (Semantic Word Similarity Threshold) และค่าขีดแบ่งของการวัดความคล้ายคลึงเชิงความหมายของวลี (Semantic Phrase Similarity Threshold) ผู้วิจัยได้ทำการทดสอบเพื่อหาค่าขีดแบ่งที่เหมาะสมโดยการกำหนดค่าขีดแบ่งเพื่อใช้ในการวัดคำศัพท์ต่างๆ ที่มักพบในแผนภาพคลาสเชิงแนวคิดของระบบงานซอฟต์แวร์และทดลองปรับ (Tuning) กับการทดสอบกับชุดของแผนภาพคลาสเชิงแนวคิดมากกว่า 10 ชุดเมื่อได้ค่าที่เหมาะสมแล้วจึงนำไปใช้ในการทดสอบกับระบบงานจริง อย่างไรก็ตามในการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดโดยใช้เครื่องมือที่นำเสนอในงานวิจัยนี้นักวิเคราะห์ระบบสามารถเปลี่ยนแปลงค่าขีดแบ่งได้ตามต้องการหากเห็นว่าผลลัพธ์ที่ได้ไม่เหมาะสม

3.4 การพิจารณาความคล้ายกันของชื่อเอนทิตี

อัลกอริทึม UMLDiff ใช้วิธี LCS ในการเปรียบเทียบความคล้ายกันของชื่อเอนทิตี แต่ในการออกแบบแผนภาพคลาสเชิงแนวคิดของระบบเดิมและระบบใหม่อาจมีการใช้คำที่เขียนไม่เหมือนกันแต่มีความหมายคล้ายกัน เช่น Customer ในภาพที่ 3.2 และ Client ในภาพที่ 3.3 ซึ่ง LCS ไม่สามารถหาความคล้ายกันเชิงความหมายได้ ในงานวิจัยนี้จึงได้นำ WordNet::Similarity

[9] มาใช้ในการเปรียบเทียบค่าเชิงความหมายด้วย โดยเลือกใช้วิธี Wu & Palmer แม้ว่าวิธีนี้จะสามารถวัดความคล้ายคลึงเชิงความหมายได้เฉพาะคำนาม (Noun) และคำกริยา (Verb) เท่านั้น แต่จากการทดสอบพบว่าวิธี Wu & Palmer เป็นวิธีที่ให้ผลลัพธ์ที่เหมาะสมที่สุดเมื่อเทียบกับผลลัพธ์ที่ได้จากวิธีต่าง ๆ อีกทั้งชื่อในแผนภาพคลาสเชิงแนวคิดมักจะถูกประกอบด้วยคำนามและคำกริยาเป็นส่วนใหญ่ โดย S-UMLDiff จะทำการเปรียบเทียบความคล้ายคลึงเชิงความหมายโดยใช้ WordNet::Similarity ก่อน หากคะแนนที่ได้น้อยกว่าค่าขีดแบ่งของความคล้ายคลึงที่กำหนดไว้ จึงจะทำการเปรียบเทียบความคล้ายกันของชื่อโดยใช้ LCS อย่างไรก็ตาม เนื่องจากชื่อเอนทิตี มักจะถูกประกอบด้วยคำมากกว่าหนึ่งคำหรือเป็นวลี (Phrase) ซึ่งเว็บริดเน็ตไม่สามารถหาความหมายของวลีได้ ในงานวิจัยนี้จึงได้นำวิธีการหาความคล้ายกันเชิงความหมายของวลีจากงานวิจัย [20] มาใช้ในการเปรียบเทียบชื่อเอนทิตีต่าง ๆ ดังนั้นหากต้องการหาความคล้ายกันเชิงความหมายระหว่างวลี a ซึ่งมี m คำ กับวลี b ซึ่งมี n คำ จะสามารถหาได้ดังสมการ (1) และ (2)

$$pSim(a,b) = \frac{\sum_{s=1}^m wpSim(a_s,b)}{m} \quad (1)$$

$$wpSim(a_s,b) = \max(wSim(a_s,b_1), \dots, wSim(a_s,b_n)) \quad (2)$$

โดยที่ $wSim(a_s, b_n)$ หมายถึง ค่าความคล้ายกันเชิงความหมายระหว่างคำที่ s ในวลี a กับคำที่ n ในวลี b ตามวิธี Wu & Palmer ซึ่งค่าจะอยู่ในช่วง $[0, 1]$

การที่สองวลีจะถือว่าเหมือนกันได้นั้น จะต้องมีค่า $pSim$ ไม่น้อยกว่าค่าขีดแบ่งของการวัดความคล้ายคลึงเชิงความหมายของวลีที่นักวิเคราะห์ระบบกำหนดในการเปรียบเทียบแผนภาพตัวอย่างเช่น ในการหาความคล้ายกันเชิงความหมายของชื่อการดำเนินการ deleteInfo ในภาพที่ 3.2 (วลี a) กับ removeInfo ในภาพที่ 3.3 (วลี b) โดยใช้สมการ (1) และ (2) จะได้ว่า

$$\begin{aligned} & pSim(deleteInfo, removeInfo) \\ &= (wpSim(delete, removeInfo) \\ & \quad + wpSim(Info, removeInfo))/2 \\ &= (\max(wSim(delete, remove), wSim(delete, Info)) + \\ & \quad \max(wSim(Info, remove), wSim(Info, Info)))/2 \\ &= (\max(0.8, 0) + \max(0.4, 1))/2 = (0.8+1)/2 = 0.9 \end{aligned}$$

หากในการเปรียบเทียบ นักวิเคราะห์ระบบกำหนดค่าขีดแบ่งของการวัดความคล้ายคลึงเชิงความหมายของวลีไว้เป็น 0.9 จะทำให้ S-UMLDiff สรุปว่าชื่อการดำเนินการ deleteInfo และ removeInfo เป็นชื่อเดียวกันในเชิงความหมายและจะไม่ทำการเปรียบเทียบโดยใช้ LCS แต่หากนักวิเคราะห์ระบบกำหนดค่าขีดแบ่งของการวัดความคล้ายคลึงเชิงความหมายของวลีไว้เป็น 0.95 จะทำให้ S-UMLDiff สรุปว่าชื่อการดำเนินการทั้งสองไม่ใช่ชื่อเดียวกันในเชิงความหมาย และจะทำการเปรียบเทียบชื่อโดยใช้ LCS ต่อไป

นอกจากนี้ในการเปรียบเทียบเพื่อหาค่าความคล้ายกันเชิงความหมายของชื่อเอนทิตี อัลกอริทึม S-UMLDiff ได้มีการพิจารณาเพิ่มเติม ดังต่อไปนี้

1) ในการหาค่าความคล้ายกันเชิงความหมายของคำแต่ละคำโดยใช้ WordNet::Similarity จำเป็นต้องระบุชนิดของคำ (Part of Speech) ที่ต้องการหาค่าความหมายด้วย เช่น คำนามและคำกริยาแต่คำหนึ่งคำอาจเป็นได้หลายชนิดเช่น name และ type เป็นทั้งคำนามและคำกริยาในการหาค่าความคล้ายกันเชิงความหมายของคำทั้งสอง S-UMLDiff จะทำการหาค่าเฉลี่ยของความคล้ายกันในกรณีที่เป็นคำนามและความคล้ายกันในกรณีที่เป็นคำกริยา ดังสมการที่ (3)

$$wSim(name, type) = (wSim(name, type)_{noun} + wSim(name, type)_{verb})/2 \quad (3)$$

2) ในการเปรียบเทียบคำแรกๆของชื่อการดำเนินการ แม้ว่าคำนั้นจะเป็นมากกว่า 1 ชนิด เช่น reviewInfo คำว่า review เป็นทั้งคำนามและคำกริยา แต่ S-UMLDiff จะถือว่าเป็นคำกริยาเท่านั้น เพราะคำแรกๆของชื่อการดำเนินการเป็นคำกริยาเสมอ

3.5 การพิจารณาความคล้ายกันของโครงสร้าง

S-UMLDiff ใช้แนวทางของ UMLDiff ในการเปรียบเทียบความคล้ายกันในเชิงโครงสร้าง โดยการจับคู่เอนทิตีประเภทเดียวกันและมีชื่อเดียวกันหรือคล้ายกัน เพื่อเปรียบเทียบรายละเอียดภายใน ดังตารางที่ 3.5

ตารางที่ 3.5 การเปรียบเทียบเชิงโครงสร้าง

ประเภท	การเปรียบเทียบ
แพ็กเกจ	คลาสที่อยู่ภายใน
คลาส	ลักษณะประจำและการดำเนินการที่อยู่ภายใน รวมทั้งคลาสอื่นที่คลาสนี้มีความสัมพันธ์ด้วย

อย่างไรก็ตาม ในการออกแบบแผนภาพคลาสเชิงแนวคิดของระบบใหม่ อาจมีเอนทิตีที่ถูกเปลี่ยนประเภทไป เช่น ลักษณะประจำ address ในภาพที่ 3.2 ถูกเปลี่ยนเป็นคลาส Address ในภาพที่ 3.3 ดังนั้นหากต้องการทราบถึงข้อมูลที่ขาดหายไปหรือเพิ่มเข้ามาในระบบใหม่ จึงจำเป็นต้องทำการเปรียบเทียบชื่อลักษณะประจำกับชื่อคลาสด้วย ซึ่งอัลกอริทึม UMLDiff ไม่สามารถตรวจสอบได้ ในงานวิจัยนี้จึงได้เพิ่มเติมความสามารถในการตรวจสอบการเปลี่ยนแปลงประเภทของส่วนย่อย ได้แก่ การเปลี่ยนจากลักษณะประจำเป็นคลาส รวมถึงการเปลี่ยนจากคลาสเป็นลักษณะประจำ

3.6 การพิจารณาความคล้ายกันโดยรวม

ในขั้นตอนของการตรวจสอบว่าเอนทิตีมีการเปลี่ยนชื่อหรือไม่ (Identify Rename) S-UMLDiff จะพิจารณาความคล้ายกันโดยรวมโดยใช้แนวทางเดียวกันกับ UMLDiff เอนทิตีแต่ละประเภทจะมีการพิจารณาความคล้ายกันโดยรวมที่แตกต่างกัน ดังนี้

กำหนดให้

MatchPoint คือ คะแนนความคล้ายกันโดยรวมของเอนทิตี x และเอนทิตี y

NamePoint คือ คะแนนความคล้ายกันของชื่อเอนทิตี x และชื่อเอนทิตี y

x คือ เอนทิตีของระบบเดิม

y คือ เอนทิตีของระบบใหม่

- 1) การพิจารณาความคล้ายกันโดยรวมของแพ็กเกจ

$$\text{MatchPoint} = (\text{NamePoint} + \text{ChildrenMatchCount}) * 1.0 / (\text{NamePoint} + (\text{xChildrenCount} + \text{yChildrenCount} - \text{ChildrenMatchCount}))$$

โดยที่

ChildrenMatchCount คือ จำนวนคลาสที่อยู่ในแพ็กเกจ x ที่มีชื่อตรงกับคลาสที่อยู่ในแพ็กเกจ y

xChildrenCount คือ จำนวนคลาสที่อยู่ในแพ็กเกจ x

yChildrenCount คือ จำนวนคลาสที่อยู่ในแพ็กเกจ y

- 2) การพิจารณาความคล้ายกันโดยรวมของคลาส

$$\text{MatchPoint} = (\text{NamePoint} + \text{ChildrenPoint} + \text{UsagePoint}) / (\text{NamePoint} + 3)$$

$$\text{ChildrenPoint} = \text{ChildrenMatchCount} * 1.0 / (\text{xChildrenCount} + \text{yChildrenCount} - \text{ChildrenMatchCount})$$

โดยที่

$\text{ChildrenMatchCount}$ คือ จำนวนลักษณะประจำของคลาส x ที่มีชื่อตรงกันกับลักษณะประจำของคลาส y รวมกับจำนวนการดำเนินการของคลาส x ที่มีชื่อตรงกันกับการดำเนินการของคลาส y

xChildrenCount คือ จำนวนลักษณะประจำและการดำเนินการของคลาส x

yChildrenCount คือ จำนวนลักษณะประจำและการดำเนินการของคลาส y

UsagePoint คือ คะแนนความคล้ายกันของชื่อคลาสที่เกี่ยวข้องกับคลาส x และชื่อคลาสที่เกี่ยวข้องกับคลาส y

3) การพิจารณาความคล้ายกันโดยรวมของลักษณะประจำ

$$\text{MatchPoint} = (\text{ParentPoint} * \text{NamePoint} + \text{UsagePoint}) / (\text{ParentPoint} * \text{NamePoint} + 2)$$

โดยที่

ParentPoint คือ คะแนนความคล้ายกันของชื่อคลาสของลักษณะประจำ x กับชื่อคลาสของลักษณะประจำ y

สำหรับ UsagePoint ใน UMLDiff กำหนดให้เป็นคะแนนความคล้ายกันของชื่อคลาสที่เรียกใช้ลักษณะประจำ x และชื่อคลาสที่เรียกใช้ลักษณะประจำ y แต่ในงานวิจัยนี้เป็นการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดซึ่งไม่มีการเรียกใช้ลักษณะประจำ ดังนั้น UsagePoint จึงมีค่าเป็น 0

4) การพิจารณาความคล้ายกันโดยรวมของการดำเนินการ

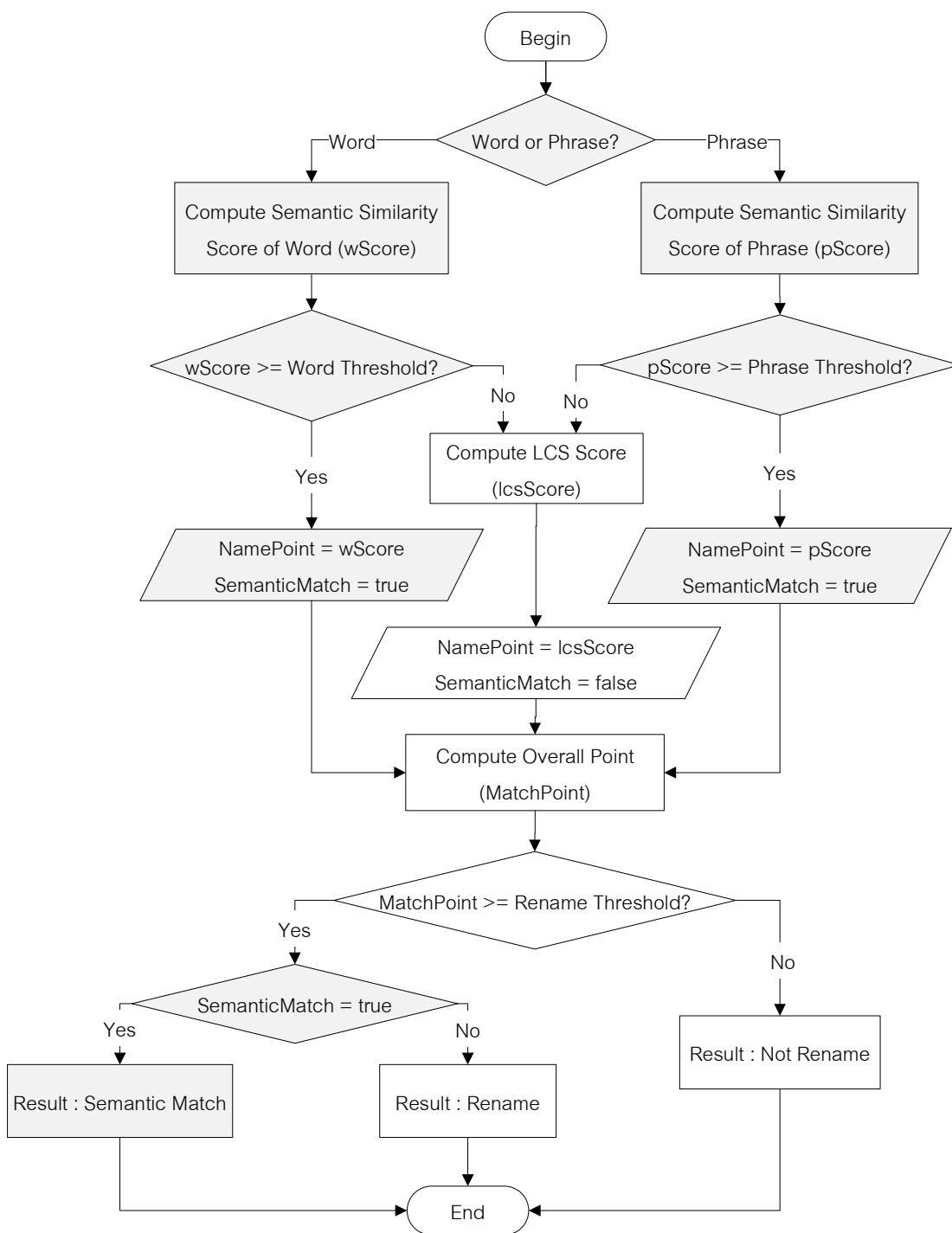
$$\text{MatchPoint} = (\text{ParentPoint} * \text{NamePoint} + \text{UsagePoint}) / (\text{ParentPoint} * \text{NamePoint} + 2)$$

โดยที่

ParentPoint คือ คะแนนความคล้ายกันของชื่อคลาสของการดำเนินการ x กับชื่อคลาสของการดำเนินการ y

สำหรับ UsagePoint ใน UMLDiff กำหนดให้เป็นคะแนนความคล้ายกันของชื่อคลาสที่เรียกใช้การดำเนินการ x และชื่อคลาสที่เรียกใช้การดำเนินการ y แต่ในงานวิจัยนี้เป็นการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดซึ่งไม่มีการเรียกใช้การดำเนินการ ดังนั้น UsagePoint จึงมีค่าเป็น 0

เมื่อทำการคำนวณคะแนนความคล้ายกันโดยรวมหรือ MatchPoint แล้ว S-UMLDiff จะนำคะแนนที่ได้มาเปรียบเทียบกับค่าขีดแบ่งของการเปลี่ยนชื่อ หาก MatchPoint มากกว่าหรือเท่ากับค่าขีดแบ่งของการเปลี่ยนชื่อ S-UMLDiff จะสรุปว่าเอนทิตี x ในระบบงานเดิมและเอนทิตี y ในระบบงานใหม่เป็นเอนทิตีเดียวกัน ซึ่งหาก NamePoint ที่ใช้ในการคำนวณ MatchPoint นั้นเป็นคะแนนที่ได้จากการเปรียบเทียบคำเชิงความหมายโดยใช้เวกเตอร์ S-UMLDiff จะสรุปว่าเอนทิตีทั้งสองมีชื่อคล้ายกันในเชิงความหมาย แต่หาก NamePoint ที่ใช้ในการคำนวณ MatchPoint เป็นคะแนนที่ได้จากการวัดความคล้ายกันของรูปคำด้วยวิธี LCS จะทำให้ S-UMLDiff สรุปว่าเอนทิตี x ในระบบงานเดิมถูกเปลี่ยนชื่อเป็นเอนทิตี y ในระบบงานใหม่ อัลกอริทึมของการตรวจสอบการเปลี่ยนชื่อของเอนทิตีแสดงดังภาพที่ 3.5 ขั้นตอนที่ไม่แรเงาคือขั้นตอนที่มีอยู่แล้วใน UMLDiff ส่วนขั้นตอนที่แรเงาคือขั้นตอนที่ผู้วิจัยเพิ่มเติมเข้ามาใน S-UMLDiff



ภาพที่ 3.5 การตรวจสอบการเปลี่ยนชื่อของอัลกอริทึม S-UMLDiff

3.7 ตัวอย่างการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดด้วยอัลกอริทึม S-UMLDiff

ผู้วิจัยขออธิบายการทำงานของอัลกอริทึม S-UMLDiff โดยใช้ตัวอย่างของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 3.2 และ 3.3 โดยกำหนดตัวแปรเพื่อใช้ในการอธิบาย ดังนี้

match คือ ชุดของเอนทิตีที่อัลกอริทึม S-UMLDiff สรุปว่าเป็นเอนทิตีเดียวกัน

first คือ ชุดของเอนทิตีของแผนภาพคลาสในภาพที่ 3.2 ที่ยังไม่ถูก match

second คือ ชุดของเอนทิตีของแผนภาพคลาสในภาพที่ 3.3 ที่ยังไม่ถูก match

การทำงานของอัลกอริทึมเริ่มจากการเปรียบเทียบความคล้ายกันของชื่อแพ็กเกจโดยใช้วิธีการของ UMLDiff คือ จะพิจารณาว่าชื่อตรงกันหรือไม่ ถ้าชื่อตรงกันจะสรุปว่าเป็นแพ็กเกจเดียวกัน แต่ถ้าชื่อไม่ตรงกันจะพิจารณาความคล้ายกันของชื่อและความคล้ายกันของโครงสร้างได้แก่ การพิจารณาความคล้ายกันของชื่อคลาสที่อยู่ภายใต้แพ็กเกจนี้ ในส่วนของการพิจารณาความคล้ายกันของชื่อนั้นจะพิจารณาความคล้ายกันเชิงความหมายโดยใช้เวิร์ดเน็ตต์ด้วย ซึ่งเป็นส่วนที่ผู้วิจัยได้เพิ่มเติมใน S-UMLDiff จากตัวอย่างนี้ทั้งสองแผนภาพมีชื่อแพ็กเกจตรงกัน คือ CreditReview ดังนั้นจึงสรุปได้ว่าเป็นแพ็กเกจเดียวกัน จากนั้นทำการเปรียบเทียบคลาสต่างๆ ที่อยู่ภายในแพ็กเกจ ดังนี้

first = {FinancialCredit, ApplicationInfo, CustRelation, Employee, Customer, LoanInfo, BankAccount, Collateral}

second = {FinancialCredit, ApplicationInfo, Committee, CustomerRelation, Officer, Client, LoanInfo, BankAccount, Address}

เริ่มจากการเปรียบเทียบชื่อว่าตรงกันหรือไม่ ผลลัพธ์ที่ได้เป็นดังนี้

match= [(FinancialCredit:FinancialCredit)(ApplicationInfo:ApplicationInfo)
(LoanInfo:LoanInfo) (BankAccount:BankAccount)]

ดังนั้นจึงมีเอนทิตีที่ยังไม่ถูก match ดังนี้

first = {CustRelation, Employee, Customer, Collateral}

second = {Committee, CustomerRelation, Officer, Client, Address}

จากนั้นจะทำการตรวจสอบว่าเป็นการเปลี่ยนชื่อหรือไม่ โดยจะพิจารณาทั้งความคล้ายกันของชื่อคลาสและความคล้ายกันของชื่อลักษณะประจำและชื่อการดำเนินการที่อยู่ภายใน

แต่ละคลาส ผลของการเปรียบเทียบในขั้นตอนนี้จะสรุปได้ว่า CustRelation ถูกเปลี่ยนชื่อเป็น CustomerRelation และจากการเพิ่มความสามารถในการเปรียบเทียบคำเชิงความหมายจึงทำให้สรุปได้ว่า Employee และ Officer เป็นคลาสเดียวกันในเชิงความหมายเช่นเดียวกันกับ Customer และ Client ชุดข้อมูลที่ได้ในขั้นตอนนี้เป็นดังนี้

```
match = [(FinancialCredit:FinancialCredit)(ApplicationInfo:ApplicationInfo)
```

```
(LoanInfo:LoanInfo)(BankAccount:BankAccount)(CustRelation:CustomerRelation)
```

```
(Employee:Officer)(Customer:Client)]
```

```
first = {Collateral}
```

```
second = {Committee, Address}
```

ขั้นตอนต่อไปเป็นการตรวจสอบว่าเอนทิตีที่เหลือถูกย้ายหรือไม่โดยพิจารณาจาก parent ของแต่ละเอนทิตี เช่น พิจารณาวามี Collateral ในแพ็คเกจอื่นๆ ของแผนภาพที่ 3.3 หรือไม่ หรือมี Committee ในแพ็คเกจอื่นๆ ของแผนภาพที่ 3.2 หรือไม่ ในตัวอย่างนี้มีเพียงแพ็คเกจเดียวดังนั้นจึงสรุปได้ว่า Collateral ถูกลบ และ Committee, Address ถูกเพิ่ม

เมื่อทำการเปรียบเทียบในระดับคลาสแล้ว S-UMLDiff จะเริ่มทำการเปรียบเทียบในระดับลักษณะประจำ โดยใช้วิธีการของ UMLDiff คือ ทำการเปรียบเทียบในแต่ละคู่ของคลาสที่ match กัน เริ่มจากเปรียบเทียบลักษณะประจำของคลาส FinancialCredit ในภาพที่ 3.2 กับลักษณะประจำของคลาส FinancialCredit ในภาพที่ 3.3 ไปจนถึงการเปรียบเทียบลักษณะประจำของคลาส Customer ในภาพที่ 3.2 กับลักษณะประจำของคลาส Client ในภาพที่ 3.3 โดยมีขั้นตอนในการเปรียบเทียบเช่นเดียวกันกับการเปรียบเทียบคลาส คือ ตรวจสอบชื่อเอนทิตีที่ตรงกันก่อน จากนั้นจึงตรวจสอบว่าเป็นการเปลี่ยนชื่อหรือไม่ และขั้นตอนสุดท้ายคือการตรวจสอบว่าเป็นการย้ายหรือไม่ เมื่อทำการตรวจสอบลักษณะประจำแล้ว S-UMLDiff จะทำการตรวจสอบการดำเนินการในแต่ละคู่ของคลาสที่ match กัน โดยใช้วิธีการเดียวกันกับการเปรียบเทียบลักษณะประจำ

หลังจากทำการตรวจสอบเงื่อนไขที่ครบถ้วนแล้วจะทำให้ได้คลาสและลักษณะประจำที่ไม่ถูก match ดังนี้

คลาสที่ถูกลบ = {Collateral}

ลักษณะประจำที่ถูกลบ = {collId,contractId,appraisalAmount,appld,appld,empld,address}

คลาสที่ถูกเพิ่ม = {Committee,Address}

ลักษณะประจำที่ถูกเพิ่ม = {approveDate,rejectDate, addrName,addrNo,street,district,city, zipCode,appNo,appNo,expiryDate,expiryDate,empNo}

นอกจากการเพิ่มความสามารถในการเปรียบเทียบค่าเชิงความหมายแล้ว S-UMLDiff ยังสามารถตรวจสอบการเปลี่ยนแปลงประเภทของส่วนย่อยได้อีกด้วย โดยทำการเปรียบเทียบคลาสที่ถูกลบกับลักษณะประจำที่ถูกเพิ่ม และเปรียบเทียบลักษณะประจำที่ถูกลบกับคลาสที่ถูกเพิ่ม ผลลัพธ์ที่ได้จะทำให้สรุปได้ว่าลักษณะประจำ address ในภาพที่ 3.2 และคลาส Address ในภาพที่ 3.3 คือข้อมูลเดียวกันแต่มีการเปลี่ยนประเภทของส่วนย่อย ผลลัพธ์ทั้งหมดที่ได้จากการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 3.2 และ 3.3 แสดงในภาคผนวก ก มีข้อสังเกตว่า S-UMLDiff สรุปว่า acctId กับ acctNo เป็นข้อมูลเดียวกันแต่ถูกเปลี่ยนชื่อ ในขณะที่ appld ถูกลบ และ appNo ถูกเพิ่ม ทั้งๆ ที่ลักษณะประจำทั้งสองควรจะเป็นข้อมูลเดียวกัน สาเหตุเกิดจากการทดสอบนี้ได้กำหนดค่าขีดแบ่งของการเปลี่ยนชื่อเป็น 0.5 ซึ่งคะแนนของ appld และ appNo ได้ค่าน้อยกว่า 0.5 ดังนั้น S-UMLDiff จึงสรุปว่าลักษณะประจำทั้งสองไม่ใช่ข้อมูลเดียวกัน ส่วน acctId และ acctNo มีคะแนนเป็น 0.5 ซึ่งมีค่าเท่ากับค่าขีดแบ่ง ดังนั้น S-UMLDiff จึงสรุปว่าลักษณะประจำทั้งสองคือข้อมูลเดียวกัน

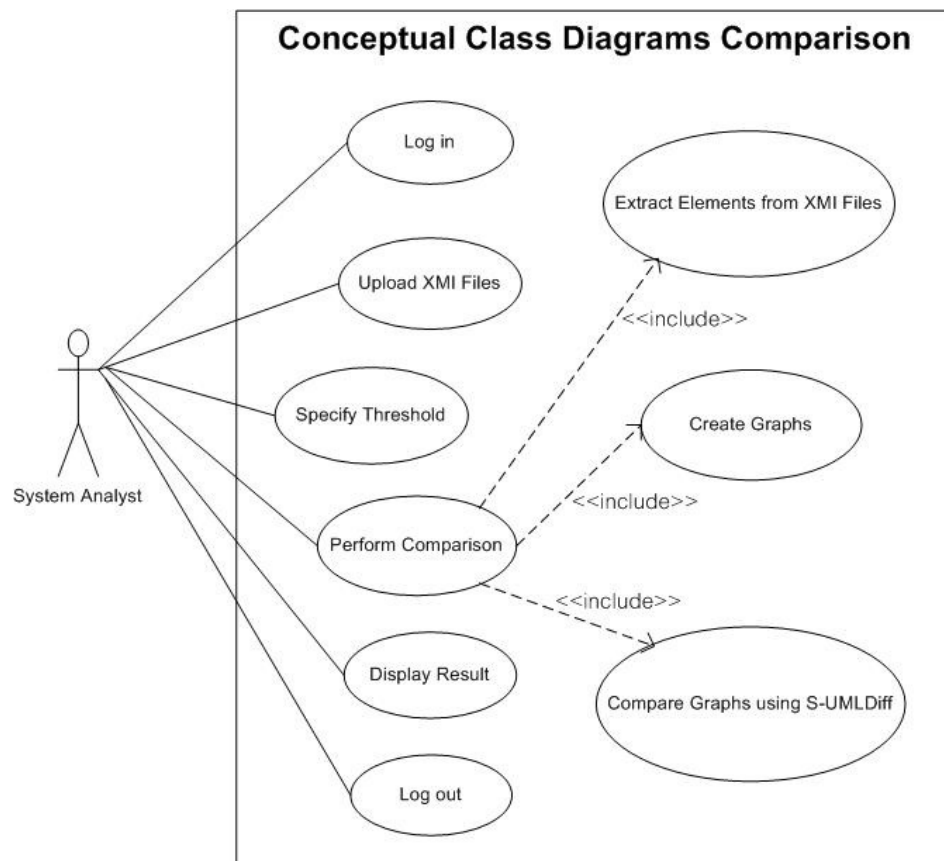
บทที่ 4

การพัฒนาเครื่องมือ

ในงานวิจัยนี้ผู้วิจัยได้ทำการพัฒนาโปรแกรมประยุกต์แบบเว็บ (Web Application) สำหรับใช้เป็นเครื่องมือในการเปรียบเทียบแผนภาพคลาสเชิงแนวคิด ซึ่งเป็นการประยุกต์ใช้อัลกอริทึม S-UMLDiff การออกแบบฟังก์ชันการทำงานการออกแบบคลาสและส่วนต่อประสานผู้ใช้ตลอดจนสภาพแวดล้อมในการพัฒนาเครื่องมือ มีดังนี้

4.1 การออกแบบฟังก์ชันการทำงานของเครื่องมือ

การออกแบบฟังก์ชันการทำงานของเครื่องมือสามารถแสดงได้ดังแผนภาพยูสเคสในภาพที่ 4.1 คำอธิบายการทำงานของแต่ละฟังก์ชันแสดงดังตารางที่ 4.1



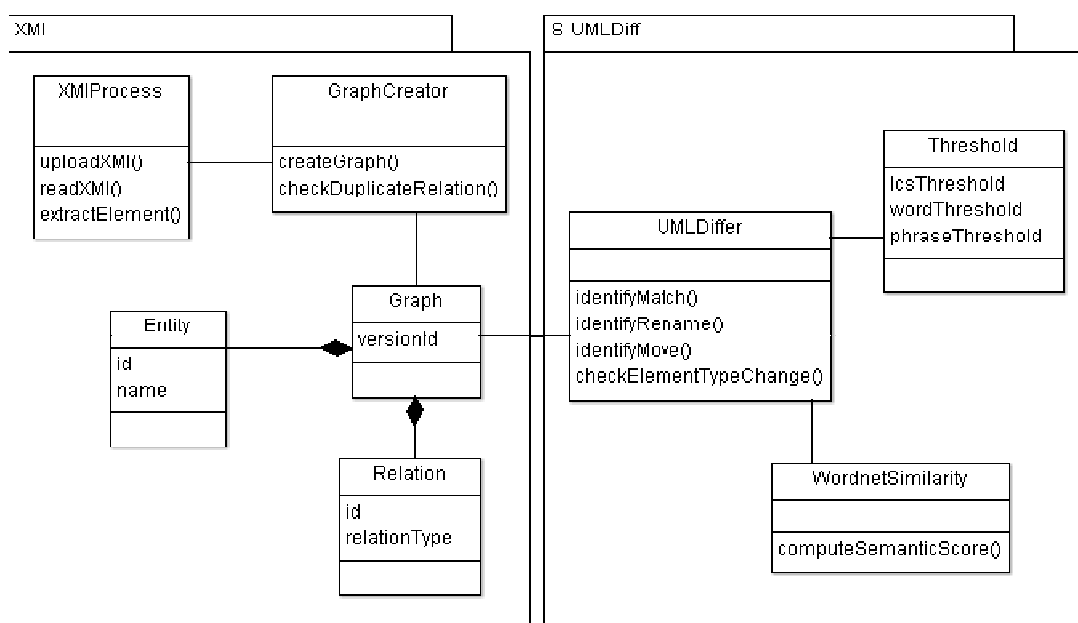
ภาพที่ 4.1 แผนภาพยูสเคสแสดงฟังก์ชันการทำงานของเครื่องมือ

ตารางที่ 4.1 คำอธิบายฟังก์ชันการทำงาน

ลำดับ	ยูสเคส	คำอธิบาย
1	Log in	นักวิเคราะห์ระบบทำการระบุรหัสผู้ใช้และรหัสผ่านเพื่อเข้าใช้งานระบบ
2	Upload XMI Files	นักวิเคราะห์ระบบระบุเอกสารเอกซ์เอ็มไอของแผนภาพคลาสเชิงแนวคิดแล้วนำเข้าสู่ระบบ
3	Specify Threshold	นักวิเคราะห์ระบบระบุค่าขีดแบ่งสำหรับใช้ในการเปรียบเทียบ
4	Perform Comparison	ระบบทำการประมวลผลโดยมีการเรียกใช้งานยูสเคส Extract Elements from XMI Files, Create Graphs และ Compare Graphs using S-UMLDiff
4.1	Extract Elements from XMI Files	ระบบทำการอ่านข้อมูลและสกัดชื่อส่วนย่อยต่างๆ จากเอกสารเอกซ์เอ็มไอ
4.2	Create Graphs	ระบบนำส่วนย่อยต่างๆ มาทำการสร้างกราฟโดยใช้วิธีการของ Xing
4.3	Compare Graphs using S-UMLDiff	ระบบทำการเปรียบเทียบกราฟโดยใช้อัลกอริทึม S-UMLDiff ตามวิธีการในหัวข้อ 3.2 ของบทที่ 3
5	Display Result	ระบบแสดงผลลัพธ์ที่ได้จากการเปรียบเทียบ
6	Log out	นักวิเคราะห์ระบบทำการ Log out เมื่อต้องการออกจากระบบ

4.2 การออกแบบแผนภาพคลาสของเครื่องมือ

ในการพัฒนาเครื่องมือผู้วิจัยได้ทำการออกแบบคลาสโดยแบ่งเป็นสองแพ็คเกจตามกลุ่มการทำงานหลัก ได้แก่ แพ็คเกจ XMI และแพ็คเกจ S-UMLDiff สำหรับแพ็คเกจ XMI จะเป็นการนำเข้าข้อมูลเอกสารเอกซ์เอ็มไอ สกัดส่วนย่อยจากเอกสารรวมถึงการสร้างกราฟ ส่วนแพ็คเกจ S-UMLDiff จะเป็นการเปรียบเทียบกราฟโดยใช้อัลกอริทึม S-UMLDiff การออกแบบคลาสและความสัมพันธ์ระหว่างคลาสแสดงดังแผนภาพคลาสเชิงแนวคิดในภาพที่ 4.2



ภาพที่ 4.2 แผนภาพคลาสเชิงแนวคิดของเครื่องมือที่พัฒนา

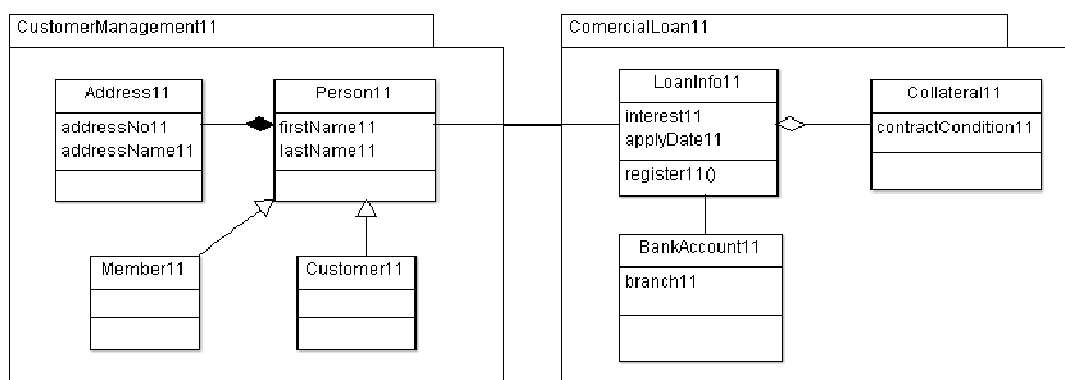
4.3 การออกแบบส่วนต่อประสานผู้ใช้

ผู้วิจัยได้ทำการออกแบบส่วนต่อประสานผู้ใช้สำหรับการนำเข้าข้อมูลเอกสารเอกซ์เอ็มไอ (ซึ่งถูกแปลงมาจากแผนภาพคลาสเชิงแนวคิดโดยใช้โปรแกรมอาร์โกยูเอ็มแอล) และออกแบบการแสดงผลของการเปรียบเทียบโดยใช้อัลกอริทึม S-UMLDiff โดยเริ่มจากหน้าจอที่ใช้ในการ Log in เข้าสู่ระบบ ดังภาพที่ 4.3 จากนั้นโปรแกรมจะแสดงหน้าจอสำหรับให้ผู้ใช้งานระบุเอกสารเอกซ์เอ็มไอของแผนภาพคลาสเชิงแนวคิดของระบบงานเดิมและระบบงานใหม่ดังภาพที่ 4.4 แต่เนื่องจากเอกสารเอกซ์เอ็มไอที่ได้จากการแปลงด้วยโปรแกรมอาร์โกยูเอ็มแอลอาจมีข้อมูลที่ไม่ถูกต้องเกินมาในเอกสารยกตัวอย่างเช่น มีแท็ก (Tag) ของคลาสที่ไม่ปรากฏในแผนภาพที่ถูกแปลง หรือมีแท็กของความสัมพัทธ์ระหว่างสองเอนทิตีมากกว่าหนึ่งความสัมพันธ์ จากปัญหานี้ผู้วิจัยแก้ไขด้วยการระบุรหัสของแผนภาพคลาสเชิงแนวคิดในทุกส่วนย่อยดังภาพที่ 4.5 จากนั้นจึงทำ

การแปลงเป็นเอกสารเอกซ์เอ็มไอ และทำการระบุรหัสเดียวกันนี้ในขั้นตอนของการนำเอกสารเอกซ์เอ็มไอเข้าสู่ระบบดังภาพที่ 4.6 ในการอ่านข้อมูลจากเอกสารเอกซ์เอ็มไอเพื่อสกัดชื่อส่วนย่อยต่างๆ นั้นโปรแกรมจะสนใจเฉพาะชื่อส่วนย่อยที่ลงท้ายด้วยรหัสที่ระบุ จากนั้นจึงนำส่วนย่อยต่างๆ มาสร้างข้อมูลกราฟตามวิธีการของ Xing โดยจะแสดงข้อมูลเอนทิตีและความสัมพันธ์ทั้งหมดบนหน้าจอ รวมถึงแจ้งให้ผู้ใช้งานทราบว่าเอนทิตีคู่ใดบ้างที่เกี่ยวข้องกันมากกว่าหนึ่งความสัมพันธ์ ดังภาพที่ 4.7 เพื่อให้ผู้ใช้งานพิจารณาและเลือกลบรายการความสัมพันธ์ที่ไม่ถูกต้องออกไป

ภาพที่ 4.3 หน้าจอ Log in เพื่อเข้าสู่ระบบ

ภาพที่ 4.4 หน้าจอที่ใช้ในการนำเข้าเอกสารเอกซ์เอ็มไอ



ภาพที่ 4.5 ตัวอย่างการระบุรหัสในแผนภาพคลาสเชิงแนวคิด

ภาพที่ 4.6 ตัวอย่างการระบุรหัสและเอกสารเอกซ์เอ็มไอ

Entity

Seq.	Entity Type	Entity Name
<input type="checkbox"/> 1		Virtual Root
<input type="checkbox"/> 2	Package	CustomerManagement
<input type="checkbox"/> 3	Package	ComercialLoan
<input type="checkbox"/> 4	Class	Address
<input type="checkbox"/> 5	Class	Customer
<input type="checkbox"/> 6	Class	Person
<input type="checkbox"/> 7	Class	Member

• Duplicated relation : row 21 and 22
row 21 : [Relation Type = USE | Entity Name = 'Address' Related Entity = 'Person']
row 22 : [Relation Type = IMPLEMENT | Entity Name = 'Person' Related Entity = 'Address']

• Duplicated relation : row 22 and 21
row 22 : [Relation Type = IMPLEMENT | Entity Name = 'Person' Related Entity = 'Address']
row 21 : [Relation Type = USE | Entity Name = 'Address' Related Entity = 'Person']

Relation

Seq.	Relation Type	Entity Name	Related Entity
<input type="checkbox"/> 1	CONTAIN	Virtual Root	CustomerManagement
<input type="checkbox"/> 2	CONTAIN	Virtual Root	ComercialLoan
<input type="checkbox"/> 3	CONTAIN	CustomerManagement	Address
<input type="checkbox"/> 4	CONTAIN	CustomerManagement	Customer
<input type="checkbox"/> 5	CONTAIN	CustomerManagement	Person
<input type="checkbox"/> 6	CONTAIN	CustomerManagement	Member
<input type="checkbox"/> 7	CONTAIN	ComercialLoan	LoanInfo

Save Back

ภาพที่ 4.7 หน้าจอแสดงผลลัพธ์ที่ได้จากการอ่านข้อมูลจากเอกสารเอกซ์เอ็มไอ

หลังจากที่ผู้ใช้งานตรวจสอบและบันทึกข้อมูลเรียบร้อยแล้ว โปรแกรมจะแสดงหน้าจอเพื่อให้ระบุค่าขีดแบ่งดังภาพที่ 4.8 โดยที่จะแสดงค่าเริ่มต้นให้ตามค่าที่ผู้วิจัยได้ทำการทดลองปรับกับตัวอย่างที่ใช้ทดสอบแล้วเห็นว่าเป็นค่าที่เหมาะสม ได้แก่ ค่าขีดแบ่งของการเปลี่ยนชื่อ คือ 0.5 ค่าขีดแบ่งของการการวัดความคล้ายคลึงเชิงความหมายของคำ คือ 0.7 และค่าขีดแบ่งของการวัดความคล้ายคลึงเชิงความหมายของวลีคือ 0.9 จากนั้นจึงเริ่มทำการประมวลผลด้วยอัลกอริทึม S-

UMLDiff ผลลัพธ์ที่ได้จากการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดจะแสดงดังภาพที่ 4.9 หากผู้ใช้งานเห็นว่าผลลัพธ์ที่ได้นั้นไม่เหมาะสมสามารถย้อนกลับมาที่หน้าจอของการกำหนดค่าขีดแบ่งเพื่อทำการแก้ไขค่าได้ตามต้องการ แล้วทำการประมวลผลใหม่เพื่อดูผลลัพธ์อีกครั้ง

ภาพที่ 4.8 หน้าจอสำหรับระบุค่าขีดแบ่ง

Seq.	Original System	New System	Type	Match Point	Semantic Match Point
1	Virtual Root	Virtual Root	match	0.0	0.0
2	ComercialLoan	ComercialLoan	match	0.0	0.0
3	CustomerManagement	ClientDirection	smatch	1.0	1.0
4	Collateral	Collateral	match	0.0	0.0
5	BankAccount	BankAccount	match	0.0	0.0
6	LoanInfo	LoanInformation	smatch	0.875	1.0
7	Address	Location	smatch	0.608145	0.823529
8	Customer	Client	smatch	0.75	1.0
9	Person	Personality	rename	0.907258	0.0
10	Member	Association	smatch	0.608145	0.823529
11	applyDate	applyDate	match	0.0	0.0
12	register()	register()	match	0.0	0.0
13	firstName	firstName	match	0.0	0.0
14	lastName	lastName	match	0.0	0.0
15	contractCondition	agreementStatus	smatch	0.944444	0.944444
16	branch	division	smatch	0.952381	0.952381
17	interest	interestRate	rename	0.666667	0.0
18	addressNo	LocationNo	smatch	0.911765	0.911765
19	addressName	LocationName	smatch	0.911765	0.911765

ภาพที่ 4.9 หน้าจอแสดงผลลัพธ์ของการเปรียบเทียบ

ภาพที่ 4.9 เป็นผลลัพธ์ที่ได้จากการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดชุดที่ 1 ในภาคผนวก ข โดยใช้อัลกอริทึม S-UMLDiff ผลลัพธ์จากภาพนี้เป็นการแสดงเอนทิตีของแผนภาพคลาสระบบเดิมและระบบใหม่ที่จับคู่กัน และแสดงให้เห็นว่าเอนทิตีแต่ละคู่หนึ่งมีการเปลี่ยนแปลงอย่างไร คอลัมน์ Match Point คือ คะแนนความคล้ายกันโดยรวม คอลัมน์ Semantic Match Point คือ คะแนนความคล้ายกันเชิงความหมายที่ได้จากเวรด์เน็ต ตัวอย่างของผลลัพธ์ที่แสดงในภาพนี้ได้แก่ ComercialLoan ของระบบงานเดิมมีชื่อตรงกัน (match) กับ ComercialLoan ของ

ระบบงานใหม่ LoanInfo ของระบบงานเดิมมีชื่อคล้ายกันในเชิงความหมาย(smatch) กับ LoanInfomation ของระบบงานใหม่ โดยมีคะแนนความคล้ายกันเชิงความหมายเท่ากับ 1.0 และมีความคล้ายกันโดยรวมเท่ากับ 0.875 ส่วน Person ในระบบงานเดิมถูกเปลี่ยนชื่อ (rename) เป็น Personality ในระบบงานใหม่ เป็นต้น

4.4 สภาพแวดล้อมในการพัฒนาเครื่องมือ

รายละเอียดของฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ที่ใช้ในการพัฒนาเครื่องมือ มีดังนี้

4.4.1 ฮาร์ดแวร์

ฮาร์ดแวร์สำหรับพัฒนาเครื่องมือ มีรายละเอียดดังนี้

- 1) เครื่องคอมพิวเตอร์โน้ตบุ๊ก หน่วยประมวลผล อินเทล คอร์ i5 2.5 กิกะเฮิร์ตซ์ (Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz)
- 2) หน่วยความจำ (Memory) 4.0 กิกะไบต์
- 3) จานบันทึกแบบแข็ง (Hard disk) ความจุ 600 กิกะไบต์

4.4.2 ซอฟต์แวร์

ซอฟต์แวร์สำหรับพัฒนาเครื่องมือ มีรายละเอียดดังนี้

- 1) ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 7 (Microsoft Windows 7)
- 2) ภาษาจาวาเวอร์ชัน 1.7 (Java Development Kit - JDK 1.7) สำหรับพัฒนาโปรแกรม
- 3) โปรแกรมอีclipse (Eclipse Java EE IDE) รุ่น Juno Service Release 1 สำหรับเขียนรหัสต้นฉบับ
- 4) อพาเชทอมแคท เว็บเซิร์ฟเวอร์ (Apache Tomcat Web Server) เวอร์ชัน 7.0 สำหรับให้เครื่องคอมพิวเตอร์มีคุณสมบัติเป็นเว็บเซิร์ฟเวอร์ (Web Server)
- 5) ฐานข้อมูล PostgreSQL เวอร์ชัน 9.2 สำหรับจัดเก็บข้อมูลที่ใช้การประมวลผลของโปรแกรม
- 6) โปรแกรมพีจีแอดมิน (pgAdmin) เวอร์ชัน 1.16.0 สำหรับใช้ในการบริหารจัดการฐานข้อมูล PostgreSQL

บทที่ 5

ผลการทดสอบ

ผู้วิจัยเห็นว่าหาก S-UMLDiff สามารถแสดงผลที่ได้ถูกต้องจะส่งผลให้สามารถนำผลที่ได้นั้นไปใช้ในการตรวจสอบความครบถ้วนของความต้องการได้อย่างมีประสิทธิภาพผู้วิจัยจึงได้ทำการทดสอบเครื่องมือโดยการนำไปใช้ในการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดที่ผู้วิจัยออกแบบเองจำนวน 10 ชุด พร้อมทั้งวัดความถูกต้องของอัลกอริทึมด้วยการคำนวณค่า Precision และ Recall ดังนี้

Precision คือ เปอร์เซ็นต์ของการรายงานผลได้อย่างถูกต้อง

$$\text{หาได้จาก } (M_{\text{actual}} \cap M_{\text{reported}}) / M_{\text{reported}}$$

Recall คือ เปอร์เซ็นต์ของการรายงานผลได้อย่างครบถ้วน

$$\text{หาได้จาก } (M_{\text{actual}} \cap M_{\text{reported}}) / M_{\text{actual}}$$

โดยที่

M_{actual} คือ จำนวนความแตกต่างระหว่างแผนภาพคลาสของระบบเดิมและระบบใหม่ที่
ได้จากการพิจารณาของผู้วิจัย

M_{reported} คือ จำนวนความแตกต่างระหว่างแผนภาพคลาสของระบบเดิมและระบบใหม่
ที่ S-UMLDiff รายงานผล

5.1 ผลการทดสอบเปรียบเทียบแผนภาพคลาสเชิงแนวคิดจำนวน 10 ชุด

ผลลัพธ์ที่ได้จากการทดสอบเปรียบเทียบแผนภาพคลาสเชิงแนวคิดด้วย S-UMLDiff แสดง
ดังตารางที่ 5.1 สำหรับชุดของแผนภาพคลาสเชิงแนวคิดที่ใช้ทดสอบแสดงในภาคผนวก ข

ตารางที่ 5.1 ผลการทดสอบการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดด้วย S-UMLDiff

ชุดที่	M_{actual}	M_{reported}	$M_{\text{actual}} \cap M_{\text{reported}}$	Precision	Recall
1	11	11	10	91%	91%
2	15	15	15	100%	100%
3	8	8	8	100%	100%

ตารางที่ 5.1 ผลการทดสอบการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดด้วย S-UMLDiff (ต่อ)

ชุดที่	M_{actual}	M_{reported}	$M_{\text{actual}} \cap M_{\text{reported}}$	Precision	Recall
4	28	34	25	74%	89%
5	20	20	19	95%	95%
6	37	37	36	97%	97%
7	24	30	22	73%	92%
8	24	30	22	73%	92%
9	20	22	18	82%	90%
10	34	34	33	97%	97%

ผลลัพธ์ที่ได้จากการทดสอบถือว่าอยู่ในระดับที่น่าพอใจเนื่องจากได้ค่า Precision และ Recall ที่ค่อนข้างสูง อีกทั้ง S-UMLDiff สามารถตรวจสอบความคล้ายกันเชิงความหมายของชื่อในแผนภาพและสามารถตรวจสอบได้ว่ามีเอนทิตีใดบ้างที่มีการเปลี่ยนประเภทของส่วนย่อย

5.2 การทดสอบเปรียบเทียบแผนภาพคลาสเชิงแนวคิดของระบบงานจริง

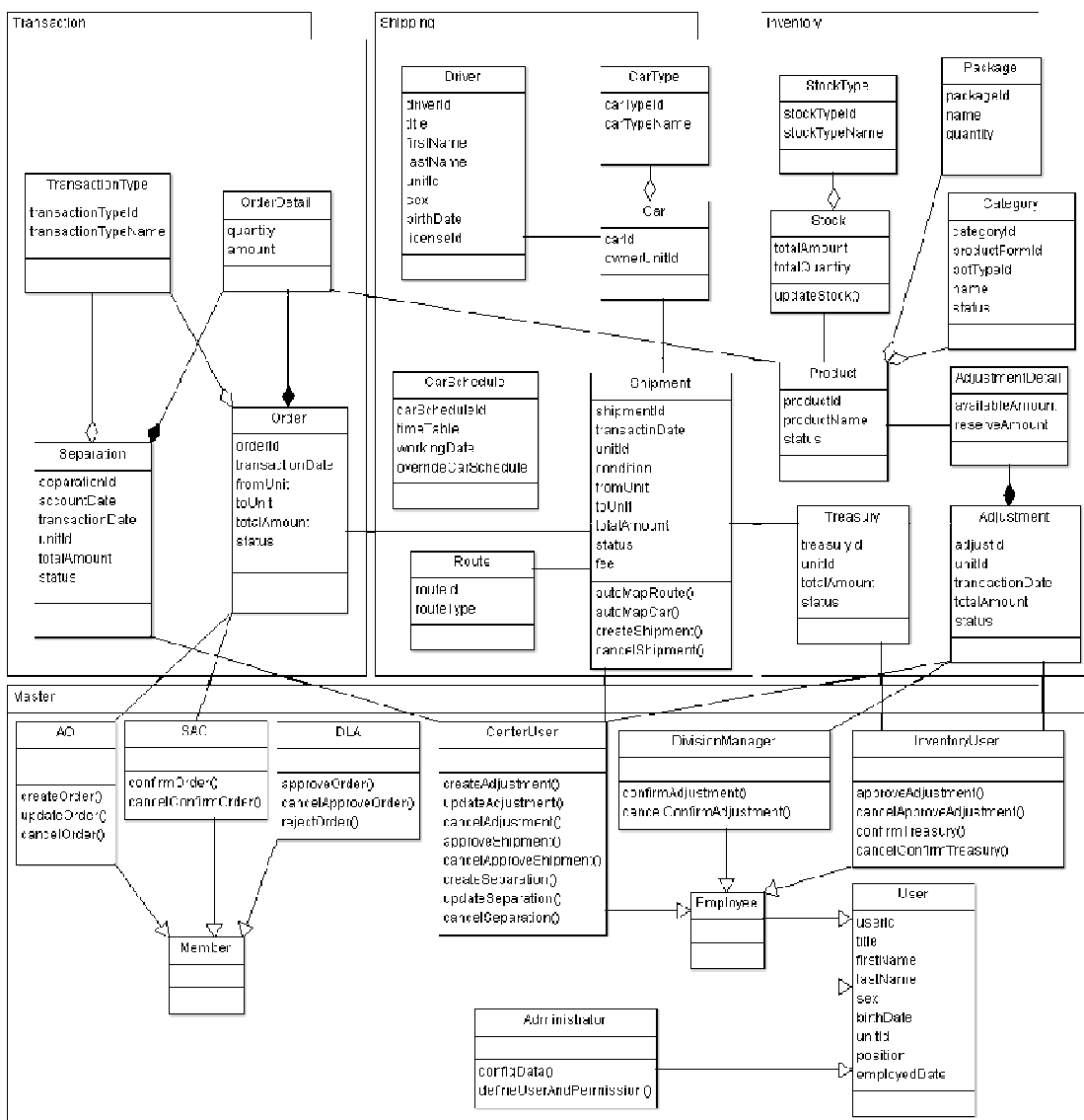
ผู้วิจัยได้ทำการทดสอบเปรียบเทียบแผนภาพคลาสเชิงแนวคิดของระบบงานจริง คือระบบจัดการเงินในคลังของธนาคารแห่งหนึ่งซึ่งเป็นการพัฒนาระบบใหม่บนพื้นฐานของข้อกำหนดของระบบเดิมจำนวน 1 ระบบงาน โดยกำหนดให้ผู้เชี่ยวชาญคนที่ 1 และ 2 เป็นผู้ออกแบบแผนภาพคลาสเชิงแนวคิดของระบบเดิมและระบบใหม่ตามลำดับ และผู้เชี่ยวชาญคนที่ 3 เป็นผู้พิจารณาเพื่อหาความแตกต่างระหว่างแผนภาพคลาสทั้งสอง (M_{actual}) คุณสมบัติของผู้เชี่ยวชาญแสดงดังตารางที่ 5.2

ตารางที่ 5.2 คุณสมบัติของผู้เชี่ยวชาญที่ร่วมทำการทดสอบ

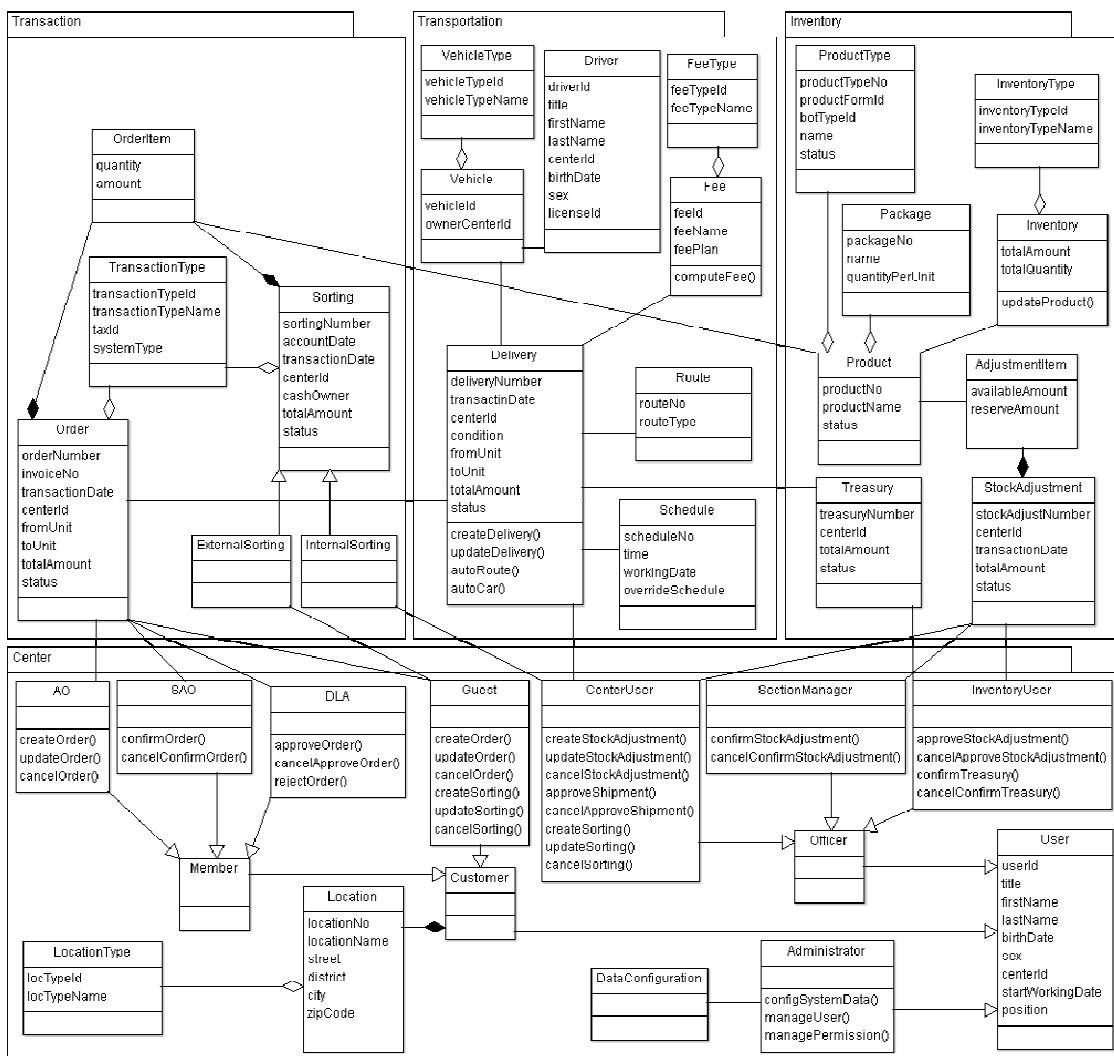
ผู้เชี่ยวชาญ	ประสบการณ์ทำงาน
คนที่ 1	มีประสบการณ์ในการทำงาน 12 ปี ดังนี้ เคยทำงานตำแหน่งโปรแกรมเมอร์ (Programmer) 10 ปี ปัจจุบันทำงานตำแหน่งนักวิเคราะห์ธุรกิจ (Business Analyst) 2 ปี

ตารางที่ 5.2 คุณสมบัติของผู้เชี่ยวชาญที่ร่วมทำการทดสอบ (ต่อ)

ผู้เชี่ยวชาญ	ประสบการณ์ทำงาน
คนที่ 2	มีประสบการณ์ในการทำงาน 12 ปี ดังนี้ เคยทำงานตำแหน่งโปรแกรมเมอร์ 7 ปี ปัจจุบันทำงานตำแหน่งนักวิเคราะห์ระบบ (System Analyst) 5 ปี
คนที่ 3	มีประสบการณ์ในการทำงานในตำแหน่งโปรแกรมเมอร์ 9 ปี



ภาพที่ 5.1 แผนภาพคลาสเชิงแนวคิดของระบบงานเดิมที่ออกแบบโดยผู้เชี่ยวชาญคนที่ 1



ภาพที่ 5.2 แผนภาพคลาสเชิงแนวคิดของระบบงานใหม่ที่ออกแบบโดยผู้เชี่ยวชาญคนที่ 2

ภาพที่ 5.1 คือ แผนภาพคลาสเชิงแนวคิดที่ออกแบบตามความต้องการซอฟต์แวร์ของระบบเดิมโดยผู้เชี่ยวชาญคนที่ 1 ส่วนภาพที่ 5.2 คือ แผนภาพคลาสเชิงแนวคิดที่ออกแบบตามความต้องการซอฟต์แวร์ของระบบใหม่ ซึ่งถูกออกแบบโดยผู้เชี่ยวชาญคนที่ 2

ผู้วิจัยได้ทำการแปลงแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 เป็นเอกสารเอกซ์เอ็มไอโดยใช้โปรแกรมอาร์โกยูเอ็มแอลจากนั้นนำเข้าเครื่องมือที่พัฒนาเพื่อทำการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดโดยใช้อัลกอริทึม S-UMLDiff เมื่อทำการเปรียบเทียบผลลัพธ์ที่ได้จาก S-UMLDiff และผลลัพธ์ที่ได้จากผู้เชี่ยวชาญคนที่ 3 สามารถคำนวณค่า Precision และ Recall ได้ดังนี้

$$M_{\text{actual}} = 92$$

$$M_{\text{reported}} = 104$$

$$\text{Precision} = (M_{\text{actual}} \cap M_{\text{reported}}) / M_{\text{reported}} = 77/104 \text{ คิดเป็น } 74\%$$

$$\text{Recall} = (M_{\text{actual}} \cap M_{\text{reported}}) / M_{\text{actual}} = 77/92 \text{ คิดเป็น } 84\%$$

นอกจากนี้ผู้วิจัยยังได้ทำการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดของระบบงานเดียวกันนี้โดยใช้อัลกอริทึม UMLDiff ผลลัพธ์ที่ได้เป็นดังนี้

$$M_{\text{actual}} = 92$$

$$M_{\text{reported}} = 177$$

$$\text{Precision} = (M_{\text{actual}} \cap M_{\text{reported}}) / M_{\text{reported}} = 49/177 \text{ คิดเป็น } 28\%$$

$$\text{Recall} = (M_{\text{actual}} \cap M_{\text{reported}}) / M_{\text{actual}} = 49/92 \text{ คิดเป็น } 53\%$$

เมื่อพิจารณาผลลัพธ์ที่ได้จากการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดด้วยอัลกอริทึม S-UMLDiff และ UMLDiff จะเห็นได้ว่า S-UMLDiff มีความสามารถในการรายงานผลได้อย่างถูกต้องมากกว่า UMLDiff ค่า Precision และ Recall ที่ได้จากทั้งสองวิธีมีความแตกต่างกันมาก เนื่องจาก UMLDiff ไม่สามารถเปรียบเทียบคำเชิงความหมายได้เมื่อเปรียบเทียบเอนทิตีที่เขียนต่างกันแต่มีความหมายเดียวกันจึงสรุปว่าเอนทิตีหนึ่งถูกลบและเอนทิตีหนึ่งถูกเพิ่ม ดังนั้น เอนทิตีทั้งหมดที่อยู่ภายในเอนทิตีที่ถูกลบจึงสรุปว่าถูกลบทั้งหมด และเอนทิตีทั้งหมดที่อยู่ภายในเอนทิตีที่ถูกเพิ่มจึงสรุปว่าถูกเพิ่มทั้งหมดด้วย ยกตัวอย่างเช่น ในการเปรียบเทียบแพ็คเกจ Shipping ในภาพที่ 5.1 และแพ็คเกจ Transportation ในภาพที่ 5.2 อัลกอริทึม UMLDiff สรุปว่า Shipping ถูกลบ และ Transportation ถูกเพิ่ม ดังนั้นจึงสรุปว่าเอนทิตีต่างๆ ที่อยู่ภายใน Shipping ถูกลบไปด้วย และสรุปว่าเอนทิตีทั้งหมดที่อยู่ภายใน Transportation ถูกเพิ่มไปด้วย ในขณะที่ S-UMLDiff สามารถตรวจสอบได้ว่า Shipping และ Transportation เป็นชื่อเดียวกันในเชิงความหมาย จึงทำการเปรียบเทียบเอนทิตีที่อยู่ภายในด้วย รายละเอียดผลการเปรียบเทียบของทั้งสองวิธีรวมถึงผลการพิจารณาจากผู้เชี่ยวชาญคนที่ 3 แสดงในภาคผนวก ค

5.3 อภิปรายผล

จากการทดสอบอัลกอริทึม S-UMLDiff ด้วยการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดจำนวน 10 ชุด และทำการทดสอบด้วยระบบงานจริง สามารถอภิปรายผลลัพธ์ที่ได้ ดังนี้

5.3.1 ค่าขีดแบ่งที่กำหนดมีผลต่อการตัดสินใจของ S-UMLDiff ยกตัวอย่างเช่น ในการเปรียบเทียบความหมายของชื่อคลาส Employee และ Officer ซึ่งคะแนนความคล้ายกันเชิงความหมายที่วัดด้วยวิธี Wu & Palmer มีค่าเท่ากับ 0.75 หากกำหนดค่าขีดแบ่งเป็น 0.7 จะทำให้ S-UMLDiff ตัดสินว่า Employee และ Officer เป็นคลาสเดียวกันในเชิงความหมาย แต่หากกำหนดค่าขีดแบ่งเป็น 0.8 จะทำให้ S-UMLDiff ตัดสินว่าทั้งสองคลาสไม่ใช่คลาสเดียวกันในเชิงความหมาย ดังนั้น หากค่าขีดแบ่งที่กำหนดมีค่าต่ำจะส่งผลให้มีเอนทิตีที่มีความคล้ายกันเชิงความหมายจำนวนมาก ในทางกลับกันหากค่าขีดแบ่งที่กำหนดมีค่าสูงจะทำให้มีจำนวนเอนทิตีที่มีความคล้ายกันเชิงความหมายน้อย

5.3.2 การที่จำนวนความแตกต่างระหว่างแผนภาพคลาสที่ได้จากการพิจารณาของผู้วิจัยหรือผู้เชี่ยวชาญ (M_{actual}) ไม่ตรงกันกับจำนวนความแตกต่างระหว่างแผนภาพคลาสที่ S-UMLDiff รายงานผล ($M_{reported}$) มีสาเหตุหลักดังนี้

1) การกำหนดค่าขีดแบ่งที่ใช้ในการเปรียบเทียบ ยกตัวอย่างเช่น ผู้วิจัยเห็นว่า appId และ appNo เป็นเอนทิตีเดียวกันที่ถูกเปลี่ยนชื่อ แต่จากการทดสอบปรากฏว่าความคล้ายคลึงโดยรวมของคำทั้งสองที่หาได้จากอัลกอริทึมมีค่าน้อยกว่าค่าขีดแบ่งที่กำหนด ดังนั้น S-UMLDiff จึงสรุปว่า appId และ appNo ไม่ใช่ชื่อเอนทิตีเดียวกัน

2) ความเข้าใจในความหมายของคำศัพท์ในแผนภาพของผู้วิจัยหรือผู้เชี่ยวชาญไม่ตรงกันกับคะแนนที่วัดได้จากเวรด์เน็ต ยกตัวอย่างเช่น ในการทดสอบแผนภาพคลาสเชิงแนวคิดชุดที่ 6 ในภาคผนวก ข ผู้วิจัยเห็นว่า custCode ในคลาส Customer ถูกเปลี่ยนชื่อเป็น code ในคลาส Client และ address ในคลาส Customer เป็นเอนทิตีเดียวกันในเชิงความหมายกับ location ในคลาส Client แต่ผลที่ได้จาก S-UMLDiff สรุปว่า custCode ถูกลบ location ถูกเพิ่ม และ address เป็นเอนทิตีเดียวกันในเชิงความหมายกับ code เนื่องจากคะแนนความคล้ายคลึงเชิงความหมายของ address และ code มีค่าเป็น 0.94 ในขณะที่คะแนนความคล้ายคลึงเชิงความหมายของ address กับ location มีค่าเป็น 0.82 ดังนั้น S-UMLDiff จึงสรุปว่า address และ code เป็นเอนทิตีเดียวกันในเชิงความหมาย เนื่องจากมีคะแนนมากกว่า address กับ location

3) ข้อจำกัดของอัลกอริทึมที่ไม่สามารถวัดความคล้ายกันเชิงความหมายของคำคุณศัพท์และคำวิเศษณ์ ดังนั้นจึงอาจทำให้ผลลัพธ์ที่ได้ไม่ตรงกับการพิจารณาของผู้วิจัยหรือผู้เชี่ยวชาญ

บทที่ 6

สรุปผลการวิจัย ข้อจำกัด และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนออัลกอริทึม S-UMLDiff สำหรับใช้ตรวจสอบความครบถ้วนของความต้องการในการเปลี่ยนแปลงซอฟต์แวร์สำหรับการพัฒนาซอฟต์แวร์ระบบใหม่บนพื้นฐานของข้อกำหนดความต้องการของระบบเดิม โดยนำแผนภาพคลาสเชิงแนวคิดของระบบเดิมและระบบใหม่มาเปรียบเทียบกันเพื่อให้เห็นถึงฟังก์ชันและข้อมูลที่เพิ่มเข้ามาหรือขาดหายไป จากการทดสอบพบว่าอัลกอริทึม S-UMLDiff สามารถตรวจหาความคล้ายกันและความแตกต่างระหว่างแผนภาพคลาสเชิงแนวคิดได้ แม้ว่าในแผนภาพนั้นจะกำหนดชื่อของส่วนย่อยแตกต่างกันหรือมีการเปลี่ยนประเภทของส่วนย่อยก็ตาม ประสิทธิภาพของอัลกอริทึมถือว่าอยู่ในระดับที่น่าพอใจ เนื่องจากได้ค่า Precision และ Recall ที่ค่อนข้างสูง ผู้วิจัยเห็นว่าวิธีการที่นำเสนอจะทำให้ผลลัพธ์ที่ได้ตรงตามวัตถุประสงค์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดสำหรับการพัฒนาซอฟต์แวร์ระบบใหม่บนพื้นฐานของข้อกำหนดความต้องการของระบบเดิม ซึ่งในการออกแบบอาจไม่ได้ดำเนินการโดยบุคคลเดียวกัน และอาจใช้คำต่างกันแต่มีความหมายเดียวกัน การเปรียบเทียบเชิงความหมายของชื่อส่วนย่อยและการตรวจสอบการเปลี่ยนประเภทของส่วนย่อยนี้จะทำให้ทราบถึงสิ่งที่ขาดหายไปหรือเพิ่มเข้ามาในระบบใหม่ นักวิเคราะห์ระบบและผู้ใช้สามารถนำผลลัพธ์ที่ได้ไปใช้ในการพิจารณาว่า ส่วนที่แตกต่างกันระหว่างแผนภาพคลาสเชิงแนวคิดทั้งสองเป็นส่วนที่ต้องการเปลี่ยนแปลงซอฟต์แวร์หรือไม่

6.2 อุปสรรคและข้อจำกัด

อุปสรรคในการดำเนินงานวิจัยและข้อจำกัดของอัลกอริทึมที่นำเสนอมีดังนี้

- 6.2.1 ในการแปลงแผนภาพคลาสเชิงแนวคิดเป็นเอกสารเอกซ์เอ็มไอโดยใช้โปรแกรมอาร์โกยูเอ็มแอล เอกสารที่ได้อาจมีแท็กของส่วนย่อยที่ไม่ปรากฏในแผนภาพคลาส แต่เป็นส่วนย่อยที่เคยมีในแผนภาพคลาสนั้นแต่ถูกลบออกไปแล้วในระหว่างการสร้างแผนภาพคลาสด้วยโปรแกรมอาร์โกยูเอ็มแอล หรืออาจมีความสัมพันธ์ของส่วนย่อยที่ไม่ถูกต้องปะปนมาในเอกสารด้วย ดังนั้นจึงจำเป็นต้องทำการพิจารณาและเลือกลบรายการที่ไม่ต้องการออกไป ซึ่งหากลบรายการที่ไม่ถูกต้องออกไปไม่หมดหรือเลือกลบรายการผิดอาจส่งผลให้ S-UMLDiff ทำงานผิดพลาดและแสดงผลลัพธ์ที่คลาดเคลื่อนได้

- 6.2.2 หากชื่อส่วนย่อยในแผนภาพคลาสเชิงแนวคิดมีคำคุณศัพท์หรือคำวิเศษณ์จำนวนมากอาจทำให้ผลลัพธ์ที่ได้จากการเปรียบเทียบด้วย S-UMLDiff คลาดเคลื่อน เนื่องจาก S-UMLDiff ใช้วิธี Wu & Palmer ในการวัดความคล้ายกันเชิงความหมายของคำ ซึ่งวิธีนี้ไม่สามารถวัดความคล้ายกันของคำคุณศัพท์และคำวิเศษณ์ได้
- 6.2.3 อัลกอริทึม S-UMLDiff สามารถตรวจสอบการเพิ่มและลบความสัมพันธ์แบบ Extend ได้ แต่ไม่สามารถเปรียบเทียบความแตกต่างของเส้นความสัมพันธ์แบบอื่นๆ ได้ เช่นเดียวกันกับอัลกอริทึม UMLDiff

6.3 ข้อเสนอแนะ

อัลกอริทึม S-UMLDiff และเครื่องมือสนับสนุนที่เสนอโดยงานวิจัยสามารถนำไปพัฒนาต่อยอดเป็นระบบจัดการการเปลี่ยนแปลงซอฟต์แวร์ที่มีความสามารถเต็มรูปแบบมากขึ้น ดังตัวอย่างความสามารถต่อไปนี้

- 6.3.1 สามารถเปรียบเทียบความแตกต่างของเส้นความสัมพันธ์แต่ละประเภทได้
- 6.3.2 สามารถแสดงผลการเปรียบเทียบแผนภาพแบบวิซวล (Visual) เพื่อให้นักวิเคราะห์ระบบสามารถเห็นจุดที่แตกต่างและประเภทของความแตกต่างได้ชัดเจนขึ้น
- 6.3.3 สามารถสนับสนุนการทำงานร่วมกันระหว่างแต่ละทีมในโครงการพัฒนาซอฟต์แวร์ ทั้งทีมออกแบบ ทีมพัฒนา และทีมทดสอบ หลังการตรวจพบความแตกต่างระหว่างการออกแบบระบบเดิมและใหม่แล้ว สามารถระบุผลกระทบต่อโครงการซึ่งจะเกิดขึ้นจากความแตกต่างของการออกแบบระบบแบบใหม่
- 6.3.4 เนื่องจากแผนภาพในงานวิจัยนี้เป็นแผนภาพของระบบงานหนึ่ง ซึ่งในความเป็นจริงอาจจะมีบางส่วนที่มีปฏิสัมพันธ์กับระบบงานอื่นซึ่งออกแบบโดยทีมออกแบบอื่น เมื่อแต่ละระบบงานเกิดการเปลี่ยนแปลงจึงควรที่จะสามารถวิเคราะห์ผลกระทบถึงระบบงานส่วนอื่น รวมถึงข้อขัดแย้งระหว่างการเปลี่ยนแปลงของแต่ละระบบ

รายการอ้างอิง

- [1] Sommerville, I. Software Engineering. 8th Ed. New York, NY: Addison-Wesley, 2007.
- [2] Pressman, R. Software Engineering A Practitioner's Approach. 7th Ed. Singapore: McGraw-Hill, 2010.
- [3] Object Management Group. XML Metadata Interchange [Online]. Available from: <http://www.omg.org/spec/XMI/2.4.1/>. 2012. [2012, Aug 1]
- [4] Xing, Z. Supporting Object-Oriented Evolutionary Development by Design Evolution Analysis. Doctoral Dissertation, Department of Computing Science, University of Alberta, 2008.
- [5] Princeton University. WordNet [Online]. Available from: [http://wordnet.princeton.edu /](http://wordnet.princeton.edu/). 2012.[2012, Jul 20]
- [6] Tigris.org. ArgoUML [Online]. Available from: <http://argouml.tigris.org/>. 2009. [2012, Aug 20]
- [7] Object Management Group. Unified Modeling Language [Online]. Available from: <http://www.omg.org/spec/UML/2.4.1/>. 2012.[2012, Jul 1]
- [8] Stein, C. Longest Common Subsequence [Online]. Available from: <http://www.columbia.edu/~cs2035/courses/csor4231.F11/lcs.pdf>. 2012. [2012, Sep 1]
- [9] Pedersen, T., Patwardhan, S., and Michelizzi, J. WordNet::Similarity - Measuring the Relatedness of Concepts. In Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI 2004) (Intelligent Systems Demonstrations), 1024-1025. San Jose, California: AAAI Press, 2004.
- [10] Kelte, U., Wehren, J., and Niere, J. A Generic Difference Algorithm for UML Models. In Proceedings of Software Engineering, Lecture Notes in Informatics, 105-116. Essen, Germany, 2005.

- [11] Girschick, M. Difference Detection and Visualization in UML Class Diagrams. In Technical Report TUD-CS-2006-5, TU Darmstadt, Germany, 2006.
- [12] Chawathe, S. S., and Garcia-Molina, H. Meaningful Change Detection in Structured Data. In Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data (SIGMOD 1997), 26-37. Tucson, Arizona, 1997.
- [13] Auxepaules, L., Py, D., and Lemeunier, T. A Diagnosis Method that Matches Class Diagrams in a Learning Environment for Object-Oriented Modeling. In Proceedings of 8th IEEE International Conference on Advanced Learning Technologies (ICALT 2008), 26-30. Santander, Cantabria, 2008.
- [14] Sorlin, S., Solnon, C., and Jolion, J.-M. A Generic Graph Distance Measure Based on Multivalent Matchings. Applied Graph Theory in Computer Vision and Pattern Recognition, Studies in Computational Intelligence 52(2007): 151-181.
- [15] Giunchiglia, F., Yatskevich, M., and Shvaiko, P. Semantic Matching: Algorithms and Implementation. Journal on Data Semantics 9(2007): 1-38.
- [16] Do, H. H., and Rahm, E. Matching Large Schemas: Approaches and Evaluation. Information Systems 32(2007): 857-885.
- [17] Kotb, Y. Improving the UML Consistency Using Text Semantic Similarity Approach. In Proceedings of 2nd International Conference on Computer Technology and Development (ICCTD 2010), 90-94. IEEE. 2010.
- [18] Corley, C., Csomai, A., and Mihalcea, R. Text Semantic Similarity, with Applications. In Proceedings of International Conference Recent Advances in Natural Language Processing (RANLP 2005), Borovets, Bulgaria, 2005.
- [19] Xing, Z. Supporting Object-Oriented Evolutionary Development by Design Evolution Analysis [Online]. Available from: <http://webdocs.cs.ualberta.ca/>

~stroulia/Zhenchang_Xing_Old_Home/. 2012.[2012, Aug 15]

- [20] Gad, W., and Kamel, M. PH-SSBM: Phrase Semantic Similarity Based Model for Document Clustering. In Proceedings of 2009 Second International Symposium on Knowledge Acquisition and Modeling, 197-200. DC, USA: IEEE Computer Society Washington, 2009.

ภาคผนวก

ภาคผนวก ก

ตัวอย่างของผลลัพธ์ที่ได้จากการเปรียบเทียบด้วย S-UMLDiff

ตารางที่ ก.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 3.2 และ 3.3

ลำดับ	เอนทิตีของภาพที่ 3.2	เอนทิตีของภาพที่ 3.3	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง	คะแนนความคล้ายคลึงเชิงความหมาย
1	Virtual Root	Virtual Root	match	0	0
2	CreditReview	CreditReview	match	0	0
3	FinancialCredit	FinancialCredit	match	0	0
4	ApplicationInfo	ApplicationInfo	match	0	0
5	LoanInfo	LoanInfo	match	0	0
6	BankAccount	BankAccount	match	0	0
7	CustRelation	CustomerRelation	rename	0.638889	0
8	Employee	Officer	smatch	0.698148	0.75
9	Customer	Client	smatch	0.875	1
10	creditType	creditType	match	0	0
11	checkDate	checkDate	match	0	0
12	checkStatus	checkStatus	match	0	0
13	checkCredit()	checkCredit()	match	0	0
14	appDate	appDate	match	0	0
15	status	status	match	0	0
16	limitAmount	limitAmount	match	0	0
17	outstandingBalance	outstandingBalance	match	0	0
18	interestRate	interestRate	match	0	0
19	interestAmount	interestAmount	match	0	0
20	updateBalance()	updateBalance()	match	0	0
21	computeInterest()	computeInterest()	match	0	0
22	acctName	acctName	match	0	0
23	acctType	acctType	match	0	0

ตารางที่ ก.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 3.2 และ 3.3 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 3.2	เอนทิตีของภาพที่ 3.3	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง	คะแนนความคล้ายคลึงเชิงความหมาย
24	relationType	relationType	match	0	0
25	titleName	titleName	match	0	0
26	firstName	firstName	match	0	0
27	lastName	lastName	match	0	0
28	position	position	match	0	0
29	createApplInfo()	createApplInfo()	match	0	0
30	titleName	titleName	match	0	0
31	firstName	firstName	match	0	0
32	lastName	lastName	match	0	0
33	deleteInfo()	removeInfo()	smatch	0.9	0.9
34	acctId	acctNo	rename	0.5	0
35	updateApplInfo()	updateApplInfo()	match	0	0
36	custId	custNo	rename	0.5	0
37	collId		remove	0	0
38	contractId		remove	0	0
39	appraisalAmount		remove	0	0
40	appld		remove	0	0
41	appld		remove	0	0
42	empld		remove	0	0
43	computeFee()		remove	0	0
44	Collateral		remove	0	0
45		approveDate	add	0	0
46		rejectDate	add	0	0
47		addrName	add	0	0
48		addrNo	add	0	0
49		street	add	0	0
50		District	add	0	0

ตารางที่ ก.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 3.2 และ 3.3 (ต่อ)

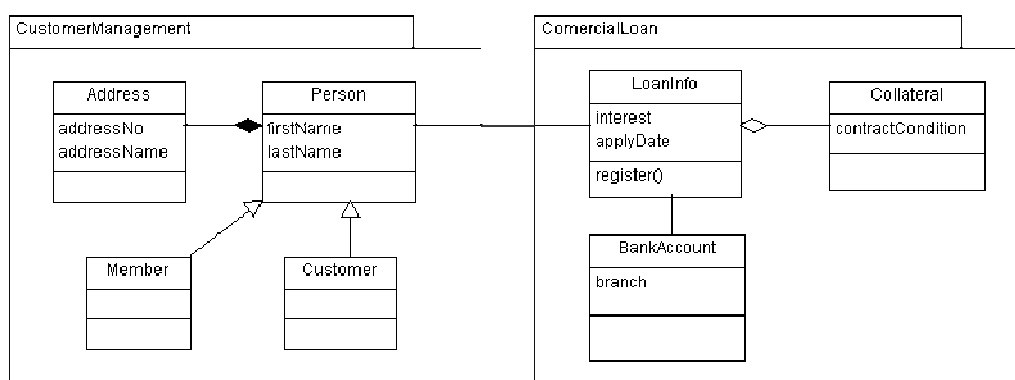
ลำดับ	เอนทิตีของภาพที่ 3.2	เอนทิตีของภาพที่ 3.3	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง	คะแนนความคล้ายคลึงเชิงความหมาย
51		city	add	0	0
52		zipCode	add	0	0
53		appNo	add	0	0
54		appNo	add	0	0
55		expiryDate	add	0	0
56		expiryDate	add	0	0
57		empNo	add	0	0
58		approveAppInfo()	add	0	0
59		rejectAppInfo()	add	0	0
60		Committee	add	0	0
61	address	Address	changetype	1	1

ภาคผนวก ข

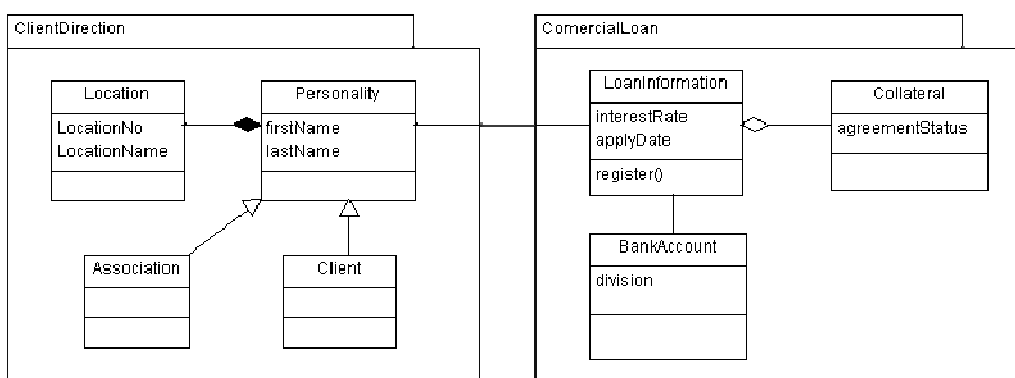
ตัวอย่างของแผนภาพคลาสเชิงแนวคิดที่ใช้ในการทดสอบ

ผู้วิจัยขอยกตัวอย่างของแผนภาพคลาสเชิงแนวคิดที่ใช้ในการทดสอบอัลกอริทึม S-UMLDiff จำนวน 10 ชุด ดังนี้

ชุดที่ 1



ภาพที่ ข.1 แผนภาพคลาสเชิงแนวคิดชุดที่ 1 ระบบงานเดิม

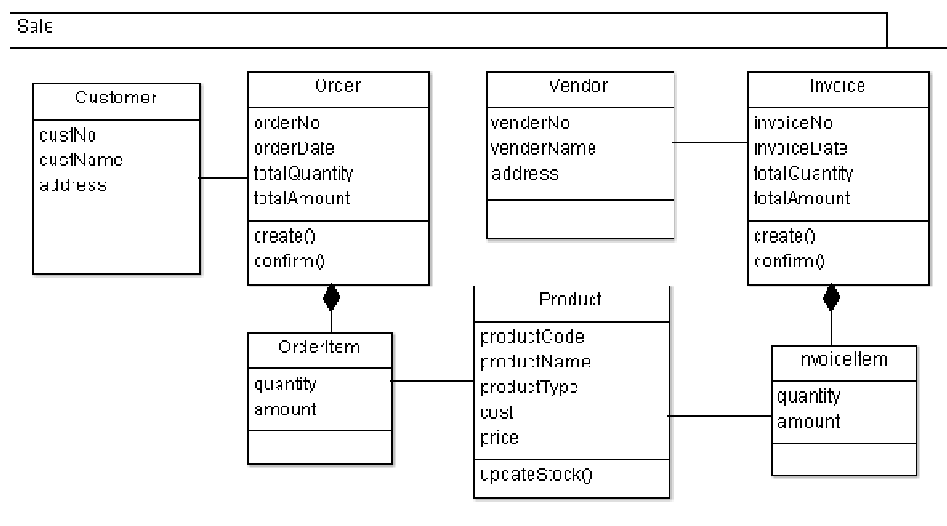


ภาพที่ ข.2 แผนภาพคลาสเชิงแนวคิดชุดที่ 1 ระบบงานใหม่

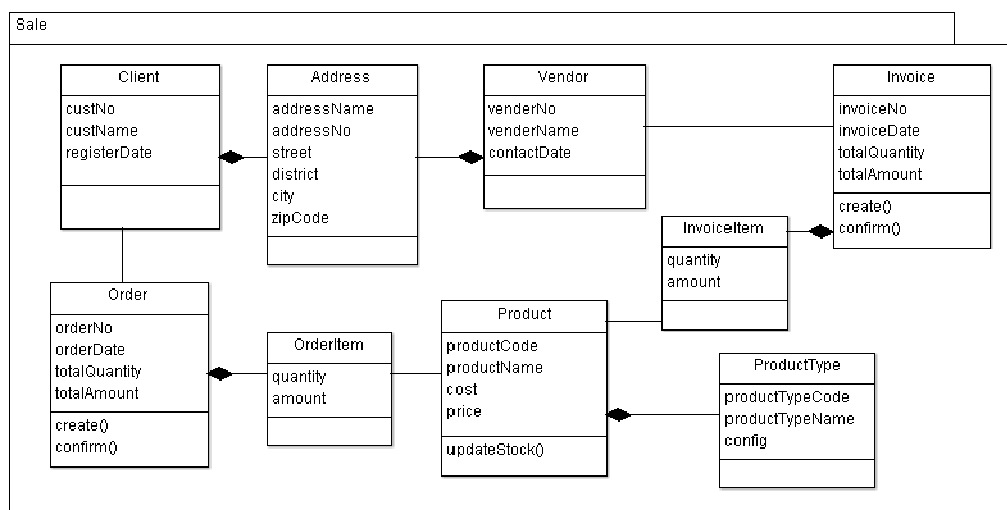
ตารางที่ ข.1 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 1

ลำดับ	เอนทิตีของภาพที่ ข.1	เอนทิตีของภาพที่ ข.2	การเปลี่ยนแปลง
1	CustomerManagement	ClientDirection	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
2	Address	Location	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
3	Person	Personality	เปลี่ยนชื่อ
4	Member	Association	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
5	Customer	Client	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
6	LoanInfo	LoanInfomation	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
7	addressNo	locationNo	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
8	addressName	locationName	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
9	interest	interestRate	เปลี่ยนชื่อ
10	contractCondition	agreementStatus	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
11	branch	division	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน

ชุดที่ 2



ภาพที่ ข.3 แผนภาพคลาสเชิงแนวคิดชุดที่ 2 ระบบงานเดิม

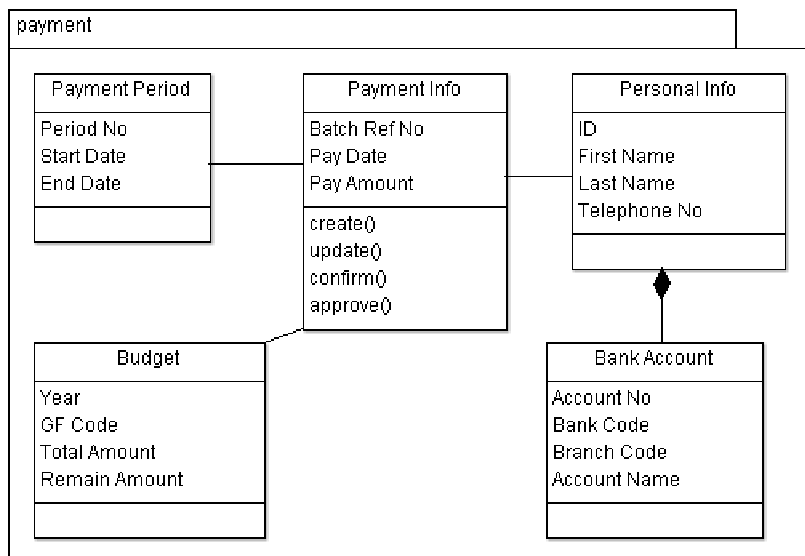


ภาพที่ ข.4 แผนภาพคลาสเชิงแนวคิดชุดที่ 2 ระบบงานใหม่

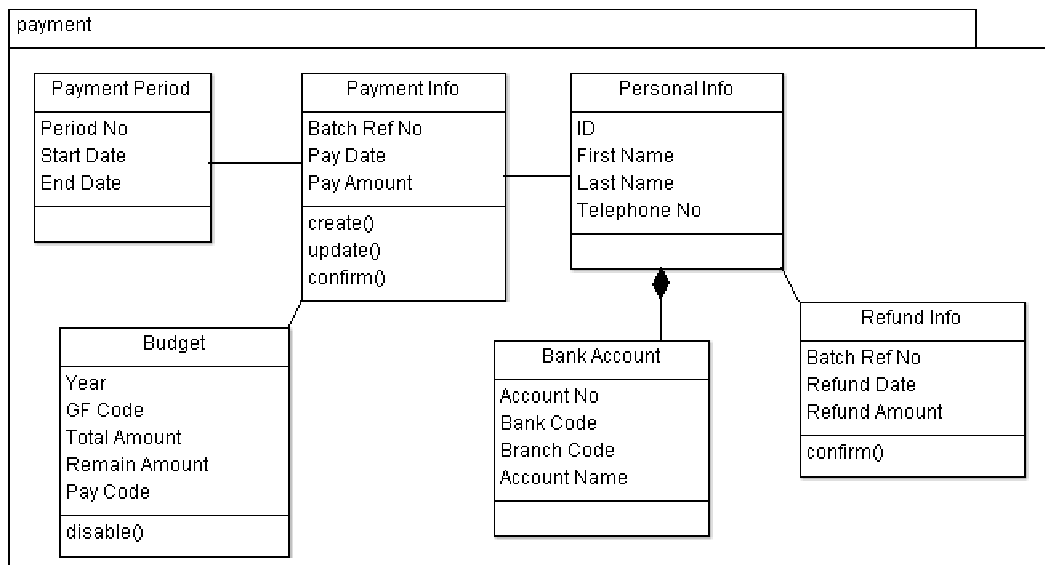
ตารางที่ ข.2 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 2

ลำดับ	เอนทิตีของภาพที่ ข.3	เอนทิตีของภาพที่ ข.4	การเปลี่ยนแปลง
1	Customer	Client	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
2	address ในคลาส Customer	Address	เปลี่ยนประเภทของเอนทิตี
3	address ในคลาส Vendor	Address	เปลี่ยนประเภทของเอนทิตี
4	productType	ProductType	เปลี่ยนประเภทของเอนทิตี
5		registerDate	เพิ่มเอนทิตี
6		addressName	เพิ่มเอนทิตี
7		addressNo	เพิ่มเอนทิตี
8		Street	เพิ่มเอนทิตี
9		district	เพิ่มเอนทิตี
10		city	เพิ่มเอนทิตี
11		zipCode	เพิ่มเอนทิตี
12		contractDate	เพิ่มเอนทิตี
13		productTypeCode	เพิ่มเอนทิตี
14		productTypeName	เพิ่มเอนทิตี
15		config	เพิ่มเอนทิตี

ชุดที่ 3



ภาพที่ ข.5 แผนภาพคลาสเชิงแนวคิดชุดที่ 3 ระบบงานเดิม

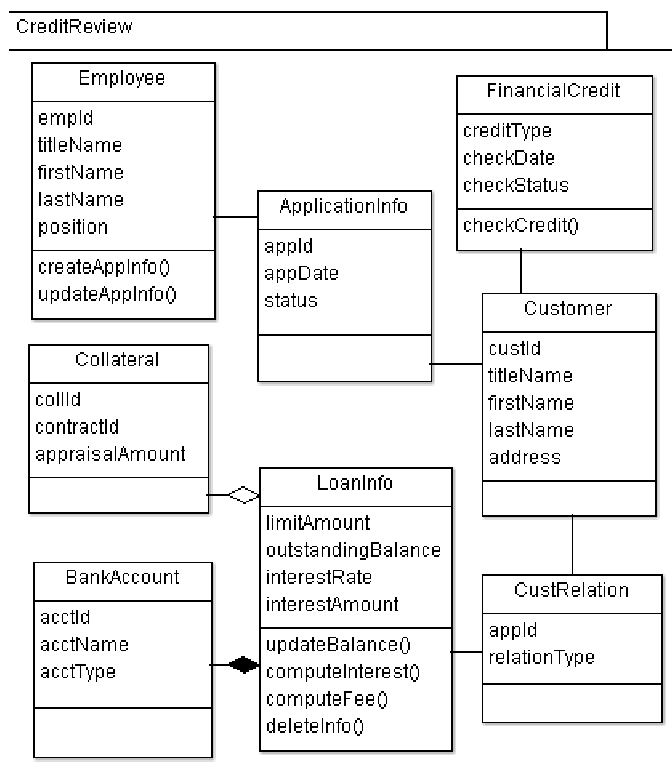


ภาพที่ ข.6 แผนภาพคลาสเชิงแนวคิดชุดที่ 3 ระบบงานใหม่

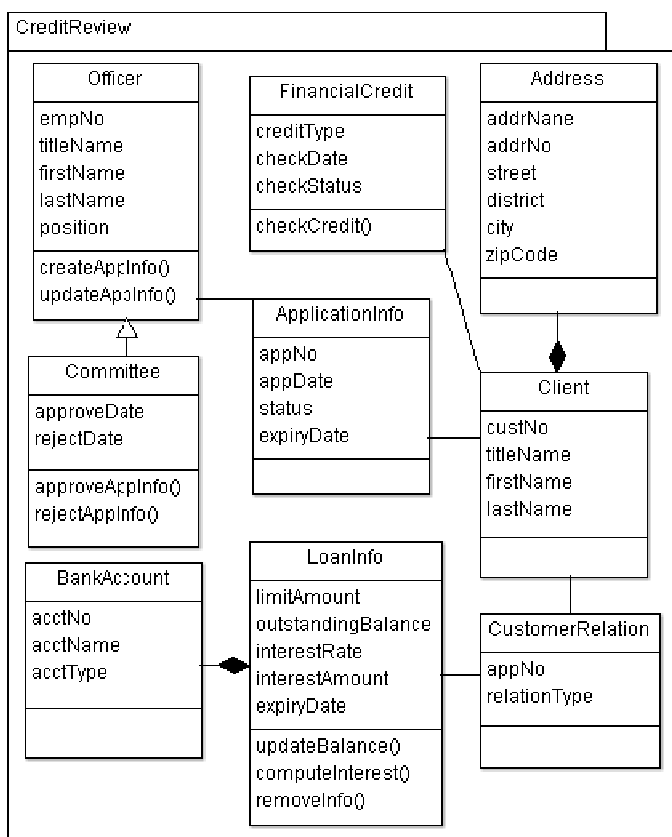
ตารางที่ ข.3 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 3

ลำดับ	เอนทิตีของภาพที่ ข.5	เอนทิตีของภาพที่ ข.6	การเปลี่ยนแปลง
1	approve()		ลบเอนทิตี
2		Pay Code	เพิ่มเอนทิตี
3		disable()	เพิ่มเอนทิตี
4		Refund Info	เพิ่มเอนทิตี
5		Batch Ref No	เพิ่มเอนทิตี
6		Refund Date	เพิ่มเอนทิตี
7		Refund Amount	เพิ่มเอนทิตี
8		confirm()	เพิ่มเอนทิตี

ชุดที่ 4



ภาพที่ ข.7 แผนภาพคลาสเชิงแนวคิดชุดที่ 4 ระบบงานเดิม



ภาพที่ ข.8 แผนภาพคลาสเชิงแนวคิดชุดที่ 4 ระบบงานใหม่

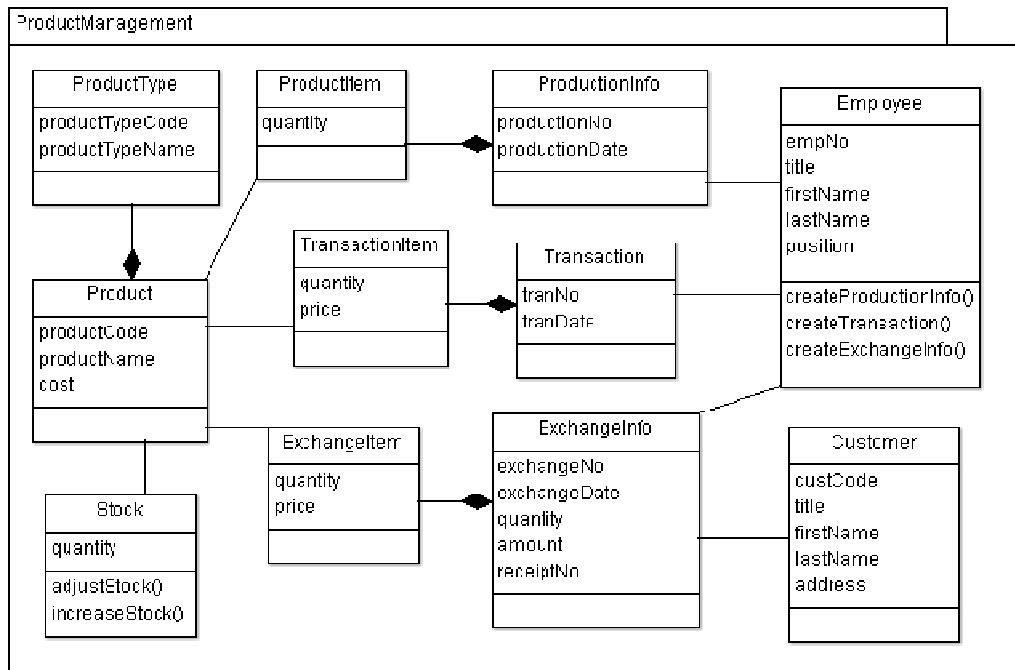
ตารางที่ ข.4 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 4

ลำดับ	เอนทิตีของภาพที่ ข.7	เอนทิตีของภาพที่ ข.8	การเปลี่ยนแปลง
1	CustRelation	CustomerRelation	เปลี่ยนชื่อ
2	Employee	Officer	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
3	Customer	Client	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
4	deleteInfo()	removeInfo()	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
5	acctId	acctNo	เปลี่ยนชื่อ
6	custId	custNo	เปลี่ยนชื่อ
7	collId		ลบเอนทิตี
8	contractId		ลบเอนทิตี

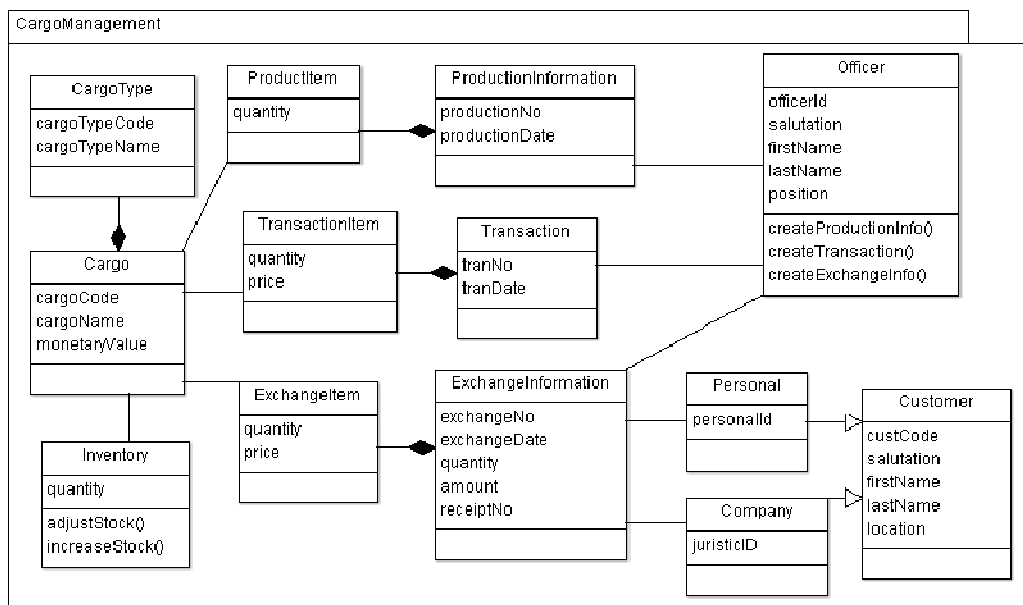
ตารางที่ ข.4 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 4 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ ข.7	เอนทิตีของภาพที่ ข.8	การเปลี่ยนแปลง
9	appraisalAmount		ลบเอนทิตี
10	appld		ลบเอนทิตี
11	appld		ลบเอนทิตี
12	empld		ลบเอนทิตี
13	computeFee()		ลบเอนทิตี
14	Collateral		ลบเอนทิตี
15		approveDate	เพิ่มเอนทิตี
16		rejectDate	เพิ่มเอนทิตี
17		addrNane	เพิ่มเอนทิตี
18		addrNo	เพิ่มเอนทิตี
19		street	เพิ่มเอนทิตี
20		district	เพิ่มเอนทิตี
21		city	เพิ่มเอนทิตี
22		zipCode	เพิ่มเอนทิตี
23		appNo	เพิ่มเอนทิตี
24		appNo	เพิ่มเอนทิตี
25		expiryDate	เพิ่มเอนทิตี
26		expiryDate	เพิ่มเอนทิตี
27		empNo	เพิ่มเอนทิตี
28		approveAppInfo()	เพิ่มเอนทิตี
29		rejectAppInfo()	เพิ่มเอนทิตี
30		Committee	เพิ่มเอนทิตี
31	address	Address	เปลี่ยนประเภทของเอนทิตี

ชุดที่ 5



ภาพที่ ข.9 แผนภาพคลาสเชิงแนวคิดชุดที่ 5 ระบบงานเดิม

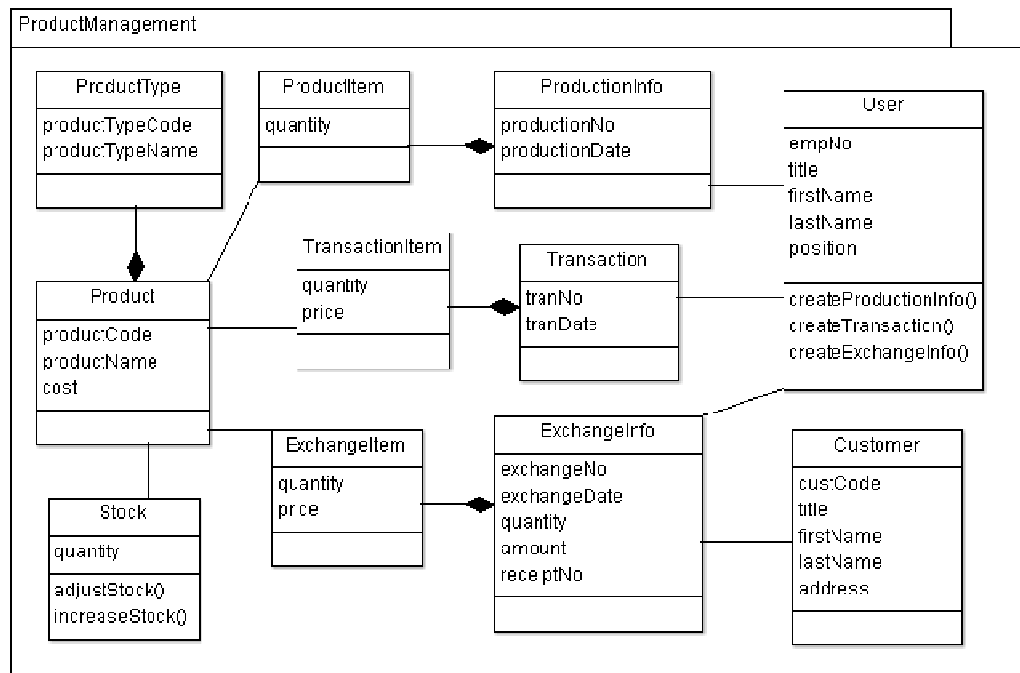


ภาพที่ ข.10 แผนภาพคลาสเชิงแนวคิดชุดที่ 5 ระบบงานใหม่

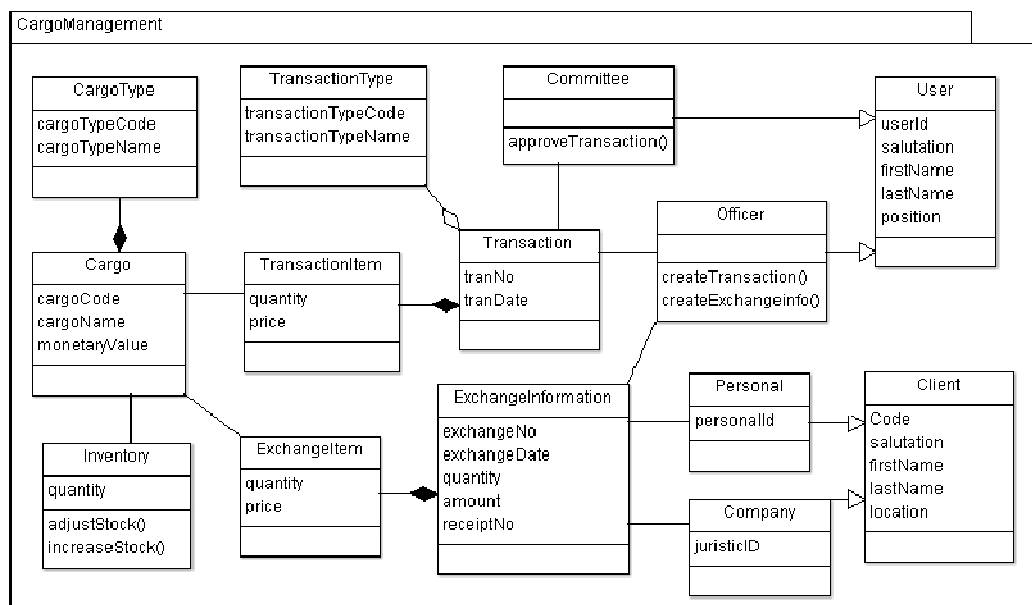
ตารางที่ ข.5 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 5

ลำดับ	เอนทิตีของภาพที่ ข.9	เอนทิตีของภาพที่ ข.10	การเปลี่ยนแปลง
1	ProductManagement	CargoManagement	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
2	ProductType	CargoType	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
3	Product	Cargo	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
4	Stock	Inventory	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
5	ProductionInfo	ProductionInformation	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
6	Employee	Officer	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
7	ExchangeInfo	ExchangeInformation	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
8	title	salutation	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
9	address	location	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
10	productTypeCode	cargoTypeCode	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
11	productTypeName	cargoTypeName	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
12	productCode	cargoCode	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
13	productName	cargoName	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
14	cost	monetaryValue	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
15	title	salutation	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
16	empNo		ลบเอนทิตี
17		personalId	เพิ่มเอนทิตี
18		juristicID	เพิ่มเอนทิตี
19		officerId	เพิ่มเอนทิตี
20		Personal	เพิ่มเอนทิตี
21		Company	เพิ่มเอนทิตี

ชุดที่ 6



ภาพที่ ข.11 แผนภาพคลาสเชิงแนวคิดชุดที่ 6 ระบบงานเดิม



ภาพที่ ข.12 แผนภาพคลาสเชิงแนวคิดชุดที่ 6 ระบบงานใหม่

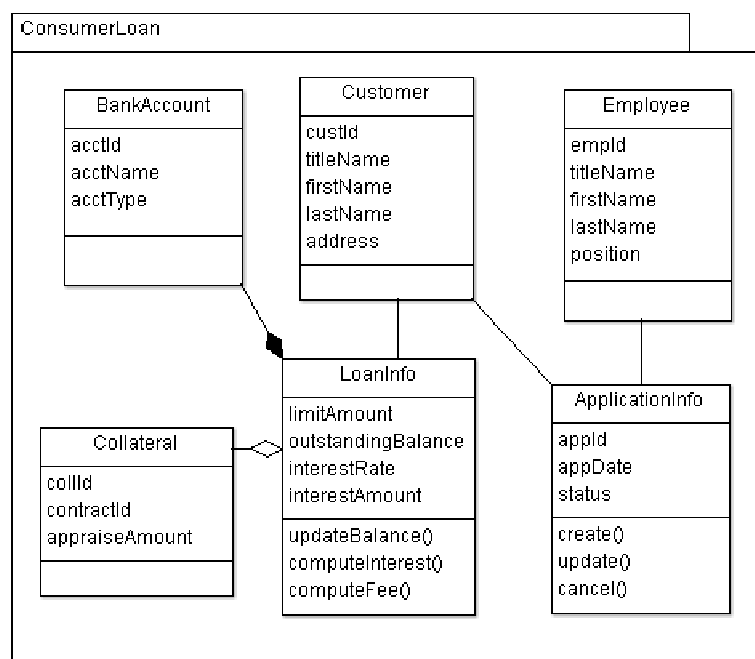
ตารางที่ ข.6 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 6

ลำดับ	เอนทิตีของภาพที่ ข.11	เอนทิตีของภาพที่ ข.12	การเปลี่ยนแปลง
1	ProductManagement	CargoManagement	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
2	ProductType	CargoType	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
3	Product	Cargo	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
4	Stock	Inventory	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
5	Customer	Client	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
6	ExchangeInfo	ExchangeInformation	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
7	title	salutation	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
8	productTypeCode	cargoTypeCode	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
9	productTypeName	cargoTypeName	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
10	productCode	cargoCode	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
11	productName	cargoName	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
12	cost	monetaryValue	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
13	title	Salutation	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
14	Address	location	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
15	productionNo		ลบเอนทิตี
16	productionDate		ลบเอนทิตี
17	quantity		ลบเอนทิตี
18	empNo		ลบเอนทิตี
19	custCode	Code	เปลี่ยนชื่อ
20	createProductionInfo()		ลบเอนทิตี
21	createTransaction()		ลบเอนทิตี
22	createExchangeInfo()		ลบเอนทิตี
23	ProductionInfo		ลบเอนทิตี
24	ProductItem		ลบเอนทิตี
25		personalId	เพิ่มเอนทิตี
26		juristicID	เพิ่มเอนทิตี
27		transactionTypeCode	เพิ่มเอนทิตี
28		transactionTypeName	เพิ่มเอนทิตี
29		userId	เพิ่มเอนทิตี

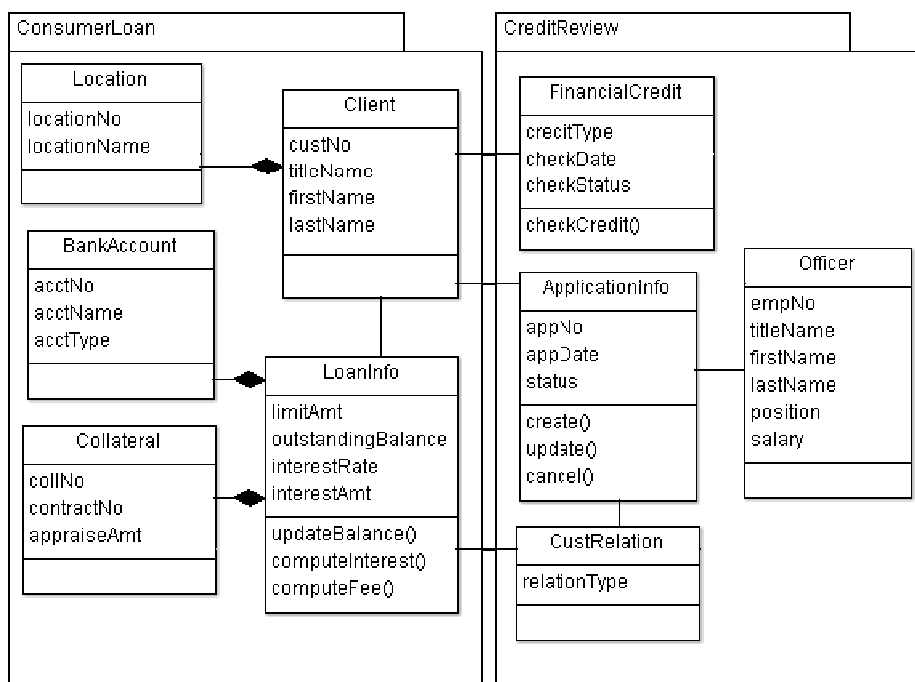
ตารางที่ ข.6 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 6 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ ข.11	เอนทิตีของภาพที่ ข.12	การเปลี่ยนแปลง
30		createTransaction()	เพิ่มเอนทิตี
31		createExchangeinfo()	เพิ่มเอนทิตี
32		approveTransaction()	เพิ่มเอนทิตี
33		Personal	เพิ่มเอนทิตี
34		Company	เพิ่มเอนทิตี
35		TransactionType	เพิ่มเอนทิตี
36		Officer	เพิ่มเอนทิตี
37		Committee	เพิ่มเอนทิตี

ชุดที่ 7



ภาพที่ ข.13 แผนภาพคลาสเชิงแนวคิดชุดที่ 7 ระบบงานเดิม



ภาพที่ ข.14 แผนภาพคลาสเชิงแนวคิดชุดที่ 7 ระบบงานใหม่

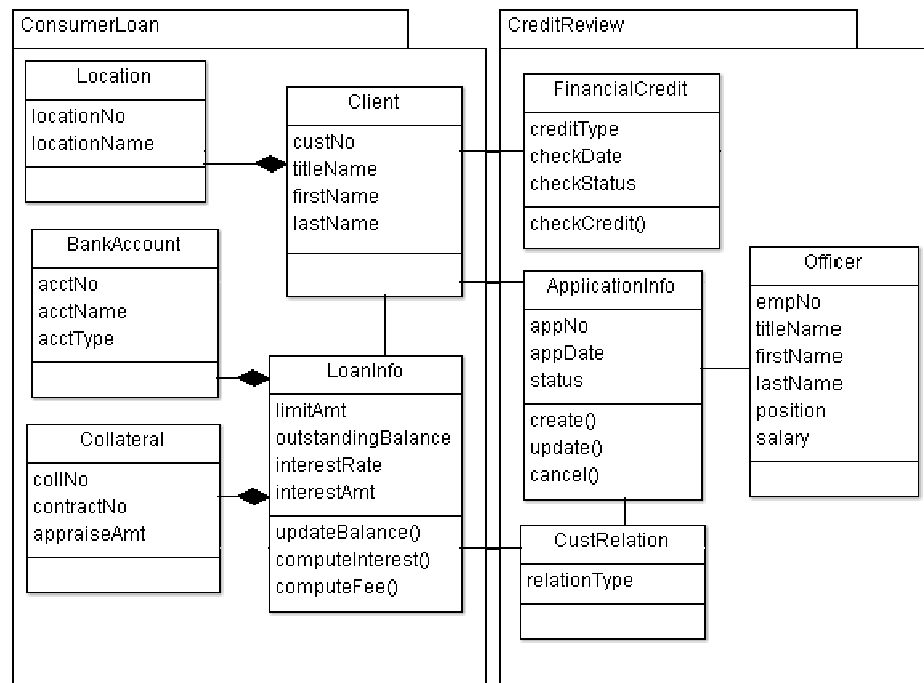
ตารางที่ ข.7 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 7

ลำดับ	เอนทิตีของภาพที่ ข.13	เอนทิตีของภาพที่ ข.14	การเปลี่ยนแปลง
1	Customer	Client	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
2	ApplicationInfo	ApplicationInfo	ย้ายเอนทิตี
3	acctId	acctNo	เปลี่ยนชื่อ
4	limitAmount	limitAmt	เปลี่ยนชื่อ
5	interestAmount	interestAmt	เปลี่ยนชื่อ
6	collId	collNo	เปลี่ยนชื่อ
7	contractId	contractNo	เปลี่ยนชื่อ
8	appraiseAmount	appraiseAmt	เปลี่ยนชื่อ
9	custId	custNo	เปลี่ยนชื่อ
10	titleName	titleName	ย้ายเอนทิตี
11	firstName	firstName	ย้ายเอนทิตี
12	lastName	lastName	ย้ายเอนทิตี
13	position	position	ย้ายเอนทิตี

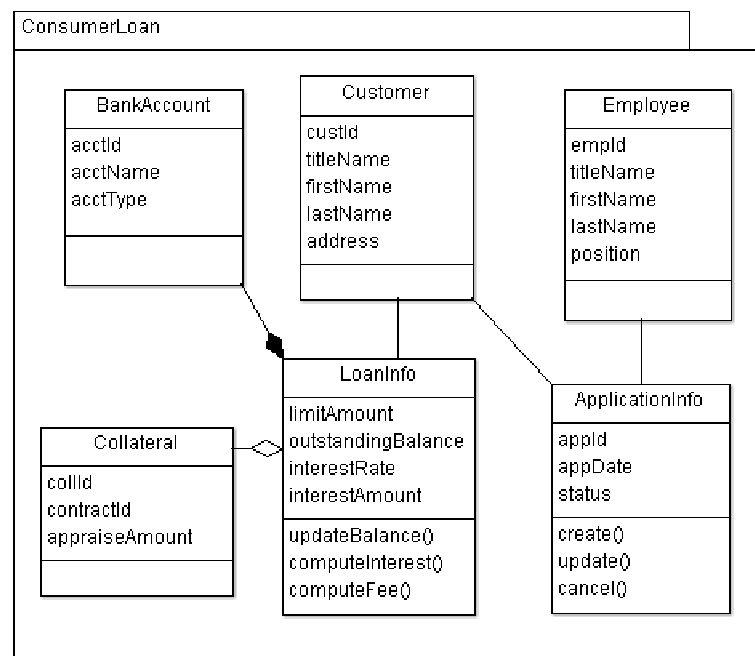
ตารางที่ ข.7 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 7 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ ข.13	เอนทิตีของภาพที่ ข.14	การเปลี่ยนแปลง
14	empld		ลบเอนทิตี
15	appld		ลบเอนทิตี
16	Employee		ลบเอนทิตี
17		CreditReview	เพิ่มเอนทิตี
18		creditType	เพิ่มเอนทิตี
19		checkDate	เพิ่มเอนทิตี
20		checkStatus	เพิ่มเอนทิตี
21		relationType	เพิ่มเอนทิตี
22		empNo	เพิ่มเอนทิตี
23		salary	เพิ่มเอนทิตี
24		locationNo	เพิ่มเอนทิตี
25		locationName	เพิ่มเอนทิตี
26		appNo	เพิ่มเอนทิตี
27		checkCredit()	เพิ่มเอนทิตี
28		FinancialCredit	เพิ่มเอนทิตี
29		CustRelation	เพิ่มเอนทิตี
30		Officer	เพิ่มเอนทิตี
31	address	Location	เปลี่ยนประเภทของเอนทิตี

ชุดที่ 8



ภาพที่ ข.15 แผนภาพคลาสเชิงแนวคิดชุดที่ 8 ระบบงานเดิม

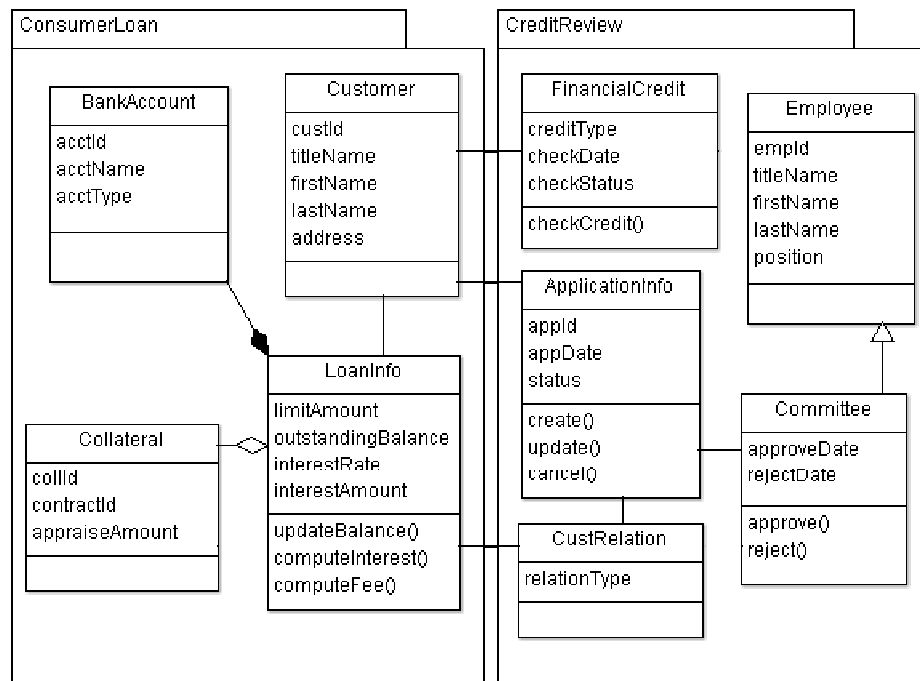


ภาพที่ ข.16 แผนภาพคลาสเชิงแนวคิดชุดที่ 8 ระบบงานใหม่

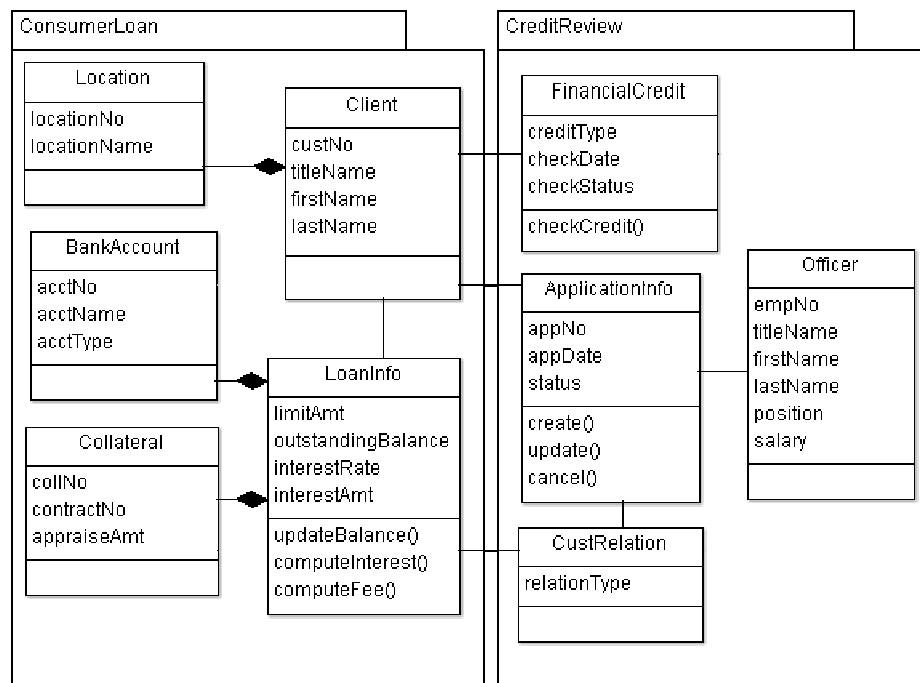
ตารางที่ ข.8 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 8

ลำดับ	เอนทิตีของภาพที่ ข.15	เอนทิตีของภาพที่ ข.16	การเปลี่ยนแปลง
1	Client	Customer	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
2	ApplicationInfo	ApplicationInfo	ย้ายเอนทิตี
3	acctNo	acctId	เปลี่ยนชื่อ
4	limitAmt	limitAmount	เปลี่ยนชื่อ
5	interestAmt	interestAmount	เปลี่ยนชื่อ
6	collNo	collId	เปลี่ยนชื่อ
7	contractNo	contractId	เปลี่ยนชื่อ
8	appraiseAmt	appraiseAmount	เปลี่ยนชื่อ
9	custNo	custId	เปลี่ยนชื่อ
10	CreditReview		ลบเอนทิตี
11	creditType		ลบเอนทิตี
12	checkDate		ลบเอนทิตี
13	checkStatus		ลบเอนทิตี
14	relationType		ลบเอนทิตี
15	empNo	empld	เปลี่ยนชื่อ
16	salary		ลบเอนทิตี
17	locationNo		ลบเอนทิตี
18	locationName		ลบเอนทิตี
19	appNo		ลบเอนทิตี
20	checkCredit()		ลบเอนทิตี
21	FinancialCredit		ลบเอนทิตี
22	CustRelation		ลบเอนทิตี
23	Officer		ลบเอนทิตี
24		appld	เพิ่มเอนทิตี
25		Employee	เพิ่มเอนทิตี
26	Location	Address	เปลี่ยนประเภทของเอนทิตี

ชุดที่ 9



ภาพที่ ข.17 แผนภาพคลาสเชิงแนวคิดชุดที่ 9 ระบบงานเดิม

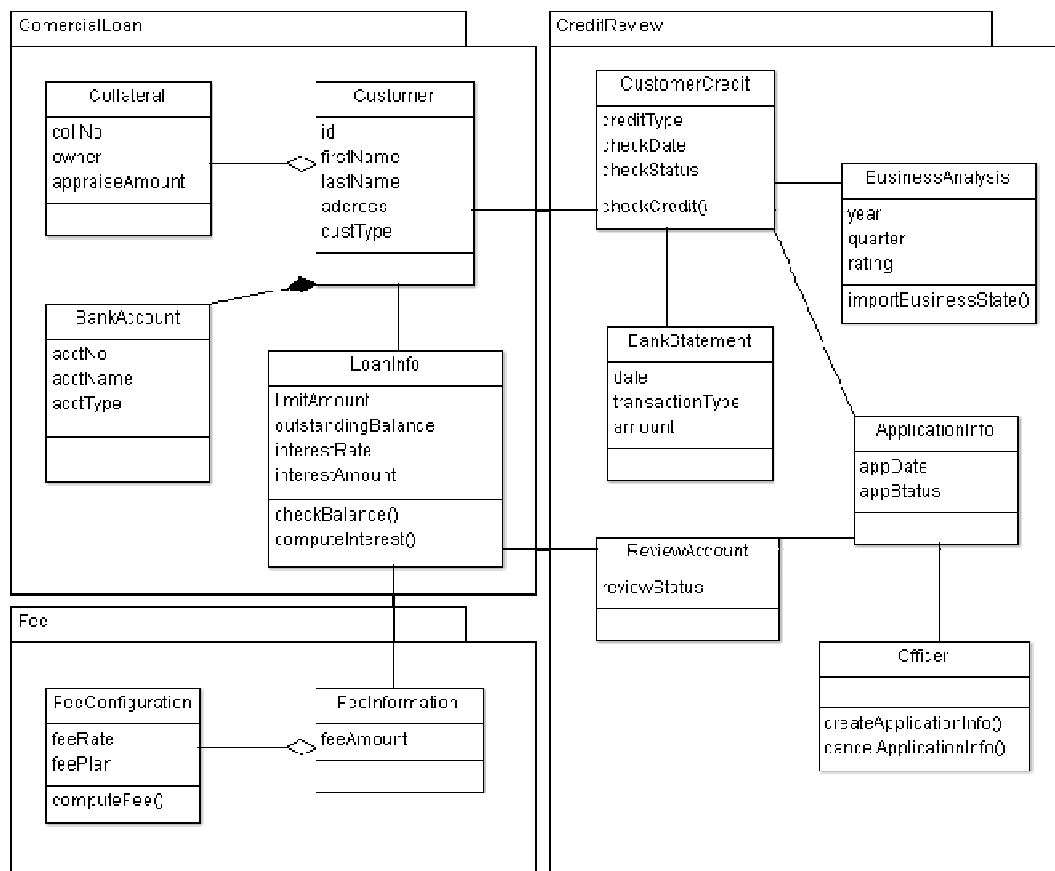


ภาพที่ ข.18 แผนภาพคลาสเชิงแนวคิดชุดที่ 9 ระบบงานใหม่

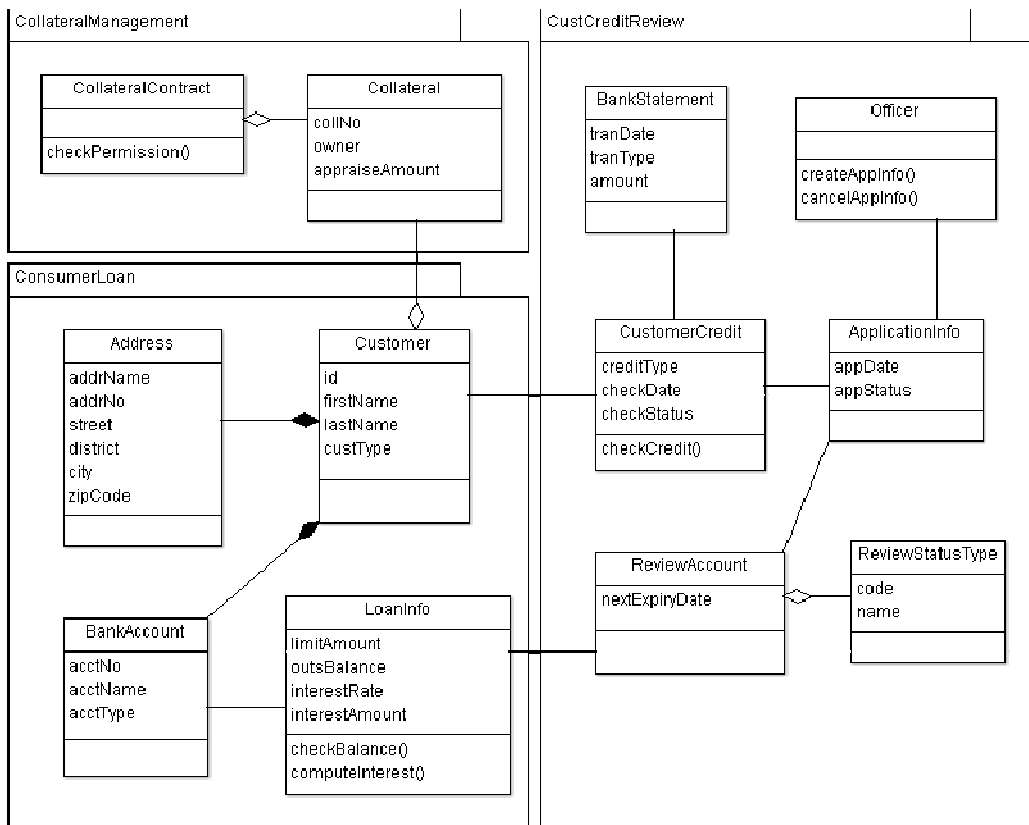
ตารางที่ ข.9 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 9

ลำดับ	เอนทิตีของภาพที่ ข.17	เอนทิตีของภาพที่ ข.18	การเปลี่ยนแปลง
1	Customer	Client	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
2	Employee	Officer	เปลี่ยนชื่อเป็นคำที่มีความหมายคล้ายกัน
3	acctId	acctNo	เปลี่ยนชื่อ
4	limitAmount	limitAmt	เปลี่ยนชื่อ
5	interestAmount	interestAmt	เปลี่ยนชื่อ
6	collId	collNo	เปลี่ยนชื่อ
7	contractId	contractNo	เปลี่ยนชื่อ
8	appraiseAmount	appraiseAmt	เปลี่ยนชื่อ
9	custId	custNo	เปลี่ยนชื่อ
10	approveDate		ลบเอนทิตี
11	rejectDate		ลบเอนทิตี
12	appld		ลบเอนทิตี
13	empld		ลบเอนทิตี
14	approve()		ลบเอนทิตี
15	reject()		ลบเอนทิตี
16	Committee		ลบเอนทิตี
17		locationNo	เพิ่มเอนทิตี
18		locationName	เพิ่มเอนทิตี
19		appNo	เพิ่มเอนทิตี
20		empNo	เพิ่มเอนทิตี
21		salary	เพิ่มเอนทิตี
22	address	Location	เปลี่ยนประเภทของเอนทิตี

ชุดที่ 10



ภาพที่ ข.19 แผนภาพคลาสเชิงแนวคิดชุดที่ 10 ระบบงานเดิม



ภาพที่ ข.20 แผนภาพคลาสเชิงแนวคิดชุดที่ 11 ระบบงานใหม่

ตารางที่ ข.10 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 10

ลำดับ	เอนทิตีของภาพที่ ข.19	เอนทิตีของภาพที่ ข.20	การเปลี่ยนแปลง
1	ComercialLoan	ConsumerLoan	เปลี่ยนชื่อ
2	CreditReview	CustCreditReview	เปลี่ยนชื่อ
3	Collateral	Collateral	ย้ายเอนทิตี
4	outstandingBalance	outsBalance	เปลี่ยนชื่อ
5	createApplicationInfo()	createAppInfo()	เปลี่ยนชื่อ
6	cancelApplicationInfo()	cancelAppInfo()	เปลี่ยนชื่อ
7	date	tranDate	เปลี่ยนชื่อ
8	transactionType	tranType	เปลี่ยนชื่อ
9	Fee		ลบเอนทิตี
10	feeRate		ลบเอนทิตี
11	feePlan		ลบเอนทิตี
12	feeAmount		ลบเอนทิตี

ตารางที่ ข.10 การเปลี่ยนแปลงแผนภาพคลาสเชิงแนวคิดชุดที่ 10 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ ข.19	เอนทิตีของภาพที่ ข.20	การเปลี่ยนแปลง
13	year		ลบเอนทิตี
14	quarter		ลบเอนทิตี
15	computeFee()		ลบเอนทิตี
16	importBusinessState()		ลบเอนทิตี
17	FeeConfiguration		ลบเอนทิตี
18	FeeInformation		ลบเอนทิตี
19	BusinessAnalysis		ลบเอนทิตี
20	rating		ลบเอนทิตี
21		CollateralManagement	เพิ่มเอนทิตี
22		addrName	เพิ่มเอนทิตี
23		addrNo	เพิ่มเอนทิตี
24		street	เพิ่มเอนทิตี
25		district	เพิ่มเอนทิตี
26		city	เพิ่มเอนทิตี
27		zipCode	เพิ่มเอนทิตี
28		code	เพิ่มเอนทิตี
29		name	เพิ่มเอนทิตี
30		nextExpiryDate	เพิ่มเอนทิตี
31		checkPermission()	เพิ่มเอนทิตี
32		CollateralContract	เพิ่มเอนทิตี
33	address	Address	เปลี่ยนประเภทของเอนทิตี
34	reviewStatus	ReviewStatusType	เปลี่ยนประเภทของเอนทิตี

ภาคผนวก ค

ผลลัพธ์ที่ได้จากการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดของระบบงานจริง

ผลลัพธ์ที่ได้จากการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดของระบบงานจริงโดยใช้ S-UMLDiff, UMLDiff และผลจากการพิจารณาของผู้เชี่ยวชาญคนที่ 3 แสดงดังตารางต่อไปนี้ สำหรับตารางที่ ค.1 และ ค.2 รายการที่แสดงด้วยตัวเอียง คือ ผลลัพธ์ที่ไม่ตรงกันกับผู้เชี่ยวชาญ และรายการที่แสดงด้วยตัวหนา คือ ผลลัพธ์ที่ตรงกันกับผู้เชี่ยวชาญ

ตารางที่ ค.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ S-UMLDiff

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง	คะแนนความคล้ายคลึงเชิงความหมาย
1	Transaction	Transaction	match	0	0
2	Inventory	Inventory	match	0	0
3	Shipping	Transportation	smatch	1	1
4	<i>Master</i>	<i>Center</i>	<i>rename</i>	<i>0.705882</i>	<i>0</i>
5	Order	Order	match	0	0
6	TransactionType	TransactionType	match	0	0
7	Treasury	Treasury	match	0	0
8	Product	Product	match	0	0
9	Package	Package	match	0	0
10	Route	Route	match	0	0
11	Driver	Driver	match	0	0
12	Administrator	Administrator	match	0	0
13	AO	AO	match	0	0
14	SAO	SAO	match	0	0
15	DLA	DLA	match	0	0
16	CenterUser	CenterUser	match	0	0
17	Member	Member	match	0	0
18	InventoryUser	InventoryUser	match	0	0

ตารางที่ ค.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ S-UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง	คะแนน ความคล้ายคลึง	คะแนน ความคล้ายคลึง เชิงความหมาย
19	User	User	match	0	0
20	OrderDetail	OrderItem	smatch	1	1
21	Separation	Sorting	rename	0.823867	0
22	Adjustment	StockAdjustment	rename	0.601732	0
23	AdjustmentDetail	AdjustmentItem	smatch	1	1
24	Category	ProductType	smatch	0.847075	0.909091
25	StockType	InventoryType	smatch	0.75	1
26	Stock	Inventory	smatch	0.875	1
27	CarSchedule	Schedule	rename	0.570478	0
28	Car	Vehicle	smatch	0.669231	0.9
29	CarType	VehicleType	smatch	0.709494	0.95
30	Shipment	Delivery	smatch	0.726455	0.869565
31	DivisionManager	SectionManager	smatch	0.75	1
32	Employee	Officer	smatch	0.55	0.75
33	transactionDate	transactionDate	match	0	0
34	fromUnit	fromUnit	match	0	0
35	toUnit	toUnit	match	0	0
36	totalAmount	totalAmount	match	0	0
37	status	status	match	0	0
38	transactionTypeId	transactionTypeId	match	0	0
39	transactionTypeNa me	transactionTypeNa me	match	0	0
40	totalAmount	totalAmount	match	0	0
41	status	status	match	0	0
42	productName	productName	match	0	0
43	status	status	match	0	0

ตารางที่ ค.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ S-UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง	คะแนน ความคล้ายคลึง	คะแนน เชิงความหมาย
44	name	name	match	0	0
45	routeType	routeType	match	0	0
46	driverId	driverId	match	0	0
47	title	title	match	0	0
48	firstName	firstName	match	0	0
49	lastName	lastName	match	0	0
50	sex	sex	match	0	0
51	birthDate	birthDate	match	0	0
52	licenseId	licenseId	match	0	0
53	createOrder()	createOrder()	match	0	0
54	updateOrder()	updateOrder()	match	0	0
55	cancelOrder()	cancelOrder()	match	0	0
56	confirmOrder()	confirmOrder()	match	0	0
57	Cancel ConfirmOrder()	Cancel ConfirmOrder()	match	0	0
58	approveOrder()	approveOrder()	match	0	0
59	Cancel ApproveOrder()	Cancel ApproveOrder()	match	0	0
60	rejectOrder()	rejectOrder()	match	0	0
61	Approve Shipment()	Approve Shipment()	match	0	0
62	CancelApprove Shipment()	CancelApprove Shipment()	match	0	0
63	confirmTreasury()	confirmTreasury()	match	0	0
64	Cancel ConfirmTreasury()	Cancel ConfirmTreasury()	match	0	0
65	userId	userId	match	0	0

ตารางที่ ค.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ S-UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง	คะแนน ความคล้ายคลึง	คะแนน เชิงความหมาย
66	title	title	match	0	0
67	firstName	firstName	match	0	0
68	lastName	lastName	match	0	0
69	sex	sex	match	0	0
70	birthDate	birthDate	match	0	0
71	position	position	match	0	0
72	quantity	quantity	match	0	0
73	amount	amount	match	0	0
74	accountDate	accountDate	match	0	0
75	transactionDate	transactionDate	match	0	0
76	totalAmount	totalAmount	match	0	0
77	status	status	match	0	0
78	transactionDate	transactionDate	match	0	0
79	totalAmount	totalAmount	match	0	0
80	status	status	match	0	0
81	availableAmount	availableAmount	match	0	0
82	reserveAmount	reserveAmount	match	0	0
83	productFormId	productFormId	match	0	0
84	botTypeId	botTypeId	match	0	0
85	name	name	match	0	0
86	status	status	match	0	0
87	totalAmount	totalAmount	match	0	0
88	totalQuantity	totalQuantity	match	0	0
89	workingDate	workingDate	match	0	0
90	transactinDate	transactinDate	match	0	0
91	condition	condition	match	0	0
92	fromUnit	fromUnit	match	0	0

ตารางที่ ค.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ S-UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง	คะแนนความคล้ายคลึงเชิงความหมาย
93	toUnit	toUnit	match	0	0
94	totalAmount	totalAmount	match	0	0
95	status	status	match	0	0
96	orderId	centerId	rename	0.9	0
97	treasuryId	treasuryNumber	smatch	0.95	0.95
98	productId	productNo	rename	0.636364	0
99	packageId	packageNo	rename	0.636364	0
100	quantity	quantityPerUnit	rename	0.533333	0
101	routeId	routeNo	rename	0.555556	0
102	configData()	configSystemData()	rename	0.625	0
103	createAdjustment()	createStock Adjustment()	smatch	1	1
104	updateAdjustment()	updateStock Adjustment()	smatch	1	1
105	cancelAdjustment()	cancelStock Adjustment()	smatch	1	1
106	createSeparation()	createSorting()	smatch	0.97619	0.97619
107	updateSeparation()	updateSorting()	smatch	0.97619	0.97619
108	cancelSeparation()	cancelSorting()	smatch	0.97619	0.97619
109	Approve Adjustment()	approveStock Adjustment()	smatch	1	1
110	cancelApprove Adjustment()	cancelApprove StockAdjustment()	rename	0.821429	0
111	separationId	sortingNumber	smatch	0.92619	0.92619
112	stockTypeId	inventoryTypeId	smatch	1	1
113	stockTypeName	inventoryTypeName	smatch	1	1
114	updateStock()	updateProduct()	smatch	0.970588	0.970588
115	carScheduleId	scheduleNo	rename	0.533333	0

ตารางที่ ค.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ S-UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง	คะแนนความคล้ายคลึงเชิงความหมาย
116	<i>overrideCar</i> <i>Schedule</i>	<i>overrideSchedule</i>	<i>smatch</i>	0.912281	0.912281
117	<i>carId</i>	<i>vehicleId</i>	<i>smatch</i>	0.95	0.95
118	<i>ownerUnitId</i>	<i>ownerCenterId</i>	<i>smatch</i>	0.912281	0.912281
119	<i>carTypeId</i>	<i>vehicleTypeId</i>	<i>smatch</i>	0.966667	0.966667
120	<i>carTypeName</i>	<i>vehicleTypeName</i>	<i>smatch</i>	0.966667	0.966667
121	<i>autoMapRoute()</i>	<i>autoRoute()</i>	<i>rename</i>	0.75	0
122	<i>autoMapCar()</i>	<i>autoCar()</i>	<i>rename</i>	0.7	0
123	<i>createShipment()</i>	<i>createDelivery()</i>	<i>smatch</i>	0.934783	0.934783
124	<i>confirmAdjustment()</i>	<i>confirmStockAdjustment()</i>	<i>smatch</i>	1	1
125	<i>cancelConfirmAdjustment()</i>	<i>cancelConfirmStockAdjustment()</i>	<i>rename</i>	0.821429	0
126	<i>unitId</i>		<i>remove</i>	0	0
127	<i>unitId</i>		<i>remove</i>	0	0
128	<i>unitId</i>		<i>remove</i>	0	0
129	<i>unitId</i>		<i>remove</i>	0	0
130	<i>unitId</i>		<i>remove</i>	0	0
131	<i>unitId</i>		<i>remove</i>	0	0
132	<i>employedDate</i>		<i>remove</i>	0	0
133	<i>adjustId</i>		<i>remove</i>	0	0
134	<i>categoryId</i>		<i>remove</i>	0	0
135	<i>timeTable</i>		<i>remove</i>	0	0
136	<i>shipmentId</i>		<i>remove</i>	0	0
137	<i>defineUserAndPermission()</i>		<i>remove</i>	0	0
138	<i>cancelShipment()</i>		<i>remove</i>	0	0

ตารางที่ ค.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ S-UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง	คะแนน ความคล้ายคลึง	คะแนน ความคล้ายคลึง เชิงความหมาย
139		feeld	add	0	0
140		feeName	add	0	0
141		feePlan	add	0	0
142		feeTypeId	add	0	0
143		feeTypeName	add	0	0
144		locationNo	add	0	0
145		locationName	add	0	0
146		street	add	0	0
147		district	add	0	0
148		city	add	0	0
149		zipCode	add	0	0
150		locTypeId	add	0	0
151		locTypeName	add	0	0
152		<i>orderNumber</i>	<i>add</i>	<i>0</i>	<i>0</i>
153		invoiceNo	add	0	0
154		taxId	add	0	0
155		systemType	add	0	0
156		<i>centerId</i>	<i>add</i>	<i>0</i>	<i>0</i>
157		<i>centerId</i>	<i>add</i>	<i>0</i>	<i>0</i>
158		<i>centerId</i>	<i>add</i>	<i>0</i>	<i>0</i>
159		<i>centerId</i>	<i>add</i>	<i>0</i>	<i>0</i>
160		<i>centerId</i>	<i>add</i>	<i>0</i>	<i>0</i>
161		<i>centerId</i>	<i>add</i>	<i>0</i>	<i>0</i>
162		<i>startWorkingDate</i>	<i>add</i>	<i>0</i>	<i>0</i>
163		cashOwner	add	0	0

ตารางที่ ค.1 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ S-UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง	คะแนน ความคล้ายคลึง	คะแนน ความคล้ายคลึง เชิงความหมาย
164		stockAdjust Number	add	0	0
165		productTypeNo	add	0	0
166		time	add	0	0
167		deliveryNumber	add	0	0
168		computeFee()	add	0	0
169		createOrder()	add	0	0
170		updateOrder()	add	0	0
171		cancelOrder()	add	0	0
172		createSorting()	add	0	0
173		updateSorting()	add	0	0
174		cancelSorting()	add	0	0
175		manageUser()	add	0	0
176		manage Permission()	add	0	0
177		updateDelivery()	add	0	0
178		ExternalSorting	add	0	0
179		InternalSorting	add	0	0
180		FeeType	add	0	0
181		DataConfiguration	add	0	0
182		Customer	add	0	0
183		Location	add	0	0
184		LocationType	add	0	0
185		Guest	add	0	0
186	fee	Fee	changetype	1	1

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง
1	Virtual Root	Virtual Root	match	0
2	Transaction	Transaction	match	0
3	Inventory	Inventory	match	0
4	Order	Order	match	0
5	TransactionType	TransactionType	match	0
6	Treasury	Treasury	match	0
7	Product	Product	match	0
8	Package	Package	match	0
9	<i>OrderDetail</i>	<i>OrderItem</i>	<i>rename</i>	<i>0.595238</i>
10	Adjustment	StockAdjustment	rename	0.601732
11	<i>AdjustmentDetail</i>	<i>AdjustmentItem</i>	<i>rename</i>	<i>0.696594</i>
12	<i>Route</i>	<i>Route</i>	<i>move</i>	<i>0.833333</i>
13	<i>Driver</i>	<i>Driver</i>	<i>move</i>	<i>0.944444</i>
14	<i>Administrator</i>	<i>Administrator</i>	<i>move</i>	<i>0.75</i>
15	AO	AO	<i>move</i>	1
16	SAO	SAO	<i>move</i>	1
17	DLA	DLA	<i>move</i>	1
18	<i>CenterUser</i>	<i>CenterUser</i>	<i>move</i>	<i>0.785714</i>
19	<i>Member</i>	<i>Member</i>	<i>move</i>	<i>0.75</i>
20	<i>InventoryUser</i>	<i>InventoryUser</i>	<i>move</i>	<i>0.833333</i>
21	<i>User</i>	<i>User</i>	<i>move</i>	<i>0.909091</i>
22	transactionDate	transactionDate	match	0
23	fromUnit	fromUnit	match	0
24	toUnit	toUnit	match	0
25	totalAmount	totalAmount	match	0

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง	คะแนน ความคล้ายคลึง
26	status	status	match	0
27	transactionTypeIcd	transactionTypeIcd	match	0
28	transactionTypeName	transactionTypeName	match	0
29	totalAmount	totalAmount	match	0
30	status	status	match	0
31	productName	productName	match	0
32	status	status	match	0
33	name	name	match	0
34	quantity	quantity	match	0
35	amount	amount	match	0
36	transactionDate	transactionDate	match	0
37	totalAmount	totalAmount	match	0
38	status	status	match	0
39	availableAmount	availableAmount	match	0
40	reserveAmount	reserveAmount	match	0
41	routeType	routeType	match	0
42	driverId	driverId	match	0
43	title	title	match	0
44	firstName	firstName	match	0
45	lastName	lastName	match	0
46	sex	sex	match	0
47	birthDate	birthDate	match	0
48	licenseId	licenseId	match	0
49	createOrder()	createOrder()	match	0
50	updateOrder()	updateOrder()	match	0

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง
51	cancelOrder()	cancelOrder()	match	0
52	confirmOrder()	confirmOrder()	match	0
53	cancelConfirmOrder()	cancelConfirmOrder()	match	0
54	approveOrder()	approveOrder()	match	0
55	cancelApproveOrder()	cancelApproveOrder()	match	0
56	rejectOrder()	rejectOrder()	match	0
57	approveShipment()	approveShipment()	match	0
58	cancelApproveShipment()	cancelApproveShipment()	match	0
59	confirmTreasury()	confirmTreasury()	match	0
60	cancelConfirmTreasury()	cancelConfirmTreasury()	match	0
61	userId	userId	match	0
62	title	title	match	0
63	firstName	firstName	match	0
64	lastName	lastName	match	0
65	sex	sex	match	0
66	birthDate	birthDate	match	0
67	position	position	match	0
68	<i>treasuryId</i>	<i>treasuryNumber</i>	<i>rename</i>	0.5
69	productId	productNo	rename	0.636364
70	packageId	packageNo	rename	0.636364
71	quantity	quantityPerUnit	rename	0.533333
72	routeId	routeNo	rename	0.555556
73	configData()	configSystemData()	rename	0.625
74	createAdjustment()	createStockAdjustment()	rename	0.761905

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง
75	updateAdjustment()	updateStock Adjustment()	rename	0.761905
76	cancelAdjustment()	cancelStock Adjustment()	rename	0.761905
77	createSeparation()	createSorting()	rename	0.611111
78	updateSeparation()	updateSorting()	rename	0.611111
79	cancelSeparation()	cancelSorting()	rename	0.611111
80	approveAdjustment()	approveStock Adjustment()	rename	0.772727
81	cancelApprove Adjustment()	cancelApprove StockAdjustment()	rename	0.821429
82	workingDate	workingDate	move	0.7
83	Shipping		remove	0
84	Master		remove	0
85	carScheduleId		remove	0
86	timeTable		remove	0
87	overrideCarSchedule		remove	0
88	carId		remove	0
89	ownerUnitId		remove	0
90	carTypeId		remove	0
91	carTypeName		remove	0
92	shipmentId		remove	0
93	transactinDate		remove	0
94	unitId		remove	0
95	unitId		remove	0
96	unitId		remove	0
97	unitId		remove	0
98	unitId		remove	0

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง	คะแนน ความคล้ายคลึง
99	<i>unitId</i>		<i>remove</i>	0
100	<i>condition</i>		<i>remove</i>	0
101	<i>fromUnit</i>		<i>remove</i>	0
102	<i>toUnit</i>		<i>remove</i>	0
103	<i>totalAmount</i>		<i>remove</i>	0
104	<i>totalAmount</i>		<i>remove</i>	0
105	<i>totalAmount</i>		<i>remove</i>	0
106	<i>status</i>		<i>remove</i>	0
107	<i>status</i>		<i>remove</i>	0
108	<i>status</i>		<i>remove</i>	0
109	<i>fee</i>		<i>remove</i>	0
110	<i>separationId</i>		<i>remove</i>	0
111	<i>accountDate</i>		<i>remove</i>	0
112	<i>transactionDate</i>		<i>remove</i>	0
113	<i>categoryId</i>		<i>remove</i>	0
114	<i>productFormId</i>		<i>remove</i>	0
115	<i>botTypeId</i>		<i>remove</i>	0
116	<i>name</i>		<i>remove</i>	0
117	<i>stockTypeId</i>		<i>remove</i>	0
118	<i>stockTypeName</i>		<i>remove</i>	0
119	<i>totalQuantity</i>		<i>remove</i>	0
120	<i>orderId</i>		<i>remove</i>	0
121	<i>adjustId</i>		<i>remove</i>	0
122	<i>employedDate</i>		<i>remove</i>	0
123	<i>autoMapRoute()</i>		<i>remove</i>	0
124	<i>autoMapCar()</i>		<i>remove</i>	0
125	<i>createShipment()</i>		<i>remove</i>	0

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง
126	cancelShipment()		remove	0
127	confirmAdjustment()		remove	0
128	cancelConfirmAdjustment()		remove	0
129	updateStock()		remove	0
130	defineUserAndPermission()		remove	0
131	CarSchedule		remove	0
132	Car		remove	0
133	CarType		remove	0
134	Shipment		remove	0
135	DivisionManager		remove	0
136	Employee		remove	0
137	Separation		remove	0
138	Category		remove	0
139	StockType		remove	0
140	Stock		remove	0
141		Transportation	add	0
142		Center	add	0
143		deliveryNumber	add	0
144		transactinDate	add	0
145		centerId	add	0
146		centerId	add	0
147		centerId	add	0
148		centerId	add	0
149		centerId	add	0
150		centerId	add	0

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง	คะแนน ความคล้ายคลึง
151		<i>centerId</i>	<i>add</i>	0
152		<i>condition</i>	<i>add</i>	0
153		<i>fromUnit</i>	<i>add</i>	0
154		<i>toUnit</i>	<i>add</i>	0
155		<i>totalAmount</i>	<i>add</i>	0
156		<i>totalAmount</i>	<i>add</i>	0
157		<i>totalAmount</i>	<i>add</i>	0
158		<i>status</i>	<i>add</i>	0
159		<i>status</i>	<i>add</i>	0
160		<i>status</i>	<i>add</i>	0
161		<i>scheduleNo</i>	<i>add</i>	0
162		time	add	0
163		<i>overrideSchedule</i>	<i>add</i>	0
164		<i>vehicleId</i>	<i>add</i>	0
165		<i>ownerCenterId</i>	<i>add</i>	0
166		<i>vehicleTypeId</i>	<i>add</i>	0
167		<i>vehicleTypeName</i>	<i>add</i>	0
168		feId	add	0
169		feeName	add	0
170		feePlan	add	0
171		feeTypeId	add	0
172		feeTypeName	add	0
173		<i>locationNo</i>	<i>add</i>	0
174		<i>locationName</i>	<i>add</i>	0
175		<i>street</i>	<i>add</i>	0

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง
176		district	add	0
177		city	add	0
178		zipCode	add	0
179		locTypeId	add	0
180		locTypeName	add	0
181		sortingNumber	add	0
182		accountDate	add	0
183		transactionDate	add	0
184		cashOwner	add	0
185		productTypeNo	add	0
186		productFormId	add	0
187		botTypeId	add	0
188		name	add	0
189		inventoryTypeId	add	0
190		inventoryTypeName	add	0
191		totalQuantity	add	0
192		orderNumber	add	0
193		invoiceNo	add	0
194		taxId	add	0
195		systemType	add	0
196		stockAdjustNumber	add	0
197		startWorkingDate	add	0
198		createDelivery()	add	0
199		updateDelivery()	add	0
200		autoRoute()	add	0

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง	คะแนน ความคล้ายคลึง
201		<i>autoCar()</i>	<i>add</i>	0
202		computeFee()	add	0
203		<i>confirmStockAdjustment()</i>	<i>add</i>	0
204		<i>cancelConfirmStock Adjustment()</i>	<i>add</i>	0
205		createOrder()	add	0
206		updateOrder()	add	0
207		cancelOrder()	add	0
208		createSorting()	add	0
209		updateSorting()	add	0
210		cancelSorting()	add	0
211		<i>updateProduct()</i>	<i>add</i>	0
212		manageUser()	add	0
213		managePermission()	add	0
214		<i>Delivery</i>	<i>add</i>	0
215		<i>Schedule</i>	<i>add</i>	0
216		<i>Vehicle</i>	<i>add</i>	0
217		<i>VehicleType</i>	<i>add</i>	0
218		<i>Fee</i>	<i>add</i>	0
219		FeeType	add	0
220		DataConfiguration	add	0
221		<i>SectionManager</i>	<i>add</i>	0
222		Customer	add	0
223		Location	add	0
224		LocationType	add	0
225		Guest	add	0

ตารางที่ ค.2 ผลลัพธ์ของการเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยใช้ UMLDiff (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง	คะแนนความคล้ายคลึง
226		<i>Officer</i>	<i>add</i>	0
227		ExternalSorting	add	0
228		Sorting	<i>add</i>	0
229		InternalSorting	add	0
230		<i>ProductType</i>	<i>add</i>	0
231		<i>InventoryType</i>	<i>add</i>	0
232		<i>Inventory</i>	<i>add</i>	0

ตารางที่ ค.3 ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง
1	Transaction	Transaction	match
2	Inventory	Inventory	match
3	Shipping	Transportation	smatch
4	Master	Center	smatch
5	Order	Order	match
6	TransactionType	TransactionType	match
7	Treasury	Treasury	match
8	Product	Product	match
9	Package	Package	match
10	Route	Route	match
11	Driver	Driver	match
12	Administrator	Administrator	match

ตารางที่ ค.3 ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง
13	AO	AO	match
14	SAO	SAO	match
15	DLA	DLA	match
16	CenterUser	CenterUser	match
17	Member	Member	match
18	InventoryUser	InventoryUser	match
19	User	User	match
20	OrderDetail	OrderItem	smatch
21	Separation	Sorting	smatch
22	Adjustment	StockAdjustment	rename
23	AdjustmentDetail	AdjustmentItem	smatch
24	Category	ProductType	smatch
25	StockType	InventoryType	smatch
26	Stock	Inventory	smatch
27	CarSchedule	Schedule	rename
28	Car	Vehicle	smatch
29	CarType	VehicleType	smatch
30	Shipment	Delivery	smatch
31	DivisionManager	SectionManager	smatch
32	Employee	Officer	smatch
33	transactionDate	transactionDate	match
34	fromUnit	fromUnit	match
35	toUnit	toUnit	match
36	totalAmount	totalAmount	match
37	status	status	match
38	transactionTypeid	transactionTypeid	match

ตารางที่ ค.3 ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง
39	transactionTypeName	transactionTypeName	match
40	totalAmount	totalAmount	match
41	status	status	match
42	productName	productName	match
43	status	status	match
44	name	name	match
45	routeType	routeType	match
46	driverId	driverId	match
47	title	title	match
48	firstName	firstName	match
49	lastName	lastName	match
50	sex	sex	match
51	birthDate	birthDate	match
52	licenseId	licenseId	match
53	createOrder()	createOrder()	match
54	updateOrder()	updateOrder()	match
55	cancelOrder()	cancelOrder()	match
56	confirmOrder()	confirmOrder()	match
57	cancelConfirmOrder()	cancelConfirmOrder()	match
58	approveOrder()	approveOrder()	match
59	cancelApproveOrder()	cancelApproveOrder()	match
60	rejectOrder()	rejectOrder()	match
61	approveShipment()	approveShipment()	match
62	cancelApproveShipment()	cancelApproveShipment()	match
63	confirmTreasury()	confirmTreasury()	match
64	cancelConfirmTreasury()	cancelConfirmTreasury()	match

ตารางที่ ค.3 ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง
65	userId	userId	match
66	title	title	match
67	firstName	firstName	match
68	lastName	lastName	match
69	sex	sex	match
70	birthDate	birthDate	match
71	position	position	match
72	quantity	quantity	match
73	amount	amount	match
74	accountDate	accountDate	match
75	transactionDate	transactionDate	match
76	totalAmount	totalAmount	match
77	status	status	match
78	transactionDate	transactionDate	match
79	totalAmount	totalAmount	match
80	status	status	match
81	availableAmount	availableAmount	match
82	reserveAmount	reserveAmount	match
83	productFormId	productFormId	match
84	botTypeid	botTypeid	match
85	name	name	match
86	status	status	match
87	totalAmount	totalAmount	match
88	totalQuantity	totalQuantity	match
89	workingDate	workingDate	match
90	transactinDate	transactinDate	match

ตารางที่ ค.3 ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง
91	condition	condition	match
92	fromUnit	fromUnit	match
93	toUnit	toUnit	match
94	totalAmount	totalAmount	match
95	status	status	match
96	orderId	orderNumber	smatch
97	treasuryId	treasuryNumber	smatch
98	productId	productNo	rename
99	packageId	packageNo	rename
100	quantity	quantityPerUnit	rename
101	routeId	routeNo	rename
102	configData()	configSystemData()	rename
103	createAdjustment()	createStockAdjustment()	rename
104	updateAdjustment()	updateStockAdjustment()	rename
105	cancelAdjustment()	cancelStockAdjustment()	rename
106	createSeparation()	createSorting()	smatch
107	updateSeparation()	updateSorting()	smatch
108	cancelSeparation()	cancelSorting()	smatch
109	approveAdjustment()	approveStockAdjustment()	rename
110	cancelApproveAdjustment()	cancelApproveStockAdjustment()	rename
111	separationId	sortingNumber	smatch
112	stockTypeId	inventoryTypeId	smatch
113	stockTypeName	inventoryTypeName	smatch
114	updateStock()	updateProduct()	smatch
115	carScheduleId	scheduleNo	rename
116	overrideCarSchedule	overrideSchedule	rename

ตารางที่ ค.3 ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง
117	carId	vehicleId	smatch
118	ownerUnitId	ownerCenterId	smatch
119	carTypeId	vehicleTypeId	smatch
120	carTypeName	vehicleTypeName	smatch
121	autoMapRoute()	autoRoute()	rename
122	autoMapCar()	autoCar()	rename
123	createShipment()	createDelivery()	smatch
124	confirmAdjustment()	confirmStockAdjustment()	rename
125	cancelConfirmAdjustment()	cancelConfirmStockAdjustment()	rename
126	unitId	centerId	smatch
127	unitId	centerId	smatch
128	unitId	centerId	smatch
129	unitId	centerId	smatch
130	unitId	centerId	smatch
131	unitId	centerId	smatch
132	employedDate	startWorkingDate	smatch
133	adjustId		remove
134	categoryId	productTypeNo	smatch
135	timeTable		remove
136	shipmentId	deliveryNumber	smatch
137	defineUserAndPermission()		remove
138	cancelShipment()		remove
139		feId	add
117	carId	vehicleId	smatch
118	ownerUnitId	ownerCenterId	smatch
119	carTypeId	vehicleTypeId	smatch
120	carTypeName	vehicleTypeName	smatch

ตารางที่ ค.3 ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง
121	autoMapRoute()	autoRoute()	rename
122	autoMapCar()	autoCar()	rename
123	createShipment()	createDelivery()	smatch
124	confirmAdjustment()	confirmStockAdjustment()	rename
125	cancelConfirmAdjustment()	cancelConfirmStockAdjustment()	rename
126	unitId	centerId	smatch
127	unitId	centerId	smatch
128	unitId	centerId	smatch
129	unitId	centerId	smatch
130	unitId	centerId	smatch
131	unitId	centerId	smatch
132	employedDate	startWorkingDate	smatch
133	adjustId		remove
134	categoryId	productTypeNo	smatch
135	timeTable		remove
136	shipmentId	deliveryNumber	smatch
137	defineUserAndPermission()		remove
138	cancelShipment()		remove
139		feeld	add
140		feeName	add
141		feePlan	add
142		feeTypeid	add
143		feeTypeName	add
144		locationNo	add
145		locationName	add
146		street	add

ตารางที่ ค.3 ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภทการเปลี่ยนแปลง
147		district	add
148		city	add
149		zipCode	add
150		locTypeId	add
151		locTypeName	add
152		invoiceNo	add
153		taxId	add
154		systemType	add
155		cashOwner	add
156		stockAdjustNumber	add
157		time	add
158		computeFee()	add
159		createOrder()	add
160		updateOrder()	add
161		cancelOrder()	add
162		createSorting()	add
163		updateSorting()	add
164		cancelSorting()	add
165		manageUser()	add
166		managePermission()	add
167		updateDelivery()	add
168		ExternalSorting	add
169		InternalSorting	add
170		FeeType	add
171		DataConfiguration	add

ตารางที่ ค.3 ผลลัพธ์ที่ได้จากการพิจารณาเปรียบเทียบแผนภาพคลาสเชิงแนวคิดในภาพที่ 5.1 และ 5.2 โดยผู้เชี่ยวชาญคนที่ 3 (ต่อ)

ลำดับ	เอนทิตีของภาพที่ 5.1	เอนทิตีของภาพที่ 5.2	ประเภท การเปลี่ยนแปลง
172		Customer	add
173		Location	add
174		LocationType	add
175		Guest	add
176	fee	Fee	changetype

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวปัทมาภรณ์ สายสิม เกิดเมื่อวันที่ 13 มิถุนายน พ.ศ. 2521 ที่จังหวัดศรีสะเกษ สำเร็จการศึกษาระดับปริญญาบัณฑิต หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ จากมหาวิทยาลัยอุบลราชธานี เมื่อ พ.ศ. 2544 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ณ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2554