การรวมแบบจำลองข้อมูลเชิงวัตถุโดยการสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาสและการกำกับลงบนแบบจำลอง

นางสาว มนัชยา ชมธวัช

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

INTEGRATING OBJECT-ORIENTED DATA MODELS BY CONSTRUCTING AN INTEGRATED ONTOLOGY FROM AN ANALYSIS OF CLASS RELATIONSHIPS AND ANNOTATING ONTO THE MODELS

Miss Manachaya Jamadhvaja

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2006

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การรวมแบบจำลองข้อมูลเชิงวัตถุโดยการสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาสและการกำกับลงบนแบบจำลอง
โดย	นางสาว มนัชยา ชมธวัช
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.ยรรยง เต็งอำนวย)

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

..... กรรมการ
(อาจารย์ จารุมาตร ปิ่นทอง)

..... กรรมการ
(รองศาสตราจารย์ ดร.วิเชียร ชุติมาสกุล)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

มนัชยา ชมธวัช : การรวมแบบจำลองข้อมูลเชิงวัตถุโดยการสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาสและการกำกับลงบนแบบจำลอง. (INTEGRATING OBJECT-ORIENTED DATA MODELS BY CONSTRUCTING AN INTEGRATED ONTOLOGY FROM AN ANALYSIS OF CLASS RELATIONSHIPS AND ANNOTATING ONTO THE MODELS) อ. ที่ปรึกษา : ผศ.ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา, 198 หน้า.

การรวมแบบจำลองข้อมูลเป็นวิธีการอันมีประสิทธิภาพวิธีการหนึ่งที่จะทำให้สามารถใช้งานข้อมูลซึ่งถูกจัดเก็บไว้ในแหล่งข้อมูลต่างแหล่งกันร่วมกันได้ ด้วยวิธีการนี้ผู้ใช้จะสามารถเข้าถึงข้อมูลทั้งหมดได้ในคราวเดียวกันโดยอาศัยการติดต่อที่จุดเดียว อย่างไรก็ตาม ความต่างแบบของข้อมูลที่นำมารวมทำให้ผู้รวมจะต้องคำนึงถึงความถูกต้องและความต้องกันของข้อมูล ปัจจัยสำคัญที่ส่งผลต่อการแก้ปัญหาความขัดแย้งของความต่างแบบระหว่างแบบจำลองก็คือความหมายที่ครบถ้วนของข้อมูลที่ได้รับการแสดงไว้ในแบบจำลอง เนื่องจากแบบจำลองข้อมูลเชิงวัตถุแสดงไว้เฉพาะความหมายของข้อมูลในเชิงโครงสร้าง (เช่น คลาส แอททริบิวต์และความสัมพันธ์ระหว่างคลาส) การจะแสดงความหมายอื่นเพิ่มเติม (เช่น ค่าข้อมูล หน่วยข้อมูล รวมถึงคำที่มีความหมายเหมือนกันและคำที่เป็นคำลูกกลุ่มกัน) จำเป็นต้องอาศัยวิธีการอื่นดังเช่น การอธิบายแบบจำลองข้อมูลด้วยออนโทโลจี

วิทยานิพนธ์ฉบับนี้นำเสนอแนวคิดใหม่ในการนำออนโทโลจีมาใช้รวมแบบจำลองข้อมูลเรียกว่า "แนวคิดแบบรวมหลายออนโทโลจี" ซึ่งเป็นวิธีการในการรวมแบบจำลองข้อมูลเชิงวัตถุสองชุดเข้าด้วยกันโดยอาศัยการวิเคราะห์จากออนโทโลจีที่แสดงแทนแบบจำลองข้อมูลแต่ละชุด ภายในงานวิจัยได้กำหนดให้นำออนโทโลจีมาใช้อธิบายความหมายของข้อมูลในแต่ละแบบจำลองข้อมูล จากนั้นออนโทโลจีจะถูกนำไปวิเคราะห์และเปรียบเทียบกันเพื่อตรวจสอบว่ามีความคล้ายคลึงหรือแตกต่างกันมากน้อยเพียงใด แล้วนำผลที่ได้มาสร้างเป็นออนโทโลจีรวม เพื่อจะนำมาใช้ในการสอบถามข้อมูลในระบบสารสนเทศรวม

งานวิจัยนี้ตั้งอยู่บนสมมติฐานที่ว่า หากนำแบบจำลองข้อมูลที่ได้รับการอธิบายและแสดงแทนความหมายในรูปแบบที่เหมาะสมและเพียงพอ มารวมเข้าด้วยกันโดยใช้ขั้นตอนวิธีที่เหมาะสม จะทำให้ได้ระบบสารสนเทศรวมที่ถูกต้องและครบถ้วน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....มนัชยา ชมธวัช.....

สาขาวิชา....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา.....ทวิตีย์ เสนีวงศ์ ณ อยุธยา.....

ปีการศึกษา....2549.....

4670439021 : MAJOR COMPUTER ENGINEERING

KEY WORD: ONTOLOGY / ONTOLOGY INTEGRATION / DATA MODEL INTEGRATION / OBJECT-ORIENTED DATA MODEL

MANACHAYA JAMADHVAJA : INTEGRATING OBJECT-ORIENTED DATA MODELS BY CONSTRUCTING AN INTEGRATED ONTOLOGY FROM AN ANALYSIS OF CLASS RELATIONSHIPS AND ANNOTATING ONTO THE MODELS. THESIS ADVISOR : ASST. PROF. TWITTIE SENIVONGSE, Ph.D., 198 pp.

Data model integration is an effective method to interoperate data that reside in different sources for the purpose of providing users with a single point of access to those data. Due to data heterogeneity, data correctness and consistency are significant for integration. Richer semantics of data is a major factor in resolving conflicts among heterogeneous data models. As object-oriented data model represents only schema-based semantics of data (e.g. classes, attributes, and class relationships), an alternative method such as ontology is useful for representing additional semantics (e.g. data values, data units, and synonym and hypernym lists).

This thesis proposes a new approach to an ontology-based integration of data models, called "Integrated Multiple Ontology approach", which provides a method for integrating two object-oriented data models by using an analysis of their ontologies. In this work, ontology will be used to describe semantics of data in each data model. Then the ontologies are analysed and compared to determine their similarities and differences. The result of the comparison is used to devise an integrated ontology that will enable querying on the integrated information.

This work is based on an assumption that the combination of a good knowledge representation that describes adequate semantics of the data model and a suitable integration algorithm leads to the correct and consistent integrated information system.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Department.....Computer Engineering..... Student's signature ...Manachaya Jamadhvaja...
Field of study.....Computer Engineering..... Advisor's signature ...Twittie Senivongse...
Academic year.....2006.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยคำแนะนำและความช่วยเหลืออย่างดียิ่งจากผู้ช่วยศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา อาจารย์ที่ปรึกษา ซึ่งคอยชี้แนะแนวทาง ตอบคำถามและให้ข้อคิดเห็นต่างๆ อันเป็นประโยชน์มาโดยตลอด

ขอขอบพระคุณอาจารย์ ดร.ยรรยง เต็งอำนวย อาจารย์จารุมาตร ปิ่นทอง และรองศาสตราจารย์ ดร.วิเชียร ชุติมาสกุล คณะกรรมการสอบวิทยานิพนธ์ ซึ่งช่วยให้แง่คิดในมุมมองที่หลากหลาย และเพิ่มประเด็นที่น่าสนใจเกี่ยวกับงานวิจัย ทั้งในระหว่างการสอบโครงร่างและในช่วงการทำวิทยานิพนธ์

ขอขอบคุณพี่ เพื่อน และน้อง สมาชิกห้องปฏิบัติการวิศวกรรมระบบสารสนเทศ และในภาควิชาวิศวกรรมคอมพิวเตอร์ โดยเฉพาะอย่างยิ่งนางสาวเนตรนภา สีหารี นางสาวรัศมีทิพย์ วิตา นายกิตติพันธ์ เจียมนภานนท์ นางสาวภัทราภรณ์ อริยปรีชากุล และนายจักกนาท วิวัฒนาวรสิน ซึ่งคอยให้คำแนะนำและความช่วยเหลือในด้านต่างๆ อยู่เสมอ ทำให้การทำงานวิจัยเป็นไปอย่างราบรื่น

สุดท้ายนี้ ขอขอบพระคุณคุณแม่และคุณพ่อสำหรับความรัก ความเชื่อมั่น โอกาส และการสนับสนุนที่มีให้กับลูกคนนี้อยู่เสมอ ขอบคุณพี่นุและน้องหนุ่มสำหรับกำลังใจและความห่วงใยในหลายต่อหลายครั้ง และขอบคุณตี๋สำหรับความอดทนต่อเรื่องไม่เป็นเรื่อง และกำลังใจอย่างมากมายเมื่อเกิดปัญหาตลอดช่วงเวลาที่ผ่านมา

หากขาดใครคนใดคนหนึ่งไป วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จสมบูรณ์ได้ถึงขนาดนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง
บทคัดย่อภาษาอังกฤษ..... จ
กิตติกรรมประกาศ..... ฉ
สารบัญ..... ช
สารบัญตาราง..... ญ
สารบัญภาพ..... ฏ

บทที่

1 บทนำ..... 1
1.1 ความเป็นมาและความสำคัญของปัญหา..... 1
1.2 วัตถุประสงค์ของการวิจัย..... 6
1.3 ขอบเขตของการวิจัย..... 6
1.4 วิธีดำเนินการวิจัย..... 7
1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 8
1.6 โครงสร้างวิทยานิพนธ์..... 8
1.7 ผลงานที่ตีพิมพ์จากการวิจัย..... 9

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 10
2.1 ทฤษฎีที่เกี่ยวข้อง..... 10
2.1.1 ออนโทโลจี..... 10
2.1.2 อาร์ดีเอฟ..... 11
2.1.3 อาวล์..... 11
2.1.4 การอนุมาน..... 14
2.1.5 เมตาโมเดลของนิยามของออนโทโลจี..... 14
2.1.6 โพรเทเจ..... 16
2.1.7 จีนา..... 17
2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง..... 18
2.2.1 งานวิจัยเกี่ยวกับการรวมแบบจำลองข้อมูลเชิงวัตถุ..... 18
2.2.2 งานวิจัยเกี่ยวกับการนำออนโทโลจีมาใช้ในการรวมแบบจำลองข้อมูล..... 20
2.2.3 งานวิจัยเกี่ยวกับการรวมออนโทโลจี..... 24

บทที่ หน้า

3 การสร้างออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุ..... 28
3.1 อัปเปอร์ออนโทโลจี..... 28
3.2 การนำอัปเปอร์ออนโทโลจีมาใช้สร้างออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุ..... 31
3.2.1 ส่วนการสร้างโลคัลออนโทโลจีเพื่ออธิบายความหมายที่ปรากฏในแบบจำลองข้อมูลเชิงวัตถุ..... 31
3.2.2 ส่วนการเพิ่มเติมความหมายของแบบจำลองข้อมูลเชิงวัตถุลงในออนโทโลจีที่สร้างไว้... 32
3.3 การสร้างออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจี..... 31
4 การเปรียบเทียบและการรวมออนโทโลจี..... 41
4.1 การเปรียบเทียบโลคัลออนโทโลจี..... 41
4.1.1 การเปรียบเทียบความหมาย..... 41
4.1.2 การเปรียบเทียบคลาสของแอททริบิวต์..... 44
4.1.3 การเปรียบเทียบคลาสของคลาส..... 52
4.2 การรวมโลคัลออนโทโลจี..... 54
4.2.1 การรวมความหมาย..... 54
4.2.2 การรวมคลาสของแอททริบิวต์..... 56
4.2.3 การรวมคลาสของคลาส..... 59
4.3 ตัวอย่างการเปรียบเทียบและการรวมโลคัลออนโทโลจี..... 76
4.3.1 ตัวอย่างการเปรียบเทียบโลคัลออนโทโลจี..... 76
4.3.2 ตัวอย่างการรวมโลคัลออนโทโลจี..... 79
5 เฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจี..... 81
5.1 แนวคิดในการออกแบบเฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจี... 81
5.2 เฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีตามแนวคิดแบบรวมหลายออนโทโลจี..... 84
5.2.1 ส่วนการสร้างโลคัลออนโทโลจี..... 84
5.2.2 ส่วนการสร้างออนโทโลจีรวม..... 95
5.2.3 ส่วนการสร้างแมปปิงระหว่างออนโทโลจีรวมกับแบบจำลองข้อมูลเชิงวัตถุ..... 110
5.3 แผ่นแบบการสอบถามข้อมูลผ่านการทำงานของเฟรมเวิร์ค..... 113
6 การทดสอบการทำงานเพื่อประเมินความสามารถของเฟรมเวิร์ค..... 117
6.1 การทดสอบผลการสร้างออนโทโลจีรวม..... 117

บทที่ หน้า

6.1.1 การทดสอบผลการสร้างออนโทโลจีรวมด้วยแบบจำลองข้อมูลเชิงวัตถุจากงานวิจัย...118
6.1.2 การทดสอบผลการสร้างออนโทโลจีรวมด้วยแบบจำลองข้อมูลเชิงวัตถุที่ประยุกต์จากแบบจำลองข้อมูลที่ออกแบบเพื่อใช้งานจริง.....120
6.2 การทดสอบความถูกต้องในการสอบถามข้อมูลผ่านระบบการรวม.....122
6.2.1 การทดสอบความถูกต้องในการสอบถามข้อมูลผ่านระบบการรวมโดยใช้ออนโทโลจีรวมที่ไม่ผ่านการอนุมาน.....123
6.2.2 การทดสอบความถูกต้องในการสอบถามข้อมูลผ่านระบบการรวมโดยใช้ออนโทโลจีรวมที่ผ่านการอนุมานแล้ว.....128
7 สรุปผลการวิจัย และข้อเสนอแนะ.....132
7.1 สรุปผลการวิจัย.....132
7.1.1 แนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูลแบบรวมหลายออนโทโลจี.....133
7.1.2 ขั้นตอนวิธีการรวมออนโทโลจีโดยการสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาส.....134
7.1.3 การอธิบายแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจี.....135
7.1.4 การกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูลเชิงวัตถุเพื่อใช้ในการสอบถามข้อมูล.....136
7.1.5 การเปรียบเทียบระหว่างแนวคิดแบบรวมหลายออนโทโลจีกับแนวคิดอื่นที่มีอยู่เดิม...138
7.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย.....140
7.3 ข้อเสนอแนะและแนวทางการทำวิจัยในอนาคต.....141
รายการอ้างอิง.....143
ภาคผนวก.....148
ภาคผนวก ก ตัวอย่างการรวมแบบจำลองข้อมูลคู่ที่ 1.....149
ภาคผนวก ข ตัวอย่างการรวมแบบจำลองข้อมูลคู่ที่ 2.....166
ภาคผนวก ค ผลงานที่ตีพิมพ์จากการวิจัย.....183
ประวัติผู้เขียนวิทยานิพนธ์.....198

สารบัญตาราง

ตารางที่ หน้า
ตารางที่ 4.1 ค่าความสัมพันธ์ r จากการเปรียบเทียบแต่ละคู่ของคลาสของชื่อความหมาย..... 42
ตารางที่ 4.2 ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของความหมายของ T1 และ T2..... 43
ตารางที่ 4.3 ค่าความสัมพันธ์จากการเปรียบเทียบแต่ละคู่ของคลาสของแอททริบิวต์.....46
ตารางที่ 4.4 ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส C1 และ C2.....49
ตารางที่ 4.5 ความสัมพันธ์จากการเปรียบเทียบคลาสของคลาสแต่ละคู่ พิจารณาจากความสัมพันธ์จากการเปรียบเทียบความหมายของคลาสของคลาสร่วมกับความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส.....53
ตารางที่ 4.6 ผลการเปรียบเทียบคลาสของคลาสภายในโลคัลออนโทโลจีที่ 1 และ 2..... 79
ตารางที่ 7.1 การเปรียบเทียบระหว่างแนวคิดแบบรวมหลายออนโทโลจีกับแนวคิดอื่นที่มีอยู่เดิม..... 139

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

รูปที่ หน้า
รูปที่ 1.1 แนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูล..... 3
รูปที่ 1.2 แนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูลแบบรวมหลายออนโทโลจี..... 4
รูปที่ 2.1 อาร์ดีเอฟกราฟ..... 11
รูปที่ 2.2 การเขียนแทนออนโทโลจีในรูปของอาร์ดีเอฟกราฟ..... 12
รูปที่ 2.3 การเขียนแทนออนโทโลจีในรูปของอาร์ดีเอฟทริปเปิล..... 12
รูปที่ 2.4 การเขียนแทนออนโทโลจีในรูปของอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล..... 12
รูปที่ 2.5 ตัวอย่างการแปลงแบบจำลองข้อมูลเชิงวัตถุ (ก) เป็นออนโทโลจีในรูปของอาร์ดีเอฟกราฟ (ข)... 15
รูปที่ 2.6 ลักษณะของคลาสแบบ Primitive และคลาสแบบ Defined ที่สร้างจากโพรเทเจ-อาวล์..... 16
รูปที่ 2.7 หน้าจอการทำงานของโปรแกรมโพรเทเจ-อาวล์..... 17
รูปที่ 2.8 ตัวอย่างการแมปจากลำดับชั้นข้อมูลไปยังโกลบอลออนโทโลจี..... 21
รูปที่ 2.9 องค์ประกอบของระบบตามงานวิจัย OBSERVER..... 22
รูปที่ 2.10 องค์ประกอบของเฟรมเวิร์คของงานวิจัย COIN..... 23
รูปที่ 2.11 ขั้นตอนการทำงานเพื่อสร้างออนโทโลจีรวมของงานวิจัย FCA-MERGE..... 26
รูปที่ 2.12 วิธีการรวมออนโทโลจีที่ใช้งานในแนวคิดเกี่ยวกับการรวมออนโทโลจี..... 27
รูปที่ 3.1 อัปเปอร์ออนโทโลจี..... 29
รูปที่ 3.2 ตัวอย่างการนำอัปเปอร์ออนโทโลจีมาใช้อธิบายยูเอ็มแอลของคลาสไดอะแกรมในรูปของออนโทโลจี..... 32
รูปที่ 3.3 ตัวอย่างการนำอัปเปอร์ออนโทโลจีมาใช้ในการเพิ่มความหมายให้กับคลาสของคลาสและคลาสของแอททริบิวต์ในโลคัลออนโทโลจี..... 34
รูปที่ 3.4 ตัวอย่างแบบจำลองข้อมูลเชิงวัตถุ 2 ชุดที่ต้องการนำมารวมกัน..... 34
รูปที่ 3.5 ตัวอย่างโลคัลออนโทโลจีสำหรับอธิบายแบบจำลองข้อมูลชุดที่ 1 (รูปที่ 3.4(ก))..... 35
รูปที่ 3.6 ตัวอย่างโลคัลออนโทโลจีสำหรับอธิบายแบบจำลองข้อมูลชุดที่ 2 (รูปที่ 3.4(ข))..... 36
รูปที่ 3.7 การระบุคำที่มีความหมายเหมือนกันและคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างแบบจำลองข้อมูลสองชุดในรูปที่ 3.4..... 39
รูปที่ 3.8 ตัวอย่างแมปปิงแสดงการระบุคำที่มีความหมายเหมือนกันและคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างโลคัลออนโทโลจีทั้งสองชุดจากรูปที่ 3.5 และ 3.6..... 40
รูปที่ 4.1 ตัวอย่างการเปรียบเทียบความหมายระหว่างคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2..... 44
รูปที่ 4.2 ขั้นตอนการเปรียบเทียบแต่ละคู่ของคลาสของแอททริบิวต์เพื่อหาระดับความสัมพันธ์..... 46

รูปที่ หน้า

รูปที่ 4.3 ตัวอย่างการเปรียบเทียบคลาสของแอททริบิวต์ระหว่างชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2..... 51
รูปที่ 4.4 ตัวอย่างคลาสของคลาสและคลาสของความหมายที่อธิบายคลาสของคลาส C1 และ C2.... 55
รูปที่ 4.5 รูปแบบในการรวมคลาสของความหมายเมื่อพิจารณาจากระดับความสัมพันธ์ระหว่างคลาสของคลาส C1 และคลาสของคลาส C2..... 56
รูปที่ 4.6 ขั้นตอนโดยรวมในการรวมคลาส..... 60
รูปที่ 4.7 ตัวอย่างการรวมคลาสของคลาส C1 และ C2 ที่มีความสัมพันธ์แบบเท่ากัน..... 62
รูปที่ 4.8 การกำจัดเส้นทางการสืบทอดระหว่างคลาสของคลาสที่ซ้ำซ้อน..... 62
รูปที่ 4.9 ตัวอย่างการรวมคลาสของคลาส C1 และ C2 ที่มีความสัมพันธ์แบบซูเปอร์คลาส/ สับคลาส. 63
รูปที่ 4.10 ตัวอย่างการรวมคลาสของคลาส C1 และ C2 ที่มีความสัมพันธ์แบบพี่น้อง..... 64
รูปที่ 4.11 ตัวอย่างการรวมคลาสของคลาส C3 ที่เกิดจากการรวมคลาสของคลาส C1 และ C2 ที่มีความสัมพันธ์แบบพี่น้องและ C4 ที่เป็นซูเปอร์คลาสของคลาสเดิม..... 65
รูปที่ 4.12 ตัวอย่างการรวมคลาสของคลาส C1 และ C2 เมื่อคลาสของคลาส C2 ถูกกำหนดให้เป็นชนิดข้อมูลของบางคลาสของแอททริบิวต์ในคลาสของคลาส C3..... 66
รูปที่ 4.13 ตัวอย่างการรวมคลาสของคลาสระหว่างคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2 โดยพิจารณาจากระดับความสัมพันธ์..... 68
รูปที่ 4.14 ตัวอย่างการเชื่อมความสัมพันธ์ระหว่างคลาสของแอททริบิวต์และคลาสของคลาส..... 69
รูปที่ 4.15 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 และ CA2 ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันตามเงื่อนไขกรณีที่ 1..... 70
รูปที่ 4.16 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 และ CA2 ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันตามเงื่อนไขกรณีที่ 2..... 71
รูปที่ 4.17 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 และ CA2 ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันตามเงื่อนไขกรณีที่ 3..... 71
รูปที่ 4.18 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 และ CA2 ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันตามเงื่อนไขกรณีที่ 4..... 72
รูปที่ 4.19 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ที่ถูกระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกันตามเงื่อนไขกรณีที่ 1..... 73
รูปที่ 4.20 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ที่ถูกระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกันตามเงื่อนไขกรณีที่ 2..... 80

รูปที่ หน้า

รูปที่ 4.21 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ที่ถูกระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกันตามเงื่อนไขกรณีที่ 3..... 74
รูปที่ 4.22 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ที่ถูกระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกันตามเงื่อนไขกรณีที่ 4..... 74
รูปที่ 4.23 ตัวอย่างการรวมคลาสของคลาสระหว่างคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2 หลังจากพิจารณาตามขั้นตอนการรวมคลาสของคลาสครบทุกขั้นตอน..... 75
รูปที่ 4.24 ออนโทโลจีรวมที่ได้จากการรวมโลคัลออนโทโลจีที่ 1 ในรูปที่ 3.5 และโลคัลออนโทโลจีที่ 2 ในรูปที่ 3.6..... 77
รูปที่ 5.1 เฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีตามแนวคิดแบบรวมหลายออนโทโลจี..... 83
รูปที่ 5.2 อัปเปอร์ออนโทโลจีในรูปของอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล..... 85
รูปที่ 5.3 หน้าจอโปรแกรมโพรเทเจเพื่อกำหนดดีฟอลต์เนมสเปซและนำเข้าอัปเปอร์ออนโทโลจี..... 87
รูปที่ 5.4 หน้าจอโปรแกรมโพรเทเจเพื่อสร้างคลาสภายในโลคัลออนโทโลจี..... 88
รูปที่ 5.5 หน้าจอโปรแกรมโพรเทเจเพื่อสร้างคลาสต่างๆ ลงในโลคัลออนโทโลจี..... 88
รูปที่ 5.6 หน้าจอโปรแกรมโพรเทเจเพื่อสร้างพรอพเพอร์ตีภายในโลคัลออนโทโลจี..... 90
รูปที่ 5.7 หน้าจอโปรแกรมโพรเทเจเพื่อสร้างพรอพเพอร์ตีต่างๆ ลงในโลคัลออนโทโลจี..... 91
รูปที่ 5.8 หน้าจอโปรแกรมโพรเทเจเพื่อเก็บบันทึกไฟล์โลคัลออนโทโลจีแต่ละชุด..... 92
รูปที่ 5.9 หน้าจอโปรแกรมโพรเทเจเพื่อกำหนดดีฟอลต์เนมสเปซและนำเข้าโลคัลออนโทโลจี..... 93
รูปที่ 5.10 หน้าจอโปรแกรมโพรเทเจเพื่อระบุว่าเป็นคำที่มีความหมายเหมือนกันระหว่างโลคัลออนโทโลจี..... 93
รูปที่ 5.11 หน้าจอโปรแกรมโพรเทเจเพื่อระบุว่าเป็นคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างโลคัลออนโทโลจี..... 94
รูปที่ 5.12 ขั้นตอนการทำงานของโปรแกรมการรวมออนโทโลจี..... 96
รูปที่ 5.13 หน้าจอการทำงานของโปรแกรมการรวมออนโทโลจี..... 97
รูปที่ 5.14 หน้าต่างสำหรับเลือกไฟล์โลคัลออนโทโลจีและออนโทโลจีความหมายของโปรแกรมการรวมออนโทโลจี..... 97
รูปที่ 5.15 หน้าจอโปรแกรมการรวมออนโทโลจีเมื่อระบุไฟล์ออนโทโลจีทั้งสามไฟล์ก่อนดำเนินการเปรียบเทียบและรวมออนโทโลจี..... 98
รูปที่ 5.16 หน้าจอโปรแกรมการรวมออนโทโลจีแสดงผลลัพธ์เมื่อไฟล์ออนโทโลจีทั้งสามได้รับการป้อนเข้าสู่โปรแกรม..... 98

รูปที่ หน้า

รูปที่ 5.17 หน้าจอโปรแกรมการรวมออนโทโลจีแสดงผลลัพธ์หลังการรวมออนโทโลจีเสร็จสิ้น.....99
รูปที่ 5.18 บางส่วนของออนโทโลจีรวมเบื้องต้นจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.owl..... 99
รูปที่ 5.19 บางส่วนของผลการรวมออนโทโลจีจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.txt..... 104
รูปที่ 5.20 หน้าจอโปรแกรมโพรเทเจเพื่อกำหนดเงื่อนไขจำเป็นให้กับคลาส.....107
รูปที่ 5.21 หน้าจอโปรแกรมโพรเทเจเพื่อกำหนดเงื่อนไขจำเป็นและเพียงพอให้กับคลาส..... 107
รูปที่ 5.22 หน้าจอโปรแกรมโพรเทเจหลังการอนุมานออนโทโลจีรวม.....108
รูปที่ 5.23 บางส่วนของออนโทโลจีรวมจากการรวมระหว่างคลาสของคลาส Children และคลาสของคลาส Baby ในโลคัลออนโทโลจีที่1 และ 2 หลังการกำหนดเงื่อนไขและการอนุมาน.....108
รูปที่ 5.24 ตัวอย่างฟังก์ชันการแปลงค่าจาก “F” เป็น “female” และ “M” เป็น “male”.....111
รูปที่ 5.25 ตัวอย่างฟังก์ชันการแปลงค่าข้อมูลจากหน่วยเซนติเมตรเป็นหน่วยเมตร..... 111
รูปที่ 5.26 ตัวอย่างวิว Student ที่กำกับลงบนฐานข้อมูลเชิงวัตถุชุดที่หนึ่ง..... 113
รูปที่ 5.27 ตัวอย่างวิว Student ที่กำกับลงบนฐานข้อมูลเชิงวัตถุชุดที่สอง.....113
รูปที่ 5.28 ตัวอย่างวิว Baby แบบมีเงื่อนไขในข้อคำถามที่กำกับลงบนฐานข้อมูลเชิงวัตถุชุดที่หนึ่ง...113
รูปที่ 5.29 การทำงานของแผ่นแบบการสอบถามข้อมูลผ่านการทำงานของเฟรมเวิร์ค.....114
รูปที่ 5.30 หน้าจอการระบุข้อคำถามของแผ่นแบบการสอบถามข้อมูลผ่านออนโทโลจีรวม.....114
รูปที่ 5.31 ตัวอย่างการระบุข้อคำถามลงในแผ่นแบบการสอบถามข้อมูลผ่านออนโทโลจีรวม.....115
รูปที่ 5.32 ตัวอย่างผลลัพธ์จากการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามในรูปที่ 5.31.....116
รูปที่ 6.1 ผลการทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่หนึ่ง..... 124
รูปที่ 6.2 ผลการทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่สอง..... 125
รูปที่ 6.3 ผลการทดสอบการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามข้อมูล..... 126
รูปที่ 6.4 ผลการทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่หนึ่ง..... 130
รูปที่ 6.5 ผลการทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่สอง..... 130
รูปที่ 6.6 ผลการทดสอบการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามข้อมูล..... 130

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

แบบจำลองข้อมูล (Data Model) [1] เป็นแนวคิดที่นำมาใช้อธิบายโครงสร้างของฐานข้อมูล ได้แก่ ชนิด (Type) ความสัมพันธ์ (Relationship) และเงื่อนไขบังคับ (Constraint) ของข้อมูลในการจัดเก็บข้อมูล โดยอาศัยแนวคิดเชิงตรรกะ (Logical Concept) เช่น อ็อบเจกต์ (Object) พรอพเพอร์ตี (Property) และความสัมพันธ์ระหว่างกัน (Interrelationship) ตัวอย่างเช่น แบบจำลองข้อมูลเชิงวัตถุ (Object-Oriented Data Model) ซึ่งแสดงแทนแนวคิดของฐานข้อมูลในรูปของอ็อบเจกต์ ประกอบด้วย คลาส (Class) แอททริบิวต์ (Attribute) เมท็อด (Method) และความสัมพันธ์ระหว่างคลาส (Class Relationship) เป็นต้น

การรวมแบบจำลองข้อมูล (Data Model Integration) มีจุดประสงค์เพื่อการรวมข้อมูล (Data Integration) [2] ที่จัดเก็บไว้ในแหล่งข้อมูลต่างกันสองแหล่ง และจัดเตรียมให้ผู้ใช้สามารถเข้าถึงข้อมูลเหล่านั้นเสมือนว่าข้อมูลทั้งหมดถูกเก็บไว้ในแหล่งเดียวกัน โดยทั่วไปแล้วการรวมแบบจำลองข้อมูลจะต้องประกอบด้วยสามองค์ประกอบสำคัญคือแบบจำลองข้อมูลที่ต้องการรวมอย่างน้อยสองชุด เรียกว่าแบบจำลองข้อมูลโลคัล (Local Data Model) แบบจำลองข้อมูลรวม (Integrated Data Model) เรียกว่าแบบจำลองข้อมูลโกลบอล (Global Data Model) และแมปปิง (Mapping) ระหว่างแบบจำลองข้อมูลที่ต้องการรวมแต่ละชุดกับแบบจำลองข้อมูลรวม ดังนั้นผู้ทำการรวมแบบจำลองข้อมูลจะต้องระบุได้ว่าชื่อข้อมูลฟิลด์ (Field) ต่างๆ ในแต่ละแบบจำลองข้อมูลโลคัลที่ต้องการรวมมีความสัมพันธ์กันอย่างไร (เช่น ฟิลด์ Student ในแบบจำลองข้อมูลแรกเท่ากับฟิลด์ GradStudent ในแบบจำลองข้อมูลที่สอง) และจะแมป (Map) เป็นชื่อฟิลด์ใดในแบบจำลองข้อมูลรวม (เช่น แมปเป็นฟิลด์ Student) ซึ่งสามารถพิจารณาได้จากวากยสัมพันธ์ (Syntax) และความหมาย (Semantic) ของข้อมูลในแบบจำลองข้อมูลทั้งสองชุด

ทั้งนี้ แบบจำลองข้อมูลที่ดีจะต้องแสดงวากยสัมพันธ์และความหมายของข้อมูลที่จัดเก็บให้เพียงพอต่อการจำแนกความขัดแย้ง (Conflict) ระหว่างแบบจำลองข้อมูลแต่ละชุดและการสร้างแมปปิงที่มีความเหมาะสมที่สุด โดยอาจแบ่งประเภทของความขัดแย้งได้เป็น 2 ระดับ [3] คือ

1. ความขัดแย้งระดับข้อมูล (Data Level Conflict) ตัวอย่างเช่น
   - ความขัดแย้งของค่าข้อมูล (Data Value Conflict) เช่น ค่าข้อมูลของฟิลด์ “Faculty” ในแบบจำลองข้อมูลแรกหมายถึง “Science” และ “Engineer” แต่ในแบบจำลองข้อมูลที่สองหมายถึง “Science”, “Account” และ “Engineer”

- ความขัดแย้งของหน่วยข้อมูล (Data Unit Conflict) เช่น หน่วยข้อมูลของฟิลด์ “Height” ในแบบจำลองข้อมูลแรกจัดเก็บค่าในหน่วยเซนติเมตร แต่ในแบบจำลองข้อมูลที่สองจัดเก็บค่าในหน่วยเมตร

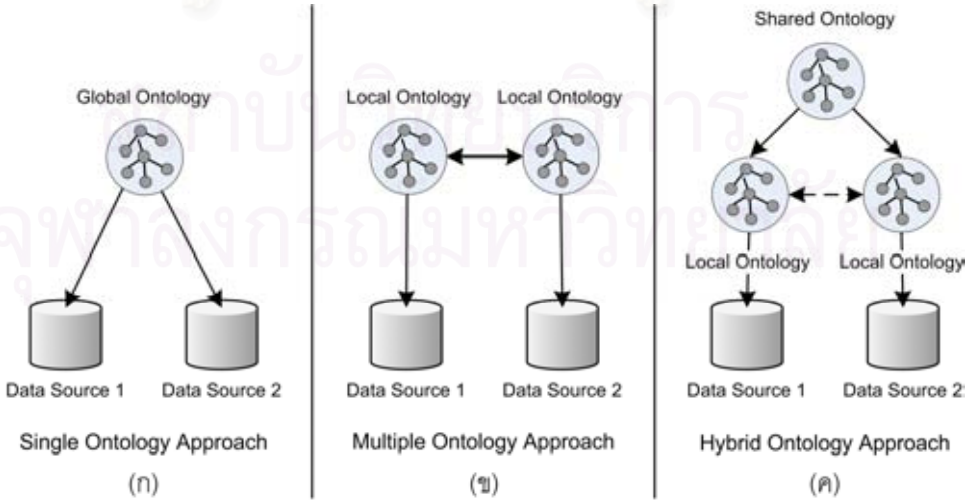
2. ความขัดแย้งระดับสกีมา (Schema Level Conflict) ตัวอย่างเช่น
   - ความขัดแย้งของชื่อ (Naming Conflict) เกิดได้สองลักษณะคือ ชื่อฟิลด์ข้อมูลต่างกันแต่หมายถึงข้อมูลเดียวกัน (Synonyms) เช่น “Student” กับ “GradStudent” หรือชื่อฟิลด์ข้อมูลเดียวกันแต่หมายถึงข้อมูลต่างกัน (Homonyms) เช่น “Address” ในแบบจำลองข้อมูลแรกหมายถึงที่ตั้งของมหาวิทยาลัย แต่ในแบบจำลองข้อมูลที่สองหมายถึงบ้านเลขที่
   - ความขัดแย้งของโครงสร้าง (Structure Conflict) เกิดขึ้นเมื่อข้อมูลเดียวกันถูกเก็บโดยใช้ชนิดของข้อมูลต่างกัน เช่น “Salary” ในแบบจำลองข้อมูลแรกเก็บเป็นจำนวนเต็ม (Integer) แต่ในแบบจำลองข้อมูลที่สองเก็บเป็นจำนวนจริง (Real)
   - ความขัดแย้งของการแอกกรีเกชัน (Aggregation Conflict) เกิดขึ้นเมื่อมีการใช้ความสัมพันธ์แบบแอกกรีเกชันในแบบจำลองข้อมูลหนึ่งแทนกลุ่มของแอททริบิวต์ในอีกแบบจำลองข้อมูล

สำหรับแบบจำลองข้อมูลเชิงวัตถุนั้น แม้ว่าตัวแบบจำลองข้อมูลจะแสดงวากยสัมพันธ์ของข้อมูลอย่างชัดเจน ดังเช่นที่ปรากฏในยูเอ็มแอลคลาสไดอะแกรม (UML Class Diagram) แต่มิได้แสดงถึงความหมายใดๆ ของข้อมูลไว้ภายในเลย ส่งผลให้ไม่สามารถจำแนกความขัดแย้งระดับข้อมูลได้ ดังนั้นการจะรวมแบบจำลองข้อมูลเชิงวัตถุสองชุดเข้าด้วยกันจำเป็นต้องอาศัยวิธีการเพิ่มเติมเพื่อสอดแทรกหรือกำกับความหมายลงบนแบบจำลองข้อมูล

วิธีการหนึ่งที่สามารถนำมาใช้อธิบายความหมายของข้อมูลในแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดที่ต้องการรวมกันทำได้โดยอาศัยแนวคิดของออนโทโลจี (Ontology) [4] ในการกำหนดคำศัพท์ (Vocabulary) สำหรับอธิบายความหมายของคลาส แอททริบิวต์ เมท็อด และความสัมพันธ์ที่ปรากฏในทั้งสองแบบจำลองข้อมูลร่วมกัน รวมไปถึงความหมายอื่นที่ผู้ทำการรวมแบบจำลองข้อมูลอาจต้องการเพิ่มเติมระหว่างการรวมเพื่อช่วยให้แบบจำลองข้อมูลรวมที่ได้มีความถูกต้องมากยิ่งขึ้น

แนวคิดเรื่องออนโทโลจีถูกนำมาใช้งานอย่างแพร่หลายในงานวิจัยด้านการรวมข้อมูลโดยอาศัยความหมาย (Semantic Data Integration) ซึ่งการใช้ออนโทโลจีจะให้ประโยชน์ทั้งในด้านการรวม (Integrate) และการสอบถาม (Query) ระหว่างแหล่งข้อมูลทั้งสอง ด้วยการทำหน้าที่เป็นตัวกลางติดต่อระหว่างผู้ใช้และแหล่งข้อมูล เพื่อจัดการเรื่องความต่างแบบกัน (Heterogeneity) และการกระจายตัว (Distribution) ของข้อมูลแทนผู้ใช้ ในปัจจุบันงานวิจัยด้านการนำออนโทโลจีมาใช้ในการรวมข้อมูลอาจแบ่งออกได้เป็น 3 แนวคิด [5] ตามวิธีการนำออนโทโลจีมาใช้อธิบายแบบจำลองข้อมูลดังรูปที่ 1.1 ได้แก่

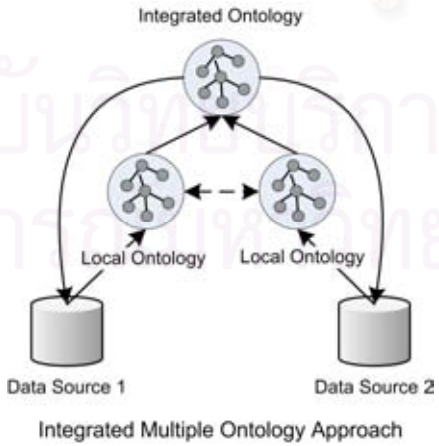
1. แบบออนโทโลจีเดียว (Single Ontology) เป็นแนวคิดที่เสนอให้ใช้ออนโทโลจีหนึ่งชุดกำหนดคำศัพท์เพื่ออธิบายแหล่งข้อมูลทั้งสองร่วมกันเรียกว่า โกลบอลออนโทโลจี (Global Ontology) ตามรูปที่ 1.1(ก) ซึ่งข้อมูลทั้งหมดจากทั้งสองแหล่งข้อมูลจะถูกแมปกับโกลบอลออนโทโลจี จากแนวคิดนี้ ผู้ทำการรวมแบบจำลองข้อมูลอาจสร้างออนโทโลจีชื่อ “Student” ขึ้นมาก่อน แล้วแมปจากฟิลด์ “Student” ในแบบจำลองข้อมูลที่ 1 และฟิลด์ “GradStudent” ในแบบจำลองข้อมูลที่ 2 หากทั้งสองฟิลด์หมายถึงสิ่งเดียวกันไปหาออนโทโลจี “Student” ตัวอย่างเช่น งานวิจัย SIMS [6] และ “Semantic Data Integration in Hierarchical Domains [7]”
2. แบบหลายออนโทโลจี (Multiple Ontologies) เป็นแนวคิดที่เสนอให้อธิบายแบบจำลองข้อมูลแต่ละชุดด้วยออนโทโลจีของตนเองแยกกันเรียกว่า โลคัลออนโทโลจี (Local Ontology) ดังรูปที่ 1.1(ข) แล้วเปรียบเทียบออนโทโลจีที่ได้เพื่อแมประหว่างสองออนโทโลจี จากแนวคิดนี้ผู้ทำการรวมแบบจำลองข้อมูลอาจสร้างออนโทโลจีชื่อ “Student” มาอธิบายฟิลด์ “Student” ในแบบจำลองข้อมูลที่ 1 และสร้างออนโทโลจีชื่อ “GradStudent” มาอธิบายฟิลด์ “GradStudent” ในแบบจำลองข้อมูลที่ 2 แล้วกำหนดว่าออนโทโลจีทั้งสองมีความสัมพันธ์แบบเท่ากัน (Equivalence) ตัวอย่างเช่น งานวิจัย OBSERVER [8]
3. แบบผสม (Hybrid) เป็นแนวคิดที่ผสมผสานระหว่างแนวคิดที่ 1 และ 2 โดยแต่ละแบบจำลองข้อมูลจะถูกอธิบายด้วยโลคัลออนโทโลจีของตนเองเช่นเดียวกับแนวคิดที่ 2 ในขณะเดียวกันจะมีการสร้างแชร์ดออนโทโลจี (Shared Ontology) ขึ้นมาอีกชั้นหนึ่ง เพื่อรองรับการสอบถามจากผู้ใช้และเก็บคำศัพท์ที่ โลคัลออนโทโลจีทั้งสองใช้ร่วมกันสำหรับเปรียบเทียบระหว่างโลคัลออนโทโลจีทำนองเดียวกับโกลบอลออนโทโลจีในแนวคิดที่ 1 แสดงได้ตามรูปที่ 1.1(ค) ตัวอย่างเช่น งานวิจัย COIN [9] และ MECOTA [10]



รูปที่ 1.1 แนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูล [5]

อย่างไรก็ตาม แต่ละแนวคิดต่างมีข้อดีข้อเสียแตกต่างกันออกไป เช่น แนวคิดแบบออนโทโลจีเดียวช่วยให้สอบถามและใช้งานข้อมูลได้สะดวก แต่ขาดความยืดหยุ่นและผู้ออกแบบโกลบอลออนโทโลจีจำเป็นต้องเข้าใจแบบจำลองข้อมูลทั้งสองเป็นอย่างดี ในขณะที่แนวคิดแบบหลายออนโทโลจีมีความยืดหยุ่นมากกว่า แต่มีความยุ่งยากในส่วนของการเปรียบเทียบโลคัลออนโทโลจีและผู้ใช้ไม่สามารถทราบถึงภาพรวมของข้อมูลทั้งหมด ส่วนแนวคิดแบบผสมแม้ว่าจะช่วยลดความยุ่งยากในส่วนการออกแบบออนโทโลจี แต่ในการใช้งานจริงอาจมีปัญหาด้านประสิทธิภาพการทำงาน เนื่องจากทุกครั้งที่มีการเรียกใช้ข้อมูลจะต้องสอบถามข้อมูลสองขั้น คือจากแชร์ดออนโทโลจีไปยังโลคัลออนโทโลจีทุกชุด และจากแต่ละชุดโลคัลออนโทโลจีไปยังข้อมูลที่โลคัลออนโทโลจีนั้นติดต่อ

จากจุดอ่อนที่เกิดขึ้นในการใช้งานออนโทโลจีเพื่อการรวมข้อมูลทั้งสามแนวทางข้างต้น ผู้วิจัยจึงมีแนวคิดที่จะนำข้อดีของแต่ละแนวทางมาประยุกต์ใช้ร่วมกัน เพื่อให้เป็นอีกวิธีการหนึ่งที่สามารถนำไปใช้ในการรวมแบบจำลองข้อมูลโดยอาศัยออนโทโลจีในการอธิบายความหมายของแบบจำลองข้อมูล โดยตั้งชื่อแนวคิดใหม่นี้ว่า “แนวคิดแบบรวมหลายออนโทโลจี (Integrated Multiple Ontologies Approach)” แสดงไว้ตามรูปที่ 1.2 แนวคิดนี้นำข้อดีจากการใช้โลคัลออนโทโลจีในการอธิบายแบบจำลองข้อมูลแต่ละชุด ซึ่งจะช่วยรักษาความหมายที่ปรากฏในแบบจำลองข้อมูลไว้ แล้วนำวิธีการรวมออนโทโลจี (Ontology Integration) [11] มาใช้เปรียบเทียบและรวมโลคัลออนโทโลจีคู่ที่ได้เข้าด้วยกันเรียกว่าออนโทโลจีรวม (Integrated Ontology) ต่อจากนั้นจะนำออนโทโลจีรวมที่ได้ไปแมปกับแบบจำลองข้อมูลทั้งสองชุดในลักษณะเดียวกับการใช้โกลบอลออนโทโลจีในการรวมแบบออนโทโลจีเดียว ด้วยวิธีการนี้จะช่วยให้เจ้าของแหล่งข้อมูลได้รับความสะดวกในการกำหนดโลคัลออนโทโลจีของตนเอง รวมทั้งสะดวกต่อผู้ใช้ในแง่ของการสอบถามข้อมูล เพราะสามารถดำเนินการได้โดยอาศัยการติดต่อกับออนโทโลจีรวมเพียงอย่างเดียว อีกทั้งผู้ใช้ยังเห็นภาพรวมทั้งหมดของข้อมูลผ่านออนโทโลจีรวมด้วย



Integrated Multiple Ontology Approach

รูปที่ 1.2 แนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูลแบบรวมหลายออนโทโลจี

สำหรับขั้นตอนวิธีในการรวมออนโทโลจีนั้น ปัจจุบันมีทั้งงานวิจัยและงานวิจัยเชิงสำรวจ [12-15] ที่เกี่ยวข้องอยู่มากมาย เช่น งานวิจัย OBSERVER [9], PROMPT [16], Chimaera [17] และ FCA-

MERGE [18] อย่างไรก็ตาม งานวิจัยที่เกี่ยวข้องกับขั้นตอนวิธีการรวมออนโลจีเหล่านี้มุ่งเน้นไปที่การรวมออนโทโลจีขนาดใหญ่เพื่ออธิบายข้อมูลที่หลากหลาย เช่น ข้อมูลบนเว็บไซต์ จึงส่งผลให้ความหมายของข้อมูลเหล่านี้ซึ่งอธิบายด้วยออนโทโลจีมีรูปแบบความสัมพันธ์ที่หลากหลายและซับซ้อนตามไปด้วย และทำให้ขั้นตอนวิธีในการรวมออนโทโลจีจำต้องมีความซับซ้อนมาก เช่น มีการนำการประมวลผลภาษาธรรมชาติ (Natural Language Processing) มาใช้ เป็นต้น

ผู้วิจัยเล็งเห็นว่า ในการรวมออนโทโลจีที่อธิบายแบบจำลองข้อมูลนั้น แม้ว่าฐานข้อมูลที่ต้องการรวมจะมีความซับซ้อน แต่โครงสร้างและความสัมพันธ์ที่ปรากฏมีรูปแบบที่ค่อนข้างชัดเจนตายตัว รูปแบบของความหมายมีจำกัด เนื่องจากถูกตีกรอบด้วยตัวแบบจำลองข้อมูลเองจึงน่าจะลดความซับซ้อนของขั้นตอนวิธีลงได้ [19] ในวิทยานิพนธ์นี้ ผู้วิจัยจึงได้นำวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุโดยใช้วิทยา-การศึกษาสำนึก (Heuristics) และการวิเคราะห์ความสัมพันธ์ระหว่างคลาสที่ใช้ในงานวิจัย [20] และแนวทางการแปลงโมเดล (Model) ระหว่างออนโทโลจีและแบบจำลองข้อมูลเชิงวัตถุ [21] มาประยุกต์ใช้ร่วมกัน เพื่อเสนอเป็นวิธีการรวมออนโทโลจีแบบง่ายโดยอาศัยการวิเคราะห์ความสัมพันธ์ระหว่างคลาส เนื่องจากจุดมุ่งหมายของการรวมออนโทโลจีในวิทยานิพนธ์นี้ก็คือการนำมาใช้กับการรวมแบบจำลองข้อมูลเชิงวัตถุนั่นเอง

ดังนั้น หากสามารถรวมโลคัลออนโทโลจีแต่ละชุดเข้าด้วยกันเป็นออนโทโลจีรวมแล้วนำไปใช้กำกับลงบนการรวมแบบจำลองข้อมูลเชิงวัตถุที่แบบจำลองข้อมูลแต่ละชุด จะได้เป็นแนวคิดใหม่ในการนำออนโทโลจีมาใช้ในการรวมข้อมูล ซึ่งจะช่วยลดทั้งความยุ่งยากในการออกแบบออนโทโลจีรวมและความซับซ้อนในการติดต่อระหว่างออนโทโลจีขณะสอบถามข้อมูล โดยอาศัยสมมติฐานว่า หากนำแบบจำลองข้อมูลที่ได้รับการอธิบายและแสดงแทนความหมายในรูปแบบที่เหมาะสมและเพียงพอ มารวมเข้าด้วยกันโดยใช้ขั้นตอนวิธีที่เหมาะสม จะทำให้การรวมแบบจำลองข้อมูลเป็นไปอย่างถูกต้องและครบถ้วน ทั้งนี้เมื่อเปรียบเทียบกับองค์ประกอบทั้งสามในการรวมแบบจำลองข้อมูลแล้ว โลคัลออนโทโลจีและออนโทโลจีรวมจะทำหน้าที่แสดงแทนแบบจำลองข้อมูลโลคัลและแบบจำลองข้อมูลโกลบอล ส่วนการกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูลเชิงวัตถุก็คือส่วนการสร้างแมปปิงระหว่างแบบจำลองข้อมูลโลคัลและแบบจำลองข้อมูลโกลบอล ตามลำดับ

ในส่วนของวิทยานิพนธ์นี้ ได้นำเสนอถึงการนำแนวคิดนี้ไปประยุกต์ใช้งาน พร้อมทั้งเปรียบเทียบข้อแตกต่างระหว่างแนวคิดนี้กับทั้งสามแนวคิดที่มีอยู่เดิม ภายในวิทยานิพนธ์นี้ผู้วิจัยจะนำเสนอวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุ โดยเลือกใช้ออนโทโลจีในการอธิบายแบบจำลองข้อมูลแต่ละชุด เพื่อแสดงแทนความหมายที่ปรากฏในแบบจำลองข้อมูลและความหมายอื่นที่ผู้ทำการรวมแบบจำลองข้อมูลอาจต้องการเพิ่มระหว่างการรวม ซึ่งขั้นตอนนี้จะเปิดให้เจ้าของฐานข้อมูลเป็นผู้สร้างออนโทโลจีขึ้นมาอธิบายแบบจำลองข้อมูลของตน แล้วนำออนโทโลจีทั้งคู่ที่ได้มารวมเข้าด้วยกันโดยอัตโนมัติ ด้วยวิธีการเปรียบเทียบความแตกต่างเพื่อกำหนดค่าความสัมพันธ์ตามวิธีการรวมออนโทโลจีที่ประยุกต์ขึ้น หลังจาก

นั้นจะนำออนโทโลจีรวมที่ได้มาใช้กำกับลงบนแบบจำลองข้อมูลแต่ละชุด และในอนาคตจะสามารถพัฒนาต่อเป็นเฟรมเวิร์คที่สมบูรณ์เพื่อรองรับการสอบถามข้อมูลได้

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อนำเสนอแนวคิดและต้นแบบในการนำออนโทโลจีมาใช้ในการรวมข้อมูลแบบรวมหลายออนโทโลจี
2. เพื่อประยุกต์ใช้แนวคิดแบบรวมหลายออนโทโลจีกับการรวมแบบจำลองข้อมูลเชิงวัตถุ
3. เพื่อกำหนดแนวทางในการอธิบายแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจี และการกำกับออนโทโลจีลงบนแบบจำลองข้อมูลเชิงวัตถุ
4. เพื่อออกแบบวิธีการรวมออนโทโลจีโดยการสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาส
5. เพื่อแสดงวิธีการนำออนโทโลจีรวมที่ได้ไปใช้ในการรวมและการสอบถามข้อมูลจากฐานข้อมูลเชิงวัตถุที่นำมารวม

1.3 ขอบเขตของการวิจัย

1. เสนอแนวทางในการสร้างออนโทโลจีรวมโดยการวิเคราะห์ความสัมพันธ์ระหว่างคลาสของแบบจำลองข้อมูลเชิงวัตถุ โดยประยุกต์จากงานวิจัย [20] ร่วมกับแนวทางการแปลงโมเดลระหว่างออนโทโลจีและแบบจำลองข้อมูลเชิงวัตถุ [21]
2. เสนอแนวทางในการกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูลเชิงวัตถุ
3. พัฒนาต้นแบบระบบการรวมแบบจำลองข้อมูลซึ่งสนับสนุนการสร้างออนโทโลจีรวม การกำกับด้วยออนโทโลจีรวม และการสอบถามข้อมูลผ่านออนโทโลจีรวม
4. การเปรียบเทียบแบบจำลองข้อมูลเชิงวัตถุจะพิจารณาเฉพาะส่วนข้อมูล ได้แก่ แอททริบิวต์ และคุณลักษณะที่เกี่ยวข้องกับแอททริบิวต์ในคลาสเท่านั้น โดยไม่คำนึงถึงส่วนการดำเนินการ (Opetation) ได้แก่ เมท็อด และคุณลักษณะที่เกี่ยวข้องกับเมท็อดในคลาส
5. ข้อมูลความสัมพันธ์ระหว่างคลาสที่จะใช้พิจารณาในการรวมออนโทโลจีเป็นไปตามที่กำหนดในงานวิจัย [20] ได้แก่ข้อมูลเกี่ยวกับ
   - คลาส
   - แอททริบิวต์
   - ชนิดของแอททริบิวต์
   - ความสัมพันธ์แบบสืบทอดคุณลักษณะ
   - ความสัมพันธ์แบบพี่น้อง
   - ความสัมพันธ์แบบแอกกรีเกชัน
   - ความหมายของคลาส
   - ความหมายของแอททริบิวต์
   - ข้อกำหนดบูรณภาพของแอททริบิวต์
   - คำที่มีความหมายเหมือนกัน
   - คำที่เป็นส่วนประกอบกัน

6. ต้นแบบที่พัฒนาสามารถใช้งานในสภาพแวดล้อมของระบบปฏิบัติการไมโครซอฟท์วินโดวส์ เอ็กซ์พี เป็นอย่างน้อย
7. ฐานข้อมูลที่จะนำมารวมจะต้องอยู่ในรูปของแบบจำลองข้อมูลเชิงวัตถุ หรือได้รับการแปลงให้อยู่ในรูปของแบบจำลองข้อมูลเชิงวัตถุแล้วเท่านั้น
8. ฐานข้อมูลที่จะนำมารวมจะต้องเป็นฐานข้อมูลที่เก็บข้อมูลเรื่องเดียวกันหรือมีความเกี่ยวข้องกัน
9. ออนโทโลจีจะได้รับการอธิบายในรูปแบบอาวล์ดีแอล (OWL DL: Description Logic) เป็นอย่างน้อย
10. ในการอนุมานออนโทโลจี ต้นแบบจะพัฒนาโดยใช้จีนาเอพีไอ รุ่นที่ 2.3 เป็นอย่างน้อย
11. การสอบถามข้อมูลจากระบบการรวมแบบจำลองข้อมูลเชิงวัตถุกระทำผ่านแผ่นแบบ (Template) โดยอ้างอิงจากออนโทโลจีที่กำหนดไว้
12. การทดสอบต้นแบบการรวมและการสอบถามข้อมูลที่พัฒนาขึ้นในขั้นแรกจะทดสอบด้วยแบบจำลองข้อมูลที่ใช้ในงานวิจัย [20] เพื่อตรวจสอบความถูกต้องในการรวม เมื่อได้ผลตรงตามต้องการแล้วจะนำไปทดสอบด้วยแบบจำลองข้อมูลที่ถูกใช้งานจริงซึ่งมีความซับซ้อนขึ้น เพื่อหาข้อจำกัดของระบบ
13. วิเคราะห์เปรียบเทียบคุณสมบัติของระบบการใช้ออนโทโลจีในการรวมข้อมูลที่ได้พัฒนาขึ้นตามแนวคิดนี้กับทั้งสามแนวคิดที่มีอยู่เดิม

1.4 วิธีดำเนินการวิจัย

1. ศึกษาทฤษฎีของการรวมสารสนเทศ
2. ศึกษาทฤษฎีของออนโทโลจี การรวมออนโทโลจี อาร์ดีเอฟ (RDF) และการอนุมาน
3. ศึกษาทฤษฎีการรวมแบบจำลองข้อมูลเชิงวัตถุ
4. ศึกษาทฤษฎีการแปลงระหว่างแบบจำลองข้อมูลเชิงวัตถุและออนโทโลจี
5. ออกแบบระบบการรวมออนโทโลจีและการกำกับออนโทโลจีรวมที่สามารถนำไปใช้ในการรวมแบบจำลองข้อมูลเชิงวัตถุ
6. พัฒนาต้นแบบระบบการรวมออนโทโลจีและการกำกับออนโทโลจีรวมเพื่อนำไปใช้ในการรวมแบบจำลองข้อมูลเชิงวัตถุ
7. ทดสอบและปรับปรุงแก้ไขต้นแบบ
8. วิเคราะห์และสรุปผล พร้อมข้อเสนอแนะ
9. จัดทำรายงานวิทยานิพนธ์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้แนวทางใหม่ในการนำออนโทโลจีมาใช้ในการรวมข้อมูล โดยใช้การสร้างออนโทโลจีรวมที่ได้จากการรวมโลคัลออนโทโลจีมาใช้เป็นตัวกลางในการรวมและการสอบถามข้อมูลระหว่างแต่ละฐานข้อมูล เรียกว่าแนวคิดแบบรวมหลายออนโทโลจี
2. ได้ขั้นตอนวิธีการรวมออนโทโลจีโดยการสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาส
3. สามารถอธิบายแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีที่เหมาะสม
4. สามารถนำแนวคิดและขั้นตอนวิธีที่ได้มาพัฒนาเป็นต้นแบบการใช้ออนโทโลจีเพื่อรวมแบบจำลองข้อมูลเชิงวัตถุ โดยใช้การสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาส แล้วนำไปกำกับลงบนแบบจำลองข้อมูลเชิงวัตถุเพื่อใช้ในการสอบถามข้อมูล
5. สามารถเปรียบเทียบข้อดีข้อเสียของแนวทางนี้กับแนวทางที่มีอยู่เดิมได้ ซึ่งแนวทางใหม่นี้สามารถนำไปประยุกต์ใช้กับฐานข้อมูลในองค์กรต่อไปได้

1.6 โครงสร้างวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้จัดแบ่งเนื้อหาออกเป็น 6 ส่วนได้แก่บทที่ 2 ถึงบทที่ 7 ตามลำดับ เริ่มจากในบทที่ 2 เป็นการสรุปถึงทฤษฎีและงานวิจัยต่างๆ ที่เกี่ยวข้องและนำมาประยุกต์ใช้กับส่วนต่างๆ ของวิทยานิพนธ์นี้ บทที่ 3 จะกล่าวถึงแนวคิดและวิธีการสร้างออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุ ซึ่งแบ่งเป็นสองส่วนคือส่วนการสร้างออนโทโลจีเพื่ออธิบายความหมายที่ปรากฏในแบบจำลองข้อมูลเชิงวัตถุ และส่วนการเพิ่มเติมความหมายของแบบจำลองข้อมูลเชิงวัตถุลงในออนโทโลจี เพื่อให้ได้โลคัลออนโทโลจีที่สามารถอธิบายความหมายของแบบจำลองข้อมูลเชิงวัตถุได้อย่างสมบูรณ์ ส่วนบทที่ 4 จะอธิบายถึงขั้นตอนวิธีในการเปรียบเทียบและการรวมออนโทโลจีอย่างละเอียด ซึ่งเป็นการนำโลคัลออนโทโลจีสองชุดที่ได้จากบทที่ 3 มาเปรียบเทียบหาความสัมพันธ์ระหว่างกัน และนำค่าความสัมพันธ์ที่ได้มาวิเคราะห์เพื่อหารูปแบบออนโทโลจีรวมที่เหมาะสม จากนั้นในบทที่ 5 จะนำเสนอเฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีที่ได้พัฒนาขึ้นตามแนวคิดของวิทยานิพนธ์นี้ประกอบด้วย แนวคิดในการออกแบบเฟรมเวิร์ค องค์ประกอบของเฟรมเวิร์ค ได้แก่ ส่วนการสร้างโลคัลออนโทโลจี ส่วนการสร้างออนโทโลจีรวม และส่วนการสร้างแมปปิงระหว่างออนโทโลจีรวมมายังฐานข้อมูลแต่ละชุด รวมถึงแผ่นแบบการสอบถามข้อมูลผ่านการทำงานของเฟรมเวิร์คเพื่อแสดงการนำแนวคิดไปใช้งานจริง ส่วนบทที่ 6 จะแสดงการทดสอบการทำงานเพื่อประเมินความสามารถของเฟรม-เวิร์ค ได้แก่ (1) การทดสอบผลการสร้างออนโทโลจีรวม โดยเลือกนำมาจากผลการทดสอบด้วยแบบจำลองข้อมูลตัวอย่าง 13 คู่ ได้แก่ แบบจำลองข้อมูลเชิงวัตถุที่ใช้ทดสอบในงานวิจัย [20] สิบคู่ และแบบจำลองข้อมูลที่ออกแบบสำหรับใช้งานในโครงการระบบสารสนเทศเพื่อการบริหารของจุฬาลงกรณ์มหาวิทยาลัย (MISCU) จากงานวิจัย [22-25] สามคู่ และ (2) การทดสอบความถูกต้องในการสอบถาม

ข้อมูลผ่านออนโทโลจีรวมที่ได้ไปยังฐานข้อมูลเชิงวัตถุที่นำมารวม ซึ่งผลลัพธ์ที่ได้จะต้องถูกต้องและครบถ้วนเทียบเท่ากับผลรวมจากการสอบถามโดยตรงไปยังฐานข้อมูลแต่ละชุด บทที่ 7 จะเป็นบทสรุปผลการวิจัย การวิเคราะห์เปรียบเทียบคุณสมบัติของแนวคิดนี้กับแนวคิดที่มีอยู่เดิม พร้อมทั้งปัญหาและข้อจำกัดที่พบจากการวิจัย รวมถึงข้อเสนอแนะเพื่อปรับปรุงแก้ไขหรือเป็นแนวทางการทำวิจัยในอนาคต

นอกจากนี้ ในตอนท้ายของวิทยานิพนธ์ยังได้แทรกภาคผนวกไว้ 3 บทเริ่มจากในภาคผนวก ก และ ข จะแสดงตัวอย่างการรวมแบบจำลองข้อมูลสองคู่ด้วยออนโทโลจีที่ได้พัฒนาขึ้นตามแนวคิดของวิทยานิพนธ์นี้ ซึ่งในแต่ละบทจะประกอบด้วยแบบจำลองข้อมูลเชิงวัตถุคู่ที่นำมารวม โลคัลออนโทโลจีสองชุดที่สร้างขึ้นจากแบบจำลองข้อมูลเชิงวัตถุคู่นั้น ออนโทโลจีรวมที่ได้จากการรวมโลคัลออนโทโลจี และตัวอย่างการนำออนโทโลจีรวมมาใช้งานโดยสร้างเป็นวิวกำกับลงบนฐานข้อมูลเชิงวัตถุเพื่อใช้ในการสอบถามข้อมูล ตามลำดับ ส่วนภาคผนวก ง เป็นการรวบรวมบทความที่ได้รับการตีพิมพ์จากวิทยานิพนธ์นี้

1.7 ผลงานที่ตีพิมพ์จากการวิจัย

ส่วนหนึ่งของผลงานจากการวิจัยนี้ได้รับการตีพิมพ์และนำเสนอเป็นบทความทางวิชาการในหัวข้อเรื่อง “An Integration of Data Sources with UML Class Models Based on Ontological Analysis” โดย Manachaya Jamadhvaja และ Twittie Senivongse ในงานประชุมวิชาการ “The ACM Workshop on Interoperability of Heterogeneous Information Systems (IHIS’05)” ซึ่งจัดขึ้นที่เมืองเบรเมน ประเทศสหพันธ์สาธารณรัฐเยอรมนี เมื่อวันที่ 4 พฤศจิกายน 2548

และบางส่วนของผลงานได้รับการตีพิมพ์และนำเสนอเป็นบทความทางวิชาการในหัวข้อเรื่อง “An Approach to Constructing an Integrated Ontology for Integrating Object-Oriented Data Models” โดย Manachaya Jamadhvaja และ Twittie Senivongse ในงานประชุมวิชาการ “The 1st National Conference on Computing and Information Technology (NCCIT 2005)” จัดโดยสถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ระหว่างวันที่ 24-25 พฤษภาคม 2548 ซึ่งบทความนี้ได้รับการตีพิมพ์ซ้ำในวารสารเทคโนโลยีสารสนเทศ ปีที่ 1 ฉบับที่ 2 กรกฎาคม – ธันวาคม 2548 โดยคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ออนโทโลจี (Ontology)

ออนโทโลจี [4] เป็นคำศัพท์ที่ยืมมาจากสาขาวิชาปรัชญา ซึ่งเดิมใช้หมายถึง “รายการอย่างมีแบบแผนของสิ่งที่มีอยู่” (“A Systematic Account of Existence”) โดยคำว่าสิ่งที่มีอยู่ในที่นี้แสดงให้เห็นว่าจะต้องสามารถแสดงแทนสิ่งนั้นด้วยถ้อยคำได้ เมื่อนำออนโทโลจีมาใช้ในทางคอมพิวเตอร์ คำจำกัดความของออนโทโลจีที่ได้รับการยอมรับอย่างแพร่หลายคือ “ข้อกำหนดเกี่ยวกับแนวคิด” (“The Specification of a Conceptualization”) ซึ่งนำมาเป็นข้อกำหนดในการใช้คำศัพท์เพื่อแสดงแทนสิ่งต่างๆ ในโดเมนที่สนใจ (Domain of Interest) ในรูปของอ็อบเจกต์ และความสัมพันธ์ระหว่างอ็อบเจกต์ เช่น การให้นิยามแก่คลาส อินสแตนซ์ (Instance) ความสัมพันธ์ พรอพเพอร์ตี กฎ (Rule) และอ็อบเจกต์อื่นๆ

ออนโทโลจีถูกออกแบบให้รองรับการแสดงแทนความรู้ (Knowledge Representation) สำหรับระบบการทำงานร่วมกัน (Sharing) และการนำกลับมาใช้ใหม่ (Reuse) [26] โดยอธิบายในรูปของข้อความที่คนทั่วไปเข้าใจได้ด้วยภาษาสำหรับแสดงแทนความรู้ (Knowledge Representational Language) ซึ่งมีความชัดเจนและเที่ยงตรงมากกว่าการใช้ภาษาธรรมชาติ (Natural Language) เพื่อให้โปรแกรมสามารถนำออนโทโลจีไปใช้ประโยชน์ได้ ซึ่งการอธิบายนี้จะอาศัยรูปแบบตามโครงสร้างของออนโทโลจีซึ่งประกอบด้วย [4,26]

1. คลาสหรือแนวคิด (Concept) สำหรับแสดงแทนกลุ่มของเอนทิตี (Entity) ในโดเมน สามารถมีสับคลาส (Subclass) เพื่อแสดงแทนแนวคิดที่เฉพาะเจาะจงกว่าของซูเปอร์คลาส (Superclass) ได้ ซึ่งเรียกว่าการแบ่งประเภท (Taxonomy)
2. ความสัมพันธ์หรือพรอพเพอร์ตี (Property) สำหรับแสดงแทนการติดต่อระหว่างคลาสในโดเมน สามารถมีสับพรอพเพอร์ตี (Subproperty) เพื่อแสดงแทนความสัมพันธ์ที่เฉพาะเจาะจงกว่าได้เช่นกัน
3. สัจพจน์ (Axiom) สำหรับกำหนดเงื่อนไขบังคับให้กับคลาส หรือกำหนดอาร์กิวเมนต์ (Argument) ของความสัมพันธ์
4. อินสแตนซ์ (Instance) สำหรับกำหนดอินสแตนซ์ของคลาส

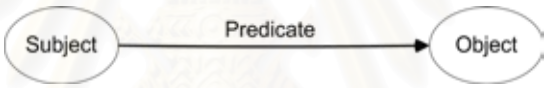
ตัวอย่างเช่น ถ้าต้องการอธิบายว่า “A student has a Name” สามารถทำได้โดยกำหนดให้ “Student” และ “Name” เป็นคลาส กำหนดความสัมพันธ์ระหว่างคลาสทั้งสองเป็น “has a” แล้วเติมอินสแตนซ์ให้กับทั้งสองคลาส อย่างไรก็ตาม แนวคิดถือเป็นเรื่องนามธรรมสำหรับการอธิบายโลกใน

มุมมองอย่างง่าย เพื่อจุดประสงค์ในการใช้งานอย่างใดอย่างหนึ่ง ดังนั้นแต่ละคนอาจสร้างออนโทโลจีที่แตกต่างกันเพื่ออธิบายเรื่องเดียวกัน [27]

2.1.2 อาร์ดีเอฟ (RDF: Resource Description Framework)

อาร์ดีเอฟ [28] เป็นภาษามาตรฐานเชิงความหมาย (W3C Recommendation) เพื่อนำมาใช้เป็นเฟรมเวิร์คในการแสดงแทนทรัพยากร (Resource) อันได้แก่สารสนเทศต่างๆ ที่อยู่บนเว็บ โดยได้รับการออกแบบให้แสดงแทนสารสนเทศได้อย่างยืดหยุ่นและมีเงื่อนไขบังคับน้อยที่สุด แต่ยังคงความสามารถในการนำสารสนเทศเหล่านั้นมาใช้งานร่วมกัน ทั้งนี้จุดมุ่งหมายหลักของภาษาอาร์ดีเอฟก็คือการสร้างแบบจำลองข้อมูลอย่างง่ายที่สามารถใช้งานกับความหมายเชิงรูปนัย (Formal Semantics) และรองรับการอนุมาน (Inference) โดยใช้ไวยากรณ์แบบภาษาเอ็กซ์เอ็มแอล (XML)

โครงสร้างทางไวยากรณ์ของอาร์ดีเอฟจะอธิบายทรัพยากรใดๆ ในรูปของกลุ่มของทริปเปิล (Triple) ซึ่งประกอบด้วยซับเจกต์ (Subject) เพรดิเคท (Predicate) หรือพรอพเพอร์ตี และอ็อบเจกต์ (Object) แสดงอยู่ในรูปของโหนดและเส้นเชื่อมระหว่างโหนด เรียกว่า อาร์ดีเอฟกราฟ (RDF Graph) ดังรูปที่ 2.1 ทั้งนี้ ทิศทางของเส้นเชื่อมมีความสำคัญมากเพราะจะชี้ไปยังปลายทางที่เป็นอ็อบเจกต์เสมอ ส่วนโหนดนั้นสามารถเป็นได้ทั้งซับเจกต์และอ็อบเจกต์



รูปที่ 2.1 อาร์ดีเอฟกราฟ

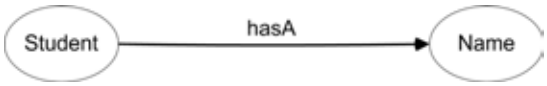
จากอาร์ดีเอฟกราฟจะสามารถบอกได้ว่าซับเจกต์มีความสัมพันธ์บางอย่างกับอ็อบเจกต์โดยความสัมพันธ์ดังกล่าวจะระบุไว้ที่เพรดิเคท

นอกเหนือจากการอธิบายในรูปแบบของอาร์ดีเอฟกราฟ ภาษาอาร์ดีเอฟยังอาจแสดงแทนได้ในรูปของอาร์ดีเอฟทริปเปิล (RDF Triple) ซึ่งนิยมเขียนในรูปของซับเจกต์ เพรดิเคทและอ็อบเจกต์เรียงต่อกันตามลำดับ หรืออาร์ดีเอฟ/ เอ็กซ์เอ็มแอล (RDF/ XML) โดยการแสดงแทนอาร์ดีเอฟตามโครงสร้างพื้นฐานของภาษาเอ็กซ์เอ็มแอล

2.1.3 อาวล์ (OWL: Web Ontology Language)

อาวล์ [29-30] เป็นภาษามาตรฐานเชิงความหมาย (W3C Recommendation) ที่นิยมใช้ประกาศออนโทโลจีเมื่อต้องการให้สามารถนำออนโทโลจีไปใช้งานร่วมกันได้ อาวล์ถูกพัฒนาขึ้นจากภาษาดีเอเอ็มแอลพลัสออยล์ (DAML + OIL: DARPA Agent Markup Language + Ontology Inference Layer) และพัฒนาขยายต่อจากการอธิบายด้วยภาษาอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล จึงทำให้เป็นที่นิยมในการเขียนออนโทโลจีที่แสดงด้วยภาษาอาวล์ในรูปของอาร์ดีเอฟกราฟ ซึ่งสามารถแปลงให้อยู่ในรูปของอาร์ดีเอฟทริปเปิลเพื่อนำไปใช้ได้โดยง่าย ตัวอย่างเช่น การเขียนออนโทโลจีเพื่อแสดงข้อความ “A

Student has a Name." สามารถทำได้ในรูปของอาร์ดีเอฟกราฟ อาร์ดีเอฟทริปเปิลและอาร์ดีเอฟ/ เอ็กซ์เอ็มแอลได้ตามรูปที่ 2.2 ถึงรูปที่ 2.4 ตามลำดับ



รูปที่ 2.2 การเขียนแทนออนโทโลจีในรูปของอาร์ดีเอฟกราฟ

```
<Student> <hasA> <Name>
```

รูปที่ 2.3 การเขียนแทนออนโทโลจีในรูปของอาร์ดีเอฟทริปเปิล

```
<owl:Class rdf:ID="Student"/>
<owl:Class rdf:ID="Name"/>
<owl:ObjectProperty rdf:ID="hasA">
   <rdf:domain rdf:resource="#Student"/>
   <rdf:range rdf:resource="#Name"/>
</owl:ObjectProperty>
```

รูปที่ 2.4 การเขียนแทนออนโทโลจีในรูปของอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล

ทั้งนี้อาวล์ได้รับการออกแบบเพื่อให้เป็นภาษาที่มีความสามารถในการอธิบายคลาสและความสัมพันธ์ระหว่างคลาสอันเป็นส่วนประกอบหลักพื้นฐานของออนโทโลจี เพื่อนำไปใช้งานในเอกสารและแอพลิเคชันบนเว็บ ซึ่งประโยชน์สำคัญที่จะได้รับจากการใช้ภาษาอาวล์คือ

- สามารถแสดงแทนข้อมูลของโดเมนใดๆ โดยการกำหนดเป็นคลาสและความสัมพันธ์ระหว่างคลาสที่เกิดขึ้นในโดเมนนั้น
- สามารถกำหนดอินสแตนซ์ที่เกี่ยวข้องให้กับคลาสและความสัมพันธ์ที่ได้กำหนดไว้ก่อนแล้วได้
- สามารถใส่กฎเพิ่มเติมเพื่อทำการอนุมานเกี่ยวกับคลาสและอินสแตนซ์ได้

นอกจากนี้ จุดเด่นของอาวล์ซึ่งทดแทนข้อจำกัดของอาร์ดีเอฟคือการที่อาวล์อนุญาตให้กำหนดลักษณะเฉพาะของความสัมพันธ์ (Property Characteristic) เช่น เป็นความสัมพันธ์แบบ Transitive, Symmetric, Functional หรือ InverseFunctional และข้อจำกัดของความสัมพันธ์ (Property Restriction) เช่น ข้อจำกัดด้านตัวบ่งปริมาณ (Quantifier Restriction) ข้อจำกัดด้านจำนวน (Cardinality Restriction) ในการอธิบายข้อมูลได้ ทำให้สามารถใช้ภาษาอาวล์ในการอธิบายว่า "นักเรียนแต่ละคนจะต้องมีชื่อเพียงชื่อเดียว" หรือ "นักเรียนทุกคนจะต้องศึกษาในคณะใดคณะหนึ่ง" ได้ โดยทั่วไปแล้ว องค์ประกอบพื้นฐานของอาวล์จะเกี่ยวข้องกับคลาส พรอพเพอร์ตีและอินสแตนซ์ ซึ่งส่วนประกอบที่มีการใช้งานในวิทยานิพนธ์นี้ได้แก่

1. คลาส แสดงด้วยข้อความ Class หรือ rdf:subClassOf เช่น

   ```
   <owl:Class rdf:ID="Student"/>
   ```

2. พรอพเพอร์ตี แสดงด้วยข้อความ ObjectProperty, DatatypeProperty, rdfs:subPropertyOf, rdfs:domain หรือ rdfs:range เช่น

```
<owl:ObjectProperty rdf:ID="hasStudentName">
   <rdf:domain rdf:resource="#Student"/>
   <rdf:range rdf:resource="#StudentName"/>
</owl:ObjectProperty>
```

3. ชนิดของข้อมูล (Datatype) สามารถเลือกกำหนดและใช้ได้ตามชนิดข้อมูลที่มีในเอ็กซ์เอ็มแอลสกีมา (XML Schema) ซึ่งอ้างอิงได้จาก http://www.w3.org/2001/XMLSchema

4. ข้อจำกัดเกี่ยวกับพรอพเพอร์ตี ได้แก่ ข้อจำกัดด้านตัวบ่งปริมาณ allValueFrom ($\forall$) และ someValueFrom ($\exists$) เป็นต้น

5. การสมมูลกัน (Equivalence) ระหว่างคลาส ใช้เมื่อต้องการระบุว่าคลาสทั้งสองคลาสหมายถึงสิ่งเดียวกัน แสดงด้วยข้อความ equivalentClass เช่น

```
<owl:Class rdf:ID="Student">
   <owl:equivalentClass rdf:resource="#GradStudent"/>
</owl:Class>
```

6. ยูเนียน (Union) ใช้เมื่อต้องการระบุว่าคลาสหนึ่งกลุ่มเป็นส่วนประกอบของอีกคลาสหนึ่ง แสดงแทนด้วยข้อความ unionOf เช่น

```
<owl:Class rdf:ID="Address">
   <owl:unionOf rdf:parseType="Collection">
         <owl:Class rdf:about="#HomeNo"/>
         <owl:Class rdf:about="#HomeCity"/>
   </owl:unionOf >
</owl:Class>
```

ในปัจจุบันดับเบิลยูทรีซี (W3C) ซึ่งเป็นผู้กำหนดมาตรฐานภาษาอาวล์ได้แบ่งอาวล์ออกเป็น 3 ระดับตามความสามารถในการอธิบายออนโทโลจีจากน้อยไปมากคือ

1. อาวล์ไลท์ (OWL Lite) เป็นภาษาย่อยระดับต้นของอาวล์ สามารถรองรับการจัดแบ่งลำดับชั้นของคลาส และการใช้งานข้อบังคับเบื้องต้นของอาวล์ได้

2. อาวล์ดีแอล (OWL DL) เป็นภาษาย่อยระดับกลางของอาวล์ โดยดีแอลในที่นี้หมายถึงการอธิบายเชิงตรรกะ (Description Logic: DL) อาวล์ดีแอลสามารถอธิบายออนโทโลจีได้ละเอียดกว่าอาวล์ไลท์ และสามารถรองรับการใส่กฎเพิ่มเติมเพื่อทำการอนุมานได้

3. อาวล์ฟูลล์ (OWL Full) เป็นภาษาย่อยระดับสูงสุดของอาวล์ นิยมใช้เมื่อต้องการอธิบายออนโทโลจีให้มีความละเอียดสูงสุดโดยไม่จำเป็นต้องยึดติดกับไวยากรณ์ของอาร์ดีเอฟอีกต่อไป

2.1.4 การอนุมาน (Inference)

การอนุมาน [31] เป็นคุณสมบัติอย่างหนึ่งของภาษาอาวล์ที่เปิดให้ผู้ใช้สามารถใส่กฎ (Rule) เพิ่มเติมเพื่อให้ได้ข้อเท็จจริง (Fact) ใหม่เกี่ยวกับคลาสและอินสแตนซ์ที่ยังไม่ถูกแสดงไว้ในออนโทโลจีได้ ประโยชน์เพิ่มเติมที่ได้จากการอนุมานคือ ทำให้สามารถตรวจสอบความไม่ต้องกัน (Inconsistency) ที่อาจเกิดขึ้นในการสร้างออนโทโลจี และช่วยให้ทราบว่าคลาสใดเป็นสับคลาสกันบ้าง การอนุมานมาตรฐานที่มีใช้ในภาษาอาวล์ดีแอลมี 4 แบบได้แก่

1. การตรวจสอบ Subsumption (Subsumption Checking) เป็นการทดสอบว่า คลาสใหม่ที่สร้างขึ้นมาภายหลังโดยการกำหนดความหมายของคลาส จะมีคลาสใดที่มีอยู่เดิมเป็นสับคลาส หรือเป็นสับคลาสของคลาสใดที่มีอยู่เดิมบ้าง
2. การตรวจสอบความสมมูล (Equivalence Checking)
3. การตรวจสอบความต้องกัน (Consistency Checking) เป็นการทดสอบว่าคลาสสามารถมีอินสแตนซ์ได้จริงโดยไม่ขัดแย้งกับคลาสอื่นหรือไม่
4. การตรวจสอบการสร้างกรณีตัวอย่าง (Instantiation Checking)

อย่างไรก็ตาม หลักการอนุมานที่ใช้ในออนโทโลจีและในภาษาอาวล์เป็นการอนุมานแบบเปิด (Open-World Reasoning) ซึ่งมีความแตกต่างจากการอนุมานแบบปิด (Closed-World Reasoning) ที่ใช้ในงานวิจัยด้านอื่นๆ ลักษณะสำคัญของการอนุมานแบบเปิดคือ ทุกข้อความจะเป็นจริงจนกว่าจะได้รับการกำหนดหรือสามารถพิสูจน์ว่าข้อความนั้นเป็นเท็จ (ในขณะที่การอนุมานแบบปิดจะให้ทุกข้อความที่ไม่ถูกกำหนดว่าเป็นจริงเป็นข้อความเท็จ) ดังนั้นการใช้ออนโทโลจีเพื่อแสดงแทนความหมายใดๆ จำเป็นต้องกำหนดเงื่อนไขให้ชัดเจนเพื่อป้องกันความผิดพลาดของการอนุมานในภายหลัง

2.1.5 เมตาโมเดลของนิยามของออนโทโลจี (Ontology Definition MetaModel)

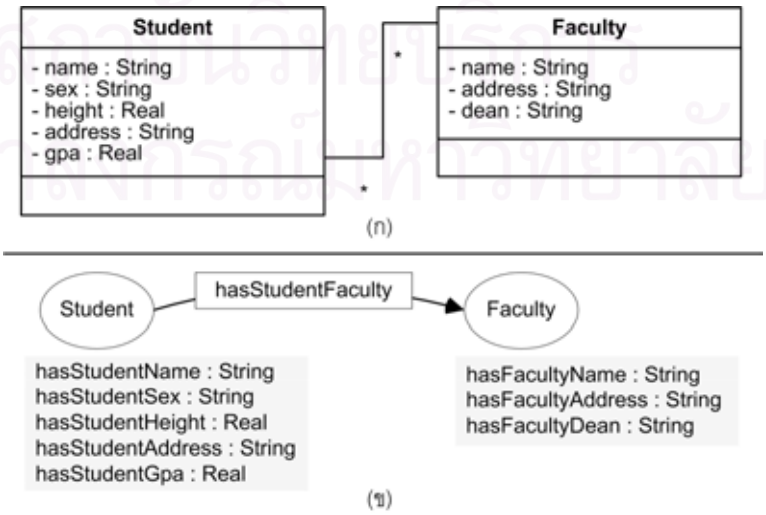
ตามที่โอเอ็มจี (OMG: Object Management Group) ได้ประกาศอาร์เอฟพี (RFP: Request For Proposal) ให้ผู้สนใจส่งเอกสารข้อเสนอเรื่อง Ontology Definition MetaModel (ODM) เพื่อสร้างเป็นข้อกำหนดเรื่องคำนิยามของออนโทโลจี ซึ่งสามารถใช้อธิบายออนโทโลจีด้วยเมตาโมเดลเอ็มโอเอฟรุ่นที่ 2 (MOF2 MetaModel) และนำมาใช้ในการแสดงแทนความรู้ รวมทั้งรองรับการทำงานแบบหลายออนโทโลจีได้ อาร์เอฟพีดังกล่าวได้กำหนดให้เอกสารข้อเสนอแต่ละชุดต้องระบุเมตาโมเดลของนิยามของออนโทโลจี ซึ่งสามารถแสดงความหมายของออนโทโลจีได้ในรูปของยูเอ็มแอล โดยใช้ข้อกำหนดของเมตาโมเดลเอ็มโอเอฟ ยูเอ็มแอล (UML) และโอซีแอล (OCL) รุ่นที่ 2.x รวมทั้งการแมประหว่างเมตาโมเดลกับยูเอ็มแอลโพรไฟล์รุ่นที่ 2 (UML2 Profile) การแมประหว่างโอดีเอ็ม (ODM) กับอาวล์และอื่นๆ

ด้วยข้อกำหนดของโอเอ็มจีนี้ ผู้วิจัยจึงได้นำเอกสารข้อเสนอในส่วนการแสดงความหมายของออนโทโลจีในรูปของยูเอ็มแอลมาใช้ประยุกต์เป็นแนวทางการแปลงระหว่างแบบจำลองข้อมูลเชิงวัตถุกับ

ออนโทโลจีได้ โดยจากผู้ยื่นข้อเสนอสี่ราย ผู้วิจัยเลือกใช้เอกสารข้อเสนอของดีเอสทีซี (DSTC) [21] ในส่วนการเทียบระหว่างออนโทโลจีที่แสดงด้วยภาษาอาวล์บนอาร์ดีเอฟสกีมา (RDF Schema) กับยูเอ็มแอลมาอ้างอิงดังนี้

1. Class ใช้เพื่อกำหนดกลุ่มอ็อบเจกต์ที่เกี่ยวข้องกันเนื่องจากใช้คุณสมบัติบางอย่างร่วมกัน เทียบได้กับคลาสในยูเอ็มแอล
2. rdfs:subClassOf ใช้เพื่อกำหนดสับคลาสให้กับคลาสใดคลาสหนึ่ง เทียบได้กับสับคลาสในยูเอ็มแอล
3. rdfs:Property ใช้แสดงความสัมพันธ์ระหว่างอินสแตนซ์หรือจากอินสแตนซ์ไปยังค่าของข้อมูล แบ่งเป็น
   - ObjectProperty ใช้แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาสสองคลาส เทียบได้กับความสัมพันธ์แบบ Association ในยูเอ็มแอล
   - DatatypeProperty ใช้แสดงความสัมพันธ์จากอินสแตนซ์ของคลาสไปยังค่าของข้อมูล เทียบได้กับแอททริบิวต์ในยูเอ็มแอล
4. rdfs:subProperty ใช้เพื่อกำหนดสับพรอพเพอร์ตีให้กับพรอพเพอร์ตีที่มีอยู่ เทียบได้กับความสัมพันธ์แบบ Sub-association หรือ Sub-type ในยูเอ็มแอล
5. rdfs:domain ใช้เพื่อกำหนดว่าพรอพเพอร์ตีที่สร้างขึ้นใช้งานกับอ็อบเจกต์ในโดเมนใดได้บ้าง เทียบได้กับต้นทางความสัมพันธ์แบบ Association ของคลาสในยูเอ็มแอล
6. rdfs:range ใช้เพื่อกำหนดว่าพรอพเพอร์ตีที่สร้างขึ้นใช้งานกับอ็อบเจกต์ในเรนจ์ใดได้บ้าง เทียบได้กับปลายทางความสัมพันธ์แบบ Association ของคลาสในยูเอ็มแอล

ตัวอย่างการแปลงแบบจำลองข้อมูลเชิงวัตถุที่แสดงด้วยยูเอ็มแอลคลาสไดอะแกรมให้แสดงในรูปของออนโทโลจีแสดงไว้ในรูปที่ 2.5



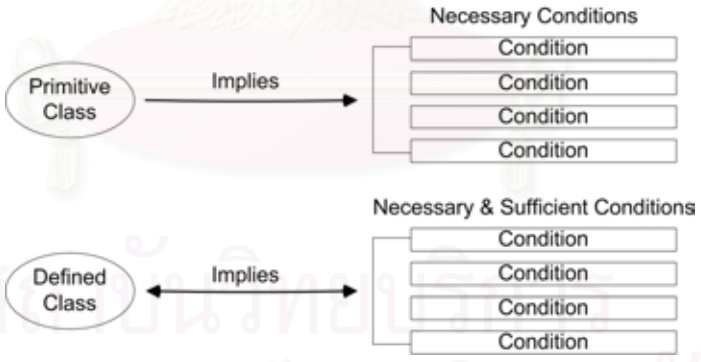
รูปที่ 2.5 ตัวอย่างการแปลงแบบจำลองข้อมูลเชิงวัตถุ (ก) เป็นออนโทโลจีในรูปของอาร์ดีเอฟกราฟ (ข)

2.1.6 โพรเทเจ (Protégé)

โพรเทเจ [32-33] เป็นเครื่องมือสำหรับสร้างออนโทโลจีเพื่อใช้เป็นแบบจำลองข้อมูลของโดเมนหรือแอพพลิเคชันเชิงฐานความรู้ใดๆ (Knowledge-Based Application) โดยมีโพรเทเจ-อาวล์ (Protégé-OWL) ซึ่งเป็นส่วนขยายของโพรเทเจทำหน้าที่สนับสนุนการใช้งานอาวล์ในการสร้างออนโทโลจี

ตามข้อกำหนดจากดับเบิลยูทรีซี อาวล์ออนโทโลจีที่สร้างขึ้นควรจะประกอบด้วยการอธิบายคลาส ความสัมพันธ์ระหว่างคลาสและอินสแตนซ์ที่เกี่ยวข้อง และเมื่อนำออนโทโลจีที่สร้างขึ้นมาพิจารณาร่วมกับ ความหมายเชิงรูปนัยจากอาวล์ (OWL Formal Semantics) ยังช่วยบอกให้ทราบได้ว่าจะได้รับข้อมูลอื่นๆ เพิ่มขึ้นตามมาได้อย่างไร เช่น การหาข้อเท็จจริงที่ไม่ได้แสดงโดยตรงในออนโทโลจีที่สร้าง แต่เป็นผลมาจากความหมายอื่นประกอบกัน ทั้งนี้ ความหมายของคลาสที่กำหนดในโพรเทเจ-อาวล์สามารถแบ่งได้เป็น 2 ประเภท [33-34] แสดงไว้ตามรูปที่ 2.6 คือ

1. คลาสแบบ Primitive (Primitive Class) หรือ Partial Class กำหนดโดยใช้เงื่อนไขจำเป็น (Necessary Condition) เพื่อระบุว่าอินสแตนซ์ใดจะเป็นสมาชิกของคลาสนั้นได้จะต้องมีคุณสมบัติตรงตามเงื่อนไขเหล่านี้
2. คลาสแบบ Defined (Defined Class) หรือ Complete Class กำหนดโดยใช้เงื่อนไขจำเป็นและเพียงพอ (Necessary & Sufficient Condition) เพื่อระบุว่าอินสแตนซ์ใดจะเป็นสมาชิกของคลาสนั้นได้จะต้องมีคุณสมบัติตรงตามเงื่อนไขเหล่านี้ และในทางกลับกันหากอินส-แตนซ์ใดมีคุณสมบัติตรงตามเงื่อนไขเหล่านี้แสดงว่าอินสแตนซ์นั้นเป็นสมาชิกของคลาส



รูปที่ 2.6 ลักษณะของคลาสแบบ Primitive และคลาสแบบ Defined ที่สร้างจากโพรเทเจ-อาวล์

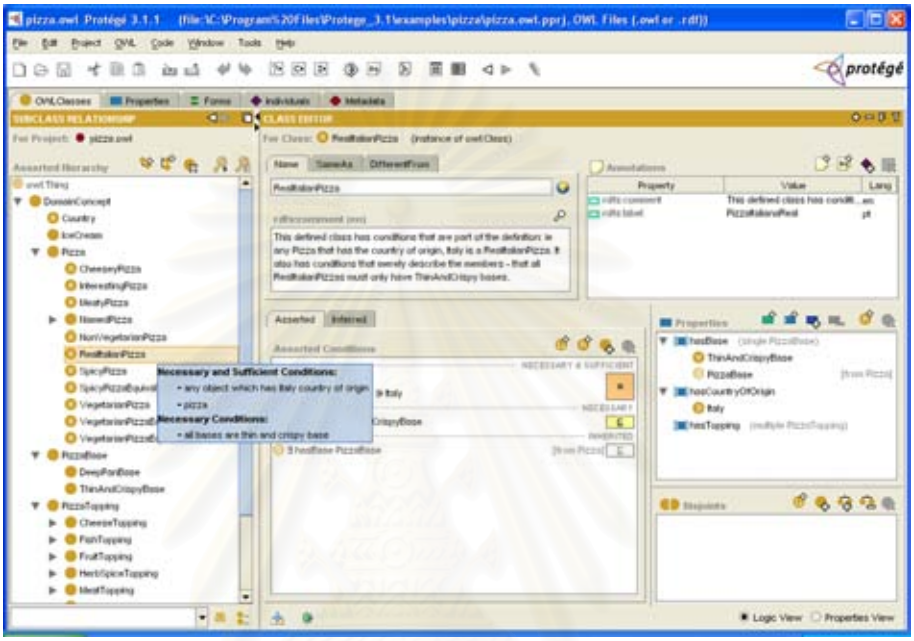
ด้วยความสามารถของโปรแกรมโพรเทเจ-อาวล์ ผู้ใช้จะสามารถสร้างออนโทโลจีได้โดยง่ายและได้รับความสะดวกในการจัดการกับออนโทโลจี ดังนี้

1. เรียกใช้ (Load) และบันทึก (Save) ออนโทโลจีในรูปแบบอาวล์และอาร์ดีเอฟ
2. ตรวจแก้ (Edit) และแสดงผลคลาสและความสัมพันธ์ที่สร้างไว้
3. กำหนดลักษณะเฉพาะเชิงตรรกะของคลาส (Logical Class Characteristic) ในรูปของนิพจน์ภาษาอาวล์ (OWL Expression)

4. เรียกใช้เครื่องมืออนุมาน (Inference Engine) อื่นได้

5. แก้ไขอินสแตนซ์ของอาวล์เพื่อนำไปใช้งานกับเว็บเชิงความหมายได้

ตัวอย่างหน้าจอการทำงานของโปรแกรมโพรเทเจ-อาวล์แสดงไว้ในรูปที่ 2.7 โดยสามารถศึกษาวิธีการใช้งานโปรแกรมโดยละเอียดได้จากเอกสาร “A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools [33]”



รูปที่ 2.7 หน้าจอการทำงานของโปรแกรมโพรเทเจ-อาวล์

2.1.7 จีนา (Jena)

จีนา [35] เป็นส่วนต่อประสานโปรแกรมประยุกต์หรือเอพีไอภาษาจาวา (Java API) เพื่อสร้างและดำเนินการกับอาร์ดีเอฟกราฟ ตัวอย่างเช่น การสร้างอาร์ดีเอฟกราฟ การกำหนดทรัพยากรและพรอพเพอร์ตีให้กับอาร์ดีเอฟกราฟ การแสดงรายการของทรัพยากร การเข้าถึงข้อมูลในอาร์ดีเอฟกราฟ การค้นคืนค่าจากอาร์ดีเอฟกราฟ และการค้นหาอาร์ดีเอฟกราฟ ดังนั้น จีนาจึงได้รับความนิยมนำมาใช้เป็นชุดเครื่องมือ (Toolkit) สำหรับพัฒนาโปรแกรมประยุกต์ในงานด้านเว็บเชิงความหมาย

ถึงแม้ว่าจุดประสงค์หลักของจีนาคือการนำไปใช้งานกับอาร์ดีเอฟก็ตาม จีนายังสามารถทำงานกับออนโทโลจีที่แสดงแทนด้วยภาษาอื่นๆ ได้แก่ อาร์ดีเอฟเอส (RDFS: Resource Description Framework Schema) อาวล์และดีเอเอ็มแอลพลัสออยล์ได้ ซึ่งจีนาจะทำหน้าที่จัดเตรียมตัวต่อประสานการโปรแกรม (Programming Interface) ไปยังออนโทโลจีเพื่อสนับสนุนการพัฒนาโปรแกรมโดยไม่คำนึงว่าออนโทโลจีดังกล่าวถูกแสดงแทนด้วยภาษาใด

นอกจากนี้ จีนายังมีความสามารถรองรับการทำงานของออนโทโลจีร่วมกับฐานข้อมูลเชิงสัมพันธ์ (Relational Database) [36] และการอนุมานโดยใช้กฎ [37] ได้อีกด้วย

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยเกี่ยวกับการรวมแบบจำลองข้อมูลเชิงวัตถุ

งานวิจัยเกี่ยวกับการรวมแบบจำลองข้อมูลเชิงวัตถุมีอยู่มากมายในปัจจุบัน ซึ่งแต่ละงานวิจัยมีจุดเด่นแตกต่างกันไปขึ้นกับการนำไปใช้งาน เนื่องจากผู้วิจัยต้องการนำขั้นตอนวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุมาประยุกต์ใช้เป็นขั้นตอนวิธีการรวมออนโทโลจี ซึ่งขั้นตอนวิธีที่ต้องการควรจะครอบคลุมถึงการวิเคราะห์ในเชิงความหมายและความสัมพันธ์ระหว่างคลาสอันเป็นลักษณะสำคัญของออนโทโลจี รวมทั้งจะส่งผลต่อส่วนของการออกแบบออนโทโลจีเพื่อแสดงแบบจำลองข้อมูลเชิงวัตถุ และขั้นตอนวิธีการรวมออนโทโลจี โดยผู้วิจัยได้เลือกงานวิจัย [20] ซึ่งมีรายละเอียดดังนี้

2.2.1.1 วิธีการรวมแบบจำลองข้อมูลเชิงวัตถุโดยใช้วิทยาการศึกษาสำนึกและการวิเคราะห์ความสัมพันธ์ระหว่างคลาส [20]

งานวิจัยนี้นำเสนอวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุหนึ่งคู่ โดยพิจารณาจากความสัมพันธ์ระหว่างคลาสและความหมายที่เพิ่มให้กับแบบจำลองข้อมูล แล้วกำหนดเป็นค่าความสัมพันธ์เพื่อวิเคราะห์ว่าแบบจำลองข้อมูลรวมควรเป็นอย่างไร หลังจากนั้นจะนำแบบจำลองข้อมูลรวมที่ได้จากวิธีการนี้ไปเปรียบเทียบกับแบบจำลองข้อมูลรวมที่สร้างขึ้นโดยผู้เชี่ยวชาญ งานวิจัยได้นำแนวคิดเรื่องค่าวิทยาการศึกษาสำนึก (Heuristics) มาใช้ปรับหาค่าความสัมพันธ์ที่เหมาะสมเพื่อกำหนดให้กับความสัมพันธ์แต่ละรูปแบบในการรวมแบบจำลองข้อมูลว่าควรเป็นค่าใด โดยมีขั้นตอนดังนี้

1. ขั้นตอนการเตรียมแบบจำลองข้อมูล ประกอบด้วย 2 ขั้นตอนย่อยคือ
   - เพิ่มความหมายให้กับแบบจำลองข้อมูล ได้แก่ การเพิ่มความหมายให้กับคลาส และการเพิ่มความหมายให้กับแอททริบิวต์
   - ระบุคำที่มีความหมายเหมือนกันหรือคำที่เป็นส่วนประกอบกัน
2. ขั้นตอนการเปรียบเทียบแบบจำลองข้อมูล ประกอบด้วย 3 ขั้นตอนย่อยคือ
   - เปรียบเทียบความหมายของคลาสหรือความหมายแอททริบิวต์ซึ่งได้ระบุเพิ่มเติมไว้ในขั้นตอนที่ 1 โดยคิดจากชื่อของความหมายและค่าของความหมายที่กำหนดให้กับคลาสหรือแอททริบิวต์แต่ละคู่ทีละคู่จนครบ ผลที่ได้จากขั้นตอนนี้คือระดับความสัมพันธ์ระหว่างคลาสหรือแอททริบิวต์แต่ละคู่
   - เปรียบเทียบแอททริบิวต์ พิจารณาจากชื่อของแอททริบิวต์ ความหมายของแอททริบิวต์ ชนิดข้อมูลของแอททริบิวต์ ข้อกำหนดบูรณภาพของแอททริบิวต์ (Integrity Constraint) และระดับความสัมพันธ์ระหว่างแอททริบิวต์ที่ได้จากขั้นตอนก่อนหน้านี้ ผลที่ได้จากขั้นตอนนี้คือระดับความเหมือนหรือแตกต่างเมื่อเทียบกับแอททริบิวต์อื่น และคำแนะนำว่าสามารถรวมแอททริบิวต์นั้นเข้ากับแอททริบิวต์อื่นได้หรือไม่

- เปรียบเทียบคลาส กำหนดค่าความสัมพันธ์ระหว่างคลาสโดยพิจารณาได้จากระดับความสัมพันธ์ระหว่างคลาสจากขั้นตอนการเปรียบเทียบความหมาย ร่วมกับระดับความเหมือนหรือแตกต่างระหว่างชุดแอททริบิวต์ในคลาสนั้นที่ได้จากขั้นตอนการเปรียบเทียบแอททริบิวต์ ผลที่ได้คือค่าความสัมพันธ์ระหว่างคลาส ซึ่งในที่นี้จะแบ่งเป็น 5 รูปแบบคือ แบบเท่ากัน (Equivalence) แบบซูเปอร์คลาส (Superclass) แบบสับคลาส (Subclass) แบบพี่น้อง (Sibling) และแบบแตกต่างกัน (Disjoint)

3. ขั้นตอนการรวมแบบจำลองข้อมูล ประกอบด้วย 3 ขั้นตอนย่อยคือ
   - รวมความหมายของคลาสหรือแอททริบิวต์ โดยการรวมความหมายของคลาสจะเกิดขึ้นเมื่อคลาสคู่นั้นมีความสัมพันธ์ระหว่างคลาสแบบเท่ากัน แบบซูเปอร์คลาสหรือแบบสับคลาส ส่วนการรวมความหมายของแอททริบิวต์จะเกิดขึ้นเมื่อมีการรวมแอททริบิวต์คู่ใดเข้าด้วยกัน ซึ่งความหมายรวมที่ได้จะต้องเป็นผลจากการยูเนียน (Union) ระหว่างความหมายเดิม
   - รวมแอททริบิวต์ ทำได้โดยการรวมความหมายของแอททริบิวต์ตามขั้นตอนก่อนหน้านี้ (ซึ่งในบางกรณีอาจต้องสร้างฟังก์ชันเพื่อแปลงความหมายด้วย) ตรวจสอบชนิดข้อมูลของแอททริบิวต์และสร้างฟังก์ชันเพื่อปรับให้ตรงกัน แล้วพิจารณาข้อกำหนดบูรณภาพของแอททริบิวต์ เพื่อหาค่าที่เป็นได้ทั้งหมดของแอททริบิวต์ พร้อมทั้งพิจารณาว่ามีแอททริบิวต์ใดหรือไม่ที่ค่าของแอททริบิวต์ขึ้นกับแอททริบิวต์ที่ถูกแปลงค่า หากมีจะต้องเพิ่มฟังก์ชันการแปลงค่าแอททริบิวต์นั้นด้วย
   - รวมคลาส พิจารณาจากค่าความสัมพันธ์ระหว่างคลาสทั้ง 5 รูปแบบที่ได้จากการเปรียบเทียบคลาสและลำดับความสำคัญของความสัมพันธ์ที่คลาสหนึ่งในแบบจำลองข้อมูลหนึ่งมีผลต่อคลาสต่างๆ ในอีกแบบจำลองหนึ่ง โดยมีลำดับการรวมคลาสเริ่มจากคลาสที่มีความสัมพันธ์แบบเท่ากัน แบบซูเปอร์คลาส / แบบสับคลาส และแบบพี่น้องตามลำดับ ส่วนความสัมพันธ์แบบแตกต่างกันจะแสดงว่าทั้งสองคลาสไม่เกี่ยวข้องกันจึงไม่นำมารวม จากนั้นจึงพิจารณาจากในแบบจำลองข้อมูลรวมที่ได้ว่า มีคลาสคู่ใดที่มีความสัมพันธ์แบบแอกกรีเกชัน (Aggregation) ปรากฏอยู่หรือไม่ ถ้ามีจะทำการรวมแอททริบิวท์ในคลาสเหล่านั้นที่มีความเกี่ยวข้องกันเข้าด้วยกันแทน

ผลลัพธ์ที่ได้จากงานวิจัยนี้คือข้อเสนอแนะว่า ในการรวมแบบจำลองข้อมูลเชิงวัตถุสองชุดเข้าด้วยกัน ควรจะได้แบบจำลองข้อมูลรวมเป็นอย่างไร แล้วจึงนำคำแนะนำดังกล่าวมาสร้างเป็นวิว (View) บนฐานข้อมูลเชิงวัตถุเดิม เพื่อเป็นส่วนติดต่อระหว่างผู้ที่ต้องการใช้งานฐานข้อมูลทั้งสองร่วมกัน กับการสอบถามไปยังฐานข้อมูลจริง

ผู้วิจัยได้นำวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุจากงานวิจัยนี้มาประยุกต์ใช้กับออนโทโลจี เพื่อพัฒนาเป็นวิธีการรวมออนโทโลจีซึ่งจะนำมาใช้กับงานของผู้วิจัย

2.2.2 งานวิจัยเกี่ยวกับการนำออนโทโลจีมาใช้ในการรวมแบบจำลองข้อมูล

งานวิจัยเกี่ยวกับการนำออนโทโลจีมาใช้ในการรวมแบบจำลองข้อมูลอาศัยความสามารถของออนโทโลจีในการทำหน้าที่เป็นตัวกลาง (Mediator) ติดต่อระหว่างผู้ใช้กับแบบจำลองข้อมูลหรือระหว่างแบบจำลองข้อมูลแต่ละชุดกับแบบจำลองข้อมูลรวมหรือทั้งสองหน้าที่ร่วมกัน ในงานวิจัย [5] ได้แบ่งแนวทางในการนำออนโทโลจีมาใช้เพื่อการรวมข้อมูลออกเป็น 3 แนวทางตามวิธีการนำออนโทโลจีมาใช้อธิบายแบบจำลองข้อมูลคือ แนวคิดแบบออนโทโลจีเดียว แนวคิดแบบหลายออนโทโลจี และแนวคิดแบบผสม ตัวอย่างงานวิจัยที่นำแนวคิดทั้งสามไปใช้ ได้แก่

2.2.2.1 Semantic Data Integration in Hierarchical Domains [7]

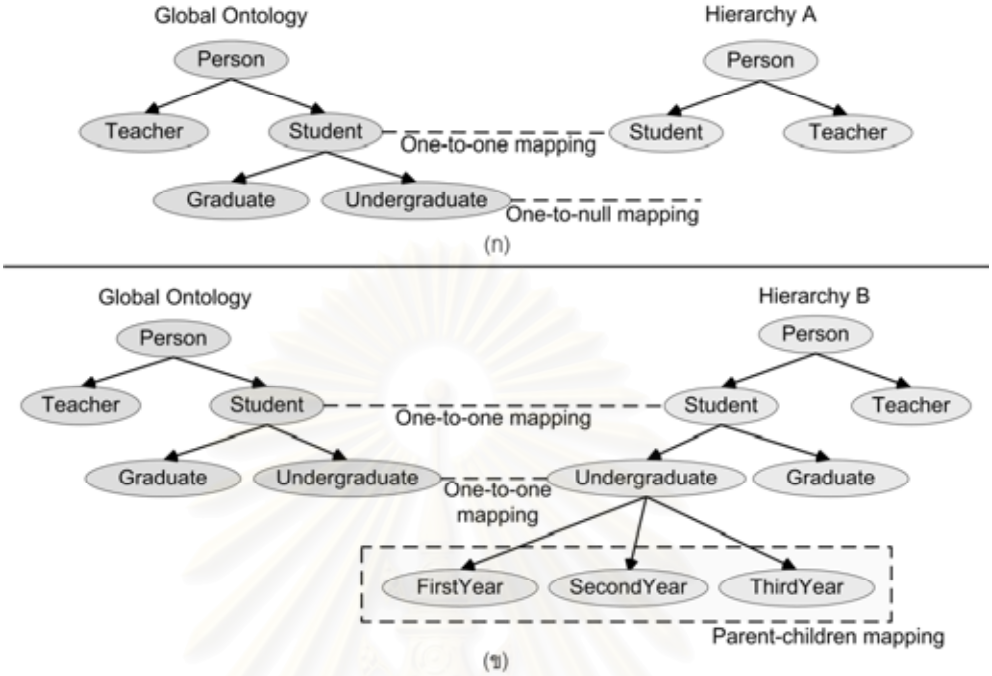
งานวิจัยนี้เป็นตัวอย่างการนำแนวคิดแบบออนโทโลจีเดียวมาใช้ โดยลักษณะสำคัญของแนวคิดแบบออนโทโลจีเดียวคือ การสร้างโกลบอลออนโทโลจีหนึ่งชุดเพื่อใช้อธิบายแบบจำลองข้อมูลทั้งหมดร่วมกัน มีจุดประสงค์สำคัญในการรวมข้อมูลจากหลายแหล่งข้อมูล และนำมาแสดงให้ผู้ใช้เห็นเสมือนว่าข้อมูลทั้งหมดถูกจัดเก็บไว้ในแหล่งข้อมูลเดียว โดยซ่อนความแตกต่างอันเป็นผลจากการออกแบบแหล่งข้อมูลจากผู้ใช้งานข้อมูล งานวิจัยนี้ยกตัวอย่างการจัดการแหล่งข้อมูลที่แสดงแทนด้วยเอกสารเอ็กซ์เอ็มแอล (XML: eXtensible Markup Language) แบบโดเมนเชิงลำดับชั้น (Hierarchical Domain) ซึ่งการจัดเรียงจะแตกต่างกันไปในแต่ละแหล่งข้อมูล ด้วยแนวคิดนี้ข้อมูลแต่ละฟิลด์ในทุกแหล่งข้อมูลจะถูกแมปไปยังโกลบอลออนโทโลจีที่สร้างขึ้นที่ส่วนกลาง

งานวิจัยนี้เสนอแนวคิดในการแมประหว่างแหล่งข้อมูลกับโกลบอลออนโทโลจีดังรูปที่ 2.8 เริ่มต้นจากการสร้างโกลบอลออนโทโลจีขึ้นมาหนึ่งชุด หากลำดับชั้น (Hierarchy) ใดของข้อมูลตรงกับที่กำหนดไว้ในโกลบอลออนโทโลจี การแมปจะเกิดขึ้นแบบหนึ่งต่อหนึ่ง (One-to-one Mapping) (รูปที่ 2.8(ก) และ (ข)) หากไม่มีลำดับชั้นตามที่โกลบอลออนโทโลจีกำหนดไว้ การแมปจะเป็นแบบหนึ่งต่อศูนย์ (One-to null Mapping) (รูปที่ 2.8(ก)) และหากลำดับชั้นใดของข้อมูลมิได้ถูกกำหนดไว้ในโกลบอลออนโทโลจี จะกำหนดให้แมปไปยังลำดับชั้นที่สูงกว่าแทน เรียกว่าการแมปแบบ Parent-children (Parent-children Mapping) (รูปที่ 2.8(ข))

งานวิจัยนี้แสดงให้เห็นได้อย่างชัดเจนว่า แม้แนวคิดแบบออนโทโลจีเดียวจะทำให้ง่ายต่อการจัดการคือมีผู้ออกแบบออนโทโลจีหนึ่งคนสร้างโกลบอลออนโทโลจีไว้ แล้วให้เจ้าของข้อมูลแต่ละคนเลือกแมปข้อมูลของตนไปยังออนโทโลจีดังกล่าว แต่การออกแบบโกลบอลออนโทโลจีเป็นไปได้ยาก เนื่องจากจำเป็นต้องเข้าใจภาพรวมของข้อมูลทั้งหมดเพื่อให้ออนโทโลจีที่ได้ถูกต้อง ครบถ้วน และสามารถรองรับการนำไปใช้ได้จริง นอกจากนั้นการแมปแบบ Parent-children แต่ละครั้งจะทำให้สูญเสียความหมายของข้อมูลในลำดับชั้นนั้นไปทันที

ดังนั้น แม้ว่าผู้วิจัยจะเลือกนำวิธีการแมปจากโกลบอลออนโทโลจีไปยังแหล่งข้อมูลมาใช้เป็นต้นแบบในการแมปจากออนโทโลจีรวมที่สร้างขึ้นไปยังแบบจำลองข้อมูลก็ตาม ผู้วิจัยยังให้

ความสำคัญกับที่มาของออนโทโลจีรวมที่จะต้องสร้างขึ้นเพื่อให้ครอบคลุมความหมายทั้งหมดของแบบจำลองข้อมูลเดิมทุกตัว และมิให้ความหมายของข้อมูลในลำดับชั้นใดๆ สูญหายระหว่างการรวม



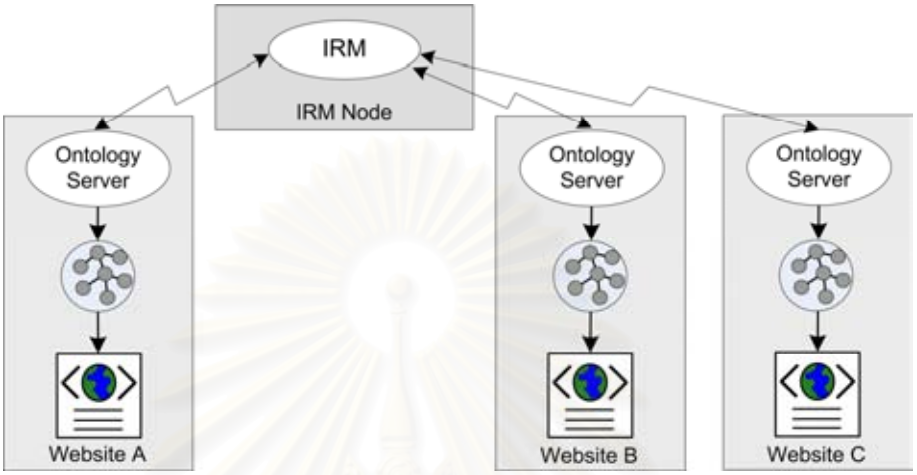
รูปที่ 2.8 ตัวอย่างการแมปจากลำดับชั้นข้อมูลไปยังโกลบอลออนโทโลจี [7]

2.2.2.2 OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies [8]

งานวิจัย OBSERVER (Ontology Based System Enhanced with Relationships for Vocabulary hEterogeneity Resolution) เป็นตัวอย่างการนำแนวคิดแบบหลายออนโทโลจีมาใช้ โดยลักษณะสำคัญของแนวคิดแบบหลายออนโทโลจีคือ การสร้างโลคัลออนโทโลจีหนึ่งชุดมาใช้อธิบายแบบจำลองข้อมูลแต่ละชุดแยกกัน แล้วสร้างแมปปิงระหว่างออนโทโลจีทั้งหมด ภายในงานวิจัยนี้นำเสนอวิธีการประมวลผลข้อคำถามในระบบสารสนเทศขนาดใหญ่ที่ข้อมูลถูกจัดเก็บในรูปของเว็บเพจ (Web Page) ซึ่งไม่มีการจัดการสารสนเทศแบบรวมศูนย์ (Centralized Information Management) ดังนั้นข้อมูลแต่ละชุดเป็นอิสระต่อกันอย่างสิ้นเชิง การใช้ออนโทโลจีช่วยให้สามารถสอบถามข้อมูลด้วยคำถามในเชิงความหมายได้เพิ่มจากการใช้คำหลัก (Keyword) เพื่อสอบถามตามแบบเดิม

ในส่วนแนวคิดแบบหลายออนโทโลจี งานวิจัยนี้เปิดให้แต่ละเว็บสร้างออนโทโลจีของตนเองอย่างอิสระโดยมีออนโทโลจีเซิร์ฟเวอร์ (Ontology Server) คอยควบคุม และในระบบจะมีตัวจัดการความสัมพันธ์ระหว่างออนโทโลจี (Interontology Relationship Manager (IRM)) ทำหน้าที่เป็นสารบัญความหมาย (Catalog of Semantics) จากทุกออนโทโลจี โดยผู้ใช้จะติดต่อผ่านออนโทโลจีที่เลือกใช้ (Selected Ontology) ชุดใดชุดหนึ่งจากออนโทโลจีทั้งหมด หลังจากนั้น เมื่อผู้ใช้สอบถาม

ไปยังออนโทโลจีที่เลือก ข้อคำถามจะถูกสร้างขึ้นโดยใช้คำศัพท์ของออนโทโลจีนั้น ก่อนจะตรวจสอบไปที่ตัวจัดการความสัมพันธ์ระหว่างออนโทโลจี เพื่อแปลเป็นข้อคำถามที่ใช้คำศัพท์ของออนโทโลจีปลายทางที่ต้องการ (Target Ontology) ตัวอย่างการทำงานของระบบ OBSERVER แสดงไว้ในรูปที่ 2.9



รูปที่ 2.9 องค์ประกอบของระบบตามงานวิจัย OBSERVER [8]

ในงานวิจัย OBSERVER นี้ ตัวจัดการความสัมพันธ์ระหว่างออนโทโลจี (ไออาร์เอ็ม) ตามรูปที่ 2.9 เป็นส่วนประกอบสำคัญที่ใช้ในการรวมออนโทโลจีโดยสร้างเป็นการสอบถามข้ามออนโทโลจี เนื่องจากเป็นส่วนที่เก็บคุณสมบัติเชิงความหมายที่กำหนดระหว่างออนโทโลจีแต่ละชุด โดยทำหน้าที่จัดการกับปัญหาเรื่องการใช้คำศัพท์ที่แตกต่างกันในแต่ละออนโทโลจี ซึ่งภายในงานวิจัยนี้เลือกแก้ปัญหาความสัมพันธ์ระหว่างคำศัพท์ 6 รูปแบบ ได้แก่

1. ความสัมพันธ์แบบความหมายเหมือนกัน (Synonym Relationship)
2. ความสัมพันธ์แบบคำลูกกลุ่ม (Hyponym Relationship)
3. ความสัมพันธ์แบบคำแม่กลุ่ม (Hypernym Relationship)
4. ความสัมพันธ์แบบซ้อนเหลื่อมกัน (Overlap Relationship)
5. ความสัมพันธ์แบบแตกต่างกัน (Disjoint Relationship)
6. ความสัมพันธ์แบบครอบคลุม (Covering Relationship)

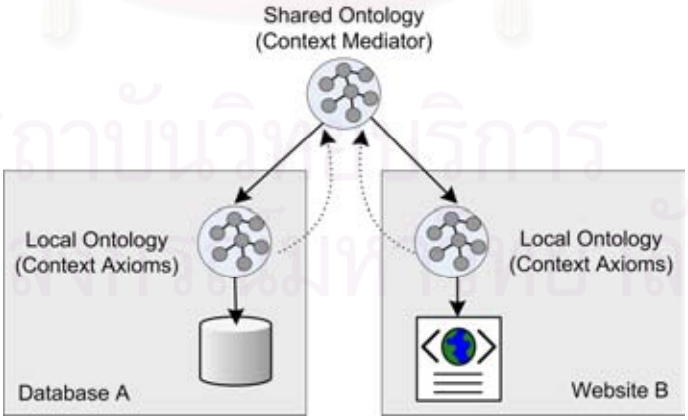
ผู้วิจัยเลือกนำเฉพาะแนวคิดการสร้างออนโทโลจีหนึ่งชุดสำหรับอธิบายข้อมูลแต่ละชุดจากงานวิจัยนี้ไปใช้ แต่มิได้นำขั้นตอนวิธีในการรวมออนโทโลจีโดยสร้างเป็นการสอบถามข้ามออนโทโลจี เนื่องจากเล็งเห็นว่าการสร้างสารบัญความหมายเพื่อเก็บความสัมพันธ์ระหว่างคำศัพท์ทั้งหมดจะทำให้ระบบยากต่อการพัฒนา และเกิดปัญหาในการใช้งานหากออนโทโลจีเปลี่ยนแปลง นอกจากนี้งานวิจัยนี้เลือกพิจารณาเฉพาะความสัมพันธ์ด้านความหมายของคำศัพท์เท่านั้น ดังนั้นหากข้อมูลที่

เก็บมีเงื่อนไขบังคับอื่น ตัวอย่างเช่น หน่วยการนับที่ต่างกัน เช่น แต่ละเว็บอาจแสดงหน่วยเงินเป็นดอลลาร์ เป็นปอนด์ หรือเป็นบาทไม่ตรงกัน ซึ่งอาจทำให้ผลการรวมคลาดเคลื่อนได้

2.2.2.3 Context Interchange: New Features and Formalisms for the Intelligent Integration of Information [9]

งานวิจัยนี้เป็นตัวอย่างการนำแนวคิดในการรวมข้อมูลโดยใช้ออนโทโลจีแบบผสมมาใช้ โดยลักษณะสำคัญของแนวคิดแบบผสมคือ การสร้างโลคัลออนโทโลจีหนึ่งชุดมาใช้อธิบายแบบจำลองข้อมูลแต่ละชุดแยกกัน แล้วสร้างแชร์ดออนโทโลจีขึ้นมาอีกชั้นหนึ่งเพื่อเก็บคำศัพท์ที่โลคัลออนโทโลจีแต่ละชุดใช้ร่วมกัน เฟรมเวิร์ค COIN (COntext INterchange) ทำงานในลักษณะของการสับเปลี่ยนบริบทโดยอาศัยแนวคิดเชิงตัวกลาง (Mediator-based) เพื่อจัดการด้านความหมายระหว่างข้อมูลต้นทางและปลายทางที่แตกต่างกัน โดยมุ่งเน้นการแก้ปัญหาความแตกต่างด้านความหมายของข้อมูลที่ไม่ได้รับการกำหนดไว้ล่วงหน้าโดยอาศัยการพิจารณาจากบริบท (Context) ของข้อมูล ดังนั้นเฟรมเวิร์คนี้ต้องมีความสามารถในการตรวจหาและปรับความแตกต่างที่พบในการใช้งานระหว่างข้อมูลแต่ละชุดไม่ว่าจะเป็นข้อมูลในแหล่งข้อมูลหรือข้อมูลจากเว็บก็ตาม

ภายในเฟรมเวิร์ค COIN (รูปที่ 2.10) จะมีตัวกลางด้านบริบท (Context Mediator) อยู่ภายในแชร์ดออนโทโลจีทำหน้าที่เปรียบเทียบสัจพจน์ของบริบท (Context Axiom) ซึ่งอยู่ในโลคัลออนโทโลจีแต่ละชุดที่ต้องการแลกเปลี่ยนข้อมูลกัน เมื่อมีการส่งการสอบถามเข้ามาในระบบตัวกลางด้านบริบทจะเข้ามารับข้อคำถามนั้นเพื่อแปลงให้อยู่ในรูปของการสอบถามของตัวกลาง (Mediator Query) แล้วจึงแปลเป็นแผนการสอบถาม (Query Plan) อีกต่อหนึ่ง ก่อนที่จะเรียกให้แผนการสอบถามทำงาน เพื่อส่งการสอบถามย่อยไปยังแต่ละระบบ เก็บผลลัพธ์ที่ได้ แปลงรูปแบบของผลลัพธ์ให้เหมาะสมและส่งคำตอบรวมที่ได้กลับไปยังผู้ใช้



รูปที่ 2.10 องค์ประกอบของเฟรมเวิร์คของงานวิจัย COIN [9]

ข้อแตกต่างสำคัญของงานวิจัยนี้กับงานของผู้วิจัยคือ วิธีที่ใช้ในการบอกความแตกต่างระหว่างโลคัลออนโทโลจี ซึ่งงานวิจัยนี้เลือกใช้วิธีการเปรียบเทียบบริบทของข้อมูลที่แชร์ดออนโทโลจีและจะเปรียบเทียบเพื่อแปลความหมายทุกครั้งที่ได้รับข้อคำถาม ในขณะที่งานของผู้วิจัยเลือก

เปรียบเทียบเพียงครั้งเดียว และนำผลที่ได้มาสร้างเป็นออนโทโลจีรวมเพื่อเก็บความแตกต่างไว้ล่วงหน้า เพื่อตัดขั้นตอนการเปรียบเทียบที่อาจใช้เวลานานออกไป

2.2.3 งานวิจัยเกี่ยวกับการรวมออนโทโลจี

งานวิจัยเกี่ยวกับการรวมออนโทโลจีได้รับความสนใจอย่างแพร่หลายในปัจจุบันตามความนิยมในการนำออนโทโลจีมาใช้แสดงแทนข้อมูลเกี่ยวกับโดเมนต่างๆ ที่สนใจ และความต้องการใช้ข้อมูลที่ได้รับการแสดงแทนด้วยออนโทโลจีร่วมกัน โดยการรวมออนโทโลจี [11] เป็นกระบวนการในการสร้างออนโทโลจีหนึ่งชุดเพื่อแสดงแทนเรื่องใดเรื่องหนึ่งขึ้นจากออนโทโลจีตั้งแต่หนึ่งชุดขึ้นไปที่แสดงแทนข้อมูลในต่างเรื่องกัน ทั้งนี้การรวมออนโทโลจีอาจประกอบด้วย [12] การแมปออนโทโลจี (Ontology Mapping) ซึ่งเป็นขั้นตอนการเชื่อมโยงคำศัพท์จากต่างออนโทโลจีที่ใช้งานในโดเมนเดียวกัน และการผสานออนโทโลจี (Ontology Merging) ซึ่งเป็นขั้นตอนการรวมหลายออนโทโลจีที่แสดงแทนข้อมูลเรื่องเดียวกันเข้าด้วยกันเป็นออนโทโลจีเดียว

ในงานวิจัย [13] ได้แบ่งแนวทางในการรวมออนโทโลจีออกเป็น 2 รูปแบบคือ แนวคิดแบบใช้การสอบถามข้ามออนโทโลจี เช่น งานวิจัย OBSERVER [9] ที่ได้กล่าวไปแล้วในหัวข้อที่ 2.2.2.2 กับแนวคิดแบบที่มีการสร้างออนโทโลจีรวมขึ้นจริง เช่น งานวิจัย PROMPT [16], Chimaera [17] และ FCA-MERGE [18] เนื่องจากผู้วิจัยเลือกใช้การรวมออนโทโลจีตามแนวคิดแบบที่มีการสร้างออนโทโลจีรวมขึ้นจริง ภายในหัวข้อนี้จึงนำเสนอเฉพาะตัวอย่างงานวิจัยที่แสดงการนำแนวคิดนี้ไปใช้เท่านั้น ได้แก่

2.2.3.1 PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment [16]

งานวิจัย PROMPT เป็นตัวอย่างหนึ่งของงานวิจัยเรื่องการรวมออนโทโลจีแบบสร้างเป็นออนโทโลจีรวม โดยมีจุดประสงค์ในการรวม (Merging) ออนโทโลจีของสารสนเทศทั้งหมดที่ต้องการให้เป็นออนโทโลจีเดียวหรือปรับแนว (Alignment) ให้ออนโทโลจีทั้งหมดอธิบายข้อมูลไปในทางเดียวกันหรือต่อเนื่องกันด้วยแนวทางแบบกึ่งอัตโนมัติ นั่นคืองานวิจัย PROMPT จะทำงานบางส่วนเองโดยอัตโนมัติ และให้คำแนะนำแก่ผู้รวมออนโทโลจีเพื่อจัดการงานส่วนที่เหลือซึ่งต้องการการตัดสินใจจากผู้รวม

การทำงานของ PROMPT เริ่มจากให้ผู้รวมออนโทโลจีป้อนออนโทโลจีคู่ที่ต้องการรวมเข้าสู่โปรแกรมเพื่อให้โปรแกรมสร้างรายการเริ่มต้นโดยการจับคู่จากชื่อคลาสที่ตรงกัน จากนั้นโปรแกรมจะให้ผู้รวมออนโทโลจีเลือกใช้การดำเนินการอย่างใดอย่างหนึ่งเพื่อรวมหรือปรับออนโทโลจีทั้งคู่เข้าด้วยกัน แล้วโปรแกรมจะดำเนินการตามที่ผู้ใช้เลือก หลังจากนั้นโปรแกรมจะตรวจสอบความขัดแย้ง (Conflict) ที่เกิดขึ้นจากการดำเนินการ และสร้างคำแนะนำใหม่เพิ่มให้กับผู้รวมออนโทโลจีโดยพิจารณาจากการดำเนินการล่าสุด เมื่อได้รับคำแนะนำจากโปรแกรม ผู้รวมออนโทโลจีจะวนกลับไปเลือกใช้การดำเนินการอย่างหนึ่งจากรายการคำแนะนำอีกครั้ง และโปรแกรมก็จะดำเนินการเช่นเดิม

ซ้ำเป็นรอบๆ จนกว่าผู้รวมออนโทโลจีจะเลือกหยุดการดำเนินการ ผลลัพธ์ที่ได้จากโปรแกรม PROMPT ก็คือคำแนะนำว่าผู้รวมออนโทโลจีควรสร้างออนโทโลจีรวมอย่างไร

ตัวอย่างการดำเนินการที่ให้ผู้รวมออนโทโลจีเลือกใช้ได้แก่ การรวมคลาส การรวมสลอต (Slot) หรือความสัมพันธ์ระหว่างคลาส การรวมแบบผูกมัด (Binding) ระหว่างสลอตและคลาส คัดลอกคลาสทั้งคลาสจากออนโทโลจีหนึ่งไปยังอีกออนโทโลจี คัดลอกบางส่วนของคลาส เป็นต้น ส่วนตัวอย่างความขัดแย้งที่โปรแกรมตรวจสอบได้แก่ ชื่อขัดแย้ง การอ้างอิงไปยังจุดที่ไม่มีจริง ความซ้ำซ้อนในลำดับชั้นของคลาส เป็นต้น

การทำงานของ PROMPT เป็นการทำงานเชิงโต้ตอบ (Interactive) ระหว่างโปรแกรมกับผู้รวมออนโทโลจี ถือเป็นแนวทางหนึ่งในการรวมออนโทโลจีขนาดใหญ่ อย่างไรก็ตามการทำงานของ PROMPT ต้องใช้ผู้รวมออนโทโลจีซึ่งมีความเชี่ยวชาญเรื่องออนโทโลจีที่จะรวมเป็นอย่างดี จึงจะตัดสินใจได้ว่าควรเลือกดำเนินการอะไรบ้างกับการรวม ถือเป็นข้อแตกต่างสำคัญจากงานของผู้วิจัยที่ต้องการให้การทำงานเป็นไปอย่างอัตโนมัติมากที่สุดเพื่อลดงานของผู้รวมออนโทโลจีลง

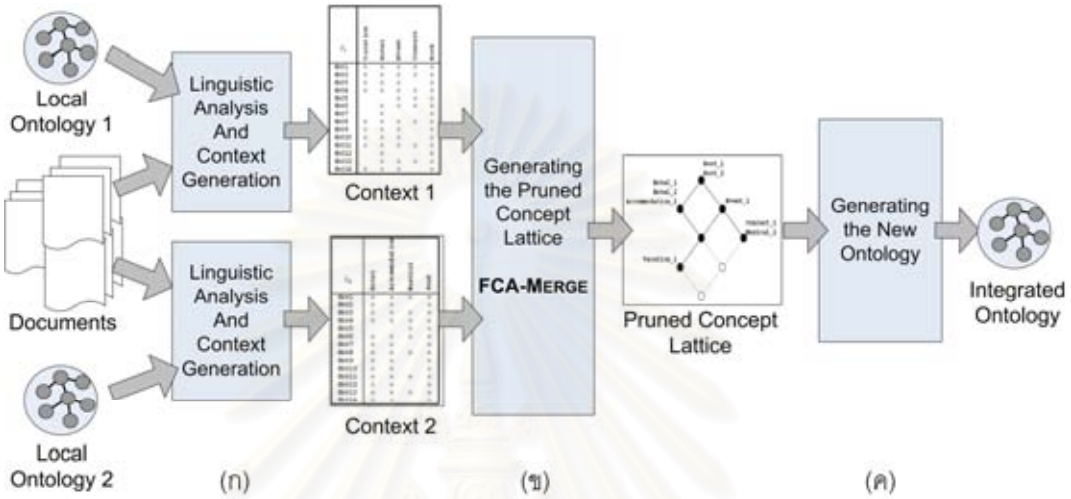
2.2.3.2 FCA-MERGE: Bottom-Up Merging of Ontologies [18]

งานวิจัย FCA-MERGE เป็นตัวอย่างหนึ่งของงานวิจัยเรื่องการรวมออนโทโลจีแบบสร้างเป็นออนโทโลจีรวม โดยขั้นตอนวิธีที่ใช้ในงานวิจัยนี้นำกลวิธีจากการประมวลผลภาษาธรรมชาติ (Natural Language Processing) และการวิเคราะห์แนวคิดเชิงรูปนัย (Formal Concept Analysis) มาใช้เปรียบเทียบออนโทโลจีที่ต้องการรวม แล้วแสดงข้อมูลทั้งหมดในรูปของโครงร่างของแนวคิด (Concept Lattice) เพื่อนำมาสร้างเป็นออนโทโลจีรวมต่อไป

ขั้นตอนการสร้างออนโทโลจีรวมในงานวิจัยนี้แบ่งการทำงานออกเป็น 3 ขั้นตอนแสดงไว้ในรูปที่ 2.11 ดังนี้

1. ขั้นตอนการวิเคราะห์ทางภาษาและการสร้างบริบท (รูปที่ 2.11(ก)) โดยนำเอกสาร (Document) ที่มีเนื้อหาเกี่ยวข้องกับออนโทโลจีทั้งสองชุดจำนวนหนึ่งกับออนโทโลจีแต่ละชุดมาเข้ากระบวนการทางภาษา เพื่อสร้างบริบท (Context) ขึ้นมาใช้กำกับออนโทโลจีแต่ละชุด เนื่องจากเอกสารที่นำมาใช้กับออนโทโลจีทั้งสองชุดเป็นเอกสารเดียวกัน ดังนั้นบริบทที่กำกับให้กับออนโทโลจีทั้งสองชุดจะมีความสัมพันธ์กัน
2. ขั้นตอนการสร้างโครงร่างของแนวคิด (รูปที่ 2.11(ข)) ถือเป็นขั้นตอนสำคัญในงานวิจัยนี้ เป็นการนำบริบทของออนโทโลจีที่ได้จากขั้นตอนแรก ซึ่งอธิบายเรื่องเดียวกันด้วยคำศัพท์จากสองออนโทโลจีมาสร้างขึ้นเป็นโครงร่างของแนวคิด เพื่อแสดงความสัมพันธ์ระหว่างออนโทโลจีทั้งสอง
3. ขั้นตอนการสร้างออนโทโลจีใหม่จากโครงร่างของแนวคิด (รูปที่ 2.11(ค)) โดยพิจารณาจากความสัมพันธ์ที่ปรากฏในโครงร่างดังกล่าว

ในการใช้งานจริง วิธีการที่ใช้ในงานวิจัย FCA-MERGE สามารถทำงานได้แบบอัตโนมัติ ผู้รวมออนโทโลจีมีหน้าที่กำหนดออนโทโลจีที่ต้องการรวม และเอกสารที่ใช้ในการสร้างบริบทให้โปรแกรมเท่านั้น แต่ความยุ่งยากของงานวิจัยนี้คือผู้รวมออนโทโลจีจำเป็นต้องเลือกใช้เอกสารที่เหมาะสมและเพียงพอต่อการกำหนดบริบทให้กับทั้งสองออนโทโลจี ออนโทโลจีรวมที่ได้จึงจะมีความถูกต้อง



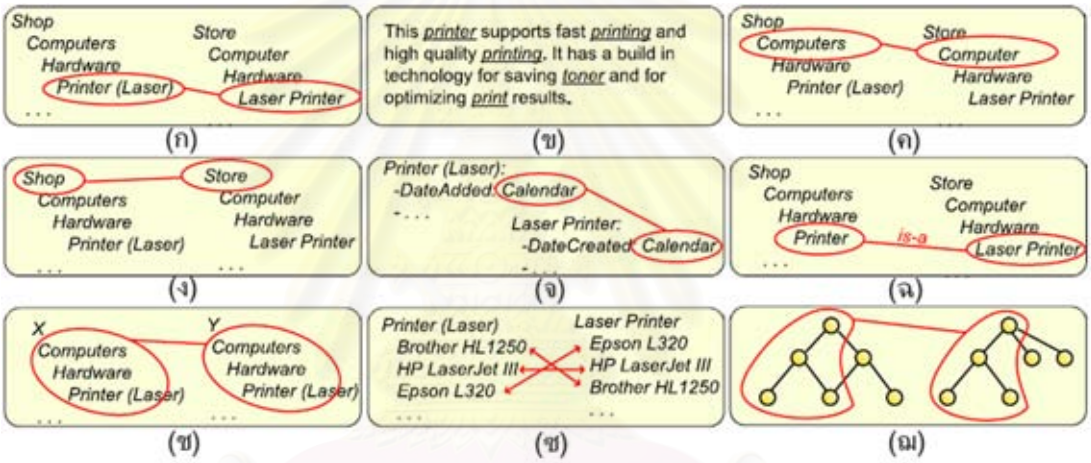
รูปที่ 2.11 ขั้นตอนการทำงานเพื่อสร้างออนโทโลจีรวมของงานวิจัย FCA- MERGE [18]

2.2.3.3 Identifying ontology integration methods and their applicability in the context of product classification and knowledge integration task [15]

งานวิจัยนี้ [15] ทำการสำรวจถึงวิธีการรวมออนโทโลจีจากงานวิจัยเกี่ยวกับแนวคิดในการรวมออนโทโลจีที่มีอยู่ในปัจจุบันกว่า 50 งานวิจัย และให้ข้อสรุปว่าสามารถจำแนกวิธีการพิจารณาในการรวมออนโทโลจีที่นิยมใช้งานภายในงานวิจัยต่างๆ ออกเป็น 9 วิธี (ตามรูปที่ 2.12) ดังต่อไปนี้

1. ความคล้ายคลึงของข้อความ (Text Similarity) เป็นวิธีที่ตรวจสอบจากความคล้ายคลึงของข้อความระหว่างสองออนโทโลจี โดยพิจารณาว่าชื่อหรือคำอธิบายในออนโทโลจีเหมือนกันหรือไม่ (รูปที่ 2.12(ก))
2. การสกัดคำหลัก (Keyword Extraction) เป็นการตรวจหาคำหลักจากคำอธิบายของคลาสเพื่อหาความหมายของคลาสและความสัมพันธ์กับคลาสอื่น (รูปที่ 2.12(ข))
3. วิธีการทางภาษา (Language-Based Methods) เช่น การพิจารณาจากข้อความที่อยู่ติดกัน การตัดคำทิ้งคำไม่สำคัญ (รูปที่ 2.12(ค))
4. การระบุความสัมพันธ์ระหว่างคำ (Identification of Word Relations) เป็นการนำทรัพยากรทางภาษาศาสตร์ เช่น คลังศัพท์ (Lexicon) มาตรวจหาคำพ้อง (รูปที่ 2.12(ง))
5. ความคล้ายคลึงของชนิดข้อมูล/ โดเมนและเรนจ์ (Type Similarity/ Domains and Ranges) เป็นการตรวจสอบว่าแอททริบิวต์ใดมีชนิดข้อมูลเหมือนกันบ้าง (รูปที่ 2.12(จ))

6. การวิเคราะห์การสืบทอดคลาส (Class Inheritance Analysis) เป็นการพิจารณาการสืบทอดระหว่างคลาส เพื่อหาความสัมพันธ์ชนิดอิสอะ (Is-A) (รูปที่ 2.12(ฉ))
7. การวิเคราะห์โครงสร้าง/ โครงสร้างของการแบ่งประเภท (Structure Analysis/ Taxonomic Structure) เป็นการระบุลักษณะร่วมกันของคลาสโดยพิจารณาจากแอททริบิวต์ของคลาสและการเชื่อมระหว่างคลาส (รูปที่ 2.12(ช))
8. การตีความข้อมูล/ คุณสมบัติหลัก (Data Interpretation/ Key Properties) ถ้าออนโทโลจีที่ต้องการรวมแสดงข้อมูลของอินสแตนซ์รวมอยู่ด้วย สามารถพิจารณาความคล้ายคลึงของคลาสได้หากอินสแตนซ์ของของสองคลาสเป็นตัวเดียวกัน (รูปที่ 2.12(ซ))
9. แมปปิงของกราฟ (Graph-Mapping) เป็นการใช้ลักษณะของส่วนที่เหมือนกันในกราฟของออนโทโลจีเพื่อระบุว่าโครงสร้างของออนโทโลจีสองชุดคล้ายคลึงกัน (รูปที่ 2.12(ฌ))



รูปที่ 2.12 วิธีการรวมออนโทโลจีที่ใช้งานในแนวคิดเกี่ยวกับการรวมออนโทโลจี [15]

สำหรับขั้นตอนวิธีในการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นนั้น เมื่อนำมาวิเคราะห์ตามวิธีการรวมออนโทโลจีที่งานวิจัยนี้ได้จำแนกไว้ ผู้วิจัยพบว่าส่วนการเปรียบเทียบออนโทโลจีภายในงานของผู้วิจัยประกอบด้วยการพิจารณาโดยใช้วิธีตรวจสอบความคล้ายคลึงของข้อความ การระบุความสัมพันธ์ระหว่างคำ ความคล้ายคลึงของชนิดข้อมูล และการวิเคราะห์โครงสร้าง

บทที่ 3
การสร้างออนโทโลจีเพื่ออธิบายและเพิ่มความหมาย ให้กับแบบจำลองข้อมูลเชิงวัตถุ

ในทางทฤษฎีแล้ว การสร้างออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุสามารถทำได้โดยไม่มีข้อจำกัดหรือข้อบังคับใดๆ ซึ่งการเปิดให้สร้างออนโทโลจีโดยอิสระส่งผลให้ออนโทโลจีที่ได้มีความหลากหลาย ความหมายของแบบจำลองข้อมูลเชิงวัตถุที่แสดงในออนโทโลจีแต่ละชุดอาจไม่ครบถ้วน ไม่ตรงกัน หรือไม่เพียงพอต่อการจำแนกความขัดแย้งระหว่างแบบจำลองข้อมูลขึ้นกับความสนใจของผู้สร้างออนโทโลจี ทำให้ยากต่อการเปรียบเทียบและการรวมออนโทโลจี

ผู้วิจัยจึงได้กำหนดแนวทางในการสร้างออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุ โดยให้เป็นไปตามโครงสร้างของอัปเปอร์ออนโทโลจี (Upper Ontology) ที่ผู้วิจัยออกแบบขึ้น เพื่อให้ออนโทโลจีที่ได้มีรูปแบบเดียวกันและเหมาะสมกับขั้นตอนวิธีการเปรียบเทียบและการรวมออนโทโลจีที่เลือกใช้ ซึ่งจะส่งผลดีต่อประสิทธิภาพของออนโทโลจีรวมที่ได้ ในบทนี้จะกล่าวถึงแนวคิดและวิธีการสร้างออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุ ซึ่งแบ่งเป็นสามขั้นตอนคือ ส่วนการสร้างออนโทโลจีเพื่ออธิบายความหมายที่ปรากฏในแบบจำลองข้อมูลเชิงวัตถุ ส่วนการเพิ่มเติมความหมายของแบบจำลองข้อมูลเชิงวัตถุลงในออนโทโลจีที่สร้างไว้ และส่วนการสร้างออนโทโลจีที่ระบุความสัมพันธ์ระหว่างออนโทโลจีแต่ละชุดที่สร้างไว้ก่อนแล้ว

3.1 อัปเปอร์ออนโทโลจี

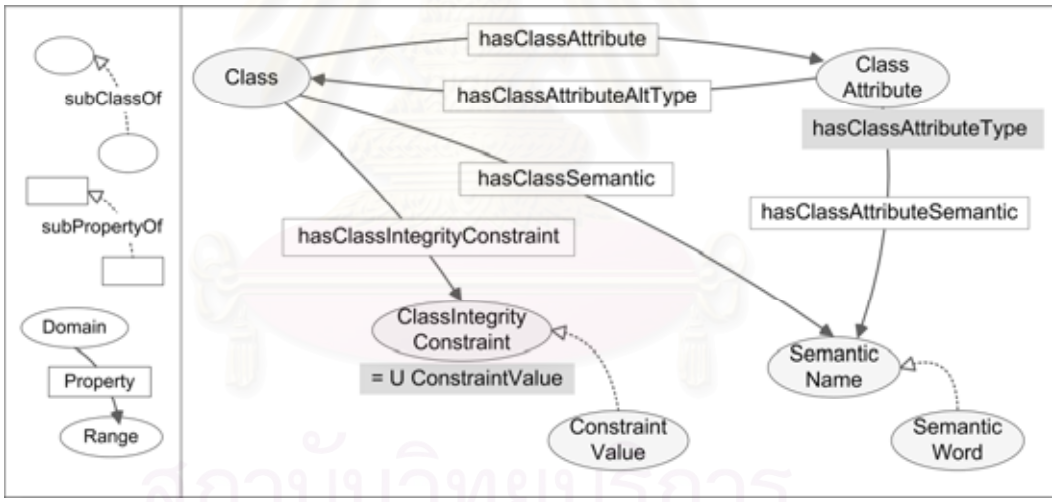
อัปเปอร์ออนโทโลจี [38] เป็นแนวคิดในการสร้างออนโทโลจีพื้นฐานที่จะสามารถนำมาใช้งานร่วมกันระหว่างโดเมน ดังนั้นอัปเปอร์ออนโทโลจีจะมีลักษณะเป็นเมตาออนโทโลจีที่แสดงข้อมูลทั่วไป เช่น โครงสร้าง และแนวคิดหลักที่จะนำไปใช้งานได้กับทุกโดเมน เพื่อให้ออนโทโลจีอื่นสามารถสร้างและระบุรายละเอียดที่เจาะจงขึ้นตามอัปเปอร์ออนโทโลจีได้

ในวิทยานิพนธ์นี้ อัปเปอร์ออนโทโลจีจะเป็นออนโทโลจีที่แสดงโครงสร้างของการกำหนดความหมาย (Semantic Structure) ให้กับแบบจำลองข้อมูลเชิงวัตถุ ทั้งนี้อัปเปอร์ออนโทโลจีถูกกำหนดขึ้นโดยผู้วิจัยเพื่อให้ผู้สร้างออนโทโลจีทุกคน ซึ่งในที่นี้อาจเป็นเจ้าของข้อมูลในแบบจำลองข้อมูลหรือผู้ที่เข้าใจโครงสร้างของแบบจำลองข้อมูลนั้นก็ได้ ใช้เป็นแนวทางในการอธิบายแบบจำลองข้อมูลเชิงวัตถุที่ต้องการนำมารวม ผู้วิจัยพบว่าความหมายที่กำหนดให้กับแบบจำลองข้อมูลเชิงวัตถุนั้นสามารถแสดงในรูปของคำศัพท์ในโดเมนของแบบจำลองข้อมูลและความสัมพันธ์ระหว่างคำศัพท์เหล่านั้น โดยอาจแบ่งได้เป็น 2 ส่วนคือ ส่วนหนึ่งจะถูกกำหนดไว้ก่อนแล้วตามโครงสร้างของแบบจำลองข้อมูล ตัวอย่างเช่น คลาสสองคลาสมีความสัมพันธ์แบบสืบทอดคุณลักษณะ เป็นต้น ซึ่งอัปเปอร์ออนโทโลจีจะต้องสามารถนำไปใช้อธิบายความหมายตามโครงสร้างของแบบจำลองข้อมูล ได้แก่ การอธิบายคลาส แอททริบิวต์

ชนิดของแอททริบิวต์ ความสัมพันธ์ระหว่างคลาสแบบสืบทอดคุณลักษณะ (Inheritance) แบบพี่น้อง (Sibling) และแบบแอกกรีเกชัน (Aggregation) ได้

อีกส่วนหนึ่ง จะเป็นส่วนอธิบายความหมายในรูปของข้อมูลเชิงความหมายที่ไม่มีการระบุไว้ตามโครงสร้างของแบบจำลองข้อมูลเชิงวัตถุ แต่จะถูกกำหนดเพิ่มเติมไว้เพื่อให้เป็นประโยชน์ต่อการรวมแบบจำลองข้อมูล ดังนั้น ผู้วิจัยได้เพิ่มส่วนอธิบายความหมายของคลาส (Class Semantics) และความหมายของแอททริบิวต์ (Attribute Semantics) และส่วนเพิ่มข้อกำหนดบูรณภาพของแอททริบิวต์ (Integrity Constraint) ซึ่งเป็นส่วนเพิ่มเติมเพื่อช่วยในการรวมแบบจำลองข้อมูลไว้ในอัปเปอร์ออนโทโลจีนี้ด้วย

จากความต้องการอัปเปอร์ออนโทโลจีที่สามารถรองรับการสร้างออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุ โดยมุ่งเน้นให้ออนโทโลจีที่ได้แสดงความหมายของแบบจำลองข้อมูลเชิงวัตถุได้อย่างถูกต้อง ตรงกันและเพียงพอต่อการจำแนกความขัดแย้ง รวมทั้งรองรับการทำงานของขั้นตอนวิธีการรวมออนโทโลจีที่เลือกใช้ ร่วมกับแนวทางการแปลงโมเดลระหว่างแบบจำลองข้อมูลเชิงวัตถุกับออนโทโลจีที่ประยุกต์จากเอกสารเมตาโมเดลของนิยามของออนโทโลจี [21] ทำให้ได้อัปเปอร์ออนโทโลจีที่แสดงในรูปของอาร์ดีเอฟกราฟดังรูปที่ 3.1



รูปที่ 3.1 อัปเปอร์ออนโทโลจี

เนื่องจากออนโทโลจีกำหนดให้แสดงแทนเอนทิตีทั้งหมดของโดเมนในรูปของคลาส ดังนั้นทุกเอนทิตีในแบบจำลองข้อมูลเชิงวัตถุ (เช่น คลาส และแอททริบิวต์) จะถูกกำหนดขึ้นเป็นคลาสในอัปเปอร์ออนโทโลจี จากรูปที่ 3.1 อัปเปอร์ออนโทโลจีประกอบด้วยคลาสต่างๆ ดังนี้

1. Class แสดงคลาสของแบบจำลองข้อมูลเชิงวัตถุ ซึ่งในวิทยานิพนธ์นี้จะเรียกรวมคลาสกลุ่มนี้ว่า “คลาสของคลาส”
2. ClassAttribute แสดงแอททริบิวต์ของคลาสของแบบจำลองข้อมูลเชิงวัตถุ ซึ่งในวิทยานิพนธ์นี้จะเรียกรวมคลาสกลุ่มนี้ว่า “คลาสของแอททริบิวต์”

3. ClassIntegrityConstraint แสดงข้อกำหนดบูรณภาพของแอททริบิวต์ในคลาสของแบบจำลองข้อมูลเชิงวัตถุ เพื่ออธิบายความหมายเพิ่มเติมให้กับแบบจำลองข้อมูลเชิงวัตถุ เมื่อต้องการระบุว่า แอททริบิวต์ดังกล่าวในคลาสถูกจำกัดขอบเขตของค่าที่เป็นไปได้อย่างไร ซึ่งในวิทยานิพนธ์นี้จะเรียกรวมคลาสกลุ่มนี้ว่า “คลาสของข้อกำหนดบูรณภาพ”
4. ConstraintValue เป็นสับคลาสของคลาสของข้อกำหนดบูรณภาพ แสดงค่าที่เป็นไปได้ของแอททริบิวต์จากข้อกำหนดบูรณภาพนั้นๆ ซึ่งในวิทยานิพนธ์นี้จะเรียกรวมคลาสกลุ่มนี้ว่า “คลาสของค่าของข้อกำหนดบูรณภาพ”
5. SemanticName แสดงชื่อของความหมายของคลาสหรือแอททริบิวต์ ซึ่งในวิทยานิพนธ์นี้จะเรียกรวมคลาสกลุ่มนี้ว่า “คลาสของชื่อความหมาย”
6. SemanticWord เป็นสับคลาสของคลาสของชื่อความหมาย แสดงค่าของความหมายของคลาสหรือแอททริบิวต์ในแต่ละชื่อของความหมายนั้นๆ ซึ่งในวิทยานิพนธ์นี้จะเรียกรวมคลาสกลุ่มนี้ว่า “คลาสของค่าความหมาย”

ทั้งนี้คลาสแต่ละคู่ในอัปเปอร์ออนโทโลจีมีความสัมพันธ์ระหว่างกันผ่านทางพรอพเพอร์ตี ซึ่งอัปเปอร์ออนโทโลจีประกอบด้วยพรอพเพอร์ตีต่างๆ ดังนี้

1. hasClassAttribute แสดงความสัมพันธ์จากคลาสของคลาส (Class) ไปยังคลาสของแอททริบิวต์ (ClassAttribute) เพื่อแสดงว่าคลาสประกอบด้วยแอททริบิวต์ใดบ้าง
2. hasClassAttributeType แสดงความสัมพันธ์จากคลาสของแอททริบิวต์ (ClassAttribute) ไปยังชนิดข้อมูลเพื่อแสดงชนิดข้อมูลของแอททริบิวต์ที่เป็นชนิดข้อมูลพื้นฐาน (Primitive Data Type)
3. hasClassAttributeAltType แสดงความสัมพันธ์จากคลาสของแอททริบิวต์ (ClassAttribute) ไปยังคลาสของคลาส (Class) เพื่อแสดงชนิดข้อมูลของแอททริบิวต์ที่เป็นชนิดข้อมูลที่ผู้ใช้กำหนด (User-Defined Type)
4. hasClassIntegrityConstraint แสดงความสัมพันธ์จากคลาสของคลาส (Class) ไปยังคลาสของข้อกำหนดบูรณภาพ (ClassIntegrityConstraint) เพื่อแสดงการกำหนดข้อกำหนดบูรณภาพของแอททริบิวต์ในคลาส
5. hasClassSemantic แสดงความสัมพันธ์จากคลาสของคลาส (Class) ไปยังคลาสของชื่อความหมาย (SemanticName) เพื่อแสดงการกำหนดความหมายเพิ่มเติมให้กับคลาส
6. hasClassAttributeSemantic แสดงความสัมพันธ์จากคลาสของแอททริบิวต์ (ClassAttribute) ไปยังคลาสของชื่อความหมาย (SemanticName) เพื่อแสดงการกำหนดความหมายเพิ่มเติมให้กับแอททริบิวต์

นอกจากนี้ อัปเปอร์ออนโทโลจียังประกอบด้วยนิพจน์ภาษาอาวล์คือ “= ∪ ConstraintValue” ซึ่งกำหนดไว้ที่คลาสของข้อกำหนดบูรณภาพ (ClassIntegrityConstraint) เพื่อเป็นการจำกัดขอบเขตของค่าที่เป็นไปได้ของคลาสนี้ว่าจะต้องมีค่าเป็นผลรวมจากการยูเนียนของคลาสของค่าของข้อกำหนดบูรณภาพ (ConstraintValue) เท่านั้น

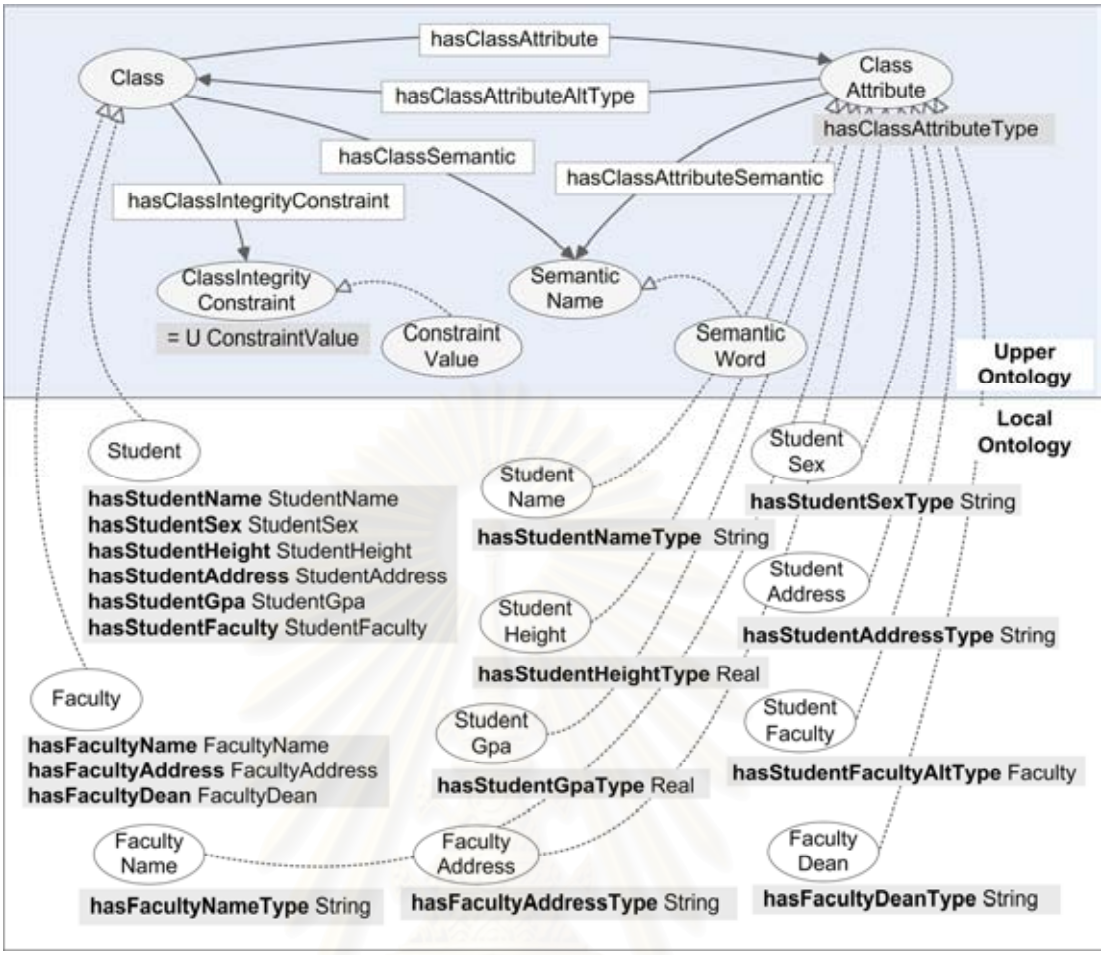
3.2 การนำอัปเปอร์ออนโทโลจีมาใช้สร้างออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุ

เจ้าของแบบจำลองข้อมูลเชิงวัตถุหรือผู้ที่เข้าใจโครงสร้างของแบบจำลองข้อมูลเชิงวัตถุจะทำหน้าที่สร้างออนโทโลจีขึ้นหนึ่งชุดตามโครงสร้างของอัปเปอร์ออนโทโลจีที่ผู้วิจัยกำหนดไว้ เพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุของตน ซึ่งจะเรียกแทนออนโทโลจีที่สร้างขึ้นตามอัปเปอร์ออนโทโลจีว่า “โลคัลออนโทโลจี (Local Ontology)” (อย่างไรก็ตาม คำว่าโลคัลออนโทโลจีในวิทยานิพนธ์นี้จะสื่อความหมายทั้งในแง่ของการเป็นออนโทโลจีที่สร้างขึ้นตามอัปเปอร์ออนโทโลจี และการเป็นออนโทโลจีที่อธิบายแบบจำลองข้อมูลที่ต้องการรวม (แบบจำลองข้อมูลโลคัล) แต่ละชุด)

3.2.1 ส่วนการสร้างโลคัลออนโทโลจีเพื่ออธิบายความหมายที่ปรากฏในแบบจำลองข้อมูลเชิงวัตถุ

เจ้าของแบบจำลองข้อมูลเชิงวัตถุจะสร้างโลคัลออนโทโลจีขึ้นหนึ่งชุดเพื่อแสดงแทนแบบจำลองข้อมูลเชิงวัตถุของตน โดยกำหนดตามอัปเปอร์ออนโทโลจีในหัวข้อที่ 3.1 (รูปที่ 3.1) ตัวอย่างเช่น เมื่อนำอัปเปอร์ออนโทโลจีมาใช้ในการอธิบายยูเอ็มแอลคลาสไดอะแกรมในรูปที่ 2.5(ก) ให้อยู่ในรูปของโลคัลออนโทโลจี ทำให้ได้โลคัลออนโทโลจีที่แสดงด้วยอาร์ดีเอฟกราฟดังแสดงได้ตามรูปที่ 3.2

จะสังเกตได้ว่า ผลจากการอธิบายแบบจำลองข้อมูลเชิงวัตถุที่แสดงไว้ในรูปของยูเอ็มแอลคลาสไดอะแกรมด้วยอัปเปอร์ออนโทโลจีจะไม่เหมือนกับผลการแปลงด้วยข้อกำหนดตามงานวิจัย [21] (รูปที่ 2.5(ข)) ทั้งหมด เช่น ผู้วิจัยกำหนดให้ทุกแอททริบิวต์ถูกอธิบายในรูปของคลาสในโลคัลออนโทโลจี (ในกลุ่มของคลาสของแอททริบิวต์) แทนการกำหนดเป็นพรอพเพอร์ตีตามที่แสดงไว้ในหัวข้อที่ 2.1.5 เนื่องจากผู้วิจัยต้องการอธิบายเพิ่มความหมายของแอททริบิวต์ดังกล่าวไว้ในโลคัลออนโทโลจีที่สร้างขึ้น หรือการกำหนดให้สร้างแอททริบิวต์ของคลาสเพิ่มเติม (ในกลุ่มของคลาสของแอททริบิวต์) และกำหนดให้ชนิดข้อมูลของแอททริบิวต์นั้นเป็นคลาสๆ หนึ่ง (ในกลุ่มของคลาสของคลาส) เพื่อใช้แทนความสัมพันธ์แบบแอกกรีเกชัน ซึ่งในยูเอ็มแอลคลาสไดอะแกรมไม่ได้แสดงไว้ เนื่องจากต้องการนำมาใช้เปรียบเทียบในลักษณะของการเปรียบเทียบแอททริบิวต์ จึงจำเป็นต้องดัดแปลงให้โลคัลออนโทโลจีที่ได้สามารถรองรับการเพิ่มความหมายเหล่านี้



รูปที่ 3.2 ตัวอย่างการนำอัปเปอร์ออนโทโลจีมาใช้อธิบายยูเอ็มแอลของคลาสไดอะแกรม ในรูปของออนโทโลจี

3.2.2 ส่วนการเพิ่มเติมความหมายของแบบจำลองข้อมูลเชิงวัตถุลงในออนโทโลจีที่สร้างไว้

เมื่อได้โลคัลออนโทโลจีที่อธิบายความหมายที่ปรากฏในแบบจำลองข้อมูลเชิงวัตถุโดยตรงแล้ว ขั้นตอนต่อไปเป็นการเพิ่มความหมายให้กับคลาสของคลาสและคลาสของแอททริบิวต์ และแทรกข้อกำหนดบูรณภาพให้กับบางคลาสของคลาสในโลคัลออนโทโลจีเพื่อรักษาความหมายเดิมของแบบจำลองข้อมูลไว้ โดยกำหนดตามอัปเปอร์ออนโทโลจีในหัวข้อที่ 3.1 (รูปที่ 3.1) เจ้าของแบบจำลองข้อมูลเชิงวัตถุจะทำหน้าที่เพิ่มเติมความหมาย และข้อกำหนดบูรณภาพทั้งหมดของแบบจำลองข้อมูลของตนลงในโลคัลออนโทโลจีที่ตนเองได้สร้างไว้ก่อนแล้ว ในที่นี้ผู้วิจัยมีข้อสมมติฐานว่า หากเจ้าของแบบจำลองข้อมูลสามารถระบุความหมายเพิ่มเติมของแบบจำลองข้อมูลให้กับโลคัลออนโทโลจีของตนได้อย่างชัดเจน การรวมแบบจำลองข้อมูลก็จะได้ผลดีใกล้เคียงกับที่ควรจะเป็น แต่หากเจ้าของแบบจำลองข้อมูลละเว้นการระบุความหมายเพิ่มเติมลงในโลคัลออนโทโลจีที่ได้ การรวมแบบจำลองข้อมูลก็ยังคงสามารถทำได้แต่อาจได้ผลด้อยกว่า

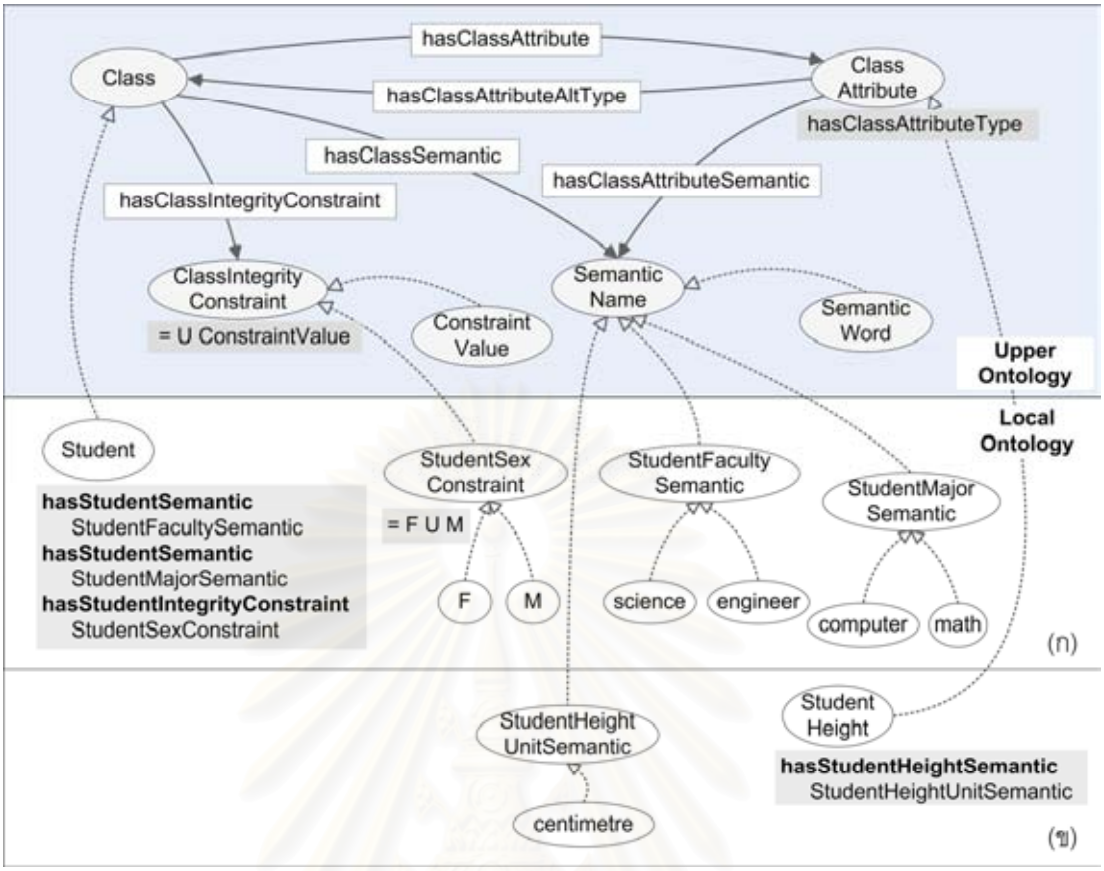
ในการสร้างโลคัลออนโทโลจีเพื่ออธิบายความหมายของแบบจำลองข้อมูลเชิงวัตถุใดก็ตาม เจ้าของแบบจำลองข้อมูลควรจะระบุไว้ว่าข้อมูลฟิลด์ต่างๆ ในแบบจำลองข้อมูลของตนที่จะนำมารวมมี

ลักษณะอย่างไร เพื่อใช้ในการเปรียบเทียบกับฟิลด์อื่นๆ ในแบบจำลองข้อมูลอีกชุดหนึ่งว่า ฟิลด์ใดมีความหมายตรงกัน ฟิลด์ใดมีความหมายแตกต่างกัน เจ้าของแบบจำลองข้อมูลจำเป็นต้องระบุว่าแต่ละฟิลด์ได้รับการออกแบบมาเพื่อเก็บข้อมูลและสื่อความหมายอย่างไร ดังนั้นการระบุความหมายที่ถูกต้องชัดเจนของแบบจำลองข้อมูลดังกล่าวลงในโลคัลออนโทโลจีจึงเป็นสิ่งจำเป็นในการรวมแบบจำลองข้อมูล

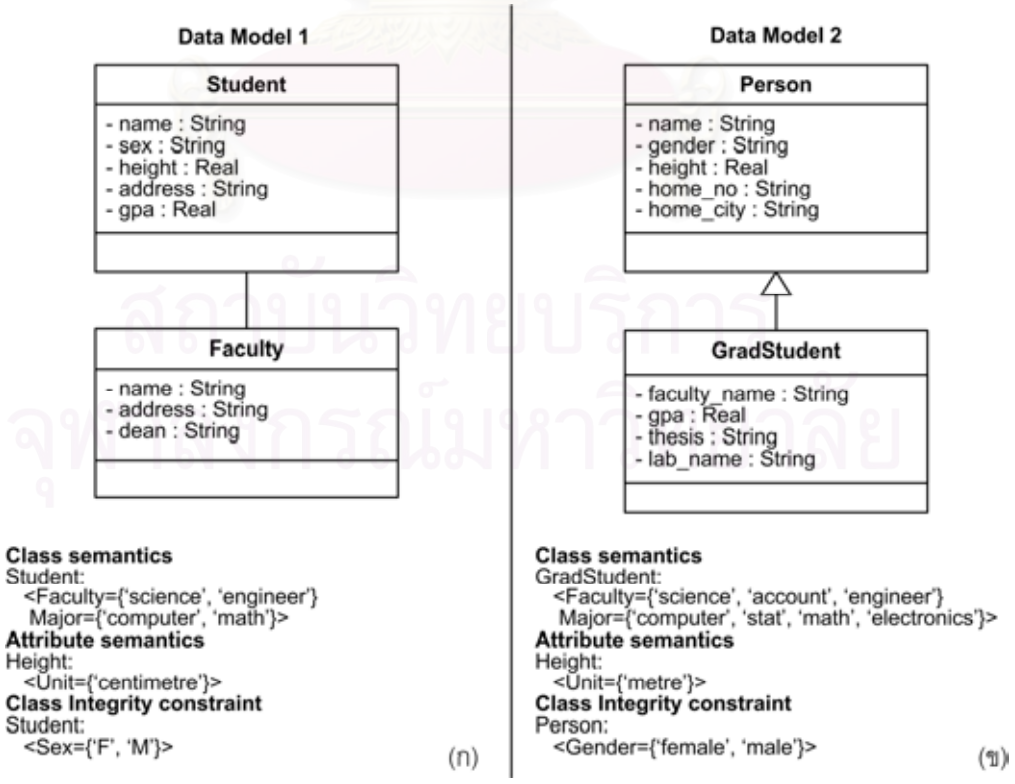
โดยเจ้าของแบบจำลองข้อมูลอาจจะระบุว่า คำศัพท์ในโลคัลออนโทโลจีที่ตนสร้างขึ้นมีความหมายว่าอย่างไร ค่าที่เป็นไปได้มีค่าใดบ้าง หน่วยของข้อมูลคืออะไร มีข้อกำหนดบูรณภาพหรือไม่ ซึ่งเป็นลักษณะเฉพาะของข้อมูลแต่ละฟิลด์ เพื่อนำมาใช้เปรียบเทียบหาระดับความเหมือนระหว่างโลคัลออนโทโลจีของตนกับโลคัลออนโทโลจีชุดอื่น ทั้งนี้ในวิทยานิพนธ์ฉบับนี้กำหนดให้เพิ่มความหมายให้กับคลาสของคลาสหรือคลาสของแอททริบิวท์ใดๆ ในโลคัลออนโทโลจีโดยกำหนดข้อมูลในรูปของชื่อของความหมายและค่าของความหมายผ่านการสร้างเป็นคลาสของชื่อความหมาย (SemanticName) และคลาสของค่าความหมาย (SemanticWord) โดยให้อธิบายผ่านโครงสร้างของอัปเปอร์ออนโทโลจีตามที่ได้แสดงไว้ในหัวข้อที่ 3.1 (รูปที่ 3.1) ในทำนองเดียวกัน หากต้องการเพิ่มข้อกำหนดบูรณภาพให้กับคลาสของคลาสใดในโลคัลออนโทโลจีก็สามารถทำได้โดยผ่านการสร้างเป็นคลาสของข้อกำหนดบูรณภาพ (ClassIntegrityConstraint) และคลาสของค่าของข้อกำหนดบูรณภาพ (ConstraintValue) ด้วยวิธีเดียวกัน ดังแสดงในรูปที่ 3.3

รูปที่ 3.3(ก) เป็นตัวอย่างการเพิ่มความหมายให้กับคลาสของคลาสที่คลาส Student ให้มีชื่อของความหมายเป็น “Faculty” โดยที่ค่าของความหมายเป็น ‘science’ กับ ‘engineer‘ และให้มีชื่อของความหมายเป็น “Major” โดยที่ค่าของความหมายเป็น ‘computer’ กับ ‘math‘ และการเพิ่มข้อกำหนดบูรณภาพของแอททริบิวต์ “Sex” ให้กับคลาสของคลาสที่คลาส Student โดยให้มีค่าที่เป็นไปได้คือ ‘F’ และ ‘M’ ส่วนรูปที่ 3.3(ข) แสดงการเพิ่มความหมายให้กับคลาสของแอททริบิวต์ที่คลาส Height ซึ่งเป็นแอททริบิวต์ของคลาส Student ให้มีชื่อของความหมายเป็น “UnitOfHeight” และค่าของความหมายเป็น ‘centimetre’

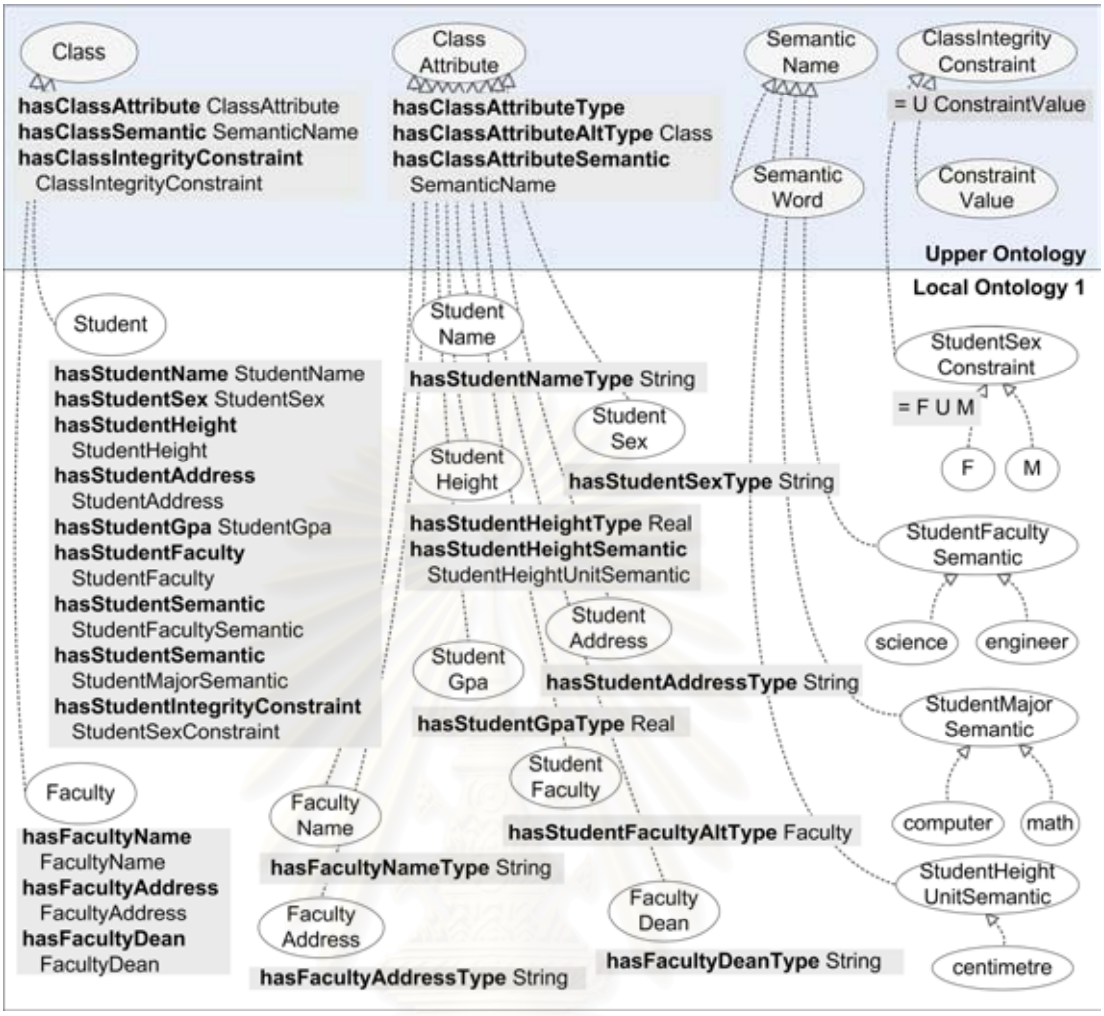
ตัวอย่างของโลคัลออนโทโลจีที่ได้จากการอธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุสองชุดในรูปที่ 3.4(ก) และ 3.4(ข) ผ่านโครงสร้างของอัปเปอร์ออนโทโลจีตามหัวข้อที่ 3.1 ได้แสดงในรูปที่ 3.5 และ 3.6 ตามลำดับ



รูปที่ 3.3 ตัวอย่างการนำอัปเปอร์ออนโทโลจีมาใช้ในการเพิ่มความหมายให้กับคลาสของคลาส และคลาสของแอททริบิวต์ในโลคัลออนโทโลจี



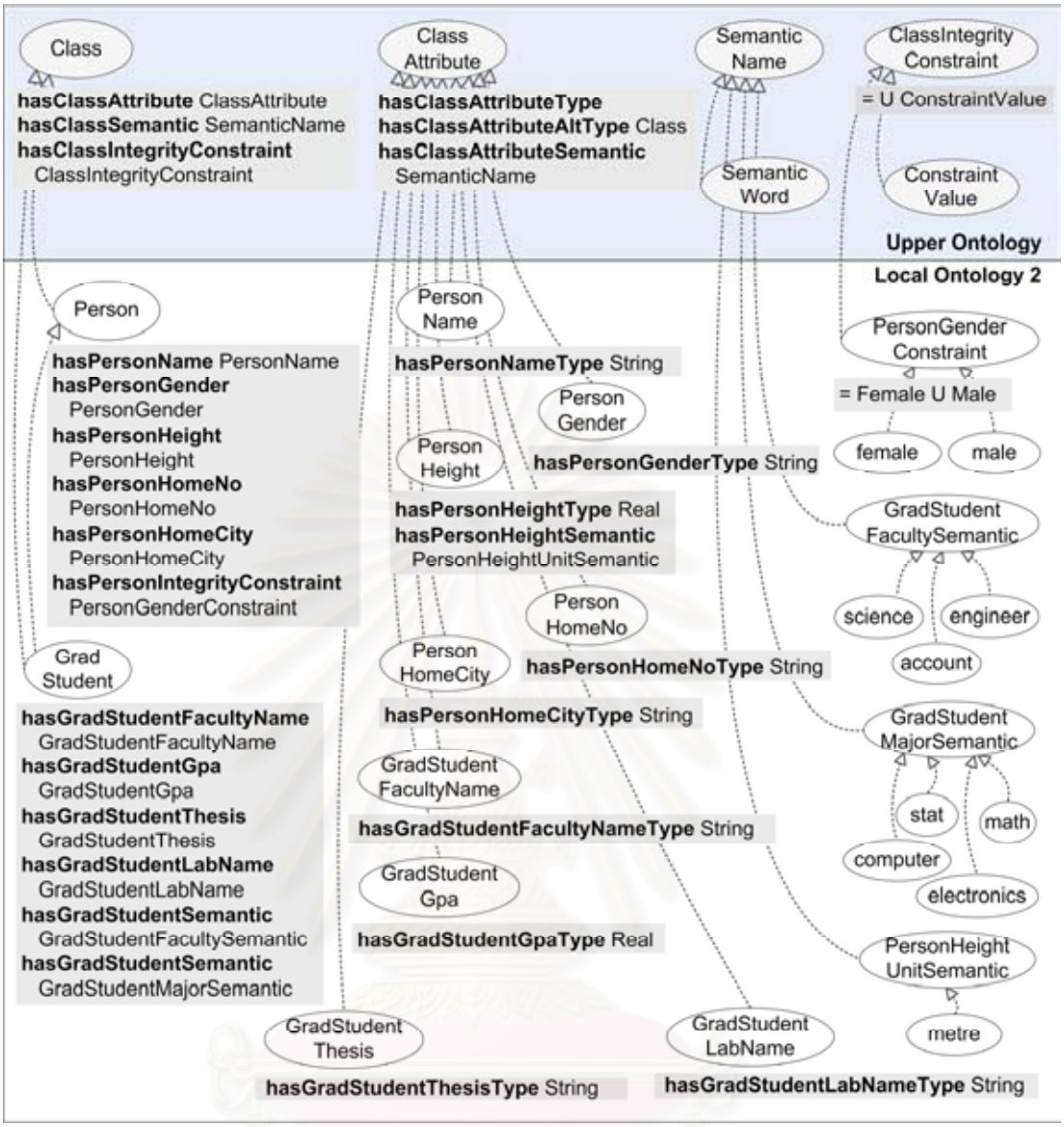
รูปที่ 3.4 ตัวอย่างแบบจำลองข้อมูลเชิงวัตถุ 2 ชุดที่ต้องการนำมารวมกัน



รูปที่ 3.5 ตัวอย่างโลคัลออนโทโลจีสำหรับอธิบายแบบจำลองข้อมูลชุดที่ 1 (รูปที่ 3.4(ก))

การเพิ่มความหมายให้กับโลคัลออนโทโลจีแต่ละชุดที่จะนำมารวมเป็นขั้นตอนสำคัญที่จะช่วยในการจำแนกความขัดแย้งระหว่างโลคัลออนโทโลจีทั้งสองชุดในขั้นตอนการเปรียบเทียบและการรวมออนโทโลจีที่จะกล่าวถึงในบทต่อไป ดังนั้นในการเพิ่มความหมายให้กับโลคัลออนโทโลจี ควรจะระบุความหมายให้เพียงพอที่จะบอกได้ว่าข้อมูลมีความเหมือนหรือแตกต่างกันอย่างไร โดยมีหลักในการพิจารณาดังนี้

- หากคลาสของคลาสหรือคลาสของแอททริบิวต์ใดมีข้อมูลเฉพาะที่สามารถบอกได้ว่าโดเมนของคลาสที่อธิบายหมายถึงอะไร ควรกำหนดความหมายให้กับคลาสของคลาสหรือคลาสของแอททริบิวต์นั้น เช่น หากคลาสของคลาส Student เกี่ยวข้องกับนิสิตในคณะวิทยาศาสตร์และคณะวิศวกรรมศาสตร์ ควรกำหนดความหมายเพิ่มเติมให้กับคลาส Student เป็น “Faculty={‘science’, ’engineer’}” หมายถึงคลาสของคลาส Student มี 1 ความหมายซึ่งชื่อของความหมายคือ “Faculty” และค่าของความหมายคือ ‘science’ และ ‘engineer’ การกำหนดความหมายด้วยวิธีการนี้จะช่วยให้สามารถจำแนกความขัดแย้งของค่าข้อมูลได้



รูปที่ 3.6 ตัวอย่างโลคัลออนโทโลจีสำหรับอธิบายแบบจำลองข้อมูลชุดที่ 2 (รูปที่ 3.4(ข))

- หากคลาสของแอททริบิวต์ใดเกี่ยวข้องกับการบอกจำนวน ควรกำหนดความหมายให้กับคลาสของแอททริบิวต์นั้นเพื่อบอกหน่วยการนับ เช่น หากคลาสของแอททริบิวต์ Height เป็นการบอกความสูงมีหน่วยเป็นเซนติเมตร ควรกำหนดความหมายเพิ่มเติมให้กับคลาส Height เป็น “Unit={‘centimetre’}” หมายถึงคลาสของแอททริบิวต์ Height มี 1 ความหมายซึ่งชื่อของความหมายคือ “Unit” และค่าของความหมายคือ ‘centimetre’ การกำหนดความหมายด้วยวิธีการนี้จะช่วยให้สามารถจำแนกความขัดแย้งของหน่วยข้อมูลที่ใช้ได้
- หากความหมายที่ต้องการอธิบายมีค่าของความหมายเป็นช่วงจำนวน การสร้างคลาสของค่าความหมายควรกำหนดความหมายให้ครอบคลุมทั้งช่วง โดยให้ใช้คำว่า

"GreaterThanOrEqual" ตามด้วยค่าต่ำสุด และ "LessThanOrEqual" ตามด้วยค่าสูงสุด เพื่อแสดงขอบเขตล่างและขอบเขตบนของค่าความหมายตามลำดับ

ทั้งนี้ การกำหนดคำศัพท์ในการเพิ่มความหมายให้กับโลคัลออนโทโลจีจำเป็นต้องให้ความระมัดระวังในเรื่องความถูกต้องด้านภาษาของคำที่เลือกใช้ เพื่อป้องกันความผิดพลาดจากการใช้คำไม่ตรงกันระหว่างโลคัลออนโทโลจีสองชุด เนื่องจากสื่อความหมายผิดพลาด เช่น การกำหนดความหมายเพิ่มเติมให้กับคลาส Student ควรเป็น "Faculty={'engineering'}" มิใช่ "Faculty={'engineer'}"

ส่วนการกำหนดข้อกำหนดบูรณภาพของแอททริบิวต์ให้กับคลาสของคลาสนั้นใช้เพื่อกำหนดว่าอะไรคือค่าที่เป็นไปได้ของแอททริบิวต์ดังกล่าวในคลาส แม้จะไม่ส่งผลต่อการรวมออนโทโลจีโดยตรง แต่เกี่ยวข้องกับการใช้งานข้อมูล โดยเฉพาะอย่างยิ่งในส่วนการสอบถามข้อมูลผ่านออนโทโลจีรวม [39] และจำเป็นต้องนำมาพิจารณาในขั้นตอนการเปรียบเทียบเพื่อให้ทราบว่าเมื่อรวมโลคัลออนโทโลจีทั้งสองชุดเข้าด้วยกันแล้ว จะต้องสร้างฟังก์ชันสำหรับแปลงข้อกำหนดบูรณภาพที่ใช้จริงในแบบจำลองข้อมูลทั้งสองให้ตรงกันหรือไม่ ดังนั้น โลคัลออนโทโลจีที่สร้างควรจะแสดงข้อกำหนดบูรณภาพที่มีทั้งหมดไว้ด้วย

3.3 การสร้างออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจี

เมื่อได้โลคัลออนโทโลจีที่อธิบายความหมายที่ครบถ้วนของแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดตามที่เจ้าของแบบจำลองข้อมูลแต่ละชุดสร้างขึ้นแล้ว ขั้นตอนต่อไปเป็นการระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดที่ได้ก่อนเข้าขั้นตอนการรวม ซึ่งผู้ทำการรวมออนโทโลจีจะสร้างออนโทโลจีขึ้นหนึ่งชุดเพื่อระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดดังกล่าวตามที่ผู้ทำการรวมทราบ โดยอาศัยวิธีการสร้างเป็นคลาสที่สมมูลกัน (Equivalence) หรือคลาสหนึ่งเป็นผลจากการยูเนียน (Union) ของคลาสกลุ่มหนึ่งอันเป็นโครงสร้างที่มีอยู่แล้วในภาษาอาวล์ ซึ่งจะช่วยให้ผลการรวมโลคัลออนโทโลจีคู่ดังกล่าวดีขึ้น เนื่องจากการระบุความสัมพันธ์ในลักษณะนี้จะปรับให้คำศัพท์ต่างๆ ที่ใช้งานในโลคัลออนโทโลจีทั้งสองชุดมีความใกล้เคียงกันมากขึ้น ส่งผลให้สามารถตรวจสอบความแตกต่างระหว่างกันได้ดียิ่งขึ้นด้วย ในที่นี้ผู้วิจัยมีข้อสมมติฐานว่า หากผู้ทำการรวมออนโทโลจีมีความรู้เกี่ยวกับโลคัลออนโทโลจีทั้งสองชุดและสามารถระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดได้ การรวมแบบจำลองข้อมูลก็จะได้ผลดีใกล้เคียงกับที่ควรจะเป็น แต่หากผู้ทำการรวมไม่มีความรู้พอที่จะระบุความสัมพันธ์เพิ่มเติมได้ การรวมแบบจำลองข้อมูลก็ยังคงสามารถทำได้แต่อาจได้ผลด้อยกว่า

วิทยานิพนธ์นี้เลือกพิจารณาความสัมพันธ์ระหว่างโลคัลออนโทโลจีเฉพาะแบบความหมายเหมือนกัน (Synonym) และแบบคำลูกกลุ่ม/ แม่กลุ่ม (Hyponym/ Hypernym) เท่านั้น ดังนั้นการเพิ่มความสัมพันธ์ระหว่างโลคัลออนโทโลจีจึงเลือกเติมเฉพาะความสัมพันธ์ที่เป็นไปตามสองรูปแบบนี้ โดยผู้ทำการรวมออนโทโลจีจะนำโลคัลออนโทโลจีที่ได้มาระบุว่า คำศัพท์ใดในโลคัลออนโทโลจีทั้งสองชุดมีความหมายเหมือนกันหรือเป็นคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกัน ซึ่งเป็นการกำหนดความสัมพันธ์ระหว่างโลคัลออนโทโลจีสองชุดได้โดยตรง ทั้งนี้ การระบุคำที่มีความหมายเหมือนกันหรือคำลูกกลุ่ม/ แม่

กลุ่มกันนี้จะช่วยจำแนกความขัดแย้งของชื่อ ซึ่งเป็นการลดปัญหาที่เกิดขึ้นทั้งจากการใช้คำศัพท์เหมือนกันแสดงแทนความหมายที่ต่างกัน เช่น “Name” อาจหมายถึงชื่อนิสิตหรือชื่อคณะ และการใช้คำศัพท์ต่างกันแสดงแทนความหมายที่เหมือนกัน เช่น “Sex” และ “Gender” ต่างหมายถึงเพศของนิสิต

การระบุเพิ่มเติมทั้งสองแบบอาศัยวิธีการสร้างเป็นคลาสที่สมมูลกันหรือคลาสหนึ่งเป็นผลจากการยูเนียนของคลาสกลุ่มหนึ่งซึ่งเป็นโครงสร้างที่มีอยู่แล้วในภาษาอาวล์ มีวิธีการดังนี้

1. การระบุคำที่มีความหมายเหมือนกัน กำหนดโดยใช้คุณสมบัติการสมมูลกันระหว่างคลาสของคลาสหรือคลาสของแอททริบิวต์ในโลคัลออนโทโลจี ซึ่งสามารถนำไปใช้เมื่อต้องการแสดงการเท่ากันของคำศัพท์คู่ใดๆ ในแบบจำลองข้อมูล ได้ทั้งระหว่างคลาสของคลาสกับคลาสของคลาส ระหว่างคลาสของแอททริบิวต์กับคลาสของแอททริบิวต์ หรือระหว่างคลาสของแอททริบิวต์กับคลาสของแอททริบิวต์อื่นโดยผ่านความสัมพันธ์ของคลาสของแอททริบิวต์หนึ่ง ตัวอย่างเช่น เมื่อต้องการระบุว่าคลาสของแอททริบิวต์ “StudentSex” ในโลคัลออนโทโลจีที่ 1 มีความหมายเหมือนกับคลาสของแอททริบิวต์ “GradStudentGender” ในในโลคัลออนโทโลจีที่ 2 สามารถทำได้โดยสร้างเป็น “equivalentClass” ระหว่างคลาสทั้งสองในโลคัลออนโทโลจีแต่ละชุด ดังนี้

```
<owl:Class rdf:ID=”StudentSex”>
   <owl:equivalentClass
         rdf:resource=”Local2.owl#GradStudentGender”/>
</owl:Class>
```

2. การระบุคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกัน กำหนดโดยใช้คุณสมบัติการยูเนียน ซึ่งจะให้คลาสของคลาสหรือคลาสของแอททริบิวต์ของคำแม่กลุ่มเป็นผลจากการยูเนียนของคลาสของคลาสหรือคลาสของแอททริบิวต์ของคำลูกกลุ่มระหว่างโลคัลออนโทโลจีทั้งสองชุด ซึ่งสามารถนำไปใช้เมื่อต้องการแสดงการเป็นส่วนประกอบกันของคำศัพท์ในแบบจำลองข้อมูล ตัวอย่างเช่น เมื่อต้องการระบุว่าคลาสของแอททริบิวต์ “StudentAddress” ในโลคัลออนโทโลจีที่ 1 เป็นคำแม่กลุ่มซึ่งประกอบด้วยคลาสของแอททริบิวต์ “GradStudentHomeNo” และคลาสของแอททริบิวต์ “GradStudentHomeCity” ในโลคัลออนโทโลจีที่ 2 สามารถทำได้โดยการสร้างกำหนดค่า “unionOf” ระหว่างคลาสลูกกลุ่มให้เท่ากับคลาสแม่กลุ่มในโลคัลออนโทโลจี ดังนี้

```
<owl:Class rdf:ID=”StudentAddress”>
   <owl:unionOf rdf:parseType=”Collection”>
         <owl:Class
               rdf:about=”Local2.owl#GradStudentHomeNo”/>
         <owl:Class
               rdf:about=”Local2.owl#GradStudentHomeCity”/>
   </owl:unionOf>
</owl:Class>
```

การระบุคำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างคำศัพท์ที่ใช้ในโลคัลออนโทโลจีทั้งสองชุดที่จะนำมารวมเป็นขั้นตอนสำคัญที่จะช่วยในการจำแนกความขัดแย้งของชื่อที่ใช้ระหว่างโลคัลออนโทโลจีทั้งสองชุดในขั้นตอนการเปรียบเทียบและการรวมออนโทโลจีที่จะกล่าวถึงในบทต่อไปเช่นกัน โดยมีหลักในการระบุคำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างโลคัลออนโทโลจีที่ควรพิจารณาดังนี้

- หากคลาสของคลาสหรือคลาสของแอททริบิวต์ใดๆ คู่หนึ่งมีชื่อต่างกัน แต่มีความหมายถึงสิ่งเดียวกัน ควรระบุว่าเป็นคำที่มีความหมายเหมือนกัน
- หากคลาสของคลาสหรือคลาสของแอททริบิวต์ใดๆ คู่หนึ่งมีชื่อเหมือนกัน แต่มีความหมายต่างกัน ควรใช้การระบุความหมายเพิ่ม (ตามหัวข้อที่ 3.2.1) ให้ต่างกัน โดยการกำหนดคลาสของชื่อความหมายให้ต่างกัน
- หากคลาสของคลาสหรือคลาสของแอททริบิวต์ใดๆ คู่หนึ่งมีชื่อเหมือนกัน และมีความหมายถึงสิ่งเดียวกัน ควรใช้การระบุความหมายเพิ่ม (ตามหัวข้อที่ 3.2.1) ให้ใกล้เคียงกันโดยการกำหนดคลาสของชื่อความหมายให้เหมือนกัน แต่คลาสของค่าความหมายอาจแตกต่างกันได้
- หากคลาสของคลาสหรือคลาสของแอททริบิวต์ใดๆ คลาสหนึ่งประกอบด้วยคลาสอื่นๆ หลายคลาส ควรระบุว่าคลาสดังกล่าวเป็นคำแม่กลุ่มของคลาสที่เหลือซึ่งเป็นคำลูกกลุ่ม

ตัวอย่างเช่น จากแบบจำลองข้อมูลเชิงวัตถุสองชุดในรูปที่ 3.4 อาจทำการระบุคำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันเพิ่มเติมได้ตามตัวอย่างในรูปที่ 3.7 ซึ่งเขียนอยู่ในรูปของประโยคสัญลักษณ์อย่างง่าย ก่อนที่จะถูกนำไปสร้างขึ้นในรูปภาษาอาวล์ตามโครงสร้างที่ได้อธิบายไปก่อนหน้านี้เพื่อใช้ประกอบการเปรียบเทียบระหว่างโลคัลออนโทโลจีทั้งสองชุด ทั้งนี้ ความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดที่เพิ่มขึ้นจากการระบุคำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันนี้จะเชื่อมโยงคำศัพท์ระหว่างบางคลาสของคลาสหรือบางคลาสของแอททริบิวต์ระหว่างโลคัลออนโทโลจีสองชุดเข้าด้วยกันในทำนองเดียวกันกับวิธีการแมปออนโทโลจี สามารถแสดงการแมปที่เกิดขึ้นในรูปของอาร์ดีเอฟกราฟได้ตามรูปที่ 3.8

Synonym-list:

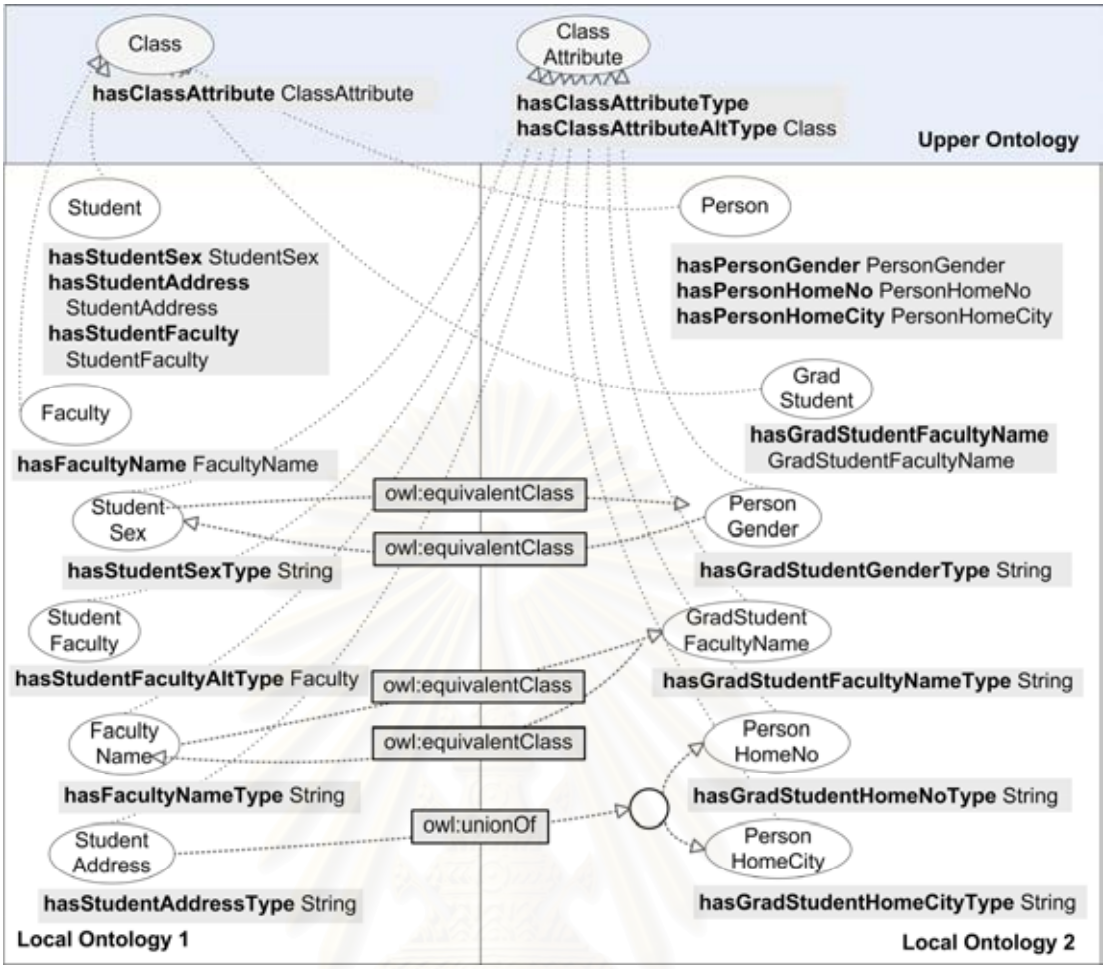
Data Model 1.Student.sex = Data Model 2.Person.gender

Data Model 1.Student.faculty.name = Data Model 2.GradStudent.faculty_name

Hypernym-list:

Data Model 1.Student.address = Data Model 2.Person.home_no +
Data Model 2.Person.home_city

รูปที่ 3.7 การระบุคำที่มีความหมายเหมือนกันและคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างแบบจำลองข้อมูลสองชุดในรูปที่ 3.4



รูปที่ 3.8 ตัวอย่างแมปปิงแสดงการระบุคำที่มีความหมายเหมือนกันและคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกัน ระหว่างโลคัลออนโทโลจีทั้งสองชุดจากรูปที่ 3.5 และ 3.6

ทั้งนี้ เมื่อได้โลคัลออนโทโลจีทั้งสองชุดตามรูปที่ 3.5 และ 3.6 พร้อมทั้งการระบุความหมายเพิ่มเติมเพื่อสร้างแมปปิงระหว่างบางส่วนของโลคัลออนโทโลจีทั้งสองชุดในรูปที่ 3.8 แล้ว ผู้ทำการรวมออนโทโลจีจะนำโลคัลออนโทโลจีที่ได้ไปใช้ในขั้นตอนการเปรียบเทียบและการรวมออนโทโลจี โดยผู้วิจัยจะนำโลคัลออนโทโลจีและการระบุความหมายที่ได้จากบทนี้ มาใช้แสดงตัวอย่างในขั้นตอนการเปรียบเทียบและการรวมออนโทโลจีในบทที่ 4 ต่อไป

บทที่ 4
การเปรียบเทียบและการรวมออนโทโลจี

ขั้นตอนวิธีในการสร้างออนโทโลจีรวมเพื่อนำมาใช้กับการรวมแบบจำลองข้อมูลเชิงวัตถุที่นำเสนอในวิทยานิพนธ์นี้ ได้รับการออกแบบเพื่อรองรับการรวมโลคัลออนโทโลจีสองชุดที่อธิบายแบบจำลองข้อมูลเชิงวัตถุที่สร้างขึ้นผ่านอัปเปอร์ออนโทโลจีตามที่ได้อธิบายไว้ในบทที่ 3 โดยเน้นการพิจารณาจากความสัมพันธ์ในเชิงความหมายหรือโครงสร้างที่บ่งบอกความหมาย การทำงานจะแบ่งเป็นสองขั้นตอนคือ ขั้นตอนการเปรียบเทียบออนโทโลจี ซึ่งจะตรวจสอบว่าโลคัลออนโทโลจีทั้งสองชุดมีระดับความสัมพันธ์กันอย่างไร หรือเรียกได้ว่าเป็นขั้นตอนการแมปออนโทโลจีในหัวข้อที่ 4.1 และขั้นตอนการรวมออนโทโลจี ซึ่งจะนำค่าความสัมพันธ์ที่ได้จากการเปรียบเทียบออนโทโลจีมาวิเคราะห์เพื่อสร้างเป็นออนโทโลจีรวม หรือเรียกได้ว่าเป็นขั้นตอนการผนวกออนโทโลจีในหัวข้อที่ 4.2 ส่วนหัวข้อที่ 4.3 จะแสดงตัวอย่างการเปรียบเทียบและการรวมโลคัลออนโทโลจีตามขั้นตอนวิธีที่นำเสนอในวิทยานิพนธ์นี้

4.1 การเปรียบเทียบโลคัลออนโทโลจี

การเปรียบเทียบโลคัลออนโทโลจีทำได้โดยการแยกเปรียบเทียบส่วนประกอบต่างๆ ภายในโลคัลออนโทโลจีทั้งสองชุดทีละคู่เพื่อหาระดับความสัมพันธ์ระหว่างแต่ละส่วนของออนโทโลจี โดยประยุกต์จากวิธีการเปรียบเทียบแบบจำลองข้อมูลในงานวิจัย [20] ซึ่งระดับความสัมพันธ์ที่ได้จากการเปรียบเทียบโลคัลออนโทโลจีนี้จะถูกนำมาพิจารณาเพื่อตัดสินใจรวมออนโทโลจีตามหัวข้อที่ 4.2 วิธีการเปรียบเทียบโลคัลออนโทโลจีในหัวข้อนี้ประกอบด้วย 3 ขั้นตอนย่อย แบ่งตามส่วนประกอบภายในโลคัลออนโทโลจีที่ต้องการเปรียบเทียบคือ การเปรียบเทียบความหมาย การเปรียบเทียบคลาสของแอททริบิวต์ และการเปรียบเทียบคลาสของคลาส

4.1.1 การเปรียบเทียบความหมาย

ขั้นตอนนี้เป็นการเปรียบเทียบส่วนความหมายของคลาสของคลาสหรือคลาสของแอททริบิวต์ภายในโลคัลออนโทโลจี (เทียบได้กับความหมายของคลาสและแอททริบิวต์ในแบบจำลองข้อมูลเชิงวัตถุ) ที่ได้เพิ่มให้กับโลคัลออนโทโลจีแต่ละชุดในหัวข้อที่ 3.2.2 โดยนำค่าวิทยาการศึกษาสำนึกที่ใช้ในงานวิจัย [20] มาใช้กำหนดค่าของผลลัพธ์ที่ได้จากการเปรียบเทียบ ซึ่งค่าวิทยาการศึกษาสำนึกเหล่านี้จะระบุความสัมพันธ์ในเชิงความหมายระหว่างคลาสของคลาส หรือคลาสของแอททริบิวต์ที่นำมาเปรียบเทียบ

เมื่อเลือกคลาสของคลาสหรือคลาสของแอททริบิวต์ที่ต้องการเปรียบเทียบจากโลคัลออนโทโลจีทั้งสองชุดแล้ว การเปรียบเทียบความหมายจะทำในสองลักษณะคือ พิจารณาจากคลาสของชื่อความหมาย (SemanticName) และจากคลาสของค่าความหมาย (SemanticWord) โดยกำหนดให้

T1 เป็นเทอม (Term) ที่จะพิจารณาเปรียบเทียบความหมายที่อยู่ในโลคัลออนโทโลจีชุดที่หนึ่ง (ซึ่งเทอมในที่นี้อาจเป็นได้ทั้งคลาสของคลาสและคลาสของแอททริบิวต์ขึ้นกับว่าต้องการเปรียบเทียบความหมายของคลาสของคลาสหรือคลาสของแอททริบิวต์)

T2 เป็นเทอมที่จะพิจารณาเปรียบเทียบความหมายที่อยู่ในโลคัลออนโทโลจีชุดที่สอง

และให้ α เป็นจำนวนคลาสของชื่อความหมายของ T1

β เป็นจำนวนคลาสของชื่อความหมายของ T2 โดย $\alpha \leq \beta$

การเปรียบเทียบจะทำที่แต่ละคู่ของคลาสของชื่อความหมายเพื่อให้ได้ค่าความสัมพันธ์ r โดยคำนวณจากอัตราส่วนระหว่างความเหมือนของคลาสของชื่อความหมายกับคลาสของค่าความหมายที่เป็นสับคลาสแต่ละคู่ในสัดส่วน 1:1 ซึ่งกำหนดให้ค่าความเหมือนเป็น 1 หากเหมือนกัน เป็น 0 หากต่างกัน และอยู่ระหว่าง 0–1 หากเหมือนกันบางส่วนหรือเป็นสับเซตกัน และให้ค่า r เป็นผลรวมของค่าอัตราส่วนจากการเปรียบเทียบคลาสของชื่อความหมายและคลาสของค่าความหมายหารด้วย 2 ดังนั้นค่า r จะอยู่ในช่วง 0-1 ซึ่งหากค่า r เป็น 0 จะแสดงว่าคลาสทั้งคู่มีความหมายต่างกัน โดยค่า r ที่เพิ่มขึ้นแสดงว่าความหมายของทั้งคู่มีความเหมือนกันมากขึ้น และหากค่า r เป็น 1 จะแสดงว่าความหมายของทั้งคู่เหมือนกัน

วิทยานิพนธ์นี้นำค่า r ที่ได้จากงานวิจัย [20] มากำหนดเป็นผลลัพธ์จากการเปรียบเทียบแต่ละคู่ของคลาสของชื่อความหมายดังที่แสดงในตารางที่ 4.1

ตารางที่ 4.1 ค่าความสัมพันธ์ r จากการเปรียบเทียบแต่ละคู่ของคลาสของชื่อความหมาย [20]

กรณี	คลาสของชื่อความหมาย	คลาสของค่าความหมาย	r
1	เหมือนกัน	เหมือนกัน	1
2	เหมือนกัน	เป็นสับเซตกัน	0.8
3	เหมือนกัน	มีบางส่วนซ้อนเหลื่อมกัน (Overlap)	0.8
4	เหมือนกัน	ต่างกัน	0.5
5	ต่างกัน	เป็นค่าใดๆ	0

จากนั้นจะนำค่า r จากการเปรียบเทียบแต่ละคู่ของคลาสของชื่อความหมายภายในเทอม T1 และ T2 มาเก็บไว้ในเวกเตอร์ (Vector) $R = [r_1, r_2, \ldots, r_\alpha]$ และให้ $Ms = \sum r_i$ โดย $r_i \in R$ และ $i = 1, \ldots, \alpha$ ทั้งนี้ค่า Ms ที่ได้จะแสดงระดับความเหมือนและความสัมพันธ์ระหว่างชุดของคลาสของความหมายของ T1 และ T2 ที่นำมาเปรียบเทียบกันซึ่งยึดตามงานวิจัย [20] ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของความหมายของ T1 และ T2 [20]

กรณี	Ms	ระดับความเหมือน	ความสัมพันธ์	s
1	α (α=β)	Equivalence	Equivalence	1
2	α (α<β)	Inclusion	Superclass	1
3	$\geq$ 3/4α (α=β)	Equivalence	Equivalence	1
4	$\geq$ 3/4α (α<β)	Inclusion	Superclass	1
5	$\geq$ 1/2α	Tight-intersection	Sibling	0.1
6	< 1/2α	Loose-intersection	Disjoint	0
7	0	Disjoint	Disjoint	0

หมายเหตุ ค่า s ในตารางเป็นค่าวิทยาการศึกษาสำนึก (Heuristics) ที่ใช้แสดงว่า จะสามารถรวมคำศัพท์ในโลคัลออนโทโลจีคู่นั้นเข้าด้วยกันได้หรือไม่ ซึ่งนำไปใช้ในขั้นตอนการเปรียบเทียบต่อไป

ทั้งนี้ ค่าความสัมพันธ์ที่ได้จากตารางที่ 4.2 จะถูกนำไปพิจารณาเป็นค่าความสัมพันธ์จากการเปรียบเทียบความหมายของคลาสของแอททริบิวต์ในหัวข้อที่ 4.1.2 และค่าความสัมพันธ์จากการเปรียบเทียบความหมายของคลาสของคลาสในหัวข้อที่ 4.1.3 ต่อไป ค่าความสัมพันธ์ระหว่างเทอม T1 และ T2 จากการเปรียบเทียบชุดของคลาสของความหมายจากตารางที่ 4.2 สามารถอธิบายได้ดังนี้

- ความสัมพันธ์แบบเท่ากัน (Equivalence) หมายถึง T1 และ T2 มีความหมายเหมือนกันทั้งหมด (กรณีที่ 1) หรือมีความหมายส่วนใหญ่เหมือนกัน (กรณีที่ 3)
- ความสัมพันธ์แบบซูเปอร์คลาส (Superclass) หมายถึง T1 มีความหมายน้อยกว่า T2 และความหมายทั้งหมดของ T1 เหมือนกับความหมายส่วนหนึ่งของ T2 (กรณีที่ 2 และ 4) ทั้งนี้การที่ T1 เป็นซูเปอร์คลาสของ T2 มีค่าเท่ากับ T2 เป็นสับคลาสของ T1
- ความสัมพันธ์แบบพี่น้อง (Sibling) หมายถึงความหมายบางส่วนของ T1 เหมือนกับความหมายส่วนหนึ่งของ T2 (กรณีที่ 5) หรืออาจเกิดได้จากการที่ T1 และ T2 มีคลาสของชื่อความหมายเหมือนกันหมด แต่คลาสของค่าความหมายต่างกัน
- ความสัมพันธ์แบบแตกต่างกัน (Disjoint) หมายถึง T1 และ T2 มีความหมายส่วนใหญ่ไม่เหมือนกัน (กรณีที่ 6 และ 7)

ตัวอย่างที่ 1 จากรูปที่ 4.1 การเปรียบเทียบความหมายระหว่างคลาสของคลาส Student ในโลคัลออน-โทโลจีที่ 1 (รูปที่ 3.5) และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2 (รูปที่ 3.6) มีวิธีการดังนี้

กำหนด T1 = Student

T2 = GradStudent

จะได้ α = 2 (StudentFacultySemantic, StudentMajorSemantic)

β = 2 (GradStudentFacultySemantic, GradStudentMajorSemantic)

เปรียบเทียบระหว่าง StudentFacultySemantic และ GradStudentFacultySemantic

r_1 = 0.8 (ตารางที่ 4.1 กรณีที่ 2)

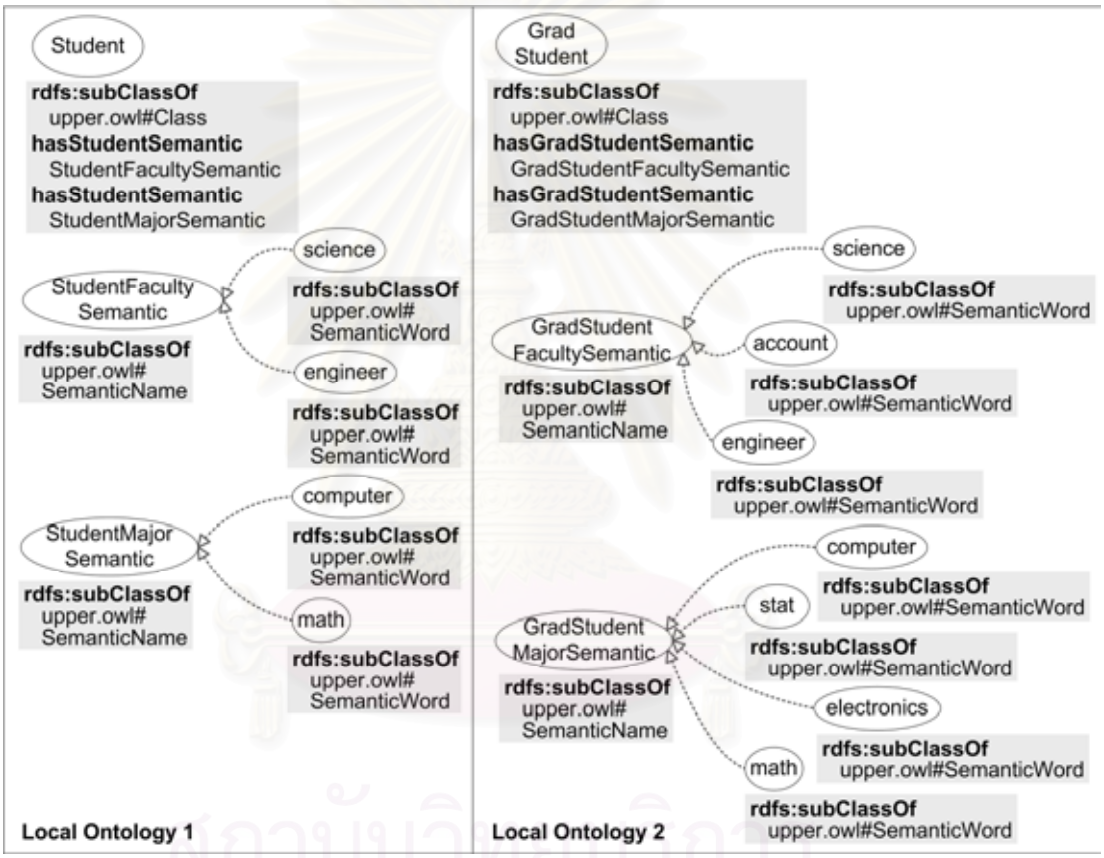
เปรียบเทียบระหว่าง StudentMajorSemantic และ GradStudentMajorSemantic

r_2 = 0.8 (ตารางที่ 4.1 กรณีที่ 2)

R = [0.8, 0.8] $\therefore$ Ms = 0.8 + 0.8 = 1.6

$\therefore$ ระดับความเหมือน = Equivalence, ความสัมพันธ์ = Equivalence และ s = 1

(ตารางที่ 4.2 กรณีที่ 3)



รูปที่ 4.1 ตัวอย่างการเปรียบเทียบความหมายระหว่างคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2

4.1.2 การเปรียบเทียบคลาสของแอททริบิวต์

ขั้นตอนนี้เป็นการเปรียบเทียบส่วนคลาสของแอททริบิวต์ภายในโลคัลออนโทโลจี (เทียบได้กับแอททริบิวต์ในแบบจำลองข้อมูลเชิงวัตถุ) ตามการอธิบายผ่านอัปเปอร์ออนโทโลจีในหัวข้อที่ 3.1 การเปรียบเทียบในหัวข้อนี้จะพิจารณาจากชื่อ ความหมาย และชนิดข้อมูลของคลาสของแอททริบิวต์ ร่วมกับคลาสของข้อกำหนดบูรณภาพของคลาสของคลาสที่กำกับคลาสของแอททริบิวต์นั้นๆ โดยค่าระดับความเหมือนหรือความแตกต่างของคลาสของแอททริบิวต์ และความเป็นไปได้ในการรวมคลาสของแอททริบิวต์

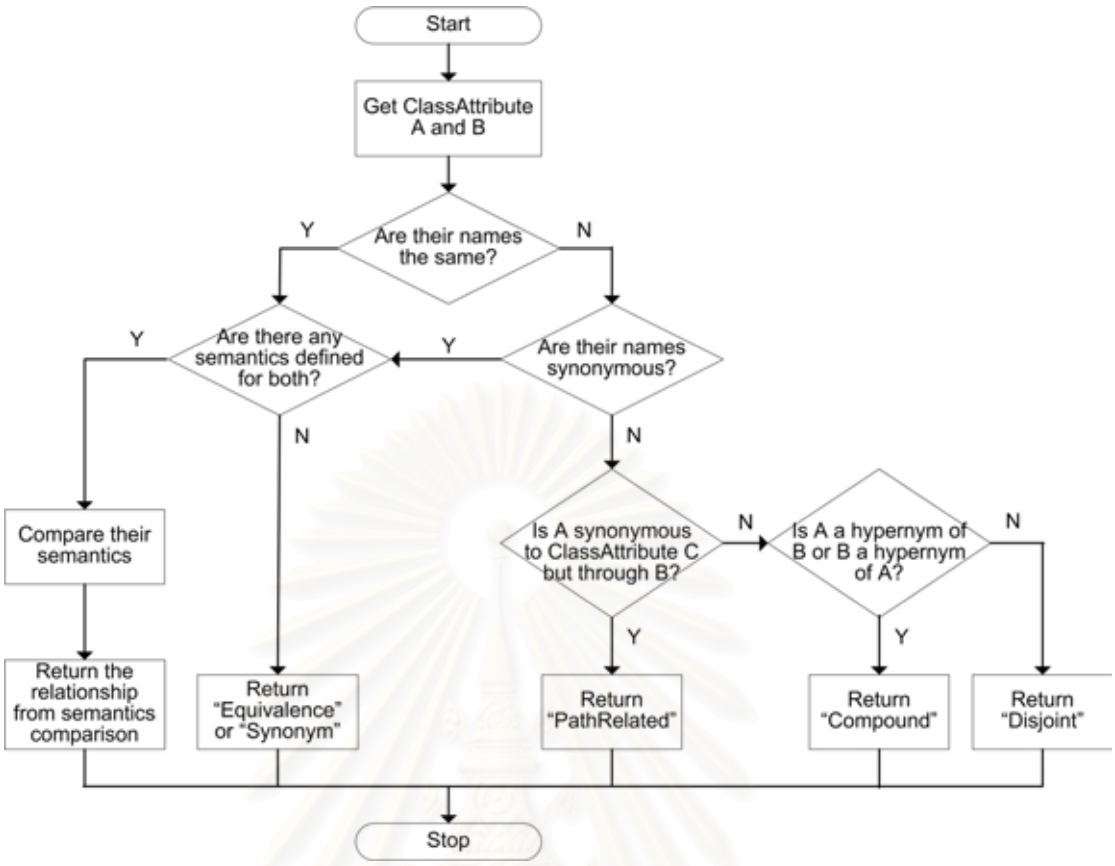
นั้นเข้ากับคลาสของแอททริบิวต์อื่นระหว่างโลคัลออนโทโลจีสองชุดพิจารณาได้จากชื่อ และความหมายของคลาสของแอททริบิวต์แต่ละคู่ ซึ่งเป็นการพิจารณาในเชิงความหมายของคลาสของแอททริบิวต์เป็นหลัก ในขณะที่การเปรียบเทียบชนิดข้อมูลและคลาสของข้อกำหนดบูรณภาพของคลาสของแอททริบิวต์นั้นจะรวบรวมไว้เพื่อนำมาใช้ในขั้นตอนการรวมคลาสของแอททริบิวต์ (หัวข้อที่ 4.2.2) ต่อไป เนื่องจากการเปรียบเทียบดังกล่าวไม่สามารถนำมาบอกได้โดยตรงว่าคลาสของแอททริบิวต์คู่ดังกล่าวเหมือนหรือต่างกัน การเปรียบเทียบในขั้นตอนนี้ประกอบด้วย 3 ขั้นตอนย่อยคือ

4.1.2.1 การหาความสัมพันธ์จากการเปรียบเทียบแต่ละคู่ของคลาสของแอททริบิวต์

ขั้นตอนนี้เป็นการเปรียบเทียบคลาสของแอททริบิวต์คู่ใดๆ จากโลคัลออนโทโลจีสองชุดว่ามีความสัมพันธ์กันหรือไม่ โดยพิจารณาจากชื่อคลาสของแอททริบิวต์ คลาสของความหมายที่ระบุเพิ่มให้กับคลาสของแอททริบิวต์ (ถ้ามี) และการระบุคำที่มีความหมายเหมือนกันหรือเป็นคำลูกกลุ่ม/ แม่กลุ่มกัน (ถ้ามี) ระหว่างคลาสของแอททริบิวต์คู่ที่ต้องการเปรียบเทียบ ขั้นตอนและเงื่อนไขการเปรียบเทียบแสดงได้ในรูปที่ 4.2 โดย A และ B เป็นคลาสของแอททริบิวต์ที่ต้องการเปรียบเทียบจากโลคัลออนโทโลจีชุดที่หนึ่งและโลคัลออนโทโลจีชุดที่สองตามลำดับ ทั้งนี้ผลที่ได้จากการเปรียบเทียบคือค่า s ซึ่งเป็นค่าวิทยาการศึกษาสำนึกที่อ้างอิงมาจากงานวิจัย [20] และระดับความสัมพันธ์ระหว่างคลาสของแอททริบิวต์ที่เปรียบเทียบ แสดงไว้ในตารางที่ 4.3 โดยค่า s ที่ได้จะอยู่ในช่วง 0-1 และค่านี้จะเป็นตัวแสดงว่าสามารถรวมคลาสของแอททริบิวต์คู่ดังกล่าวเข้าด้วยกันได้หรือไม่ หากค่า s เป็น 0 แสดงว่าคลาสคู่นั้นไม่มีความสัมพันธ์กันเพียงพอที่จะรวมเข้าด้วยกัน แต่หากค่า s เป็น 1 แสดงว่าคลาสคู่นั้นมีความสัมพันธ์กันและยอมให้รวมเข้าเป็นคลาสเดียวกันได้

หลังจากนั้น ค่า s ที่ได้จากตารางที่ 4.3 จะถูกนำไปใช้ในขั้นตอนการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในแต่ละคลาสของคลาสในหัวข้อที่ 4.1.2.3 ต่อไป ค่า s และความสัมพันธ์ที่ได้จากการเปรียบเทียบระหว่างคลาสของแอททริบิวต์แต่ละคู่จากตารางที่ 4.3 สามารถอธิบายได้ดังนี้

- ความสัมพันธ์แบบเท่ากัน (Equivalence) แสดงว่าคลาสของแอททริบิวต์ทั้งคู่เหมือนกัน จะเกิดได้สองแบบคือเมื่อคลาสของแอททริบิวต์ทั้งสองคลาสได้รับการอธิบายความหมายเพิ่มและผลจากการเปรียบเทียบความหมาย (ตารางที่ 4.2 หัวข้อที่ 4.1.1) ของคลาสคู่ดังกล่าวเป็นความสัมพันธ์แบบเท่ากัน (กรณีที่ 1 และ 2) หรือเมื่อคลาสของแอททริบิวต์ทั้งสองคลาสมีชื่อเหมือนกันโดยที่ทั้งสองคลาสต่างไม่ได้รับการอธิบายความหมายเพิ่ม (กรณีที่ 3) ทั้งสามกรณีนี้แสดงว่าคลาสของแอททริบิวต์ทั้งคู่เหมือนกันมากพอที่จะรวมเป็นคลาสเดียวกันได้ และกำหนดให้ค่า s = 1



รูปที่ 4.2 ขั้นตอนการเปรียบเทียบแต่ละคู่ของคลาสของแอททริบิวต์เพื่อหาระดับความสัมพันธ์ [20]

ตารางที่ 4.3 ค่าความสัมพันธ์จากการเปรียบเทียบแต่ละคู่ของคลาสของแอททริบิวต์ [20]

กรณี	การเปรียบเทียบคลาสของแอททริบิวต์	ความสัมพันธ์	s
1	ชื่อคลาสเหมือนกัน และทั้งคู่ได้รับการอธิบายความหมายเพิ่ม	จากตารางที่ 4.2	จากตารางที่ 4.2
2	ชื่อคลาสต่างกัน แต่เป็นคำที่มีความหมายเหมือนกัน และทั้งคู่ได้รับการอธิบายความหมายเพิ่ม	จากตารางที่ 4.2	จากตารางที่ 4.2
3	ชื่อคลาสเหมือนกัน แต่ไม่มีการอธิบายความหมายเพิ่ม	Equivalence	1
4	ชื่อคลาสต่างกัน แต่เป็นคำที่มีความหมายเหมือนกัน และไม่มีการอธิบายความหมายเพิ่ม	Synonym	1
5	ชื่อคลาสเป็นคำที่มีความหมายเหมือนกับชื่อคลาสอื่น โดยผ่านทางความสัมพันธ์กับคลาสที่นำมาเปรียบเทียบ	PathRelated	1
6	ชื่อคลาสเป็นคำที่เป็นคำลูกกลุ่ม / แม่กลุ่มกัน	Compound	1
7	ไม่มีความสัมพันธ์กัน	Disjoint	0

- ความสัมพันธ์แบบซูเปอร์คลาส (Superclass) แสดงว่าคลาสของแอททริบิวต์ทั้งคู่ได้รับการอธิบายความหมายเพิ่ม และผลจากการเปรียบเทียบความหมาย (ตารางที่ 4.2 หัวข้อที่ 4.1.1) ของคลาสคู่ดังกล่าวเป็นความสัมพันธ์แบบซูเปอร์คลาส (กรณีที่ 1 และ 2) ในกรณีนี้แสดงว่าคลาสของแอททริบิวต์หนึ่งมีความหมายครอบคลุมอีกคลาสของแอททริบิวต์หนึ่งทั้งหมด ทำให้สามารถรวมกันเป็นคลาสเดียวที่มีความหมายครอบคลุมทั้งสองคลาสได้ และกำหนดให้ค่า s = 1
- ความสัมพันธ์แบบพี่น้อง (Sibling) แสดงว่าคลาสของแอททริบิวต์ทั้งคู่ได้รับการอธิบายความหมายเพิ่ม และผลจากการเปรียบเทียบความหมาย (ตารางที่ 4.2 หัวข้อที่ 4.1.1) ของคลาสคู่ดังกล่าวเป็นความสัมพันธ์แบบพี่น้อง (กรณีที่ 1 และ 2) ในกรณีนี้แสดงว่าความหมายส่วนหนึ่งของคลาสของแอททริบิวต์หนึ่งเหมือนกับความหมายบางส่วนของอีกคลาสของแอททริบิวต์หนึ่ง ทำให้สามารถสร้างคลาสใหม่เพื่อแสดงความหมายทั่วไปที่คลาสทั้งคู่ใช้ร่วมกันได้ และกำหนดให้ค่า s = 1
- ความสัมพันธ์แบบความหมายเหมือนกัน (Synonym) แสดงว่าคลาสของแอททริบิวต์ทั้งคู่ได้รับการระบุว่าเป็นคำที่มีความหมายเหมือนกัน (กรณีที่ 4) ทำให้สามารถรวมเป็นคลาสเดียวกันได้ และกำหนดให้ค่า s = 1
- ความสัมพันธ์แบบการมีเส้นทางเกี่ยวข้อง (PathRelated) เกิดขึ้นจากการที่แอททริบิวต์เดิมในแบบจำลองข้อมูลมีความสัมพันธ์แบบแอกกรีเกชัน ส่งผลให้คลาสของแอททริบิวต์ที่อธิบายแอททริบิวต์นั้นในโลคัลออนโทโลจีมีชนิดข้อมูลเป็นคลาสของคลาสซึ่งเกี่ยวข้องกับคลาสของแอททริบิวต์กลุ่มหนึ่ง เมื่อนำคลาสของแอททริบิวต์ดังกล่าวมาเปรียบเทียบกับคลาสของแอททริบิวต์อีกคลาสหนึ่ง โดยผลการเปรียบเทียบพบว่าคลาสของแอททริบิวต์ที่นำมาเปรียบเทียบมีความสัมพันธ์กับคลาสของแอททริบิวต์อื่นในกลุ่มนั้นผ่านคลาสของแอททริบิวต์แรกที่นำมาเปรียบเทียบ (กรณีที่ 5) ทำให้สามารถรวมกันเป็นคลาสเดียวในลักษณะของความสัมพันธ์แบบความหมายเหมือนกันได้ (โดยวิธีการรวมนี้จะกล่าวในหัวข้อที่ 4.2.3.3) และกำหนดให้ค่า s = 1
- ความสัมพันธ์แบบการเป็นส่วนประกอบกัน (Compound) แสดงว่าคลาสของแอททริบิวต์ทั้งคู่ได้รับการระบุว่าเป็นคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกัน (กรณีที่ 6) ทำให้สามารถรวมกันเป็นคลาสเดียวในลักษณะของการเป็นส่วนประกอบกันได้ และกำหนดให้ค่า s = 1
- ความสัมพันธ์แบบแตกต่างกัน (Disjoint) แสดงว่าคลาสของแอททริบิวต์ทั้งคู่ไม่มีความสัมพันธ์ใดๆ กัน (กรณีที่ 7) ทำให้ไม่สามารถรวมกันได้ และกำหนดให้ค่า s = 0

4.1.2.2 การเปรียบเทียบชนิดข้อมูลและค่าที่เป็นไปได้สำหรับคลาสของแอททริบิวต์

เมื่อค่า s และความสัมพันธ์ระหว่างคลาสของแอททริบิวต์จากหัวข้อที่ 4.1.2.1 แล้ว ขั้นตอนนี้จะเป็นการเปรียบเทียบเพื่อนำไปใช้ในขั้นตอนการรวมคลาสของแอททริบิวต์เหล่านี้ว่า หากมีการ

รวมคลาสคู่ใดๆ เข้าด้วยกันคลาสรวมที่ได้ควรเป็นอย่างไร จะต้องสร้างฟังก์ชันเพิ่มเติมเพื่อแปลงข้อมูลระหว่างคลาสทั้งสองหรือไม่ ซึ่งจะเก็บบันทึกผลการเปรียบเทียบที่ได้ไปใช้ในขั้นตอนการรวมคลาสของแอททริบิวต์ต่อไป ทั้งนี้หากคลาสของแอททริบิวต์คู่หนึ่งมีความสัมพันธ์กัน (ความสัมพันธ์แบบใดๆ ยกเว้นความสัมพันธ์แบบแตกต่างกัน) จะนำคลาสคู่ดังกล่าวมาเปรียบเทียบชนิดของข้อมูลและค่าที่เป็นไปได้ระหว่างคลาสของแอททริบิวต์ทั้งสอง โดยมีวิธีการเปรียบเทียบดังนี้

1. การเปรียบเทียบชนิดข้อมูลของคลาสของแอททริบิวต์ (ClassAttributeType) หากชนิดข้อมูลของคลาสคู่ที่นำมาเปรียบเทียบต่างกัน ขั้นตอนนี้จะบันทึกไว้ว่า ถ้าคลาสทั้งสองถูกรวมกันในขั้นตอนการรวมคลาส ผู้ทำการรวมจะต้องสร้างฟังก์ชันการแปลงเพิ่มเติมสำหรับการแปลงให้คลาสทั้งสองมีชนิดข้อมูลเหมือนกัน
2. การเปรียบเทียบคลาสของข้อกำหนดบูรณภาพของคลาสของคลาสที่กำกับคลาสของแอท-ทริบิวต์ (ClassIntegrityConstraint) ใช้เพื่อพิจารณาถึงค่าที่เป็นไปได้ของอินสแตนซ์ในคลาสของแอททริบิวต์นั้นซึ่งแสดงไว้ในรูปคลาสของค่าของข้อกำหนดบูรณภาพ โดยหากคลาสของค่าของข้อกำหนดบูรณภาพของคลาสของแอททริบิวต์หนึ่งมีขอบเขตของค่าไม่อยู่ในขอบเขตของค่าของอีกคลาสหนึ่ง ขั้นตอนนี้จะบันทึกไว้ว่า ถ้าคลาสของแอททริบิวต์ทั้งสองถูกรวมกันในขั้นตอนการรวมคลาสของแอททริบิวต์ ผู้ทำการรวมจะต้องสร้างฟังก์ชันการแปลงเพิ่มเติมสำหรับการแปลงคลาสของค่าของข้อกำหนดบูรณภาพของอินสแตนซ์ในคลาสของแอททริบิวต์ทั้งสองให้ตรงกัน

4.1.2.3 การหาความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในแต่ละคลาสของคลาส

จากการเปรียบเทียบแต่ละคู่ของคลาสของแอททริบิวต์ในหัวข้อที่ 4.1.2.1 จะทำให้ได้ค่า s และระดับความสัมพันธ์ระหว่างคลาสของแอททริบิวต์แต่ละคู่ ซึ่งนอกจากจะนำค่า s ดังกล่าวไปใช้ระบุว่าคลาสคู่นั้นสามารถรวมกันได้หรือไม่แล้ว ค่า s ยังถูกนำมาใช้ในขั้นตอนนี้เพื่อหาความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในแต่ละคลาสของคลาสในหัวข้อนี้ โดยชุดของคลาสของแอททริบิวต์ที่จะนำมาเปรียบเทียบกันจะถูกกำหนดตามกลุ่มของแอททริบิวต์ที่อยู่ภายในคลาสเดียวกันของแบบจำลองข้อมูลเชิงวัตถุเดิม ซึ่งขั้นตอนนี้จะเป็นขั้นตอนสำคัญในการวิเคราะห์โครงสร้างของคลาสภายในโลคัลออนโทโลจีแต่ละชุดที่มีความสัมพันธ์ลักษณะเดียวกัน นั่นคือการเป็นคลาสของแอททริบิวต์ของคลาสใดคลาสหนึ่ง โดยกำหนดให้

C1 เป็นคลาสของคลาสที่จะพิจารณาเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสนั้นจากโลคัลออนโทโลจีชุดที่หนึ่ง

C2 เป็นคลาสของคลาสที่จะพิจารณาเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสนั้นจากโลคัลออนโทโลจีชุดที่สอง

θ_{C1} เป็นจำนวนคลาสของแอททริบิวต์ที่มีอยู่จริงภายในคลาสของคลาส C1

θ_{C2} เป็นจำนวนคลาสของแอททริบิวต์ที่มีอยู่จริงภายในคลาสของคลาส C2

α เป็นจำนวนคลาสของแอททริบิวต์ทั้งหมดภายในคลาสของคลาส C1

β เป็นจำนวนคลาสของแอททริบิวต์ทั้งหมดภายในคลาสของคลาส C2 โดย $\alpha \leq \beta$

วิธีการคำนวณหาค่า α และ β มีขั้นตอนดังนี้

1. ให้ $\alpha = \theta_{C1}$ และ $\beta = \theta_{C2}$
2. ที่คลาสของคลาส C1 ถ้าคลาสของแอททริบิวต์หนึ่งภายใน C1 เป็นคำแม่กลุ่มของคลาสของแอททริบิวต์กลุ่มหนึ่งภายใน C2 ให้ $\alpha = \alpha$ - 1 + จำนวนคลาสของแอททริบิวต์ใน C2 ทั้งหมดที่เป็นคำลูกกลุ่ม และทำซ้ำสำหรับทุกคลาสของแอททริบิวต์ภายใน C1 ที่มีลักษณะเช่นนี้จนครบ
3. ที่คลาสของคลาส C2 ถ้าคลาสของแอททริบิวต์หนึ่งภายใน C2 เป็นคำแม่กลุ่มของคลาสของแอททริบิวต์กลุ่มหนึ่งภายใน C1 ให้ $\beta = \beta$ - 1 + จำนวนคลาสของแอททริบิวต์ใน C1 ทั้งหมดที่เป็นคำลูกกลุ่ม และทำซ้ำสำหรับทุกคลาสของแอททริบิวต์ภายใน C1 ที่มีลักษณะเช่นนี้จนครบ

จากนั้นจะนำค่า s จากการเปรียบเทียบแต่ละคู่ของคลาสของแอททริบิวต์ภายในคลาสของคลาส C1 และ C2 มาเก็บไว้ในเวกเตอร์ $R_{all\text{-}ClassAttribute} = [s_1, s_2, \ldots, s_\alpha]$ และให้ $Ms_{all\text{-}ClassAttribute} = \sum s_i$ โดย $s_i \in R_{all\text{-}ClassAttribute}$ และ $i = 1, \ldots, \alpha$ ทั้งนี้ค่า $Ms_{all\text{-}ClassAttribute}$ ที่ได้จะแสดงระดับความเหมือนและความสัมพันธ์ระหว่างชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส C1 และ C2 ที่นำมาเปรียบเทียบกันซึ่งยึดตามงานวิจัย [20] ดังแสดงในตารางที่ 4.4

ตารางที่ 4.4 ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส C1 และ C2 [20]

กรณี	$Ms_{all\text{-}ClassAttribute}$	ระดับความเหมือน	ความสัมพันธ์
1	α (α=β)	Equivalence	Equivalence
2	α (α<β)	Inclusion	Superclass
3	$\geq$ 3/4α (α=β)	Equivalence	Equivalence
4	$\geq$ 3/4α (α<β)	Inclusion	Superclass
5	$\geq$ 1/2α	Tight-intersection	Sibling
6	< 1/2α	Loose-intersection	Disjoint
7	0	Disjoint	Disjoint

ทั้งนี้ ค่าความสัมพันธ์ที่ได้จากตารางที่ 4.4 จะถูกนำไปพิจารณาเป็นค่าความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาสแต่ละคู่ในหัวข้อที่ 4.1.3 ต่อไป

<u>ตัวอย่างที่ 2</u> จากรูปที่ 4.3 การเปรียบเทียบคลาสของแอททริบิวต์ระหว่างชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 (รูปที่ 3.5) และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2 (รูปที่ 3.6) มีวิธีการดังนี้

กำหนด C1 = Student

C2 = GradStudent

จะได้ θ_{C1} = 6 (StudentName, StudentSex, StudentHeight, StudentAddress, StudentGpa, StudentFaculty)

θ_{C2} = 9 (GradStudentName, GradStudentGender, GradStudentHeight, GradStudentHomeNo, GradStudentHomeCity, GradStudentFacultyName, GradStudentGpa, GradStudentThesis, GradStudentLabName)

คำนวณค่า $\alpha = 6 - 1 + 2 = 7$

$\beta = 9$

เปรียบเทียบระหว่าง StudentName และ GradStudentName

ความสัมพันธ์ = Equivalence, $s_1 = 1$ (ตารางที่ 4.3 กรณีที่ 3)

เปรียบเทียบระหว่าง StudentSex และ GradStudentGender

ความสัมพันธ์ = Synonym, $s_2 = 1$ (ตารางที่ 4.3 กรณีที่ 4)

เปรียบเทียบระหว่าง StudentHeight และ GradStudentHeight

ความสัมพันธ์ = Sibling, $s_3 = 0.1$

(ตารางที่ 4.3 กรณีที่ 1 และ ตารางที่ 4.2 กรณีที่ 5)

เปรียบเทียบระหว่าง StudentAddress และ GradStudentHomeNo

ความสัมพันธ์ = Compound, $s_4 = 1$ (ตารางที่ 4.3 กรณีที่ 6)

เปรียบเทียบระหว่าง StudentAddress และ GradStudentHomeCity

ความสัมพันธ์ = Compound, $s_5 = 1$ (ตารางที่ 4.3 กรณีที่ 6)

เปรียบเทียบระหว่าง StudentGpa และ GradStudentGpa

ความสัมพันธ์ = Equivalence, $s_6 = 1$ (ตารางที่ 4.3 กรณีที่ 3)

เปรียบเทียบระหว่าง StudentFaculty และ GradStudentFacultyName

ความสัมพันธ์ = PathRelated, $s_7 = 1$ (ตารางที่ 4.3 กรณีที่ 5)

$R_{all\text{-}ClassAttribute}$ = [1, 1, 0.1, 1, 1, 1, 1] $\therefore Ms_{all\text{-}ClassAttribute} = 1 + 1 + 0.1 + 1 + 1 + 1 + 1 = 6.1$

$\therefore$ ระดับความเหมือน = Inclusion และ ความสัมพันธ์ = Superclass

(ตารางที่ 4.4 กรณีที่ 4)

รูปที่ 4.3 ตัวอย่างการเปรียบเทียบคลาสของแอททริบิวต์ระหว่างชุดของคลาสของแอททริบิวต์
ภายในคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1
และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2

4.1.3 การเปรียบเทียบคลาสของคลาส

ขั้นตอนนี้เป็นการเปรียบเทียบส่วนคลาสของคลาสภายในโลคัลออนโทโลจี (เทียบได้กับคลาสในแบบจำลองข้อมูลเชิงวัตถุ) ตามการอธิบายผ่านอัปเปอร์ออนโทโลจีในหัวข้อที่ 3.1 การเปรียบเทียบในหัวข้อนี้จะพิจารณาค่าระดับความสัมพันธ์จากการเปรียบเทียบความหมายของคลาสของคลาสในหัวข้อที่ 4.1.1 ร่วมกับค่าระดับความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาสนั้นในหัวข้อที่ 4.1.2.3 โดยเมื่อพิจารณาจากระดับความสัมพันธ์สองลักษณะข้างต้นร่วมกัน จะได้เป็นระดับความสัมพันธ์ของคลาสของคลาสแต่ละคู่ระหว่างโลคัลออนโทโลจีทั้งสองชุดที่นำมาเปรียบเทียบกันดังแสดงในตารางที่ 4.5

<u>ตัวอย่างที่ 3</u> จากรูปที่ 4.1 และ 4.3 การเปรียบเทียบคลาสของคลาสระหว่างคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2 มีวิธีการดังนี้

จากตัวอย่างที่ 1 ค่าความสัมพันธ์จากการเปรียบเทียบความหมายของคลาสของคลาส
= Equivalence

จากตัวอย่างที่ 2 ค่าความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส = Superclass

∴ ความสัมพันธ์ = Superclass (ตารางที่ 4.5 กรณีที่ 2)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4.5 ความสัมพันธ์จากการเปรียบเทียบคลาสของคลาสแต่ละคู่ พิจารณาจาก
ความสัมพันธ์จากการเปรียบเทียบความหมายของคลาสของคลาสร่วมกับ
ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส [20]

กรณี	ความสัมพันธ์จากการเปรียบเทียบความหมายของคลาสของคลาส	ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส	ความสัมพันธ์
1	Equivalence	Equivalence	Equivalence
2	Equivalence	Superclass	Superclass
3	Equivalence	Subclass	Subclass
4	Equivalence	Sibling	Sibling
5	Equivalence	Disjoint	Disjoint
6	Superclass	Equivalence	Equivalence
7	Superclass	Superclass	Superclass
8	Superclass	Subclass	Subclass
9	Superclass	Sibling	Sibling
10	Superclass	Disjoint	Disjoint
11	Subclass	Equivalence	Equivalence
12	Subclass	Superclass	Superclass
13	Subclass	Subclass	Subclass
14	Subclass	Sibling	Sibling
15	Subclass	Disjoint	Disjoint
16	Sibling	Equivalence	Equivalence
17	Sibling	Superclass	Sibling
18	Sibling	Subclass	Sibling
19	Sibling	Sibling	Sibling
20	Sibling	Disjoint	Disjoint
21	Disjoint	Equivalence	Disjoint
22	Disjoint	Superclass	Disjoint
23	Disjoint	Subclass	Disjoint
24	Disjoint	Sibling	Sibling
25	Disjoint	Disjoint	Disjoint
26	ไม่มีการระบุความหมาย	Equivalence	Equivalence
27	ไม่มีการระบุความหมาย	Superclass	Superclass
28	ไม่มีการระบุความหมาย	Subclass	Subclass
29	ไม่มีการระบุความหมาย	Sibling	Sibling
30	ไม่มีการระบุความหมาย	Disjoint	Disjoint

4.2 การรวมโลคัลออนโทโลจี

การรวมโลคัลออนโทโลจีเป็นการนำค่าความสัมพันธ์ระหว่างส่วนประกอบต่างๆ ภายในโลคัลออนโทโลจีทั้งสองชุด ซึ่งเป็นผลจากการเปรียบเทียบโลคัลออนโทโลจีตามวิธีในหัวข้อที่ 4.1 มาวิเคราะห์เพื่อหารูปแบบที่เหมาะสมสำหรับการรวมออนโทโลจีทั้งสองชุดเข้าด้วยกัน โดยประยุกต์จากวิธีการรวมแบบจำลองข้อมูลในงานวิจัย [20] ซึ่งผลที่ได้จากขั้นตอนนี้จะนำไปเป็นแนวทางในการสร้างออนโทโลจีรวม เพื่อนำไปใช้งานในการรวมแบบจำลองข้อมูลที่ต้องการต่อไป ขั้นตอนการรวมโลคัลออนโทโลจีในหัวข้อนี้ประกอบด้วย 3 ขั้นตอนย่อย แบ่งตามส่วนประกอบภายในโลคัลออนโทโลจีที่ต้องการรวมคือ การรวมความหมาย การรวมคลาสของแอททริบิวต์ และการรวมคลาสของคลาส

4.2.1 การรวมความหมาย

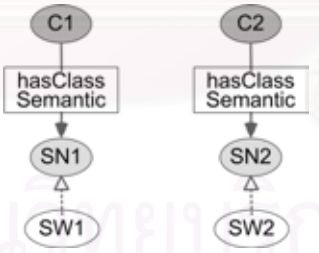
ขั้นตอนนี้เป็นการรวมส่วนความหมายของคลาสของคลาสจากโลคัลออนโทโลจีทั้งสองชุดที่ได้เพิ่มเติมให้กับโลคัลออนโทโลจีแต่ละชุดในหัวข้อที่ 3.2.2 ได้แก่ การรวมส่วนคลาสของชื่อความหมายและคลาสของค่าความหมาย โดยพิจารณาจากระดับความสัมพันธ์ระหว่างคลาสของคลาสที่คลาสของความหมายนั้นอธิบาย ทั้งนี้ ในหัวข้อนี้จะพิจารณาการรวมคลาสของความหมายเฉพาะของคลาสของคลาสเท่านั้น (การรวมคลาสของความหมายของคลาสของแอททริบิวต์จะกล่าวถึงในหัวข้อที่ 4.2.2.1 ต่อไป) รูปแบบในการรวมคลาสของความหมายจะขึ้นกับความสัมพันธ์ที่ได้จากการเปรียบเทียบคลาสของคลาสในหัวข้อที่ 4.1.3 และจะนำคลาสของความหมายรวมที่ได้ไปกำหนดเป็นส่วนความหมายของคลาสนั้นๆ หลังจากการรวมคลาสของคลาสในขั้นตอนการสร้างออนโทโลจีรวม ทั้งนี้คลาสของความหมายรวมที่ได้จะต้องครอบคลุมความหมายของคลาสของคลาสเดิมเสมอ

เมื่อพิจารณาจากความสัมพันธ์ระหว่างคลาสของคลาสแต่ละคู่ที่ได้จากการเปรียบเทียบในหัวข้อที่ 4.1.3 การรวมความหมายจะทำได้ 4 รูปแบบตามระดับความสัมพันธ์ดังนี้

- <u>กรณีที่ 1</u> หากคลาสของคลาสทั้งคู่มีความสัมพันธ์แบบเท่ากัน (Equivalence) คลาสของความหมายที่อธิบายคลาสของคลาสรวมในออนโทโลจีรวมจะได้จากผลการยูเนียน (Union) ระหว่างคลาสของความหมายจากคลาสของคลาสเดิมทั้งสองคลาสในโลคัลออน-โทโลจีคู่ที่นำมารวม
- <u>กรณีที่ 2</u> หากคลาสของคลาสทั้งคู่มีความสัมพันธ์แบบซูเปอร์คลาส/ สับคลาส (Superclass/ Subclass) คลาสของความหมายที่อธิบายคลาสของคลาสที่เป็นซูเปอร์คลาสในออนโทโลจีรวมจะได้จากผลการยูเนียนระหว่างคลาสของความหมายจากคลาสของคลาสเดิมทั้งสองคลาสในโลคัลออนโทโลจีคู่ที่นำมารวม ในขณะที่คลาสของความหมายที่อธิบายคลาสของคลาสที่เป็นสับคลาสในออนโทโลจีรวมจะเท่ากับคลาสของความหมายเดิมที่อธิบายคลาสของคลาสนั้นในโลคัลออนโทโลจีเดิม

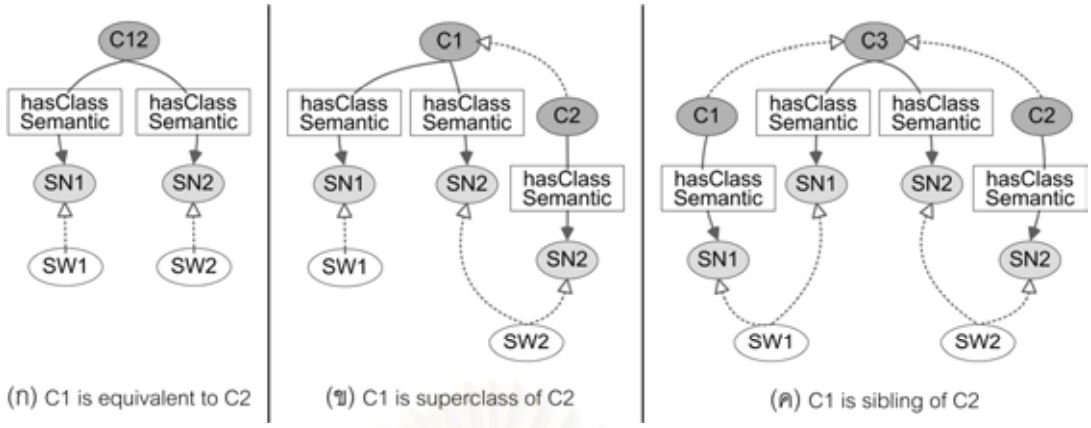
- กรณีที่ 3 หากคลาสของคลาสทั้งคู่มีความสัมพันธ์แบบพี่น้อง (Sibling) คลาสของความหมายที่อธิบายคลาสของคลาสที่เป็นซูเปอร์คลาสของคลาสทั้งสองในออนโทโลจีรวมจะได้จากผลการยูเนียนระหว่างคลาสของความหมายจากคลาสของคลาสเดิมทั้งสองคลาสในโลคัล ออนโทโลจีคู่ที่นำมารวม ในขณะที่คลาสของความหมายที่อธิบายคลาสของคลาสเดิมแต่ละคลาสในออนโทโลจีรวมจะเท่ากับคลาสของความหมายเดิมที่อธิบายคลาสของคลาสนั้นในโลคัลออนโทโลจีแต่ละชุดที่นำมารวม
- กรณีที่ 4 หากคลาสของคลาสทั้งคู่มีความสัมพันธ์แบบแตกต่างกัน (Disjoint) คลาสของคลาสคู่ดังกล่าวจะไม่ถูกนำมารวมในออนโทโลจีรวม ดังนั้นจึงไม่มีการรวมคลาสของความหมายที่อธิบายคลาสของคลาสคู่นั้นๆ

รูปแบบในการรวมคลาสของความหมายเมื่อพิจารณาจากระดับความสัมพันธ์ระหว่างคลาสของคลาสแสดงไว้ในรูปที่ 4.4 และ 4.5 โดยรูปที่ 4.4 แสดงตัวอย่างคลาส C1 ซึ่งเป็นคลาสของคลาสในโลคัลออนโทโลจีชุดที่หนึ่ง และคลาส C2 เป็นคลาสของคลาสในโลคัลออนโทโลจีชุดที่สอง กำหนดให้คลาสของคลาส C1 มีคลาสของชื่อความหมายและคลาสของค่าความหมายเป็น SN1 และ SW1 ส่วนคลาสของคลาส C2 มีคลาสของชื่อความหมายและคลาสของค่าความหมายเป็น SN2 และ SW2 ตามลำดับ รูปที่ 4.5(ก) แสดงการรวมความหมายเมื่อคลาสของคลาส C1 มีความสัมพันธ์แบบเท่ากันกับ C2 (สมมติให้คลาส C12 เป็นคลาสรวมที่เกิดจากการรวม C1 และ C2) รูปที่ 4.5(ข) แสดงการรวมความหมายเมื่อคลาสของคลาส C1 มีความสัมพันธ์แบบซูเปอร์คลาสกับ C2 และรูปที่ 4.5(ค) แสดงการรวมความหมายเมื่อคลาสของคลาส C1 มีความสัมพันธ์แบบพี่น้องกับ C2 (สมมติให้คลาส C3 เป็นซูเปอร์คลาสของ C1 และ C2)



รูปที่ 4.4 ตัวอย่างคลาสของคลาสและคลาสของความหมายที่อธิบายคลาสของคลาส C1 และ C2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.5 รูปแบบในการรวมคลาสของความหมายเมื่อพิจารณาจากระดับความสัมพันธ์ระหว่างคลาสของคลาส C1 และคลาสของคลาส C2

4.2.2 การรวมคลาสของแอททริบิวต์

ขั้นตอนนี้เป็นการรวมส่วนคลาสของแอททริบิวต์ภายในโลคัลออนโทโลจีทั้งสองชุดตามการอธิบายผ่านอัปเปอร์ออนโทโลจีในหัวข้อที่ 3.1 โดยพิจารณาจากระดับความสัมพันธ์ระหว่างคลาสของแอททริบิวต์นั้นกับอีกคลาสของแอททริบิวต์หนึ่งตามผลการเปรียบเทียบแต่ละคู่ของคลาสของแอททริบิวต์ในหัวข้อที่ 4.1.2.1 ซึ่งค่า s ที่ได้จากขั้นตอนดังกล่าวจะเป็นตัวกำหนดรูปแบบในการรวมคลาสของแอททริบิวต์แต่ละคู่ภายในโลคัลออนโทโลจีทั้งสองชุด ทั้งนี้การรวมคลาสของแอททริบิวต์จะเกิดได้เพียงสองลักษณะคือ ทำการรวมคลาสทั้งคู่เข้าด้วยกันหากค่า s > 0 (ความสัมพันธ์ระหว่างคลาสของแอททริบิวต์คู่ดังกล่าวเป็นแบบใดๆ ยกเว้นแบบแตกต่างกัน) และไม่ทำการรวมคลาสหากค่า s = 0 (ความสัมพันธ์ระหว่างคลาสของแอททริบิวต์คู่ดังกล่าวเป็นแบบแตกต่างกัน) หลังจากนั้น คลาสของแอททริบิวต์รวมที่ได้จะถูกนำไปกำหนดเป็นคลาสของแอททริบิวต์ภายในคลาสของคลาสที่เกี่ยวข้องหลังจากการรวมคลาสของคลาสในขั้นตอนการสร้างออนโทโลจีรวม

อย่างไรก็ตาม เนื่องจากคลาสของแอททริบิวต์เหล่านี้ทำหน้าที่แสดงแทนแอททริบิวต์ในแบบจำลองข้อมูล จึงเกี่ยวข้องโดยตรงกับอินสแตนซ์ต่างๆ ในฐานข้อมูล ส่งผลให้ในบางครั้งคลาสของแอททริบิวต์เหล่านี้สามารถบอกถึงความขัดแย้งระหว่างแบบจำลองข้อมูลได้ ซึ่งความขัดแย้งเหล่านี้ส่งผลให้ไม่สามารถรวมคลาสของแอททริบิวต์บางคู่จากโลคัลออนโทโลจีแต่ละชุดเข้าด้วยกันโดยตรงได้ ดังนั้น หากเลือกรวมคลาสของแอททริบิวต์คู่ใดๆ เข้าด้วยกันจึงจำเป็นต้องพิจารณาถึงความหมาย และชนิดข้อมูลของคลาสของแอททริบิวต์ ร่วมกับคลาสของข้อกำหนดบูรณภาพของคลาสของคลาสที่กำกับคลาสของแอททริบิวต์นั้น โดยการพิจารณาประกอบด้วย 3 ขั้นตอนย่อยคือ

4.2.2.1 ความหมายของคลาสของแอททริบิวต์

เมื่อเลือกรวมคลาสของแอททริบิวต์แต่ละคู่เข้าด้วยกันจะต้องพิจารณาถึงการรวมคลาสของความหมายที่ได้เพิ่มเติมให้กับคลาสของแอททริบิวต์ดังกล่าวทั้งส่วนคลาสของชื่อความหมายและคลาสของค่าความหมาย โดยวิธีการรวมความหมายของคลาสของแอททริบิวต์มีความใกล้เคียงกับ

วิธีการรวมความหมายของคลาสของคลาส (หัวข้อที่ 4.2.1) อย่างไรก็ตาม ความหมายของคลาสของแอททริบิวต์กลุ่มนี้อาจบอกได้ถึงความขัดแย้งระหว่างแบบจำลองข้อมูล เช่น บอกความขัดแย้งของหน่วยข้อมูลที่ใช้ เนื่องจากหน่วยการนับของข้อมูลที่ใช้ระหว่างคลาสของแอททริบิวต์ทั้งคู่ที่ต้องการรวมไม่ตรงกัน (มักเกิดขึ้นในกรณีที่คลาสของแอททริบิวต์ทั้งคู่มีคลาสของชื่อความหมายเหมือนกัน แต่คลาสของค่าความหมายต่างกัน ทำให้ได้ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของความหมายเป็นแบบพี่น้อง (ค่า s จากตารางที่ 4.2 และ 4.3 เป็น 0.1)) ทั้งนี้ หากพบความขัดแย้งระหว่างความหมายของคลาสของแอททริบิวต์คู่ที่ต้องการรวมจากโลคัลออนโทโลจีแต่ละชุด ผู้ทำการรวมจะต้องสร้างฟังก์ชันเพื่อแปลงความหมายของคลาสของแอททริบิวต์ทั้งคู่ให้ตรงกันก่อน แล้วจึงนำความหมายรวมที่ได้ไปกำหนดเป็นคลาสของความหมายที่อธิบายคลาสของแอททริบิวต์รวมนั้นๆ หลังจากการรวมคลาสของแอททริบิวต์คู่ดังกล่าวในขั้นตอนการสร้างออนโทโลจีรวม นอกจากนั้น เมื่อนำออนโทโลจีรวมไปใช้ในการเข้าถึงข้อมูลจริงในฐานข้อมูลเชิงวัตถุ ผู้ทำการรวมยังต้องสร้างฟังก์ชันเพื่อแปลงค่าข้อมูลจากทั้งสองฐานข้อมูลให้ตรงกันตามหน่วยข้อมูลที่เลือกใช้ด้วย

<u>ตัวอย่างที่ 4</u> การรวมคลาสของแอททริบิวต์ระหว่างคลาสของแอททริบิวต์ StudentName ในโลคัลออนโทโลจีที่ 1 และคลาสของแอททริบิวต์ GradStudentName ในโลคัลออนโทโลจีที่ 2 (จากการเปรียบเทียบในรูปที่ 4.3) มีวิธีการดังนี้

จากผลการเปรียบเทียบในตัวอย่างที่ 2 ค่าความสัมพันธ์จากการเปรียบเทียบคลาสของแอททริบิวต์ = Equivalence และ s = 1

∴ คลาสของแอททริบิวต์คู่นี้สามารถรวมกันได้

<u>ตัวอย่างที่ 5</u> การรวมคลาสของแอททริบิวต์ระหว่างคลาสของแอททริบิวต์ StudentHeight ในโลคัลออนโทโลจีที่ 1 และคลาสของแอททริบิวต์ GradStudentHeight ในโลคัลออนโทโลจีที่ 2 (จากการเปรียบเทียบในรูปที่ 4.3) มีวิธีการดังนี้

จากผลการเปรียบเทียบในตัวอย่างที่ 2 ค่าความสัมพันธ์จากการเปรียบเทียบคลาสของแอททริบิวต์ = Sibling และ s = 0.1

∴ คลาสของแอททริบิวต์คู่นี้สามารถรวมกันได้ แต่ต้องการฟังก์ชันเพื่อแปลงความหมายส่วนคลาสของค่าความหมายที่ขัดแย้งกันของคลาสของแอททริบิวต์รวมจาก ‘Centimetre’ และ ’Metre’ ให้มีค่าตรงกัน เช่น เลือกเป็น ‘Centemetre’ นอกจากนี้ เมื่อนำออนโทโลจีรวมไปใช้งานจำเป็นต้องสร้างฟังก์ชันเพื่อแปลงค่าข้อมูลจากทั้งสองฐานข้อมูลให้ตรงกันอีกด้วย ตัวอย่างเช่น หากเลือกคลาสของค่าความหมายเป็น ‘Centimetre’ จะต้องสร้างฟังก์ชันเพื่อแปลงค่าข้อมูลที่ได้จากการสอบถามจากทั้งสองฐานข้อมูลให้อยู่ในหน่วยเซนติเมตร

4.2.2.2 ชนิดข้อมูลของคลาสของแอททริบิวต์

เมื่อเลือกรวมคลาสของแอททริบิวต์แต่ละคู่เข้าด้วยกันจะต้องพิจารณาถึงชนิดข้อมูลของคลาสของแอททริบิวต์แต่ละคลาสที่นำมารวมด้วย เนื่องจากชนิดข้อมูลของคลาสของแอททริบิวต์ที่

ไม่ตรงกันระหว่างโลคัลออนโทโลจีทั้งสองชุดแสดงให้เห็นอย่างชัดเจนถึงความขัดแย้งของโครงสร้างระหว่างแบบจำลองข้อมูลเชิงวัตถุที่นำมารวม ดังนั้น หากพบความขัดแย้งระหว่างชนิดข้อมูลของคลาสของแอททริบิวต์ที่ต้องการรวมจากโลคัลออนโทโลจีทั้งสองชุด (ซึ่งพิจารณาได้จากผลการเปรียบเทียบชนิดข้อมูลของคลาสของแอททริบิวต์ในหัวข้อที่ 4.1.2.2) ผู้ทำการรวมจะต้องสร้างฟังก์ชันเพื่อแปลงชนิดข้อมูลของคลาสของแอททริบิวต์ทั้งคู่ให้ตรงกันก่อน แล้วจึงนำชนิดข้อมูลที่ได้ไปกำหนดเป็นชนิดข้อมูลของคลาสของแอททริบิวต์รวมนั้นๆ หลังจากการรวมคลาสของแอททริบิวต์คู่ดังกล่าวในขั้นตอนการสร้างออนโทโลจีรวม โดยมีเงื่อนไขในการแปลงชนิดข้อมูลดังนี้

- กรณีที่ 1 หากชนิดข้อมูลของคลาสของแอททริบิวต์ทั้งสองคลาสเป็นชนิดข้อมูลพื้นฐานที่ต่างชนิดกัน ได้แก่ String, Integer และ Real ให้ดำเนินการแปลงชนิดข้อมูลให้ตรงกันเพื่อใช้กำหนดเป็นชนิดข้อมูลของคลาสของแอททริบิวต์รวม เช่น ระหว่าง Integer กับ Real ให้แปลงชนิดข้อมูลของทั้งคู่เป็น Real
- กรณีที่ 2 หากชนิดข้อมูลของคลาสของแอททริบิวต์หนึ่งเป็นชนิดข้อมูลพื้นฐาน แต่อีกคลาสของแอททริบิวต์หนึ่งเป็นชนิดข้อมูลที่ผู้ใช้กำหนด (อันเป็นผลจากความสัมพันธ์แบบแอกกรีเกชันระหว่างคลาสในแบบจำลองข้อมูลเชิงวัตถุ) หรือเรียกได้ว่าเป็นความขัดแย้งของการแอกกรีเกชัน ซึ่งตามการอธิบายผ่านอัปเปอร์ออนโทโลจีในวิทยานิพนธ์นี้ ชนิดข้อมูลที่ผู้ใช้กำหนดจะถูกสร้างขึ้นเป็นคลาสของคลาสฯ หนึ่งในโลคัลออนโทโลจีชุดนั้น ในกรณีนี้ชนิดข้อมูลของคลาสของแอททริบิวต์รวมที่ได้จะถูกกำหนดให้เป็นคลาสของคลาสด้วย โดยจะยึดคลาสของแอททริบิวต์ที่มีชนิดข้อมูลเป็นคลาสของคลาสอยู่ก่อนแล้วเป็นหลัก แล้วสร้างฟังก์ชันเพื่อแปลงชนิดข้อมูลของคลาสของแอททริบิวต์ที่เดิมเป็นชนิดข้อมูลพื้นฐานให้แสดงค่าข้อมูลทั้งหมดตรงกัน
- กรณีที่ 3 หากชนิดข้อมูลของคลาสของแอททริบิวต์ทั้งสองคลาสเป็นชนิดข้อมูลที่ผู้ใช้กำหนดทั้งคู่ นั่นคือชนิดข้อมูลของคลาสของแอททริบิวต์แต่ละคลาสจะเป็นคลาสของคลาสใดคลาสหนึ่งภายในโลคัลออนโทโลจีแต่ละชุด ให้พิจารณาชนิดข้อมูลของคลาสของแอท-ทริบิวต์รวมจากความระดับสัมพันธ์ระหว่างคลาสของคลาสที่เป็นชนิดข้อมูลของคลาสของแอททริบิวต์ทั้งสอง (หัวข้อที่ 4.1.3) หากความสัมพันธ์ระหว่างคลาสของคลาสทั้งคู่เป็นแบบซูเปอร์คลาสและสับคลาส ให้เลือกคลาสของคลาสที่เป็นซูเปอร์คลาสเป็นชนิดข้อมูลของคลาสของแอททริบิวต์รวม แต่หากความสัมพันธ์ระหว่างคลาสของคลาสทั้งคู่เป็นแบบพี่น้องชนิดข้อมูลของคลาสของแอททริบิวต์รวมจะเป็นคลาสของคลาสที่เป็นซูเปอร์คลาสของคลาสพี่น้องทั้งสองคลาส

4.2.2.3 คลาสของข้อกำหนดบูรณภาพของคลาสของคลาสที่กำกับคลาสของแอททริบิวต์

คลาสของข้อกำหนดบูรณภาพของคลาสของคลาสที่กำกับคลาสของแอททริบิวต์เป็นตัวกำหนดเงื่อนไขบังคับของคลาสของแอททริบิวต์นั้นๆ ว่าค่าที่เป็นไปได้ของอินสแตนซ์ของคลาส

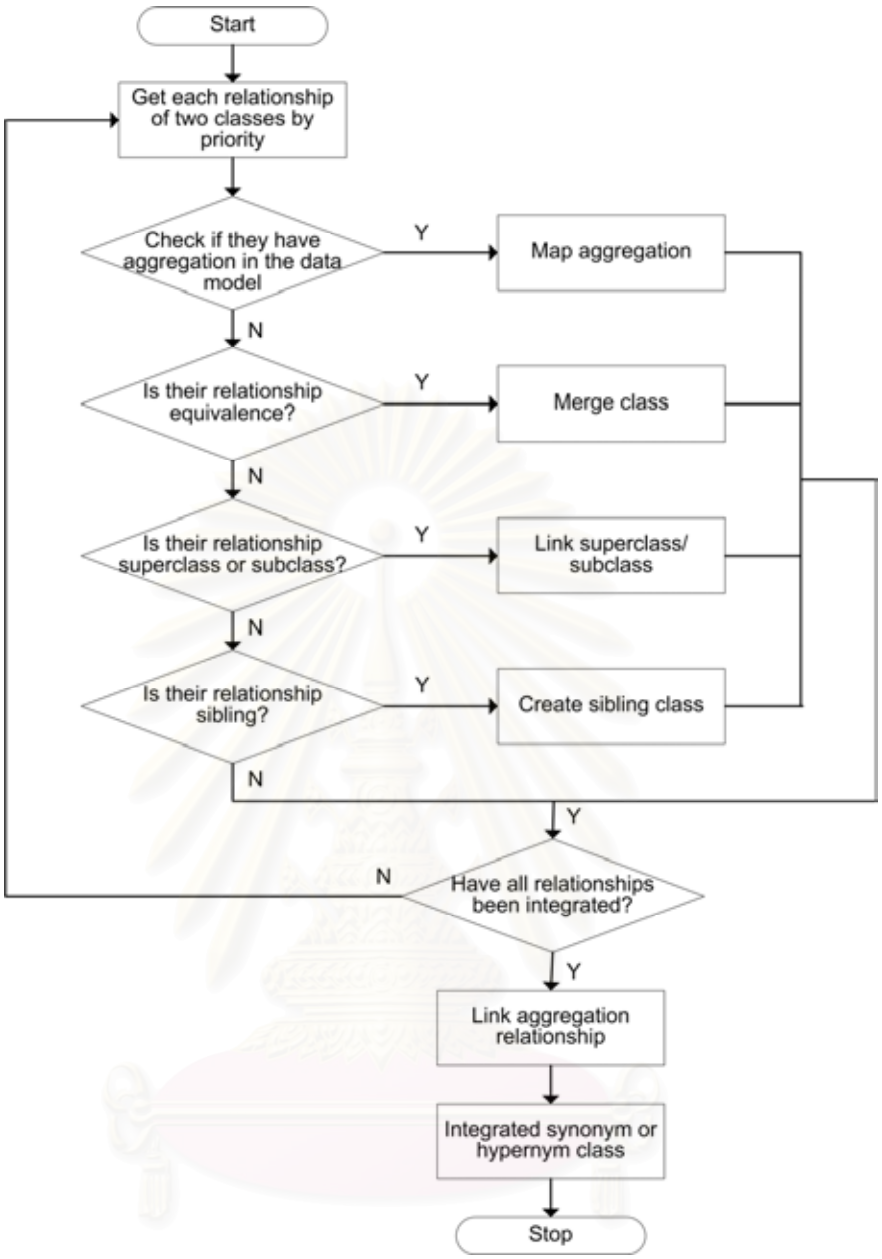
ของแอททริบิวต์นั้นคืออะไรและจะต้องเป็นไปตามที่กำหนดไว้เท่านั้น ดังนั้น คลาสของข้อกำหนดบูรณภาพของคลาสของคลาสที่กำกับคลาสของแอททริบิวต์ที่ไม่ตรงกันระหว่างโลคัลออนโทโลจีทั้งสองชุดแสดงให้เห็นอย่างชัดเจนว่า มีความขัดแย้งของค่าข้อมูลระหว่างคลาสของแอททริบิวต์คู่ดังกล่าว หากพบความขัดแย้งในลักษณะนี้ (ซึ่งพิจารณาได้จากผลการเปรียบเทียบชนิดข้อมูลของคลาสของแอททริบิวต์ในหัวข้อที่ 4.1.2.2) ผู้ทำการรวมจะต้องสร้างฟังก์ชันเพื่อแปลงค่าคลาสของข้อกำหนดบูรณภาพคู่ดังกล่าวให้ตรงกันก่อน แล้วจึงนำคลาสของข้อกำหนดบูรณภาพที่ได้ไปกำหนดเป็นคลาสของข้อกำหนดบูรณภาพของคลาสของคลาสที่กำกับคลาสของแอททริบิวต์รวมนั้นๆ หลังจากการรวมคลาสของแอททริบิวต์คู่ดังกล่าวในขั้นตอนการสร้างออนโทโลจีรวม

การแปลงคลาสของข้อกำหนดบูรณภาพของคลาสที่กำกับคลาสของแอททริบิวต์จะพิจารณาจากคลาสของค่าของข้อกำหนดบูรณภาพที่เป็นสับคลาสของคลาสของข้อกำหนดบูรณ-ภาพนั้น โดยผู้ทำการรวมสามารถเลือกยึดคลาสของค่าของข้อกำหนดบูรณภาพที่กำกับคลาสของแอททริบิวต์คลาสใดคลาสหนึ่งภายในโลคัลออนโทโลจีชุดหนึ่งเป็นหลัก แล้วสร้างฟังก์ชันเพื่อแปลงคลาสของค่าของข้อกำหนดบูรณภาพที่กำกับคลาสของแอททริบิวต์อีกคลาสให้ตรงกัน ก่อนจะนำคลาสของข้อกำหนดบูรณภาพที่ได้ไปกำหนดเป็นคลาสของข้อกำหนดบูรณภาพที่กำกับคลาสของแอททริบิวต์รวมนั้น หลังจากการรวมคลาสของแอททริบิวต์คู่ดังกล่าวในขั้นตอนการสร้างออนโทโลจีรวม

4.2.3 การรวมคลาสของคลาส

ขั้นตอนนี้เป็นการรวมส่วนคลาสของคลาสภายในโลคัลออนโทโลจีทั้งสองชุดตามการอธิบายผ่านอัปเปอร์ออนโทโลจีในหัวข้อที่ 3.1 โดยพิจารณาจากระดับความสัมพันธ์ระหว่างคลาสของคลาสนั้นกับอีกคลาสของคลาสหนึ่งตามผลการเปรียบเทียบแต่ละคู่ของคลาสของคลาสในหัวข้อที่ 4.1.3 ได้แก่ ความสัมพันธ์แบบเท่ากัน ความสัมพันธ์แบบซูเปอร์คลาส ความสัมพันธ์แบบสับคลาส ความสัมพันธ์แบบพี่น้อง และความสัมพันธ์แบบแตกต่างกัน ซึ่งค่าความสัมพันธ์เหล่านี้แสดงถึงระดับความเหมือนและความแตกต่างระหว่างคลาสแต่ละคู่ในโลคัลออนโทโลจีสองชุดในระดับที่มากน้อยต่างกัน และเป็นตัวตัดสินว่าควรจะรวมคลาสของคลาสหนึ่งในโลคัลออนโทโลจีชุดที่หนึ่งกับคลาสของคลาสใดบ้างในโลคัลออนโทโลจีชุดที่สอง ด้วยการรวมแบบใด รวมทั้งมีลำดับการรวมก่อนหลังอย่างไร ทั้งนี้ ลำดับการรวมคลาสของคลาสที่ต่างกันจะส่งผลให้รูปแบบของออนโทโลจีรวมที่ได้แตกต่างกันด้วย

วิทยานิพนธ์ฉบับนี้ยึดแนวทางในการจัดลำดับความสำคัญ (Priority) ของความสัมพันธ์ที่คลาสของคลาสหนึ่งในโลคัลออนโทโลจีชุดหนึ่งมีต่อคลาสของคลาสต่างๆ แต่ละคลาสในโลคัลออนโทโลจีอีกชุดหนึ่งตามงานวิจัย [20] ซึ่งเรียงระดับความสัมพันธ์จากมากไปน้อย ดังนี้ (1) ความสัมพันธ์แบบเท่ากัน (2) ความสัมพันธ์แบบซูเปอร์คลาส/ แบบสับคลาส (3) ความสัมพันธ์แบบพี่น้อง และ (4) ความสัมพันธ์แบบแตกต่างกัน โดยคลาสของคลาสคู่ที่มีระดับความสัมพันธ์ระหว่างกันที่มากกว่าจะได้รับการพิจารณาเพื่อรวมเข้าด้วยกันก่อน



รูปที่ 4.6 ขั้นตอนโดยรวมในการรวมคลาส [20]

รูปที่ 4.6 แสดงขั้นตอนโดยรวมในการรวมคลาสของคลาส โดยสามารถอธิบายลำดับและเงื่อนไขในการรวมคลาสของคลาสหนึ่งในโลคัลออนโทโลจีชุดที่หนึ่ง กับคลาสของคลาสอื่นๆ ในโลคัลออนโทโลจีชุดที่สองได้ดังนี้ สมมติว่าต้องการรวมคลาสของคลาส C1 ในโลคัลออนโทโลจีชุดที่หนึ่งกับคลาสของคลาส C2, C3 และ C4 ในโลคัลออนโทโลจีชุดที่สอง ซึ่งค่าความสัมพันธ์ระหว่างคลาสของคลาส C1 กับคลาสของคลาส C2, C3 และ C4 จากการเปรียบเทียบคลาสของคลาสแต่ละคู่มีค่าเป็นความสัมพันธ์แบบเท่ากัน แบบซูเปอร์คลาส และแบบพี่น้องตามลำดับ เมื่อจัดลำดับความสำคัญของความสัมพันธ์จากมากไปน้อยจะได้เป็นแบบเท่ากัน แบบซูเปอร์คลาส และแบบพี่น้อง ดังนั้นในการรวมออนโทโลจีจึงเริ่มจากการรวมคลาสของคลาส C1 และ C2 ตามรูปแบบการรวมคลาสของคลาสที่มีความสัมพันธ์แบบเท่ากัน จากนั้นจึงรวมคลาสของคลาส C1 และ C3 ตามรูปแบบการรวมคลาสของ

คลาสที่มีความสัมพันธ์แบบซูเปอร์คลาส และสุดท้ายจึงรวมคลาสของคลาส C1 และ C4 ตามรูปแบบการรวมคลาสของคลาสที่มีความสัมพันธ์แบบพี่น้อง

ทั้งนี้ ในการรวมคลาสของคลาสแต่ละคู่ภายในโลคัลออนโทโลจีสองชุดเข้าด้วยกันจำเป็นต้องพิจารณาถึงลำดับความสำคัญของระดับความสัมพันธ์ระหว่างคลาสของคลาสทุกคู่ในโลคัลออนโทโลจีทั้งสองชุด การจัดการกับบางคลาสของคลาสที่ถูกกำหนดให้เป็นชนิดข้อมูลของบางคลาสของแอททริบิวต์ (เทียบได้กับความสัมพันธ์แบบแอกกรีเกชันในแบบจำลองข้อมูลเชิงวัตถุ) และการรวมคลาสของแอททริบิวต์ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันหรือ เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างโลคัลออนโทโลจีสองชุดที่นำมารวม ดังนั้นในการรวมคลาสของคลาสอาจแยกพิจารณาได้เป็น 3 ส่วนย่อยคือ

4.2.3.1 การรวมคลาสของคลาสโดยการพิจารณาจากระดับความสัมพันธ์

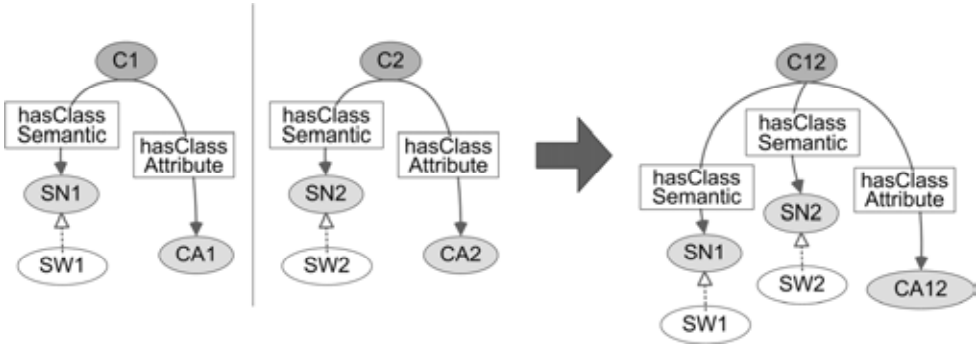
จากค่าความสัมพันธ์จากการเปรียบเทียบคลาสของคลาสแต่ละคู่ที่ได้จากการเปรียบเทียบในขั้นตอนที่ 4.1.3 ได้แก่ ความสัมพันธ์แบบเท่ากัน ความสัมพันธ์แบบซูเปอร์คลาส/ สับคลาส ความสัมพันธ์แบบพี่น้องและความสัมพันธ์แบบแตกต่างกัน รวมกับความสัมพันธ์ระหว่างคลาสของคลาสบางคู่ที่มีคลาสหนึ่งถูกกำหนดให้เป็นชนิดข้อมูลของบางคลาสของแอททริบิวต์ การรวมคลาสของคลาสโดยการพิจารณาจากระดับความสัมพันธ์สามารถทำได้ 5 แบบดังนี้คือ

- <u>กรณีที่ 1</u> การรวมคลาสของคลาสที่มีความสัมพันธ์แบบเท่ากัน (Merge class ในรูปที่ 4.6)

  หากคลาสของคลาส C1 ในโลคัลออนโทโลจีชุดที่หนึ่งมีความสัมพันธ์แบบเท่ากันกับคลาสของคลาส C2 ในโลคัลออนโทโลจีชุดที่สอง การรวมคลาสของคลาสมีขั้นตอนคือ

  1. รวมความหมายของคลาสของคลาส C1 และ C2 ตามวิธีการรวมความหมายในหัวข้อที่ 4.2.1 แล้วกำหนดเป็นความหมายของคลาสของคลาสรวม
  2. รวมคลาสของแอททริบิวต์ภายในคลาสของคลาส C1 และ C2 โดยพิจารณาจากความสัมพันธ์ระหว่างคลาสของแอททริบิวต์แต่ละคู่ตามวิธีการรวมคลาสของแอททริบิวต์ในหัวข้อที่ 4.2.2 แล้วกำหนดเป็นคลาสของแอททริบิวต์ของคลาสของคลาสรวม

  เมื่อกำหนดให้คลาสของคลาส C1 มีคลาสของแอททริบิวต์เป็น CA1 และมีคลาสของชื่อความหมายเป็น SN1 (มีคลาสของค่าความหมายเป็น SW1) ในขณะที่คลาสของคลาส C2 มีคลาสของแอททริบิวต์เป็น CA2 และมีคลาสของชื่อความหมายเป็น SN2 (มีคลาสของค่าความหมายเป็น SW2) ตามลำดับ และกำหนดให้คลาสของแอททริบิวต์ CA1 และ CA2 มีความสัมพันธ์แบบเท่ากัน (s = 1) รูปที่ 4.7 แสดงตัวอย่างการรวมคลาสของคลาสที่มีความสัมพันธ์แบบเท่ากัน โดยสมมติให้คลาสของคลาส C12 และคลาสของแอททริบิวต์ CA12 เป็นคลาสของคลาสรวมและคลาสของแอททริบิวต์รวมที่เกิดจากการรวมคลาสของคลาส C1 กับ C2 และคลาสของแอททริบิวต์ CA1 กับ CA2 ตามลำดับ

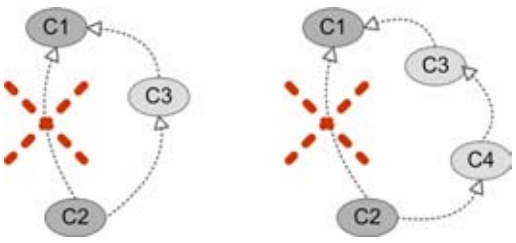


รูปที่ 4.7 ตัวอย่างการรวมคลาสของคลาส C1 และ C2 ที่มีความสัมพันธ์แบบเท่ากัน

- <u>กรณีที่ 2</u> การรวมคลาสของคลาสที่มีความสัมพันธ์แบบซูเปอร์คลาส/ สับคลาส (Link superclass/ subclass ในรูปที่ 4.6)

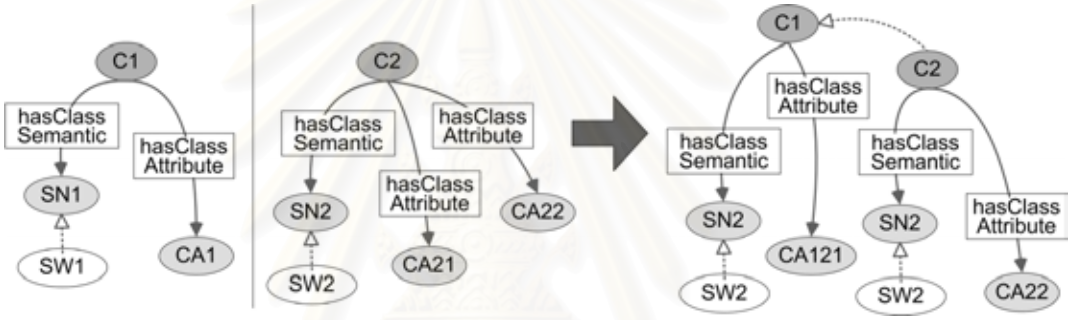
หากคลาสของคลาส C1 ในโลคัลออนโทโลจีชุดที่หนึ่งมีความสัมพันธ์แบบซูเปอร์คลาสกับคลาสของคลาส C2 ในโลคัลออนโทโลจีชุดที่สองหรือในอีกแง่หนึ่งคลาสของคลาส C2 ในโลคัลออนโทโลจีชุดที่สองมีความสัมพันธ์แบบสับคลาสกับคลาสของคลาส C1 ในโลคัลออนโทโลจีชุดที่หนึ่ง การรวมคลาสของคลาสมีขั้นตอนคือ

1. เชื่อมความสัมพันธ์ระหว่างคลาสของคลาสทั้งสองโดยกำหนดให้คลาสของคลาส C1 เป็นซูเปอร์คลาสของคลาส C2 ภายในออนโทโลจีรวม
2. รวมความหมายของคลาสของคลาส C1 และ C2 ตามวิธีการรวมความหมายในหัวข้อที่ 4.2.1 แล้วกำหนดเป็นความหมายของคลาสของคลาส C1 และ C2
3. รวมคลาสของแอททริบิวต์ภายในคลาสของคลาส C1 และ C2 โดยพิจารณาจากความสัมพันธ์ระหว่างคลาสของแอททริบิวต์แต่ละคู่ตามวิธีการรวมคลาสของแอททริบิวต์ในหัวข้อที่ 4.2.2 แล้วกำหนดเป็นคลาสของแอททริบิวต์ของคลาสของคลาส C1 และเมื่อคลาสของแอททริบิวต์คู่ใดถูกนำมารวมไว้ที่คลาส C1 แล้วให้ตัดคลาสดังกล่าวออกจากการเป็นคลาสของแอททริบิวต์ของคลาส C2
4. กำจัดเส้นทางการสืบทอดระหว่างคลาสของคลาสที่ซ้ำซ้อน โดยตรวจสอบว่าหากภายในออนโทโลจีรวมมีเส้นทางการสืบทอดจากคลาสของคลาส C2 ไปยังคลาสของคลาสอื่นซึ่งเป็นสับคลาสของคลาส C1 อยู่ก่อนแล้ว ให้ตัดความสัมพันธ์ที่เชื่อมระหว่างคลาสของคลาส C1 และ C2 ทิ้ง และใช้การสืบทอดที่มีอยู่ก่อนแล้วแทนเพื่อกำจัดลูป (Loop) ที่เกิดขึ้นระหว่างคลาสของคลาส C1 กับ C2 ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 การกำจัดเส้นทางการสืบทอดระหว่างคลาสของคลาสที่ซ้ำซ้อน

เมื่อกำหนดให้คลาสของคลาส C1 มีคลาสของแอททริบิวต์เป็น CA1 และมีคลาสของชื่อความหมายเป็น SN1 (มีคลาสของค่าความหมายเป็น SW1) ในขณะที่คลาสของคลาส C2 มีคลาสของแอททริบิวต์เป็น CA21 และ CA22 และมีคลาสของชื่อความหมายเป็น SN2 (มีคลาสของค่าความหมายเป็น SW2) ตามลำดับ และกำหนดให้คลาสของแอททริบิวต์ CA1 และ CA21 มีความสัมพันธ์แบบเท่ากัน (s = 1) และคลาสของความหมายของคลาสของคลาส C1 มีความสัมพันธ์แบบสับคลาสกับคลาสของความหมายของคลาสของคลาส C2 รูปที่ 4.9 แสดงตัวอย่างการรวมคลาสของคลาสที่มีความสัมพันธ์แบบซูเปอร์คลาส โดยสมมติให้คลาสของแอททริบิวต์ CA121 เป็นคลาสของแอททริบิวต์รวมที่เกิดจากการรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และคลาสของชื่อความหมาย SN2 ครอบคลุมคำศัพท์ทั้งหมดของคลาส SN1 (ในขณะเดียวกันคลาสของค่าความหมาย SW2 ครอบคลุมคำศัพท์ทั้งหมดของคลาส SW1 ด้วย)



รูปที่ 4.9 ตัวอย่างการรวมคลาสของคลาส C1 และ C2 ที่มีความสัมพันธ์แบบซูเปอร์คลาส/ สับคลาส

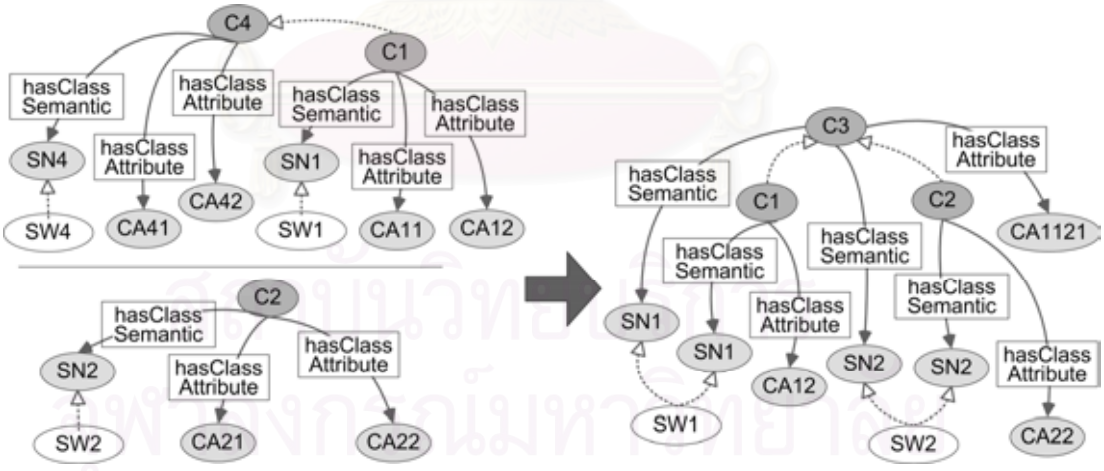
- <u>กรณีที่ 3</u> การรวมคลาสที่มีความสัมพันธ์แบบพี่น้อง (Create sibling class ในรูปที่ 4.6)

หากคลาสของคลาส C1 ในโลคัลออนโทโลจีชุดที่หนึ่งมีความสัมพันธ์แบบพี่น้องกับคลาสของคลาส C2 ในโลคัลออนโทโลจีชุดที่สอง ก่อนจะรวมคลาสของคลาสคู่นี้ตามความสัมพันธ์แบบพี่น้องจะต้องตรวจสอบให้แน่ใจก่อนว่า ทั้งคลาสของคลาส C1 และ C2 มิได้เป็นสับคลาสของคลาสใดภายในออนโทโลจีรวมอยู่ก่อนแล้ว (ซึ่งหมายความว่าคลาสของคลาส C1 และ C2 เป็นคลาสพี่น้องกันอยู่แล้วจึงไม่จำเป็นต้องรวมกันอีก) รวมทั้งต้องแน่ใจว่า คลาสของคลาสทั้งคู่ไม่มีซูเปอร์คลาสโดยตรง (Immediate Superclass) ในแต่ละโลคัลออนโทโลจีเดิม (มิฉะนั้นจะใช้การรวมคลาสของคลาสที่เหลือตามความสัมพันธ์แบบสับคลาสกับซูเปอร์คลาสที่มีอยู่แทน) การรวมคลาสของคลาสมีขั้นตอนคือ

1. สร้างคลาสของคลาส C3 ขึ้นใหม่ภายในออนโทโลจีรวม เพื่อกำหนดให้ทำหน้าเป็นซูเปอร์คลาสโดยตรงของทั้งคลาสของคลาส C1 และ C2
2. เชื่อมความสัมพันธ์ระหว่างคลาสของคลาส C1, C2 และ C3 โดยกำหนดให้ทั้งคลาสของคลาส C1 และ C2 เป็นสับคลาสของคลาส C3 ภายในออนโทโลจีรวม
3. รวมความหมายของคลาสของคลาส C1 และ C2 ตามวิธีการรวมความหมายในหัวข้อที่ 4.2.1 แล้วกำหนดเป็นความหมายของคลาสของคลาส C1, C2 และ C3

4. รวมคลาสของแอททริบิวต์ภายในคลาสของคลาส C1 และ C2 โดยพิจารณาจากความสัมพันธ์ระหว่างคลาสของแอททริบิวต์แต่ละคู่ตามวิธีการรวมคลาสของแอททริบิวต์ในหัวข้อที่ 4.2.2 แล้วกำหนดเป็นคลาสของแอททริบิวต์ของคลาสของคลาส C3 และเมื่อคลาสของแอททริบิวต์คู่ใดถูกนำมารวมไว้ที่คลาส C3 แล้วให้ตัดคลาสดังกล่าวออกจากการเป็นคลาสของแอททริบิวต์ของทั้งคลาส C1 และ C2
5. ตรวจสอบความสัมพันธ์ระหว่างคลาสของคลาส C3 ที่สร้างขึ้นใหม่กับคลาสของคลาสอื่นทั้งหมดที่เป็นซูเปอร์คลาสของคลาส C1 และ C2 ภายในออนโทโลจีรวม ถ้าพบว่าคลาสของคลาส C3 มีความสัมพันธ์รูปแบบใดก็ตามกับคลาสเหล่านั้นให้รวมคลาสของคลาส C3 กับคลาสดังกล่าวตามรูปแบบความสัมพันธ์ที่ได้รับจากการเปรียบเทียบ

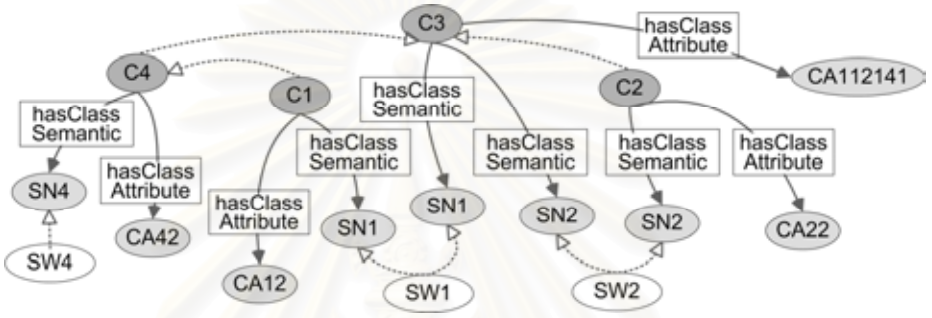
เมื่อกำหนดให้คลาสของคลาส C1 เป็นสับคลาสของคลาส C4 โดยมีคลาสของแอททริบิวต์เป็น CA11 และ CA12 และมีคลาสของชื่อความหมายเป็น SN1 (มีคลาสของค่าความหมายเป็น SW1) ในขณะที่คลาสของคลาส C2 มีคลาสของแอททริบิวต์เป็น CA21 และ CA22 และมีคลาสของชื่อความหมายเป็น SN2 (มีคลาสของค่าความหมายเป็น SW2) ตามลำดับ และกำหนดให้คลาสของแอททริบิวต์ CA11 และ CA21 มีความสัมพันธ์แบบเท่ากัน (s = 1) รูปที่ 4.10 แสดงตัวอย่างการรวมคลาสของคลาสที่มีความสัมพันธ์แบบพี่น้อง โดยกำหนดให้สร้างคลาสของคลาส C3 ขึ้นใหม่เพื่อทำหน้าที่เป็นซูเปอร์คลาสโดยตรงของคลาสของคลาส C1 และ C2 และสมมติให้คลาสของแอททริบิวต์ CA1121 เป็นคลาสของแอททริบิวต์รวมที่เกิดจากการรวมคลาสของแอททริบิวต์ CA11 กับ CA21



รูปที่ 4.10 ตัวอย่างการรวมคลาสของคลาส C1 และ C2 ที่มีความสัมพันธ์แบบพี่น้อง

จากรูปที่ 4.10 หลังจากสร้างคลาสของคลาส C3 เพื่อรวมคลาสของคลาส C1 และ C2 เข้าด้วยกันตามการรวมด้วยความสัมพันธ์แบบพี่น้อง จะต้องตรวจสอบความสัมพันธ์ระหว่างคลาสของคลาส C3 และ C4 ซึ่งมีคลาสของแอททริบิวต์เป็น CA41 และ CA42 และมีคลาสของชื่อความหมายเป็น SN4 (มีคลาสของค่าความหมายเป็น SW4) ในที่นี้กำหนดให้คลาสของคลาส C3 มีความสัมพันธ์แบบซูเปอร์คลาสกับคลาสของคลาส C4

โดยคลาสของแอททริบิวต์ CA1121 และ CA41 มีความสัมพันธ์แบบเท่ากัน (s = 1) และคลาสของความหมายของคลาสของคลาส C3 มีความสัมพันธ์แบบซูเปอร์คลาสกับคลาสของความหมายของคลาสของคลาส C4 รูปที่ 4.11 แสดงตัวอย่างการรวมคลาสของคลาสที่เกิดขึ้น โดยสมมติให้คลาสของแอททริบิวต์ CA112141 เป็นคลาสของแอททริบิวต์รวมที่เกิดจากการรวมคลาสของแอททริบิวต์ CA1121 กับ CA41 และคลาสของชื่อความหมาย SN1 และ SN2 ของคลาสของคลาส C3 ครอบคลุมคำศัพท์ทั้งหมดของคลาส SN4 (ในขณะเดียวกันคลาสของค่าความหมาย SW1 และ SW2 ครอบคลุมคำศัพท์ทั้งหมดของคลาส SW4 ด้วย)



รูปที่ 4.11 ตัวอย่างการรวมคลาสของคลาส C3 ที่เกิดจากการรวมคลาสของคลาส C1 และ C2 ที่มีความสัมพันธ์แบบพี่น้องและ C4 ที่เป็นซูเปอร์คลาสของคลาสเดิม

- <u>กรณีที่ 4</u> การรวมคลาสที่มีความสัมพันธ์แบบแตกต่างกัน

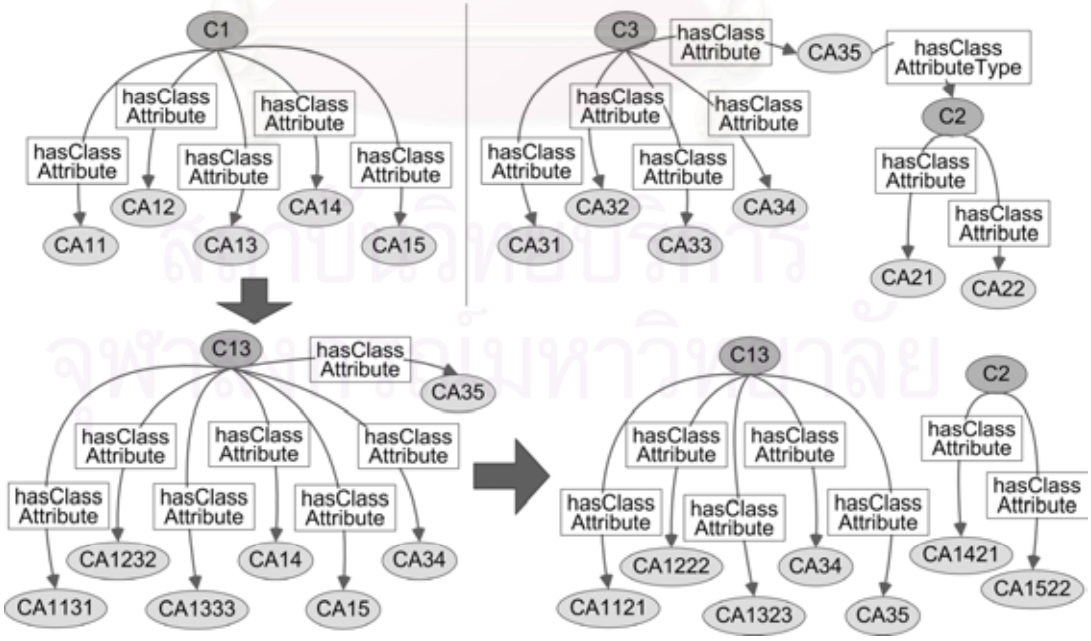
  หากคลาสของคลาส C1 ในโลคัลออนโทโลจีชุดที่หนึ่งมีความสัมพันธ์แบบแตกต่างกันกับคลาสของคลาส C2 ในโลคัลออนโทโลจีชุดที่สอง จะหมายความว่าคลาสของคลาสทั้งคู่ไม่มีเกี่ยวข้องใดๆ กันเลย ดังนั้นคลาสคู่นี้จึงไม่ถูกรวมเข้าด้วยกันภายในออนโทโลจีรวม

- <u>กรณีที่ 5</u> การรวมคลาสของคลาสที่ถูกกำหนดให้เป็นชนิดข้อมูลของบางคลาสของแอททริ-บิวต์ในโลคัลออนโทโลจีชุดใดชุดหนึ่ง (Map aggregation ในรูปที่ 4.6)

  หากคลาสของคลาส C1 ในโลคัลออนโทโลจีชุดที่หนึ่งมีความสัมพันธ์แบบเท่ากัน ความสัมพันธ์แบบซูเปอร์คลาส ความสัมพันธ์แบบสับคลาส หรือความสัมพันธ์แบบพี่น้องกับคลาสของคลาส C2 ในโลคัลออนโทโลจีชุดที่สอง แต่คลาสของคลาส C1 มีความสัมพันธ์แบบเท่ากันกับคลาสของคลาส C3 ในโลคัลออนโทโลจีชุดที่สองและได้ถูกรวมกันไว้ภายในออนโทโลจีรวมมาก่อนแล้ว การรวมคลาสของคลาส C1 และ C2 จะต้องตรวจสอบความสัมพันธ์ระหว่างคลาสของคลาส C2 และ C3 ภายในโลคัลออนโทโลจีชุดที่สองก่อน หากพบว่าคลาสของแอททริบิวต์หนึ่งภายในคลาสของคลาส C3 มีชนิดข้อมูลเป็นคลาสของคลาส C2 (เทียบได้กับความสัมพันธ์แบบแอกกรีเกชันในแบบจำลองข้อมูลเชิงวัตถุ) การรวมคลาสของคลาส C1 และ C2 จะไม่ทำตามรูปแบบความสัมพันธ์ที่ได้จากการเปรียบเทียบ เนื่องจากต้องการยึดความสัมพันธ์ที่ปรากฏในโลคัลออนโทโลจีเดิมเป็นหลัก แต่จะเลือกรวมคลาสของแอททริบิวต์ในคลาสของคลาส C1 ภายในออนโทโลจีรวมกับ

คลาสของคลาสแอททริบิวต์ภายในคลาสของคลาส C2 ทุกคู่ที่มีความสัมพันธ์กัน (s > 0) แล้วกำหนดเป็นคลาสของแอททริบิวต์ของคลาสของคลาส C2 ภายในออนโทโลจีรวมและตัดคลาสของแอททริบิวต์ที่ถูกรวมไว้ในคลาส C2 ออกจากการเป็นคลาสของแอททริบิวต์ของคลาส C1 จากนั้นจึงทำการเชื่อมความสัมพันธ์ระหว่างคลาสของคลาส C1 และ C2 ในออนโทโลจีรวมซึ่งจะอธิบายวิธีการในหัวข้อที่ 4.2.3.2 ต่อไป

เมื่อกำหนดให้คลาสของคลาส C1 จากโลคัลออนโทโลจีที่ 1 มีคลาสของแอททริบิวต์เป็น CA11, CA12, CA13, CA14 และ CA15 ในขณะที่คลาสของคลาส C2 และ C3 จากโลคัลออนโทโลจีที่ 2 มีคลาสของแอททริบิวต์เป็น CA21, CA22 และ CA31, CA32, CA33, CA34, CA35 ตามลำดับ หากคลาสของคลาส C1 มีความสัมพันธ์แบบเท่ากันกับคลาสของคลาส C3 โดยคลาสของแอททริบิวต์ CA11 กับ CA31, CA12 กับ CA32 และ CA13 กับ CA33 แต่ละคู่มีความสัมพันธ์แบบเท่ากัน (s = 1) ในขณะเดียวกันคลาสของคลาส C1 มีความสัมพันธ์แบบซูเปอร์คลาสกับคลาสของคลาส C2 โดยคลาสของแอททริบิวต์ CA14 กับ CA21 และ CA15 กับ CA22 แต่ละคู่มีความสัมพันธ์แบบเท่ากัน (s = 1) นอกจากนี้ ชนิดข้อมูลของคลาสของแอททริบิวต์ CA35 ถูกกำหนดเป็นคลาสของคลาส C2 การรวมคลาสของคลาส C1, C2 และ C3 เหล่านี้ทำได้โดยเลือกรวมคลาสของคลาส C1 กับคลาสของคลาส C3 ด้วยความสัมพันธ์แบบเท่ากันก่อน แล้วจึงพิจารณารวมคลาสของแอททริบิวต์ภายในคลาสของคลาส C1 กับ C2 ตามวิธีการด้านบน รูปที่ 4.12 แสดงตัวอย่างการรวมคลาสของคลาสที่ถูกกำหนดให้เป็นชนิดข้อมูลของบางคลาสของแอททริบิวต์ในโลคัลออนโทโลจีชุดใดชุดหนึ่ง



รูปที่ 4.12 ตัวอย่างการรวมคลาสของคลาส C1 และ C2 เมื่อคลาสของคลาส C2 ถูกกำหนดให้เป็นชนิดข้อมูลของบางคลาสของแอททริบิวต์ในคลาสของคลาส C3

<u>ตัวอย่างที่ 6</u> การรวมระหว่างคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2 (จากการเปรียบเทียบในรูปที่ 4.1 และ 4.3) มีวิธีการดังนี้

จากผลการเปรียบเทียบในตัวอย่างที่ 3 ค่าความสัมพันธ์จากการเปรียบเทียบคลาสของคลาส

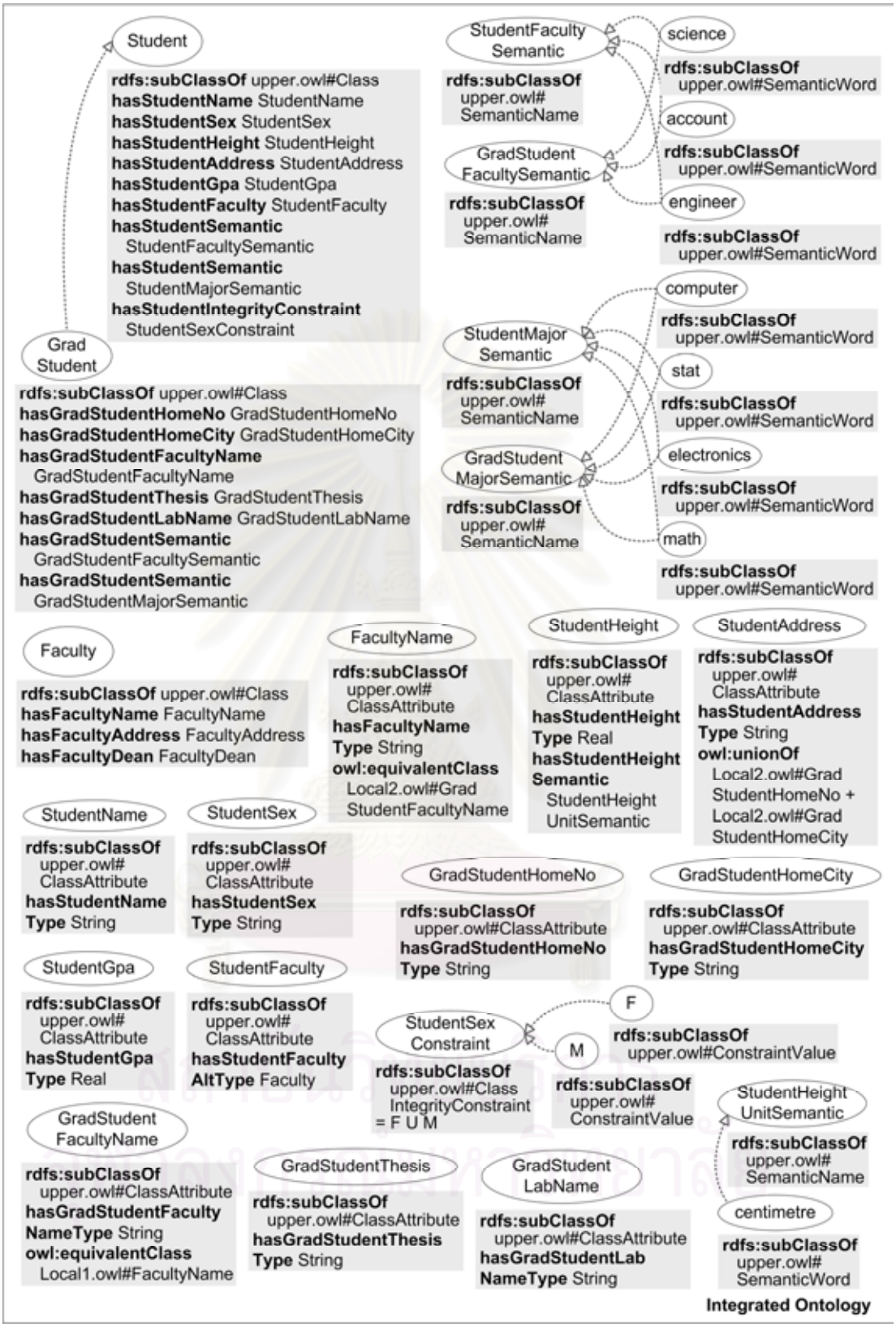
= Superclass

∴ การรวมคลาสของคลาสเป็นการรวมแบบซูเปอร์คลาส ดังรูปที่ 4.13

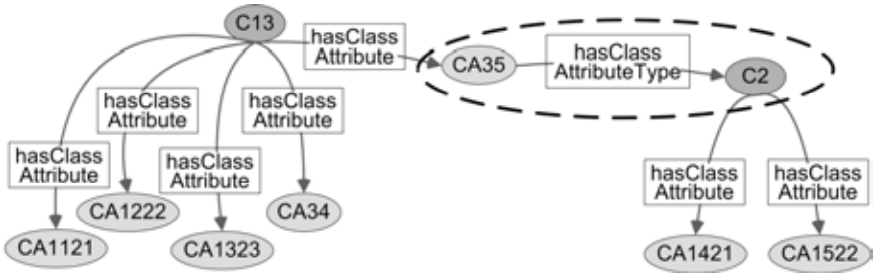
4.2.3.2 การเชื่อมความสัมพันธ์ระหว่างบางคลาสของแอททริบิวต์ไปยังคลาสของคลาสเพื่อแสดงว่าชนิดข้อมูลของคลาสของแอททริบิวต์ดังกล่าวเป็นคลาสของคลาสภายในออนโทโลจีรวม (Link aggregation relationship ในรูปที่ 4.6)

หลังจากการรวมคลาสของคลาสทุกคู่จากโลคัลออนโทโลจีทั้งสองชุดโดยการพิจารณาระดับความสัมพันธ์ และสร้างขึ้นเป็นออนโทโลจีรวมแล้ว การทำงานในส่วนนี้จะเป็นการเชื่อมความสัมพันธ์ระหว่างบางคลาสของแอททริบิวต์ไปยังคลาสของคลาสเพื่อแสดงว่าชนิดข้อมูลของคลาสของแอททริบิวต์ดังกล่าวเป็นคลาสของคลาสภายในออนโทโลจีรวม (เทียบได้กับความสัมพันธ์แบบแอกกรีเกชันในแบบจำลองข้อมูลเชิงวัตถุ) โดยจะพิจารณาว่าคลาสของแอททริบิวต์ที่ถูกกำหนดให้มีชนิดข้อมูลเป็นคลาสของคลาสในโลคัลออนโทโลจีเดิมถูกรวมเป็นคลาสของแอททริบิวต์ใดในออนโทโลจีรวม และพิจารณาว่าคลาสของคลาสที่ถูกกำหนดให้เป็นชนิดข้อมูลของคลาสของแอททริบิวต์ดังกล่าวถูกรวมเป็นคลาสของคลาดใดในออนโทโลจีรวม แล้วเชื่อมความสัมพันธ์ระหว่างคลาสของแอททริบิวต์และคลาสของคลาสคู่นั้น ด้วยการกำหนดให้ชนิดข้อมูลของคลาสของแอททริ-บิวต์เป็นคลาสของคลาส เพื่อให้การเชื่อมความสัมพันธ์ระหว่างคลาสเป็นไปอย่างถูกต้อง รูปที่ 4.14 แสดงตัวอย่างการเชื่อมความสัมพันธ์ระหว่างคลาสของแอททริบิวต์และคลาสของคลาสจากผลการรวมคลาสของคลาสในรูปที่ 4.12

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.13 ตัวอย่างการรวมคลาสของคลาสระหว่างคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2 โดยพิจารณาจากระดับความสัมพันธ์



รูปที่ 4.14 ตัวอย่างการเชื่อมความสัมพันธ์ระหว่างคลาสของแอททริบิวต์และคลาสของคลาสเพื่อแสดงว่าชนิดข้อมูลของคลาสของแอททริบิวต์ดังกล่าวเป็นคลาสของคลาสภายในออนโทโลจีรวม

4.2.3.3 การรวมคลาสของแอททริบิวต์ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันหรือเป็นคำลูกกลุ่ม/ แม่กลุ่มกัน

การระบุว่าคลาสของแอททริบิวต์ใดในโลคัลออนโทโลจีชุดที่หนึ่งเป็นคำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันกับคลาสของแอททริบิวต์อื่นๆ ในโลคัลออนโทโลจีชุดที่สองในหัวข้อที่ 3.3 แสดงให้ทราบว่าคลาสของแอททริบิวต์ทั้งสองหมายถึงสิ่งเดียวกันหรือคลาสของแอททริบิวต์คลาสใดคลาสหนึ่งประกอบด้วยคลาสของแอททริบิวต์อื่นๆ หลายคลาสตามลำดับ ดังนั้น เมื่อคลาสของแอททริบิวต์ทั้งคู่หมายถึงสิ่งเดียวกัน คลาสของแอททริบิวต์คู่ดังกล่าวควรจะได้รับการรวมเป็นคลาสของแอททริบิวต์เดียวกันในออนโทโลจีรวมด้วย ในทำนองเดียวกันเมื่อคลาสของแอททริบิวต์ใดประกอบด้วยคลาสของแอททริบิวต์อีกกลุ่มหนึ่ง คลาสของแอททริบิวต์ที่เป็นส่วนประกอบก็ควรจะถูกรวมเป็นคลาสของแอททริบิวต์หลัก เพื่อป้องกันมิให้เกิดความซ้ำซ้อนของข้อมูลที่อาจปรากฏในออนโทโลจีรวม ทั้งนี้ การรวมคลาสของแอททริบิวต์โดยการพิจารณาว่าเป็นคำที่มีความหมายเหมือนกันหรือเป็นคำลูกกลุ่ม/ แม่กลุ่มกันจะตรวจสอบจากคลาสของแอททริบิวต์ภายในออนโทโลจีรวม ซึ่งสร้างขึ้นจากคลาสของแอททริบิวต์ในโลคัลออนโทโลจีแต่ละชุดที่ได้รับการระบุว่าเป็นคำที่มีความหมายเหมือนกันหรือเป็นคำลูกกลุ่ม/ แม่กลุ่มกัน หลังจากได้ออนโทโลจีรวมจากการพิจารณาและการเชื่อมความสัมพันธ์ในขั้นตอนก่อนหน้านี้

1. การรวมคลาสของแอททริบิวต์ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกัน

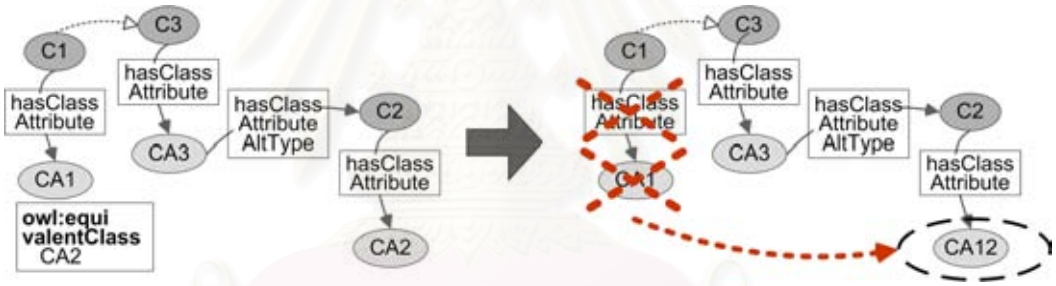
หากคลาสของแอททริบิวต์ CA1 ของคลาส C1 ภายในโลคัลออนโทโลจีชุดที่หนึ่งถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันกับคลาสของแอททริบิวต์ CA2 ของคลาส C2 ภายในโลคัลออนโทโลจีชุดที่สอง ซึ่งโดยทั่วไปค่าความสัมพันธ์จากการเปรียบเทียบคลาสของแอททริบิวต์ CA1 กับ CA2 ในหัวข้อที่ 4.1.2.1 จะเป็นความสัมพันธ์แบบการมีเส้นทางเกี่ยวข้อง (s = 1)

อย่างไรก็ตาม การรวมคลาสของแอททริบิวต์ในขั้นตอนนี้เป็นส่วนเสริมจากขั้นตอนการรวมคลาสของแอททริบิวต์ในหัวข้อที่ 4.2.2 โดยจะพิจารณาเฉพาะคลาสของแอททริบิวต์ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันคู่ที่ยังไม่ถูกรวมกันเท่านั้น ซึ่งโดยทั่วไปแล้วมักจะเกิดขึ้นจากการที่คลาสของแอททริบิวต์หนึ่งในนั้นเป็นคลาสแอททริบิวต์ของคลาสของ

คลาสที่ถูกกำหนดเป็นชนิดข้อมูลของคลาสแอททริบิวต์ใดๆ ทำให้ไม่สามารถรวมคลาสของคลาสรวมถึงคลาสของแอททริบิวต์เหล่านั้นกับคลาสของคลาสและคลาสของแอททริบิวต์อื่นภายในออนโทโลจีรวมได้โดยตรง

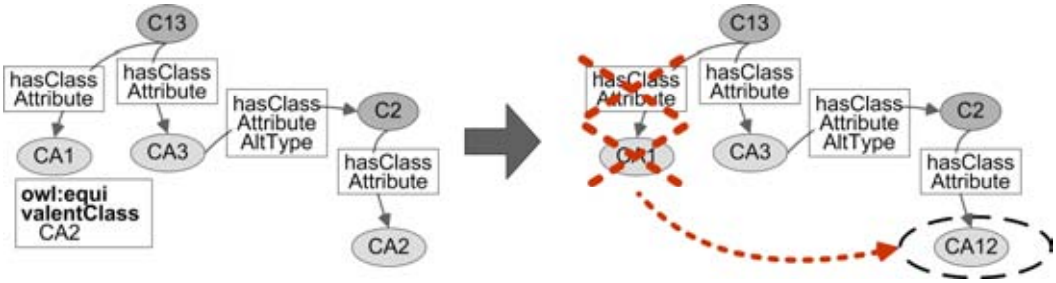
เมื่อนำโลคัลออนโทโลจีทั้งสองชุดมาสร้างเป็นออนโทโลจีรวม กำหนดให้คลาสของคลาส C3 ภายในออนโทโลจีรวมมีคลาสของแอททริบิวต์เป็น CA3 และมีชนิดข้อมูลเป็นคลาสของคลาส C2 การรวมคลาสของแอททริบิวต์ CA1 และ CA2 เข้าด้วยกันภายในออนโทโลจีรวมจะพิจารณาความสัมพันธ์ระหว่างคลาสของคลาส C1 และ C3 เพื่อตรวจสอบเส้นทางการเข้าถึง (Path) คลาสของแอททริบิวต์ CA1 และ CA2 เป็นหลัก โดยมีเงื่อนไขดังนี้

- <u>กรณีที่ 1</u> หากคลาสของคลาส C3 เป็นซูเปอร์คลาสของคลาส C1 ภายในออนโทโลจีรวม คลาสของแอททริบิวต์ CA1 จะถูกนำไปรวมกับคลาสของแอททริบิวต์ CA2 เป็นคลาสของแอททริบิวต์ CA12 ภายในคลาสของคลาส C2 แล้วตัดคลาสของแอททริบิวต์ CA1 ออกจากคลาสของคลาส C1 ตัวอย่างการรวมคลาสของแอททริบิวต์ตามเงื่อนไขนี้แสดงไว้ในรูปที่ 4.15



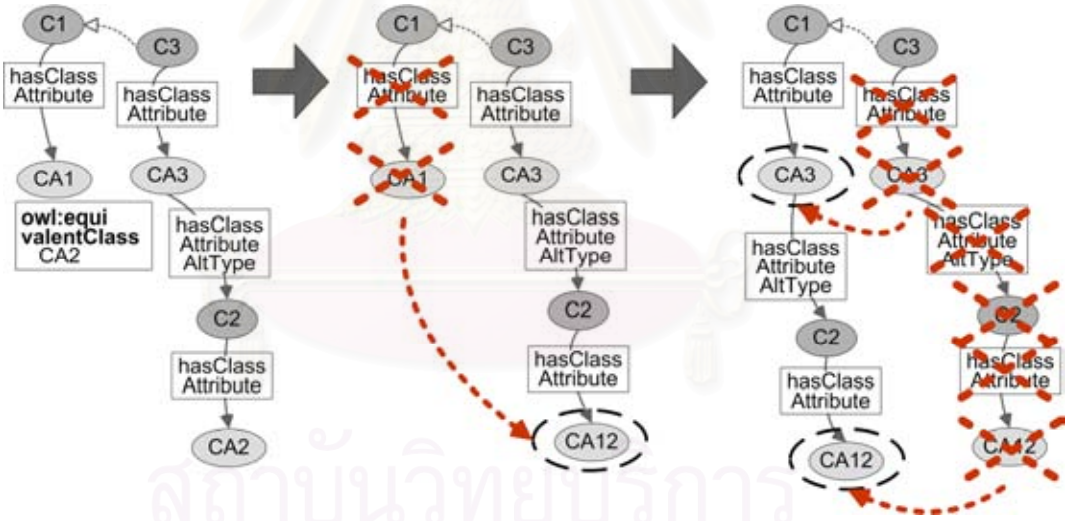
รูปที่ 4.15 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 และ CA2 ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันตามเงื่อนไขกรณีที่ 1

- <u>กรณีที่ 2</u> หากคลาสของคลาส C3 เป็นคลาสเดียวกันกับคลาสของคลาส C1 (ทั้งสองคลาสได้รับการรวมกันโดยกำหนดให้ C13 เป็นคลาสของคลาสรวมจากคลาสคู่นี้) ภายในออนโทโลจีรวม คลาสของแอททริบิวต์ CA1 จะถูกนำไปรวมกับคลาสของแอท-ทริบิวต์ CA2 เป็นคลาสของแอททริบิวต์ CA12 ภายในคลาสของคลาส C2 แล้วตัดคลาสของแอททริบิวต์ CA1 ออกจากคลาสของคลาส C13 ตัวอย่างการรวมคลาสของแอททริบิวต์ตามเงื่อนไขนี้แสดงไว้ในรูปที่ 4.16



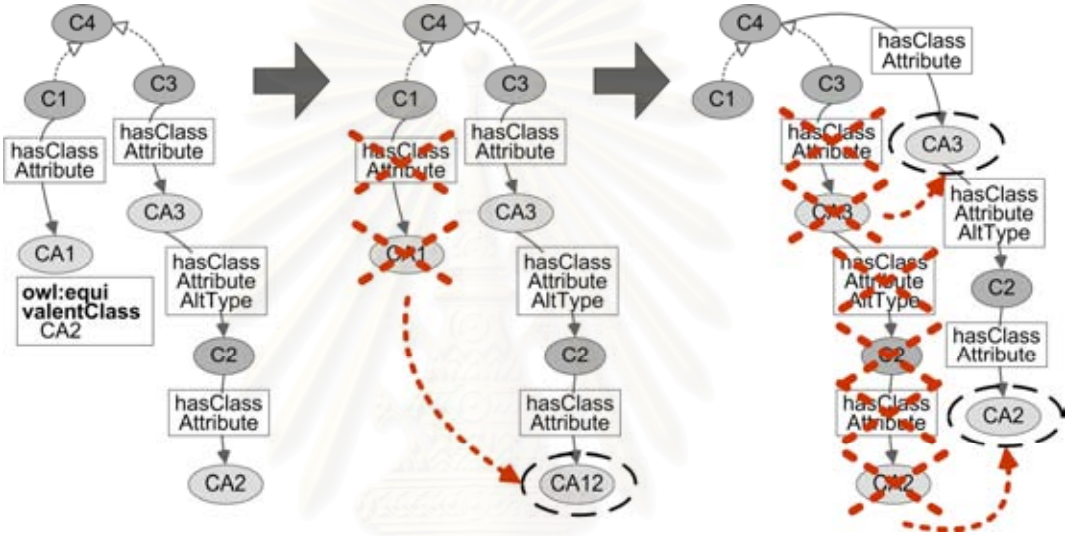
รูปที่ 4.16 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 และ CA2 ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันตามเงื่อนไขกรณีที่ 2

- กรณีที่ 3 หากคลาสของคลาส C3 เป็นสับคลาสของคลาส C1 ภายในออนโทโลจีรวมคลาสของแอททริบิวต์ CA1 จะถูกนำไปรวมกับคลาสของแอททริบิวต์ CA2 เป็นคลาสของแอททริบิวต์ CA12 ภายในคลาสของคลาส C2 แล้วตัดคลาสของแอททริบิวต์ CA1 ออกจากคลาสของคลาส C1 หลังจากนั้นจะย้ายคลาสของแอททริบิวต์ CA3 ส่วนที่มีชนิดข้อมูลย่อยเป็นคลาสของแอททริบิวต์ CA12 ไปกำหนดเป็นคลาสของแอททริบิวต์ภายในคลาสของคลาส C1 แทน ตัวอย่างการรวมคลาสของแอททริบิวต์ตามเงื่อนไขนี้แสดงไว้ในรูปที่ 4.17



รูปที่ 4.17 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 และ CA2 ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันตามเงื่อนไขกรณีที่ 3

- กรณีที่ 4 หากคลาสของคลาส C3 เป็นคลาสพี่น้องกับคลาสของคลาส C1 ภายในออนโทโลจีรวมโดยมีซูเปอร์คลาสร่วมกันคือคลาสของคลาส C4 คลาสของแอททริบิวต์ CA1 จะถูกนำไปรวมกับคลาสของแอททริบิวต์ CA2 เป็นคลาสของแอททริบิวต์ CA12 ภายในคลาสของคลาส C2 แล้วตัดคลาสของแอททริบิวต์ CA1 ออกจากคลาสของคลาส C1 หลังจากนั้นจะย้ายคลาสของแอททริบิวต์ CA3 ส่วนที่มีชนิดข้อมูลย่อยเป็นคลาสของแอททริบิวต์ CA12 ไปกำหนดเป็นคลาสของแอททริบิวต์ภายในคลาสของคลาส C4 แทน ตัวอย่างการรวมคลาสของแอททริบิวต์ตามเงื่อนไขนี้แสดงไว้ในรูปที่ 4.18



รูปที่ 4.18 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 และ CA2 ที่ถูกระบุว่าเป็นคำที่มีความหมายเหมือนกันตามเงื่อนไขกรณีที่ 4

2. การรวมคลาสของแอททริบิวต์ที่ถูกระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกัน

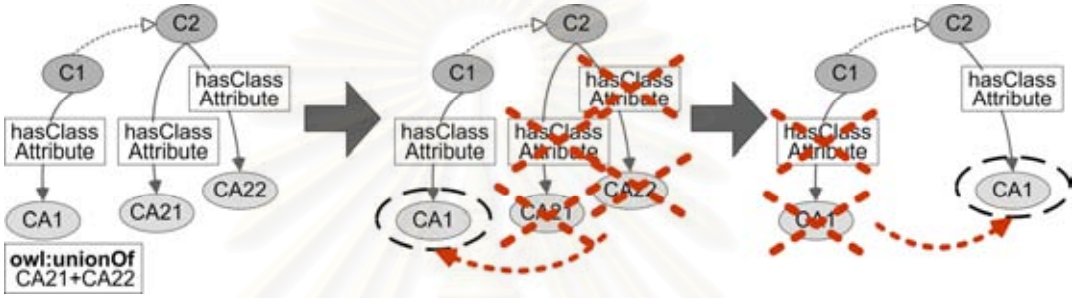
หากคลาสของแอททริบิวต์ CA1 ของคลาส C1 ภายในโลคัลออนโทโลจีชุดที่หนึ่งถูกระบุว่าเป็นคำแม่กลุ่มของคลาสของแอททริบิวต์ CA21 และ CA22 ของคลาส C2 ภายในโลคัลออนโทโลจีชุดที่สอง (หรือในอีกแง่หนึ่งคลาสของแอททริบิวต์ CA21 และ CA22 ของคลาส C2 ภายในโลคัลออนโทโลจีชุดที่สองถูกระบุว่าเป็นคำลูกกลุ่มของคลาสของแอททริบิวต์ CA1 ของคลาส C1 ภายในโลคัลออนโทโลจีชุดที่หนึ่ง) ซึ่งโดยทั่วไปค่าความสัมพันธ์จากการเปรียบเทียบคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ในหัวข้อที่ 4.1.2.1 จะเป็นความสัมพันธ์แบบการเป็นส่วนประกอบกัน (s = 1)

เมื่อนำโลคัลออนโทโลจีทั้งสองชุดมาสร้างเป็นออนโทโลจีรวม การรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ภายในออนโทโลจีรวมจะพิจารณาจากเส้นทางการเข้าถึง (Path) คลาสของแอททริบิวต์ CA1 และ CA2 โดยมีเงื่อนไขดังนี้

เมื่อนำโลคัลออนโทโลจีทั้งสองชุดมาสร้างเป็นออนโทโลจีรวม การรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 เข้าด้วยกันภายในออนโทโลจีรวมจะพิจารณา

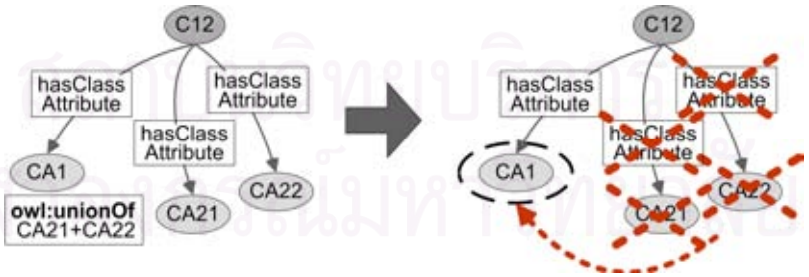
ความสัมพันธ์ระหว่างคลาสของคลาส C1 และ C2 เพื่อตรวจสอบเส้นทางการเข้าถึง (Path) คลาสของแอททริบิวต์ CA1, CA21 และ CA22 เป็นหลัก โดยมีเงื่อนไขดังนี้

- <u>กรณีที่ 1</u> หากคลาสของคลาส C2 เป็นซูเปอร์คลาสของคลาส C1 ภายในออนโทโลจีรวม คลาสของแอททริบิวต์ CA21 และ CA22 จะถูกนำไปรวมกับคลาสของแอททริบิวต์ CA1 ในลักษณะของการเป็นส่วนประกอบ แล้วตัดคลาสของแอททริบิวต์ CA21 และ CA22 ออกจากคลาสของคลาส C2 หลังจากนั้นจะย้ายคลาสของแอททริบิวต์ CA1 ไปกำหนดเป็นคลาสของแอททริบิวต์ภายในคลาสของคลาส C2 แทน ตัวอย่างการรวมคลาสของแอททริบิวต์ตามเงื่อนไขนี้แสดงไว้ในรูปที่ 4.19



รูปที่ 4.19 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ที่ถูกระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกันตามเงื่อนไขกรณีที่ 1

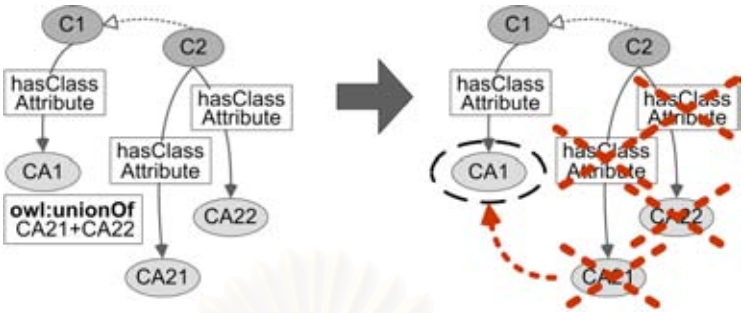
- <u>กรณีที่ 2</u> หากคลาสของคลาส C2 เป็นคลาสเดียวกันกับคลาสของคลาส C1 (ทั้งสองคลาสได้รับการรวมกันโดยกำหนดให้ C12 เป็นคลาสของคลาสรวมจากคลาสคู่นี้) ภายในออนโทโลจีรวม คลาสของแอททริบิวต์ CA21 และ CA22 จะถูกนำไปรวมกับคลาสของแอททริบิวต์ CA1 ในลักษณะของการเป็นส่วนประกอบ แล้วตัดคลาสของแอททริบิวต์ CA21 และ CA22 ออกจากคลาสของคลาส C12 ตัวอย่างการรวมคลาสของแอททริบิวต์ตามเงื่อนไขนี้แสดงไว้ในรูปที่ 4.20



รูปที่ 4.20 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ที่ถูกระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกันตามเงื่อนไขกรณีที่ 2

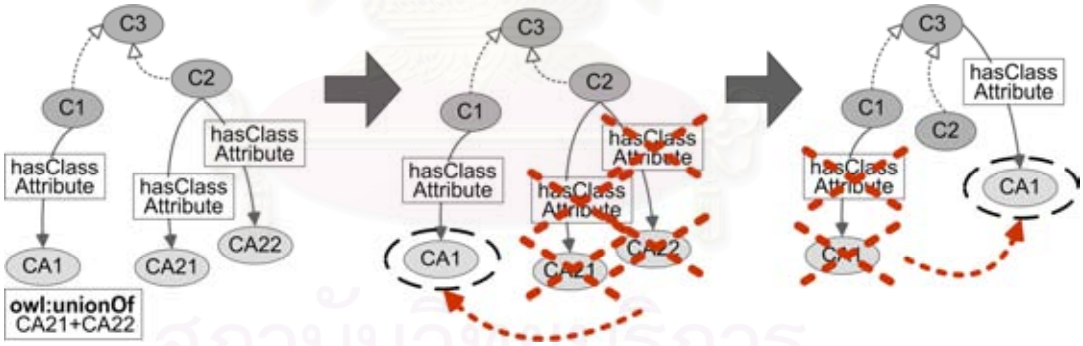
- <u>กรณีที่ 3</u> หากคลาสของคลาส C2 เป็นสับคลาสของคลาส C1 ภายในออนโทโลจีรวม คลาสของแอททริบิวต์ CA21 และ CA22 จะถูกนำไปรวมกับคลาสของแอททริบิวต์ CA1 ในลักษณะของการเป็นส่วนประกอบ แล้วตัดคลาสของแอททริบิวต์ CA21 และ

CA22 ออกจากคลาสของคลาส C2 ตัวอย่างการรวมคลาสของแอททริบิวต์ตามเงื่อนไขนี้แสดงไว้ในรูปที่ 4.21



รูปที่ 4.21 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ที่ถูกระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกันตามเงื่อนไขกรณีที่ 3

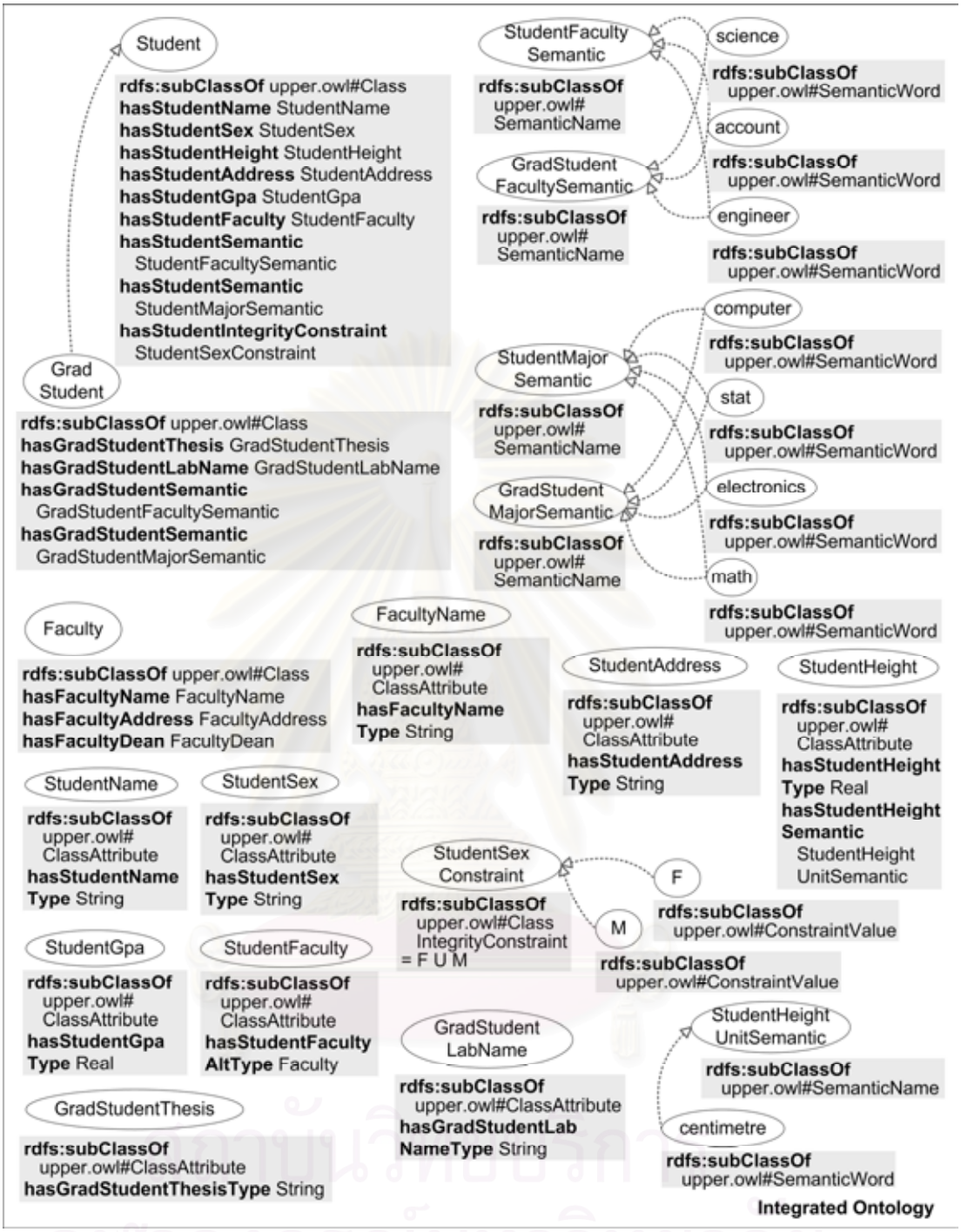
- <u>กรณีที่ 4</u> หากคลาสของคลาส C2 เป็นคลาสพี่น้องกับคลาสของคลาส C1 ภายในออนโทโลจีรวมโดยมีซูเปอร์คลาสร่วมกันคือคลาสของคลาส C3 คลาสของแอททริบิวต์ CA21 และ CA22 จะถูกนำไปรวมกับคลาสของแอททริบิวต์ CA1 ในลักษณะของการเป็นส่วนประกอบ แล้วตัดคลาสของแอททริบิวต์ CA21 และ CA22 ออกจากคลาสของคลาส C2 หลังจากนั้นจะย้ายคลาสของแอททริบิวต์ CA1 ไปกำหนดเป็นคลาสของแอททริบิวต์ภายในคลาสของคลาส C3 แทน ตัวอย่างการรวมคลาสของแอททริบิวต์ตามเงื่อนไขนี้แสดงไว้ในรูปที่ 4.22



รูปที่ 4.22 ตัวอย่างการรวมคลาสของแอททริบิวต์ CA1 กับ CA21 และ CA22 ที่ถูกระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกันตามเงื่อนไขกรณีที่ 4

<u>ตัวอย่างที่ 7</u> การรวมคลาสของคลาสระหว่างคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2 เมื่อพิจารณาเพิ่มเติมจากผลการรวมในตัวอย่างที่ 6 ตามหัวข้อที่ 4.2.3.2 และ 4.2.3.3

∴ ผลการรวมคลาสของคลาสระหว่างคลาสของคลาส Student และ GradStudent หลังจากพิจารณาตามขั้นตอนการรวมคลาสของคลาสครบทุกขั้นตอน แสดงได้ดังรูปที่ 4.23



รูปที่ 4.23 ตัวอย่างการรวมคลาสของคลาสระหว่างคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2 หลังจากพิจารณาตามขั้นตอนการรวมคลาสของคลาสครบทุกขั้นตอน

หลังจากทำการรวมโลคัลออนโทโลจีส่วนที่เป็นคลาสในแบบจำลองข้อมูลเสร็จสิ้นแล้ว ผลที่ได้จากขั้นตอนนี้คือรูปแบบของออนโทโลจีรวมที่สามารถนำไปสร้างเป็นออนโทโลจีรวมเพื่อนำไปใช้งานกับเฟรมเวิร์คที่จะนำเสนอในบทที่ 5 ต่อไป

4.3 ตัวอย่างการเปรียบเทียบและการรวมโลคัลออนโทโลจี

ตัวอย่างการเปรียบเทียบและการรวมโลคัลออนโทโลจีในหัวข้อนี้เป็นการแสดงวิธีการทำงานของขั้นตอนวิธีการเปรียบเทียบและการรวมโลคัลออนโทโลจีที่ได้นำเสนอไปแล้วในหัวข้อที่ 4.1 และ 4.2 โดยแสดงการคิดคำนวณ และการวิเคราะห์ค่าความสัมพันธ์ผ่านการรวมโลคัลออนโทโลจีสองชุดที่ได้สร้างขึ้นและแสดงไว้ในรูปที่ 3.5 และ 3.6 ร่วมกับแมปปิงแสดงการระบุความสัมพันธ์ของคำศัพท์ระหว่างโลคัลออนโทโลจีทั้งสองชุดในรูปที่ 3.8 ในบทที่ 3 นอกเหนือจากบางส่วนที่ได้แสดงไปแล้วในตัวอย่างที่ 1 ถึง 6 มีขั้นตอนดังนี้

4.3.1 ตัวอย่างการเปรียบเทียบโลคัลออนโทโลจี

4.3.1.1 ตัวอย่างการเปรียบเทียบความหมาย

- ความหมายของคลาสของแอททริบิวต์ StudentHeight ในโลคัลออนโทโลจีที่ 1 กับคลาสของแอททริบิวต์ PersonHeight ในโลคัลออนโทโลจีที่ 2

  T1 = StudentHeight, T2 = PersonHeight

  α = 1 (StudentHeightUnitSemantic)

  β = 1 (PersonHeightUnitSemantic)

  r_1 = 0.5 (ตารางที่ 4.1 กรณีที่ 4)

  R = [0.5] $\therefore$ Ms = 0.5

  $\therefore$ ระดับความเหมือน = Tight-intersection, ความสัมพันธ์ = Sibling และ s = 0.1 (ตารางที่ 4.2 กรณีที่ 5)

- ความหมายของคลาสของแอททริบิวต์ StudentHeight ในโลคัลออนโทโลจีที่ 1 กับคลาสของแอททริบิวต์ GradStudentHeight ในโลคัลออนโทโลจีที่ 2

  T1 = StudentHeight, T2 = GradStudentHeight

  α = 1 (StudentHeightUnitSemantic)

  β = 1 (GradStudentHeightUnitSemantic)

  r_1 = 0.5 (ตารางที่ 4.1 กรณีที่ 4)

  R = [0.5] $\therefore$ Ms = 0.5

  $\therefore$ ระดับความเหมือน = Tight-intersection, ความสัมพันธ์ = Sibling และ s = 0.1 (ตารางที่ 4.2 กรณีที่ 5)

4.3.1.2 ตัวอย่างการเปรียบเทียบคลาสของแอททริบิวต์

- ชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส Person ในโลคัลออนโทโลจีที่ 2

  C1 = Student, C2 = Person

θ_{C1} = 6 (StudentName, StudentSex, StudentHeight, StudentAddress, StudentGpa, StudentFaculty)

θ_{C2} = 5 (PersonName, PersonGender, PersonHeight, PersonHomeNo, PersonHomeCity)

α = 6 – 1 + 2 = 7, β = 5

เปรียบเทียบระหว่าง StudentName และ PersonName

ความสัมพันธ์ = Equivalence, s_1 = 1 (ตารางที่ 4.3 กรณีที่ 3)

เปรียบเทียบระหว่าง StudentSex และ PersonGender

ความสัมพันธ์ = Synonym, s_2 = 1 (ตารางที่ 4.3 กรณีที่ 4)

เปรียบเทียบระหว่าง StudentHeight และ PersonHeight

ความสัมพันธ์ = Sibling, s_3 = 0.1

(ตารางที่ 4.3 กรณีที่ 1 และ ตารางที่ 4.2 กรณีที่ 5)

เปรียบเทียบระหว่าง StudentAddress และ PersonHomeNo

ความสัมพันธ์ = Compound, s_4 = 1 (ตารางที่ 4.3 กรณีที่ 6)

เปรียบเทียบระหว่าง StudentAddress และ PersonHomeCity

ความสัมพันธ์ = Compound, s_5 = 1 (ตารางที่ 4.3 กรณีที่ 6)

$R_{all\text{-}ClassAttribute}$ = [1, 1, 0.1, 1, 1] $\therefore$ $Ms_{all\text{-}ClassAttribute}$ = 1 + 1 + 0.1 + 1 + 1 = 4.1

$\therefore$ ระดับความเหมือน = Inclusion และความสัมพันธ์ = Subclass

(ตารางที่ 4.4 กรณีที่ 4)

- ชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส Faculty ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส Person ในโลคัลออนโทโลจีที่ 2

C1 = Faculty, C2 = Person

θ_{C1} = 3 (FacultyName, FacultyAddress, FacultyDean)

θ_{C2} = 5 (PersonName, PersonGender, PersonHeight, PersonHomeNo, PersonHomeCity)

α = 3, β = 5

เปรียบเทียบระหว่าง FacultyName และ PersonName

ความสัมพันธ์ = Equivalence, s_1 = 1 (ตารางที่ 4.3 กรณีที่ 3)

$R_{all\text{-}ClassAttribute}$ = [1] $\therefore$ $Ms_{all\text{-}ClassAttribute}$ = 1

$\therefore$ ระดับความเหมือน = Loose-intersection และความสัมพันธ์ = Disjoint

(ตารางที่ 4.4 กรณีที่ 6)

- ชุดของคลาสของแอททริบิวต์ภายในคลาสของคลาส Faculty ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2

  C1 = Faculty, C2 = GradStudent

  θ_{C1} = 3 (FacultyName, FacultyAddress, FacultyDean)

  θ_{C2} = 9 (GradStudentName, GradStudentGender, GradStudentHeight, GradStudentHomeNo, GradStudentHomeCity, GradStudentFacultyName, GradStudentGpa, GradStudentThesis, GradStudentLabName)

  α = 3, β = 9

  เปรียบเทียบระหว่าง FacultyName และ GradStudentName

  ความสัมพันธ์ = Equivalence, s_1 = 1 (ตารางที่ 4.3 กรณีที่ 3)

  เปรียบเทียบระหว่าง FacultyName และ GradStudentFacultyName

  ความสัมพันธ์ = Synonym, s_2 = 1 (ตารางที่ 4.3 กรณีที่ 4)

  $R_{all\text{-}ClassAttribute}$ = [1, 1] $\therefore Ms_{all\text{-}ClassAttribute}$ = 1 + 1 = 2

  $\therefore$ ระดับความเหมือน = Tight-intersection และความสัมพันธ์ = Sibling (ตารางที่ 4.4 กรณีที่ 5)

4.3.1.3 ตัวอย่างการเปรียบเทียบคลาสของคลาส

- คลาสของคลาส Student ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส Person ในโลคัลออนโทโลจีที่ 2

  ความสัมพันธ์จากการเปรียบเทียบความหมาย = ไม่มีการระบุความหมาย

  ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ = Subclass

  $\therefore$ ความสัมพันธ์ = Subclass (ตารางที่ 4.5 กรณีที่ 28)

- คลาสของคลาส Faculty ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส Person ในโลคัลออนโทโลจีที่ 2

  ความสัมพันธ์จากการเปรียบเทียบความหมาย = ไม่มีการระบุความหมาย

  ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ = Disjoint

  $\therefore$ ความสัมพันธ์ = Disjoint (ตารางที่ 4.5 กรณีที่ 30)

- คลาสของคลาส Faculty ในโลคัลออนโทโลจีที่ 1 และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่ 2

  ความสัมพันธ์จากการเปรียบเทียบความหมาย = ไม่มีการระบุความหมาย

  ความสัมพันธ์จากการเปรียบเทียบชุดของคลาสของแอททริบิวต์ = Sibling

  $\therefore$ ความสัมพันธ์ = Sibling (ตารางที่ 4.5 กรณีที่ 29)

จากผลการเปรียบเทียบคลาสของคลาสสามารถแสดงความสัมพันธ์ระหว่างคลาสของคลาสภายในโลคัลออนโทโลจีที่ 1 และโลคัลออนโทโลจีที่ 2 ได้ดังตารางที่ 4.6

ตารางที่ 4.6 ผลการเปรียบเทียบคลาสของคลาสภายในโลคัลออนโทโลจีที่ 1 และ 2

Local Ontology 2 / Local Ontology 1	Person	GradStudent
Student	Subclass	Superclass
Faculty	Disjoint	Sibling

4.3.2 ตัวอย่างการรวมโลคัลออนโทโลจี

จากผลจากการเปรียบเทียบระหว่างโลคัลออนโทโลจีที่ 1 และโลคัลออนโทโลจีที่ 2 ในหัวข้อที่ 4.3.1 สามารถนำมาสร้างขึ้นเป็นออนโทโลจีรวมโดยยึดตามค่าความสัมพันธ์ระหว่างคลาสของคลาสในตางรางที่ 4.6 แล้วรวมคลาสของความหมายและคลาสของคลาสแอททริบิวต์ตามผลการเปรียบเทียบส่วนประกอบต่าง ออนโทโลจีรวมที่ได้ผลการเปรียบเทียบแสดงได้ดังรูปที่ 4.24

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 4.24 ตัวอย่างออนโทโลจีรวมที่ได้จากการรวมโลคัลออนโทโลจีที่ 1 ในรูปที่ 3.5 และ โลคัลออนโทโลจีที่ 2 ในรูปที่ 3.6

บทที่ 5
เฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจี

ในบทนี้จะกล่าวถึงการออกแบบ องค์ประกอบและต้นแบบการทำงานของเฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีที่พัฒนาขึ้นตามแนวคิดแบบรวมหลายออนโทโลจี เพื่อรวมแบบจำลองข้อมูลเชิงวัตถุสองชุดซึ่งนำมาจากฐานข้อมูลเชิงวัตถุสองแหล่งที่เก็บข้อมูลเรื่องเดียวกันหรือมีความเกี่ยวข้องกัน เนื้อหาภายในบทจะอธิบายถึงแนวคิดในการออกแบบเฟรมเวิร์ค (หัวข้อที่ 5.1) องค์ประกอบของเฟรมเวิร์ค และต้นแบบการทำงานของเฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจี (หัวข้อที่ 5.2) พร้อมทั้งแผ่นแบบการสอบถามข้อมูลผ่านการทำงานของเฟรมเวิร์คซึ่งสร้างขึ้นตามออนโทโลจีรวมที่ได้ เพื่อแสดงการนำแนวคิดแบบรวมหลายออนโทโลจีไปใช้งานจริงในการสอบถามข้อมูลจากฐานข้อมูลเชิงวัตถุ (หัวข้อที่ 5.3) ตามลำดับ

ทั้งนี้ เฟรมเวิร์คทั้งหมดที่นำเสนอในวิทยานิพนธ์ฉบับนี้ได้รับการออกแบบให้สอดคล้องกับการใช้งานออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุตามที่ได้กล่าวไปแล้วในบทที่ 3 และการนำขั้นตอนวิธีการรวมออนโทโลจีที่ประยุกต์จากงานวิจัย [20] ดังรายละเอียดในบทที่ 4 มาใช้งาน

5.1 แนวคิดในการออกแบบเฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจี

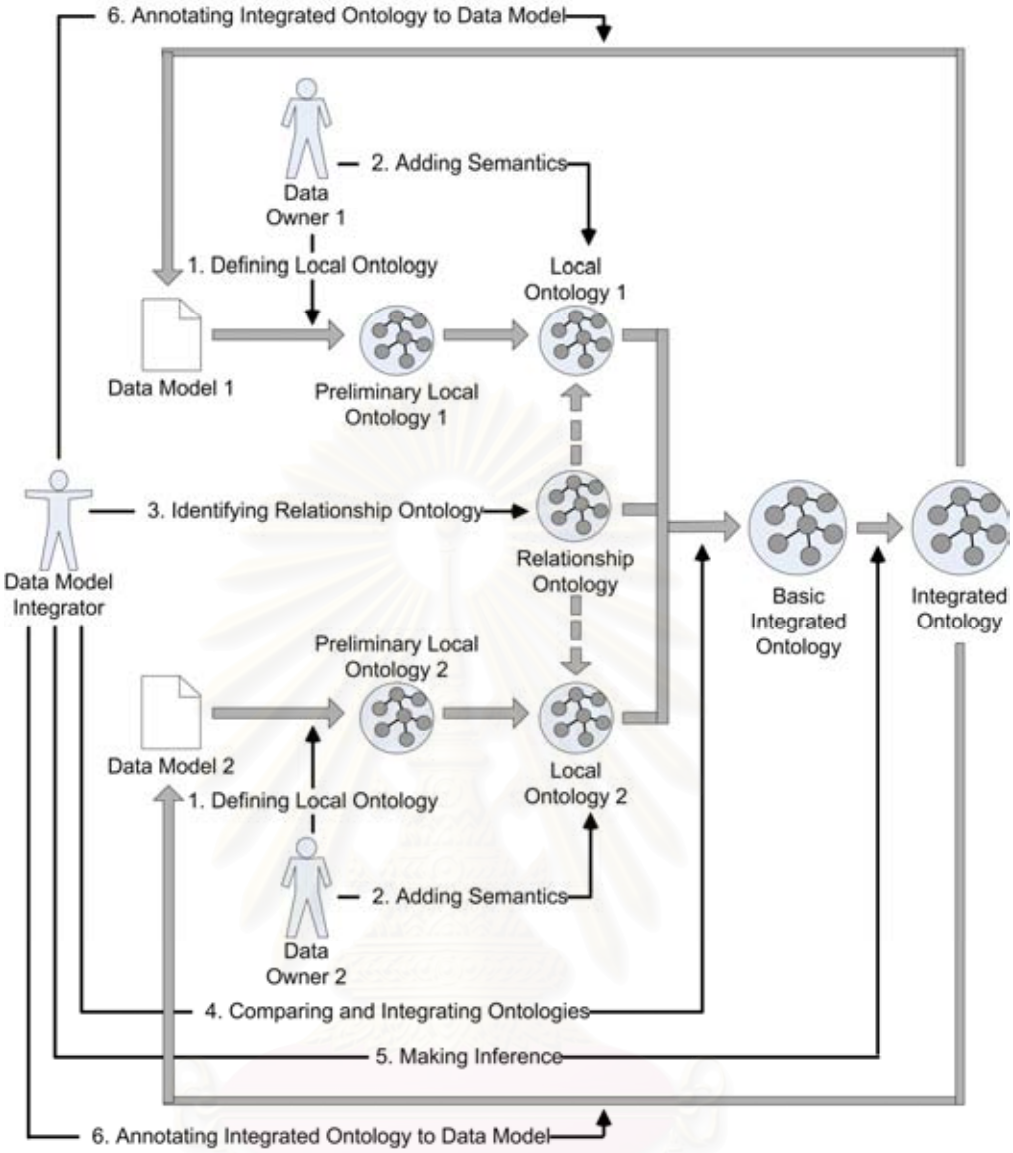
เฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีที่นำเสนอในวิทยานิพนธ์ฉบับนี้ออกแบบขึ้นตามแนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูลแบบรวมหลายออนโทโลจี โดยลักษณะสำคัญของแนวคิดแบบรวมหลายออนโทโลจีคือ การสร้างโลคัลออนโทโลจีหนึ่งชุดมาใช้อธิบายแบบจำลองข้อมูลแต่ละชุดแยกกัน แล้วนำโลคัลออนโทโลจีที่ได้มารวมกันเป็นออนโทโลจีรวม จากนั้นนำออนโทโลจีรวมที่ได้ไปกำกับลงบนแบบจำลองข้อมูลเชิงวัตถุทุกชุดเพื่อทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับแบบจำลองข้อมูลแต่ละชุดร่วมกัน

ผู้วิจัยมีแนวคิดว่า การดำเนินการเพื่อการรวมแบบจำลองข้อมูลเชิงวัตถุใดๆ ด้วยออนโทโลจีตามแนวคิดแบบรวมหลายออนโทโลจีนั้น เฟรมเวิร์คควรจะประกอบด้วยขั้นตอนการทำงาน และจัดเตรียมรายละเอียดดังต่อไปนี้ไว้ในขั้นตอนการรวมแบบจำลองข้อมูล

1. การกำหนดขั้นตอนวิธีในการรวมแบบจำลองข้อมูลเชิงวัตถุที่จะใช้ในการรวม ซึ่งในวิทยานิพนธ์ฉบับนี้เลือกใช้ขั้นตอนวิธีในงานวิจัย [20]
2. การกำหนดอัปเปอร์ออนโทโลจีสำหรับการอธิบายความหมายของแบบจำลองข้อมูล ซึ่งความหมายที่อธิบายจะต้องสอดคล้องกับขั้นตอนวิธีที่เลือกใช้ในข้อที่ 1 โดยวิทยานิพนธ์นี้นำเอกสารเมตาโมเดลของนิยามของออนโทโลจี [21] มาใช้ประกอบเป็นแนวทางการกำหนดอัปเปอร์ออนโทโลจี และได้อัปเปอร์ออนโทโลจีตามที่ได้อธิบายในหัวข้อที่ 3.1

3. จากข้อที่ 1 และ 2 ทำให้สามารถสร้างขั้นตอนวิธีที่เหมาะสมในการรวมออนโทโลจีได้ ซึ่งขั้นตอนวิธีที่ใช้ในวิทยานิพนธ์นี้ได้แสดงไว้ในบทที่ 4
4. เฟรมเวิร์คจะรับเข้าแบบจำลองข้อมูลเชิงวัตถุสองชุดที่ต้องการรวมจากเจ้าของแบบจำลองข้อมูล
5. จากอัปเปอร์ออนโทโลจีในข้อที่ 2 และแบบจำลองข้อมูลเชิงวัตถุที่ได้รับในข้อที่ 4 ทำให้สามารถสร้างโลคัลออนโทโลจีเบื้องต้น (Preliminary Local Ontology) สองชุด (ตามวิธีการในหัวข้อที่ 3.2.1)
6. เฟรมเวิร์คจะเปิดให้เจ้าของแบบจำลองข้อมูลกำหนดความหมายเพิ่มเติมให้กับโลคัลออนโทโลจีเบื้องต้น ทำให้สามารถสร้างโลคัลออนโทโลจีสองชุด (ตามวิธีการในหัวข้อที่ 3.2.2)
7. หลังจากนั้นจะเปิดให้ผู้ทำการรวมแบบจำลองข้อมูลระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดที่ได้รับในข้อที่ 6 ในรูปของออนโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจี (ตามวิธีการในหัวข้อที่ 3.3)
8. จากขั้นตอนวิธีในการรวมออนโทโลจีในข้อที่ 3 และโลคัลออนโลจีทั้งสามชุดในข้อที่ 6 และ 7 ทำให้สามารถรวมโลคัลออนโทโลจีสองชุดเข้าด้วยกัน และสร้างเป็นออนโทโลจีรวม
9. จากออนโทโลจีรวมที่ได้ในข้อที่ 8 เฟรมเวิร์คจะเปิดให้ผู้รวมออนโทโลจีกำหนดกฎเพื่อนำไปใช้อนุมานกับออนโทโลจีรวม และกำกับออนโทโลจีรวมที่ได้ลงบนแบบจำลองข้อมูลเชิงวัตถุแต่ละชุดที่ได้รับเข้าในข้อที่ 4

จากแนวคิดข้างต้นนี้ร่วมกับสมมติฐานของผู้วิจัยที่ว่า หากนำแบบจำลองข้อมูลที่ได้รับการอธิบายและแสดงแทนความหมายในรูปแบบที่เหมาะสมและเพียงพอ มารวมเข้าด้วยกันโดยใช้ขั้นตอนวิธีที่เหมาะสมจะทำให้การรวมแบบจำลองข้อมูลเป็นไปอย่างถูกต้องและครบถ้วน ผู้วิจัยจึงได้ออกแบบเฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีตามแนวคิดแบบรวมหลายออนโทโลจีขึ้นเพื่อให้สอดคล้องกับแนวคิดและสมมติฐานดังกล่าว โดยเฟรมเวิร์คที่ออกแบบและนำเสนอในบทนี้จะครอบคลุมเฉพาะขั้นตอนในข้อที่ 4 ถึง 9 เนื่องจากได้อธิบายถึงการกำหนดอัปเปอร์ออนโทโลจี การเลือกและการสร้างขั้นตอนวิธีในการรวมออนโทโลจี (ขั้นตอนในข้อที่ 1 ถึง 3) ไว้โดยละเอียดแล้วในบทที่ 3 และ 4 นอกจากนั้น จะเห็นได้ว่าการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีโดยใช้แนวคิดแบบรวมหลายออนโทโลจีนี้มิได้ยึดติดกับวิธีการกำหนดอัปเปอร์ออนโทโลจี หรือขั้นตอนวิธีในการรวมออนโทโลจีแต่อย่างใด ดังนั้นจึงสามารถนำแนวคิดนี้ไปพัฒนาต่อเพื่อเพิ่มความซับซ้อนของขั้นตอนวิธีในการรวมหรือขยายความสามารถให้นำไปใช้กับการรวมแบบจำลองข้อมูลชนิดอื่นๆ ได้ในอนาคต การทำงานของเฟรมเวิร์คนี้ประกอบด้วย 6 ขั้นตอนตามที่แสดงไว้ในรูปที่ 5.1 ดังนี้



รูปที่ 5.1 เฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจี ตามแนวคิดแบบรวมหลายออนโทโลจี

1. การกำหนดโลคัลออนโทโลจี (Defining Local Ontology) เป็นขั้นตอนที่เจ้าของแบบจำลองข้อมูลหรือผู้ที่เข้าใจแบบจำลองข้อมูลเป็นอย่างดีสร้างโลคัลออนโทโลจีเพื่ออธิบายแบบจำลองข้อมูลเชิงวัตถุของตนตามโครงสร้างของอัปเปอร์ออนโทโลจีที่ได้ประกาศไว้ และเรียกออนโทโลจีที่ได้ว่า “โลคัลออนโทโลจีเบื้องต้น (Preliminary Local Ontology)”
2. การเพิ่มความหมายให้กับโลคัลออนโทโลจี (Adding Semantics) เป็นขั้นตอนที่เจ้าของแบบจำลองข้อมูลหรือผู้ที่เข้าใจแบบจำลองข้อมูลเป็นอย่างดีกำหนดความหมายเพิ่มเติมให้กับโลคัลออนโทโลจีแต่ละชุด และโลคัลออนโทโลจีที่ได้จากขั้นตอนนี้จะนำไปใช้ในขั้นตอนการเปรียบเทียบและการรวมออนโทโลจี

3. การระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจี (Identifying Relationship Ontology) เป็นขั้นตอนที่ผู้ทำการรวมแบบจำลองข้อมูลเชิงวัตถุระบุสร้างออนโทโลจีขึ้นหนึ่งชุดเพื่อระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุด ซึ่งออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีที่ได้จากขั้นตอนนี้จะนำไปใช้ในขั้นตอนการเปรียบเทียบและการรวมออนโทโลจีเช่นกัน
4. การเปรียบเทียบและการรวมออนโทโลจี (Comparing and Integrating Ontologies) เป็นขั้นตอนการทำงานแบบกึ่งอัตโนมัติโดยการรับเข้าโลคัลออนโทโลจีสองชุด และนำเปรียบเทียบกันแล้วสร้างเป็นออนโทโลจีรวมพื้นฐาน (Basic Integrated Ontology)
5. การอนุมานออนโทโลจี (Making Inference) เป็นขั้นตอนการอนุมานโดยผู้ทำการรวมแบบจำลองข้อมูลจะเพิ่มกฎตามความหมายและข้อกำหนดบูรณภาพของแบบจำลองข้อมูลแต่ละชุด เพื่อให้ได้ข้อเท็จจริงใหม่เกี่ยวกับข้อมูลภายในแบบจำลองข้อมูลที่นำมารวม
6. การกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูล (Annotating Integrated Ontology onto Data Model) เป็นขั้นตอนที่ผู้ทำการรวมแบบจำลองข้อมูลนำออนโทโลจีรวมที่ได้ไปกำกับลงในแบบจำลองข้อมูลแต่ละชุด

ทั้งนี้ เมื่อนำเฟรมเวิร์คนี้ไปพัฒนาขึ้นเพื่อใช้งานจริง ขั้นตอนการทำงานทั้งห้าขั้นตอนนี้จะถูกแบ่งออกเป็นสามส่วนย่อยตามองค์ประกอบสำคัญในการรวมแบบจำลองข้อมูล ได้แก่ ส่วนการสร้างโลคัลออนโทโลจี (ขั้นตอนที่ 1 ถึง 3) ส่วนการสร้างออนโทโลจีรวม (ขั้นตอนที่ 4 และ 5) และส่วนการสร้างแมป-ปิงระหว่างออนโทโลจีรวมกับแบบจำลองข้อมูลเชิงวัตถุ (ขั้นตอนที่ 6) ซึ่งจะนำเสนอรายละเอียดในหัวข้อที่ 5.2

5.2 เฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีตามแนวคิดแบบรวมหลายออนโทโลจี

องค์ประกอบของเฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีตามแนวคิดแบบรวมหลายออนโทโลจีประกอบด้วย 3 ส่วนซึ่งทำงานต่อเนื่องกันตามลำดับ ได้แก่ ส่วนการสร้างโลคัลออนโทโลจี ส่วนการสร้างออนโทโลจีรวม และส่วนการสร้างแมปปิงระหว่างออนโทโลจีรวมกับแบบจำลองข้อมูลเชิงวัตถุ ต้นแบบการทำงานของเฟรมเวิร์คแต่ละส่วนได้รับการพัฒนาแยกกัน และนำผลลัพธ์ที่ได้จากส่วนก่อนหน้ามาใช้พิจารณาในส่วนถัดไป โดยแต่ละส่วนมีการทำงานดังต่อไปนี้

5.2.1 ส่วนการสร้างโลคัลออนโทโลจี

เฟรมเวิร์คส่วนการสร้างโลคัลออนโทโลจีประกอบด้วยขั้นตอนการสร้างออนโทโลจีเพื่ออธิบายแบบจำลองข้อมูลเชิงวัตถุที่ต้องการรวมแต่ละชุด และการเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุแต่ละชุดลงในออนโทโลจีที่สร้างขึ้น (ขั้นตอนที่ 1 ถึง 3 ในรูปที่ 5.1) การสร้างออนโทโลจีเพื่ออธิบายแบบจำลองข้อมูลเชิงวัตถุแต่ละชุดแยกกันนี้มีลักษณะเช่นเดียวกับแนวคิดแบบหลายออนโทโลจีและ

แนวคิดแบบผสม ซึ่งมีข้อดีคือการเปิดให้ผู้เป็นเจ้าของข้อมูลทำหน้าที่อธิบายแบบจำลองข้อมูลของตนเอง ส่งผลให้ออนโทโลจีที่ได้มีความถูกต้องตรงตามความเป็นจริง อีกทั้งเจ้าของข้อมูลสามารถเลือกได้ว่าต้องการให้ข้อมูลฟิลด์ใดบ้างปรากฏในแบบจำลองข้อมูลรวม

ในการสร้างออนโทโลจีเพื่ออธิบายความหมายของแบบจำลองข้อมูลเชิงวัตถุและการเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุลงในออนโทโลจีที่สร้างขึ้นนั้น วิทยานิพนธ์ฉบับนี้กำหนดให้ผู้เป็นเจ้าของข้อมูลและผู้ทำการรวมแบบจำลองข้อมูลสร้างโลคัลออนโทโลจีตามโครงสร้างของอัปเปอร์ออนโทโลจีที่ผู้วิจัยได้กำหนดไว้ในบทที่ 3 ซึ่งประกาศไว้ที่ http://www.upper-ontology.com/upper.owl ดังรูปที่ 5.2

ทั้งนี้ โลคัลออนโทโลจีแต่ละชุดจะถูกสร้างโดยอาศัยการทำงานผ่านโปรแกรมโพรเทเจ [32] และเก็บบันทึกไว้ในรูปของไฟล์ภาษาอาวล์หนึ่งไฟล์ หลังจากนั้น ผู้ทำการรวมแบบจำลองข้อมูลอาจเพิ่มความหมายของแบบจำลองข้อมูลลงในโลคัลออนโทโลจีที่ได้ และสร้างออนโทโลจีอีกชุดหนึ่งในรูปของไฟล์ภาษาอาวล์เช่นกัน เพื่อเก็บบันทึกความสัมพันธ์จากการระบุคำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างโลคัลออนโทโลจีทั้งสองชุดที่สร้างไว้ก่อนแล้ว

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://www.upper-ontology.com/upper.owl#"                    (ก)
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.upper-ontology.com/upper.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Class"/>                                           (ข)
  <owl:Class rdf:ID="ClassAttribute"/>                                  (ค)
  <owl:Class rdf:ID="ClassIntegrityConstraint">                         (ง)
    <owl:equivalentClass>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:ID="ConstraintValue"/>
          <owl:Class rdf:about="#ConstraintValue"/>
        </owl:unionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:about="#ConstraintValue">                              (จ)
    <rdfs:subClassOf rdf:resource="#ClassIntegrityConstraint"/>
  </owl:Class>
  <owl:Class rdf:ID="SemanticName"/>                                    (ฉ)
  <owl:Class rdf:ID="SemanticWord">                                     (ช)
```

รูปที่ 5.2 อัปเปอร์ออนโทโลจีในรูปของอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล

```
    <rdfs:subClassOf rdf:resource="#SemanticName"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="hasClassAttribute">                    (ช)
    <rdfs:range rdf:resource="#ClassAttribute"/>
    <rdfs:domain rdf:resource="#Class"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasClassAttributeAltType">             (ฌ)
    <rdfs:range rdf:resource="#Class"/>
    <rdfs:domain rdf:resource="#ClassAttribute"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasClassIntegrityConstraint">          (ญ)
    <rdfs:domain rdf:resource="#Class"/>
    <rdfs:range rdf:resource="#ClassIntegrityConstraint"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasClassSemantic">                     (ฎ)
    <rdfs:range rdf:resource="#SemanticName"/>
    <rdfs:domain rdf:resource="#Class"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasClassAttributeSemantic">            (ฏ)
    <rdfs:domain rdf:resource="#ClassAttribute"/>
    <rdfs:range rdf:resource="#SemanticName"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="hasClassAttributeType">              (ฐ)
    <rdfs:domain rdf:resource="#ClassAttribute"/>
  </owl:DatatypeProperty>
</rdf:RDF>
```

รูปที่ 5.2 อัปเปอร์ออนโทโลจีในรูปของอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล (ต่อ)

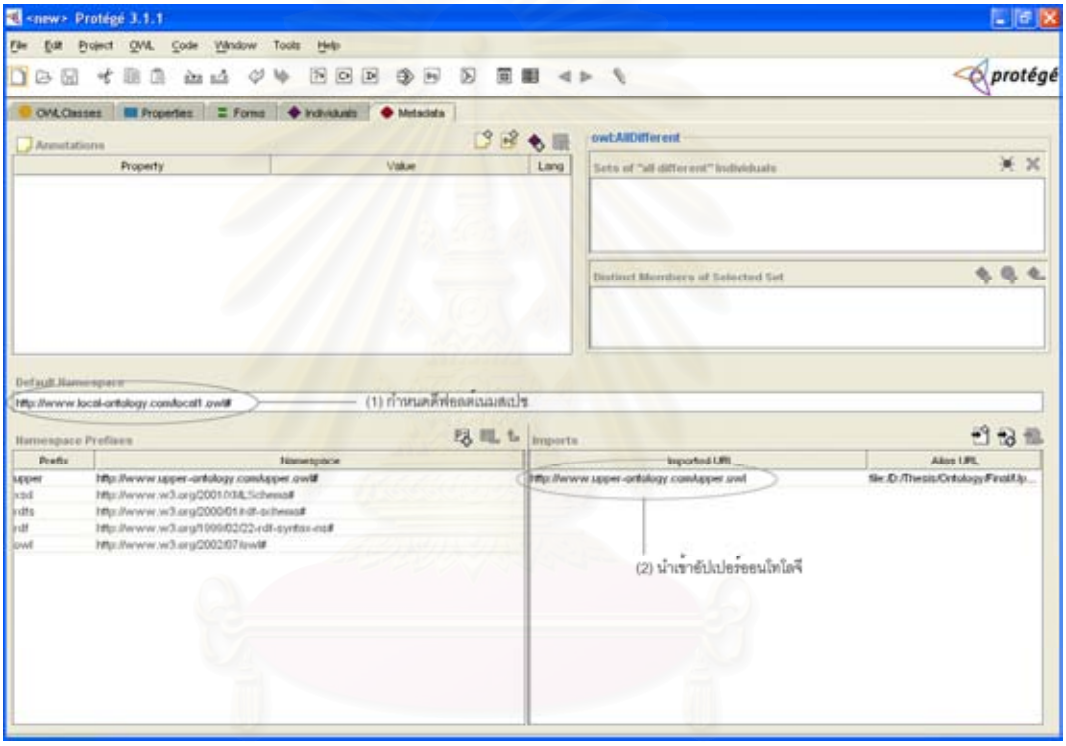
การทำงานของเฟรมเวิร์คในส่วนนี้อาจแบ่งได้เป็น 2 ส่วนย่อยคือ การสร้างโลคัลออนโทโลจีเพื่ออธิบายและเพิ่มความหมายของแบบจำลองข้อมูลเชิงวัตถุแต่ละชุด (ขั้นตอนที่ 1 และ 2 ในรูปที่ 5.1) และการสร้างออนโทโลจีหนึ่งชุดเพื่อระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดที่ได้สร้างไว้ (ขั้นตอนที่ 3 ในรูปที่ 5.1) วิทยานิพนธ์ฉบับนี้เลือกใช้การสร้างออนโทโลจีทั้งสองส่วนโดยอาศัยโปรแกรมโพรเทเจ [33] ซึ่งมีวิธีการดังนี้

5.2.1.1 การสร้างโลคัลออนโทโลจีเพื่ออธิบายและเพิ่มความหมายของแบบจำลองข้อมูลเชิงวัตถุ

จากแบบจำลองข้อมูลเชิงวัตถุหนึ่งชุด ขั้นตอนนี้เป็นการสร้างโลคัลออนโทโลจีหนึ่งชุดเพื่อแสดงแทนแบบจำลองข้อมูลเชิงวัตถุชุดดังกล่าว โดยโลคัลออนโทโลจีที่ได้นี้จะต้องสามารถอธิบายความหมายของแบบจำลองข้อมูลเชิงวัตถุทั้งความหมายตามโครงสร้างของแบบจำลองข้อมูล ได้แก่ คลาส แอททริบิวต์ ชนิดข้อมูลของแอททริบิวต์และความสัมพันธ์ระหว่างคลาส และความหมายในรูปของข้อมูลเชิงความหมาย ได้แก่ ความหมายของคลาส ความหมายของแอททริบิวต์ และข้อกำหนดบูรณภาพ เป็นต้น ซึ่งการสร้างโลคัลออนโทโลจีแต่ละชุดจะเป็นไปตามแนวทางที่ได้

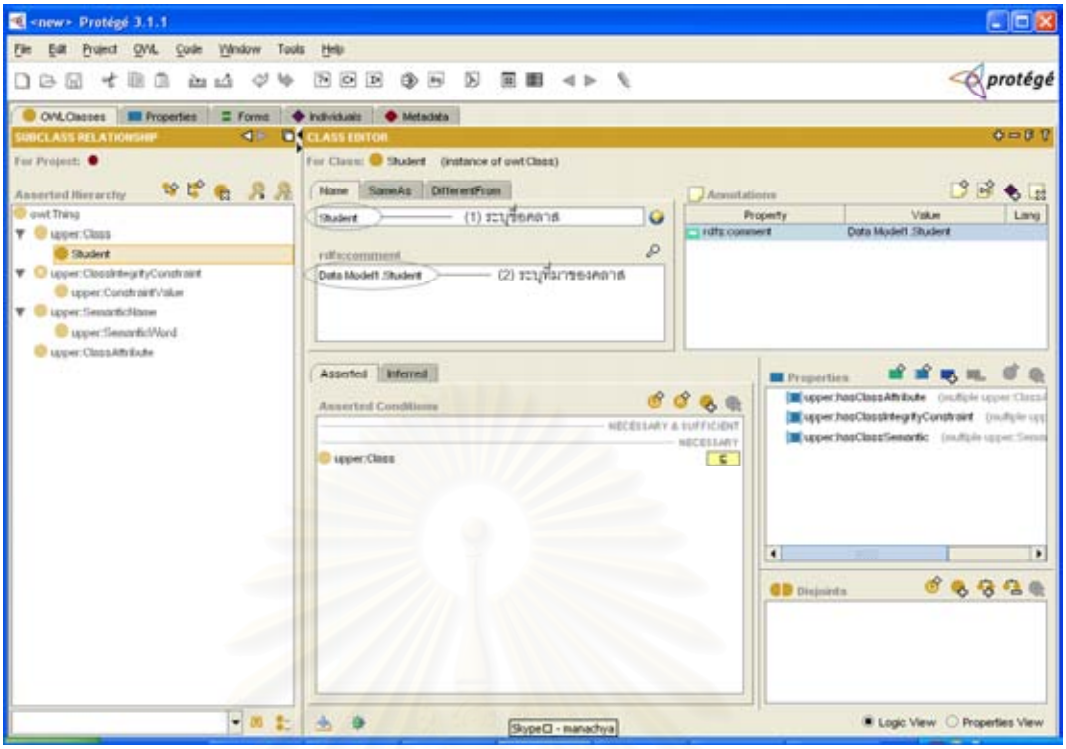
อธิบายไว้ในหัวข้อที่ 3.2 การสร้างโลคัลออนโทโลจีเพื่ออธิบายและเพิ่มความหมายของแบบจำลองข้อมูลเชิงวัตถุมีขั้นตอนดังต่อไปนี้

1. เปิดโปรแกรมโพรเทเจและเลือก “สร้างโปรเจ็กต์ใหม่ (Create New Project)” แล้วเลือกชนิดของโปรเจ็กต์เป็น “ไฟล์อาวล์ (OWL Files (.owl or .rdf))”
2. จากรูปที่ 5.3 กำหนดดีฟอลต์เนมสเปซ (Default Namespace) ของโลคัลออนโทโลจีที่ต้องการสร้าง เช่น http://www.local-ontology.com/local1.owl# และนำเข้า (Imports) อัปเปอร์ออนโทโลจีจาก http://www.upper-ontology.com/upper.owl ตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ก) เพื่อใช้เป็นโครงสร้างของโลคัลออนโทโลจีที่จะสร้างต่อไป

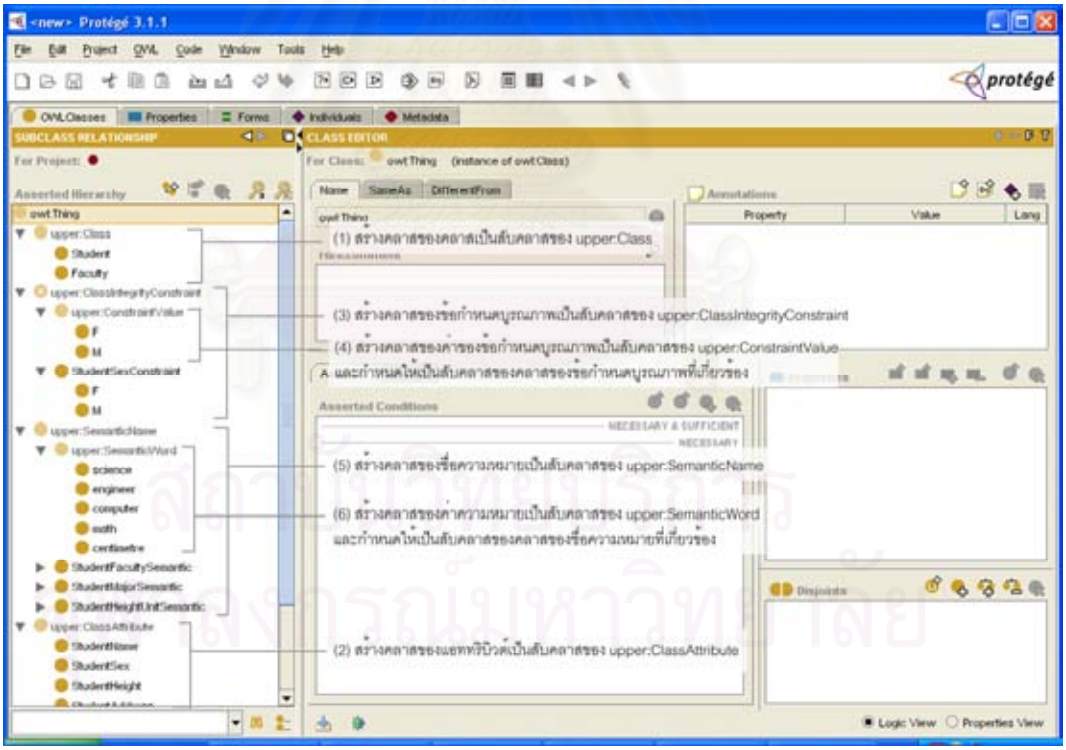


รูปที่ 5.3 หน้าจอโปรแกรมโพรเทเจเพื่อกำหนดดีฟอลต์เนมสเปซและนำเข้าอัปเปอร์ออนโทโลจี

3. จากรูปที่ 5.4 วิธีการสร้างคลาสต่างๆ ภายในโลคัลออนโทโลจี ประกอบด้วยการระบุชื่อของคลาสที่ต้องการสร้าง และระบุที่มาของคลาสที่สร้างว่าเดิมเป็นส่วนใดในแบบจำลองข้อมูลเพื่อนำไปใช้ประโยชน์ในขั้นตอนการกำกับออนโทโลจีรวมลงในแบบจำลองข้อมูลเชิงวัตถุ
4. จากรูปที่ 5.5 สร้างคลาสต่างๆ ลงในโลคัลออนโทโลจีตามวิธีการในรูปที่ 5.4 เพื่อแสดงแทนคลาส แอททริบิวต์ ข้อกำหนดบูรณภาพ ค่าที่เป็นไปได้ของแอททริบิวต์จากข้อกำหนดบูรณภาพ ชื่อของความหมายและค่าของความหมายของแบบจำลองข้อมูล โดยมีข้อกำหนดดังนี้
   - สร้างคลาสของคลาสเพื่อแสดงแทนคลาส โดยกำหนดเป็นสับคลาสของ upper:Class ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ข)



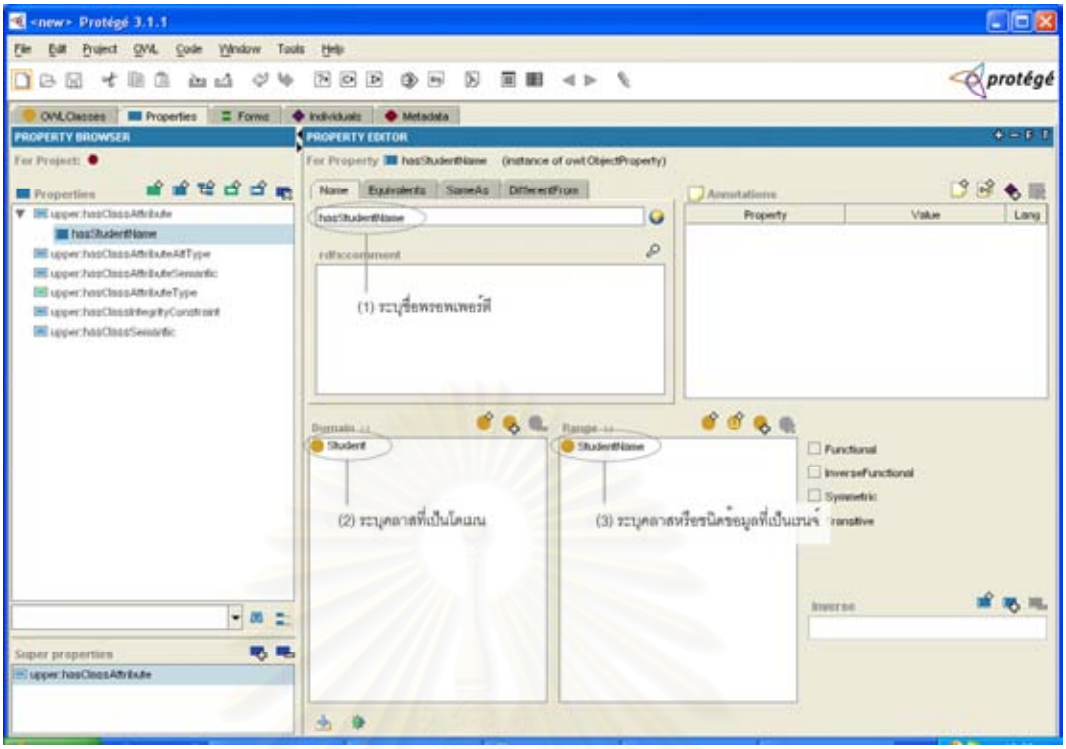
รูปที่ 5.4 หน้าจอโปรแกรมโพรเทเจเพื่อสร้างคลาสภายในโลคัลออนโทโลจี



รูปที่ 5.5 หน้าจอโปรแกรมโพรเทเจเพื่อสร้างคลาสต่างๆ ลงในโลคัลออนโทโลจี

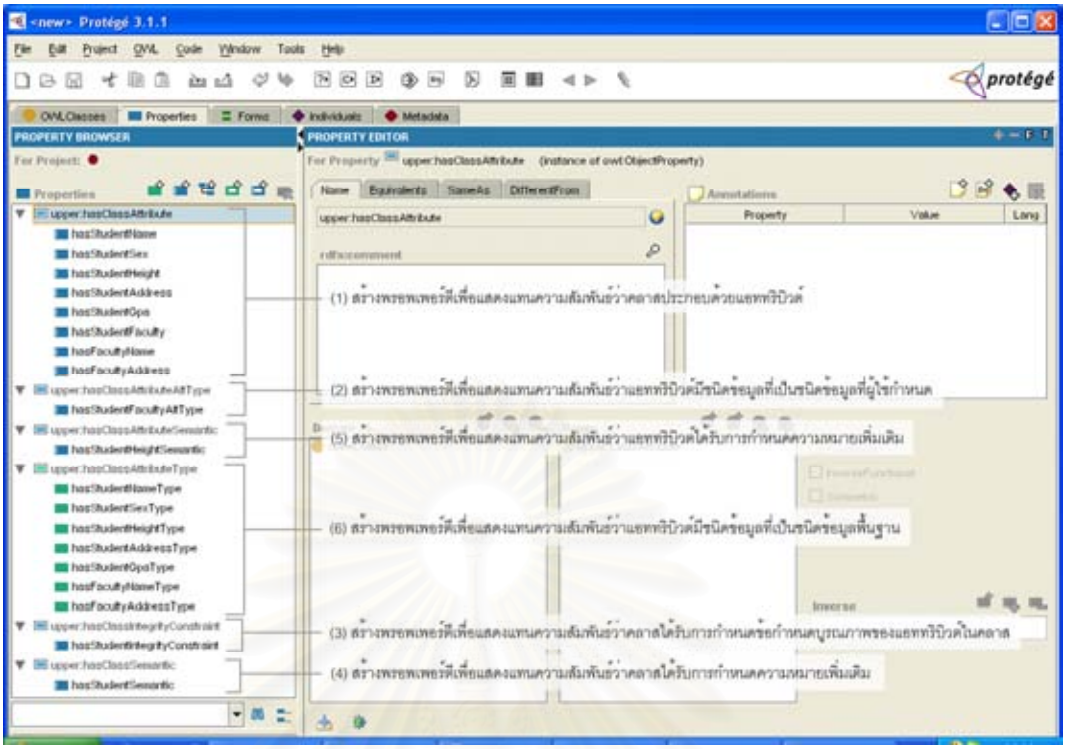
- หากมีความสัมพันธ์แบบสืบทอดคุณลักษณะปรากฏอยู่ในแบบจำลองข้อมูลเชิงวัตถุ ให้สร้างคลาสของคลาสคู่นั้นในรูปของซูเปอร์คลาส/ สับคลาส เพื่อแสดงแทนความสัมพันธ์ดังกล่าว โดยคลาสทั้งคู่ยังคงเป็นสับคลาสของ upper:Class ของอัปเปอร์ออนโทโลจี
- สร้างคลาสของแอททริบิวต์เพื่อแสดงแทนแอททริบิวต์ โดยกำหนดเป็นสับคลาสของ upper:ClassAttribute ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ค)
- หากมีความสัมพันธ์แบบแอกกรีเกชันปรากฏอยู่ในแบบจำลองข้อมูลเชิงวัตถุ ให้สร้างคลาสของแอททริบิวต์เพื่อแสดงแทนความสัมพันธ์ดังกล่าว โดยกำหนดเป็นสับคลาสของ upper:ClassAttribute ของอัปเปอร์ออนโทโลจี เช่นกัน
- สร้างคลาสของข้อกำหนดบูรณภาพเพื่อแสดงแทนข้อกำหนดบูรณภาพของแอททริบิวต์ภายในคลาส โดยกำหนดเป็นสับคลาสของ upper:ClassIntegrityConstraint ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ง)
- สร้างคลาสของค่าของข้อกำหนดบูรณภาพเพื่อแสดงแทนค่าที่เป็นไปได้ของแอททริบิวต์จากข้อกำหนดบูรณภาพ โดยกำหนดเป็นสับคลาสของ upper:ConstraintValue ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(จ) พร้อมทั้งกำหนดให้เป็นสับคลาสของคลาสของข้อกำหนดบูรณภาพที่เกี่ยวข้อง
- กำหนดเงื่อนไขจำเป็นและเพียงพอให้กับคลาสของข้อกำหนดบูรณภาพว่า ค่าที่เป็นไปได้ของแอททริบิวต์ภายในคลาสจากข้อกำหนดบูรณภาพนั้นจะต้องเท่ากับผลการยูเนียนของคลาสของค่าของข้อกำหนดบูรณภาพที่เกี่ยวข้องกัน ซึ่งตรงกับเงื่อนไขที่กำหนดไว้ภายในอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ง) ว่า “ClassIntegrityConstraint = $\cup$ ConstraintValue”
- สร้างคลาสของชื่อความหมายเพื่อแสดงแทนชื่อของความหมายของคลาสหรือแอททริบิวต์ โดยกำหนดเป็นสับคลาสของ upper:SemanticName ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ฉ)
- สร้างคลาสของค่าความหมายเพื่อแสดงแทนค่าของความหมายของคลาสหรือแอททริบิวต์ โดยกำหนดเป็นสับคลาสของ upper:SemanticWord ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ช) พร้อมทั้งกำหนดให้เป็นสับคลาสของคลาสของชื่อความหมายที่เกี่ยวข้อง

5. จากรูปที่ 5.6 วิธีการสร้างพรอพเพอร์ตีต่างๆ ภายในโลคัลออนโทโลจี ประกอบด้วยการระบุชื่อของพรอพเพอร์ตีที่ต้องการสร้าง ระบุคลาสที่เป็นโดเมน (Domain) ของพรอพเพอร์ตี และระบุคลาส (สำหรับพรอพเพอร์ตีแบบอ็อบเจกต์) หรือชนิดข้อมูล (สำหรับพรอพเพอร์ตีแบบชนิดข้อมูล) ที่เป็นเรนจ์ (Range) ของพรอพเพอร์ตีนั้น



รูปที่ 5.6 หน้าจอโปรแกรมโพรเทเจเพื่อสร้างพรอพเพอร์ตีภายในโลคัลออนโทโลจี

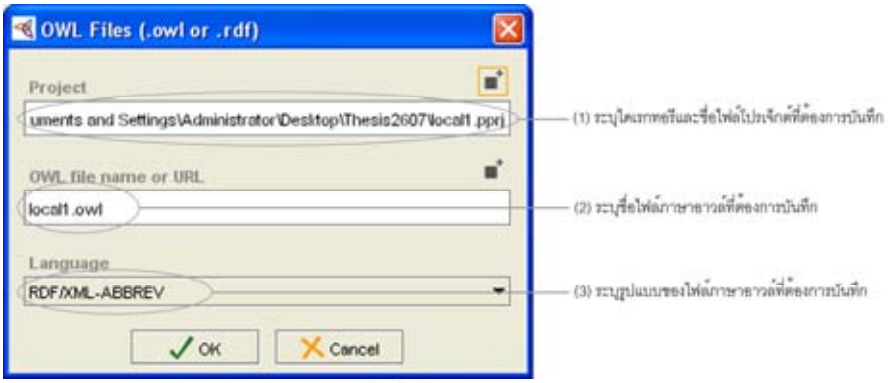
6. จากรูปที่ 5.7 สร้างพรอพเพอร์ตีต่างๆ ลงในโลคัลออนโทโลจีเพื่อแสดงแทนความสัมพันธ์ระหว่างคลาสชนิดต่างๆ ที่ได้สร้างไว้ในขั้นตอนก่อนหน้านี้ โดยมีข้อกำหนดดังนี้
   - สร้างพรอพเพอร์ตีเพื่อแสดงแทนความสัมพันธ์ว่าคลาสประกอบด้วยแอททริบิวต์ จากโดเมนคือคลาสของคลาสไปยังเรนจ์คือคลาสของแอททริบิวต์ โดยกำหนดเป็นสับพรอพเพอร์ตีของ upper:hasClassAttribute ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ซ)
   - สร้างพรอพเพอร์ตีเพื่อแสดงแทนความสัมพันธ์ว่าแอททริบิวต์มีชนิดข้อมูลที่เป็นชนิดข้อมูลที่ผู้ใช้กำหนด (ใช้แสดงแทนความสัมพันธ์แบบแอกกรีเกชันในแบบจำลองข้อมูล) จากโดเมนคือคลาสของแอททริบิวต์ไปยังเรนจ์คือคลาสของคลาส โดยกำหนดเป็นสับพรอพเพอร์ตีของ upper:hasClassAttributeAltType ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ฌ)
   - สร้างพรอพเพอร์ตีเพื่อแสดงแทนความสัมพันธ์ว่าคลาสได้รับการกำหนดข้อกำหนดบูรณภาพของแอททริบิวต์ในคลาส จากโดเมนคือคลาสของคลาสไปยังเรนจ์คือคลาสของข้อกำหนดบูรณภาพ โดยกำหนดเป็นสับพรอพเพอร์ตีของ upper:hasClassIntegrityConstraint ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ญ)



รูปที่ 5.7 หน้าจอโปรแกรมโพรเทเจเพื่อสร้างพรอพเพอร์ตีต่างๆ ลงในโลคัลออนโทโลจี

- สร้างพรอพเพอร์ตีเพื่อแสดงแทนความสัมพันธ์ว่าคลาสได้รับการกำหนดความหมายเพิ่มเติม จากโดเมนคือคลาสของคลาสไปยังเรนจ์คือคลาสของชื่อความหมาย โดยกำหนดเป็นสับพรอพเพอร์ตีของ upper:hasClassSemantic ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ฎ)
- สร้างพรอพเพอร์ตีเพื่อแสดงแทนความสัมพันธ์ว่าแอททริบิวต์ได้รับการกำหนดความหมายเพิ่มเติม จากโดเมนคือคลาสของแอททริบิวต์ไปยังเรนจ์คือคลาสของชื่อความหมาย โดยกำหนดเป็นสับพรอพเพอร์ตีของ upper:hasClassAttributeSemantic ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ฎ)
- สร้างพรอพเพอร์ตีเพื่อแสดงแทนความสัมพันธ์ว่าแอททริบิวต์มีชนิดข้อมูลที่เป็นชนิดข้อมูลพื้นฐาน จากโดเมนคือคลาสของแอททริบิวต์ไปยังเรนจ์คือชนิดข้อมูล โดยกำหนดเป็นสับพรอพเพอร์ตีของ upper:hasClassAttributeType ของอัปเปอร์ออนโทโลจีตามที่ผู้วิจัยประกาศไว้ดังรูปที่ 5.2(ฐ)

7. จากรูปที่ 5.8 วิธีการเก็บบันทึกไฟล์โลคัลออนโทโลจีแต่ละชุดที่สร้างเสร็จแล้ว ประกอบด้วยการระบุไดเรกทอรีและชื่อไฟล์โปรเจ็กต์ที่ต้องการบันทึกโลคัลออนโทโลจีนั้น และระบุชื่อไฟล์ภาษาอาวล์ที่ต้องการบันทึก ซึ่งไฟล์ดังกล่าวนี้จะนำไปใช้ในส่วนการสร้างออนโทโลจีรวมในขั้นตอนที่ 5.2.2 ต่อไป พร้อมทั้งกำหนดรูปแบบของไฟล์ภาษาอาวล์ที่ต้องการบันทึก ซึ่งในที่นี้กำหนดให้บันทึกในรูปของอาร์ดีเอฟ/ เอ็กซ์เอ็มแอลเพื่อให้เหมาะสมกับการประมวลผลด้วยจีนา [35] ในการโปรแกรมส่วนการสร้างออนโทโลจีรวม



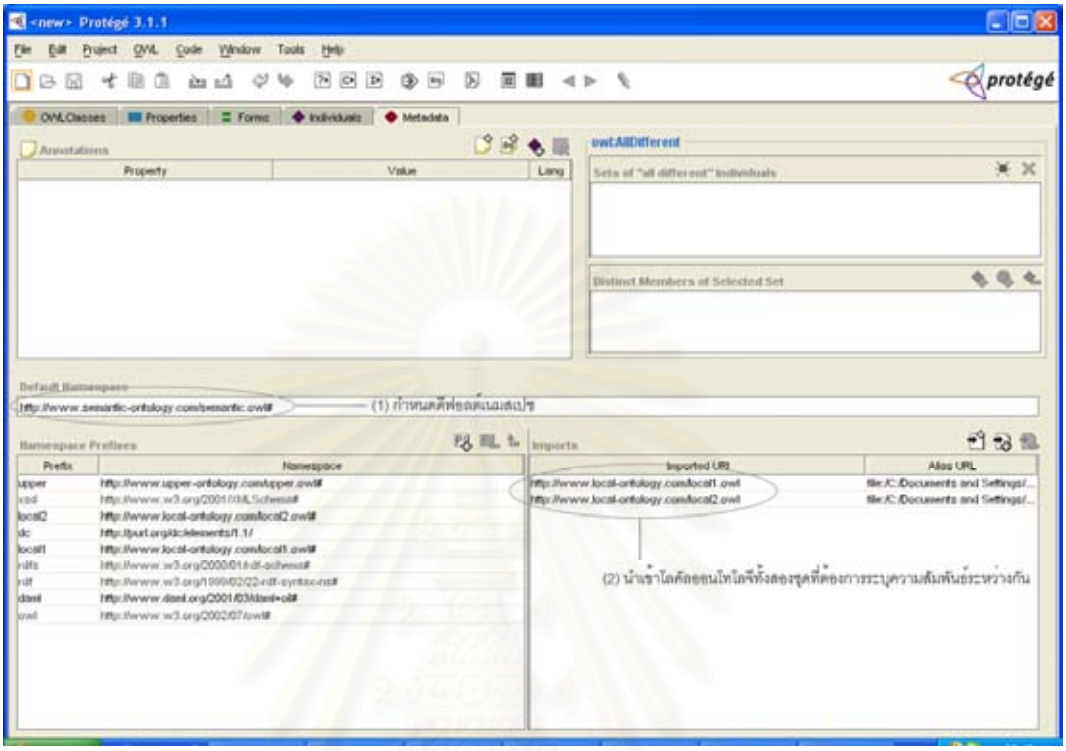
รูปที่ 5.8 หน้าจอโปรแกรมโพรเทเจเพื่อเก็บบันทึกไฟล์โลคัลออนโทโลจีแต่ละชุด

5.2.1.2 การสร้างออนโทโลจีเพื่อระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีสองชุดที่ได้สร้างไว้

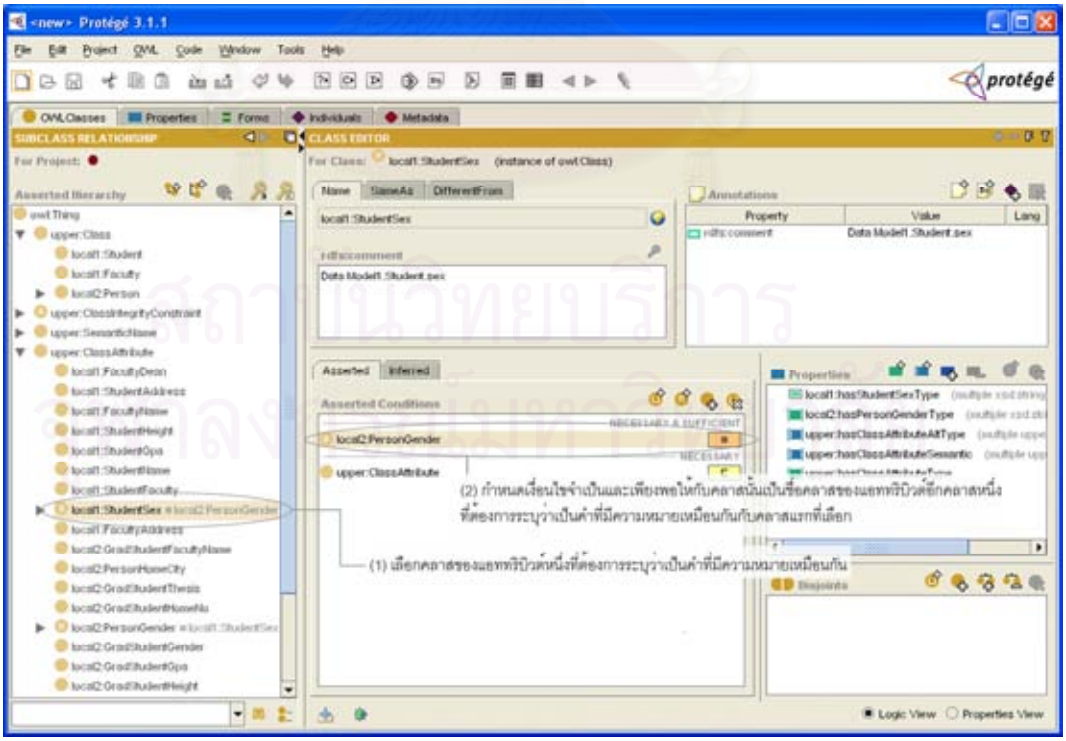
หลังจากสร้างโลคัลออนโทโลจีสองชุดเพื่ออธิบายและเพิ่มความหมายของแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดที่ต้องการรวมด้วยวิธีการในหัวข้อที่ 5.2.1.1 แล้ว ขั้นตอนนี้เป็นการสร้างออนโทโลจีหนึ่งชุดเพื่อระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดที่ได้สร้างไว้แล้ว โดยออนโทโลจีที่ได้นี้จะต้องสามารถแสดงให้ทราบได้ว่ามีคำคู่ใดในโลคัลออนโทโลจีทั้งสองชุดเป็นคำที่มีความหมายเหมือนกัน หรือมีคำใดในโลคัลออนโทโลจีชุดที่หนึ่งเป็นคำแม่กลุ่มของคำลูกกลุ่มๆ หนึ่งในโลคัลออนโทโลจีชุดที่สอง ซึ่งการระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีแต่ละคู่จะเป็นไปตามแนวทางที่ได้อธิบายไว้ในหัวข้อที่ 3.3 การสร้างโลคัลออนโทโลจีระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีสองชุดมีขั้นตอนดังต่อไปนี้

1. เปิดโปรแกรมโพรเทเจและเลือก “สร้างโปรเจ็กต์ใหม่ (Create New Project)” แล้วเลือกชนิดของโปรเจ็กต์เป็น “ไฟล์อาวล์ (OWL Files (.owl or .rdf))”
2. จากรูปที่ 5.9 กำหนดดีฟอลต์เนมสเปซของออนโทโลจีที่ต้องการสร้าง เช่น http://www.semantic-ontology.com/semantic.owl# และนำเข้าโลคัลออนโทโลจีทั้งสองชุดที่ต้องการระบุความสัมพันธ์ระหว่างกัน เช่น จาก http://www.local-ontology.com/local1.owl และจาก http://www.local-ontology.com/local2.owl เพื่อใช้สร้างแมปปิงระหว่างโลคัลออนโทโลจีทั้งสองชุด ทั้งนี้ หลังจากนำเข้าโลคัลออนโทโลจีทั้งสองชุดแล้วคลาสและพรอพเพอร์ตีทั้งหมดของโลคัลออนโทโลจีทั้งสองชุดจะปรากฏขึ้น
3. จากรูปที่ 5.10 วิธีการระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีโดยกำหนดว่าเป็นคำที่มีความหมายเหมือนกัน ประกอบด้วยการเลือกคลาสของคลาสหรือคลาสของแอททริบิวต์คลาสใดคลาสหนึ่งที่ต้องการระบุว่าเป็นคำที่มีความหมายเหมือนกัน แล้วกำหนดเงื่อนไขจำเป็นและเพียงพอให้กับคลาสนั้นเป็นชื่อคลาสของคลาสหรือคลาสของแอททริบิวต์อีกคลาสหนึ่งที่ต้องการระบุว่าเป็นคำที่มีความหมายเหมือนกันกับคลาสแรกที่เลือก ทั้งนี้การระบุความสัมพันธ์ด้วยวิธีนี้ ความสัมพันธ์ที่ได้จะเป็นแบบสมมาตร (Symmetry) นั่นคือเมื่อกำหนดว่า คลาสของแอททริบิวต์ CA1 ในโลคัลออนโทโลจีชุดที่หนึ่งเป็นคำที่มีความหมาย

เหมือนกับคลาสของแอททริบิวต์ CA2 ในโลคัลออนโทโลจีชุดที่สองแล้ว จะสามารถสรุปได้ว่าคลาสของแอททริบิวต์ CA2 เป็นคำที่มีความหมายเหมือนกับคลาสของแอททริบิวต์ CA1 โดยอัตโนมัติ

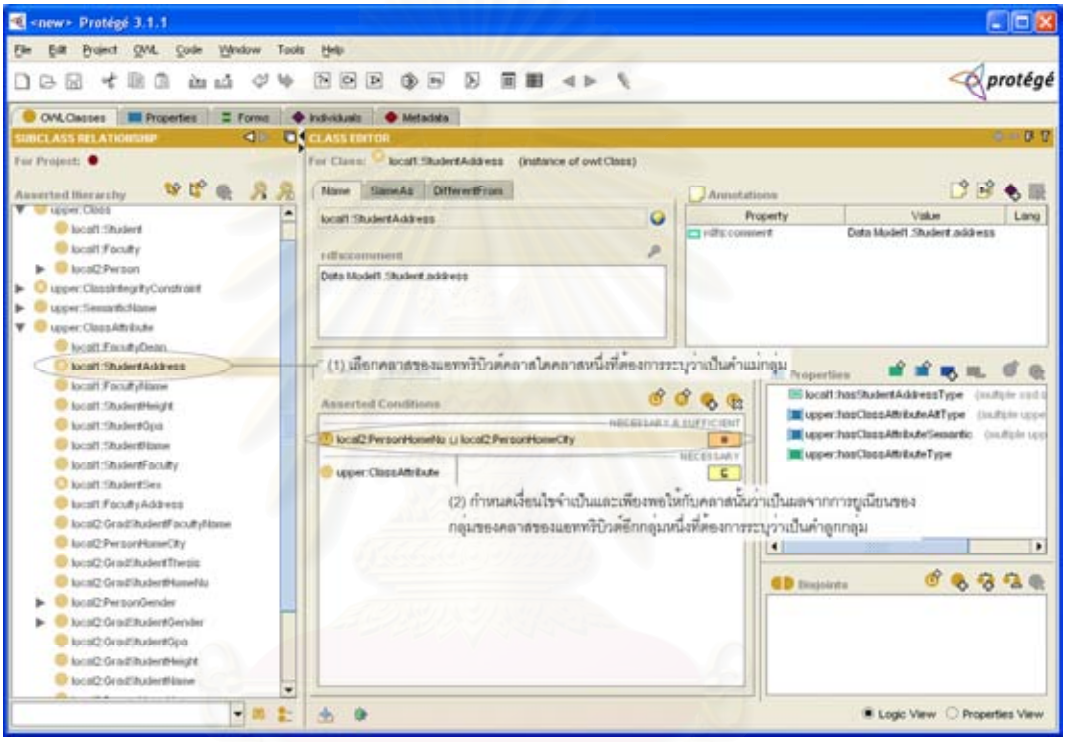


รูปที่ 5.9 หน้าจอโปรแกรมโพรเทเจเพื่อกำหนดดีฟอลต์เนมสเปซและนำเข้าโลคัลออนโทโลจี



รูปที่ 5.10 หน้าจอโปรแกรมโพรเทเจเพื่อระบุว่าเป็นคำที่มีความหมายเหมือนกันระหว่างโลคัลออนโทโลจี

4. จากรูปที่ 5.11 วิธีการระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีโดยกำหนดว่าเป็นคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกัน ประกอบด้วยการเลือกคลาสของคลาสหรือคลาสของแอททริบิวต์คลาสใดคลาสหนึ่งที่ต้องการระบุว่าเป็นคำแม่กลุ่ม แล้วกำหนดเงื่อนไขจำเป็นและเพียงพอให้กับคลาสนั้นว่าเป็นผลจากการยูเนียนของกลุ่มของคลาสของคลาสหรือกลุ่มของคลาสของแอททริบิวต์อีกกลุ่มหนึ่งที่ต้องการระบุว่าเป็นคำลูกกลุ่ม
5. เลือกเก็บบันทึกไฟล์ออนโทโลจีที่สร้างเสร็จแล้วด้วยวิธีการเดียวกับการเก็บบันทึกไฟล์โลคัลออนโทโลจีในหัวข้อที่ 5.2.1.1



รูปที่ 5.11 หน้าจอโปรแกรมโพรเทเจเพื่อระบุว่าเป็นคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างโลคัลออนโทโลจี

จากการทำงานของเฟรมเวิร์คส่วนการสร้างโลคัลออนโทโลจีในทั้งสองขั้นตอนนี้ ผู้ทำการรวมแบบจำลองข้อมูลจะได้รับไฟล์โลคัลออนโทโลจีสองไฟล์ และไฟล์ออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุด ซึ่งทั้งสามไฟล์นี้จะถูกนำไปใช้งานในเฟรมเวิร์คส่วนการสร้างออนโทโลจีรวมที่จะอธิบายให้หัวข้อต่อไป ดังนั้น ผู้ทำการรวมแบบจำลองข้อมูลและผู้เป็นเจ้าของแบบจำลองมูลควรจะให้ความสำคัญกับความถูกต้องและครบถ้วนในการสร้างออนโทโลจีทั้งสามชุดเพื่ออธิบายแบบจำลองข้อมูลที่ต้องการรวม โดยพยายามอธิบายแบบจำลองข้อมูลทั้งสองชุดให้ชัดเจนที่สุดเพื่อให้สามารถจำแนกความขัดแย้งของข้อมูลที่มีได้ อันจะส่งผลให้การรวมออนโทโลจีเป็นไปอย่างเหมาะสมและถูกต้องที่สุด

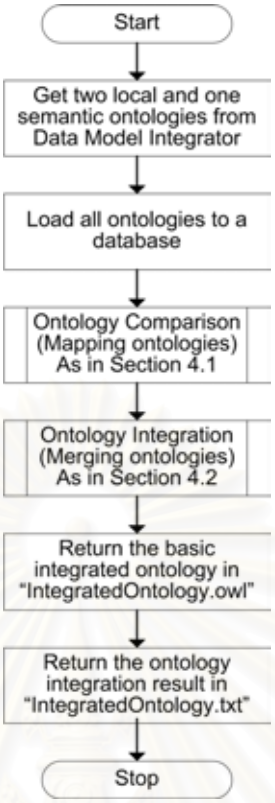
5.2.2 ส่วนการสร้างออนโทโลจีรวม

เฟรมเวิร์คส่วนการสร้างออนโทโลจีรวมประกอบด้วยขั้นตอนการรับไฟล์โลคัลออนโทโลจีสองชุดที่ต้องการรวม พร้อมทั้งไฟล์ออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุด จากผู้ทำการรวมแบบจำลองข้อมูลเข้าสู่โปรแกรมการรวมออนโทโลจี หลังจากนั้นโปรแกรมจะดำเนินการเปรียบเทียบโลคัลออนโทโลจีทั้งสองชุดตามขั้นตอนวิธีในหัวข้อที่ 4.1 เพื่อหาค่าความสัมพันธ์ระหว่างคลาสของคลาสทุกคู่ในโลคัลออนโทโลจีทั้งสองชุด และนำผลการเปรียบเทียบที่ได้มาใช้สร้างออนโทโลจีรวมพื้นฐานในรูปของไฟล์ภาษาอาวล์ (ขั้นตอนที่ 4 ในรูปที่ 5.1) ในขณะเดียวกันก็สร้างไฟล์ผลการรวมออนโทโลจีเพื่อแสดงผลการเปรียบเทียบและรายละเอียดที่จำเป็นในการสร้างออนโทโลจีรวมตามขั้นตอนวิธีในหัวข้อที่ 4.2 เมื่อได้รับไฟล์ผลการรวมออนโทโลจี รายละเอียดในไฟล์ดังกล่าวจะช่วยให้ผู้ทำการรวมแบบจำลองข้อมูลสามารถปรับแต่งออนโทโลจีรวมพื้นฐานและเพิ่มกฎเพื่ออนุมานหาข้อเท็จจริงเพิ่มเติมทำให้ออนโทโลจีรวมที่ได้มีความสมบูรณ์ยิ่งขึ้นก่อนที่จะนำไปใช้งาน (ขั้นตอนที่ 5 ในรูปที่ 5.1) ทั้งนี้ การทำงานของเฟรมเวิร์คในส่วนนี้มุ่งเน้นไปที่การสร้างออนโทโลจีรวมขึ้นจริง เพื่อให้สามารถนำออนโทโลจีรวมที่ได้ไปใช้เป็นตัวกลางในการสอบถามข้อมูล และเป็นแมปปิงไปยังแบบจำลองข้อมูลเชิงวัตถุที่นำมารวม

การทำงานของเฟรมเวิร์คในส่วนนี้อาจแบ่งได้เป็น 2 ส่วนย่อยคือ การเปรียบเทียบโลคัลออนโทโลจีเพื่อสร้างออนโทโลจีรวมพื้นฐาน และการอนุมานออนโทโลจีรวมพื้นฐานเพื่อให้ได้ออนโทโลจีรวมที่สมบูรณ์สำหรับนำไปใช้สร้างแมปปิงจากออนโทโลจีรวมดังกล่าวไปยังแบบจำลองข้อมูลเชิงวัตถุที่นำมารวม ซึ่งมีวิธีการดังนี้

5.2.2.1 การเปรียบเทียบโลคัลออนโทโลจีเพื่อสร้างออนโทโลจีรวมพื้นฐาน

จากไฟล์โลคัลออนโทโลจีสองชุดที่ต้องการรวม และไฟล์ออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุด ขั้นตอนนี้เป็นการสร้างออนโทโลจีรวมพื้นฐานจากผลการเปรียบเทียบโลคัลออนโทโลจีคู่ดังกล่าว โดยอาศัยโปรแกรมการรวมออนโทโลจี (Ontology Integration Program) ที่ผู้วิจัยพัฒนาขึ้นด้วยภาษาจาวา ร่วมกับการใช้งานจีนาเอพีไอรุ่นที่ 2.4 [35] เพื่อจัดการการเข้าถึงข้อมูลในออนโทโลจี และฐานข้อมูลมายเอสคิวแอล (MySQL) เวอร์ชัน 5.0.24a [40] ซึ่งสนับสนุนการจัดเก็บออนโทโลจีในรูปของฐานข้อมูลโดยใช้การติดต่อผ่านจีนาเอพีไอ เพื่อจัดเก็บออนโทโลจีและผลลัพธ์ที่ได้จากขั้นตอนการเปรียบเทียบและการรวมโลคัลออนโทโลจี โปรแกรมการรวมออนโทโลจีจะกำหนดให้ผู้ทำการรวมแบบจำลองข้อมูลป้อนไฟล์ภาษาอาวล์ของออนโทโลจีทั้งสามไฟล์ที่ได้รับจากเฟรมเวิร์คส่วนการสร้างโลคัลออนโทโลจีเข้าสู่โปรแกรม แล้วดำเนินการเปรียบเทียบและรวมโลคัลออนโทโลจีที่ต้องรวมทั้งสองชุดเข้าด้วยกันแบบอัตโนมัติ หลังจากนั้น เมื่อโปรแกรมทำงานเสร็จสิ้นผู้ทำการรวมแบบจำลองข้อมูลจะได้รับไฟล์ออนโทโลจีรวมพื้นฐานชื่อ IntegratedOntology.owl และไฟล์ผลการรวมออนโทโลจีชื่อ IntegratedOntology.txt ขั้นตอนการทำงานของโปรแกรมการรวมออนโทโลจีแสดงได้ดังรูปที่ 5.12

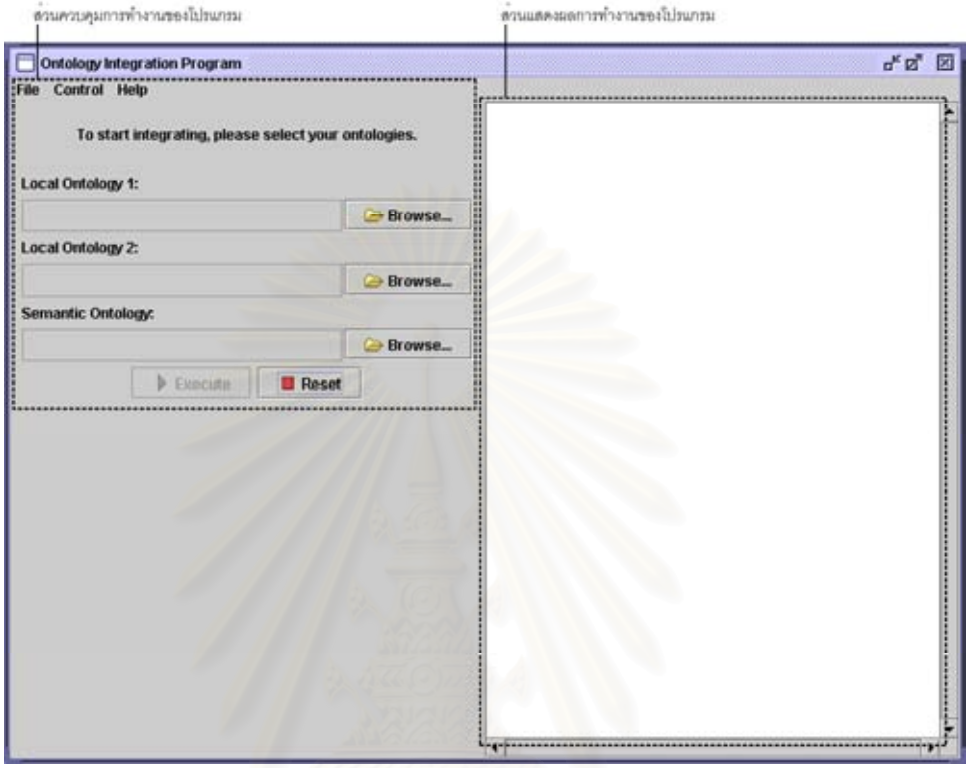


รูปที่ 5.12 ขั้นตอนการทำงานของโปรแกรมการรวมออนโทโลจี

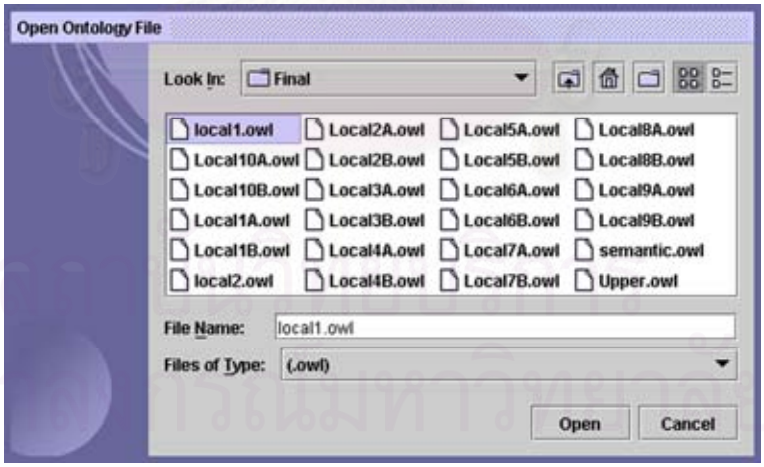
การทำงานของโปรแกรมการรวมออนโทโลจีประกอบด้วยหน้าจอต่างๆ ที่แสดงการทำงานแต่ละขั้นตอนดังรูปที่ 5.13 ถึง 5.17 มีรายละเอียดดังนี้

1. รูปที่ 5.13 แสดงส่วนประกอบของหน้าจอโปรแกรมการรวมออนโทโลจี ประกอบด้วยสองส่วนคือ ส่วนควบคุมการทำงานของโปรแกรม (ด้านซ้าย) ซึ่งใช้สำหรับป้อนไฟล์ออนโทโลจีให้กับโปรแกรมและสั่งการทำงานของโปรแกรม และส่วนแสดงผลการทำงานของโปรแกรม (ด้านขวา) ซึ่งใช้แสดงผลลัพธ์จากการทำงานของโปรแกรม
2. จากรูปที่ 5.14 เมื่อผู้ทำการรวมแบบจำลองข้อมูลกดปุ่ม Browse จะปรากฏหน้าต่างนี้ขึ้นเพื่อให้เลือกระบุตำแหน่งและชื่อไฟล์ออนโทโลจีแต่ละไฟล์สำหรับใช้ในการรวม ทั้งนี้โปรแกรมได้รับการกำหนดให้ทำงานได้กับไฟล์ออนโทโลจีที่อยู่ในรูปภาษาอาวล์ (.owl) เท่านั้น
3. จากรูปที่ 5.15 หลังจากผู้ทำการรวมแบบจำลองข้อมูลระบุชื่อไฟล์ออนโทโลจีทั้งสามไฟล์ที่จำเป็นต้องใช้ในการรวมออนโทโลจีแล้ว ผู้ทำการรวมจะสามารถเลือกสั่งให้โปรแกรมเริ่มดำเนินการเปรียบเทียบและรวมออนโทโลจีโดยการกดปุ่ม Execute หรืออาจกดปุ่ม Reset เพื่อเริ่มการระบุชื่อไฟล์ออนโทโลจีใหม่อีกครั้ง

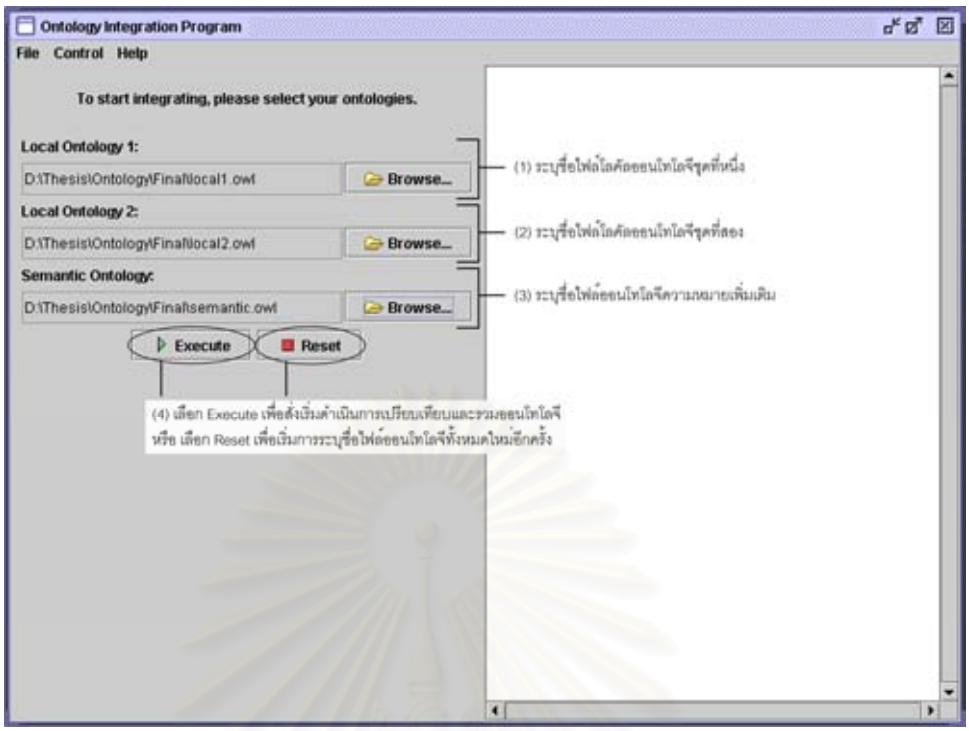
4. จากรูปที่ 5.16 หลังจากผู้ทำการรวมแบบจำลองข้อมูลเลือกสั่งให้โปรแกรมเริ่มทำงาน โปรแกรมจะเริ่มจากการจัดก็บออนโทโลจีทั้งสามชุดลงในฐานข้อมูล เพื่อให้สะดวกต่อการเปรียบเทียบ และแสดงผลการทำงานเมื่อขั้นตอนดังกล่าวเสร็จสมบูรณ์ดังรูป



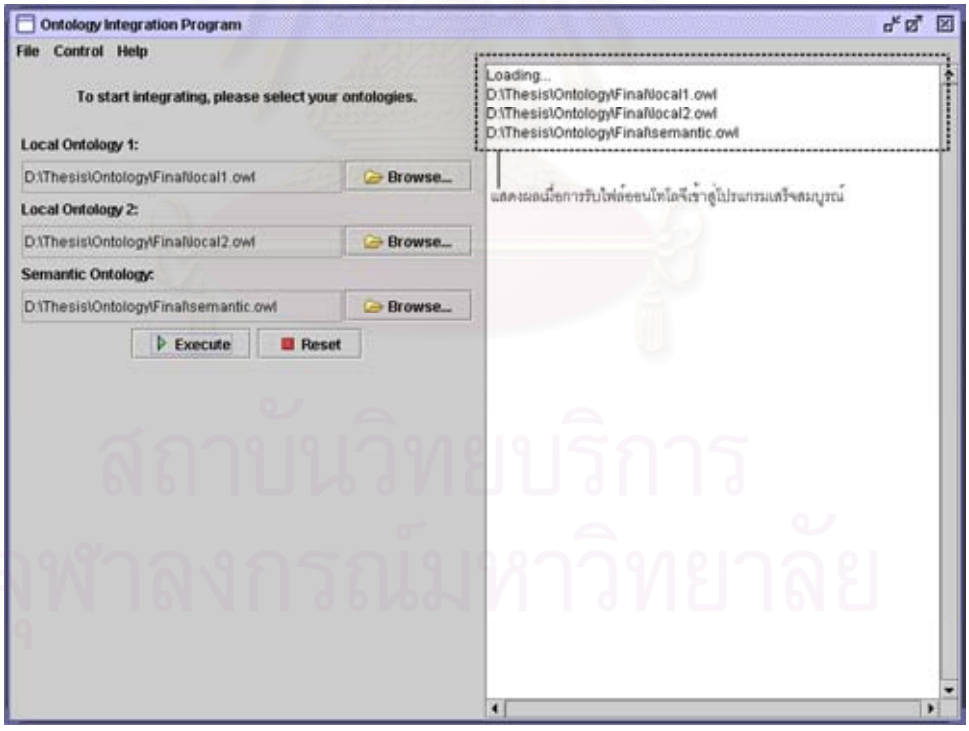
รูปที่ 5.13 หน้าจอการทำงานของโปรแกรมการรวมออนโทโลจี



รูปที่ 5.14 หน้าต่างสำหรับเลือกไฟล์โลคัลออนโทโลจีและออนโทโลจีความหมายของโปรแกรมการรวมออนโทโลจี

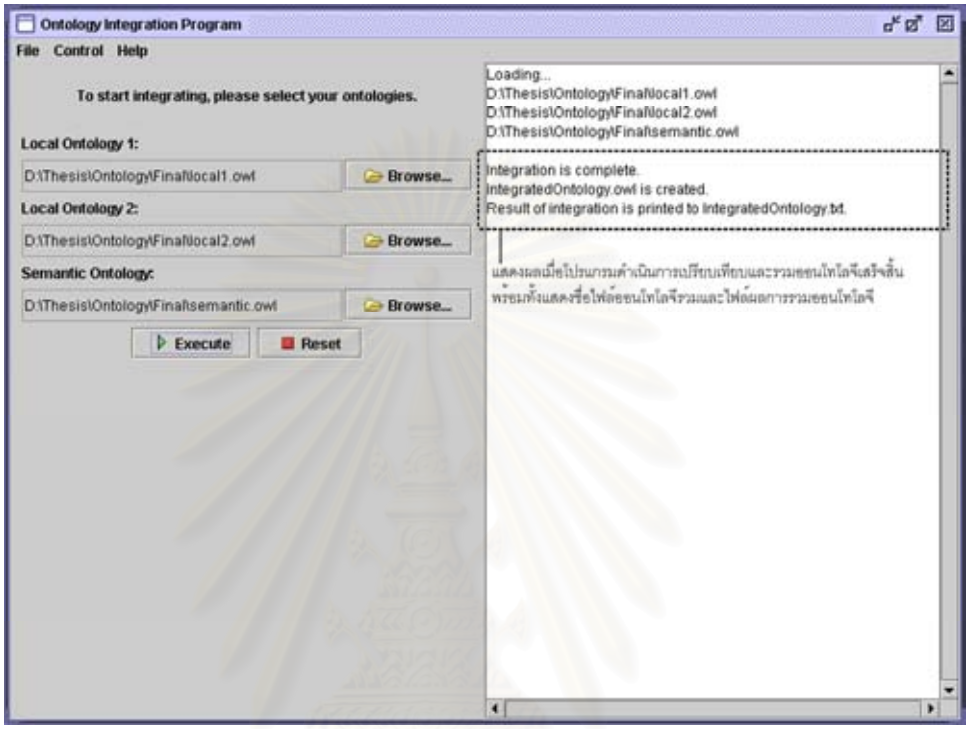


รูปที่ 5.15 หน้าจอโปรแกรมการรวมออนโทโลจีเมื่อระบุไฟล์ออนโทโลจีทั้งสามไฟล์ก่อนดำเนินการเปรียบเทียบและรวมออนโทโลจี



รูปที่ 5.16 หน้าจอโปรแกรมการรวมออนโทโลจีแสดงผลลัพธ์เมื่อไฟล์ออนโทโลจีทั้งสามได้รับการป้อนเข้าสู่โปรแกรม

5. จากรูปที่ 5.17 หลังจากนั้นโปรแกรมจะดำเนินการเปรียบเทียบและรวมออนโทโลจีตามขั้นตอนวิธีที่ได้อธิบายในหัวข้อที่ 4.1 และ 4.2 ตามลำดับ แล้วส่งกลับผลลัพธ์ในรูปของไฟล์ออนโทโลจีรวมพื้นฐานชื่อ IntegratedOntology.owl และไฟล์ผลการรวมออนโทโลจีชื่อ IntegratedOntology.txt เมื่อขั้นตอนดังกล่าวเสร็จสมบูรณ์ดังรูป



รูปที่ 5.17 หน้าจอโปรแกรมการรวมออนโทโลจีแสดงผลลัพธ์หลังการรวมออนโทโลจีเสร็จสิ้น

ตัวอย่างผลลัพธ์ที่ได้จากโปรแกรมการรวมออนโทโลจีในไฟล์ IntegratedOntology.owl และ IntegratedOntology.txt แสดงไว้ดังรูปที่ 5.18 และ 5.19 ตามลำดับ

```
<owl:Class rdf:ID="Student">              // สร้างคลาสของคลาส Student
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    equivalent to local1.Student</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="GradStudent">          // สร้างคลาสของคลาส GradStudent
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    equivalent to local2.GradStudent</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Student"/>
</owl:Class>
<owl:Class rdf:ID="Faculty">              // สร้างคลาสของคลาส Faculty
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    equivalent to local1.Faculty</rdfs:comment>
```

รูปที่ 5.18 บางส่วนของออนโทโลจีรวมเบื้องต้นจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.owl

```
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#Class"/>
</owl:Class>
<owl:Class rdf:ID="StudentFaculty">      // สร้างคลาสของแอททริบิวต์ StudentFaculty
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="StudentGpa">          // สร้างคลาสของแอททริบิวต์ StudentGpa
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="GradStudentThesis">  // สร้างคลาสของแอททริบิวต์ GradStudentThesis
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="GradStudentLabName">// สร้างคลาสของแอททริบิวต์ GradStudentLabName
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#StudentFacultySemantic">       // สร้างคลาสของชื่อความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:about="#StudentMajorSemantic">         // สร้างคลาสของชื่อความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:about="#GradStudentFacultySemantic">   // สร้างคลาสของชื่อความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:about="#GradStudentMajorSemantic">     // สร้างคลาสของชื่อความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="Science">                          // สร้างคลาสของค่าความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentFacultySemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentFacultySemantic"/>
  </rdfs:subClassOf>
</owl:Class>
```

รูปที่ 5.18 บางส่วนของออนโทโลจีรวมเบื้องต้นจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.owl (ต่อ)

```
<owl:Class rdf:ID="Account">                                  // สร้างคลาสของค่าความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentFacultySemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentFacultySemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Engineer">                                 // สร้างคลาสของค่าความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentFacultySemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentFacultySemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Computer">                                 // สร้างคลาสของค่าความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentMajorSemantic"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentMajorSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Stat">                                     // สร้างคลาสของค่าความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentMajorSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Math">                                     // สร้างคลาสของค่าความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentMajorSemantic"/>
  </rdfs:subClassOf>
```

รูปที่ 5.18 บางส่วนของออนโทโลจีรวมเบื้องต้นจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.owl (ต่อ)

```
  <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentMajorSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Electronics">                          // สร้างคลาสของค่าความหมาย
  <rdfs:subClassOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentMajorSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
// สร้างพรอพเพอร์ตีจากคลาสของคลาสไปยังคลาสของแอททริบิวต์
<owl:ObjectProperty rdf:ID="hasStudentGpa">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#StudentGpa"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentFaculty">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#StudentFaculty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentThesis">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#GradStudent"/>
  <rdfs:range rdf:resource=#GradStudentThesis"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentLabName">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#GradStudent"/>
  <rdfs:range rdf:resource="#GradStudentLabName"/>
</owl:ObjectProperty>
// สร้างพรอพเพอร์ตีจากคลาสของคลาสไปยังคลาสของชื่อความหมาย
<owl:ObjectProperty rdf:ID="hasStudentSemantic">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range>
```

รูปที่ 5.18 บางส่วนของออนโทโลจีรวมเบื้องต้นจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.owl (ต่อ)

```
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#StudentMajorSemantic"/>
        <owl:Class rdf:about="#StudentFacultySemantic"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentSemantic">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:domain rdf:resource="#GradStudent"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#GradStudentFacultySemantic"/>
        <owl:Class rdf:about="#GradStudentMajorSemantic"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
// สร้างพรอพเพอร์ตีจากคลาสของแอททริบิวต์ไปยังคลาสของคลาสเพื่อกำหนดชนิดของข้อมูลเป็นคลาสของคลาส
<owl:ObjectProperty rdf:ID="hasStudentFacultyAltType">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassAttributeAltType"/>
    <rdfs:domain rdf:resource="#StudentFaculty"/>
    <rdfs:range rdf:resource="#Faculty"/>
  </owl:ObjectProperty>
// สร้างพรอพเพอร์ตีจากคลาสของแอททริบิวต์ไปยังชนิดข้อมูลพื้นฐาน
<owl:DatatypeProperty rdf:ID="hasStudentGpaType">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#StudentGpa"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasGradStudentThesisType">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#GradStudentThesis"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasGradStudentLabNameType">
  <rdfs:subPropertyOf rdf:resource=
    "http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#GradStudentLabName"/>
```

รูปที่ 5.18 บางส่วนของออนโทโลจีรวมเบื้องต้นจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.owl (ต่อ)

```
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
...
```

รูปที่ 5.18 บางส่วนของออนโทโลจีรวมเบื้องต้นจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.owl (ต่อ)

```
[Class]: Student (equivalent to Local1.Student)       << คลาสของคลาส Student
<subClassOf> Person, upper.owl#Class
<hasClassAttribute>                                   << ส่วนคลาสของแอททริบิวต์
  [ClassAttribute]: StudentFaculty
  (equivalent to Local1.Student.StudentFaculty)
  <hasClassAttributeAltType> Faculty
    Type conversion: no
  <hasClassAttributeSemantic> none
  [ClassAttribute]: StudentGpa (equivalent to Local1.Student.StudentGpa,
  Local2.GradStudent.GradStudentGpa)
  <hasClassAttributeType> String
    Type conversion: no
  <hasClassAttributeSemantic> none
<hasClassIntegrityConstraint> none                    << ส่วนคลาสของข้อกำหนดบูรณภาพ
<hasClassSemantic>                                    << ส่วนคลาสของความหมาย
  [SemanticName]: StudentFacultySemantic
    [SemanticWord]: Science, Account, Engineer
  [SemanticName]: StudentMajorSemantic
    [SemanticWord]: Computer, Stat, Math, Electronics
[Class]: GradStudent (equivalent to Local2.GradStudent)
                                                      << คลาสของคลาส GradStudent
<subClassOf> Student, Person, upper.owl#Class
<hasClassAttribute>                                   << ส่วนคลาสของแอททริบิวต์
  [ClassAttribute]: GradStudentThesis
  (equivalent to Local2.GradStudent.GradStudentThesis)
  <hasClassAttributeType> String
    Type conversion: no
  <hasClassAttributeSemantic> none
  [ClassAttribute]: GradStudentLabName
  (equivalent to Local2.GradStudent.GradStudentLabName)
  <hasClassAttributeAltType> String
    Type conversion: no
  <hasClassAttributeSemantic> none
<hasClassIntegrityConstraint> none                    << ส่วนคลาสของข้อกำหนดบูรณภาพ
<hasClassSemantic>                                    << ส่วนคลาสของความหมาย
  [SemanticName]: GradStudentFacultySemantic
    [SemanticWord]: Science, Account, Engineer
```

รูปที่ 5.19 บางส่วนของผลการรวมออนโทโลจีจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.txt

```
  [SemanticName]: GradStudentMajorSemantic
    [SemanticWord]: Computer, Stat, Math, Electronics
[Class]: Faculty (equivalent to Local1.Faculty)  << บางส่วนของคลาสของคลาส Faculty
<subClassOf> upper.owl#Class
...
```

รูปที่ 5.19 บางส่วนของผลการรวมออนโทโลจีจากการรวมระหว่างคลาสของคลาส Student และคลาสของคลาส GradStudent ในโลคัลออนโทโลจีที่1 และ 2 ในไฟล์ IntegratedOntoloty.txt (ต่อ)

5.2.2.2 การอนุมานออนโทโลจีรวมพื้นฐานเพื่อให้ได้ออนโทโลจีรวมที่สมบูรณ์

จากไฟล์ออนโทโลจีรวมพื้นฐานที่ได้จากขั้นตอนที่ 5.2.2.1 ขั้นตอนนี้จะเปิดให้ผู้ทำการรวมแบบจำลองข้อมูลแก้ไขบางส่วนของออนโทโลจีรวมพื้นฐาน หรือเพิ่มกฎตามความหมายและข้อกำหนดบูรณภาพของแบบจำลองข้อมูลแต่ละชุด เพื่อสร้างข้อเท็จจริงใหม่เกี่ยวกับข้อมูลลงในออนโทโลจีรวมพื้นฐานได้ตามต้องการ ก่อนจะนำมาใช้เป็นออนโทโลจีรวมสำหรับส่วนการสร้างแมปปิงไปยังแบบจำลองข้อมูลเชิงวัตถุในหัวข้อที่ 5.2.3 โดยพิจารณาจากไฟล์ผลการรวมออนโทโลจีร่วมกับความเข้าใจเกี่ยวกับข้อมูลในแบบจำลองข้อมูลแต่ละชุดที่นำมารวม และอาศัยความสามารถของออนโทโลจีที่อยู่ในรูปของภาษาอาวล์ที่อนุญาตให้ผู้ใช้เพิ่มกฎใดๆ เพื่ออนุมานออนโทโลจี ซึ่งจะเป็นประโยชน์ต่อการนำออนโทโลจีรวมที่ได้ไปใช้ในการสอบถามข้อมูลในขั้นตอนต่อไป

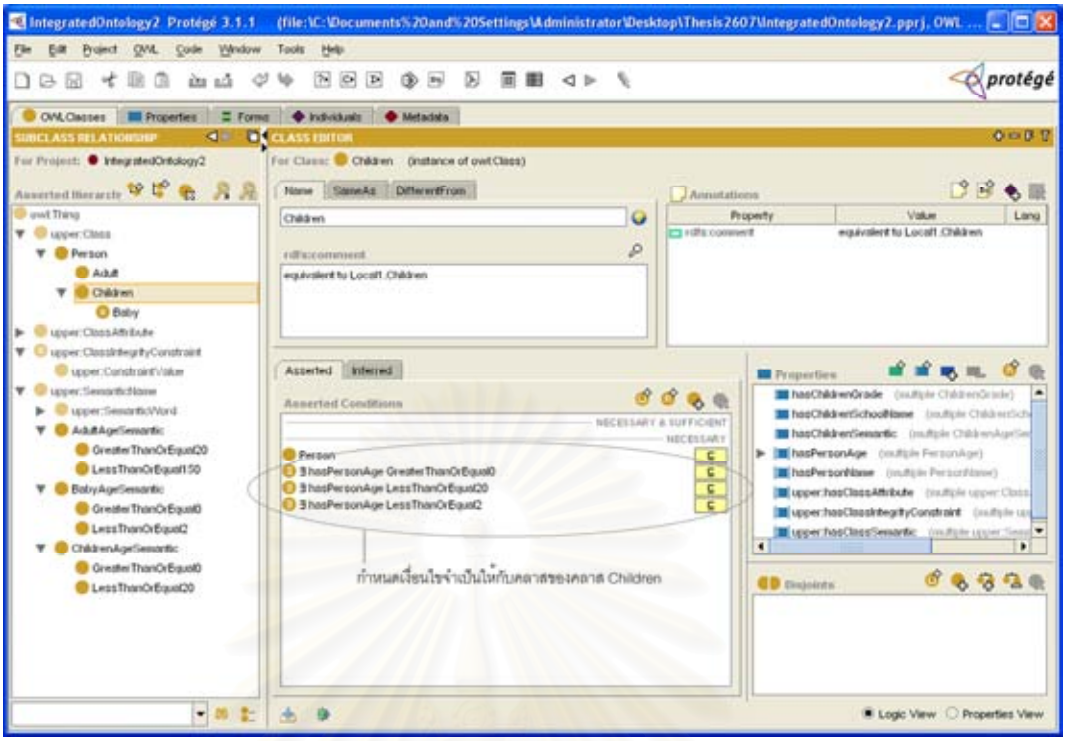
จุดประสงค์หลักของการอนุมานออนโทโลจีในขั้นตอนนี้คือ เพื่อให้โปรแกรมตรวจสอบและจำแนกการแบ่งประเภทของออนโทโลจีรวมตามโครงสร้างและเงื่อนไขที่กำหนดก่อนจะนำไปใช้งาน รวมทั้งอาจช่วยแก้จุดบกพร่อง (Debug) ของออนโทโลจีรวมที่สร้างขึ้น [41] โดยอาศัยการทำงานของโปรแกรมโพรเทเจในการแสดงผลของออนโทโลจีรวมที่ได้ และการระบุกฎในรูปของเงื่อนไขของคลาสลงในคลาสที่ต้องการเพิ่มเติม ร่วมกับโปรแกรมเรเซอร์โปร (RacerPro) เวอร์ชัน 1.9 [42] ซึ่งเรียกใช้งานได้จากหน้าจอโปรแกรมโพรเทเจในการอนุมานออนโทโลจีหลังจากการระบุเงื่อนไข ทั้งนี้การอนุมานจะตรวจสอบความเป็นไปได้ของเงื่อนไขดังกล่าว แล้วแสดงผลข้อเท็จจริงใหม่ที่ได้เพิ่มขึ้นให้กับผู้ทำการรวมแบบจำลองข้อมูลไว้ภายในออนโทโลจีรวมที่ได้ เพื่อให้ผู้ทำการรวมแบบจำลองข้อมูลนำออนโทโลจีรวมดังกล่าวไปใช้ในขั้นตอนต่อไป

ประโยชน์จากการอนุมานออนโทโลจีรวมนอกเหนือจากความถูกต้องของออนโทโลจีรวมที่ได้รับ คือเมื่อนำออนโทโลจีรวมที่ได้ไปใช้เป็นตัวกลางในการสอบถามข้อมูลจะทำให้ได้ผลลัพธ์ของการสอบถามเพิ่มขึ้น ซึ่งเป็นผลจากข้อเท็จจริงที่ได้เพิ่มจากการอนุมาน ตัวอย่างเช่น จากงานวิจัย [43] ออนโทโลจีรวมที่ได้ประกอบด้วยคลาสของคลาส Children ซึ่งมีคลาสของชื่อความหมายเกี่ยวกับอายุเป็น ChildrenAgeSemantic (มีคลาสของค่าความหมายเป็น “GreaterThanOrEqual0” และ “LessThanOrEqual20” ซึ่งหมายความว่ามีอายุอยู่ในช่วง 0 ถึง 20 ปี) และมีสับคลาสเป็นคลาสของคลาส Baby ซึ่งมีคลาสของชื่อความหมายเกี่ยวกับอายุเป็น BabyAgeSemantic (มีคลาสของค่าความหมายเป็น “GreaterThanOrEqual0” และ

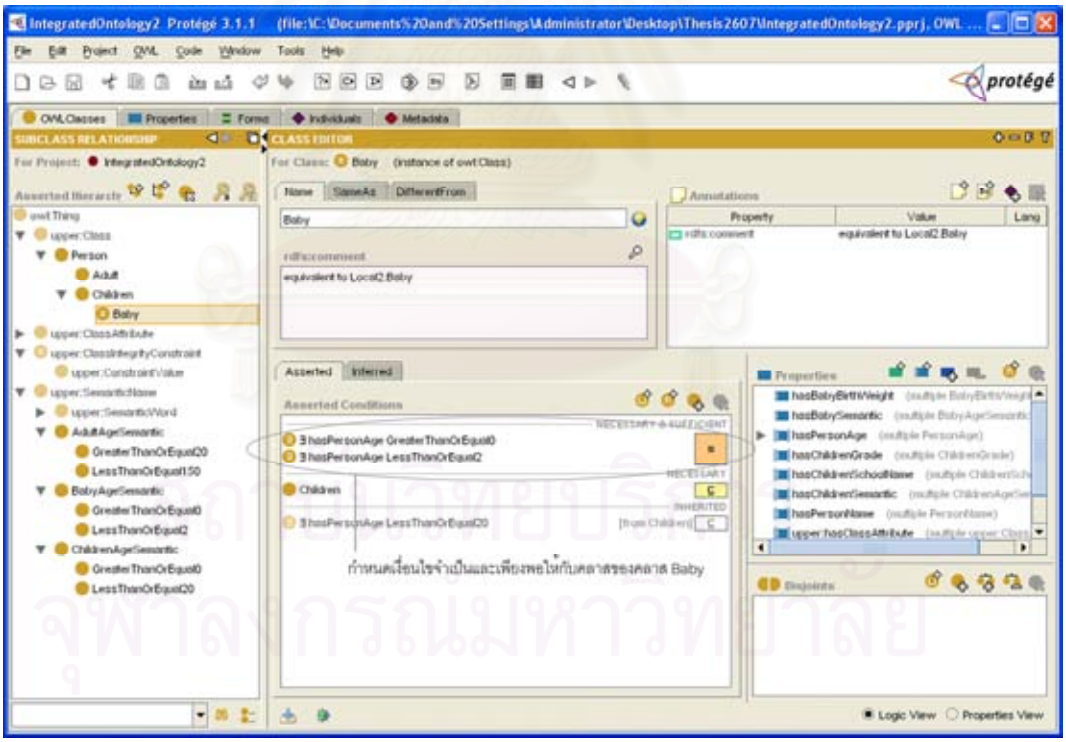
“LessThanOrEqual2” ซึ่งหมายความว่ามีอายุอยู่ในช่วง 0 ถึง 2 ปี) ตามลำดับ เมื่อพิจารณาจากความหมายของคลาสของคลาสคู่นี้จะเห็นได้ว่าความหมายของคลาสของคลาส Baby เป็นส่วนหนึ่งของความหมายของคลาสของคลาส Children จึงอาจสรุปได้ว่า หากอินสแตนซ์ใดของคลาสของคลาส Children มีค่าอายุอยู่ในช่วง 0 ถึง 2 ปี อินสแตนซ์ดังกล่าวควรปรากฏอยู่ในผลลัพธ์จากการสอบถามข้อมูลที่คลาสของคลาส Baby ด้วย

แต่เนื่องจากความสัมพันธ์แบบซูเปอร์คลาส/ สับคลาสระหว่างคลาสทั้งคู่ การนำออนโทโลจีรวมที่ได้ไปใช้ในการสอบถามข้อมูลทันทีโดยไม่ผ่านขั้นตอนการอนุมานนี้จะทำให้ผลลัพธ์ดังกล่าวไม่ปรากฏ ดังนั้นหากผู้ทำการรวมแบบจำลองข้อมูลต้องการได้ผลลัพธ์จากการสอบถามเพิ่มขึ้นตามความหมายที่ได้กำหนดให้กับคลาส ผู้ทำการรวมแบบจำลองข้อมูลจำเป็นต้องกำหนดเงื่อนไขเพิ่มเติมลงในคลาสของคลาสทั้งคู่ภายในออนโทโลจีรวมและอนุมานออนโทโลจีดังกล่าว ซึ่งมีขั้นตอนดังต่อไปนี้

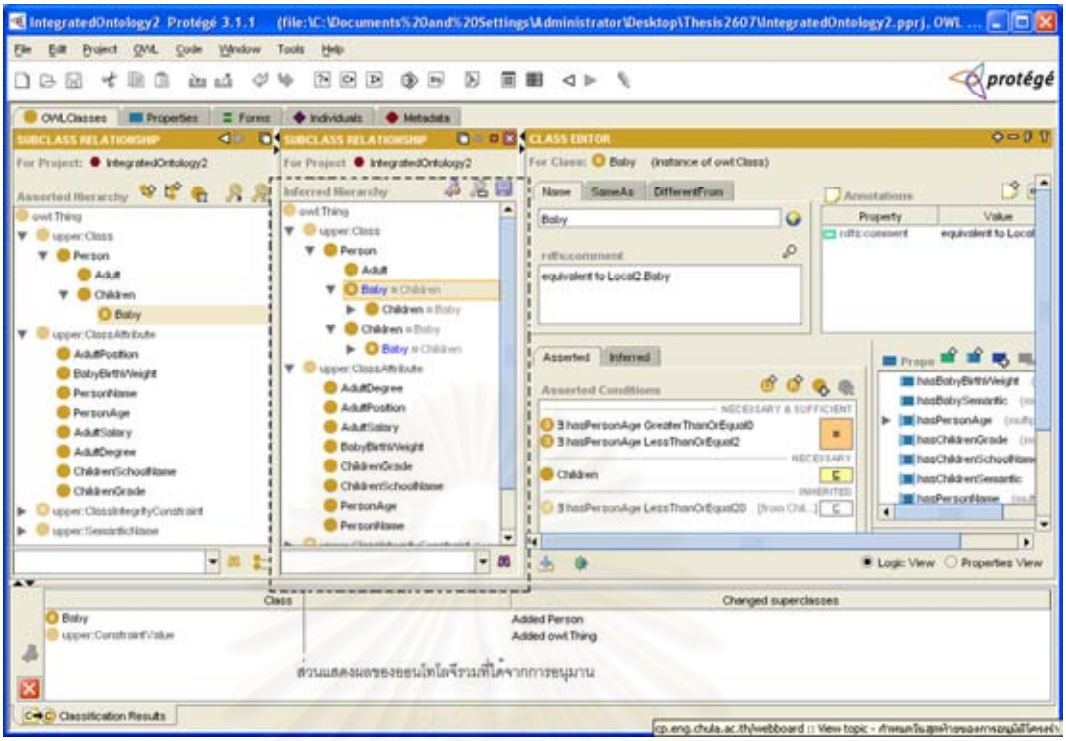
1. เปิดโปรแกรมโพรเทเจและสร้างออนโทโลจีรวมขึ้นจากไฟล์ผลการรวมออนโทโลจีโดยใช้วิธีการเดียวกับการสร้างโลคัลออนโทโลจีในขั้นตอนที่ 5.2.1.1
2. ปรับแต่งแก้ไขออนโทโลจีรวม เช่น เปลี่ยนชื่อคลาสของคลาส หรือคลาสของแอททริบิวต์ที่ปรากฏในออนโทโลจีรวมตามต้องการ
3. จากรูปที่ 5.20 หลังจากได้ออนโทโลจีรวมตามที่ต้องการแล้ว กำหนดเงื่อนไขจำเป็นให้กับคลาสของคลาส Children (คลาสที่เป็นซูเปอร์คลาส) โดยกำหนดให้มีค่าเป็นคลาสของค่าความหมายทั้งหมดที่เป็นของคลาสของคลาส Children และคลาสของคลาส Baby
4. จากรูปที่ 5.21 กำหนดเงื่อนไขจำเป็นและเพียงพอให้กับคลาสของคลาส Baby (คลาสที่เป็นสับคลาส) โดยกำหนดให้มีค่าเป็นคลาสของค่าความหมายทั้งหมดของคลาสของคลาส Baby เท่านั้น
5. เปิดโปรแกรมเรเซอร์โปร
6. ที่โปรแกรมโพรเทเจ เริ่มการอนุมานโดยเลือกคำสั่ง OWL >> Classify taxonomy จากเมนูด้านบนของโปรแกรม
7. จากรูปที่ 5.22 โปรแกรมเรเซอร์โปรจะทำหน้าที่ตรวจสอบและจำแนกการแบ่งประเภทของออนโทโลจีรวมที่สร้างขึ้นในโปรแกรมโพรเทเจ แล้วแสดงความสัมพันธ์ของออนโทโลจีรวมที่ได้จากการอนุมานลงในหน้าต่างที่ปรากฏเพิ่มของโปรแกรมโพรเทเจ
8. เลือกเก็บบันทึกไฟล์ออนโทโลจีรวมที่ได้เพื่อนำไปใช้ต่อด้วยวิธีการเดียวกับการเก็บบันทึกไฟล์โลคัลออนโทโลจีในหัวข้อที่ 5.2.1.1



รูปที่ 5.20 หน้าจอโปรแกรมโพรเทเจเพื่อกำหนดเงื่อนไขจำเป็นให้กับคลาส



รูปที่ 5.21 หน้าจอโปรแกรมโพรเทเจเพื่อกำหนดเงื่อนไขจำเป็นและเพียงพอให้กับคลาส



รูปที่ 5.22 หน้าจอโปรแกรมโพรเทเจหลังการอนุมานออนโทโลจีรวม

ผลจากการเพิ่มเงื่อนไขให้กับคลาสของคลาส Children และคลาสของคลาส Baby ลงในโปรแกรมโพรเทเจ จะทำให้เงื่อนไขดังกล่าวปรากฏอยู่ในคลาสทั้งคู่ภายในออนโทโลจีรวมดังแสดงในรูปที่ 5.23 โดยเงื่อนไขเหล่านี้จะถูกนำไปพิจารณาสร้างเป็นเงื่อนไขในส่วนแมปปิงที่สร้างระหว่างออนโทโลจีรวมกับแบบจำลองข้อมูลเชิงวัตถุในหัวข้อที่ 5.3 ต่อไป

```
<owl:Class rdf:ID="Children">                          // คลาสของคลาส Children
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    equivalent to Local1.Children</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="#GreaterThanOrEqual0"/>     // เงื่อนไข 1
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasPersonAge"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="#LessThanOrEqual2"/>        // เงื่อนไข 2
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasPersonAge"/>
```

รูปที่ 5.23 บางส่วนของออนโทโลจีรวมจากการรวมระหว่างคลาสของคลาส Children และคลาสของคลาส Baby ในโลคัลออนโทโลจีที่1 และ 2 [43] หลังการกำหนดเงื่อนไขและการอนุมาน

```
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="# LessThanOrEqual20"/>     // เงื่อนไข 3
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasPersonAge"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Baby">                          //   คลาสของคลาส Baby
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    equivalent to Local2.Baby</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Children"/>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:someValuesFrom rdf:resource="#GreaterThanOrEqual0"/> // เงื่อนไข 1
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#hasPersonAge"/>
          </owl:onProperty>
        </owl:Restriction>
        <owl:Restriction>
          <owl:someValuesFrom rdf:resource="#LessThanOrEqual2"/>   // เงื่อนไข 2
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#hasPersonAge"/>
          </owl:onProperty>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

รูปที่ 5.23 บางส่วนของออนโทโลจีรวมจากการรวมระหว่างคลาสของคลาส Children และคลาสของคลาส Baby ในโลคัลออนโทโลจีที่1 และ 2 [43] หลังการกำหนดเงื่อนไขและการอนุมาน (ต่อ)

ทั้งนี้ วิทยานิพนธ์ฉบับนี้ได้กำหนดหลักเบื้องต้นในการระบุเงื่อนไขของคลาสเพิ่มเติมไว้ว่า หากคลาสของคลาส A จากโลคัลออนโทโลจีชุดที่หนึ่งถูกรวมกับคลาสของคลาส B จากโลคัลออนโทโลจีชุดที่สองในออนโทโลจีรวมด้วยความสัมพันธ์แบบซูเปอร์คลาส (คลาสของคลาส A เป็นซูเปอร์คลาสของคลาส B ในโลคัลออนโทโลจีรวม) และคลาสของความหมายของคลาสของคลาส B เป็นสับเซตของคลาสของความหมายของคลาสของคลาส A ให้กำหนดเงื่อนไขจำเป็นให้กับคลาสของคลาส A เป็นคลาสของค่าความหมายทั้งหมดของทั้งคลาสของคลาส A และคลาสของคลาส B

พร้อมกับกำหนดเงื่อนไขจำเป็นและเพียงพอให้กับคลาสของคลาส B เป็นคลาสของค่าความหมายทั้งหมดของคลาสของคลาส B เท่านั้น

อย่างไรก็ตาม หากผู้ทำการรวมแบบจำลองข้อมูลพิจารณาแล้วพบว่า ออนโทโลจีรวมพื้นฐานที่ได้เพียงพอและตรงตามความต้องการที่จะนำไปใช้เป็นออนโทโลจีรวม ผู้ทำการรวมแบบจำลองข้อมูลอาจข้ามการอนุมานในขั้นตอนนี้ไปและนำออนโทโลจีรวมพื้นฐานที่ได้จากขั้นตอนที่ 5.2.2.1 ไปใช้เป็นออนโทโลจีรวมในส่วนการสร้างแมปปิงระหว่างออนโทโลจีรวมกับแบบจำลองข้อมูลเชิงวัตถุในหัวข้อที่ 5.2.3 ทั้งนี้ หากนำไฟล์ออนโทโลจีรวมพื้นฐานไปใช้เป็นออนโทโลจีรวมโดยมิได้ปรับแต่งใดๆ ผลการรวมแบบจำลองข้อมูลที่ได้จะใกล้เคียงกับผลที่ได้จากการรวมแบบจำลองข้อมูลเชิงวัตถุในงานวิจัย [20] นอกจากนั้น ในการกำหนดเงื่อนไขใดๆ เพื่อการอนุมานออนโทโลจี ผู้ทำการรวมแบบจำลองข้อมูลจะต้องคำนึงอยู่เสมอว่า เงื่อนไขดังกล่าวจะต้องไม่ขัดแย้งกับข้อกำหนดบูรณภาพของแบบจำลองข้อมูลเดิม

5.2.3 ส่วนการสร้างแมปปิงระหว่างออนโทโลจีรวมกับแบบจำลองข้อมูลเชิงวัตถุ

เฟรมเวิร์คส่วนการสร้างแมปปิงระหว่างออนโทโลจีรวมกับแบบจำลองข้อมูลเชิงวัตถุเป็นขั้นตอนการนำออนโทโลจีรวมที่ได้จากขั้นตอนก่อนหน้านี้มากำกับลงบนแบบจำลองข้อมูลแต่ละชุด (ขั้นตอนที่ 6 ในรูปที่ 5.1) การกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูลเชิงวัตถุนี้มีลักษณะเช่นเดียวกับแนวคิดแบบออนโทโลจีเดียว ซึ่งจะช่วยลดความยุ่งยากซับซ้อนในส่วนการสอบถามข้อมูลจากฐานข้อมูลเชิงวัตถุทั้งสองชุดผ่านออนโทโลจีรวม เนื่องจากข้อคำถามจากผู้ใช้จะถูกส่งจากออนโทโลจีที่ทำหน้าที่เป็นตัวกลางไปยังฐานข้อมูลเชิงวัตถุแต่ละชุดโดยตรง หลังจากนั้นแมปปิงที่สร้างขึ้นในขั้นตอนนี้จะทำหน้าที่แปลงข้อคำถามที่ได้รับดังกล่าวให้อยู่ในรูปของข้อคำถามที่ใช้งานกับฐานข้อมูลเชิงวัตถุชุดนั้น แล้วส่งผลลัพธ์ที่ได้ไปยังออนโทโลจีรวมเพื่อแสดงแก่ผู้ใช้ ดังนั้น หน้าที่ของแมปปิงที่สร้างขึ้นในส่วนนี้คือการแมปฟิลด์ต่างๆ ในแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดกับออนโทโลจีรวมที่ได้จากขั้นตอนที่ 5.2.2 รวมถึงจัดการกับความขัดแย้งต่างๆ ที่ปรากฏระหว่างแบบจำลองข้อมูลแต่ละชุดทั้งความขัดแย้งระดับข้อมูลและความขัดแย้งระดับสกีมา ทำให้สามารถแก้ไขปัญหาเรื่องความต่างแบบระหว่างข้อมูลแต่ละชุดแทนผู้ใช้ โดยอาศัยออนโทโลจีรวมเป็นตัวกลางการติดต่อไปยังแบบจำลองข้อมูลแต่ละชุด

ในการสร้างแมปปิงระหว่างออนโทโลจีรวมกับแบบจำลองข้อมูลเชิงวัตถุนั้น วิทยานิพนธ์ฉบับนี้กำหนดให้ผู้ทำการรวมแบบจำลองข้อมูลสร้างแมปปิงในรูปของวิว (View) [1] เพื่อแมประหว่างคลาสของคลาสและคลาสของแอททริบิวต์ในออนโทโลจีรวมไปยังฟิลด์ต่างๆ ภายในฐานข้อมูลเชิงวัตถุ โดยโครงสร้างของวิวที่ใช้กำกับลงบนฐานข้อมูลเชิงวัตถุทั้งสองชุดจะถูกกำหนดขึ้นตามออนโทโลจีรวมที่ได้จากส่วนการสร้างออนโทโลจีรวม เพื่อให้สัมพันธ์กับแผ่นแบบการสอบถามข้อมูลที่ใช้เป็นตัวกลางในการติดต่อระหว่างผู้ใช้งานกับฐานข้อมูลเชิงวัตถุแต่ละชุดที่จะพัฒนาขึ้นต่อไป

จากออนโทโลจีรวมที่สร้างขึ้นจากการรวมโลคัลออนโทโลจีที่อธิบายแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุด ขั้นตอนนี้จะนำข้อมูลจากออนโทโลจีรวมดังกล่าวมาสร้างเป็นวิวที่กำกับลงบนฐานข้อมูลเชิงวัตถุแต่ละชุด เพื่อใช้รองรับข้อคำถามที่เกิดจากการสอบถามข้อมูลผ่านออนโทโลจีรวม โดยอาศัยความสามารถของระบบจัดการฐานข้อมูล (Databse Management System: DBMS) ของฐานข้อมูลแต่ละชุดที่เปิดให้ผู้ใช้งานสามารถสร้างชั้นของวิวเพื่อรองรับการเลือกสอบถามข้อมูลเฉพาะฟิลด์ที่ต้องการได้ ทั้งนี้ วิวที่สร้างขึ้นแต่ละชุดจะถูกกำกับลงบนฐานข้อมูลเชิงวัตถุทั้งสองชุดในรูปแบบเดียวกัน ซึ่งเป็นไปตามแนวคิดแบบรวมหลายออนโทโลจีของผู้วิจัยที่ต้องการนำออนโทโลจีรวมที่ได้เพียงชุดเดียวมาใช้งานในการสอบถามข้อมูลจากระบบการรวมแบบจำลองข้อมูลเชิงวัตถุที่พัฒนาขึ้น

การสร้างวิวเพื่อนำไปกำกับลงบนฐานข้อมูลเชิงวัตถุแต่ละชุดจะกำหนดขึ้นจากออนโทโลจีรวม โดยมีวิธีการดังนี้

1. สร้างฟังก์ชันเพื่อจัดการกับความขัดแย้งของค่าข้อมูล เช่น ฟังก์ชันการแปลงค่าจาก “F” เป็น “female” และ “M” เป็น “male” (รูปที่ 5.24)
2. สร้างฟังก์ชันเพื่อจัดการกับความขัดแย้งของหน่วยข้อมูล เช่น ฟังก์ชันการแปลงค่าข้อมูลจากหน่วยเป็นเซนติเมตรเป็นหน่วยเมตร (รูปที่ 5.25)

```
CREATE FUNCTION Sex_to_Gender (old_sex in VARCHAR2)
RETURN VARCHAR2
IS
new_sex VARCHAR2(50);
BEGIN
IF (old_sex='F') THEN
new_sex:='female';
ELSIF (old_sex='M') THEN
new_sex:='male';
END IF;
RETURN new_sex;
END;
```

รูปที่ 5.24 ตัวอย่างฟังก์ชันการแปลงค่าจาก “F” เป็น “female” และ “M” เป็น “male”

```
CREATE FUNCTION Height_to_Metre (old_height in NUMBER)
RETURN NUMBER
IS
new_height NUMBER;
BEGIN
new_height:= old_height /100;
RETURN new_height;
END;
```

รูปที่ 5.25 ตัวอย่างฟังก์ชันการแปลงค่าข้อมูลจากหน่วยเซนติเมตรเป็นหน่วยเมตร

3. สร้างฟังก์ชันเพื่อจัดการกับความขัดแย้งของโครงสร้าง เช่น ฟังก์ชันการแปลงชนิดของข้อมูลที่เก็บให้ตรงกัน
4. สร้างวิวหนึ่งชุดจากคลาสของคลาสหนึ่งคลาสในออนโทโลจีรวม แล้วกำกับลงบนฐานข้อมูลทั้งสองชุด ซึ่งวิวที่สร้างประกอบด้วยรายละเอียดดังนี้
   - ชื่อวิว (View Name) ให้กำหนดตามชื่อคลาสของคลาสในออนโทโลจีรวม และแมปเป็นชื่อของคลาสที่ใช้ในแบบจำลองข้อมูลแต่ละชุด เพื่อจัดการกับความขัดแย้งของชื่อ
   - รายชื่อแอททริบิวต์ของวิว (Attribute List) ให้กำหนดรายชื่อแอททริบิวต์ของวิวตามกลุ่มของคลาสของแอททริบิวต์ภายในคลาสของคลาสนั้น เพื่อจัดการกับความขัดแย้งของชื่อ
   - ข้อคำถาม (Query) ให้กำหนดตามแอททริบิวต์ของแต่ละคลาสที่แมปได้กับแต่ละฟิลด์ของรายชื่อแอททริบิวต์ หากฟิลด์ใดไม่ปรากฏจริงในฐานข้อมูลให้แทนที่ด้วยสัญลักษณ์ “—“
   - ภายในข้อคำถาม หากฟิลด์ใดเกิดขึ้นจากคลาสของแอททริบิวต์ที่ได้รับการระบุว่าเป็นคำที่มีความหมายเหมือนกันให้แมปแอททริบิวต์ภายในฐานข้อมูลเชิงวัตถุให้ตรงกับชื่อที่เลือกในออนโทโลจีรวม เพื่อจัดการกับความขัดแย้งของชื่อ หรือความขัดแย้งของการแอกกรีเกชัน
   - ภายในข้อคำถาม หากฟิลด์ใดเกิดขึ้นจากคลาสของแอททริบิวต์ที่ได้รับการระบุว่าเป็นคำลูกกลุ่ม/ แม่กลุ่มกันให้แมปกลุ่มของแอททริบิวต์ภายในฐานข้อมูลเชิงวัตถุทั้งกลุ่มให้ตรงกับชื่อคำแม่กลุ่มที่เลือกในออนโทโลจีรวม โดยอาศัยฟังก์ชัน CONCAT [44] ของระบบจัดการฐานข้อมูล เพื่อจัดการกับความขัดแย้งของโครงสร้าง หรือความขัดแย้งของการแอกกรีเกชัน
5. หากคลาสของคลาสใดถูกกำหนดเงื่อนไขเพิ่มเติมจากขั้นตอนการอนุมาน ในส่วนของข้อคำถามให้เพิ่มเงื่อนไขในการสร้างวิวให้ตรงกับเงื่อนไขที่ปรากฏในออนโทโลจีรวม

ตัวอย่างวิวสำหรับกำกับลงบนฐานข้อมูลเชิงวัตถุชุดที่หนึ่งและชุดที่สองดังแสดงในรูปที่ 5.26 และ 5.27 ตามลำดับ ทำหน้าที่เป็นแมปปิงระหว่างคลาสของคลาส Student ในออนโทโลจีรวมกับแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุด (ในรูปที่ 3.4) โดยสร้างวิวขึ้นตามออนโทโลจีที่ได้ในรูปที่ 4.24 ส่วนตัวอย่างวิวที่สร้างจากคลาสของคลาส Baby ซึ่งได้รับการเพิ่มเงื่อนไขลงในข้อคำถามสำหรับกำกับลงบนฐานข้อมูลเชิงวัตถุชุดที่หนึ่งจากแบบจำลองข้อมูลในงานวิจัย [43] แสดงไว้ในรูปที่ 5.28

```
CREATE OR REPLACE VIEW Student1(name, gender, height, address, gpa,
faculty_name, faculty_address, faculty_dean)
AS
SELECT a.name, a.sex, a.height, a.address, a.gpa, a.faculty.name,
a.faculty.address, a.faculty.dean
FROM StudentTab a;
```

รูปที่ 5.26 ตัวอย่างวิว Student ที่กำกับลงบนฐานข้อมูลเชิงวัตถุชุดที่หนึ่ง

```
CREATE OR REPLACE VIEW Student2 (name, gender, height, address, gpa,
faculty_name, faculty_address, faculty_dean)
AS
SELECT a.person.name, a.person.gender, a.person.height, a.person.home_no ||
', ' || a.person.home_city, a.gpa, a.faculty_name, '-', '-'
FROM GradStudentTab a;
```

รูปที่ 5.27 ตัวอย่างวิว Student ที่กำกับลงบนฐานข้อมูลเชิงวัตถุชุดที่สอง

```
CREATE OR REPLACE VIEW Baby1 (name, age, grade, school_name, birth_weight)
AS
SELECT a.person.name, a.person.age, a.grade, a.school_name, '-'
FROM ChildrenTab1 a
WHERE a.person.age <= 2;
```

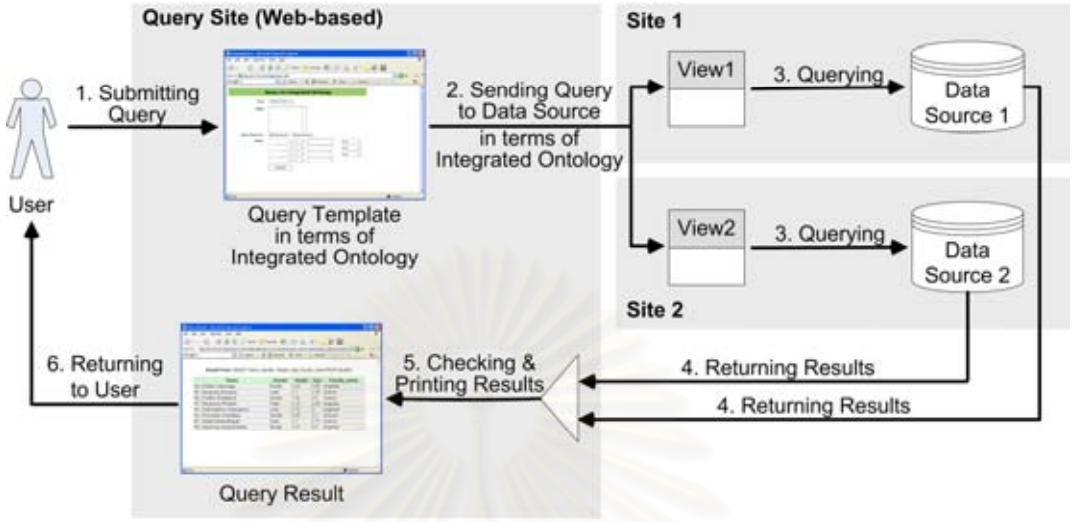
รูปที่ 5.28 ตัวอย่างวิว Baby แบบมีเงื่อนไขในข้อคำถามที่กำกับลงบนฐานข้อมูลเชิงวัตถุชุดที่หนึ่ง

5.3 แผ่นแบบการสอบถามข้อมูลผ่านการทำงานของเฟรมเวิร์ค

แผ่นแบบการสอบถามข้อมูลผ่านการทำงานของเฟรมเวิร์คซึ่งพัฒนาขึ้นตามแนวคิดแบบรวมหลายออนโทโลจีที่นำเสนอในวิทยานิพนธ์ฉบับนี้ เป็นแผ่นแบบการสอบถามข้อมูลอย่างง่ายที่ทำงานผ่านเว็บเพจ โดยใช้ภาษาพีเอชพี (PHP: Hypertext Preprocessor) [45] ในการโปรแกรมเพื่อจัดการกับข้อคำถาม การติดต่อกับฐานข้อมูล และการแสดงผลลัพธ์ โครงสร้างของแผ่นแบบได้รับการออกแบบให้สอดคล้องกับออนโทโลจีรวมที่ได้จากเฟรมเวิร์คส่วนการสร้างออนโทโลจีรวมในขั้นตอนที่ 5.2.2 เพื่อให้สามารถติดต่อและส่งข้อคำถามจากผู้ใช้ไปยังฐานข้อมูลแต่ละชุดได้โดยตรง นอกจากนี้ แผ่นแบบจะช่วยให้ผู้ใช้ทราบถึงภาพรวมทั้งหมดของฐานข้อมูลทั้งสองชุดว่าจะสามารถสอบถามข้อมูลใดได้บ้าง

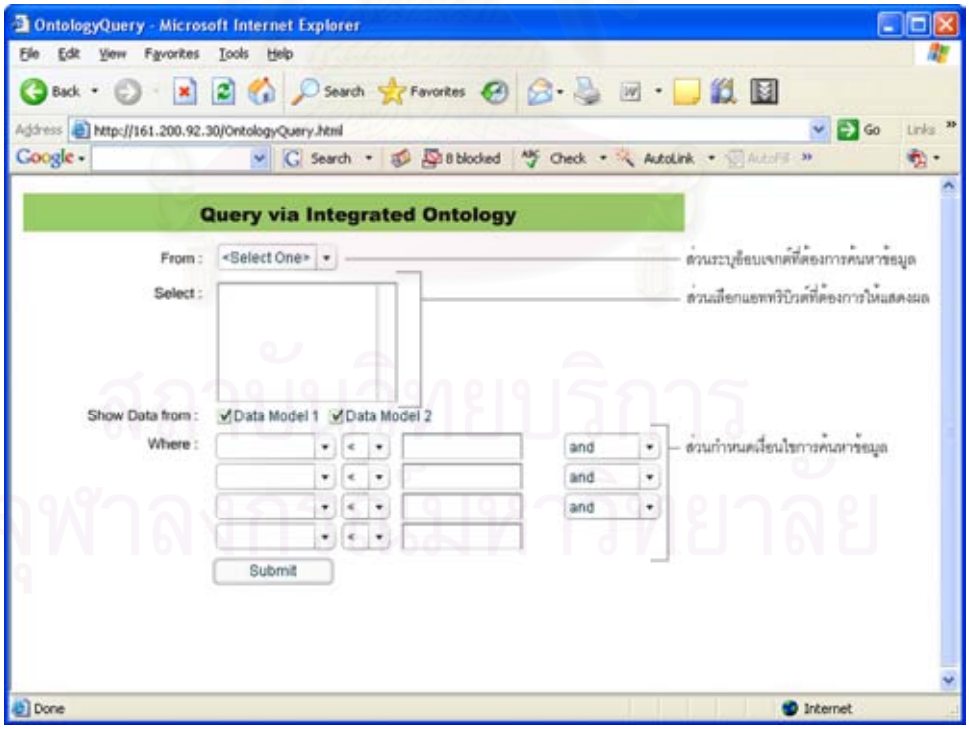
หลังจากผู้ใช้ระบุข้อคำถามลงในแผ่นแบบแล้ว โปรแกรมจะส่งข้อคำถามดังกล่าวต่อไปยังฐานข้อมูลทั้งสองชุดพร้อมกันเพื่อดำเนินการสอบถามข้อมูล โดยการสอบถามจะกระทำผ่านวิวซึ่งสร้างไว้ในขั้นตอนก่อนหน้านี้ ส่งผลให้สามารถแปลงข้อคำถามเป็นข้อคำถามที่ใช้จริงกับฐานข้อมูลได้ เมื่อฐานข้อมูลส่งกลับผลลัพธ์จากการสอบถาม โปรแกรมจะนำผลลัพธ์ที่ได้จากทั้งสองฐานข้อมูลมาตรวจสอบเพื่อตัดข้อมูลที่ซ้ำซ้อนระหว่างฐานข้อมูลออก ก่อนจะนำไปแสดงให้กับผู้ใช้ ด้วยวิธีการดังกล่าวนี้ ทำให้สามารถแก้ไขปัญหาเรื่องการกระจายตัวของข้อมูลแต่ละชุดแทนผู้ใช้ โดยอาศัยออน-

โทโลจีรวมเป็นตัวกลางในการติดต่อระหว่างผู้ใช้ไปยังฐานข้อมูลทุกชุด การทำงานของแผ่นแบบการสอบถามข้อมูลผ่านการทำงานของเฟรมเวิร์คนี้แสดงไว้ในรูปที่ 5.29



รูปที่ 5.29 การทำงานของแผ่นแบบการสอบถามข้อมูลผ่านการทำงานของเฟรมเวิร์ค

หน้าจอการทำงานของแผ่นแบบจะประกอบด้วยเมนูการระบุข้อคำถาม ได้แก่ ส่วนระบุอ็อบเจกต์ที่ต้องการสอบถามข้อมูล (From) ส่วนเลือกแอททริบิวต์ที่ต้องการให้แสดงผล (Select) และส่วนกำหนดเงื่อนไขการสอบถามข้อมูล (Where) ดังแสดงในรูปที่ 5.30

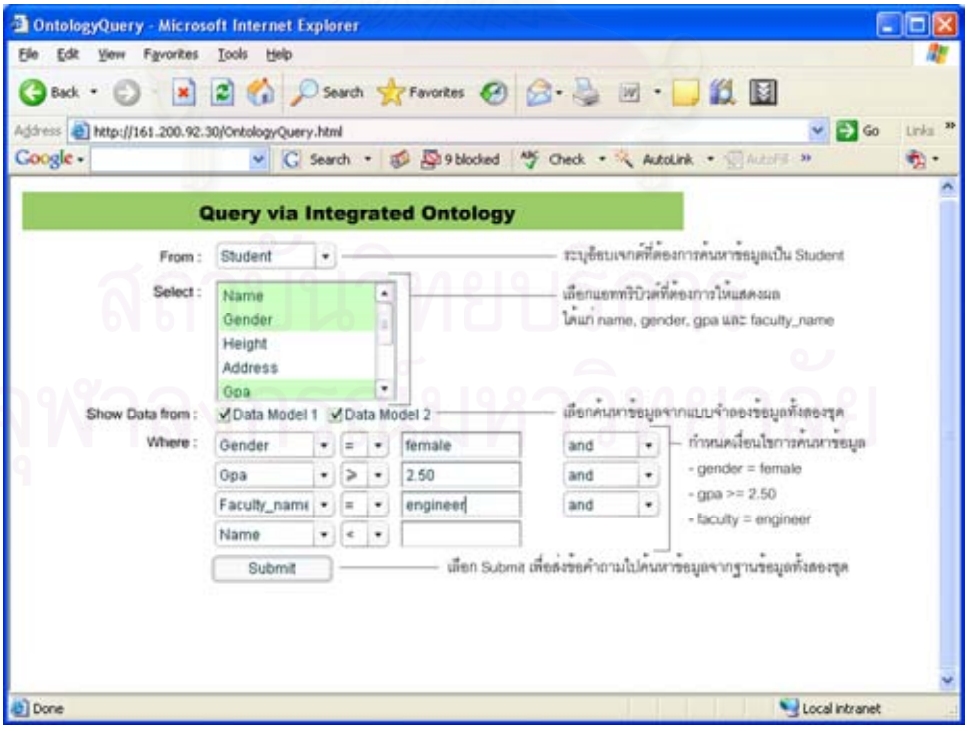


รูปที่ 5.30 หน้าจอการระบุข้อคำถามของแผ่นแบบการสอบถามข้อมูลผ่านออนโทโลจีรวม

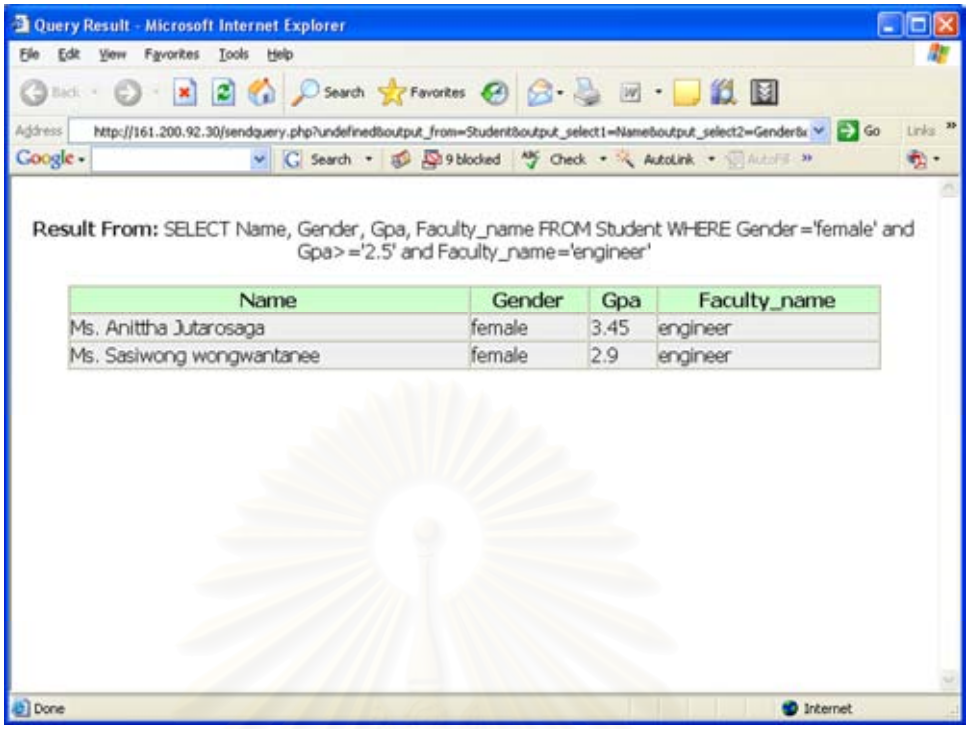
การสร้างแผ่นแบบการสอบถามข้อมูลเพื่อแสดงวิธีการสอบถามข้อมูลผ่านออนโทโลจีรวมในการทำงานของเฟรมเวิร์คที่พัฒนาขึ้นตามแนวคิดแบบรวมหลายออนโทโลจีนั้น เมนูต่างๆ ในแผ่นแบบจะถูกกำหนดขึ้นให้ผู้ใช้เลือกสอบถามข้อมูลตามออนโทโลจีรวมที่ได้จากขั้นตอนที่ 5.2.2 โดยมีวิธีการดังนี้

1. สร้างเมนูอ็อบเจกต์ที่ต้องการสอบถามข้อมูลจากแต่ละคลาสของคลาสในออนโทโลจีรวม
2. จากคลาสของคลาสที่สร้างขึ้นเป็นอ็อบเจกต์ที่ต้องการสอบถามข้อมูลแต่ละคลาส สร้างเมนูชุดของแอททริบิวต์ที่ต้องการให้แสดงผลตามกลุ่มของคลาสของแอททริบิวต์ภายในคลาสของคลาสนั้น
3. สร้างเมนูเงื่อนไขการสอบถามข้อมูลจากกลุ่มของคลาสของแอททริบิวต์ภายในคลาสของคลาสนั้น โดยให้ผู้ใช้สามารถกำหนดตัวดำเนินการ (Operator) ในการเปรียบเทียบได้เป็น $<, \leq, =, \geq$ และ $>$ พร้อมทั้งกำหนดค่าของเงื่อนไขที่เปรียบเทียบได้ตามที่ผู้ใช้ต้องการ
4. สร้างเมนูให้ผู้ใช้เลือกได้ว่า ต้องการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดใดบ้าง โดยกำหนดค่าเริ่มต้นให้สอบถามจากแบบจำลองข้อมูลทั้งสองชุด (Data Model 1 และ Data Model 2)

ตัวอย่างการระบุข้อคำถามลงในแผ่นแบบการสอบถามข้อมูลที่สร้างขึ้นตามออนโทโลจีรวมที่ได้จากผลการรวมออนโทโลจีในรูปที่ 4.24 เพื่อสอบถามข้อมูลจากแบบจำลองข้อมูลทั้งสองชุดในรูปที่ 3.4 แสดงได้ดังรูปที่ 5.31 โดยรูปที่ 5.32 แสดงตัวอย่างผลลัพธ์จากการสอบถามข้อมูลด้วยข้อคำถามดังกล่าว



รูปที่ 5.31 ตัวอย่างการระบุข้อคำถามลงในแผ่นแบบการสอบถามข้อมูลผ่านออนโทโลจีรวม



รูปที่ 5.32 ตัวอย่างผลลัพธ์จากการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามในรูปที่ 5.31

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6
การทดสอบการทำงานเพื่อประเมินความสามารถของเฟรมเวิร์ค

จุดประสงค์หลักของการทดสอบการทำงานเพื่อประเมินความสามารถของเฟรมเวิร์คในบทนี้คือ การทดสอบต้นแบบการทำงานของเฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีที่ได้รับการพัฒนาขึ้นตามแนวคิดแบบรวมหลายออนโทโลจีในบทที่ 5 ว่าสามารถทำงานได้จริงตรงตามที่ต้องการหรือไม่ วิทยานิพนธ์ฉบับนี้เลือกใช้การทดสอบในสองลักษณะคือ

1. การทดสอบผลการสร้างออนโทโลจีรวมที่สร้างขึ้นจากการรวมโลคัลออนโทโลจีสองชุดที่ใช้อธิบายแบบจำลองข้อมูลเชิงวัตถุที่ต้องการรวม เพื่อตรวจสอบความถูกต้องของวิธีการรวมออนโทโลจี โดยออนโทโลจีรวมที่ได้จะต้องตรงกับผลการรวมแบบจำลองข้อมูลเชิงวัตถุที่ได้จากการรวมตามขั้นตอนวิธีในงานวิจัย [20] หลังจากนั้น จะนำโปรแกรมการรวมออนโทโลจีไปใช้ทดสอบด้วยแบบจำลองข้อมูลที่มีความซับซ้อนมากขึ้นเพื่อหาข้อจำกัดของระบบ
2. การทดสอบความถูกต้องในการสอบถามข้อมูลผ่านระบบการรวมแบบจำลองข้อมูลเชิงวัตถุที่สร้างขึ้นตามแนวคิดแบบรวมหลายออนโทโลจี เพื่อตรวจสอบว่าข้อมูลที่ได้จากการสอบถามผ่านแผ่นแบบของระบบการรวมแบบจำลองข้อมูลถูกต้องและครบถ้วน เมื่อเปรียบเทียบกับการสอบถามโดยตรงที่ฐานข้อมูลแต่ละชุด ซึ่งจะแสดงให้เห็นว่าแนวคิดที่นำเสนอในวิทยานิพนธ์ฉบับนี้สามารถนำไปใช้งานได้จริงโดยไม่มีข้อมูลใดสูญหาย หรือทำให้ความหมายของข้อมูลผิดไปจากเดิม

6.1 การทดสอบผลการสร้างออนโทโลจีรวม

การทดสอบผลการสร้างออนโทโลจีรวมซึ่งสร้างขึ้นตามขั้นตอนวิธีในการรวมโลคัลออนโทโลจีที่นำเสนอในวิทยานิพนธ์ฉบับนี้มีวัตถุประสงค์ดังต่อไปนี้

- เพื่อตรวจสอบผลการแปลงโมเดลระหว่างออนโทโลจีและแบบจำลองข้อมูลเชิงวัตถุ โดยใช้การสร้างโลคัลออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุตามโครงสร้างของอัปเปอร์ออนโทโลจีที่ผู้วิจัยกำหนดขึ้น ดังนั้นหากการแปลงถูกต้อง โลคัลออนโทโลจีแต่ละชุดจะต้องแสดงแทนความหมายของแบบจำลองข้อมูลเชิงวัตถุได้เพียงพอต่อการรวม
- เพื่อตรวจสอบขั้นตอนวิธีการรวมโลคัลออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นว่ามีความถูกต้องและเหมาะสมสำหรับนำไปใช้เป็นขั้นตอนวิธีการรวมออนโทโลจีในการแสดงแทนแบบจำลองข้อมูลเชิงวัตถุซึ่งสร้างขึ้นตามโครงสร้างที่ผู้วิจัยกำหนดไว้
- เพื่อตรวจสอบความถูกต้องของโปรแกรมการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้น ทั้งในส่วนการเปรียบเทียบและส่วนการรวมออนโทโลจี

- เพื่อตรวจหาข้อจำกัดของขั้นตอนวิธีการรวมโลคัลออนโทโลจี และข้อจำกัดของโปรแกรมการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้น เมื่อนำไปใช้งานกับแบบจำลองข้อมูลจริงซึ่งมีขนาดใหญ่ และมีความซับซ้อน

ในการทดสอบผลการสร้างออนโทโลจีรวมที่สร้างขึ้นตามขั้นตอนวิธีในการรวมโลคัลออนโทโลจีที่นำเสนอในวิทยานิพนธ์ฉบับนี้นั้น ผู้วิจัยได้เลือกนำแบบจำลองข้อมูลเชิงวัตถุสองชุดจำนวน 13 คู่ แบ่งเป็น แบบจำลองข้อมูลเชิงวัตถุจากงานวิจัย [20] สิบคู่ และแบบจำลองข้อมูลเชิงวัตถุที่ประยุกต์จากแบบจำลองข้อมูลที่ออกแบบสำหรับใช้งานในโครงการระบบสารสนเทศเพื่อการบริหารของจุฬาลงกรณ์มหาวิทยาลัยจากงานวิจัย [22-25] อีกสามคู่มาทดลองสร้างออนโทโลจีรวมโดยอาศัยโปรแกรมต้นแบบที่พัฒนาขึ้นตามเฟรมเวิร์คสำหรับการรวมแบบจำลองข้อมูลเชิงวัตถุในหัวข้อที่ 5.2 แล้วพิจารณาเปรียบเทียบกับผลการรวมแบบจำลองข้อมูลเชิงวัตถุคู่นั้นๆ ในงานวิจัย [20] หรือผลการรวมจากผู้เชี่ยวชาญด้วยขั้นตอนวิธีเดียวกัน โดยผู้วิจัยได้แบ่งการทดสอบนี้ออกเป็นสองขั้นคือ การทดสอบด้วยแบบจำลองข้อมูลเชิงวัตถุจากงานวิจัย [20] และการทดสอบด้วยแบบจำลองข้อมูลเชิงวัตถุที่ประยุกต์จากแบบจำลองข้อมูลในงานวิจัย [22-25] ที่ออกแบบเพื่อใช้งานจริง

6.1.1 การทดสอบผลการสร้างออนโทโลจีรวมด้วยแบบจำลองข้อมูลเชิงวัตถุจากงานวิจัย [20]

ขั้นตอนการทดสอบ

1. เลือกแบบจำลองข้อมูลเชิงวัตถุที่ต้องการรวมหนึ่งคู่จากแบบจำลองข้อมูลตัวอย่างที่ใช้ในการทดสอบของงานวิจัย [20]
2. สร้างโลคัลออนโทโลจีเพื่ออธิบายและเพิ่มความความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดโดยใช้โปรแกรมโพรเทเจ ตามวิธีการในหัวข้อที่ 5.2.1.1
3. สร้างออนโทโลจีความหมายเพิ่มเติมที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดโดยใช้โปรแกรมโพรเทเจ ตามวิธีการในหัวข้อที่ 5.2.1.2
4. สร้างออนโทโลจีรวมพื้นฐานและไฟล์ผลการรวมออนโทโลจีจากโลคัลออนโทโลจีทั้งสองชุดและออนโทโลจีความหมายเพิ่มเติม โดยใช้โปรแกรมการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นตามวิธีการในหัวข้อที่ 5.2.2.1
5. เปรียบเทียบออนโทโลจีรวมพื้นฐานที่ได้ กับแบบจำลองข้อมูลรวมที่ได้จากการรวมแบบจำลองข้อมูลเชิงวัตถุคู่ที่เลือกในงานวิจัย [20]

ผลการทดสอบ

หลังจากนำแบบจำลองข้อมูลเชิงวัตถุทั้งสิบคู่จากแบบจำลองข้อมูลตัวอย่างที่ใช้ทดสอบในงานวิจัย [20] มาทดลองสร้างออนโทโลจีรวมตามขั้นตอนการทดสอบ ผู้วิจัยพบว่าออนโทโลจีรวมพื้นฐานที่ได้จากโปรแกรมการรวมออนโทโลจีซึ่งผู้วิจัยพัฒนาขึ้นมีโครงสร้างและส่วนประกอบ ได้แก่ คลาสของคลาส คลาสของแอททริบิวต์ ชนิดของคลาสของแอททริบิวต์ ความสัมพันธ์ระหว่างคลาสแบบสืบทอด

คุณลักษณะ ความสัมพันธ์ระหว่างคลาสแบบพี่น้อง ความสัมพันธ์ระหว่างคลาสแบบแอกกรีเกชัน คลาสของความหมายของคลาส คลาสของความหมายของคลาสของแอททริบิวต์ และคลาสของ ข้อกำหนดบูรณภาพของแอททริบิวต์ในคลาส เหมือนกับโครงสร้างและส่วนประกอบของแบบจำลองข้อมูลรวมที่ได้จากการรวมในงานวิจัยดังกล่าว นอกจากนั้น หากนำข้อมูลจากไฟล์ผลการรวมออนโทโลจี (IntegratedOntology.txt) มาสร้างกลับเป็นแบบจำลองข้อมูลเชิงวัตถุในรูปของคลาส แอททริบิวต์และความสัมพันธ์ระหว่างคลาสทั้งสามแบบ ซึ่งกำหนดไว้ด้วยคำหลักต่างๆ เช่น [Class], <subClassOf>, <hasClassAttribute>, [ClassAttribute] เป็นต้น แบบจำลองข้อมูลเชิงวัตถุที่ได้จะตรงกับแบบจำลองข้อมูลรวมที่ได้ในงานวิจัย [20]

ในวิทยานิพนธ์ฉบับนี้ผู้วิจัยได้เลือกนำเสนอตัวอย่างการทดสอบผ่านการรวมแบบจำลองข้อมูลเชิงวัตถุตัวอย่างคู่ที่ 6 ในงานวิจัย [20] โดยได้แสดงรายละเอียดพร้อมทั้งผลการทดสอบไว้ในภาคผนวก ก ประกอบด้วย ภาคผนวก ก.1 แสดงแบบจำลองข้อมูลเชิงวัตถุคู่ที่นำมารวม ภาคผนวก ก.2 แสดงไฟล์โลคัลออนโทโลจีที่ใช้อธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุด รวมทั้งไฟล์ออนโทโลจีความหมายเพิ่มเติมที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจี ส่วนภาคผนวก ก.3 แสดงไฟล์ออนโทโลจีรวมพื้นฐาน (IntegratedOntology.owl) และไฟล์ผลการรวมออนโทโลจีที่ได้จากการรวมโลคัลออนโทโลจีคู่ที่สร้างขึ้นด้วยโปรแกรมการรวมออนโทโลจี ตามลำดับ โดยไฟล์ออนโทโลจีทั้งหมดในภาคผนวกเป็นไฟล์ภาษาอาวล์ที่แสดงในรูปของอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล

เมื่อนำผลลัพธ์ที่ได้ภายในไฟล์ผลการรวมออนโทโลจีมาสร้างกลับเป็นแบบจำลองข้อมูลเชิงวัตถุแล้วนำไปเปรียบเทียบกับแบบจำลองข้อมูลรวมจากการรวมตัวอย่างคู่ที่ 6 ในงานวิจัย [20] จะพบว่าแบบจำลองข้อมูลทั้งสองชุดเหมือนกัน แสดงให้เห็นว่าผลการสร้างออนโทโลจีรวมจากการรวมโลคัลออน-โทโลจีที่อธิบายแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดด้วยวิธีการตามเฟรมเวิร์คของผู้วิจัยให้ผลตรงกับผลการสร้างแบบจำลองข้อมูลรวมเพื่อรวมแบบจำลองข้อมูลเชิงวัตถุด้วยวิธีการที่ใช้ในงานวิจัย [20]

<u>สรุปผลการทดสอบ</u>

จากการทดสอบและเปรียบเทียบออนโทโลจีรวมพื้นฐานที่ได้จากการทำงานตามเฟรมเวิร์คของวิทยานิพนธ์ฉบับนี้ กับแบบจำลองข้อมูลรวมจากการรวมตัวอย่างในงานวิจัย [20] ทั้งสิบคู่ ผู้วิจัยพบว่าผลรวมที่ได้ตรงกัน ทำให้สามารถสรุปได้ว่าเฟรมเวิร์คที่พัฒนาขึ้นสามารถนำไปใช้ในการรวมแบบจำลองข้อมูลเชิงวัตถุได้อย่างถูกต้อง โดยจะให้ผลลัพธ์จากการรวมเป็นออนโทโลจีรวมพื้นฐานที่ดีเทียบเท่ากับแบบจำลองข้อมูลรวมที่ได้จากการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยวิธีการในงานวิจัย [20]

ทั้งนี้ เนื่องจากงานวิจัย [20] ได้นำเมตริกซ์เชิงวัตถุ (Object-Oriented Metrics) มาใช้ทดสอบเพื่อปรับแต่งให้แบบจำลองข้อมูลรวมที่ได้มีลักษณะตรงตามหลักการออกแบบแบบจำลองข้อมูลเชิงวัตถุที่ดี เมื่อผู้วิจัยนำวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุจากงานวิจัย [20] มาใช้ ผู้วิจัยจึงได้ออกแบบและพัฒนาขั้นตอนการสร้างออนโทโลจีรวมโดยมุ่งให้ผลการรวมที่ได้มีโครงสร้างตรงตามผลที่ได้จากงานวิจัย

เดิม เพื่อให้สามารถนำเมตริกซ์เชิงวัตถุที่มีมาประยุกต์ใช้ในการวัดคุณภาพของออนโทโลจีรวมได้ในอนาคต ซึ่งผลรวมที่ได้จะต้องดีเช่นเดียวกัน

นอกจากนั้น เมื่อพิจารณาจากผลการรวมแบบจำลองข้อมูลเชิงวัตถุแต่ละคู่ที่นำมาใช้ทดสอบการที่ออนโทโลจีรวมพื้นฐานที่สร้างขึ้นจากโปรแกรมการรวมออนโทโลจีมีโครงสร้างและส่วนประกอบเหมือนกับแบบจำลองข้อมูลรวมที่ได้จาการรวมแบบจำลองข้อมูลตัวอย่างคู่เดียวกันในงานวิจัย [20] ยังแสดงให้เห็นว่า

- วิธีการแปลงโมเดลระหว่างออนโทโลจีและแบบจำลองข้อมูลเชิงวัตถุที่ผู้วิจัยกำหนดขึ้น โดยใช้การสร้างโลคัลออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุตามโครงสร้างของอัปเปอร์ออนโทโลจี สามารถนำมาใช้งานได้จริงในการแสดงแทนความหมายของแบบจำลองข้อมูลเชิงวัตถุ โดยไม่ส่งผลกระทบต่อความหมายใดๆ ของแบบจำลองข้อมูลเชิงวัตถุที่นำมารวมที่จำเป็นต้องใช้ในการรวม
- ขั้นตอนวิธีการรวมโลคัลออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นมีความถูกต้องและเหมาะสมสำหรับนำไปใช้เป็นขั้นตอนวิธีการรวมออนโทโลจีที่แสดงแทนแบบจำลองข้อมูลเชิงวัตถุซึ่งสร้างขึ้นตามโครงสร้างที่ผู้วิจัยกำหนดไว้
- โปรแกรมการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นทำงานได้ถูกต้องตรงตามวิธีการเปรียบเทียบและการรวมออนโทโลจีที่ผู้วิจัยประยุกต์จากงานวิจัย [20]

6.1.2 การทดสอบผลการสร้างออนโทโลจีรวมด้วยแบบจำลองข้อมูลเชิงวัตถุที่ประยุกต์จากแบบจำลองข้อมูลที่ออกแบบเพื่อใช้งานจริง

การทดสอบในหัวข้อนี้เป็นการนำแบบจำลองข้อมูลเชิงวัตถุที่มีความซับซ้อนมากขึ้นมาใช้ทดสอบเพื่อหาข้อจำกัดของระบบ โดยผู้วิจัยเลือกนำแบบจำลองข้อมูลที่ออกแบบสำหรับใช้งานในโครงการระบบสารสนเทศเพื่อการบริหารของจุฬาลงกรณ์มหาวิทยาลัย (MISCU) จากงานวิจัย [22-25] จำนวนสามคู่มาใช้ทดสอบ โดยจุดประสงค์ของโครงการนี้คือการออกแบบและพัฒนาระบบสารสนเทศในรูปของฐานข้อมูลเชิงสัมพันธ์ (Relational Database) จำนวนห้าระบบแยกกันตามความต้องการใช้ข้อมูล เพื่อนำมาใช้งานร่วมกันในการบริหารงาน ได้แก่ ระบบการเงิน ระบบบุคลากร ระบบทะเบียนนักศึกษาและรายวิชา ระบบอาคารสถานที่และพัสดุครุภัณฑ์ และระบบทรัพย์สิน ซึ่งแต่ละระบบจะมีฐานข้อมูลของตนเองแยกจากระบบอื่น และมีข้อมูลบางส่วนเกี่ยวข้องกับระบบที่เหลือ รวมทั้งมีส่วนเชื่อมโยงถึงกัน [22] ในที่นี้ ผู้วิจัยเลือกนำแบบจำลองข้อมูลมาใช้ทดสอบจำนวนสามคู่ ได้แก่ (1) แบบจำลองข้อมูลของระบบทะเบียนนักศึกษาและรายวิชา [23] กับแบบจำลองข้อมูลของระบบบุคลากร [24] (2) แบบจำลองข้อมูลของระบบทะเบียนนักศึกษาและรายวิชา กับแบบจำลองข้อมูลของระบบการเงิน [25] และ (3) แบบจำลองข้อมูลของระบบบุคลากร กับแบบจำลองข้อมูลของระบบการเงิน

ขั้นตอนการทดสอบ

1. เลือกแบบจำลองข้อมูลที่ต้องการรวมสองชุดจากแบบจำลองข้อมูลสามชุดที่ได้รับการออกแบบขึ้นภายในงานวิจัย [23-25]
2. เนื่องจากแบบจำลองข้อมูลทั้งหมดภายในงานวิจัย [23-25] เป็นแบบจำลองข้อมูลเชิงสัมพันธ์ ดังนั้นต้องแปลงแบบจำลองข้อมูลทั้งสองชุดที่เลือกให้อยู่ในรูปของแบบจำลองข้อมูลเชิงวัตถุ ซึ่งในที่นี้ผู้วิจัยได้เลือกใช้วิธีการแปลงแบบจำลองข้อมูลตามที่อธิบายไว้ในหนังสือ “Object-Oriented Modeling and Design for Database Applications [46]”
3. สร้างโลคัลออนโทโลจีเพื่ออธิบายและเพิ่มความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดที่ได้โดยใช้โปรแกรมโพรเทเจ ตามวิธีการในหัวข้อที่ 5.2.1.1
4. สร้างออนโทโลจีความหมายเพิ่มเติมที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดโดยใช้โปรแกรมโพรเทเจ ตามวิธีการในหัวข้อที่ 5.2.1.2 โดยอ้างอิงจากรายละเอียดที่ผู้ออกแบบฐานข้อมูลแต่ละชุดอธิบายไว้ในพจนานุกรมข้อมูล (Data Dictionary) และถือว่าผู้วิจัยเป็นผู้ที่เข้าใจโครงสร้างของแบบจำลองข้อมูลทั้งสองชุดที่ต้องการรวม
5. ให้ผู้เชี่ยวชาญทดลองนำโลคัลออนโทโลจีทั้งสองชุด และออนโทโลจีความหมายเพิ่มเติมมาสร้างออนโทโลจีรวมพื้นฐาน สำหรับนำไปใช้เปรียบเทียบกับออนโทโลจีรวมพื้นฐานที่ได้จากโปรแกรมการรวมออนโทโลจี
6. สร้างออนโทโลจีรวมพื้นฐานและไฟล์ผลการรวมออนโทโลจีจากโลคัลออนโทโลจีทั้งสองชุดและออนโทโลจีความหมายเพิ่มเติม โดยใช้โปรแกรมการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นตามวิธีการในหัวข้อที่ 5.2.2.1
7. เปรียบเทียบออนโทโลจีรวมพื้นฐานที่ได้ กับออนโทโลจีรวมพื้นฐานที่ได้จากการรวมโดยผู้เชี่ยวชาญ

ผลการทดสอบ

หลังจากนำแบบจำลองข้อมูลทั้งสามคู่ที่ได้รับการออกแบบขึ้นภายในงานวิจัย [23-25] ทั้งสามคู่มาทดลองสร้างออนโทโลจีรวมตามขั้นตอนการทดสอบ ผู้วิจัยพบว่าออนโทโลจีรวมพื้นฐานที่ได้จากโปรแกรมการรวมออนโทโลจีซึ่งผู้วิจัยพัฒนาขึ้นมีโครงสร้างและส่วนประกอบตรงกับโครงสร้างและส่วนประกอบของออนโทโลจีรวมที่ได้จากการรวมโดยผู้เชี่ยวชาญ ทั้งนี้ เมื่อพิจารณาจากผลการรวมออน-โทโลจีที่ได้ ผู้วิจัยพบว่าการเพิ่มความหมายและความสัมพันธ์ของคำศัพท์ระหว่างโลคัลออนโทโลจีในขั้นตอนการทดสอบที่ 3 และ 4 มีส่วนสำคัญอย่างมากต่อออนโทโลจีรวมพื้นฐานที่ได้ เนื่องจากความหมายและความสัมพันธ์ทั้งหมดที่เพิ่มเติมตามความเข้าใจของผู้วิจัยในขั้นตอนดังกล่าวมีส่วนช่วยให้การรวมโลคัลออนโทโลจีมีความถูกต้องมากขึ้น อีกทั้งยังส่งผลให้การรวมโลคัลออนโทโลจีเป็นการรวมในเชิงความหมาย นอกเหนือจากการพิจารณาจากโครงสร้างของแบบจำลองข้อมูลเพียงด้านเดียว

เมื่อนำผลลัพธ์ที่ได้ภายในไฟล์ผลการรวมออนโทโลจีไปเปรียบเทียบกับออนโทโลจีรวมจากการรวมโดยผู้เชี่ยวชาญ ผู้วิจัยพบว่าออนโทโลจีรวมที่ได้จากโปรแกรมการรวมออนโทโลจีมีโครงสร้างและส่วนประกอบตรงกับออนโทโลจีรวมดังกล่าว

<u>สรุปผลการทดสอบ</u>

จากการทดสอบและเปรียบเทียบออนโทโลจีรวมพื้นฐานที่ได้จากการทำงานตามเฟรมเวิร์คของวิทยานิพนธ์ฉบับนี้ กับออนโทโลจีรวมที่ได้จากการรวมโดยผู้เชี่ยวชาญทั้งสามคู่ที่ประยุกต์จากแบบจำลองข้อมูลที่ได้รับการออกแบบขึ้นภายในงานวิจัย [23-25] ผู้วิจัยพบว่าผลรวมที่ได้ตรงกัน ทำให้สามารถสรุปได้ว่าเฟรมเวิร์คที่พัฒนาขึ้นสามารถนำไปใช้ในการรวมแบบจำลองข้อมูลเชิงวัตถุที่มีความซับซ้อนได้ในระดับหนึ่ง โดยผลลัพธ์ที่ได้จากการรวมยังคงถูกต้องตามขั้นตอนวิธีที่เลือกใช้

อย่างไรก็ตาม เมื่อพิจารณาจากผลการรวมออนโทโลจีที่ได้จากการทดสอบ ผู้วิจัยพบว่าการรวมโลคัลออนโทโลจีคู่ใดๆ ให้ได้ผลการรวมที่ดีนั้น โลคัลออนโทโลจีคู่ดังกล่าวจำเป็นต้องได้รับการระบุความหมายเพิ่มเติมและการระบุความสัมพันธ์ระหว่างกันในขั้นตอนการสร้างโลคัลออนโทโลจีทั้งสองชุดให้มากที่สุด เพื่อรองรับการทำงานของขั้นตอนวิธีการเปรียบเทียบออนโทโลจีในการจำแนกความขัดแย้งระหว่างโลคัลออนโทโลจีคู่นั้น

6.2 การทดสอบความถูกต้องในการสอบถามข้อมูลผ่านระบบการรวม

การทดสอบความถูกต้องในการสอบถามข้อมูลผ่านระบบการรวมแบบจำลองข้อมูลเชิงวัตถุที่สร้างขึ้นตามแนวคิดแบบรวมหลายออนโทโลจีมีวัตถุประสงค์ดังต่อไปนี้

- เพื่อแสดงตัวอย่างการทำงานของระบบการสอบถามข้อมูลที่พัฒนาขึ้นตามแนวคิดแบบรวมหลายออนโทโลจี ซึ่งกำหนดให้ออนโทโลจีรวมจะต้องถูกกำกับลงบนแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุด ในขณะเดียวกันแผ่นแบบการสอบถามข้อมูลก็จะต้องสร้างขึ้นโดยอ้างอิงจากออนโทโลจีรวมเช่นเดียวกัน
- เพื่อตรวจสอบวิธีการกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูลเชิงวัตถุ โดยใช้การสร้างวิวตามออนโทโลจีรวมที่ได้เพื่อกำกับลงบนแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดที่นำมารวม และความสามารถของออนโทโลจีในการจัดการกับความต่างแบบของข้อมูล
- เพื่อตรวจสอบการทำงานของแผ่นแบบการสอบถามข้อมูลที่ผู้วิจัยพัฒนาขึ้นโดยอ้างอิงจากออนโทโลจีรวมที่ได้ เพื่อใช้เป็นตัวกลางในการสอบถามข้อมูลจากฐานข้อมูลเชิงวัตถุทั้งสองชุดที่นำมารวม และความสามารถของออนโทโลจีในการจัดการกับการกระจายตัวของข้อมูล
- เพื่อตรวจสอบความถูกต้องในการแปลงข้อคำถามที่ได้รับจากแผ่นแบบการสอบถามข้อมูลไปเป็นข้อคำถามที่ใช้งานกับฐานข้อมูลเชิงวัตถุทั้งสองชุด

- เพื่อแสดงให้เห็นถึงประโยชน์จากสอบถามข้อมูลผ่านออนโทโลจีรวมที่ได้รับการอนุมานแล้ว ซึ่งจะส่งผลให้ผู้ใช้จะได้รับข้อมูลจากการสอบถามเพิ่มขึ้น และการสอบถามข้อมูลเป็นไปในเชิงความหมายมากยิ่งขึ้น

ในการทดสอบความถูกต้องในการสอบถามข้อมูลผ่านระบบการรวมแบบจำลองข้อมูลเชิงวัตถุที่สร้างขึ้นตามแนวคิดแบบรวมหลายออนโทโลจีนั้น ผู้วิจัยจำเป็นต้องสร้างฐานข้อมูลเชิงวัตถุขึ้นจริงเพื่อนำมาใช้ทดลองสอบถามข้อมูลและเปรียบเทียบผลลัพธ์ที่ได้ โดยวิทยานิพนธ์ฉบับนี้เลือกใช้ระบบจัดการฐานข้อมูลออราเคิล เอ็กซ์เพรส เอดิชัน 10จี (Oracle Express Edition 10g) [44] เป็นฐานข้อมูลในการทดสอบเนื่องจากสนับสนุนการใช้งานฐานข้อมูลเชิงวัตถุ การทดสอบประกอบด้วยการนำออนโทโลจีรวมไปกำกับลงบนแบบจำลองข้อมูลเชิงวัตถุแต่ละชุด พร้อมทั้งนำไปสร้างแผ่นแบบการสอบถามข้อมูล แล้วตรวจสอบความถูกต้องและครบถ้วนของผลลัพธ์ที่ได้จากการสอบถามผ่านแผ่นแบบของระบบการรวมไปยังฐานข้อมูลทั้งสองชุด โดยเปรียบเทียบกับผลรวมจากการสอบถามโดยตรงที่ฐานข้อมูลแต่ละชุดที่สร้างไว้ ทั้งนี้ ผู้วิจัยได้แบ่งการทดสอบนี้ออกเป็นสองส่วนคือ การทดสอบด้วยออนโทโลจีรวมที่ไม่ผ่านการอนุมาน (ข้ามขั้นตอนการอนุมานในหัวข้อที่ 5.2.2.2) และการทดสอบด้วยออนโทโลจีรวมที่ผ่านการอนุมานแล้ว

6.2.1 การทดสอบความถูกต้องในการสอบถามข้อมูลผ่านระบบการรวมโดยใช้ออนโทโลจีรวมที่ไม่ผ่านการอนุมาน

<u>ขั้นตอนการทดสอบ</u>

1. เลือกแบบจำลองข้อมูลเชิงวัตถุที่ต้องการทดสอบหนึ่งคู่จากแบบจำลองข้อมูลตัวอย่างที่ใช้ในงานวิจัย [20]
2. นำแบบจำลองข้อมูลเชิงวัตถุที่เลือกแต่ละชุดมาสร้างเป็นฐานข้อมูลเชิงวัตถุลงในระบบจัดการฐานข้อมูลออราเคิล เอ็กซ์เพรส
3. สร้างโลคัลออนโทโลจีเพื่ออธิบายและเพิ่มความความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุด และสร้างออนโทโลจีความหมายที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดตามวิธีการในหัวข้อที่ 5.2.1
4. สร้างออนโทโลจีรวมพื้นฐานและไฟล์ผลการรวมออนโทโลจีจากโลคัลออนโทโลจีทั้งสองชุดและออนโทโลจีความหมายเพิ่มเติม โดยใช้โปรแกรมการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นตามวิธีการในหัวข้อที่ 5.2.2.1
5. นำออนโทโลจีรวมพื้นฐานที่ได้ไปสร้างเป็นวิวเพื่อกำกับลงบนฐานข้อมูลเชิงวัตถุที่สร้างไว้แต่ละชุดตามวิธีการในหัวข้อที่ 5.2.3
6. สร้างแผ่นแบบการสอบถามข้อมูลโดยอ้างอิงจากออนโทโลจีรวมพื้นฐานที่ได้ตามวิธีการในหัวข้อที่ 5.3

7. ทดลองสอบถามข้อมูลผ่านแผ่นแบบ และเปรียบเทียบผลลัพธ์ที่ได้จากการสอบถามโดยตรงที่ฐานข้อมูลแต่ละชุด

ผลการทดสอบ

ในวิทยานิพนธ์ฉบับนี้ผู้วิจัยเลือกทดสอบการสอบถามข้อมูลผ่านระบบการรวมแบบจำลองข้อมูลเชิงวัตถุที่สร้างขึ้นจากการรวมแบบจำลองข้อมูลเชิงวัตถุตัวอย่างคู่ที่ 6 ในงานวิจัย [20] เมื่อนำแบบจำลองข้อมูลคู่ดังกล่าวมาสร้างโลคัลออนโทโลจี และสร้างออนโทโลจีรวมตามลำดับแล้ว จะได้ไฟล์ออนโทโลจีรวมพื้นฐานภาษาอาวล์ในรูปของอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล (IntegratedOntology.owl) และไฟล์ผลการรวมออนโทโลจี (IntegratedOntology.txt) ดังแสดงในภาคผนวก ก.3 ซึ่งเมื่อนำออนโทโลจีรวมพื้นฐานที่ได้ไปสร้างเป็นวิวเพื่อกำกับลงบนฐานข้อมูลเชิงวัตถุแต่ละชุด วิวที่สร้างขึ้นมีรายละเอียดดังแสดงในภาคผนวก ก.4 จากนั้นผู้วิจัยจะนำออนโทโลจีรวมพื้นฐานไปใช้สร้างแผ่นแบบการสอบถามข้อมูลเพื่อใช้ทดสอบการสอบถามข้อมูลผ่านระบบการรวม รูปที่ 6.1 ถึง 6.3 แสดงผลลัพธ์จากการทดสอบดังต่อไปนี้

- การทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่หนึ่งโดยตรง แสดงไว้ในรูปที่ 6.1
- การทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่สองโดยตรง แสดงไว้ในรูปที่ 6.2
- การทดสอบการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามข้อมูล แสดงไว้ในรูปที่ 6.3

```
SELECT a.NAME, a.SEX, a.HEIGHT, a.ADDRESS, a.GPA, a.FACULTY.NAME,
  a.FACULTY.ADDRESS, a.FACULTY.DEAN FROM StudentTab a;
```

NAME	SEX	HEIGHT	ADDRESS	GPA	FACULTY. NAME	FACULTY. ADDRESS	FACULTY. DEAN
Ms.Anittha Jutarosaga	F	165	265 Sukhumvit Road, Bangkok	3.45	engineer	Building B	Mr. Vasu Pornlikitpiam chote
Ms. Chotika Jindaapirat	F	162	25 Thon Buri Road, Bangkok	2.8	science	Building A	Ms. Preeyachat Tammapimook
Mr. Peerapong Phraestri	M	172	3758 Ratchasima Road, Bangkok	2.65	engineer	Building B	Mr. Vasu Pornlikitpiam chote
Mr. Panupong Wongma	M	170	201 Phaya Thai Road, Bangkok	2.55	science	Building A	Ms. Preeyachat Tammapimook

```
SELECT a.* FROM FacultyTab a;
```

NAME	ADDRESS	DEAN
science	Building A	Ms. Preeyachat Tammapimook
engineer	Building B	Mr. Vasu Pornlikitpiamchote

รูปที่ 6.1 ผลการทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่หนึ่ง

```
SELECT a.* FROM PersonTab a;
```

NAME	GENDER	HEIGHT	HOME_NO	HOME_CITY
Mr. Akaraphol Charochsopa	male	1.7	54 Bang Bon Road	Bangkok
Mr. Kasem Un-sri	male	1.75	149 Sathon Road	Bangkok
Ms. Narukamol Poomipark	female	1.6	677 Phaya Thai Road	Bangkok
Mr. Arthit Larpmahapaisarn	male	1.75	8489 Bang Khae Road	Bangkok
Mr. Siripat Sahamethapat	male	1.7	8 Ranong Road	Bangkok
Ms. Sasiwong wongwantanee	female	1.57	151 Nakhon Sawan Road	Bangkok
Ms. Thunyaluk Limpittaya	female	1.65	48 Lat Phrao Road	Bangkok
Mr. Chakchaiphun whaiyaphun	male	1.72	636 Asoke Road	Bangkok

```
SELECT a.PERSON.NAME, a.PERSON.GENDER, a.PERSON.HEIGHT, a.PERSON.HOME_NO,
   a.PERSON.HOME_CITY, a.FACULTY_NAME, a.GPA, a.THESIS, a.LAB_NAME
   FROM GradStudentTab a;
```

PERSON. NAME	PERSON. GENDER	PERSON. HEIGHT	PERSON. HOME_NO	PERSON. HOME_CITY
Mr. Siripat Sahamethapat	male	1.7	8 Ranong Road	Bangkok
Ms. Sasiwong wongwantanee	female	1.57	151 Nakhon Sawan Road	Bangkok
Ms. Thunyaluk Limpittaya	female	1.65	48 Lat Phrao Road	Bangkok
Mr. Chakchaiphun whaiyaphun	male	1.72	636 Asoke Road	Bangkok

PERSON. NAME	FACULTY_NAME	GPA	THESIS	LAB_NAME
Mr. Siripat Sahamethapat	science	2.75	OCR Development Designing	Usability Testing Lab
Ms. Sasiwong wongwantanee	engineer	2.9	Robot Vision Development	Project Lab
Ms. Thunyaluk Limpittaya	account	3.2	Design Optimizing algorithms	Computational Mathematics Lab
Mr. Chakchaiphun whaiyaphun	engineer	3	Image content extraction	Artificial Intelligence Lab

รูปที่ 6.2 ผลการทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่สอง

Result From: SELECT Name, Gender, Height, Address FROM Person

Name	Gender	Height	Address
Ms. Anittha Jutarosaga	female	1.65	265 Sukhumvit Road, Bangkok
Ms. Chotika Jindaapirat	female	1.62	25 Thon Buri Road, Bangkok
Mr. Peerapong Phraestri	male	1.72	3758 Ratchasima Road, Bangkok
Mr. Panupong Wongma	male	1.7	201 Phaya Thai Road, Bangkok
Mr. Akaraphol Charochsopa	male	1.7	54 Bang Bon Road, Bangkok
Mr. Arthit Larpmahapaisarn	male	1.75	8489 Bang Khae Road, Bangkok
Mr. Chakchaiphun whaiyaphun	male	1.72	636 Asoke Road, Bangkok
Mr. Kasem Un-sri	male	1.75	149 Sathon Road, Bangkok
Mr. Siripat Sahamethapat	male	1.7	8 Ranong Road, Bangkok
Ms. Narukamol Poomipark	female	1.6	677 Phaya Thai Road, Bangkok
Ms. Sasiwong wongwantanee	female	1.57	151 Nakhon Sawan Road, Bangkok
Ms. Thunyaluk Limpittaya	female	1.65	48 Lat Phrao Road, Bangkok

Result From: SELECT Name, Gender, Height, Address, Gpa, Faculty_name FROM Student

Name	Gender	Height	Address	Gpa	Faculty_name
Ms. Anittha Jutarosaga	female	1.65	265 Sukhumvit Road, Bangkok	3.45	engineer
Ms. Chotika Jindaapirat	female	1.62	25 Thon Buri Road, Bangkok	2.8	science
Mr. Peerapong Phraestri	male	1.72	3758 Ratchasima Road, Bangkok	2.65	engineer
Mr. Panupong Wongma	male	1.7	201 Phaya Thai Road, Bangkok	2.55	science
Mr. Siripat Sahamethapat	male	1.7	8 Ranong Road, Bangkok	2.75	science
Ms. Sasiwong wongwantanee	female	1.57	151 Nakhon Sawan Road, Bangkok	2.9	engineer
Ms. Thunyaluk Limpittaya	female	1.65	48 Lat Phrao Road, Bangkok	3.2	account
Mr. Chakchaiphun whaiyaphun	male	1.72	636 Asoke Road, Bangkok	3	engineer

รูปที่ 6.3 ผลการทดสอบการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามข้อมูล

Result From: SELECT Name, Gender, Height, Address, Gpa, Faculty_name, Lab_name FROM GradStudent

Name	Gender	Height	Address	Gpa	Faculty_name	Lab_name
Mr.Siripat Sahamethapat	male	1.7	8 Ranong Road, Bangkok	2.75	science	Usability Testing Lab
Ms.Sasiwong wongwantanee	female	1.57	151 Nakhon Sawan Road, Bangkok	2.9	engineer	Project Lab
Ms.Thunyaluk Limpittaya	female	1.65	48 Lat Phrao Road, Bangkok	3.2	account	Computational Mathematics Lab
Mr.Chakchaiphun whaiyaphun	male	1.72	636 Asoke Road, Bangkok	3	engineer	Artificial Intelligence Lab

Result From: SELECT Name FROM Faculty

Name
science
engineer
account

รูปที่ 6.3 ผลการทดสอบการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามข้อมูล (ต่อ)

เมื่อนำผลการสอบถามข้อมูลที่ได้เปรียบเทียบกัน จะเห็นได้ว่าข้อมูลที่ได้จากการสอบถามผ่านแผ่นแบบการสอบถามข้อมูลจะเท่ากับผลรวมที่ได้จากการสอบถามโดยตรงจากแบบจำลองข้อมูลทั้งสองชุดโดยไม่มีการสูญหายของข้อมูล ทั้งนี้ อาจมีข้อมูลบางฟิลด์ที่ถูกเปลี่ยนแปลงไปจากเดิม เช่น ข้อมูลฟิลด์ “Home_No” และ “Home_City” ในแบบจำลองข้อมูลชุดที่สองถูกรวมกันเป็นฟิลด์ “Address” ในผลการสอบถามผ่านแผ่นแบบ หรือข้อมูลฟิลด์ “Height” ในแบบจำลองข้อมูลชุดที่หนึ่งถูกปรับให้มีหน่วยเป็นเมตรในผลการสอบถามผ่านแผ่นแบบ เป็นต้น

สรุปผลการทดสอบ

จากผลการทดสอบการสอบถามข้อมูลผ่านระบบการรวมโดยใช้ออนโทโลจีรวมที่ไม่ผ่านการอนุมาน และเปรียบเทียบกับผลการสอบถามข้อมูลจากฐานข้อมูลเชิงวัตถุโดยตรง ผู้วิจัยพบว่าข้อมูลที่ได้จากการสอบถามผ่านแผ่นแบบของระบบการรวมข้อมูลมีความถูกต้อง เมื่อเปรียบเทียบกับผลที่ได้จากการสอบถามโดยตรงจากฐานข้อมูลแต่ละชุด นอกจากข้อมูลบางฟิลด์ถูกเปลี่ยนแปลงจากผลการทำงานของฟังก์ชันที่สร้างขึ้นในวิว เพื่อปรับให้ผลลัพธ์ที่ได้จากฐานข้อมูลทั้งสองชุดตรงกัน และแก้ไขความขัดแย้งระหว่างข้อมูล ทั้งนี้การแปลงผลลัพธ์ที่เกิดขึ้นต่างไม่ทำให้ความหมายของข้อมูลผิดไปจากเดิม

นอกจากนั้น เมื่อพิจารณาจากผลการสอบถามผ่านแผ่นแบบที่ได้จากการทดสอบ การที่ผลลัพธ์ที่ได้มีความถูกต้อง ครบถ้วน และไม่สูญเสียความหมายเดิมข้องข้อมูลยังแสดงให้เห็นว่า

- การทำงานของระบบการสอบถามข้อมูลที่พัฒนาขึ้นตามแนวคิดแบบรวมหลายออนโทโลจี ซึ่งผู้วิจัยพัฒนาขึ้น โดยกำหนดให้นำออนโทโลจีรวมไปกำกับลงบนแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดเพื่อเป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับฐานข้อมูลแต่ละชุด ในขณะเดียวกันแผ่นแบบการสอบถามข้อมูลก็จะต้องสร้างขึ้นโดยอ้างอิงจากออนโทโลจีรวมนี้ สามารถทำงานได้อย่างถูกต้อง และให้ผลลัพธ์ที่เหมาะสม
- วิธีการกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูลเชิงวัตถุ โดยใช้การสร้างวิวเป็นไปอย่างถูกต้อง วิวที่สร้างขึ้นจากออนโทโลจีรวมสามารถจัดการกับความต่างแบบของข้อมูลโดยการแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม และแก้ไขความขัดแย้งระหว่างข้อมูลจากต่างฐานข้อมูลกันได้ โดยไม่กระทบต่อความหมายเดิมของข้อมูล
- การทำงานของแผ่นแบบการสอบถามข้อมูลที่ผู้วิจัยพัฒนาขึ้นโดยอ้างอิงจากออนโทโลจีรวมมีความถูกต้อง แผ่นแบบสามารถรับข้อคำถามจากผู้ใช้และส่งต่อไปยังฐานข้อมูลแต่ละชุด พร้อมทั้งรับผลที่ได้จากฐานข้อมูลเหล่านั้นกลับมาแสดงได้อย่างถูกต้อง ครบถ้วนและกำจัดความซ้ำซ้อนที่อาจเกิดขึ้นระหว่างฐานข้อมูลทั้งสองชุด เช่น การกำจัดความซ้ำซ้อนจากผลลัพธ์การสอบถามฟิลด์ “Name” ของ “Faculty” ซึ่งแสดงให้เห็นถึงความสามารถของออนโทโลจีในการจัดการกับการกระจายตัวของข้อมูล
- การแปลงข้อคำถามที่ได้รับจากแผ่นแบบการสอบถามข้อมูลไปเป็นข้อคำถามที่ใช้งานกับฐานข้อมูลเชิงวัตถุทั้งสองชุดเป็นไปอย่างถูกต้อง ผลลัพธ์ทั้งหมดที่ได้จากการสอบถามสอดคล้องกับข้อคำถามที่เลือกระบุลงในแผ่นแบบการสอบถามข้อมูล

6.2.2 การทดสอบความถูกต้องในการสอบถามข้อมูลผ่านระบบการรวมโดยใช้ออนโทโลจีรวมที่ผ่านการอนุมานแล้ว

<u>ขั้นตอนการทดสอบ</u>

1. เลือกแบบจำลองข้อมูลเชิงวัตถุที่ต้องการทดสอบหนึ่งคู่จากแบบจำลองข้อมูลตัวอย่างที่ใช้ในงานวิจัย [20]
2. นำแบบจำลองข้อมูลเชิงวัตถุที่เลือกแต่ละชุดมาสร้างเป็นฐานข้อมูลเชิงวัตถุลงในระบบจัดการฐานข้อมูลออราเคิล เอ็กซ์เพรส
3. สร้างโลคัลออนโทโลจีเพื่ออธิบายและเพิ่มความความหมายให้กับแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุด และสร้างออนโทโลจีความหมายที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีทั้งสองชุดตามวิธีการในหัวข้อที่ 5.2.1

4. สร้างออนโทโลจีรวมพื้นฐานและไฟล์ผลการรวมออนโทโลจีจากโลคัลออนโทโลจีทั้งสองชุด และออนโทโลจีความหมายเพิ่มเติม โดยใช้โปรแกรมการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นตามวิธีการในหัวข้อที่ 5.2.2.1
5. กำหนดกฎเพื่ออนุมานออนโทโลจีรวมพื้นฐาน ตามวิธีการในหัวข้อที่ 5.2.2.2 ก่อนจะนำออนโทโลจีรวมที่ได้ไปใช้งาน
6. นำออนโทโลจีรวมที่ได้ไปสร้างเป็นวิวเพื่อกำกับลงบนฐานข้อมูลเชิงวัตถุที่สร้างไว้แต่ละชุดตามวิธีการในหัวข้อที่ 5.2.3
7. สร้างแผ่นแบบการสอบถามข้อมูลโดยอ้างอิงจากออนโทโลจีรวมที่ได้ตามวิธีการในหัวข้อที่ 5.3
8. ทดลองสอบถามข้อมูลผ่านแผ่นแบบ และเปรียบเทียบผลลัพธ์ที่ได้จากการสอบถามโดยตรงที่ฐานข้อมูลแต่ละชุด

<u>ผลการทดสอบ</u>

ในวิทยานิพนธ์ฉบับนี้ผู้วิจัยได้เลือกทดสอบการสอบถามข้อมูลผ่านระบบการรวมแบบจำลองข้อมูลเชิงวัตถุที่สร้างขึ้นจากการรวมแบบจำลองข้อมูลเชิงวัตถุตัวอย่างคู่ที่ 4 ในงานวิจัย [20] (เป็นแบบจำลองข้อมูลชุดเดียวกับที่ใช้ในงานวิจัย [43]) เมื่อนำแบบจำลองข้อมูลคู่ดังกล่าวมาสร้างโลคัลออนโทโลจี และสร้างออนโทโลจีรวมตามลำดับแล้ว ผู้วิจัยจะกำหนดกฎเพื่ออนุมานออนโทโลจีรวมพื้นฐานที่ได้ โดยในที่นี้จะกำหนดเงื่อนไขให้ข้อมูลใน “Children” ที่มีฟิลด์ “Age” ระหว่าง ‘0’ ถึง ‘2’ ถูกแสดงในการสอบถามข้อมูล “Baby” ด้วย ทำให้ได้ไฟล์ออนโทโลจีรวมภาษาอาวล์ในรูปของอาร์ดีเอฟ/เอ็กซ์เอ็มแอลดังแสดงในภาคผนวก ค.3 ซึ่งเมื่อนำออนโทโลจีรวมที่ได้ไปสร้างเป็นวิวเพื่อกำกับลงบนฐานข้อมูลเชิงวัตถุแต่ละชุด วิวที่สร้างขึ้นมีรายละเอียดดังแสดงในภาคผนวก ค.4 จากนั้น ผู้วิจัยจะนำออนโทโลจีรวมดังกล่าวไปใช้สร้างแผ่นแบบการสอบถามข้อมูลเพื่อใช้ทดสอบการสอบถามข้อมูล รูปที่ 6.4 ถึง 6.6 แสดงผลลัพธ์จากการทดสอบดังต่อไปนี้

- การทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่หนึ่งโดยตรง แสดงไว้ในรูปที่ 6.4
- การทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่สองโดยตรง แสดงไว้ในรูปที่ 6.5
- การทดสอบการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามข้อมูล แสดงไว้ในรูปที่ 6.6

```
SELECT a.PERSON.NAME, a.PERSON.AGE, a.GRADE, a.SCHOOL_NAME
   FROM ChildrenTab1 a;
```

PERSON.NAME	PERSON.AGE	GRADE	SCHOOL_NAME
Joshua	2	3.5	Bangkok Patana School
Madison	10	3.25	Chiang Mai International School
Matthew	17	3.1	Ruamrudee International School

รูปที่ 6.4 ผลการทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่หนึ่ง

```
SELECT a.PERSON.NAME, a.PERSON.AGE, a.GRADE, a.KINDERGARTEN_NAME,
   a.BIRHT_WEIGHT FROM BabyTab2 a;
```

PERSON.NAME	PERSON.AGE	GRADE	KINDERGARTEN_NAME	BIRTH_WEIGHT
Abigail	1	4	Harrow Kindergarten International School	2.85
Ethan	2	4	Wells International Kindergarten	3.3
Olivia	2	3.75	IPC International Kindergarten	3

รูปที่ 6.5 ผลการทดสอบการสอบถามข้อมูลจากแบบจำลองข้อมูลชุดที่สอง

เมื่อนำผลการสอบถามข้อมูลที่ได้เปรียบเทียบกันและพิจารณาโดยรวม จะเห็นได้ว่าข้อมูลที่ได้จากการสอบถามผ่านแผ่นแบบการสอบถามข้อมูลจะเท่ากับผลรวมที่ได้จากการสอบถามโดยตรงจากแบบจำลองข้อมูลทั้งสองชุดเช่นเดียวกับผลการสอบถามผ่านระบบการรวมที่สร้างขึ้นจากออนโทโลจีรวมที่ไม่ผ่านการอนุมาน โดยไม่มีการสูญหายของข้อมูล

Result From: SELECT Name, Age, Grade, School_name FROM Children

Name	Age	Grade	School_name
Joshua	2	3.5	Bangkok Patana School
Madison	10	3.25	Chiang Mai International School
Matthew	17	3.1	Ruamrudee International School
Abigail	1	4	Harrow Kindergarten International School
Ethan	2	4	Wells International Kindergarten
Olivia	2	3.75	IPC International Kindergarten

รูปที่ 6.6 ผลการทดสอบการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามข้อมูล

Result From: SELECT Name, Age, Grade, School_name, Birth_weight FROM Baby

Name	Age	Grade	School_name	Birth_weight
Joshua	2	3.5	Bangkok Patana School	-
Abigail	1	4	Harrow Kindergarten International School	2.85
Ethan	2	4	Wells International Kindergarten	3.3
Olivia	2	3.75	IPC International Kindergarten	3

รูปที่ 6.6 ผลการทดสอบการสอบถามข้อมูลผ่านแผ่นแบบการสอบถามข้อมูล (ต่อ)

อย่างไรก็ตาม เมื่อพิจารณาผลการสอบถามข้อมูล “Baby” ผ่านแผ่นแบบการสอบถามข้อมูลจะเห็นได้ว่า ผลลัพธ์ที่ได้เพิ่มขึ้นจากผลการสอบถามข้อมูล “Baby” จากแบบจำลองข้อมูลชุดที่สองโดยตรง ซึ่งเป็นผลมาจากการอนุมานออนโทโลจีรวมก่อนที่จะนำมาใช้สร้างวิวและแผ่นแบบที่ใช้ในการสอบถามข้อมูลผ่านระบบการรวม โดยข้อมูลที่เพิ่มขึ้นยังคงถูกต้องตามความหมายที่กำหนด และไม่กระทบต่อข้อมูลส่วนอื่นๆ

สรุปผลการทดสอบ

จากผลการทดสอบการสอบถามข้อมูลผ่านระบบการรวมโดยใช้ออนโทโลจีรวมที่ผ่านการอนุมานแล้ว และเปรียบเทียบกับผลการสอบถามข้อมูลจากฐานข้อมูลเชิงวัตถุโดยตรง ผู้วิจัยพบว่าข้อมูลที่ได้จากการสอบถามผ่านแผ่นแบบของระบบการรวมข้อมูลมีความถูกต้อง เมื่อเปรียบเทียบกับผลที่ได้จากการสอบถามโดยตรงจากฐานข้อมูลแต่ละชุด นอกจากนั้น ผลลัพธ์ที่ได้จากการสอบถามผ่านแผ่นแบบยังเพิ่มขึ้น ซึ่งเป็นผลจากการอนุมานออนโทโลจีรวมก่อนที่จะนำมาใช้สร้างวิวและแผ่นแบบที่ใช้ในการสอบถามข้อมูลผ่านระบบการรวม ทั้งนี้ผลลัพธ์ที่ได้ทั้งหมดยังคงมีความหมายของข้อมูลคงเดิม ทำให้สามารถสรุปได้ว่าการอนุมานออนโทโลจีรวมก่อนจะนำมาใช้จะทำให้ผู้ใช้ได้รับประโยชน์จากสอบถามข้อมูลผ่านออนโทโลจีรวมดังกล่าว โดยจะส่งผลให้ผู้ใช้จะได้รับข้อมูลจากการสอบถามเพิ่มขึ้น และการสอบถามข้อมูลเป็นไปในเชิงความหมายมากยิ่งขึ้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 7
สรุปผลการวิจัย และข้อเสนอแนะ

จากการศึกษาวิจัยเรื่องการรวมแบบจำลองข้อมูลเพื่อนำไปใช้ในการรวมสารสนเทศ ผู้วิจัยได้ออกแบบและพัฒนาต้นแบบระบบการรวมแบบจำลองข้อมูลเชิงวัตถุด้วยแนวคิดแบบรวมหลายออนโทโลจีขึ้น โดยวิทยานิพนธ์ฉบับนี้มุ่งเน้นที่การนำเสนองานในสองส่วนหลักคือ ส่วนขั้นตอนวิธีในการสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาส และส่วนแนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูลแบบรวมหลายออนโทโลจี หลังจากทดสอบระบบต้นแบบที่ได้ด้วยแบบจำลองข้อมูลเชิงวัตถุตัวอย่างและแบบจำลองข้อมูลจริง ผู้วิจัยพบว่าระบบสามารถทำงานได้อย่างถูกต้องครบถ้วน นอกจากนั้น การใช้ออนโทโลจีเป็นตัวกลางในการรวมข้อมูลยังส่งผลให้การรวมและการสอบถามข้อมูลเป็นไปในเชิงความหมายมากยิ่งขึ้น ทั้งจากการใช้โลคัลออนโทโลจีเพื่ออธิบายแบบจำลองข้อมูลเชิงวัตถุแต่ละชุด การสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาสในโลคัลออนโทโลจี การนำออนโทโลจีรวมที่ได้ไปกำกับลงบนแบบจำลองข้อมูล และการสอบถามข้อมูลผ่านแผ่นแบบที่สร้างขึ้นจากออนโทโลจีรวม ดังรายละเอียดที่ได้นำเสนอไปแล้วในบทที่ 3 ถึงบทที่ 6 ซึ่งสามารถสรุปและเปรียบเทียบผลการวิจัย วิเคราะห์ถึงปัญหาและข้อจำกัดที่พบ พร้อมทั้งเสนอแนะแนวทางเพื่อการทำวิจัยต่อในอนาคตได้ดังนี้

7.1 สรุปผลการวิจัย

ถึงแม้ว่าการรวมแบบจำลองข้อมูลเชิงวัตถุซึ่งเป็นข้อมูลที่มีโครงสร้างค่อนข้างชัดเจนอยู่แล้ว สามารถทำได้โดยพิจารณาจากวากยสัมพันธ์หรือความหมายเชิงโครงสร้าง เช่น คลาส แอททริบิวต์ และความสัมพันธ์ระหว่างคลาสที่ปรากฏในตัวแบบจำลองข้อมูล แต่ในบางครั้งความหมายเชิงโครงสร้างเหล่านั้นไม่เพียงพอต่อการจำแนกความขัดแย้งบางประเภท (โดยเฉพาะความขัดแย้งระดับข้อมูล) ระหว่างแบบจำลองข้อมูลคู่ที่นำมารวมออกจากกันได้ ทำให้ผลการรวมที่ได้ไม่ดีเท่าที่ควร งานวิจัยในปัจจุบันจึงได้พยายามนำออนโทโลจีมาใช้ช่วยอธิบายความหมายทั้งหมดของข้อมูลภายในแบบจำลองข้อมูล เพื่อให้ความหมายที่ปรากฏในออนโทโลจีที่ได้ครอบคลุมทั้งความหมายเชิงโครงสร้างและความหมายเพิ่มเติมอื่นๆ ส่งผลให้สามารถจำแนกความขัดแย้งประเภทต่างๆ ได้อย่างถูกต้องยิ่งขึ้น

จากความสามารถของออนโทโลจีในการอธิบายความหมายดังกล่าว วิทยานิพนธ์ฉบับนี้ได้นำออนโทโลจีมาใช้แสดงแทนแบบจำลองข้อมูลแต่ละชุด รวมทั้งผลลัพธ์ที่ได้จากการรวมแบบจำลองข้อมูลเชิงวัตถุคู่นั้น ทำให้การพิจารณาความแตกต่างระหว่างแบบจำลองข้อมูลที่ต้องการรวมเป็นไปในเชิงความหมายมากขึ้น และสามารถจำแนกความขัดแย้งระหว่างแบบจำลองข้อมูลที่ต้องการรวมได้ดียิ่งขึ้น ด้วยวิธีการนี้ ออนโทโลจีจะทำหน้าที่เป็นตัวกลางในการรวมแบบจำลองข้อมูลระหว่างแบบจำลองข้อมูลเชิงวัตถุแต่ละชุดและกับผู้ใช้ รวมทั้งช่วยจัดการกับความต่างแบบและการกระจายตัวที่เกิดขึ้นในการใช้งานระบบการรวมแบบจำลองข้อมูลเชิงวัตถุที่ได้

ทั้งนี้ ผู้วิจัยพบว่าผลการทดสอบที่ได้จากวิทยานิพนธ์ฉบับนี้สอดคล้องกับสมมติฐานที่ได้ตั้งไว้ว่า หากนำแบบจำลองข้อมูลที่ได้รับการอธิบายและแสดงแทนความหมายในรูปแบบที่เหมาะสมและเพียงพอ (ในที่นี้เลือกใช้ออนโทโลจี) มารวมเข้าด้วยกันโดยใช้ขั้นตอนวิธีที่เหมาะสม (ในที่นี้หมายถึงขั้นตอนการทำงานของระบบการรวมที่ได้พัฒนาขึ้น) จะทำให้การรวมแบบจำลองข้อมูลเป็นไปอย่างถูกต้องและครบถ้วน

7.1.1 แนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูลแบบรวมหลายออนโทโลจี

แนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูลแบบรวมหลายออนโทโลจีถูกพัฒนาขึ้นจากข้อดีของทั้งสามแนวคิดที่มีอยู่เดิม ได้แก่ แบบออนโทโลจีเดียว แบบหลายออนโทโลจีและแบบผสม โดยผู้วิจัยต้องการให้แนวคิดใหม่ที่ได้นำไปประยุกต์ใช้งานได้ง่ายขึ้น ลดภาระของผู้ทำการรวมข้อมูลลง ระบบการรวมข้อมูลที่ได้จากแนวคิดนี้สามารถทำงานได้จริง มีขั้นตอนที่ไม่ยุ่งยากซับซ้อน และข้อมูลรวมที่ได้ยังคงถูกต้องครบถ้วน

เมื่อพิจารณาตามองค์ประกอบในการรวมแบบจำลองข้อมูล การทำงานของแนวคิดแบบรวมหลายออนโทโลจีประกอบด้วยส่วนต่างๆ ดังนี้

1. *แบบจำลองข้อมูลโลคัล* แนวคิดแบบรวมหลายออนโทโลจีเลือกใช้การแสดงแทนแบบจำลองข้อมูลที่ต้องการรวมหนึ่งชุดในรูปของโลคัลออนโทโลจีหนึ่งชุด เพื่อให้ความหมายทั้งหมดของแบบจำลองข้อมูลที่ต้องการรวมได้รับการอธิบายไว้ในโลคัลออนโทโลจี
2. *แบบจำลองข้อมูลโกลบอล* แนวคิดแบบรวมหลายออนโทโลจีกำหนดให้สร้างแบบจำลองข้อมูลรวมหนึ่งชุดในรูปของออนโทโลจีรวมที่ได้จากการรวมโลคัลออนโทโลจีทั้งสองชุด เพื่อให้ออนโทโลจีรวมที่ได้แสดงแทนความหมายทั้งหมดของแบบจำลองข้อมูลที่นำมารวม
3. *แมปปิงระหว่างแบบจำลองข้อมูลโลคัลกับแบบจำลองข้อมูลโกลบอล* แนวคิดแบบรวมหลายออนโทโลจีใช้การสร้างวิวในรูปของออนโทโลจีรวมกำกับลงบนแบบจำลองข้อมูลที่นำมารวมแต่ละชุด เพื่อทำหน้าที่เป็นแมปปิงระหว่างแบบจำลองข้อมูลทั้งสองชนิด

การทำงานของแนวคิดในการรวมแบบจำลองข้อมูลเชิงวัตถุแบบรวมหลายออนโทโลจีประกอบด้วย การให้ผู้เป็นเจ้าของข้อมูลแต่ละคนสร้างโลคัลออนโทโลจีขึ้นหนึ่งชุดเพื่ออธิบายความหมายทั้งหมดของแบบจำลองข้อมูลของตน แล้วให้ผู้ทำการรวมแบบจำลองข้อมูลนำโลคัลออนโทโลจีคู่ที่จะรวมมาเปรียบเทียบและสร้างออนโทโลจีรวมด้วยโปรแกรมการรวมออนโทโลจี จากนั้นจึงนำออนโทโลจีรวมที่ได้ไปกำกับลงบนแบบจำลองข้อมูลเชิงวัตถุแต่ละชุด พร้อมทั้งนำไปใช้สร้างแผ่นแบบเพื่อเป็นตัวกลางสำหรับผู้ใช้ในการสอบถามข้อมูลจากแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดที่นำมารวม

ทั้งนี้ เมื่อพิจารณาจากขั้นตอนการทำงานโดยรวมแล้วจะพบได้ว่า แนวคิดในการนำออนโทโลจีมาใช้ในการรวมข้อมูลแบบรวมหลายออนโทโลจีของผู้วิจัยค่อนข้างใกล้เคียงกับแนวคิดแบบออนโทโลจี

เดียวมากที่สุด เนื่องจากทั้งสองแนวคิดนี้ต่างต้องการหลีกเลี่ยงความยุ่งยากซับซ้อน ทั้งในส่วนการสร้างระบบการรวมข้อมูล และส่วนการสอบถามข้อมูลจากระบบการรวมที่ได้ ข้อแตกต่างสำคัญระหว่างสองแนวคิดนี้ก็คือ แนวคิดของผู้วิจัยยังมุ่งเน้นไปที่การกำหนดวิธีการที่ชัดเจนในการสร้างออนโทโลจีรวม ซึ่งทำหน้าที่เหมือนกับโกลบอลออนโทโลจีในแนวคิดแบบออนโทโลจีเดียว เพื่อให้ออนโทโลจีรวมที่ได้มีความเหมาะสมมากที่สุด

นอกจากนั้น ผู้วิจัยยังได้เลือกใช้การสร้างโลคัลออนโทโลจีในการแสดงแทนความหมายทั้งหมดของแบบจำลองข้อมูลเชิงวัตถุแต่ละชุดตามแนวคิดแบบหลายออนโทโลจีและแบบผสม เพื่อให้แบบจำลองข้อมูลแต่ละชุดได้รับการอธิบายอย่างครบถ้วน ชัดเจน และเป็นอิสระจากแบบจำลองข้อมูลชุดอื่น ทำให้เกิดการกระจายงานจากผู้ทำการรวมแบบจำลองข้อมูลไปยังเจ้าของข้อมูลแต่ละชุด ซึ่งจะช่วยลดภาระงานของผู้ทำการรวมแบบจำลองข้อมูลได้อีกด้วย

อย่างไรก็ตาม สำหรับการนำออนโทโลจีมาใช้ในการรวมแบบจำลองข้อมูลเชิงวัตถุตามแนวคิดแบบรวมหลายออนโทโลจีที่ผู้ทำวิจัยได้เสนอขึ้นนี้ ผู้วิจัยพบว่าสามารถทำงานได้ดีกับการรวมฐานข้อมูลที่มีการกำหนดโครงสร้างของข้อมูลไว้แล้ว แต่ไม่เหมาะสมสำหรับฐานข้อมูลที่มีการเปลี่ยนแปลงโครงสร้างบ่อย และข้อมูลที่ไม่มีโครงสร้างแน่นอน เช่น ข้อมูลบนเว็บเพจ

7.1.2 ขั้นตอนวิธีการรวมออนโทโลจีโดยการสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาส

ในวิทยานิพนธ์ฉบับนี้ ผู้วิจัยได้พัฒนาขั้นตอนวิธีการรวมออนโทโลจีโดยการสร้างออนโทโลจีรวมจากการวิเคราะห์ความสัมพันธ์ระหว่างคลาสขึ้นจากวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุโดยใช้วิทยา-การศึกษาสำนึกและการวิเคราะห์ความสัมพันธ์ระหว่างคลาสในงานวิจัย [20] มีรายละเอียดของขั้นตอนวิธีการรวมตามที่นำเสนอไว้ในบทที่ 4 ซึ่งประกอบด้วยส่วนการเปรียบเทียบออนโทโลจี (ได้แก่ การเปรียบเทียบความหมาย การเปรียบเทียบคลาสของแอททริบิวต์และการเปรียบเทียบคลาสของคลาส) และส่วนการรวมออนโทโลจี (ได้แก่ การรวมความหมาย การรวมคลาสของแอททริบิวต์และการรวมคลาสของคลาส) ขั้นตอนวิธีดังกล่าวนี้ได้รับการออกแบบให้สามารถนำไปใช้รวมโลคัลออนโทโลจีสองชุดที่สร้างขึ้นตามวิธีการที่ผู้วิจัยกำหนด เพื่อสร้างออนโทโลจีรวมซึ่งจะนำไปใช้เป็นตัวกลางในการสอบถามข้อมูลจากแบบจำลองข้อมูลเชิงวัตถุทั้งสองชุดที่นำมารวม โดยออนโทโลจีรวมที่ได้ต้องสามารถปรับความต่างแบบ และจัดการกับความขัดแย้งระหว่างโลคัลออนโทโลจีแต่ละชุดที่นำมารวมได้

การเลือกนำวิธีการในงานวิจัย [20] มาประยุกต์ใช้ต่อเป็นขั้นตอนวิธีการสร้างออนโทโลจีรวมในวิทยานิพนธ์ฉบับนี้เนื่องจากผู้วิจัยต้องการสร้างขั้นตอนวิธีการรวมออนโทโลจีที่ทำงานแบบกึ่งอัตโนมัติ ซึ่งสามารถเปรียบเทียบและรวมออนโทโลจีหนึ่งคู่ที่แสดงแทนความหมายทั้งหมดของแบบจำลองข้อมูลเชิงวัตถุได้อย่างถูกต้อง โดยมีวิธีการทำงานที่ไม่ยุ่งยากซับซ้อนมากนักและช่วยลดภาระงานของผู้ทำการรวมแบบจำลองข้อมูลเชิงวัตถุ เพื่อนำมาใช้ประกอบการทำงานของแนวคิดในการนำออนโทโลจีมาใช้ใน

การรวมแบบจำลองข้อมูลเชิงวัตถุแบบรวมหลายออนโทโลจีที่ออกแบบขึ้น ทั้งนี้ ผลลัพธ์ที่ได้จากการรวมออนโทโลจีด้วยขั้นตอนวิธีของผู้วิจัยมีความถูกต้อง ออนโทโลจีรวมที่ได้มีโครงสร้างตรงตามต้องการ ดังที่ได้แสดงผลการทดสอบไว้ในบทที่ 6 และยังสามารถนำไปใช้งานในการรวมออนโทโลจีขนาดใหญ่ได้

เนื่องจากวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุในงานวิจัย [20] ได้รับการทดสอบด้วยเมตริกซ์เชิงวัตถุแล้วว่าแบบจำลองข้อมูลรวมที่ได้มีลักษณะที่ดีตามหลักการออกแบบแบบจำลองข้อมูลเชิงวัตถุ ดังนั้นการนำวิธีการดังกล่าวมาใช้ส่งผลให้ผลลัพธ์ที่ได้จากขั้นตอนวิธีการรวมนี้อยู่ในรูปของแบบจำลองข้อมูลที่ดีอีกด้วย

นอกจากนั้น เมื่อพิจารณาจากงานวิจัยเชิงสำรวจเกี่ยวกับแนวคิดในการนำออนโทโลจีมาใช้เพื่อการรวมเชิงความหมาย [13] งานวิจัยดังกล่าวได้จำแนกส่วนประกอบของออนโทโลจีที่นิยมใช้ในการเปรียบเทียบออนโทโลจีไว้ดังนี้

- ชื่อคลาส
- ลำดับชั้นของคลาส
- ความสัมพันธ์ (โดเมน เรนจ์และข้อจำกัดของความสัมพันธ์)
- อินสแตนซ์ของคลาส
- คำจำกัดความของคลาส

เมื่อนำมาเปรียบเทียบกับขั้นตอนวิธีในส่วนการเปรียบเทียบออนโทโลจีของวิทยานิพนธ์ฉบับนี้ จะพบว่า ส่วนการเปรียบเทียบความหมายตรงกับการเปรียบเทียบชื่อคลาส และคำจำกัดความของคลาส การเปรียบเทียบคลาสของแอททริบิวต์ตรงกับการเปรียบเทียบชื่อคลาส ความสัมพันธ์และคำจำกัดความของคลาส ส่วนการเปรียบเทียบคลาสของคลาสตรงกับการเปรียบเทียบชื่อคลาส ลำดับชั้นของคลาส ความสัมพันธ์และคำจำกัดความของคลาส ในขณะที่การเปรียบเทียบอินสแตนซ์ของคลาสนั้น แม้ว่าจะไม่ปรากฏโดยตรงแต่อาจเกิดขึ้นในส่วนการเปรียบเทียบความหมายเนื่องจากอินสแตนซ์บางชุดอาจถูกระบุไว้เป็นความหมายของออนโทโลจี ซึ่งแสดงให้เห็นว่าวิธีการเปรียบเทียบที่เลือกใช้ในขั้นตอนวิธีการสร้างออนโทโลจีรวมในวิทยานิพนธ์นี้ครอบคลุมถึงส่วนประกอบต่างๆ ในออนโทโลจีได้ดีในระดับหนึ่ง

7.1.3 การอธิบายแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจี

การนำอัปเปอร์ออนโทโลจีมาใช้เป็นออนโทโลจีพื้นฐานเพื่อนำมาใช้สร้างโลคัลออนโทโลจีที่สามารถทำงานร่วมกันระหว่างโดเมนเป็นวิธีการที่ช่วยควบคุมให้โลคัลออนโทโลจีทุกชุดที่สร้างขึ้นเป็นไปในแนวทางเดียวกัน มีโครงสร้างและแนวคิดหลักตรงกัน ส่งผลให้การใช้งานโลคัลออนโทโลจีทั้งหมดร่วมกัน (เช่น การเปรียบเทียบระหว่างออนโทโลจีหลายชุด) เป็นไปได้โดยง่าย

นอกจากนั้น การสร้างโลคัลออนโทโลจีที่อธิบายแบบจำลองข้อมูลเชิงวัตถุมาใช้ในการรวมแบบจำลองข้อมูลยังช่วยให้เจ้าของแบบจำลองข้อมูลมั่นใจได้ว่า ผลจากการรวมแบบจำลองข้อมูลเชิง

วัตถุที่ได้จะไม่กระทบกับการใช้งานแบบจำลองข้อมูลเดิม เนื่องจากในขั้นตอนการรวม ผู้ทำการรวมไม่ได้เกี่ยวข้องหรือดัดแปลงแก้ไขใดๆ กับแบบจำลองข้อมูลเดิมเลย

ผู้วิจัยได้กำหนดให้เจ้าของแบบจำลองข้อมูลแต่ละคนสร้างโลคัลออนโทโลจีเพื่ออธิบายแบบจำลองข้อมูลเชิงวัตถุของตนตามโครงสร้างของอัปเปอร์ออนโทโลจีที่ผู้วิจัยออกแบบไว้ตามวิธีการที่ได้แสดงไว้ในบทที่ 3 ซึ่งประโยชน์ที่ได้รับนอกเหนือจากที่โลคัลออนโทโลจีทุกชุดอธิบายความหมายของแบบจำลองข้อมูลแต่ละชุดในแนวทางเดียวกันแล้ว การสร้างโลคัลออนโทโลจีตามโครงสร้างของอัปเปอร์ออนโทโลจียังช่วยให้ความหมายของแบบจำลองข้อมูลเชิงวัตถุที่ปรากฏในโลคัลออนโทโลจีเพียงพอต่อการจำแนกความขัดแย้งระหว่างโลคัลออนโทโลจี และสอดคล้องกับขั้นตอนวิธีการรวมออนโทโลจีที่ผู้วิจัยได้พัฒนาขึ้น โดยการออกแบบอัปเปอร์ออนโทโลจีได้คำนึงถึงประเด็นดังกล่าวเป็นสำคัญ

อย่างไรก็ตาม ปัญหาหนึ่งที่อาจเกิดขึ้นจากขั้นตอนการอธิบายแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีและกระทบต่อผลการรวมออนโทโลจีที่ได้คือ การเลือกใช้คำศัพท์ในการสร้างโลคัลออนโทโลจีของเจ้าของแบบจำลองข้อมูลเชิงวัตถุแต่ละชุด ซึ่งในบางครั้งอาจขาดความเข้าใจด้านภาษา ส่งผลให้คำศัพท์ที่ใช้สื่อความหมายผิดจากที่ต้องการ หรือคำศัพท์ดังกล่าวไม่ตรงกับความเป็นจริง เมื่อนำไปเปรียบเทียบกับโลคัลออนโทโลจีอีกชุดก็จะทำให้ผลการเปรียบเทียบผิดเพี้ยนตามไปด้วย

ความถูกต้องของความหมายของโลคัลออนโทโลจีที่สร้างขึ้นเป็นปัจจัยสำคัญซึ่งจะส่งผลให้การสร้างออนโทโลจีรวมเป็นไปอย่างถูกต้องเหมาะสม ดังนั้นผู้เป็นเจ้าของแบบจำลองข้อมูลแต่ละคน รวมถึงผู้ทำการรวมแบบจำลองข้อมูลจำเป็นต้องสร้างโลคัลออนโทโลจีให้สามารถแสดงแทนความหมายของแบบจำลองข้อมูลเชิงวัตถุแต่ละชุดให้มีความครบถ้วน ชัดเจนมากที่สุด และเลือกใช้คำศัพท์ต่างๆ ด้วยความระมัดระวัง

ทั้งนี้ วิทยานิพนธ์ฉบับนี้ยังได้กำหนดให้แบบจำลองข้อมูลเชิงวัตถุที่นำมาสร้างเป็นโลคัลออน-โทโลจีซึ่งจะนำมารวมควรเป็นแบบจำลองข้อมูลเชิงวัตถุที่อธิบายข้อมูลเรื่องเดียวกันหรือมีความเกี่ยวข้องกัน เพื่อให้สร้างวิทยานิพนธ์ฉบับนี้ในขณะที่ออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีคู่ที่นำมารวมนั้น จะช่วยลดระยะห่าง (Distance) ระหว่างความแตกต่างของโลคัลออนโทโลจีทั้งสองชุดลงและให้ผลการรวมที่ดียิ่งขึ้น

7.1.4 การกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูลเชิงวัตถุเพื่อใช้ในการสอบถามข้อมูล

การกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูลเชิงวัตถุเพื่อใช้ในการสอบถามข้อมูลทำหน้าที่เป็นแมปปิงระหว่างแบบจำลองข้อมูลเชิงวัตถุแต่ละชุด กับออนโทโลจีรวมซึ่งเป็นตัวกลางในสร้างข้อคำถามเพื่อใช้ในการสอบถามเชิงความหมายจากผู้ใช้ ส่งผลให้สามารถแปลงข้อคำถามที่ผู้ใช้กำหนดเป็นข้อคำถามที่ใช้งานจริงกับฐานข้อมูลแต่ละชุดได้ แม้ว่าชื่อฟิลด์ข้อมูลในแบบจำลองข้อมูลทั้งสองชุดจะต่างกันก็ตาม

ผู้วิจัยเลือกกำกับออนโทโลจีรวมลงบนแบบจำลองข้อมูลเชิงวัตถุแต่ละชุดในรูปของวิวซึ่งสร้างขึ้นบนฐานข้อมูลเหล่านั้นเพื่อแมประหว่างคลาสภายในออนโทโลจีรวมไปยังฟิลด์ต่างๆ ภายในฐานข้อมูลเชิงวัตถุ โดยไม่กระทบต่อโครงสร้างและการใช้งานฐานข้อมูลที่มีอยู่เดิม โครงสร้างของวิวที่สร้างขึ้นจะถูกกำหนดขึ้นตามออนโทโลจีรวมเป็นสำคัญ ในขณะที่การแมปไปยังฟิลด์ในฐานข้อมูลเชิงวัตถุสามารถแบ่งได้เป็น 7 ประเภทตามผลการรวมออนโทโลจีที่ได้ คือ

- หากคลาสของออนโทโลจีรวมตรงกับแอททริบิวต์ในแบบจำลองข้อมูลเชิงวัตถุ จะแมปจากคลาสของออนโทโลจีรวมไปยังฟิลด์ที่ตรงกันโดยตรง
- หากคลาสของออนโทโลจีรวมตรงกับแอททริบิวต์ในแบบจำลองข้อมูลเชิงวัตถุ แต่ค่าข้อมูลไม่ตรงกัน จะต้องสร้างฟังก์ชันการแปลงค่าข้อมูลให้ตรงกัน ก่อนแมปจากคลาสของออน-โทโลจีรวมไปยังฟิลด์ที่ตรงกันผ่านฟังก์ชันที่สร้างขึ้น
- หากคลาสของออนโทโลจีรวมตรงกับแอททริบิวต์ในแบบจำลองข้อมูลเชิงวัตถุ แต่หน่วยข้อมูลไม่ตรงกัน จะต้องสร้างฟังก์ชันการแปลงค่าข้อมูลให้หน่วยตรงกัน ก่อนแมปจากคลาสของออนโทโลจีรวมไปยังฟิลด์ที่ตรงกันผ่านฟังก์ชันที่สร้างขึ้น
- หากคลาสของออนโทโลจีรวมตรงกับแอททริบิวต์ในแบบจำลองข้อมูลเชิงวัตถุ แต่ชนิดข้อมูลไม่ตรงกัน จะต้องสร้างฟังก์ชันการแปลงชนิดข้อมูลให้หน่วยตรงกัน ก่อนแมปจากคลาสของออนโทโลจีรวมไปยังฟิลด์ที่ตรงกันผ่านฟังก์ชันที่สร้างขึ้น
- หากคลาสของออนโทโลจีรวมตรงกับกลุ่มของแอททริบิวต์ในแบบจำลองข้อมูลเชิงวัตถุ จะแมปจากคลาสของออนโทโลจีรวมไปยังฟิลด์ของกลุ่มของแอททริบิงต์เหล่านั้นโดยอาศัยฟังก์ชัน CONCAT (Concatenate)
- หากไม่มีแอททริบิวต์ใดในแบบจำลองข้อมูลเชิงวัตถุตรงกับคลาสของออนโทโลจีรวม จะแมปจากคลาสของออนโทโลจีรวมไปยัง “—“ เพื่อแสดงว่าไม่มีข้อมูลดังกล่าว
- หากคลาสของออนโทโลจีรวมที่ได้รับการกำหนดเงื่อนไขจากขั้นตอนการอนุมาน ตรงกับแอททริบิวต์ในแบบจำลองข้อมูลเชิงวัตถุ จะแมปจากคลาสของออนโทโลจีรวมไปยังฟิลด์ที่ตรงกันโดยกำหนดเงื่อนไขของข้อคำถามในการแมปตามเงื่อนไขของคลาสนั้น

การแมประหว่างออนโทโลจีรวมไปยังแบบจำลองข้อมูลเชิงวัตถุโดยการสร้างวิวกำกับลงบนฐานข้อมูลเชิงวัตถุแต่ละชุดตามวิธีการนี้ ผู้วิจัยได้ทดสอบผลการสอบถามข้อมูลผ่านระบบการรวมแบบจำลองข้อมูลเชิงวัตถุ เปรียบเทียบกับการสอบถามโดยตรงจากฐานข้อมูลแต่ละชุดดังรายละเอียดในบทที่ 6 และพบว่าผลการสอบถามที่ได้มีความถูกต้อง ครบถ้วน ถึงแม้ว่าข้อมูลบางฟิลด์ถูกเปลี่ยนแปลงจากการทำงานของฟังก์ชันที่สร้างขึ้นในวิว เพื่อปรับให้ผลลัพธ์ที่ได้จากฐานข้อมูลทั้งสองชุดตรงกัน ซึ่งเป็นส่วนของการแก้ไขความขัดแย้งระหว่างข้อมูล โดยที่ความหมายของข้อมูลที่ได้จากการสอบถามตรงตามข้อคำถามที่กำหนด และไม่ผิดไปจากความหมายเดิมของข้อมูลแต่อย่างใด

7.1.5 การเปรียบเทียบระหว่างแนวคิดแบบรวมหลายออนโทโลจีกับแนวคิดอื่นที่มีอยู่เดิม

เพื่อเปรียบเทียบระหว่างแนวคิดในการนำออนโทโลจีมาใช้ในการรวมแบบจำลองข้อมูลแบบรวมหลายออนโทโลจีกับแนวคิดอื่นๆ ที่มีอยู่เดิม ผู้วิจัยได้ทำการเปรียบเทียบทั้งสี่แนวคิดตามหัวข้อการเปรียบเทียบที่ใช้ในงานวิจัย [5] ผลการเปรียบเทียบที่ได้แสดงไว้ในตารางที่ 7.1

จากการเปรียบเทียบจะพบว่า แนวคิดแบบรวมหลายออนโทโลจีได้ประโยชน์จากการนำข้อดีของแนวคิดอื่นๆ ส่งผลให้แนวคิดนี้ง่ายต่อการนำไปใช้งาน และได้จำกัดความความต่างแบบเชิงความหมายโดยกำหนดให้ทุกแหล่งข้อมูลถูกแมปเข้ากับออนโทโลจีรวม ซึ่งไม่กระทบต่อความหมายเดิมของข้อมูลแต่ละชุด ในขณะเดียวกันแนวคิดแบบรวมหลายออนโทโลจีพยายามลดความซับซ้อนในส่วนของวิธีการเปรียบเทียบระหว่างออนโทโลจีแต่ละชุด โดยอาศัยการอธิบายความหมายและความสัมพันธ์ระหว่างออนโทโลจีแต่ละชุดให้ชัดเจนก่อนจะทำการเปรียบเทียบ และการเปรียบเทียบจะเกิดขึ้นเพียงครั้งเดียวคือเมื่อสร้างระบบการรวมแบบจำลองข้อมูลเท่านั้น จึงช่วยลดการทำงานของระบบในขั้นตอนการสอบถามข้อมูลเมื่อได้รับข้อคำถามจากผู้ใช้

อย่างไรก็ตาม แนวคิดแบบรวมหลายออนโทโลจีมีความยืดหยุ่นต่อการเพิ่ม/ ลดแหล่งข้อมูลที่นำมารวมน้อยกว่าแนวคิดแบบผสม แต่ใกล้เคียงกับแนวคิดแบบหลายออนโทโลจี เนื่องจากออนโทโลจีรวมที่ได้จะเปลี่ยนแปลงตามผลการรวมออนโทโลจีทั้งหมดทุกชุด ทำให้ต้องสร้างออนโทโลจีรวมใหม่ทุกครั้งที่มีการเปลี่ยนแปลงแหล่งข้อมูลที่นำมารวม ซึ่งการอภิปรายถึงปัญหาและข้อจำกัดของแนวคิดแบบรวมหลายออนโทโลจีได้แสดงไว้ในหัวข้อที่ 7.2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 7.1 การเปรียบเทียบระหว่างแนวคิดแบบรวมหลายออนโทโลจีกับแนวคิดอื่นที่มีอยู่เดิม

แนวคิดที่ใช้ / การเปรียบเทียบ	แบบออนโทโลจีเดียว	แบบหลายออนโทโลจี	แบบผสม	แบบรวมหลายออนโทโลจี
ความยุ่งยากในการนำไปใช้งาน	ง่าย—ตรงไปตรงมา	ยุ่งยาก—ค่าใช้จ่ายสูง	ปานกลาง—สมเหตุสมผล	ง่าย—ลดภาระของผู้ทำการรวม โดยใช้การกระจายงานบางส่วนให้ผู้เป็นเจ้าของแหล่งข้อมูล
ความต่างแบบเชิงความหมาย	เหมือนกันทุกแหล่งข้อมูล	สามารถจัดการกับความต่างแบบได้ โดยให้ผู้ใช้เลือกมุมมองของข้อมูลที่ต้องการ	สามารถจัดการกับความต่างแบบได้ โดยให้ผู้ใช้เลือกมุมมองของข้อมูลที่ต้องการ	เหมือนกันทุกแหล่งข้อมูล โดยยึดออนโทโลจีรวมเป็นตัวกลาง
การเพิ่ม/ลดแหล่งข้อมูลที่นำมารวม	ต้องแก้ไขที่โกลบอลออนโทโลจี	สร้างโลคัลออนโทโลจีใหม่เพื่ออธิบาย และเชื่อมโยงไปยัง โลคัลออนโทโลจีเดิมทุกชุด	สร้างโลคัลออนโทโลจีใหม่เพื่ออธิบาย	สร้างโลคัลออนโทโลจีใหม่เพื่ออธิบาย และสร้างออนโทโลจีรวมใหม่ด้วยโปรแกรมการรวมที่มีอยู่แล้ว
วิธีการเปรียบเทียบระหว่างออนโทโลจีแต่ละชุด	ไม่มีการเปรียบเทียบออนโทโลจี	ยาก—เพราะไม่มีการเก็บคำศัพท์ที่ออนโทโลจีแต่ละชุดใช้ร่วมกันไว้ก่อน	ง่าย—เพราะได้รวบรวมคำศัพท์ที่ออนโทโลจีใช้งานร่วมกันไว้ก่อนแล้ว	ง่าย—เพราะออนโทโลจีทุกชุดได้รับการอธิบายความหมายและความสัมพันธ์ระหว่างกันไว้อย่างชัดเจน

7.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย

- เนื่องจากวิทยานิพนธ์นี้ประยุกต์ใช้ขั้นตอนวิธีการรวมออนโทโลจีมาจากขั้นตอนวิธีการรวมแบบจำลองข้อมูลที่ใช้ในงานวิจัย [20] ดังนั้นข้อจำกัดที่ปรากฏในงานวิจัยดังกล่าวยังคงปรากฏในวิทยานิพนธ์นี้ เช่น ในส่วนของการเปรียบเทียบจะไม่มีการพิจารณาถึงเมท็อดของวัตถุในแบบจำลองข้อมูลเชิงวัตถุแต่อย่างใด
- จากขั้นตอนการเปรียบเทียบโลคัลออนโทโลจีเพื่อการสร้างเป็นออนโทโลจีรวมนั้น จะเห็นได้ว่าการจะได้ออนโทโลจีรวมที่ถูกต้องจำเป็นต้องอาศัยการเพิ่มความหมาย การระบุคำที่มีความหมายเหมือนกัน และคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกัน ดังนั้นหากต้องการนำเฟรมเวิร์คนี้ไปใช้งานจริง ผู้ทำการรวมแบบจำลองข้อมูลต้องให้ความสำคัญกับความหมายของข้อมูลด้วย
- การทำงานของโปรแกรมเปรียบเทียบและรวมออนโทโลจีเป็นแบบกึ่งอัตโนมัติ อย่างไรก็ตาม หากผู้ทำการรวมแบบจำลองข้อมูลพบว่า ออนโทโลจีรวมที่แสดงแบบจำลองข้อมูลรวมที่ได้ไม่เป็นไปตามที่ต้องการ ผู้ทำการรวมแบบจำลองข้อมูลควรจะกลับไปเริ่มที่ขั้นตอนของการเพิ่มความหมายของแบบจำลองข้อมูลลงในโลคัลออนโทโลจีใหม่ และทดลองรวมโลคัลออนโทโลจีคู่ดังกล่าวอีกครั้ง
- ต้นแบบโปรแกรมการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นสามารถรวมโลคัลออนโทโลจีได้ครั้งละสองชุดเท่านั้น ดังนั้น หากต้องการรวมแบบจำลองข้อมูลเชิงวัตถุสามชุด ผู้ทำการรวมแบบจำลองข้อมูลจำเป็นต้องเลือกรวมโลคัลออนโทโลจีสองครั้งคือ รวมโลคัลออนโทโลจีที่อธิบายแบบจำลองข้อมูลคู่หนึ่งก่อน แล้วนำออนโทโลจีรวมที่ได้ไปใช้เป็นโลคัลออนโทโลจีอีกชุดที่จะนำไปรวมกับโลคัลออนโทโลจีที่อธิบายแบบจำลองข้อมูลชุดที่เหลือ
- ในการรวมออนโทโลจีมากกว่าสองชุดด้วยขั้นตอนวิธีที่ผู้วิจัยพัฒนาขึ้นนั้น ลำดับก่อนหลังในการเลือกรวมโลคัลออนโทโลจีแต่ละคู่จะส่งผลต่อออนโทโลจีรวมที่ได้ ดังนั้นผู้ทำการรวมจำเป็นต้องเลือกว่าจะรวมโลคัลออนโทโลจีคู่ใดก่อน-หลัง ซึ่งลำดับที่ต่างกันอาจทำให้ผลการรวมออนโทโลจีต่างกันด้วย
- วิธีการอธิบายแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีที่นำเสนอในวิทยานิพนธ์ฉบับนี้เป็นการสร้างออนโทโลจีอย่างง่ายเพื่ออธิบายความหมายของแบบจำลองข้อมูลเชิงวัตถุ ซึ่งต้องการให้ออนโทโลจีที่ได้รองรับการทำงานของขั้นตอนวิธีในการสร้างออนโทโลจีรวมที่พัฒนาขึ้นเป็นหลัก และไม่ต้องการให้ผู้เป็นเจ้าของแบบจำลองข้อมูลจำเป็นต้องมีความรู้เรื่องออนโทโลจีมากนัก ผู้วิจัยจึงได้เน้นไปที่ความถูกต้องของความหมาย และความสัมพันธ์ระหว่างคลาสในออนโทโลจีเป็นสำคัญ แต่ยังมิได้ใช้ความสามารถทั้งหมดที่มีของภาษาอาวล์ในการแสดงแทนออนโทโลจี ตัวอย่างเช่น การกำหนดความหมายของคลาสด้วยเงื่อนไขในรูปของนิพจน์ภาษาอาวล์ หรือการกำหนดอินสแตนซ์ของคลาส ซึ่งอาจนำมาใช้

แทนการสร้างคลาสของข้อกำหนดบูรณภาพ หรือคลาสของความหมายได้ รวมไปถึงการใช้ความสามารถในการอนุมานของภาษาอาวล์อีกด้วย

7.3 ข้อเสนอแนะและแนวทางการทำวิจัยในอนาคต

- ในส่วนของการปรับปรุงขั้นตอนวิธีการรวมออนโทโลจีนั้น สามารถเพิ่มเติมการวิเคราะห์ในส่วนอื่นๆ เช่น ความสัมพันธ์รูปแบบอื่นที่ต้องการพิจารณาจากแบบจำลองข้อมูลเชิงวัตถุ การวิเคราะห์ด้านคำศัพท์ที่ใช้ในการเปรียบเทียบความหมาย และอื่นๆ ได้อีกมากเพื่อให้สามารถนำขั้นตอนวิธีไปใช้ได้กว้างขวางขึ้น
- ในทำนองเดียวกัน หากมีการเปลี่ยนแปลงขั้นตอนวิธีการรวมออนโทโลจี ก็อาจทำการปรับเปลี่ยนอัปเปอร์ออนโทโลจีตามความเหมาะสมเพื่อให้ได้โลคัลออนโทโลจีที่สามารถอธิบายข้อมูลได้ตรงกับขั้นตอนวิธีที่เลือกใช้ ซึ่งจะเห็นได้ว่าในการเปลี่ยนแปลงขั้นตอนวิธีการรวมออนโทโลจีจะต้องคำนึงถึงโครงสร้างของอัปเปอร์ออนโทโลจีที่ใช้ด้วย
- หากผู้เป็นเจ้าของแบบจำลองข้อมูลแต่ละชุดมีความรู้ ความเข้าใจเรื่องออนโทโลจีอยู่ก่อนแล้ว การสร้างโลคัลออนโทโลจีอาจจะกำหนดให้ใช้ความสามารถของภาษาอาวล์ให้มากขึ้น ซึ่งจะส่งผลให้การรวมออนโทโลจีเป็นการรวมในเชิงความหมายมากกว่าเดิม เนื่องจากสามารถนำความสามารถในการอนุมานของออนโทโลจีมาใช้ได้อย่างครบถ้วนยิ่งขึ้น
- ผู้ทำการรวมแบบจำลองข้อมูลควรจะมีความรู้ถึงภาพรวม และรายละเอียดของแบบจำลองข้อมูลที่นำมารวม เพื่อให้สามารถระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีที่นำมาใช้รวมได้ ทั้งนี้ผู้ทำการรวมไม่จำเป็นต้องเป็นคนเพียงคนเดียว แต่อาจเป็นทีมผู้เชี่ยวชาญที่เข้าใจถึงแต่ละส่วนของแบบจำลองข้อมูล
- ในส่วนของการอธิบายแบบจำลองข้อมูลเชิงวัตถุด้วยออนโทโลจีนั้นจะเห็นว่าในวิทยานิพนธ์นี้ยังต้องอาศัยเจ้าของฐานข้อมูลมาทำหน้าที่สร้างโลคัลออนโทโลจีของตนเองอยู่ ซึ่งอาจแก้ไขได้โดยการสร้างโปรแกรมเพื่อดึงแบบจำลองข้อมูลที่มีอยู่แล้วมาสร้างเป็นออนโทโลจีได้โดยอัตโนมัติ
- ในทำนองเดียวกัน ในส่วนของการระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีที่นำมารวม ซึ่งในวิทยานิพนธ์ฉบับนี้ยังต้องอาศัยผู้ทำการรวมออนโทโลจีมาทำหน้าที่ระบุความสัมพันธ์ต่างๆ ซึ่งในอนาคตอาจลดภาระส่วนนี้ลงโดยนำเครื่องมือด้านภาษาที่มีอยู่ เช่น คลังศัพท์ (Lexicon) มาใช้วิเคราะห์ความสัมพันธ์ระหว่างคำศัพท์ต่างๆ แทน
- ผู้เป็นเจ้าของแบบจำลองข้อมูล และผู้ทำการรวมแบบจำลองข้อมูลจำเป็นต้องสร้างโลคัลออนโทโลจีให้สามารถแสดงแทนความหมายของแบบจำลองข้อมูลเชิงวัตถุแต่ละชุดให้มีความครบถ้วน ชัดเจนมากที่สุด และเลือกใช้คำศัพท์ต่างๆ ด้วยความระมัดระวัง ตัวอย่างเช่น เมื่อต้องการกำหนดค่าความหมายของ ‘faculty’ ก็ควรกำหนดให้ค่าความหมายทุกคำเป็นชื่อคณะ เช่น ‘science’, ‘accountancy’ และ ‘engineering’ มิใช่ละ

หรือย่อหรือปรับหน้าที่ของคำตามความสะดวกของผู้สร้าง เนื่องจากความผิดพลาดจากการสร้างออนโทโลจีในตอนแรกจะส่งผลกระทบโดยตรงต่อความถูกต้องของออนโทโลจีรวมที่ได้

- จากการทดลองรวมออนโทโลจีมากกว่าสองชุดด้วยขั้นตอนวิธีที่ในวิทยานิพนธ์ฉบับนี้ ผู้วิจัยพบว่าลำดับในการรวมมีผลต่อออนโทโลจีรวมที่ได้ ดังนั้น งานวิจัยในอนาคตควรจะมีข้อแนะนำให้ผู้ทำการรวมทราบว่า หากจำเป็นต้องรวมออนโทโลจีมากกว่าสองชุด ลำดับการควรเป็นอย่างไร
- เนื่องจากระบบสารสนเทศขององค์กรเริ่มได้รับการขยายความสามารถให้ทำงานเชิงความหมายได้ดียิ่งขึ้น ดังนั้น องค์กรต่างๆ สามารถนำแนวคิดเรื่องการใช้ออนโทโลจีในการอธิบายแบบจำลองข้อมูลและข้อมูลต่างขององค์กร เพื่อรองรับการใช้งานทั้งการรวมแบบจำลองข้อมูล และงานสารสนเทศด้านอื่นต่อไปในอนาคต

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Elmasri, R. and Navathe, S. B. Fundamentals of database systems. 4th ed. Reading, MA: Addison-Wesley, 2003.
2. Lenzerini, M. Data integration: A theoretical perspective. Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2002), Madison, WI, June 3-5, 2002, pp. 233-246.
3. Ram, S. and Park, J. Semantic conflict resolution ontology (SCROL): An ontology for detecting and resolving data and schema-level semantic conflicts. IEEE Transactions on Knowledge and Data Engineering 16, 2, (February 2004): 189-202.
4. Gruber, T. R. A translation approach to portable ontology specifications. Knowledge Acquisition 5, 2, (June 1993): 199-220.
5. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hübner, S. Ontology–based integration of information — A survey of existing approaches. Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Workshop: Ontologies and Information Sharing, Seattle, WA, August 4-6, 2001, pp. 108-117.
6. Arens, Y., Hsu, C. -N. and Knoblock, C. A. Query processing in the SIMS information mediator. In Tate, A. (ed.), Advanced Planning Technology, pp. 61--69. Menlo Park, CA: AAAI Press, 1996.
7. Cruz, I. F. and Rajendran, A. Semantic data integration in hierarchical domains. IEEE Intelligent Systems 18, 2, (2003): 66-73.
8. Mena, E. Illarramendi, A. Kashyap, V. and Sheth, A. P. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. DISTRIBUTED AND PARALLEL DATABASES 8, 2, (April 2000): 223-271.
9. Goh, C. H., Bressan, S., Madnick, S. and Siegel, M. Context interchange: New features and formalisms for the intelligent integration of information. ACM Transactions on Information Systems 17, 3, (July 1999): 270-293.
10. Wache, H., Scholz, T., Stieghahn, H. and Konig-Ries, B. An integration method for the specification of rule-oriented mediators. Proceedings of International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), Kyoto, Japan, November 28-30, 1999, pp. 109-112.

11. Pinto, H. S. and Martims, J. P. A methodology for ontology integration. Proceedings of the 1st International Conference on Knowledge Capture (K-CAP'01), British Columbia, Canada, October 22 – 23, 2001, pp. 131-138.
12. D'Antonio, F., Lenzerini, M., Jeusfeld, M. and Johannesson, P. Deliverable D8.1: State of the art and state of the practice including initial possible research orientations [PDF file]. INTEROP: Interoperability Research for Networked Enterprises Applications and Software, Version 1.2, November 18, 2004. Available from: http://interop-noe.org/deliv/D8/attach/D8.1.pdf [February 20, 2006]
13. Noy, N. F. Semantic integration: A survey of ontology-based approaches. ACM SIGMOD Record, Special Issue on Semantic Integration 33, 4, (December 2004): 65-70.
14. Zipparo, M., Turgut, D. and Bölöni, L. A survey of merging techniques and tools for ontologies. Proceedings of The 2004 International Conference on Information and Knowledge Engineering (IKE'04), Las Vegas, NV, June 21-24, 2004.
15. Abels, S., Haak, L. and Hahn, A. Identifying ontology integration methods and their applicability in the context of product classification and knowledge integration task. Technical Report of Department of Business Information Systems, University of Oldenburg, Oldenburg, Germany, Report No. WI-OL-TR-01-2005, October, 2005.
16. Noy, N. F. and Musen, M. A. PROMPT: Algorithm and tool for automated ontology merging and alignment. Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, August 2000, pp. 450-455.
17. McGuinness, D. L., Fikes, R., Rice, J. and Wilder, S. An environment for merging and testing large ontologies. Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR 2000), Breckenridge, CO, April 12-15, 2000, pp. 483-493.
18. Stumme, G. and Maedche, A. FCA-MERGE: Bottom-up merging of ontologies. Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Seattle, WA, August 4-10, 2001, pp. 225-230.
19. McComb, D. Semantics in business systems: The savvy managers guide. Amsterdam, Netherlands: Morgan Kaufmann, 2004.
20. สุพัตรา สวัสดิ์รักเกียรติ. วิธีการรวมแบบจำลองข้อมูลเชิงวัตถุโดยใช้วิทยาการศึกษาสำนึกและการวิเคราะห์ความสัมพันธ์ระหว่างคลาส. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2542.

21. DSTC. Ontology Definition MetaModel: DSTC initial submission. Submitted to the ADTF's RFP for an Ontology Definition Metamodel (ad/03-03-40), Object Management Group, Inc. (OMG), ad/2003-08-01, August 18, 2003.

22. ศิรินทร เอนกพีระศักดิ์. การออกแบบและพัฒนาระบบพจนานุกรมข้อมูลสำหรับระบบสารสนเทศเพื่อการบริหารมหาวิทยาลัย. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2534.

23. อุรฉัตร จินตนาวงศ์. การวางแนวทางเพื่อพัฒนาระบบสารสนเทศเพื่อการจัดการทะเบียนนักศึกษาและรายวิชา. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2534.

24. จินดารัตน์ วิเศษเรืองโรจน์. การพัฒนาโมเดลของข้อมูลสำหรับงานบริหารฝ่ายบุคลากรของมหาวิทยาลัย. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2535.

25. พิบูลย์ศรี พิณรัตน์. การวิเคราะห์ระบบการเงินในเชิงข้อมูลเพื่อเสริมระบบสารสนเทศสำหรับบริหารสถาบันการศึกษา. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2535.

26. Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies (IJHCS) 43, 5-6, (November 1995): 907-928.

27. Noy, N. F. and McGuinness, D. L. Ontology development 101: A guide to creating your first ontology [PDF file]. Stanford Knowledge Systems Laboratory Technical Report and Stanford Medical Informatics Technical Report, Stanford University, Stanford, CA, Report No. KSL-01-05 and SMI-2001-0880, March, 2001. Available from: http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf [February 20, 2006]

28. Resource Description Framework (RDF): Concepts and abstract syntax [Online]. In Klyne, G. and Carroll, J. J. (eds.), W3C Recommendation 10 February 2004, Available from: http://www.w3.org/TR/rdf-concepts/ [February 20, 2006]

29. OWL Web Ontology Language guide [Online]. In Smith, M. K., Welty, C. and McGuinness, D. L. (eds.), W3C Recommendation 10 February 2004, Available from: http://www.w3.org/TR/2004/REC-owl-guide-20040210/ [February 20, 2006]

30. Bechhofer, S., Harmelen, F. V., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F. and Stein, L. A. OWL Web Ontology Language reference [Online]. In

Dean, M. and Schreiber, G. (eds.), W3C Recommendation 10 February 2004, Available from: http://www.w3.org/TR/owl-ref/ [February 20, 2006]

31. Supekar, K. and Dameron, O. Advanced reasoning with OWL [Presentation]. The 2005 Protégé Conference, Available from: http://protege.stanford.edu/conference/2005/slides/T5_supekar-dameron_AdvancedOWLReasoningTutorial.pdf [February 20, 2006]
32. The Protégé ontology editor and knowledge acquisition system [Online]. Available from: http://protege.stanford.edu/ [February 20, 2006]
33. Horridge, M., Knublauch, H., Rector, A., Stevens, R. and Wroe, C. A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools [PDF file]. The University Of Manchester, Available from: http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf [February 20, 2006]
34. Knublauch, H., Musen, M. A. and Rector, A. L. Editing description logic ontologies with the Protégé OWL plugin. Proceedings of the 2004 International Workshop on Description Logics (DL2004), British Columbia, Canada, 2004.
35. Jena – A semantic web framework for Java [Online]. Available from: http://jena.sourceforge.net/ [April 20, 2006]
36. Jena2 database interface - Release notes [Online]. Hewlett-Packard Development Company, 2006. Available from: http://jena.sourceforge.net/DB/index.html [April 20, 2006]
37. Reynolds, D. Jena 2 inference support [Online]. Hewlett-Packard Development Company, LP, 2006. Available from: http://jena.sourceforge.net/inference/index.html [April 20, 2006]
38. Schoening, J. IEEE P1600.1 Standard Upper Ontology Working Group (SUO WG) home page [Online]. IEEE, 2003. Available from: http://suo.ieee.org/ [February 20, 2006]
39. Calì, A., Calvanese, D., De Giacomo, G. and Lenzerini, M. Data integration under integrity constraints. Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE'02), Toronto, Ontario, Canada, May 27-31, 2002, pp. 262-279.
40. MySQL AB :: The world's most popular open source database [Online]. Available from: http://www.mysql.com/ [April 20, 2006]
41. Wang, H., Horridge, M., Rector, A., Drummond, N. and Seidenberg, J. Debugging OWL-DL ontologies: A heuristic approach. Proceedings of the 4th International Semantic

Web Conference (ICWC 2005), Galway, Ireland, November 6-10, 2005, pp. 745-757.

42. Racer systems GmbH & Co. KG [Online]. Available from: http://www.racer-systems.com/index.phtml [April 20, 2006]
43. Jamadhvaja, M. and Senivongse, T. An integration of data sources with UML class models based on ontological analysis. Proceedings of the 1st International ACM Workshop on Interoperability of Heterogeneous Information Systems (IHIS'05), Bremen, Germany, November 4, 2005, pp. 1-8.
44. Oracle database 10g [Online]. Available from: http://www.oracle.com/technology/products /database/oracle10g/index.html [April 20, 2006]
45. PHP: Hypertext preprocessor [Online]. Available from: http://www.php.net/ [April 20, 2006]
46. Blaha, M. and Premerlani, W. Object-oriented modeling and design for database applications. Upper Saddle River, NJ: Prentice Hall, 1998.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

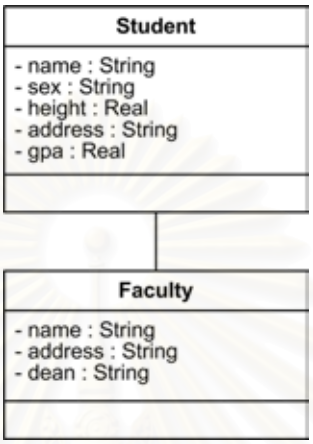
ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก
ตัวอย่างการรวมแบบจำลองข้อมูลคู่ที่ 1

ก.1 แบบจำลองข้อมูลเชิงวัตถุที่นำมารวม

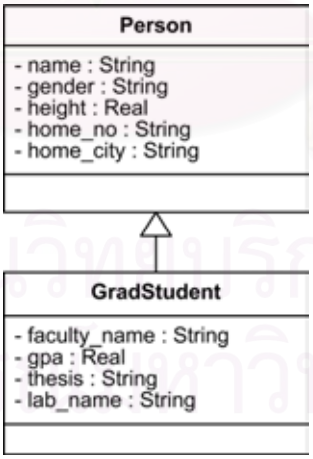
1. แบบจำลองข้อมูลเชิงวัตถุชุดที่ 1



นำมาจากแบบจำลองข้อมูลคู่ที่ 6 ในงานวิจัย [20]

Class semantics
Student: <Faculty = {'science', 'engineer'}
Major = {'computer', 'math'}>
Attribute semantics
Height: <Unit = {'centimetre'}>
Class Integrity constraint
Student: <Sex = {'F', 'M'}>

2. แบบจำลองข้อมูลเชิงวัตถุชุดที่ 2



นำมาจากแบบจำลองข้อมูลคู่ที่ 6 ในงานวิจัย [20]

Class semantics
GradStudent: <Faculty = {'science', 'account', 'engineer'}
Major = {'computer', 'stat', 'math', 'electronics'}>
Attribute semantics
Height: <Unit = {'metre'}>
Class Integrity constraint
Person: <Gender = {'female', 'male'}>

3. คำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1 และ 2

Synonym-list:

Data Model1.Student.sex = Data Model2.Person.gender
Data Model1.Student.faculty.name = Data Model2.GradStudent.faculty_name

Hypernym-list:

Data Model1.Student.address = Data Model2.Person.home_no + Data Model2.Person.home_city

ก.2 โลคัลออนโทโลจี

1. โลคัลออนโทโลจีที่ 1 สร้างจากแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns="http://www.local-ontology.com/local6A.owl#"
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.local-ontology.com/local6A.owl">
  <owl:Ontology rdf:about="">
   <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="StudentSex">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Student.sex
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="math">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
   <rdfs:subClassOf>
     <owl:Class rdf:ID="StudentMajorSemantic"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="M">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
   <rdfs:subClassOf>
     <owl:Class rdf:ID="StudentSexConstraint"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="StudentFacultySemantic">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
  </owl:Class>
  <owl:Class rdf:ID="science">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
   <rdfs:subClassOf rdf:resource="#StudentFacultySemantic"/>
  </owl:Class>
  <owl:Class rdf:ID="StudentFaculty">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Student.faculty
   &lt;/p></rdfs:comment>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  </owl:Class>
  <owl:Class rdf:ID="Faculty">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Faculty
   &lt;/p></rdfs:comment>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
  </owl:Class>
  <owl:Class rdf:ID="StudentHeight">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Student.height
```

```
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="StudentGpa">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Student.gpa
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="FacultyName">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Faculty.name
  &lt;/p>
  &lt;p style="margin-top: 0">
    Data Model1.Student.Faculty.name
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="StudentName">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Student.name
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="FacultyDean">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Faculty.dean
  &lt;/p>
  &lt;p style="margin-top: 0">
    Data Model1.Student.Faculty.dean
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="F">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#StudentSexConstraint"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="centimetre">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="StudentHeightUnitSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="StudentAddress">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Student.address
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="engineer">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#StudentFacultySemantic"/>
</owl:Class>
<owl:Class rdf:about="#StudentMajorSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="Student">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Student
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
</owl:Class>
<owl:Class rdf:ID="FacultyAddress">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Faculty.address
  &lt;/p>
  &lt;p style="margin-top: 0">
```

```
   Data Model1.Student.Faculty.address
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#StudentSexConstraint">
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#M"/>
        <owl:Class rdf:about="#F"/>
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassIntegrityConstraint"/>
</owl:Class>
<owl:Class rdf:ID="computer">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#StudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:about="#StudentHeightUnitSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasStudentGpa">
  <rdfs:range rdf:resource="#StudentGpa"/>
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentSemantic">
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#StudentFacultySemantic"/>
        <owl:Class rdf:about="#StudentMajorSemantic"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:domain rdf:resource="#Student"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentFacultyAltType">
  <rdfs:domain rdf:resource="#StudentFaculty"/>
  <rdfs:range rdf:resource="#Faculty"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeAltType"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentName">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#StudentName"/>
  <rdfs:domain rdf:resource="#Student"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyDean">
  <rdfs:domain rdf:resource="#Faculty"/>
  <rdfs:range rdf:resource="#FacultyDean"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyAddress">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Faculty"/>
  <rdfs:range rdf:resource="#FacultyAddress"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentHeightSemantic">
  <rdfs:domain rdf:resource="#StudentHeight"/>
  <rdfs:range rdf:resource="#StudentHeightUnitSemantic"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentFaculty">
  <rdfs:range rdf:resource="#StudentFaculty"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Student"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentAddress">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#StudentAddress"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentSex">
```

```
  <rdfs:range rdf:resource="#StudentSex"/>
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasStudentIntegrityConstraint">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassIntegrityConstraint"/>
  <rdfs:range rdf:resource="#StudentSexConstraint"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasStudentHeight">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#StudentHeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasFacultyName">
  <rdfs:range rdf:resource="#FacultyName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Faculty"/>
 </owl:ObjectProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentAddressType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#StudentAddress"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentSexType">
  <rdfs:domain rdf:resource="#StudentSex"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasFacultyNameType">
  <rdfs:domain rdf:resource="#FacultyName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasFacultyAddressType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#FacultyAddress"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentGpaType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#StudentGpa"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasFacultyDeanType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#FacultyDean"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentNameType">
  <rdfs:domain rdf:resource="#StudentName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasStudentHeightType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#StudentHeight"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
 </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

2. โลคัลออนโทโลจีที่ 2 สร้างจากแบบจำลองข้อมูลเชิงวัตถุชุดที่ 2

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns="http://www.local-ontology.com/local6B.owl#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.local-ontology.com/local6B.owl">
```

```
<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
</owl:Ontology>
<owl:Class rdf:ID="PersonGender">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Person.gender
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="male">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="PersonGenderConstraint"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="PersonName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Person.name
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="GradStudentMajorSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="stat">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#GradStudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:ID="female">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#PersonGenderConstraint"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="GradStudentLabName">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.GradStudent.lab_name
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="account">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="GradStudentFacultySemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#PersonGenderConstraint">
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#male"/>
        <owl:Class rdf:about="#female"/>
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassIntegrityConstraint"/>
</owl:Class>
<owl:Class rdf:ID="GradStudent">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Person"/>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.GradStudent
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PersonHomeCity">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Person.home_city
  &lt;/p></rdfs:comment>
```

```
</owl:Class>
<owl:Class rdf:ID="science">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#GradStudentFacultySemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="computer">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#GradStudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:about="#Person">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Person
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
</owl:Class>
<owl:Class rdf:about="#GradStudentFacultySemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="PersonHeight">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Person.height
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="metre">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="PersonHeightUnitSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="GradStudentFacultyName">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.GradStudent.faculty_name
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="PersonHomeNo">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Person.home_no
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="GradStudentGpa">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.GradStudent.gpa
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#PersonHeightUnitSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="math">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#GradStudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:ID="electronics">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#GradStudentMajorSemantic"/>
</owl:Class>
<owl:Class rdf:ID="engineer">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#GradStudentFacultySemantic"/>
</owl:Class>
<owl:Class rdf:ID="GradStudentThesis">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.GradStudent.thesis
  &lt;/p></rdfs:comment>
```

```
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasPersonName">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#PersonName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentSemantic">
  <rdfs:domain rdf:resource="#GradStudent"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#GradStudentMajorSemantic"/>
        <owl:Class rdf:about="#GradStudentFacultySemantic"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHeightSemantic">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeSemantic"/>
  <rdfs:range rdf:resource="#PersonHeightUnitSemantic"/>
  <rdfs:domain rdf:resource="#PersonHeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentThesis">
  <rdfs:domain rdf:resource="#GradStudent"/>
  <rdfs:range rdf:resource="#GradStudentThesis"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHomeCity">
  <rdfs:range rdf:resource="#PersonHomeCity"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonIntegrityConstraint">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassIntegrityConstraint"/>
  <rdfs:range rdf:resource="#PersonGenderConstraint"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonGender">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#PersonGender"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHomeNo">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#PersonHomeNo"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentGpa">
  <rdfs:domain rdf:resource="#GradStudent"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#GradStudentGpa"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHeight">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#PersonHeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentFacultyName">
  <rdfs:domain rdf:resource="#GradStudent"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#GradStudentFacultyName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentLabName">
  <rdfs:range rdf:resource="#GradStudentLabName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#GradStudent"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasGradStudentLabNameType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#GradStudentLabName"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonGenderType">
```

```
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonGender"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasGradStudentThesisType">
  <rdfs:domain rdf:resource="#GradStudentThesis"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonHomeCityType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonHomeCity"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasGradStudentFacultyNameType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#GradStudentFacultyName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonHeightType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#PersonHeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonHomeNoType">
  <rdfs:domain rdf:resource="#PersonHomeNo"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonNameType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasGradStudentGpaType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#GradStudentGpa"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

3. ออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีที่ 1 และโลคัลออนโทโลจีที่ 2

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:local1="http://www.local-ontology.com/local6A.owl#"
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:local2="http://www.local-ontology.com/local6B.owl#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns="http://www.semantic-ontology.com/semantic6.owl#"
  xml:base="http://www.semantic-ontology.com/semantic6.owl">
 <owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.local-ontology.com/local6B.owl"/>
  <owl:imports rdf:resource="http://www.local-ontology.com/local6A.owl"/>
  <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
 </owl:Ontology>
 <rdf:Description rdf:about="http://www.local-ontology.com/local6A.owl#StudentSex">
  <owl:equivalentClass>
   <rdf:Description rdf:about="http://www.local-ontology.com/local6B.owl#PersonGender">
    <owl:equivalentClass rdf:resource="http://www.local-ontology.com/local6A.owl#StudentSex"/>
   </rdf:Description>
  </owl:equivalentClass>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.local-ontology.com/local6A.owl#StudentAddress">
  <owl:equivalentClass>
   <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
     <rdf:Description rdf:about="http://www.local-ontology.com/local6B.owl#PersonHomeCity"/>
     <rdf:Description rdf:about="http://www.local-ontology.com/local6B.owl#PersonHomeNo"/>
```

```
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.local-ontology.com/local6B.owl#GradStudentFacultyName">
  <owl:equivalentClass>
   <rdf:Description rdf:about="http://www.local-ontology.com/local6A.owl#FacultyName">
    <owl:equivalentClass rdf:resource="http://www.local-
ontology.com/local6B.owl#GradStudentFacultyName"/>
   </rdf:Description>
  </owl:equivalentClass>
 </rdf:Description>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

ก.3 ออนโทโลจีรวม

1. ออนโทโลจีรวมพื้นฐาน (IntegratedOntology.owl)

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns="http://www.integrate-ontology.com/IntegratedOntology6.owl#"
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.integrate-ontology.com/IntegratedOntology6.owl">
 <owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
 </owl:Ontology>
 <owl:Class rdf:ID="Faculty">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#Faculty
  &lt;/p></rdfs:comment>
 </owl:Class>
 <owl:Class rdf:ID="computer">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="GradStudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Class rdf:ID="StudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 </owl:Class>
 <owl:Class rdf:ID="FacultyDean">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#FacultyDean
  &lt;/p></rdfs:comment>
 </owl:Class>
 <owl:Class rdf:ID="science">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="GradStudentFacultySemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Class rdf:ID="StudentFacultySemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
 </owl:Class>
 <owl:Class rdf:ID="GradStudentLabName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6B.owl#GradStudentLabName
  &lt;/p></rdfs:comment>
 </owl:Class>
 <owl:Class rdf:ID="engineer">
```

```
  <rdfs:subClassOf>
    <owl:Class rdf:about="#GradStudentFacultySemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#StudentFacultySemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="GradStudent">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Student"/>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6B.owl#GradStudent
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="StudentFaculty">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#StudentFaculty
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="metre">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="PersonHeightUnitSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="PersonGender">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#StudentSex
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local6B.owl#PersonGender
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="electronics">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#GradStudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#StudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="FacultyAddress">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#FacultyAddress
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="GradStudentThesis">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6B.owl#GradStudentThesis
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="math">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#GradStudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#StudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="PersonHeight">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
  >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#StudentHeight
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local6B.owl#PersonHeight
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="F">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="PersonGenderConstraint"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="M">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ConstraintValue"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#PersonGenderConstraint"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#StudentFacultySemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="stat">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#GradStudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#StudentMajorSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:about="#Student">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#Student
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Person"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#GradStudentFacultySemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="account">
  <rdfs:subClassOf rdf:resource="#GradStudentFacultySemantic"/>
  <rdfs:subClassOf rdf:resource="#StudentFacultySemantic"/>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:about="#Person">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6B.owl#Person
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PersonName">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#StudentName
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local6B.owl#PersonName
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#PersonHeightUnitSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="StudentGpa">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local6A.owl#StudentGpa
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local6B.owl#GradStudentGpa
```

```
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#StudentMajorSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:about="#GradStudentMajorSemantic">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="FacultyName">
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#FacultyName
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local6B.owl#GradStudentFacultyName
 &lt;/p></rdfs:comment>
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#PersonGenderConstraint">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassIntegrityConstraint"/>
</owl:Class>
<owl:Class rdf:ID="PersonAddress">
 <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
  equivalent to local6A.owl#StudentAddress
 &lt;/p>
 &lt;p style="margin-top: 0">
  equivalent to local6B.owl#PersonHomeNo + local6B.owl#PersonHomeCity
 &lt;/p></rdfs:comment>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasPersonHeight">
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#PersonHeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentThesis">
 <rdfs:domain rdf:resource="#GradStudent"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#GradStudentThesis"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonAddress">
 <rdfs:range rdf:resource="#PersonAddress"/>
 <rdfs:domain rdf:resource="#Person"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentSemantic">
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
 <rdfs:domain rdf:resource="#GradStudent"/>
 <rdfs:range>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#GradStudentFacultySemantic"/>
    <owl:Class rdf:about="#StudentMajorSemantic"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyDean">
 <rdfs:domain rdf:resource="#Faculty"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#FacultyDean"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentFaculty">
 <rdfs:domain rdf:resource="#Student"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 <rdfs:range rdf:resource="#StudentFaculty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentFacultyAltType">
 <rdfs:domain rdf:resource="#StudentFaculty"/>
 <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeAltType"/>
 <rdfs:range rdf:resource="#Faculty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonIntegrityConstraint">
```

```
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassIntegrityConstraint"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#PersonGenderConstraint"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyAddress">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Faculty"/>
  <rdfs:range rdf:resource="#FacultyAddress"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacultyName">
  <rdfs:range rdf:resource="#FacultyName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Faculty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonGender">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#PersonGender"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonHeightSemantic">
  <rdfs:range rdf:resource="#PersonHeightUnitSemantic"/>
  <rdfs:domain rdf:resource="#PersonHeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentSemantic">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#StudentFacultySemantic"/>
        <owl:Class rdf:about="#StudentMajorSemantic"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStudentGpa">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#StudentGpa"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#PersonName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasGradStudentLabName">
  <rdfs:domain rdf:resource="#GradStudent"/>
  <rdfs:range rdf:resource="#GradStudentLabName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasPersonAddressType">
  <rdfs:domain rdf:resource="#PersonAddress"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasStudentGpaType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#StudentGpa"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasFacultyDeanType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#FacultyDean"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonHeightType">
  <rdfs:domain rdf:resource="#PersonHeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasPersonGenderType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonGender"/>
```

```
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasGradStudentThesisType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#GradStudentThesis"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasFacultyAddressType">
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:domain rdf:resource="#FacultyAddress"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasGradStudentLabNameType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:domain rdf:resource="#GradStudentLabName"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasPersonNameType">
    <rdfs:domain rdf:resource="#PersonName"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasFacultyNameType">
    <rdfs:domain rdf:resource="#FacultyName"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

2. ผลการรวมออนโทโลจี (IntegratedOntology.txt)

```
[Class] : Person (equivalent to local6B.owl#Person)
<subClassOf> upper.owl#Class
<hasClassAttribute>
      [ClassAttribute] : PersonName (equivalent to local6A.owl#StudentName, local6B.owl#PersonName)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> String
      Type conversion : no
      [ClassAttribute] : PersonGender (equivalent to local6A.owl#StudentSex, local6B.owl#PersonGender)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> String
      Type conversion : no
      [ClassAttribute] : PersonHeight (equivalent to local6A.owl#StudentHeight, local6B.owl#PersonHeight)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> String
      Type conversion : no
      <hasClassAttributeSemantic>
            [SemanticName] : PersonHeightUnitSemantic
            <subClassOf> upper.owl#SemanticName
                  [SemanticWord] : metre
                  <subClassOf> upper.owl#SemanticWord, PersonHeightUnitSemantic
      [ClassAttribute] : PersonAddress (equivalent to local6A.owl#StudentAddress, local6B.owl#PersonHomeNo
+ local6B.owl#PersonHomeCity)
      <hasClassAttributeType> String
      Type conversion : no
<hasClassIntegrityConstraint>
      [ClassIntegrityConstraint] : PersonGenderConstraint
      <subClassOf> upper.owl#ClassIntegrityConstraint
            [ConstraintValue] : F
            <subClassOf> upper.owl#ConstraintValue, PersonGenderConstraint
            Mapping function : yes
            [ConstraintValue] : M
            <subClassOf> upper.owl#ConstraintValue, PersonGenderConstraint
            Mapping function : yes
<hasClassSemantic> none

[Class] : Student (equivalent to local6A.owl#Student)
<subClassOf> upper.owl#Class, Person
<hasClassAttribute>
      [ClassAttribute] : StudentGpa (equivalent to local6A.owl#StudentGpa, local6B.owl#GradStudentGpa)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> Float
      Type conversion : no
      [ClassAttribute] : StudentFaculty (equivalent to local6A.owl#StudentFaculty)
```

<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeAltType> Faculty
Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
[SemanticName] : StudentFacultySemantic
<subClassOf> upper.owl#SemanticName
[SemanticWord] : science
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticWord] : account
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticWord] : engineer
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticName] : StudentMajorSemantic
<subClassOf> upper.owl#SemanticName
[SemanticWord] : computer
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : stat
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : electronics
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : math
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic

[Class] : GradStudent (equivalent to local6B.owl#GradStudent)
<subClassOf> upper.owl#Class, Student
<hasClassAttribute>
[ClassAttribute] : GradStudentThesis (equivalent to local6B.owl#GradStudentThesis)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : no
[ClassAttribute] : GradStudentLabName (equivalent to local6B.owl#GradStudentLabName)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
[SemanticName] : GradStudentFacultySemantic
<subClassOf> upper.owl#SemanticName
[SemanticWord] : science
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticWord] : account
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticWord] : engineer
<subClassOf> upper.owl#SemanticWord, StudentFacultySemantic, GradStudentFacultySemantic
[SemanticName] : GradStudentMajorSemantic
<subClassOf> upper.owl#SemanticName
[SemanticWord] : computer
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : stat
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : electronics
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic
[SemanticWord] : math
<subClassOf> upper.owl#SemanticWord, StudentMajorSemantic, GradStudentMajorSemantic

[Class] : Faculty (equivalent to local6A.owl#Faculty)
<subClassOf> upper.owl#Class
<hasClassAttribute>
[ClassAttribute] : FacultyName (equivalent to local6A.owl#FacultyName,
local6B.owl#GradStudentFacultyName)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : yes
[ClassAttribute] : FacultyAddress (equivalent to local6A.owl#FacultyAddress)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : no
[ClassAttribute] : FacultyDean (equivalent to local6A.owl#FacultyDean)
<subClassOf> upper.owl#ClassAttribute
<hasClassAttributeType> String
Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic> none

ก.4 ตัวอย่างวิวที่กำกับลงบนฐานข้อมูล

1. วิวที่กำกับลงบนแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1

```
CREATE FUNCTION Sex_to_Gender (old_sex in VARCHAR2)
      RETURN VARCHAR2
      IS new_sex            VARCHAR2(50);
BEGIN
      IF (old_sex='F') THEN
            new_sex:='female';
      ELSIF (old_sex='M') THEN
            new_sex:='male';
      END IF;
      RETURN new_sex;
END;

CREATE FUNCTION Height_to_Metre (old_height in NUMBER)
      RETURN NUMBER
      IS new_height  NUMBER;
BEGIN
      new_height:= old_height /100;
      RETURN new_height;
END;

CREATE OR REPLACE VIEW Person(name, gender, height, address)
      AS
      SELECT a.name, SEX_TO_GENDER(a.sex), Height_to_Metre(a.height), a.address
      FROM StudentTab a;

CREATE OR REPLACE VIEW Student(name, gender, height, address, gpa, faculty_name, faculty_address,
faculty_dean)
      AS
SELECT a.name, SEX_TO_GENDER(a.sex), Height_to_Metre(a.height), a.address, a.gpa, a.faculty.name,
a.faculty.address, a.faculty.dean
      FROM StudentTab a;

CREATE OR REPLACE VIEW Faculty(name, address, dean)
      AS
      SELECT a.name, a.address, a.dean
      FROM FacultyTab a;
```

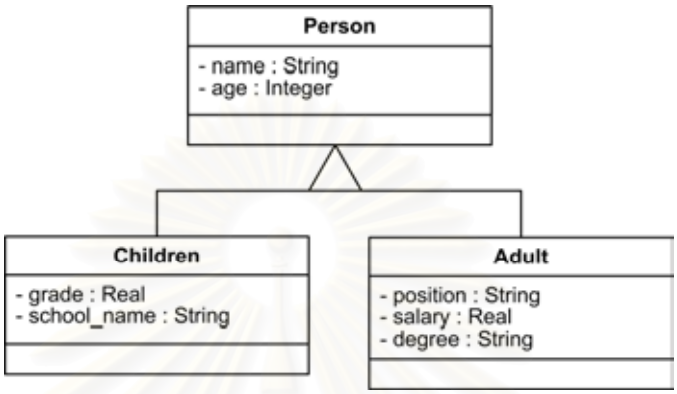
2. วิวที่กำกับลงบนแบบจำลองข้อมูลเชิงวัตถุชุดที่ 2

```
CREATE OR REPLACE VIEW Person(name, gender, height, address)
      AS
      SELECT a.name, a.gender, a.height, a.home_no || ', ' || a.home_city
      FROM PersonTab a
      UNION
SELECT b.person.name, b.person.gender, b.person.height, b.person.home_no || ', ' || b.person.home_city
      FROM GradStudentTab b;

CREATE OR REPLACE VIEW Student(name, gender, height, address, gpa, faculty_name, faculty_address,
faculty_dean)
      AS
SELECT a.person.name, a.person.gender, a.person.height, a.person.home_no || ', ' || a.person.home_city,
a.gpa, a.faculty_name, '—', '—'
      FROM GradStudentTab a;

CREATE OR REPLACE VIEW GradStudent(name, gender, height, address, gpa, faculty_name,
faculty_address, faculty_dean, thesis, lab_name)
      AS
SELECT a.person.name, a.person.gender, a.person.height, a.person.home_no || ', ' || a.person.home_city,
a.gpa, a.faculty_name, '—', '—', a.thesis, a.lab_name
      FROM GradStudentTab a;

CREATE OR REPLACE VIEW Faculty(name, address, dean)
      AS
      SELECT a.faculty_name, '—', '—'
      FROM GradStudentTab a;
```

ภาคผนวก ข
ตัวอย่างการรวมแบบจำลองข้อมูลคู่ที่ 2

ข.1 แบบจำลองข้อมูลเชิงวัตถุที่นำมารวม

1. แบบจำลองข้อมูลเชิงวัตถุชุดที่ 1

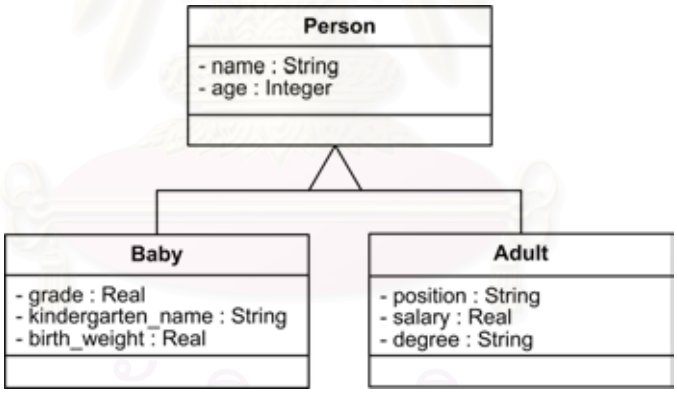


นำมาจากแบบจำลองข้อมูลคู่ที่ 4 ในงานวิจัย [20]

Class semantics

Children: <Age = [0, 20]>
Adult: <Age = [30, 150]>

2. แบบจำลองข้อมูลเชิงวัตถุชุดที่ 2



นำมาจากแบบจำลองข้อมูลคู่ที่ 4 ในงานวิจัย [20]

Class semantics

Baby: <Age = [0, 2]>
Adult: <Age = [20, 120]>

3. คำที่มีความหมายเหมือนกันหรือคำที่เป็นคำลูกกลุ่ม/ แม่กลุ่มกันระหว่างแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1 และ 2

Synonym-list:

Data Model1.Childten.school_name = Data Model2.Baby.kindergarten_name

ข.2 โลคัลออนโทโลจี

1. โลคัลออนโทโลจีที่ 1 สร้างจากแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns="http://www.local-ontology.com/local4A.owl#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.local-ontology.com/local4A.owl">
  <owl:Ontology rdf:about="">
   <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Adult">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Adult
   &lt;/p></rdfs:comment>
   <rdfs:subClassOf>
     <owl:Class rdf:ID="Person"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="AdultSalary">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Adult.salary
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="GreaterThanOrEqual30">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
   <rdfs:subClassOf>
     <owl:Class rdf:ID="AdultAgeSemantic"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Person">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Person
   &lt;/p></rdfs:comment>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
  </owl:Class>
  <owl:Class rdf:ID="AdultDegree">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Adult.degree
   &lt;/p></rdfs:comment>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  </owl:Class>
  <owl:Class rdf:ID="LessThanOrEqual20">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
   <rdfs:subClassOf>
     <owl:Class rdf:ID="ChildrenAgeSemantic"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#AdultAgeSemantic">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
  </owl:Class>
  <owl:Class rdf:ID="PersonName">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Person.name
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="AdultPosition">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     Data Model1.Adult.position
```

```
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#ChildrenAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="PersonAge">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Person.age
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Children">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Children
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:Class rdf:ID="ChildrenSchoolName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Children.school_name
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="ChildrenGrade">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model1.Children.grade
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual0">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
</owl:Class>
<owl:Class rdf:ID="LessThanOrEqual150">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#AdultAgeSemantic"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasChildrenGrade">
  <rdfs:range rdf:resource="#ChildrenGrade"/>
  <rdfs:domain rdf:resource="#Children"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultPosition">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:range rdf:resource="#AdultPosition"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSemantic">
  <rdfs:domain rdf:resource="#Children"/>
  <rdfs:range rdf:resource="#ChildrenAgeSemantic"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSemantic">
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:range rdf:resource="#AdultAgeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSchoolName">
  <rdfs:range rdf:resource="#ChildrenSchoolName"/>
  <rdfs:domain rdf:resource="#Children"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSalary">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:range rdf:resource="#AdultSalary"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#PersonName"/>
```

```
    <rdfs:domain rdf:resource="#Person"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasAdultDegree">
    <rdfs:domain rdf:resource="#Adult"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
    <rdfs:range rdf:resource="#AdultDegree"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasPersonAge">
    <rdfs:range rdf:resource="#PersonAge"/>
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="hasPersonAgeType">
    <rdfs:domain rdf:resource="#PersonAge"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasAdultSalaryType">
    <rdfs:domain rdf:resource="#AdultSalary"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasAdultDegreeType">
    <rdfs:domain rdf:resource="#AdultDegree"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasAdultPositionType">
    <rdfs:domain rdf:resource="#AdultPosition"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasChildrenGradeType">
    <rdfs:domain rdf:resource="#ChildrenGrade"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasChildrenSchoolNameType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:domain rdf:resource="#ChildrenSchoolName"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasPersonNameType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:domain rdf:resource="#PersonName"/>
  </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

2. โคดัลออนโทโลจีที่ 2 สร้างจากแบบจำลองข้อมูลเชิงวัตถุชุดที่ 2

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:upper="http://www.upper-ontology.com/upper.owl#"
    xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.local-ontology.com/local4B.owl#"
  xml:base="http://www.local-ontology.com/local4B.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Adult">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >&lt;p style="margin-top: 0">
      Data Model2.Adult
    &lt;/p></rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Person"/>
    </rdfs:subClassOf>
  </owl:Class>
```

```
<owl:Class rdf:ID="LessThanOrEqual20">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="AdultAgeSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="PersonAge">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Person.age
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#AdultAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual20">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#AdultAgeSemantic"/>
</owl:Class>
<owl:Class rdf:ID="AdultPosition">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Adult.position
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="BabyKindergartenName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Baby.kindergarten_name
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="BabyAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="BabyBirthWeight">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Baby.birth_weight
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="LessThanOrEqual2">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#BabyAgeSemantic"/>
</owl:Class>
<owl:Class rdf:about="#Person">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Person
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual0">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#BabyAgeSemantic"/>
</owl:Class>
<owl:Class rdf:ID="AdultSalary">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Adult.salary
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="BabyGrade">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Baby.grade
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="AdultDegree">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```
  >&lt;p style="margin-top: 0">
    Data Model2.Adult.degree
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="PersonName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Person.name
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Baby">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    Data Model2.Baby
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasBabyKindergartenName">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Baby"/>
  <rdfs:range rdf:resource="#BabyKindergartenName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultDegree">
  <rdfs:range rdf:resource="#AdultDegree"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#PersonName"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabySemantic">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:range rdf:resource="#BabyAgeSemantic"/>
  <rdfs:domain rdf:resource="#Baby"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultPosition">
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:range rdf:resource="#AdultPosition"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabyGrade">
  <rdfs:domain rdf:resource="#Baby"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#BabyGrade"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonAge">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#PersonAge"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabyBirthWeight">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Baby"/>
  <rdfs:range rdf:resource="#BabyBirthWeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSemantic">
  <rdfs:range rdf:resource="#AdultAgeSemantic"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSalary">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#AdultSalary"/>
  <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasBabyBirthWeightType">
  <rdfs:domain rdf:resource="#BabyBirthWeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasBabyKindergartenNameType">
```

```
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#BabyKindergartenName"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasBabyGradeType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:domain rdf:resource="#BabyGrade"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasPersonAgeType">
    <rdfs:domain rdf:resource="#PersonAge"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasAdultDegreeType">
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:domain rdf:resource="#AdultDegree"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasAdultSalaryType">
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:domain rdf:resource="#AdultSalary"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasAdultPositionType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#AdultPosition"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasPersonNameType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:domain rdf:resource="#PersonName"/>
  </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

3. ออนโทโลจีที่ระบุความสัมพันธ์ระหว่างโลคัลออนโทโลจีที่ 1 และโลคัลออนโทโลจีที่ 2

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:upper="http://www.upper-ontology.com/upper.owl#"
    xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:local1="http://www.local-ontology.com/local4A.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:local2="http://www.local-ontology.com/local4B.owl#"
    xmlns="http://www.semantic-ontology.com/semantic4.owl#"
  xml:base="http://www.semantic-ontology.com/semantic4.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.local-ontology.com/local4B.owl"/>
    <owl:imports rdf:resource="http://www.local-ontology.com/local4A.owl"/>
    <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
  </owl:Ontology>
  <rdf:Description rdf:about="http://www.local-ontology.com/local4A.owl#ChildrenSchoolName">
    <owl:equivalentClass>
      <rdf:Description rdf:about="http://www.local-ontology.com/local4B.owl#BabyKindergartenName">
        <owl:equivalentClass rdf:resource="http://www.local-ontology.com/local4A.owl#ChildrenSchoolName"/>
      </rdf:Description>
    </owl:equivalentClass>
  </rdf:Description>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

ข.3 ออนโทโลจีรวม

1. ออนโทโลจีรวมพื้นฐาน (IntegratedOntology.owl)

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns="http://www.integrate-ontology.com/IntegratedOntology4.owl#"
  xml:base="http://www.integrate-ontology.com/IntegratedOntology4.owl">
  <owl:Ontology rdf:about="">
   <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="ChildrenAgeSemantic">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
  </owl:Class>
  <owl:Class rdf:ID="Children">
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     equivalent to local4A.owl#Children
   &lt;/p></rdfs:comment>
   <rdfs:subClassOf>
     <owl:Class rdf:ID="Person"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="LessThanOrEqual20">
   <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  </owl:Class>
  <owl:Class rdf:ID="PersonName">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     equivalent to local4A.owl#PersonName
   &lt;/p>
   &lt;p style="margin-top: 0">
     equivalent to local4B.owl#PersonName
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="ChildrenGrade">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     equivalent to local4A.owl#ChildrenGrade
   &lt;/p>
   &lt;p style="margin-top: 0">
     equivalent to local4B.owl#BabyGrade
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="AdultSalary">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     equivalent to local4A.owl#AdultSalary
   &lt;/p>
   &lt;p style="margin-top: 0">
     equivalent to local4B.owl#AdultSalary
   &lt;/p></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="AdultPosition">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
     equivalent to local4A.owl#AdultPosition
   &lt;/p>
   &lt;p style="margin-top: 0">
     equivalent to local4B.owl#AdultPosition
   &lt;/p></rdfs:comment>
  </owl:Class>
```

```
<owl:Class rdf:ID="GreaterThanOrEqual0">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="BabyAgeSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="LessThanOrEqual150">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="AdultAgeSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="BabyBirthWeight">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4B.owl#BabyBirthWeight
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Baby">
  <rdfs:subClassOf rdf:resource="#Children"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4B.owl#Baby
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="ChildrenSchoolName">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#ChildrenSchoolName
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#BabyKindergartenName
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:about="#Person">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#Person
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#Person
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PersonAge">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#PersonAge
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#PersonAge
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="AdultDegree">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#AdultDegree
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#AdultDegree
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="Adult">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#Adult
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#Adult
```

```
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#AdultAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="LessThanOrEqual2">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#BabyAgeSemantic"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual20">
  <rdfs:subClassOf rdf:resource="#AdultAgeSemantic"/>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
</owl:Class>
<owl:Class rdf:about="#BabyAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasChildrenGrade">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Children"/>
  <rdfs:range rdf:resource="#ChildrenGrade"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSemantic">
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:range rdf:resource="#AdultAgeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSemantic">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:range rdf:resource="#ChildrenAgeSemantic"/>
  <rdfs:domain rdf:resource="#Children"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#PersonName"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonAge">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#PersonAge"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabySemantic">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:domain rdf:resource="#Baby"/>
  <rdfs:range rdf:resource="#BabyAgeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultPosition">
  <rdfs:range rdf:resource="#AdultPosition"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultDegree">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#AdultDegree"/>
  <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabyBirthWeight">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Baby"/>
  <rdfs:range rdf:resource="#BabyBirthWeight"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSalary">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:range rdf:resource="#AdultSalary"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSchoolName">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#ChildrenSchoolName"/>
  <rdfs:domain rdf:resource="#Children"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasAdultPositionType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
```

```
    <rdfs:domain rdf:resource="#AdultPosition"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasAdultDegreeType">
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#AdultDegree"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasChildrenGradeType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:domain rdf:resource="#ChildrenGrade"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasPersonNameType">
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#PersonName"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasBabyBirthWeightType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:domain rdf:resource="#BabyBirthWeight"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasPersonAgeType">
    <rdfs:domain rdf:resource="#PersonAge"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasAdultSalaryType">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:domain rdf:resource="#AdultSalary"/>
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasChildrenSchoolNameType">
    <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
    <rdfs:domain rdf:resource="#ChildrenSchoolName"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

2. ผลการรวมออนโทโลจี (IntegratedOntology.txt)

```
[Class] : Person (equivalent to local4A.owl#Person, local4B.owl#Person)
<subClassOf> upper.owl#Class
<hasClassAttribute>
      [ClassAttribute] : PersonName (equivalent to local4A.owl#PersonName, local4B.owl#PersonName)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> String
      Type conversion : no
      [ClassAttribute] : PersonAge (equivalent to local4A.owl#PersonAge, local4B.owl#PersonAge)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> Int
      Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic> none

[Class] : Children (equivalent to local4A.owl#Children)
<subClassOf> upper.owl#Class, Person
<hasClassAttribute>
      [ClassAttribute] : ChildrenGrade (equivalent to local4A.owl#ChildrenGrade, local4B.owl#BabyGrade)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> Float
      Type conversion : no
      [ClassAttribute] : ChildrenSchoolName (equivalent to local4A.owl#ChildrenSchoolName,
local4B.owl#BabyKindergartenName)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeAltType> String
      Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
      [SemanticName] : ChildrenAgeSemantic
      <subClassOf> upper.owl#SemanticName
            [SemanticWord] : GreaterThanOrEqual0
```

```
            <subClassOf> upper.owl#SemanticWord, ChildrenAgeSemantic, BabyAgeSemantic
            [SemanticWord] : LessThanOrEqual20
            <subClassOf> upper.owl#SemanticWord, ChildrenAgeSemantic

[Class] : Baby (equivalent to local4B.owl#Baby)
<subClassOf> upper.owl#Class, Children
<hasClassAttribute>
      [ClassAttribute] : BabyBirthWeight (equivalent to local4B.owl#BabyBirthWeight)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> Float
      Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
      [SemanticName] : BabyAgeSemantic
      <subClassOf> upper.owl#SemanticName
            [SemanticWord] : GreaterThanOrEqual0
            <subClassOf> upper.owl#SemanticWord, ChildrenAgeSemantic, BabyAgeSemantic
            [SemanticWord] : LessThanOrEqual2
            <subClassOf> upper.owl#SemanticWord, BabyAgeSemantic

[Class] : Adult (equivalent to local4A.owl#Adult, local4B.owl#Adult)
<subClassOf> upper.owl#Class, Person
<hasClassAttribute>
      [ClassAttribute] : AdultPosition (equivalent to equivalent to local4A.owl#AdultPosition,
local4B.owl#AdultPosition)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> String
      Type conversion : no
      [ClassAttribute] : AdultSalary (equivalent to local4A.owl#AdultSalary, local4B.owl#AdultSalary)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> Float
      Type conversion : no
      [ClassAttribute] : AdultDegree (equivalent to local4A.owl#AdultDegree, local4B.owl#AdultDegree)
      <subClassOf> upper.owl#ClassAttribute
      <hasClassAttributeType> String
      Type conversion : no
<hasClassIntegrityConstraint> none
<hasClassSemantic>
      [SemanticName] : AdultAgeSemantic
      <subClassOf> upper.owl#SemanticName
            [SemanticWord] : GreaterThanOrEqual20
            <subClassOf> upper.owl#SemanticWord, AdultAgeSemantic
            [SemanticWord] : LessThanOrEqual150
            <subClassOf> upper.owl#SemanticWord, AdultAgeSemantic
```

3. ออนโทโลจีรวมที่ผ่านการอนุมานแล้ว

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:upper="http://www.upper-ontology.com/upper.owl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns="http://www.integrate-ontology.com/IntegratedOntology4.owl#"
  xml:base="http://www.integrate-ontology.com/IntegratedOntology4.owl">
  <owl:Ontology rdf:about="">
   <owl:imports rdf:resource="http://www.upper-ontology.com/upper.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="ChildrenAgeSemantic">
   <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
  </owl:Class>
  <owl:Class rdf:ID="Children">
   <rdfs:subClassOf>
    <owl:Restriction>
     <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasPersonAge"/>
     </owl:onProperty>
     <owl:someValuesFrom>
      <owl:Class rdf:ID="LessThanOrEqual20"/>
     </owl:someValuesFrom>
    </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
```

```
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasPersonAge"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="LessThanOrEqual2"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasPersonAge"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="GreaterThanOrEqual0"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Person"/>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#Children
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#LessThanOrEqual20">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
</owl:Class>
<owl:Class rdf:ID="ChildrenGrade">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#ChildrenGrade
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#BabyGrade
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PersonName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#PersonName
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#PersonName
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="AdultSalary">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#AdultSalary
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#AdultSalary
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="AdultPosition">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#AdultPosition
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#AdultPosition
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#GreaterThanOrEqual0">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#ChildrenAgeSemantic"/>
  <rdfs:subClassOf>
```

```
    <owl:Class rdf:ID="BabyAgeSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="LessThanOrEqual150">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="AdultAgeSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="BabyBirthWeight">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4B.owl#BabyBirthWeight
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Baby">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:someValuesFrom rdf:resource="#GreaterThanOrEqual0"/>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#hasPersonAge"/>
          </owl:onProperty>
        </owl:Restriction>
        <owl:Restriction>
          <owl:someValuesFrom>
            <owl:Class rdf:about="#LessThanOrEqual2"/>
          </owl:someValuesFrom>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#hasPersonAge"/>
          </owl:onProperty>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
  <rdfs:subClassOf rdf:resource="#Children"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4B.owl#Baby
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="ChildrenSchoolName">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#ChildrenSchoolName
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#BabyKindergartenName
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#Person">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#Person
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#Person
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#Class"/>
</owl:Class>
<owl:Class rdf:ID="AdultDegree">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#AdultDegree
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#AdultDegree
  &lt;/p></rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
</owl:Class>
<owl:Class rdf:ID="PersonAge">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#ClassAttribute"/>
```

```
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#PersonAge
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#PersonAge
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Adult">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
    equivalent to local4A.owl#Adult
  &lt;/p>
  &lt;p style="margin-top: 0">
    equivalent to local4B.owl#Adult
  &lt;/p></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#AdultAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:Class rdf:ID="GreaterThanOrEqual20">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf rdf:resource="#AdultAgeSemantic"/>
</owl:Class>
<owl:Class rdf:about="#LessThanOrEqual2">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticWord"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#BabyAgeSemantic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#BabyAgeSemantic">
  <rdfs:subClassOf rdf:resource="http://www.upper-ontology.com/upper.owl#SemanticName"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasChildrenGrade">
  <rdfs:range rdf:resource="#ChildrenGrade"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Children"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultSemantic">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:range rdf:resource="#AdultAgeSemantic"/>
  <rdfs:domain rdf:resource="#Adult"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasChildrenSemantic">
  <rdfs:range rdf:resource="#ChildrenAgeSemantic"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:domain rdf:resource="#Children"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonName">
  <rdfs:range rdf:resource="#PersonName"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasPersonAge">
  <rdfs:range rdf:resource="#PersonAge"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabySemantic">
  <rdfs:domain rdf:resource="#Baby"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassSemantic"/>
  <rdfs:range rdf:resource="#BabyAgeSemantic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultPosition">
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#AdultPosition"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAdultDegree">
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#AdultDegree"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasBabyBirthWeight">
```

```
  <rdfs:domain rdf:resource="#Baby"/>
  <rdfs:range rdf:resource="#BabyBirthWeight"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasAdultSalary">
  <rdfs:range rdf:resource="#AdultSalary"/>
  <rdfs:domain rdf:resource="#Adult"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasChildrenSchoolName">
  <rdfs:domain rdf:resource="#Children"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttribute"/>
  <rdfs:range rdf:resource="#ChildrenSchoolName"/>
 </owl:ObjectProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultPositionType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#AdultPosition"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultDegreeType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#AdultDegree"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasChildrenGradeType">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#ChildrenGrade"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonNameType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:domain rdf:resource="#PersonName"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasBabyBirthWeightType">
  <rdfs:domain rdf:resource="#BabyBirthWeight"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasPersonAgeType">
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#PersonAge"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasAdultSalaryType">
  <rdfs:domain rdf:resource="#AdultSalary"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="hasChildrenSchoolNameType">
  <rdfs:domain rdf:resource="#ChildrenSchoolName"/>
  <rdfs:subPropertyOf rdf:resource="http://www.upper-ontology.com/upper.owl#hasClassAttributeType"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.2.1, Build 365)  http://protege.stanford.edu -->
```

ข.4 ตัวอย่างวิวที่กำกับลงบนฐานข้อมูล

1. วิวที่กำกับลงบนแบบจำลองข้อมูลเชิงวัตถุชุดที่ 1

```
CREATE OR REPLACE VIEW Person(name, age)
    AS
    SELECT a.name, a.age
    FROM PersonTab a
    UNION
    SELECT b.person.name, b.person.age
    FROM ChildrenTab b
    UNION
    SELECT c.person.name, c.person.age
    FROM AdultTab c;

CREATE OR REPLACE VIEW Children(name, age, grade, school_name)
```

```
    AS
    SELECT a.person.name, a.person.age, a.grade, a.school_name
    FROM ChildrenTab a;

CREATE OR REPLACE VIEW Baby(name, age, grade, school_name, birth_weight)
    AS
    SELECT a.person.name, a.person.age, a.grade, a.school_name, '-'
    FROM ChildrenTab a
    WHERE a.person.age <=2;

CREATE OR REPLACE VIEW Adult(name, age, position, salary, degree)
    AS
    SELECT a.person.name, a.person.age, a.position, a.salary, a.degree
    FROM AdultTab a;
```

2. วิวที่กำกับลงบนแบบจำลองข้อมูลเชิงวัตถุชุดที่ 2

```
CREATE OR REPLACE VIEW Person(name, age)
    AS
    SELECT a.name, a.age
    FROM PersonTab a
    UNION
    SELECT b.person.name, b.person.age
    FROM BabyTab b
    UNION
    SELECT c.person.name, c.person.age
    FROM AdultTab c;

CREATE OR REPLACE VIEW Children(name, age, grade, school_name)
    AS
    SELECT a.person.name, a.person.age, a.grade, a.kindergarten_name
    FROM BabyTab a;

CREATE OR REPLACE VIEW Baby(name, age, grade, school_name, birth_weight)
    AS
SELECT a.person.name, a.person.age, a.grade, a.kindergarten_name, a.birth_weight
    FROM BabyTab a;

CREATE OR REPLACE VIEW Adult(name, age, position, salary, degree)
    AS
    SELECT a.person.name, a.person.age, a.position, a.salary, a.degree
    FROM AdultTab a;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค
ผลงานที่ตีพิมพ์จากการวิจัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

An Integration of Data Sources with UML Class Models Based on Ontological Analysis

Manachaya Jamadhvaja
Department of Computer Engineering
Chulalongkorn University
Phyathai Road, Pathumwan, Bangkok 10330 Thailand
+66 2 218 6991
manachaya.j@student.chula.ac.th

Twittie Senivongse
Department of Computer Engineering
Chulalongkorn University
Phyathai Road, Pathumwan, Bangkok 10330 Thailand
+66 2 218 6996
twittie.s@chula.ac.th

ABSTRACT
Data integration is an effective method to interoperate data that reside in different sources for the purpose of providing users with a single point of access to those data. Due to data heterogeneity, data correctness and consistency are significant for integration. Richer semantics of data is a major factor in resolving conflicts among heterogeneous data sources. As UML class model represents only schema-based semantics of data (e.g. classes, attributes, and class structures), alternative methods such as ontology is useful for representing additional semantics (e.g. data values, data units, and synonym and hypernym lists). This paper proposes a method for integrating two data sources with UML class models by using an analysis of their ontologies. In our framework, ontology will be applied to describe semantics of data in each source. Then the ontologies are analysed and compared to determine their similarities and differences. The result of the comparison is used to devise an integrated ontology that will enable querying on the integrated information.

Categories and Subject Descriptors
H.2.5 [**Heterogeneous Databases**]

General Terms
Algorithms.

Keywords
Data integration, heterogeneous data sources, UML class model, semantic integration, ontology integration.

1. INTRODUCTION
Data integration has been an information systems issue concerning the techniques to combine heterogeneous data from different sources, and to allow users to transparently access all data from multiple sources via a single view [14]. The difficulty with data integration is data heterogeneity (i.e. schema heterogeneity, semantic heterogeneity) which is comparable to the problem of federated database [22]. Several research work proposes frameworks comprising three components—global schema, source schema, and mapping between global and source schemas—for resolving the problem [4]. In such frameworks, both global and source schemas can be represented in any forms of knowledge representation which is able to reveal semantics of global and source data. And the mapping specifies the relationship between two types of the schemas.

The data integration process depends on the form of knowledge representation; therefore, good representation will sort out the heterogeneity and facilitate conflict solving for the source schemas. Some examples of schema conflicts are naming conflicts (the same data represented by different names), and structure conflicts (e.g. same data represented by different data types), while some examples of semantic or data level conflicts are data value conflict (e.g. the different ranges of 'age' values), and data unit conflict (e.g. the different units of 'age') [20]. Referring to UML class model, it reveals adequate semantics for resolving only the schema conflicts. Alternative knowledge representation that has the ability to reveal additional semantics is required. Otherwise, the data integrator has to manually solve the data level conflicts.

Ontology, as an explicit specification of a conceptualisation, has the ability to represent any knowledge in the form of a set of concepts (or terms) and describable relationships between them [12]. It is an adaptive knowledge representation for defining shared vocabularies to reveal both schema-based semantics and additional semantics. In our research, a data integration process consists of a form of knowledge representation that has the ability to represent rich semantics for sorting out data heterogeneity, an integration algorithm, a data source schema which is associated with a data source, and a technique to transform the data source schema into the knowledge representation.

This paper demonstrates an application of the integration process to two data sources with UML class models. First, we select ontology as a knowledge representation. Second, the ontology integration algorithm is adapted from the methodology for integrating object-oriented data models in [24]. This algorithm compares and integrates two object-oriented data models by using some heuristics from an experiment over ten different pairs of data models. The algorithm gives satisfactory results with regards to the results of the integration conducted manually by database integrators, and also the resulting integrated data models show

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IHIS'05, November 4, 2005, Bremen, Germany.
Copyright 2005 ACM 1-59593-184-5/05/0011...$5.00.

characteristics of good object-oriented design according to some object-oriented measurement. We therefore rely on this algorithm to implement our ontology integration. In our approach, data source schemas in UML class models will be transformed into ontology-based schemas according to DSTC's Ontology Definition MetaModel which is proposed to the Object Management Group (OMG) [8]. The ontology-based schemas will be enriched with additional data semantics.

While several research work deals with the use of ontologies for integrating heterogeneous data (e.g. SIMS [1], OBSERVER [17], COIN [11], and MECOTA [25]), a number of researches specifically propose ontology integration techniques, for example OBSERVER [17], FCA-MERGE [23], PROMPT [19] and Chimaera [16]. However, these techniques tend to integrate unstructured or semi-structured data such as Web pages. By nature of such data, it is unavoidable that the proposed algorithms are complex (e.g. natural language processing is employed). On the contrary, our integration method, which is adapted from [24], focuses on an integration of two UML class models, so it is assumed that their ontologies are more structured and the algorithm proposed here should be less complex.

In Section 2, we briefly explain the use of ontology for data inference and integration. Section 3 reviews related work and discusses a comparison to our research. Our data integration framework in Section 4 is described with an example of the integration of the UML class models of two simple population databases. We demonstrate the use of the integration framework in Section 5. Finally, Section 6 concludes the paper.

2. ONTOLOGICAL INFERENCE AND INTEGRATION

Ontology provides means for sharing knowledge among information systems by defining a common vocabulary in which shared knowledge is represented [12]. As a result, ontology is a shared and common understanding that reaches across people and systems [10]. By defining UML class models as ontologies, shared vocabularies facilitate the comparison and analysis between them. Ontology will act as a mediator to handle the heterogeneity and distribution across all data sources in the integration process, and provide and interface between users and data sources in the query process.

Currently, ontology has been applied to many web applications. A number of organisations set standards to the use of ontology. For example, W3C has recommended Web Ontology Language (OWL) for publishing and sharing ontologies [2]. OWL DL (Description Logic) is an OWL sublanguage that was designed to provide a language subset that has desirable computational properties for reasoning systems. The reasoner places a number of constraints on supporting the inference capabilities. Inference engines for ontologies can reason about an ontology's schema definition [10]. Such reasoners help build ontologies and use them for advanced information access and navigation. For example, if 'baby' is subclass of 'children' in an ontology, we can automatically derive the fact that 'baby' is 'children'. Besides, if we put a rule 'children whose age are less than or equal to 2 is baby', a new fact will be inferred for some 'children' who meets this condition, such that those 'children' are also 'baby'.

OWL is developed as a vocabulary extension to RDF (Resource Description Framework) [15], so two of the representations of OWL are RDF graph and RDF triple [3], which we will use in this paper. Also Jena [7] and RDQL [21] are used as inference engine and query language for RDF triple.

In summary, the representation of the schemas and other semantic knowledge in the form of ontology give us advantages by the power of ontology integration and inference.

3. RELATED WORK

3.1 Ontology-Based Data Integration

Several researches propose approaches to apply ontologies to integrate heterogeneous data. The approaches can be classified into three groups [26] (Figure 1):

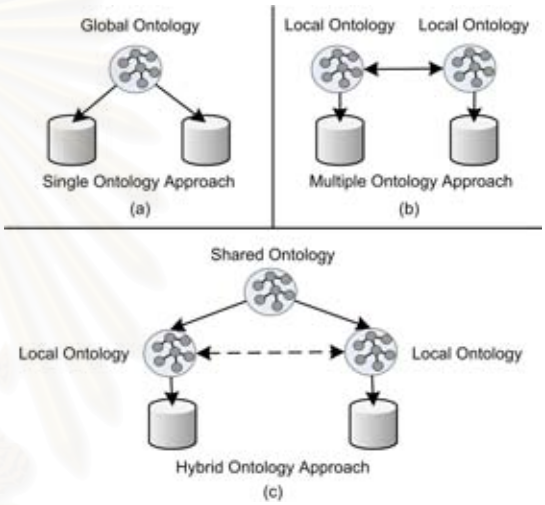


Figure 1. Ontology-based approaches for information integration [26].

1. Single Ontology Approach: One global ontology is created to describe data in all sources (Figure 1(a)), and query on the integrated data is via the global ontology. This approach looks straightforward but needs an expert who knows the meaning of all data sources to define the global ontology (e.g. SIMS [1]).
2. Multiple Ontology Approach: Data from each source is conveniently described by its own local ontology but a mapping between the local ontologies has to be established (Figure 1(b)). Query on the integrated data is via a local ontology and the mapping is used to compose local queries on other local ontologies. This approach looks more adaptive but the user has to select a suitable ontology that provides the required data as the mediator (e.g. OBSERVER [17]).
3. Hybrid Ontology Approach: This combination of the first two approaches describes data in each source with a local ontology and a shared ontology is built for sharing vocabularies among local ontologies (Figure 1(c)). This approach takes the advantages of the two approaches: ease of defining ontologies locally and of querying via the shared ontology. But mapping shared ontology onto local ontologies each time there is a query is an overhead (e.g. COIN [11] and MECOTA [25]).

Our research will combine the hybrid approach with the single ontology approach. Following the hybrid approach, data in each source is described by a local ontology and an integrated ontology is constructed for merging two local ontologies. The integrated ontology will then be associated directly to the data sources and query on the integrated data will be through the integrated ontology only, hence the single approach. Section 4 will go into more details on our approach.

3.2 Transformation of UML Class Model into Ontology

Ontology is a standard knowledge representation language which allows users to define common vocabularies as shared terms for semantically communicating and exchanging knowledge. By using ontological concepts, knowledge will be described as classes, properties and restrictions on the properties [13, 18]. As ontology language is flexible and easy to use, it has become popular in many fields of research including knowledge engineering and semantic web. Not only representing general knowledge which is unstructured or semi-structured data, ontology also represents structured data such as UML class model. Many research work deals with the transformation from UML class model into ontology. In [5], mappings between UML, DL, ER, and Ontology are proposed. Java2OWL [6] converts Java classes and instances to OWL using cardinality constraints and XML Schema datatypes.

Since the OMG has announced an RFP (Request For Proposal) for an Ontology Definition MetaModel, its requirements are to represent the semantics of ontologies, depict the ontology by using UML and so on [9]. Four organizations, including DSTC, submitted their proposal. In our research, we adopt the DSTC's proposal [8] to define the transformation of UML class model into ontology.

4. OUR FRAMEWORK

As discussed in Section 3.1, our data integration approach is a combination of the single ontology and the hybrid approaches, and we simplify the ontology integration methodology by focusing on how to integrate ontologies which describe structured information such as two UML class models. Our framework is divided into four main steps as follows (Figure 2):

1. Define: The data owner of each source, using an upper ontology for UML class models (see Section 4.1), defines their data in terms of their local ontology.
2. Add semantics: The data integrator takes part in adding some semantics such as synonym and hypernym lists to local ontologies.
3. Compare: Compare is an automatic step to compare and analyse the degree of relationships between two local ontologies. This step results in five types of relationships, i.e. equivalence, superclass, subclass, sibling and disjoint.
4. Construct & Inference: By using the output from previous step, an integrated ontology is constructed. Rules can be added to generate new facts to the resulting integrated ontology, and after that, an inference engine is employed to derive more facts about the data models. This helps refine the integrated ontology. The integrated ontology is then ready to act as the single point of access to the data sources.

Our approach gains from the expressiveness of ontology which can describe both the semantics already present in the UML class models (i.e. classes, attributes, types of attributes, inheritance, aggregation, sibling) and other additional semantics (i.e. other class and attribute semantics, synonym, and hypernym). For simplicity and due to limited space, the proposed approach will be explained through an example of the integration of two simple population data models in Figures 3(a) and 3(b). Additional semantics are also added to the example, i.e. the semantic lists for classes are attached to both models together with a synonym list in Figure 3(c). The integration process will follow the four steps mentioned above.

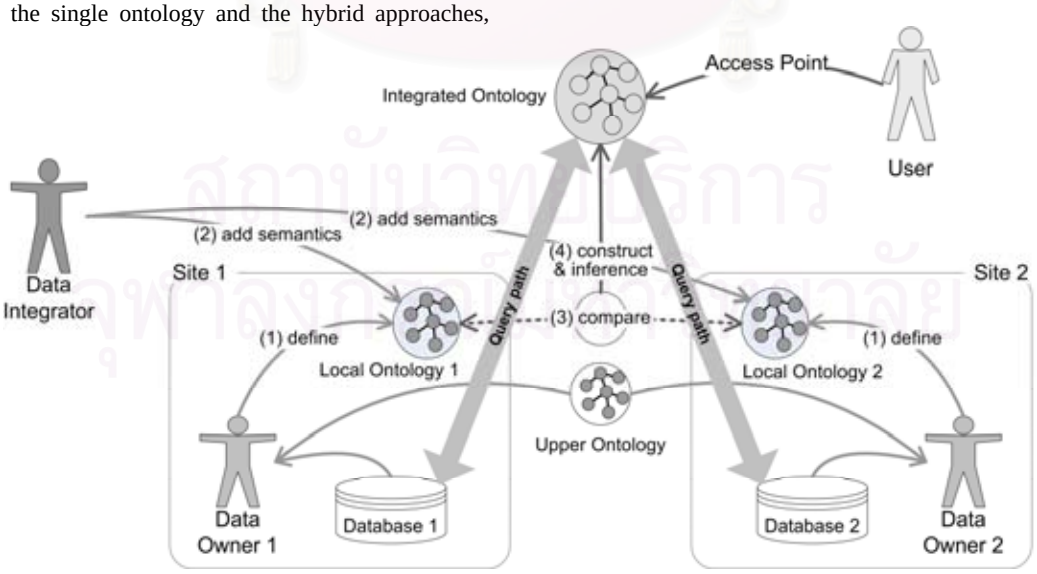


Figure 2. Our data integration framework.

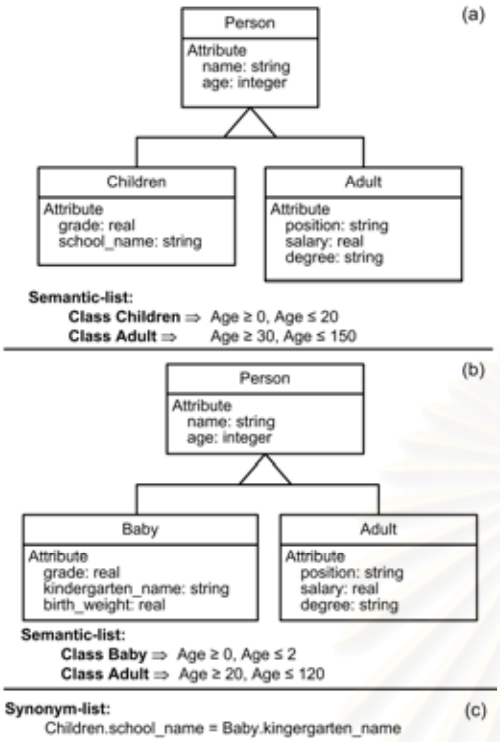


Figure 3. Example of UML class models and additional semantics.

4.1 Defining Local Ontologies

Upper ontology (the upper part of Figure 4) is a meta-ontology from which the local ontologies of the two data models will be derived (OWL ontology here is represented in RDF graph [15]). It explains what semantics should be described for the data models. Both data owners and integrator need to define or add any kinds of semantic data that conform to the given upper ontology.

From the upper ontology, the semantics of the UML class model is classified into two types: schema-based semantics and additional semantics. Schema-based semantics defines the semantics that is already present in the UML class model. In Figure 3, schema-based semantics refers to the UML class diagrams (the upper part of Figures 3(a) and 3(b)). We use two ontology classes, i.e. "Class" and "ClassAttribute", to represent object-oriented class and attribute respectively. The ontological object property "hasClassAttribute" is used to describe the property of a class having an attribute, and the ontological datatype property "hasClassAttributeType" is used to describe the data type of an attribute. Additional semantics defines other semantics that a data integrator may add to local ontologies for the purpose of integration; for example, in Figure 3, additional semantics is represented as Semantic-list (in the lower part of Figures 3(a) and 3(b)) and Synonym list (in Figure 3(c)). According to the upper ontology, the data owners/ integrator can add the additional semantics to both classes and attributes of UML class models in two ways:

1. By using "hasClassSemantic" or "hasClass AttributeSemantic" property for class semantics and attribute semantics, the semantics will be defined as "SemanticName" and "Semantic Value". For example, in Figure 5, we define semantics of Children as "ChildrenAgeSemantic" with value "≥ 0" and "≤ 20".
2. By using "owl:equivalentClass" and "owl:unionOf" property for both class and attribute, class semantics and attribute semantics will be defined for a synonym or hypernym relationship between a term in a local ontology and another term in the other local ontology. For example, in Figure 5, we define a synonym of "ChildrenSchoolName" as "owl:equivalentClass BabyKindergartenName" to show that they have the same meaning.

The semantics of the two models in Figure 3 will be represented as local ontology 1 (the middle part) and local ontology 2 (the lower part) of Figure 4 respectively. They are derived from the upper ontology in the upper part.

4.2 Comparing Local Ontologies

Ontology comparison is based on class comparison in order to find the relationships between two UML classes, which are now represented as subclasses of "Class" in the upper ontology. Class comparison (Section 4.2.3) is a bottom-up process that results from semantic comparison (Section 4.2.1) and ClassAttribute comparison (Section 4.2.2). First, each pair of the classes from the two local ontologies is selected. Second, class semantic comparison and ClassAttribute comparison are performed respectively. Finally, the relationship that this pair of classes has with each other is returned by class comparison.

4.2.1 Semantic Comparison

Semantic comparison is based on some heuristics that is derived from an experiment on the integration of ten pairs of OO data models and from a data integration expert [24]. It gives a numeric value that tells the degree of relationships between two terms that are compared. Let T1 and T2 be the terms in local ontology 1 and local ontology 2 respectively whose semantics are to be compared. α is the number of SemanticName of T1, β is the number of SemanticName of T2, and $\alpha \leq \beta$.

Each pair of "SemanticName and SemanticValue" will be compared and the heuristic value will be returned according to Table 1. The heuristic scores that reflect the result from the comparison of each semantic pair will be gathered in a set $R = [r_1, r_2, \ldots, r_\alpha]$, and $Ms = \sum r_i$ where $r_i \in R$ and $i = 1, \ldots, \alpha$ is computed. Table 2 shows the kind of relationships that T1 has with T2.

Table 1. Results of comparison of each semantics of T1 with that of T2.

Case	Semantic name	Semantic values	r
1	Same	Same	1
2	Same	Subset	0.8
3	Same	Overlap	0.8
4	Same	Disjoint	0.5
5	Different	Any	0

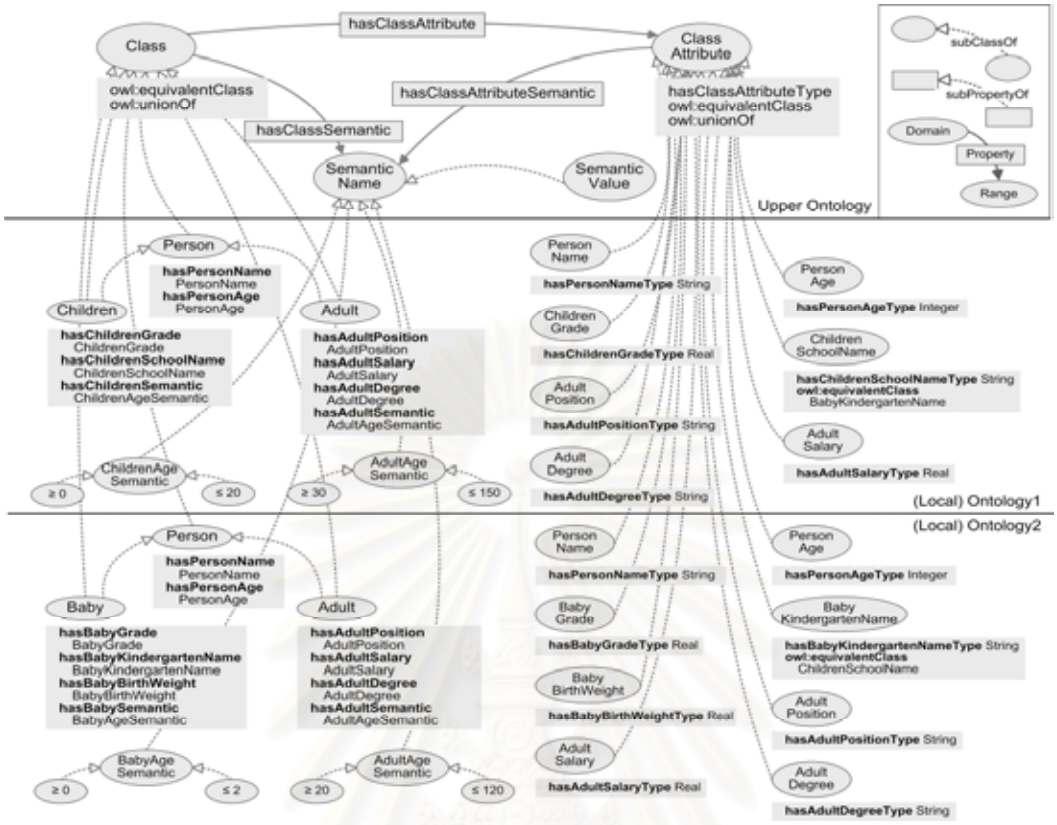


Figure 4. Upper and local ontologies.

Table 2. Relationships of terms with semantic consideration.

Case	Ms	Result	Relationship	s
1	α (α = β)	Equivalence	Equivalence	1
2	α (α < β)	Inclusion	Superclass*	1
3	≥ ¾α (α = β)	Equivalence	Equivalence	1
4	≥ ¾α (α < β)	Inclusion	Superclass*	1
5	≥ ½α	Tight-intersection	Sibling	0.1
6	< ½α	Loose-intersection	Disjoint	0
7	0	Disjoint	Disjoint	0

*T1 is Superclass of T2 = T2 is Subclass of T1

4.2.2 ClassAttribute Comparison

ClassAttribute comparison is based on the comparison of ClassAttribute and the comparison of the set of ClassAttributes.

1. ClassAttribute comparison is a process to compare each pair of ClassAttribute in a class. Each pair of ClassAttribute will be compared and the heuristic value will be returned according to Table 3.
2. Set of ClassAttributes comparison is a process to compare a set of ClassAttributes of a class to another one. Set of ClassAttributes comparison is also based on heuristics scoring from [24]. The heuristics value tells the degree of relationships between two classes that are compared. Let C1 and C2 be the classes in local ontology 1 and local ontology 2 respectively whose ClassAttributes are to be compared. θ_{C1} is the number of ClassAttributes of C1, θ_{C2} is the number of ClassAttributes of C2. Let $\alpha = \theta_{C1}$, $\beta = \theta_{C2}$, and $\alpha \leq \beta$. If a ClassAttribute in C1 has "owl:unionOf" property with other group of ClassAttributes in C2, then $\alpha = \alpha - 1$ + the number of ClassAttributes in the group. And if a ClassAttribute in C2 has "owl:unionOf" property with other group of ClassAttributes in C1, $\beta = \beta - 1$ + the number of ClassAttributes in the group.

Table 3. Results of comparison of each ClassAttribute A with ClassAttribute B.

Case	Comparison condition	s	Relationship
1	Name of A is sameAs B and their additional semantics are defined	(see Table 2)	(see Table 2)
2	Name of A is differentFrom B but both are equivalentClass, and their additional semantics are defined	(see Table 2)	(see Table 2)
3	Name of A is sameAs B but no additional semantics is defined	1	Equivalence
4	Name of A is differentFrom B but both are equivalentClass, and no additional semantics is defined	1	Synonym
5	A is unionOf B or B is unionOf A	1	Hypernym
6	A is AllDifferent from B	0	Disjoint

Each pair of ClassAttributes in C1 and C2 will be compared using the method in the ClassAttribute comparison process. The result from the comparison of each ClassAttribute pair will be gathered in a set $S = [s_1, s_2, \ldots, s_\alpha]$, and $Ms = \sum s_i$ where $s_i \in S$ and $i = 1, \ldots, \alpha$ is computed. Table 4 shows the kind of relationships that C1 has with C2.

Table 4. Relationships of classes C1 and C2, with consideration on the set of ClassAttribute.

Case	Ms	Result	Relationship
1	α (α = β)	Equivalence	Equivalence
2	α (α < β)	Inclusion	Superclass*
3	≥ ¾α (α = β)	Equivalence	Equivalence
4	≥ ¾α (α < β)	Inclusion	Superclass*
5	≥ ½α	Tight-intersection	Sibling
6	< ½α	Loose-intersection	Disjoint
7	0	Disjoint	Disjoint

*C1 is Superclass of C2 = C2 is Subclass of C1

4.2.3 Class Comparison

The relationship that a class C1 has with a class C2 is based on the relationship between their class semantics and the relationship between all of their ClassAttributes as a whole. From the method in section 4.2.1 and 4.2.2, we can obtain Relationship of ClassSemantic which indicates the kind of relationships between all of their class semantics and Relationship of the set of ClassAttributes which indicates the kind of relationships between a particular pair of their attributes. In this process, we can obtain the relationship that C1 has with C2 from Table 5.

Table 5. Summary of class relationships.

Case	Result of ClassSemantic comparison (Section 4.2.1)	Result of set of ClassAttributes comparison (Section 4.2.2)	Relationship
1	- Equivalence - Superclass - Subclass - Somenone*	Equivalence	Equivalence
2	- Equivalence - Superclass - Subclass - Somenone*	Superclass	Superclass
3	- Equivalence - Superclass - Subclass - Somenone*	Subclass	Subclass
4	- Equivalence - Superclass - Subclass - Somenone*	Sibling	Sibling
5	- Disjoint	Sibling	Sibling

*Either or both of the two classes do not define class semantics

Figure 5 shows the comparison of classes 'Children' and 'Baby' from Figure 4. The comparison follows the steps in section 4.2.1, 4.2.2 and 4.2.3. The result is that 'Children' has ClassSemantics and the set of ClassAttribute that are superclass of those of 'Baby', and therefore 'Children' is superclass of 'Baby'. Other pairs of classes in the two ontologies will be compared in a similar manner.

4.3 Constructing Integrated Ontology

Ontology comparison results in the relationship between each pair of the classes that are compared being identified (i.e. equivalence, superclass, subclass, or sibling). From the result, we can construct a new integrated ontology based on the resulting relationships. Table 6 shows the result of the comparison between pairs of the classes in our example.

Table 6. Result of the comparison between local ontology 1 and local ontology 2.

	Ontology1. Person	Ontology1. Children	Ontology1. Adult
Ontology2.Person	Equivalence	Superclass	Superclass
Ontology2.Baby	Subclass	Subclass	Sibling
Ontology2.Adult	Subclass	Sibling	Equivalence

The resulting relationship between each pair of the compared classes determines the order of class integration. Before integrating a class in one ontology with any of the classes in the other ontology, the relationship that the class has with each class in the other ontology must be determined. We suggest that the equivalence relationship has the highest priority, and so the two classes with this kind of relationship should be integrated first. This can be followed by the integration of the classes with the lower-priority relationships, i.e. superclass, subclass, and sibling respectively. Figure 6 presents the integrated ontology of our population example which depicts 'Person' as the root class with 'Children' and 'Adult' as its subclasses whereas 'Baby' also inherits from 'Children'.

From integrated ontology, adding some inference rules from data integrator's knowledge will increase the ability and correctness of the integrated ontology when processing the query. For example, with the semantics of 'Children' and 'Baby', the integrator can add the fact that 'Children' whose age is less than or equal to 2 is also 'Baby'.

In the case that the data integrator is not satisfied with the integrated ontology, it is recommended to return to the 'Add semantic' step to add more meanings to the local ontologies so as to make them capture richer semantics. The 'Compare' and 'Construct' steps will then be repeated.

4.4 Applying Integrated Ontology to Data Sources

Since the integrated ontology is created and inferred as shown in Figure 6, all resulting classes in the integrated ontology will be mapped to the local ontologies. By using this mapping, the integrated ontology will be annotated to the data sources, e.g. as a view over the DBMS which manages the data sources. A query through the integrated ontology will be passed and translated as queries on the data sources. Example of query will be demonstrated in the next section.

5. QUERY EXAMPLE

We present two query examples for demonstrating paths and translations of query which is submitted to our framework. Query is in terms of RDQL (RDL Data Query Language) [21], which is suitable for extracting information from RDF triples. After the query is submitted to the integrated ontology, the integrated ontology passes RDQL to the DBMS of all data sources. Then a view, which has been created previously for supporting annotation of the integrated ontology, takes part to translate RDQL to other query language based on the DBMS language.

Semantic comparison	
Ontology2.Baby ⊂ Ontology1.Children ⇒ Ontology1.Children.Age has Ontology2.Baby.Age as a subset ∴ R = [0.8], Ms = 0.8, α = 1, β = 1 Result = Equivalence **Relationship = Equivalence**	- Case 2 in Table 1 - Case 3 in Table 2
ClassAttribute comparison	
Ontology1.Children.PersonName is sameAs Ontology2.Baby.PersonName, but they have no additional semantics. ⇒ return Equivalent s = 1	- Case 3 in Table 3
Ontology1.Children.PersonAge is sameAs Ontology2.Baby.PersonAge, and semantics of Ontology2.Baby.PersonAge ⊂ semantics of Ontology1.Children.PersonAge ⇒ return Superclass s = 1	- Case 1 in Table 3 - Case 3 in Table 2 (also from semantic comparison)
Ontology1.Children.ChildrenGrade is sameAs Ontology2.Baby.BabyGrade, but they have no additional semantics. ⇒ return Equivalent s = 1	- Case 3 in Table 3
Ontology1.Children.ChildrenSchoolName is differentFrom Ontology2.Baby.BabyKindergartenName, but Ontology1.Children.ChildrenSchoolName is equivalentClass to Ontology2.Baby.BabyKindergartenName, and they have no additional semantics. ⇒ return Synonym s = 1	- Case 4 in Table 3
∴ $Ms_{SetOfClassAttribute}$ = 1+1+1+1 = 4, α = 4, β = 5 Result = Inclusion **Relationship = Superclass**	- Case 2 in Table 4
Class comparison	
∴ From Table 5, Relationship of ClassSemantic + Relationship of set of ClassAttribute **= Equivalence + Superclass = Superclass**	- Case 2 in Table 5

Figure 5. Comparison between class 'Children' in local ontology 1 and class 'Baby' in local ontology 2.

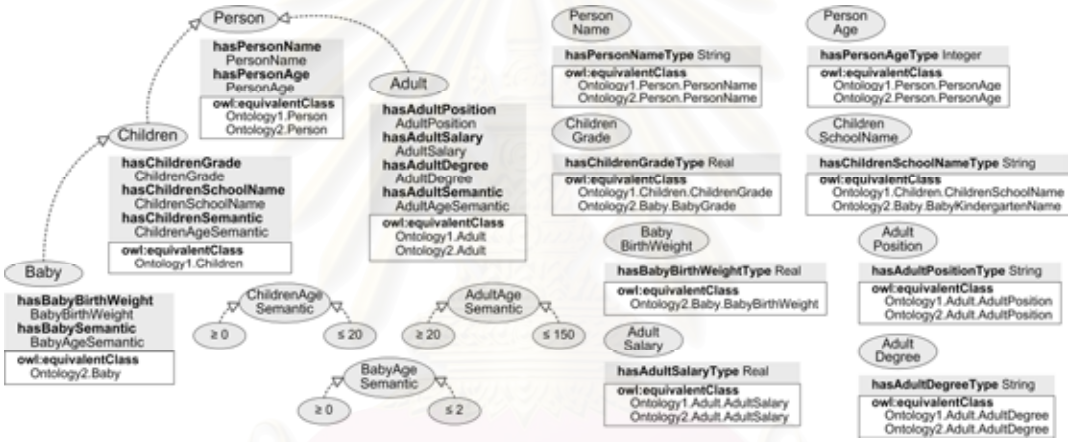


Figure 6. Integrated ontology.

Example 1.
```
SELECT ?AdultPosition, ?AdultSalary
WHERE  (?AdultPosition info:position
        ?AdultSalary)
AND ?AdultSalary <= 20000
USING info FOR <http://example.org/peopleInfo#>
```

From integrated ontology in Figure 6, the query in Example 1 wants to find the positions with a salary less than or equal to 20000. This query will be passed to the DBMS of both data sources. At data sources 1 and 2 which are described by local ontologies 1 and 2 respectively, this query will be translated to such an SQL query as shown below. The DBMS will manage to send the new SQL query and return the result back to the integrated ontology.

```
SELECT Adult.position, Adult.salary
FROM Adult
WHERE Adult.salary <= 20000
```

Example 2.
```
SELECT ?ChildrenName, ?ChildrenAge
WHERE  (?Children info:age ?ChildrenAge)
AND ?ChildrenAge > 1
USING info FOR <http://example.org/peopleInfo#>
```

From integrated ontology in Figure 6, the query in Example 2 wants to find the names and ages of the children who are older than one-year old. This query will be passed to the DBMS of both data sources. At source 1, this query will be translated to such an SQL query as shown below.

```
SELECT Children.name, Children.age
FROM Children
WHERE Children.age > 1
```

At source 2, this query will be translated to an SQL query as shown below.

```
SELECT Baby.name, Baby.age
FROM Baby
WHERE Children.age > 1
```

6. CONCLUSION

The use of ontology for integrating structured data such as UML class models gains advantage from the expressiveness of ontology, and achieves good result for integration. A query through the integrated ontology will be passed and translated as

queries onto the data sources. Users can also benefit from describing semantics by ontology because query can then be semantics-based by the inference power of ontology. For example, a query for the data of all children will result in the data of children from data source 1 and of baby from data source 2 to be returned.

This paper reports the progress of the prototype of the ontology integration framework. We are also in the process of refining the ontology integration approach. We expect more kinds of semantics to be included in the local ontologies, such as integrity constraints on attributes. Further evaluation of the approach is also a future work. As mentioned before, the integration algorithm has been evaluated by data integration experts and object-oriented measurement scheme. We expect to try it with data models that are used in real applications and compare the performance against other research works.

7. ACKNOWLEDGMENTS

This work is part of "A Design and Development of Enterprise System with Distributed System Technology" research supported by the Thai Government-Industry Linkage Research Grant (Fiscal Year 2004).

8. REFERENCES

[1] Arens, Y., Hsu, C.-N., et al. Query Processing in the SIMS Information Mediator. in *Readings in agents*, Morgan Kaufmann Publishers Inc., 1998, 82 - 90.

[2] Bechhofer, S., Harmelen, F.v., et al. OWL Web Ontology Language Reference. Dean, M., et al. eds. *W3C Recommendation*, W3C, 2004.

[3] Beckett, D. RDF/XML Syntax Specification (Revised). McBride, B. ed. *W3C Recommendation*, W3C, 2004.

[4] Cali, A., Reasoning in Data Integration Systems: Why LAV and GAV Are Siblings. in *The 14th International Symposium on Methodologies for Intelligent Systems (ISMIS 2003)*, (2003), 562-571.

[5] Colomb, R.M., Gerber, A., et al., Issues in Mapping Metamodels in the Ontology Development Metamodel Using QVT. in *The 1st International Workshop on the Model-Driven Semantic Web (MSDW 2004)*, (Monterey, California, USA, 2004).

[6] Dean, M. Java2OWL, DAML.org, 2003.

[7] Dickinson, I. The Jena 2 Ontology API, Jena Semantic Web Framework, 2005.

[8] DSTC. Ontology Definition MetaModel - DSTC Initial Submission, Object Management Group, 2003, 70.

[9] DSTC, Gentleware, et al. Ontology Definition MetaModel *Preliminary Revised Submission to OMG RFP ad/2003-03-40 Volume 1*, 2004, 169.

[10] Fensel, D., van Harmelen, F., et al. OIL: an ontology infrastructure for the Semantic Web. *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, *16* (2). 38-45.

[11] Goh, C.H., Bressan, S.p., et al. Context Interchange: New Features and Formalisms for the Intelligent Integration of Information. *ACM Transactions on Information Systems (TOIS)*, *17* (3). 270-293.

[12] Gruber, T.R. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition (1042-8143)*, *5* (2). 199 - 220.

[13] Horridge, M., Knublauch, H., et al. *A Practical Guide To Building OWL Ontologies Using The Protege-OWL Plugin and CO-ODE Tools*. The University Of Manchester, 2004.

[14] Lenzerini, M., Data Integration: A Theoretical Perspective. in *The 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (Madison, Wisconsin, 2002), ACM Press New York, NY, USA, 233 - 246.

[15] McBride, B. An Introduction to RDF and the Jena RDF API, http://jena.sourceforge.net/tutorial/RDF_API/index.html, 2005.

[16] McGuinness, D.L., Fikes, R., et al., An Environment for Merging and Testing Large Ontologies. in *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, (Breckenridge, Colorado, 2000).

[17] Mena, E., Illarramendi, A., et al. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. *Distributed and Parallel Databases -- An International Journal*, *8* (2). 223-271.

[18] Noy, N.F. and McGuinness, D.L. Ontology Development 101: A Guide to Creating Your First Ontology *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*, 2001.

[19] Noy, N.F. and Musen, M.A., PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. in *The Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, (2000), AAAI Press / The MIT Press, 450 - 455.

[20] Ram, S. and Park, J. Semantic conflict resolution ontology (SCROL): an ontology for detecting and resolving data and schema-level semantic conflicts. *Knowledge and Data Engineering, IEEE Transactions on*, *16* (2). 189-202.

[21] Seaborne, A. RDQL - A Query Language for RDF *W3C Member Submission 9*, W3C, 2004.

[22] Sheth, A.P. and Larson, J.A. Fedetated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, *22* (3). 183-236.

[23] Stumme, G. and Maedche, A., FCA-MERGE: Bottom-Up Merging of Ontologies. in *The Seventeenth International Joint Conference on Artificial Intelligence*, (Seattle, Washington, USA, 2001).

[24] Swasrukkiet, S. A Methodology for Integration of Object-Oriented Data Models Using Heuristics and Analysis of Relationships between Class. *Department of Computer Engineering*, Chulalongkorn University, Bangkok, 1999, 162.

[25] Wache, H., Scholz, T., et al., An integration method for the specification of rule-oriented mediators. in *Database Applications in Non-Traditional Environments, 1999. (DANTE '99) Proceedings. 1999 International Symposium on*, (1999), 109-112.

[26] Wache, H., Vogele, T., et al., Ontology-Based Integration of Information — A Survey of Existing Approaches. in *The Seventeenth International Joint Conference on Artificial Intelligence*, (Seattle, Washington, USA, 2001).

An Approach to Constructing an Integrated Ontology for Integrating Object-Oriented Data Models

Manachaya Jamadhvaja* , Twittie Senivongse**

ABSTRACT

Semantically determining similarities and differences between data from multiple data sources plays an important role in the process of data integration. The similarities and differences identify the degree of relationships and are used for constructing the integrated data model. In structured data such as object-oriented database, their data models reveal only structure-based semantics without other additional semantics. This paper proposes an approach to integrate two object-oriented data models based on the integration of ontologies that describe their semantics. The ontology for an object model describes semantics based on the model structure and additional semantics, i.e. class and attribute semantics, and synonym and hypernym lists. The ontologies are analysed to devise an integrated data model that will be used to facilitate query on the integrated data.

KEYWORD : Ontology Integration, Data Model Integration, Ontology, Object-Oriented Data Model.

1. INTRODUCTION

The growth of information technology affects the use of data in information systems. In organisations, using a single data source is not much enough to run the business. Hence, data model integration is required for combining heterogeneous data from different data sources and for providing users with a single point of access to all data sources [9]. The data models that are integrated tend to be in the same context but are designed by different owners (e.g. the data models represent the population databases of different departments in a district). This will result in several kinds of semantic heterogeneity, for example, name conflict (the same information represented by different names or different information represented by the same name), structure conflict (e.g. same information represented by different data types), and scaling conflict (e.g. an attribute 'salary' with different currencies in different data models) [14]. To integrate two data models, a database integrator must understand well the meaning of both data models and can identify the similarities and differences in the semantics that underlies the data models.

For structured data such as object-oriented databases, structure-based semantics are represented as shown in UML class diagram in Figure 1. UML class diagram describes the semantics of an object-oriented data model in terms of classes and attributes that form the model as well as the relationships between them, but it does not reveal any other meaning inside the data.

The use of ontologies for describing the meaning inside the data is a practical method to solve the problem of semantic heterogeneity. Ontology, an explicit specification of a conceptualization [13], represents knowledge in a domain of interest in the form of a set of concepts (or terms) and describable relationships between them. W3C has recommended Web Ontology Language (OWL) for publishing and sharing ontologies [11]. OWL is developed as a vocabulary extension to RDF (Resource Description Framework), so one of the representations of OWL is RDF graph [1]. By using ontological concepts to represent semantics in the data models, shared vocabularies are defined to illustrate both structure-based semantics and other additional semantics in the models. As a result, ontologies will handle the heterogeneity and distribution across all data sources, and be the mediator between users and data sources. Ontologies will facilitate flexible semantics-based query on the data sources. Several research work deals with the use of ontologies for integrating heterogeneous data (e.g. SIMS [15], OBSERVER [5], COIN [2] and MECOTA [7]).

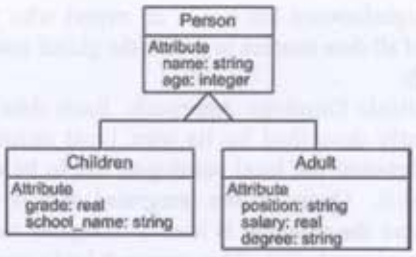


Fig. 1. Example of object-oriented data model

This paper proposes an approach to integrate two object-oriented data models whose semantics are described by ontologies. Since there is a mapping between object model and ontology, we follow such a suggestion from DSTC's Ontology Definition MetaModel [4] to develop an ontology integration approach by adapting from the methodology for integrating object-oriented data models in [12]. All classes from the two data models are analysed in order to determine the degree of relationships between them before the integration can be applied. Several kinds of relationships between ontological concepts within the models are considered, i.e. inheritance, aggregation, sibling, equivalence, synonym, and hypernym. The degree of similarities and differences between the ontologies are identified by heuristics-based scoring.

In Section 2, we review related work and discuss a comparison to our research. Our ontology integration method in Section 3 is described with an example of the integration of the object-oriented data models of two population databases. Finally, Section 4 discusses the use of this approach and concludes the paper.

2. RELATED WORK

2.1 Ontology-based information integration

Several research work suggests ways to use ontologies for heterogeneous data integration. In summary, all research can be categorised into three approaches [8]:

1. Single Ontology Approach: One global ontology is used to describe all data sources (Figure 2(a)), and query on the integrated data is via the global ontology. This approach

*Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University
**Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University

looks straightforward but needs an expert who knows the meaning of all data sources to define the global ontology (e.g. SIMS [15])

2. Multiple Ontology Approach: Each data source is conveniently described by its own local ontology but a mapping between the local ontologies has to be established (Figure 2(b)). Query on the integrated data is via a local ontology and the mapping is used to compose local queries on other local ontologies. This approach looks more adaptive but mapping local ontologies each time there is a query is an overhead (e.g. OBSERVER [5]).

3. Hybrid Ontology Approach: This combination of the first two approaches describes each data source with a local ontology and a shared ontology is built to for sharing vocabularies among local ontologies (Figure 2(c)). This approach takes the advantages of the two approaches: ease of defining ontologies locally and of querying via the shared ontology (e.g. COIN [2] and MECOTA [7]). Our research follows this hybrid approach.

2.2 Ontology integration

Most of research in ontology integration presents techniques for integrating ontologies which describe knowledge about unstructured or semi-structured data such as Web pages. By nature of such data, it is unavoidable that the proposed algorithms are complex (e.g. natural language processing is employed). For example, OBSERVER [1] presents a method that performs the translation from a local ontology into other local ontologies for cross-ontology query; no integrated ontology is actually created. On the contrary, FCA-MERGE [6], PROMPT [10], and Chimaera [3] propose ideas for constructing integrated ontology. FCA-MERGE [6] applies mathematical techniques to derive concepts of ontology, and produces suggestions for transforming into an integrated ontology. PROMPT [10] and Chimaera [3] are interactive tools that can perform some integration tasks automatically and guide the ontology integrator in performing other tasks.
Our research focuses on integrating two object-oriented data models, so it is assumed that the semantics of these models are more structured and the algorithm proposed here should be less complex.

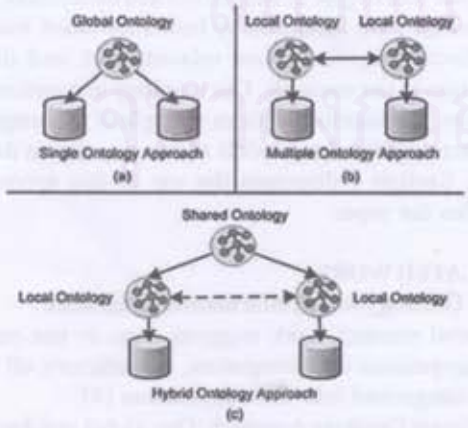


Fig. 2. Ontology-based approaches for information integration [8]

3. INTEGRATING ONTOLOGIES

For simplicity and due to limited space, the proposed approach to integrating ontologies for object-oriented data models will be explained through an example of the integration of two simple population data models in Figures 1 and 3. First, the semantics of these two models will be represented as two local ontologies based on the guideline in [4] (Section 3.1). Then they are compared in Section 3.2 and integrated in Section 3.3 using the methodology adapted from [12].

Our approach gains semantic richness of ontology and describe both the semantics that is already present in the data models (i.e. class, attribute, inheritance, aggregation, sibling) and the other additional semantics (i.e. other class and attribute semantics, equivalence, synonym, hypernym).

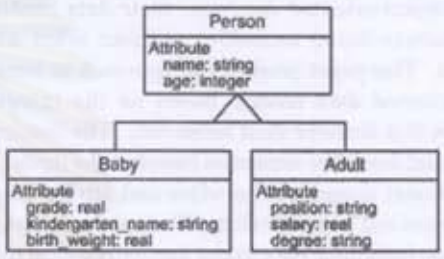


Fig. 3. Another example of object-oriented data model

3.1 Using upper ontology to derive local ontologies

Upper ontology (Figure 4) is a meta-ontology from which the local ontologies of the two object-oriented data models will be derived (OWL ontology here is represented in RDF graph). It explains what semantics should be described for the data models.

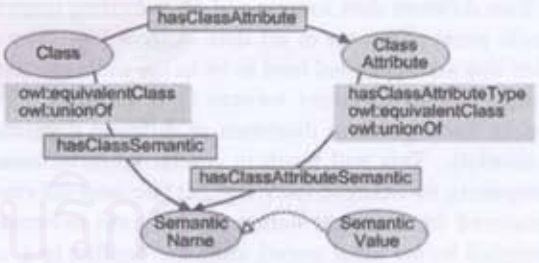


Fig. 4. Upper ontology

From the upper ontology, the semantics of the object-oriented data model are classified into two types: structure-based semantics and additional semantics. Structure-based semantics defines the semantics that is already present in the object-oriented data models. We use two ontology classes, i.e. "Class" and "ClassAttribute", to represent object-oriented class and attribute respectively. The ontology object property "hasClassAttribute" is used to describe the property of a class having an attribute, and the ontology datatype property "hasClassAttributeType" is used to describe data type of an attribute. Additional semantics defines other semantics that a data model integrator may add to the ontology for the purpose

of integration. According to the upper ontology, the additional semantics can be added to both OO classes and attributes in two ways:

1. By using "hasClassSemantic" or "hasClass Attribute Semantic" property for class semantics and attribute semantics, the semantics will be defined as "SemanticName" and "Semantic Value". For example, in Figure 5, we define semantics of Children as "ChildrenAgeSemantic" with value "≥ 0" and "≤ 20".

2. By using "owl:equivalentClass" and "owl:unionOf" property for both class and attribute, class semantics and attribute semantics will be defined as what term in a local ontology is a synonym or hypernym of what term in the other local ontology. For example, in Figure 5, we define a synonym of "ChildrenSchoolName" as "owl:equivalentClass BabyKindergartenName" to show that their meanings are the same.

The semantics of the two models in Figures 1 and 3 will be represented as local ontology 1 (upper side) and 2 (lower side) in Figure 5 respectively. They are derived from the upper ontology in the middle.

3.2 Ontology comparison

Ontology comparison is based on class comparison to find out the relationships between object-oriented classes, which are represented by the ontology class inherited from "Class" in the upper ontology. Class comparison (Section 3.2.3) is a bottom-up process, resulting from semantic comparison (Section 3.2.1) and ClassAttribute comparison (Section 3.2.2). Firstly, each pair of the classes from the two local ontologies is picked one by one. Secondly, class semantic comparison and ClassAttribute comparison are performed respectively. Finally, the relationship that this pair of classes has with each other is returned by class comparison. The process of ontology comparison is depicted in Figure 6.

3.2.1 Semantic comparison

Semantic comparison is based on some heuristics that is derived from an experiment on the integration of ten pairs of OO data models and from a data integration expert [11]. It gives a numeric value that tells the degree of relationship between two terms that are compared. Let T1 and T2 be the terms in local ontology 1 and 2 respectively whose semantics are to be compared. α is the number of SemanticName of T1, β is the number of SemanticName of T2, and $\alpha \leq \beta$.

Each pair of "SemanticName and SemanticValue" will be compared and the heuristic value will be returned according to Table 1. The result of each semantic pair will be gathered in a set $R = [r_1, r_2, \ldots, r_\alpha]$, and $Ms = \Sigma r_i$ where $r_i \in R$ and $i = 1, \ldots, \alpha$ is computed. Table 2 shows the kind of relationship that T1 has with T2.

3.2.2 ClassAttribute comparison

ClassAttribute comparison is based on the comparison of ClassAttribute and the comparison of the set of ClassAttribute.

1. ClassAttribute comparison is a process to compare each pair of ClassAttribute in a class. Each pair of Class Attribute will be compared and the heuristic value will be returned according to Figure 7.

Table 1. Result of comparison of each semantics of T1 with that of T2

Semantic name	Semantic values	r
Same	Same	1
Same	Subset	0.8
Same	Overlap	0.8
Same	Disjoint	0.5
Differrent	Any	0

Table 2. Relationships of Terms with Semantic Consideration

Ms	Result	Relationship	s
α ($\alpha = \beta$)	Equivalence	Equivalence	1
α ($\alpha < \beta$)	Inclusion	Superclass*	1
$\geq 3/4\alpha$ ($\alpha = \beta$)	Equivalence	Equivalence	1
$\geq 3/4\alpha$ ($\alpha < \beta$)	Inclusion	Superclass*	1
$\geq 1/2\alpha$	Tight-intersection	Sibling	0.1
$< 1/2\alpha$	Lose-intersection	Disjoint	0
0	Disjoint	Disjoint	0

*T1 is Superclass of T2 = T2 is Subclass of T1

2. Set of ClassAttribute comparison is a process to compare a set of ClassAttribute of a class to another one. Set of ClassAttribute comparison is also based on heuristics scoring from [11]. The heuristics value tells the degree of relationship between two classes that are compared. Let C1 and C2 be the classes in local ontology 1 and 2 respectively whose ClassAttributes are to be compared. θ_{C1} is the number of ClassAttribute of C1, θ_{C2} is the number of ClassAttribute of C2. Let $\alpha = \theta_{C1}, \beta = \theta_{C2}$, and $\alpha \leq \beta$. If a ClassAttribute in C1 has "owl:unionOf" property with other group of ClassAttribute in C2, then $\alpha = \alpha - 1 +$ number of ClassAttribute in the group. And if a ClassAttribute in C2 has "owl:unionOf" property with other group of ClassAttribute in C1, $\beta = \beta - 1 +$ number of ClassAttribute in the group.

Each pair of ClassAttribute in C1 and C2 will be compared using the method in the ClassAttribute comparison process. The result of each ClassAttribute pair will be gathered in a set $S = [s_1, s_2, \ldots, s_\alpha]$, and $Ms = \Sigma s_i$ where $s_i \in S$ and $i = 1, \ldots, \alpha$ is computed. Table 3 shows the kind of relationship that C1 has with C2.

Table. 3 Relationships of classes C1 and C2, with consideration on the set of ClassAttribute

Ms	Result	Relationship
α ($\alpha = \beta$)	Equivalence	Equivalence
α ($\alpha < \beta$)	Inclusion	Superclass*
$\geq 3/4\alpha$ ($\alpha = \beta$)	Equivalence	Equivalence
$\geq 3/4\alpha$ ($\alpha < \beta$)	Inclusion	Superclass*
$\geq 1/2\alpha$	Tight-intersection	Sibling
$< 1/2\alpha$	Lose-intersection	Disjoint
0	Disjoint	Disjoint

*C1 is Superclass of C2 = C2 is Subclass of C1

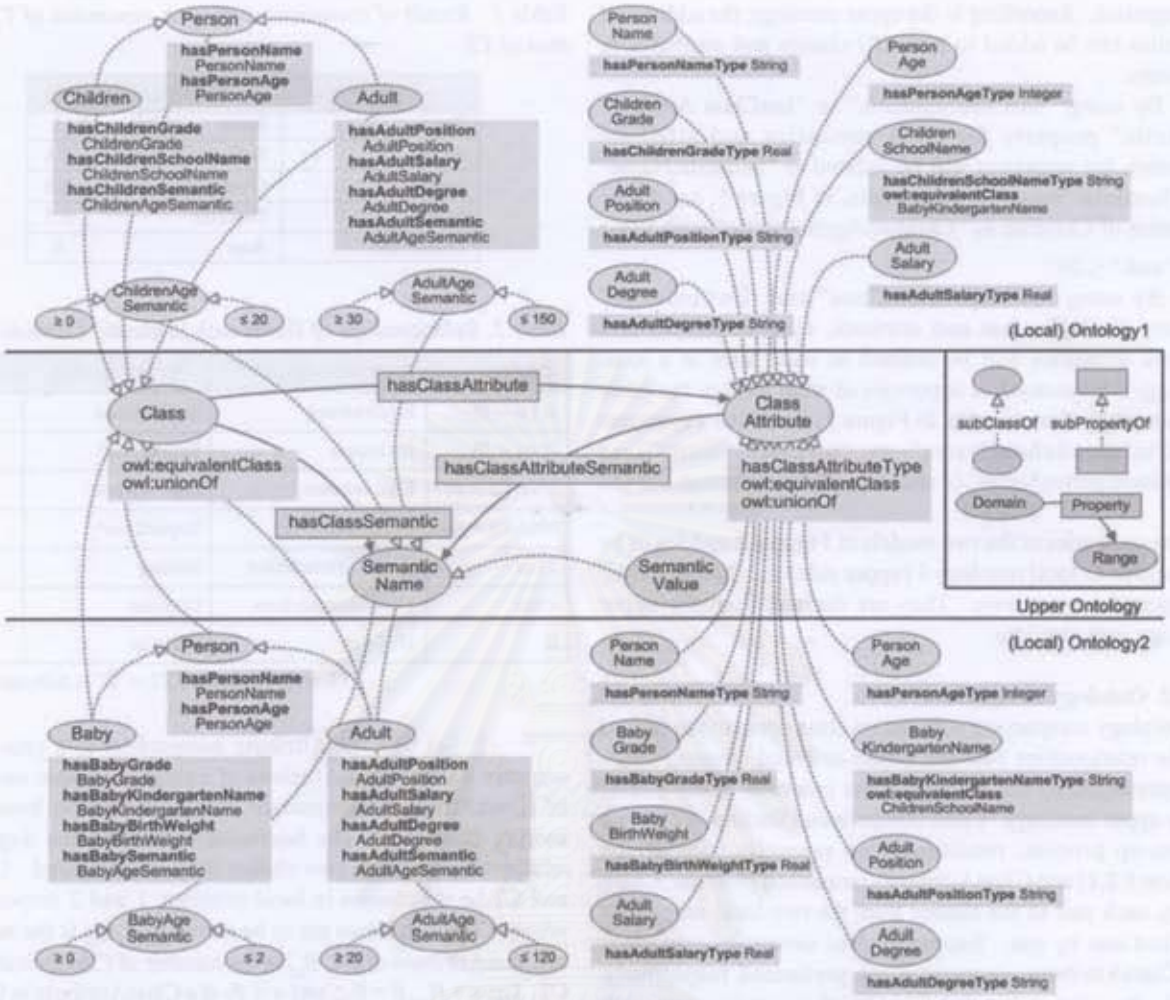


Fig. 5. Local ontologies

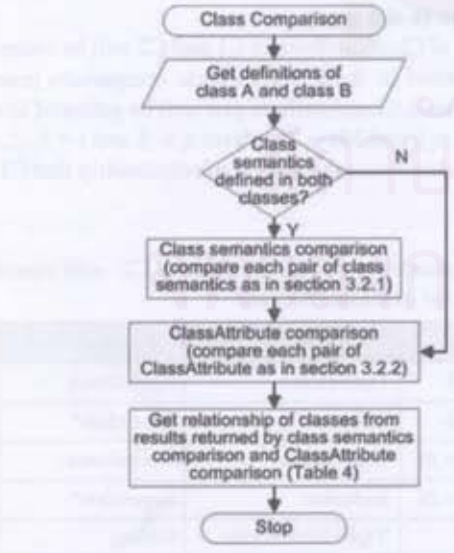


Fig. 6. Class comparison

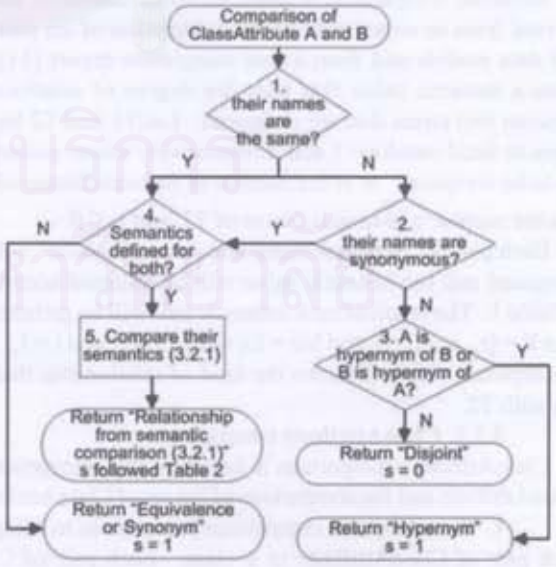


Fig. 7. ClassAttribute comparison

3.2.3 Class comparison

The relationship that a class C1 has with a class C2 is based on the relationship between their class semantics and the relationship between all of their ClassAttributes as a whole. From the method in section 3.2.1 and 3.2.2, we can obtain Relationship of ClassSemantic which indicates the kind of relationship between all of their class semantics and Relationship of the set of ClassAttribute which indicates the kind of relationship between a particular pair of their attributes. In this process, we can obtain the relationship that C1 has with C2 from Table 4.

Figure 8 shows the comparison of classes 'Children' and 'Baby' from Figure 5. The comparison follows the steps in section 3.2.1, 3.2.2 and 3.2.3. The result is that 'Children' has ClassSemantics and the set of ClassAttribute that are superclass of those of 'Baby', and therefore 'Children' is superclass of 'Baby'. Other pairs of classes in the two ontologies will be compared in a similar manner.

3.3 Ontology integration

Ontology comparison results in the relationship between each pair of the classes that are compared being identified (i.e. equivalence, superclass, subclass, or sibling). From the result, we can construct a new integrated ontology based on the resulting relationships. Table 5 shows the result of the comparison between pairs of the classes in our example. The class relationships in this table lead to the creation of the integrated ontology. Figure 9 presents the final integrated ontology of our population example which depicts 'Person' as the root class with 'Children' and 'Adult' as its subclass whereas 'Baby' also inherits from 'Children'.

Table 4: Summary of class relationships

Result of Class Semantic comparison	Result of set of Class Attributes comparison	Relationship
• Equivalence • Superclass • Subclass • Someone*	Equivalence	Equivalence
• Equivalence • Superclass • Subclass • Someone*	Superclass	Superclass
• Equivalence • Superclass • Subclass • Someone*	Subclass	Subclass
• Equivalence • Superclass • Subclass • Someone*	Sibling	Sibling
• Disjoint	Sibling	Sibling

*Either or none of the two classes have ClassSemantics

Semantic comparison

Ontology1.Children.Age has Ontology2.Baby.Age as a subset
$\therefore$ R = [0.8], Ms = 0.8, $\alpha = 1$, $\beta = 1$ Result = Equivalence **Relationship = Equivalence**

Class Attribute comparison

Ontology1.Children.PersonName has no additional semantics but has the same name as Ontology2.Baby.PersonName
$\Rightarrow$ return Equivalent s = 1
Ontology1.Children.PersonAge and Ontology2.Baby.PersonAge have class semantics that are subset and also have the same name
$\Rightarrow$ return Superclass s = 1
Ontology1.Children.ChildrenGrade has no additional semantics but has the same name as Ontology2.Baby.BabyGrade
$\Rightarrow$ return Equivalent s = 1
Ontology1.Children.ChildrenSchoolName and Ontology2.Baby.BabyKindergartenName have no additional semantics but they are synonymous
$\Rightarrow$ return Synonym s = 1
$\therefore Ms_{SetOfClassAttribute} = 1+1+1+1 = 4$, $\alpha = 4$, $\beta = 5$ Result = Inclusion **Relationship = Superclass**

Class comparison

$\therefore$ From Table 4, Relationship of ClassSemantic + Relationship of set of ClassAttribute
= Equivalence + Superclass = Superclass

Fig. 8. Comparison between class Children in local ontology 1 and class Baby in local ontology 2

Table. 5 Result of the comparison between local ontology 1 and local ontology 2

	Ontology1.Person	Ontology1.Children	Ontology1.Adult
Ontology2.Person	Equivalence	Superclass	Superclass
Ontology2.Baby	Subclass	Subclass	Sibling
Ontology2.Adult	Subclass	Sibling	Equivalence

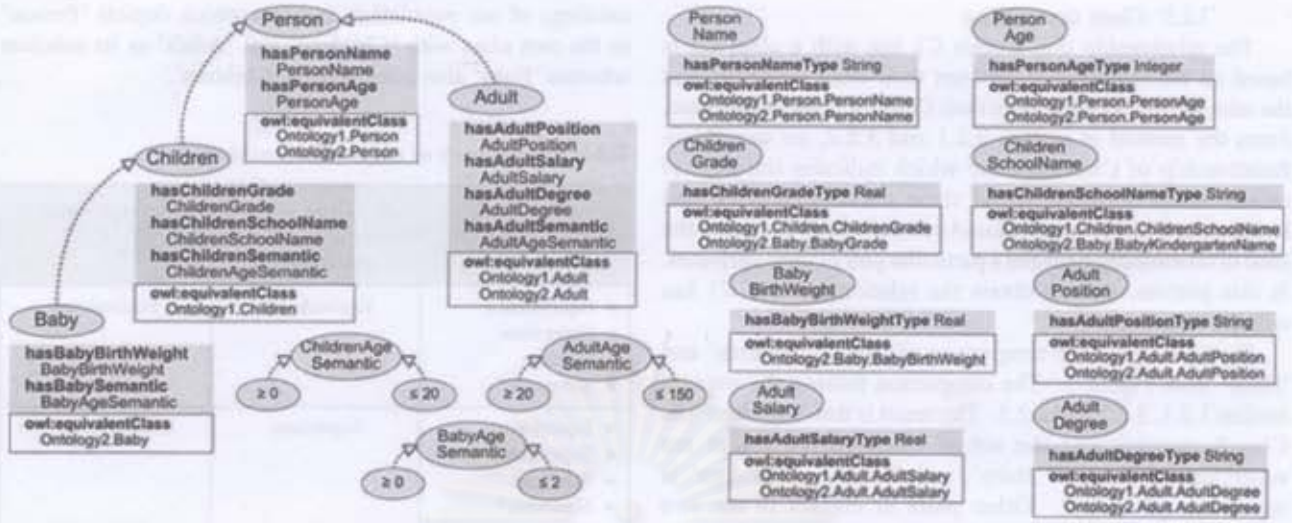


Fig. 9. Integrated ontology

4. CONCLUSION

From ontology integration, all resulting classes in the integrated ontology will be mapped to the local ontologies. A query through the integrated ontology will be translated as queries on the local ontologies of the underlying databases. Users can also benefit from describing semantics by ontology because query can then be semantics-based by the inference power of ontology. For example, a query for the data of all children will result in the data of children from data model 1 and of baby from data model 2 to be returned.

We are in the process of refining the ontology integration approach. We expect more kinds of semantics to be included in the local ontologies, such as integrity constraints on attributes. The prototype of the ontology integration and query framework is also in progress.

5. REFERENCE

[1] B. McBride, "An Introduction to RDF and the Jena RDF API," vol. 2005: http://jena.sourceforge.net/tutorial/RDF_API/index.html, 2005.

[2] C. H. Goh, et al, "Context Interchange: New Features and Formalisms for the Intelligent Integration of Information," ACM Transactions on Information Systems (TOIS), vol. 17, pp. 270-293, 1999.

[3] D. L. McGuinness, et al, "An Environment for Merging and Testing Large Ontologies," Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, 2000.

[4] DSTC, "Ontology Definition MetaModel – DSTC Initial Submission," submit to Object Management Group, 2003.

[5] E. Mena, et al, "OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies," Distributed and Parallel Databases, vol. 8, pp. 223-271, 2000.

[6] G. Stumme and A. Maedche, "FCA-MERGE: Bottom-Up Merging of Ontologies," Proceedings of 17th International Joint Conference on Artificial Intelligence, Seattle, Washington, USA, 2001.

[7] H. Wache, et al, "An integration method for the specification of rule-oriented mediators," Proceedings of 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE '99), 1999.

[8] H. Wache, et al, "Ontology-Based Integration of Information — A Survey of Existing Approaches," Proceedings of 17th International Joint Conference on Artificial Intelligence, Seattle, Washington, USA, 2001.

[9] M. Lenzerini, "Data Integration: A Theoretical Perspective," Proceedings of 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Madison, Wisconsin, 2002.

[10] N. F. Noy and M. A. Musen, "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment," Proceedings of 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence, 2000.

[11] S. Bechhofer, et al, "OWL Web Ontology Language Reference," W3C Recommendation, vol. 2005, M. Dean and G. Schreiber, Eds., February 10, 2004.

[12] S. Swasrukkiet, "A Methodology for Integration of Object-Oriented Data Models Using Heuristics and Analysis of Relationships between Class," Master Thesis, Department of Computer Engineering, Chulalongkorn University, Bangkok, 1999, pp. 162.

[13] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," Knowledge Acquisition (1042-8143), vol. 5, pp. 199 - 220, 1993.

[14] W. Sull and R.L. Kashyap, "A Self-Organizing Knowledge Representation Scheme for Extensible Heterogeneous Information Environment", IEEE Transactions on Knowledge and Data Engineering, vol. 4 no. 2, pp. 185-191, 1992.

[15] Y. Arens, et al, "Query Processing in the SIMS Information Mediator," in Readings in agents: Morgan Kaufmann Publishers Inc., pp. 82 - 90, 1998.

ประวัติผู้เขียนวิทยานิพนธ์

นางสาว มนัชยา ชมธวัช เกิดเมื่อวันที่ 26 มิถุนายน 2522 ที่จังหวัดตรัง สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2542 แล้วเข้าทำงานที่ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค) ในตำแหน่งวิศวกร 1 เป็นเวลา 3 ปี และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี 2546 เคยได้รับทุนอุดหนุนวิทยานิพนธ์สำหรับนิสิต ประจำภาคการศึกษาต้น ปีการศึกษา 2549 จากบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย