

แบบจำลองฐานข้อมูลเชิงแนวคิดโอโอในแอมและการแปลงรูปเป็นฐานข้อมูล
เชิงวัตถุ

THE OONIAM CONCEPTUAL SCHEMA MODEL AND A
TRANSFORMATION TO OBJECT ORIENTED DATABASE SCHEMA



สุธีรา พันธุ์ธีรานุรักษ์
SUTHEERA PUNTHEERANURAK

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2543

ISBN 974-622-978-8

เลขหมู่.....
เลขทะเบียน.....38530
ปี.....5 ๓.๑. 2544

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**THE OONIAM CONCEPTUAL SCHEMA MODEL AND A
TRANSFORMATION TO OBJECT ORIENTED DATABASE SCHEMA**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S OF TECHNOLOGY LADKRABANG**

2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ ISBN 974-622-978-8 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2000

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	แบบจำลองฐานข้อมูลเชิงแนวคิด โอโอโนแอมและการแปลงรูป เป็นฐานข้อมูลเชิงวัตถุ
นักศึกษา	น.ส.สุธีรา พันธุ์ธีรานุรักษ์
รหัสประจำตัว	36061211
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2543
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.สุกุมิตร จิตตะยโสธร

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เป็นการนำเสนอถึงวิธีการจัดการระบบสารสนเทศที่มีการเปลี่ยนแปลงสูง โดยได้มีการนำแบบจำลองระดับแนวคิดโนแอมที่ใช้ในการออกแบบฐานข้อมูลเชิงสัมพันธ์มาทำการปรับปรุงให้สามารถรองรับการข้อมูลที่มีความซับซ้อนมากขึ้น เพื่อให้สอดคล้องกับแนวการออกแบบระบบเชิงวัตถุรวมไปถึงการออกแบบฐานข้อมูลเชิงวัตถุด้วย แบบจำลองระดับแนวคิดที่ได้ทำการสร้างขึ้นใหม่นี้เรียกว่าแบบจำลองข้อมูลระดับแนวคิด OONIAM ซึ่งจะนำมาทำการแก้ปัญหาในส่วนงานของระบบสารสนเทศที่มักมีการเปลี่ยนแปลงอยู่ตลอดเวลาให้อยู่ในลักษณะเชิงวัตถุที่มีความสะดวกในการแก้ไขเปลี่ยนแปลงมากขึ้น ได้ทำการสร้างอัลกอริทึมในการแปลงจากแบบจำลองระดับแนวคิด OONIAM อีกทั้งยังได้ทำการสร้างระบบอัตโนมัติเพื่อออกแบบแบบจำลองข้อมูล OONIAM ซึ่งจะได้นำมาใช้งานในการแปลงฐานข้อมูลเดิมที่มีการใช้งานบนระบบจัดการฐานข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุอย่างต้องการ

Thesis Title	The OONAIM Conceptual Schema Model and A Transformation to Object Oriented Database Schema
Student	Miss. Sutheera Puntheeranurak
Student ID.	36061211
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2000
Thesis Advisor	Assoc.Prof.Dr. Suphamit Chittayasothorn

ABSTRACT

In this thesis we propose an approach for management of dynamic information system. This approach adopts the basic method of NIAM conceptual schema and adds new semantics for supporting complex data types, abstract data types and methods. The NIAM conceptual Schema is used for relation database design. This thesis supports an object oriented database design. we also present an Object Oriented Nijssen's Information Analysis Methodology (OONIAM) which is used to solve many problems in a dynamic information system. Because object oriented methodology updates system more conveniently than a structured approach does. We create an algorithm for transforming OONIAM to Object Definition Language and implementing Automatic System for designing OONIAM conceptual schema which is used for schema transformation from a relational database system management to an object oriented database system.

กิตติกรรมประกาศ

การจัดทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ฉันนั้น ทางผู้จัดทำได้รับการแนะนำและการสนับสนุนรวมทั้งกำลังใจจากบุคคลหลายฝ่าย ดังนั้นจึงใคร่กล่าวนามดังนี้

- ขอกราบขอบพระคุณ ของบิดา มารดา ของข้าพเจ้า ซึ่งได้ให้ความสนับสนุนในด้านการศึกษาของข้าพเจ้ามาโดยตลอด เป็นผู้ที่ยกย่องให้ทั้งคำสั่งสอนและกำลังใจกับข้าพเจ้าเสมอมา
- ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.ศุภมิตร จิตตะยโสธร อาจารย์ที่ปรึกษาในการจัดทำวิทยานิพนธ์ฉบับนี้ ท่านได้ให้ความรู้ในทุกด้านทั้งในด้านการทวิจยและการทำงานแก่ข้าพเจ้ามาโดยตลอด
- ขอขอบพระคุณ อาจารย์ทุกท่านที่ได้อบรมสั่งสอน ให้ความรู้แก่ข้าพเจ้าเสมอมา
- ขอขอบพระคุณ อาจารย์ประจำภาควิชาเทคนิคอุตสาหกรรมทุกท่าน ที่ได้ให้กำลังใจและคอยชี้แนะแนวทางในการทำวิจย
- ขอขอบพระคุณ อาจารย์ประจำภาควิชาคณิตศาสตร์ ม.เทคโนโลยีมหานครทุกท่าน ที่ได้ให้กำลังใจมาโดยตลอด
- ขอขอบคุณ บรรดาพี่น้องของข้าพเจ้า ที่คอยช่วยเหลือและให้กำลังใจเสมอมา
- ขอขอบคุณ รุ่นพี่ และเพื่อน ๆ ที่คอยให้กำลังใจและข้อเสนอแนะตลอดมา ซึ่งมีอาจกล่าวนามมา ณ. ที่นี้ได้หมด
- ขอขอบคุณ พี่ ๆ เจ้าหน้าที่หน่วยงานต่าง ๆ ในคณะวิศวกรรมศาสตร์ พระจอมเกล้าลาดกระบังที่คอยให้กำลังใจข้าพเจ้าเสมอมา

ข้าพเจ้าขอขอบความดีทั้งหมดที่ได้รับจากวิทยานิพนธ์ฉบับนี้ให้กับ มารดาผู้ล่วงลับไปแล้วของข้าพเจ้า ซึ่งเป็นบุคคลที่รักยิ่งของข้าพเจ้าตลอดไป

สุธีรา พันธุ์ธีรานุกษ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	2
1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย	3
1.4 ขอบเขตของการศึกษา	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีพื้นฐานที่เกี่ยวข้อง	5
2.1 แบบจำลองระดับแนวคิดในแอม (NIAM Conceptual Schema)	5
2.1.1 ส่วนประกอบพื้นฐานของแบบจำลองระดับแนวคิดในแอม	5
2.1.2 สัญลักษณ์พื้นฐานที่ใช้ในแบบจำลองระดับแนวคิดในแอม	6
2.1.3 กฎข้อบังคับความถูกต้องของข้อมูลที่ใช้ในแบบจำลองระดับแนวคิด ในแอม	7
2.1.4 อัลกอริทึมในการแปลงแบบจำลองระดับแนวคิดในแอมให้อยู่ในรูป ของโครงสร้างฐานข้อมูลเชิงสัมพันธ์	20
2.2 แบบจำลองข้อมูลเชิงวัตถุ (Object Oriented Database Model)	21
2.2.1 ส่วนประกอบพื้นฐานและสัญลักษณ์ของแบบจำลองฐานข้อมูลเชิงวัตถุ	22
2.2.2 อัลกอริทึมที่ใช้ในการแปลงแบบจำลองฐานข้อมูลเชิงวัตถุให้อยู่ในรูป ของโครงสร้างฐานข้อมูลเชิงวัตถุ	30
2.3 การเปรียบเทียบแบบจำลองฐานข้อมูลเชิงวัตถุกับแบบจำลองฐานข้อมูลเชิงสัมพันธ์	31
2.3.1 ชนิดของข้อมูล (Data Types)	31
2.3.2 ความแน่นอนของข้อมูล (Data Integrity)	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.3 วิวัฒนาการของโครงสร้าง (Schema Evolution)	32
---	----

บทที่ 3 OONIAM Conceptual Schema	34
--	----

3.1 ส่วนประกอบและสัญลักษณ์ที่ใช้ใน OONIAM Conceptual Schema	34
---	----

3.1.1 ส่วนประกอบและสัญลักษณ์ที่นำมาจากแบบจำลองระดับแนวคิด ในแอม	34
--	----

3.1.2 ส่วนประกอบและสัญลักษณ์เพิ่มเติมที่ใช้ในแบบจำลองระดับแนวคิด OONIAM	35
--	----

3.2 ตัวอย่างการใช้สัญลักษณ์ต่าง ๆ ในแบบจำลองระดับแนวคิด OONIAM	37
--	----

3.3 อัลกอริทึมในการแปลงแบบจำลองระดับแนวคิด OONIAM ให้อยู่ในรูปของ Object Oriented Definitions Language	39
---	----

3.4 อัลกอริทึมของเมธอดที่ทำการแปลงมาจากกฎข้อบังคับความถูกต้องของข้อมูล	42
--	----

3.5 ตัวอย่างการใช้แบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM ในการออกแบบฐาน ข้อมูลเชิงวัตถุ	44
--	----

บทที่ 4 สถาปัตยกรรมของระบบ	50
----------------------------------	----

4.1 สถาปัตยกรรมของระบบ	50
------------------------------	----

4.2 เครื่องมือที่ใช้ในการพัฒนาระบบ	51
--	----

4.2.1 Visual Basic	51
--------------------------	----

4.2.2 Cache OODBMS	53
--------------------------	----

4.3 อุปกรณ์ที่ใช้ในการพัฒนาระบบ	58
---------------------------------------	----

บทที่ 5 การพัฒนาระบบอัตโนมัติสำหรับการออกแบบ OONIAM Conceptual Schema	59
--	----

5.1 Use case diagram	59
----------------------------	----

5.2 MetaModel	61
---------------------	----

5.3 Sequence diagram และ Collaboration diagram.....	65
---	----

5.4 ระบบอัตโนมัติสำหรับการออกแบบ OONIAM Conceptual Schema	67
---	----

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 6 การเปรียบเทียบวิธีการออกแบบฐานข้อมูลเชิงวัตถุ	70
6.1 การออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ Rational Rose	70
6.2 การออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ Visio Professional	74
6.3 การออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ OONIAM	79
6.4 เปรียบเทียบวิธีการออกแบบฐานข้อมูลเชิงวัตถุ	84
บทที่ 7 สรุปผลการวิจัยและข้อเสนอแนะ	85
7.1 สรุปผลการวิจัย	85
7.2 ข้อเสนอแนะ	86
เอกสารอ้างอิง	87
ประวัติผู้เขียน	89

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางแสดงสัญลักษณ์ที่ใช้แทนความหมายของ Multiplicity of Associations	25



สารบัญรูป

รูปที่	หน้า
2.1 แสดงสัญลักษณ์ของชนิดเอนตีตี้ A และตัวอย่างของชนิดเอนตีตี้คน	6
2.2 แสดงสัญลักษณ์ของชนิดเลเบล B และตัวอย่างของชนิดเลเบลชื่อคน	6
2.3 แสดงสัญลักษณ์ชนิดความจริงแบบ 2 บทบาท และ 3 บทบาท	6
2.4 แสดงสัญลักษณ์ของชนิดความจริงแบบเนส	6
2.5 แสดงสัญลักษณ์ของชนิดอ้างอิงที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วย	7
2.6 แสดงสัญลักษณ์ของชนิดอ้างอิงที่มีความสัมพันธ์หนึ่งต่อหนึ่งกับชนิดเลเบลแบบตัวเลข	7
2.7 แสดงความสัมพันธ์แบบหนึ่งต่อหลายหน่วย	7
2.8 แสดงความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วย	8
2.9 แสดงความสัมพันธ์แบบหลายต่อหลายหน่วย	8
2.10 ภาพแสดง Inter fact type uniqueness constraints	9
2.11 ภาพแสดง Mandatory role constraints	9
2.12 ภาพแสดง Inclusion mandatory role constraints	10
2.13 ภาพแสดง Entity type constraints	11
2.14 ภาพแสดง Subset constraints	12
2.15 ภาพแสดง Equality constraints	13
2.16 ภาพแสดง Exclusion constraints	13
2.17 ภาพแสดง Subtype constraints	14
2.18 ภาพแสดง Occurrence frequency constraints	15
2.19 ภาพแสดง Irreflexive constraints	16
2.20 ภาพแสดง Asymmetric constraints	16
2.21 ภาพแสดง Antisymmetric constraints	17
2.22 ภาพแสดง Intransitive constraints	18
2.23 ภาพแสดง Acyclic Ring constraints	18
2.24 ภาพแสดง Cardinality constraints	19
2.25 ภาพแสดง Relative closure constraints	19
2.26 ภาพแสดง Textual constraints	20
2.27 ภาพแสดงสัญลักษณ์และตัวอย่างของคลาส	23
2.28 ภาพแสดงสัญลักษณ์ความสัมพันธ์แบบแอสโซซิเอชัน	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.29 ภาพแสดงชื่อของเอสโซซิเอชัน	24
2.30 ภาพแสดงสัญลักษณ์ของช็อบทบาท	24
2.31 ภาพแสดงตัวอย่างของ Multiplicity of Associations	25
2.32 ภาพแสดงคลาสเอสโซซิเอชัน	26
2.33 ภาพแสดงตัวอย่างของแอสโซซิเอชัน	26
2.34 ภาพแสดงตัวอย่างความสัมพันธ์แบบ N-ary Associations	27
2.35 ภาพแสดงความสัมพันธ์แบบแอสโซซิเอชัน	27
2.36 ภาพแสดงความสัมพันธ์แบบแอสโซซิเอชันในลักษณะของการเป็นส่วนประกอบ	28
2.37 ภาพแสดงความสัมพันธ์แบบเจนเนอรัลไลเซชัน	28
2.38 ภาพแสดงกฎข้อบังคับความถูกต้องของข้อมูลแบบ Ordered	29
2.39 ภาพแสดงกฎข้อบังคับความถูกต้องของข้อมูลแบบ subset	29
2.40 ภาพแสดงกฎข้อบังคับความถูกต้องของข้อมูลแบบ exclusive or	29
2.41 ภาพแสดงกฎข้อบังคับความถูกต้องแบบ reflexive association	30
3.1 ภาพแสดงสัญลักษณ์พื้นฐานที่นำมาจากแบบจำลองระดับแนวคิดในแอม	35
3.2 แสดงสัญลักษณ์ชนิดคอมเพลกซ์ออบเจกต์	36
3.3 ภาพแสดงสัญลักษณ์ของเมธอด	36
3.4 แสดงสัญลักษณ์เมธอดที่ใช้ในแบบจำลอง	37
3.5 แสดงตัวอย่างการใช้สัญลักษณ์ในแบบจำลองระดับแนวคิด OONIAM	38
3.6 ภาพตัวอย่างแสดงการใช้เมธอดกับชนิดเอนทิตี	39
3.7 ตัวอย่างแบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM	44
3.8 ตัวอย่างการสร้างเมธอดใน OONIAM	45
3.9 แสดงผลลัพธ์ที่ได้จากการแปลงให้อยู่ในรูปของคลาสไดอะแกรม	45
3.10 ตัวอย่างแบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM	46
4.1 สถาปัตยกรรมของระบบ OONIAM	50
4.2 ภาพแสดงสถาปัตยกรรมในการติดต่อของ Cache	55
4.3 ภาพแสดงโครงสร้างและการใช้งานของ Cache Objects	55
5.1 ภาพแสดง Use Case Diagram ของระบบ	60
5.2 เมตาโมเดลที่ใช้ในการจัดเก็บแบบจำลองระดับแนวคิด OONIAM	62

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้ในวงจำกัดเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.3 เมตาโมเดลที่ใช้ในการจัดเก็บผลลัพธ์ที่ได้จากการแปลง	64
5.4 ภาพแสดง Sequence diagram ของระบบ	65
5.5 ภาพแสดง Collaboration diagram ของระบบ	66
5.6 ภาพแสดงตัวอย่างการสร้าง OONIAM Conceptual Schema	67
5.7 ภาพแสดงผลลัพธ์ที่อยู่ในรูปของ Class Diagram	68
5.8 ภาพแสดงผลลัพธ์ที่ได้อยู่ในรูปของ Object Definition Language	68
6.1 ภาพแสดงการออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ Rational Rose	70
6.2 ภาพแสดงตัวอย่างในการกำหนดรายละเอียดของชื่อคลาส	71
6.3 ภาพแสดงตัวอย่างในการกำหนดรายละเอียดต่าง ๆ ของคลาส	71
6.4 ภาพแสดงตัวอย่างในการกำหนดแอตทริบิวต์ของคลาส	72
6.5 ภาพแสดงตัวอย่างการกำหนดรายละเอียดของแอตทริบิวต์	72
6.6 ภาพแสดงตัวอย่างของคลาสไคอะแกรม	73
6.7 ภาพแสดงตัวอย่างการกำหนดรายละเอียดของความสัมพันธ์	73
6.8 ภาพแสดงการเลือก Template ที่ใช้ในการสร้างแบบจำลอง	74
6.9 ภาพแสดงรายละเอียดของกลุ่มของเครื่องมือ	75
6.10 ภาพแสดงหน้าจอสำหรับการทำกรออกแบบแบบจำลอง	75
6.11 ภาพแสดงตัวอย่างการกำหนดรายละเอียดของคลาส	76
6.12 ภาพแสดงการกำหนดแอตทริบิวต์ของคลาส	76
6.13 ภาพแสดงตัวอย่างในการกำหนดรายละเอียดของแอตทริบิวต์	77
6.14 ภาพแสดงตัวอย่างในการกำหนดเมธอดของคลาส	77
6.15 ภาพแสดงการกำหนดรายละเอียดของแต่ละเมธอด	78
6.16 ภาพแสดงหน้าจอสำหรับกำหนดรายละเอียดของความสัมพันธ์	78
6.17 ภาพแสดงผลลัพธ์ที่ได้จากการออกแบบ	79
6.18 ภาพแสดงการติดต่อกับฐานข้อมูลเชิงวัตถุ Cache	80
6.19 ภาพแสดงการป้อนข้อมูลโครงการที่จะทำการออกแบบ	80
6.20 ภาพแสดงหน้าจอของระบบ	81
6.21 ภาพแสดงตัวอย่างการป้อนข้อมูลของชนิดเลเบิล	81
6.22 ภาพแสดงตัวอย่างการป้อนข้อมูลของชนิดความจริงแบบเนส	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในระบบสารสนเทศที่ได้พัฒนาขึ้นมานั้นสามารถแบ่งองค์ประกอบออกเป็น 2 ส่วน คือ ส่วนของขั้นตอนการทำงาน (procedure) ซึ่งจะเป็นส่วนที่ทำการรวบรวมขั้นตอนการคำนวณต่าง ๆ และรวมไปถึงการควบคุมกฎข้อบังคับความถูกต้องของข้อมูลที่จะทำการจัดเก็บ อีกส่วนหนึ่งคือ ส่วนของข้อมูล (Data) ซึ่งแสดงถึงโครงสร้างของข้อมูลและข้อมูลที่จัดเก็บจริง ซึ่งในระบบงานสารสนเทศที่มีการเปลี่ยนแปลงสูงนั้นมักจะมีการเปลี่ยนแปลงใน 2 ส่วนนี้อยู่เสมอ เช่นอาจมีการเปลี่ยนแปลงในส่วนของขั้นตอนการทำงานทำให้มีผลกระทบกับโครงสร้างของข้อมูลที่เคยออกแบบไว้ จึงทำให้ต้องทำการปรับปรุงทั้งโครงสร้างข้อมูลและขั้นตอนการทำงานของระบบสารสนเทศดังกล่าวให้ถูกต้องเพื่อให้ใช้งานได้จริงกับเหตุการณ์ปัจจุบันอยู่ตลอดเวลา โดยส่วนใหญ่ระบบงานสารสนเทศที่ได้พัฒนาขึ้นมานั้นมักจะทำการออกแบบฐานข้อมูล ในรูปของฐานข้อมูลเชิงสัมพันธ์ และทำการจัดเก็บข้อมูลบนระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS) ซึ่งใช้งานกันอย่างแพร่หลายมาเป็นเวลานาน

แต่ในปัจจุบันระบบสารสนเทศที่ได้ใช้งานกันอยู่นั้น มักพบปัญหาเกี่ยวกับการจัดการและดูแลกฎข้อบังคับความถูกต้องของฐานข้อมูลที่มีโอกาสเกิดการเปลี่ยนแปลงได้ เมื่อระบบเริ่มใช้งานมาระยะเวลาหนึ่ง ซึ่งในการจัดการเกี่ยวกับกฎข้อบังคับความถูกต้องของข้อมูล ผู้พัฒนาระบบอาจทำการจัดการได้สองลักษณะคือ ลักษณะแรกจะทำการจัดเก็บกฎข้อบังคับความถูกต้องของข้อมูลรวมไว้ในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ในรูปแบบของ Stored procedure [3] หรือ Database Trigger [3] ของฐานข้อมูลนั้น ๆ แต่ลักษณะของกฎข้อบังคับความถูกต้องของข้อมูลบางประเภทไม่สามารถสร้างและจัดเก็บในลักษณะดังกล่าวได้จึงทำให้ผู้พัฒนาระบบต้องทำการสร้างกฎข้อบังคับความถูกต้องของข้อมูลไว้ใน source code ของโปรแกรมประยุกต์ที่สร้างขึ้น หรือในอีกลักษณะหนึ่ง ก็อาจทำการจัดเก็บกฎข้อบังคับความถูกต้องของข้อมูลทั้งหมดไว้ในรูปของ source code ของโปรแกรมประยุกต์ที่สร้างขึ้น ซึ่งจะทำให้ source code นั้น ๆ มีขนาดใหญ่มากและแยกออกเป็นหลาย ๆ ไฟล์ การที่จะทราบถึงส่วนที่ต้องทำการแก้ไขว่าอยู่ในไฟล์ไหนและส่วนใดของโปรแกรมนั้น ต้องใช้เวลาค่อนข้างมากและต้องมีความคุ้นเคยกับ source code ของโปรแกรمدังกล่าวด้วย ซึ่งพบว่าการจัดการทั้งสองลักษณะดังกล่าวนี้ จะทำให้เกิดความยุ่งยากในการทำการแก้ไขกฎข้อบังคับความถูกต้องของข้อมูลดังกล่าวให้ถูกต้องตามความต้องการอยู่เสมอ

และถึงแม้ว่าระบบดังกล่าวจะได้ถูกพัฒนาขึ้นมาและสามารถใช้งานได้อย่างถูกต้องใน

ระยะเวลาหนึ่งแล้วก็ตาม ก็ยังอาจจะเกิดปัญหาในการบำรุงรักษาระบบให้สามารถใช้งานได้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตลอดไป เพราะเมื่อเวลาผ่านไปอาจมีการเปลี่ยนแปลงกฎข้อบังคับความต้องการของข้อมูลและกฎต่างๆ ในการดำเนินงาน ซึ่งส่งผลกระทบต่อให้ต้องมีการแก้ไขโปรแกรมระบบสารสนเทศเพื่อให้สามารถรองรับกับกฎใหม่ๆ และยกเลิกกฎที่มีอยู่เดิมบางกฎลงไปได้ ซึ่งถ้าหากผู้พัฒนาโปรแกรมยังคงเป็นกลุ่มบุคคลเดิมที่มีความรู้ความเข้าใจในระบบเป็นอย่างดี โดยที่ได้มีการทำเอกสารอ้างอิงและได้บันทึกคำแนะนำไว้ใน source code ของโปรแกรมไว้อย่างครบถ้วน ก็จะทำให้การแก้ไขกฎระเบียบต่างๆ นั้นสามารถทำได้โดยไม่เป็นเรื่องที่ยากมากนัก แต่โดยมากแล้วการพัฒนาโปรแกรมที่มีอายุการใช้งานนานๆ การดูแลรักษาโปรแกรมมักจะถูกส่งมอบเป็นทอดๆ ไป เมื่อผู้พัฒนาโปรแกรมเดิมไม่อยู่ทำให้การแก้ไขโปรแกรมโดยผู้ที่ได้รับสืบทอดมานั้นเป็นเรื่องที่ยากมาก ถึงแม้จะมีองค์ประกอบต่างๆ เตรียมไว้ให้ครบถ้วนก็ตาม เพราะผู้พัฒนาโปรแกรมใหม่จะต้องใช้เวลาในการทำความเข้าใจกับระบบงานเดิมมานานมาก ซึ่งเป็นที่ทราบกันดีว่าระยะเวลาที่ใช้ในการบำรุงรักษาระบบให้สามารถดำเนินงานไปได้อย่างถูกต้องนั้น ใช้เวลามากกว่า 70 เปอร์เซ็นต์ของเวลาทั้งหมด

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

ในปัจจุบันการพัฒนาระบบสารสนเทศนั้น ได้นำเอาการออกแบบระบบในลักษณะเชิงวัตถุเข้ามาใช้งานกันอย่างแพร่หลาย เพราะในการออกแบบลักษณะดังกล่าวทำให้ผู้พัฒนาระบบสามารถที่จะได้รับการสืบทอดความเข้าใจของขั้นตอนการทำงานของระบบได้ง่ายมากขึ้น และสามารถเปลี่ยนแปลงและแก้ไขขั้นตอนการทำงานต่าง ๆ ได้สะดวกมากขึ้น (reusability) ทั้งนี้ทำให้ระบบจัดการฐานข้อมูลเชิงวัตถุได้เข้ามามีบทบาทมากขึ้นในการจัดการฐานข้อมูลทำให้การพัฒนากระบวนการสารสนเทศสอดคล้องกับการออกแบบระบบ ในลักษณะเชิงวัตถุ (Object Oriented System) อีกทั้งระบบจัดการฐานข้อมูลเชิงวัตถุยังสามารถที่จะรองรับข้อมูลได้หลากหลายประเภท รวมไปถึงข้อมูลที่มีความซับซ้อนมาก (Complex Data Type) และลักษณะของ Abstract Data Type [2, 3, 4] ทั้งนี้ยังรวมข้อมูลประเภทภาพและเสียงได้อีกด้วย อีกทั้งในระบบจัดการฐานข้อมูลเชิงวัตถุยังสามารถจัดการเกี่ยวกับกฎข้อบังคับความต้องการของข้อมูลได้สะดวกมากยิ่งขึ้น ทำให้เมื่อเกิดการเปลี่ยนแปลงกฎข้อบังคับความต้องการต่าง ๆ สามารถทำการแก้ไขได้อย่างง่ายดายมากขึ้น

ดังนั้นในวิทยานิพนธ์ฉบับนี้จึงมีวัตถุประสงค์ในการจัดหาวิธีการที่จะบำรุงรักษาระบบสารสนเทศที่มีการเปลี่ยนแปลงสูงให้สามารถใช้งานได้อย่างถูกต้องตลอดเวลา รวมไปถึงแนวทางในการจัดการเกี่ยวกับกฎข้อบังคับความต้องการของข้อมูล อีกทั้งหาแนวทางในการปรับปรุงให้ระบบงานสารสนเทศที่ได้มีการใช้งานอยู่ในปัจจุบัน สามารถที่จะปรับเปลี่ยนให้รองรับกับแนวทางการเปลี่ยนแปลงในอนาคต เพื่อให้สอดคล้องกับทรัพยากรและเทคโนโลยีที่มีอยู่ในปัจจุบัน

เช่นแนวทางในการปรับโครงสร้างฐานข้อมูลให้อยู่ในลักษณะของฐานข้อมูลเชิงวัตถุ เพื่อรองรับกับการขยายการจัดเก็บข้อมูลประเภทภาพและเสียงเพิ่มเติมเข้าไปในระบบสารสนเทศเดิมที่มีอยู่ได้ ไม่ว่าจะเป็นกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ไม่ต้องทำการออกแบบโครงสร้างของฐานข้อมูลใหม่ ซึ่งเป็นการสิ้นเปลืองทรัพยากรที่มีอยู่อย่างมาก อีกทั้งยังใช้เวลาในการออกแบบระบบงานใหม่ใช้เวลาค่อนข้างมากอีกด้วย

1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

หลักการที่ได้มีการนำเสนอในวิทยานิพนธ์ฉบับนี้คือ หาแนวทางในการแปลงฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุ เพื่อสามารถนำข้อมูลของระบบงานที่ได้มีการจัดเก็บข้อมูลบนระบบจัดการฐานข้อมูลเชิงสัมพันธ์ มาใช้งานบนระบบจัดการฐานข้อมูลเชิงวัตถุได้ เป็นการเพิ่มความสะดวกต่อผู้พัฒนาระบบในการที่จะปรับปรุงระบบงานสารสนเทศเพื่อให้ระบบสามารถรองรับกับการเปลี่ยนแปลงที่อาจจะเกิดขึ้น โดยนำการออกแบบฐานข้อมูลเชิงสัมพันธ์ด้วยวิธีการของ NIAM Conceptual Schema [1] ซึ่งเป็นวิธีที่ใช้งานกันอย่างแพร่หลายมาทำการแปลงให้อยู่ในรูปของแบบจำลองเชิงวัตถุ (Object Oriented Model) โดยอาศัยรูปแบบในการนำเสนอแบบจำลองเชิงวัตถุ ตามมาตรฐานของ UML (Unified Modeling Language) [6, 7, 8, 9] ซึ่งใช้งานกันอย่างแพร่หลาย สำหรับการออกแบบระบบงานเชิงวัตถุ โดยเลือกรูปแบบที่จะนำเสนอในลักษณะของ Class Diagram [6, 7, 8, 9, 10] หรือ Object Definition Language [2, 4, 5] เพื่อผู้ใช้งานสามารถนำมาใช้กับระบบจัดการฐานข้อมูลเชิงวัตถุต่อไป

รวมไปถึงแนวทางในการที่จะนำเสนอ ให้ผู้พัฒนาระบบที่คุ้นเคยกับการออกแบบฐานข้อมูลเชิงสัมพันธ์ สามารถใช้ NIAM Conceptual Schema ในการออกแบบฐานข้อมูลเชิงวัตถุได้อีกด้วย โดยจะจัดหาวิธีการนำเสนอคุณสมบัติของฐานข้อมูลเชิงวัตถุให้ปรากฏอยู่บน NIAM Conceptual Schema และจะจัดทำระบบอัตโนมัติขึ้นเพื่อให้ผู้พัฒนาระบบสามารถใช้งานได้อย่างสะดวกมากขึ้นทั้งในการแปลงฐานข้อมูลที่มีอยู่ และการออกแบบฐานข้อมูลเชิงวัตถุด้วย

1.4 ขอบเขตการวิจัย

ในงานวิจัยนี้ได้กำหนดขอบเขตในการทำงานโดยสามารถแบ่งออกเป็นข้อ ๆ ได้ดังนี้

1. ศึกษาการออกแบบระบบงานสารสนเทศในลักษณะของการออกแบบเชิงวัตถุ
2. ทำการสร้างอัลกอริทึมในการแปลงฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุ
3. ทำการศึกษาเพื่อสร้างแบบจำลองฐานข้อมูลเชิงแนวคิดเพื่อนำมาใช้ในการออกแบบฐานข้อมูลเชิงวัตถุ โดยอาศัยความรู้พื้นฐานจากการออกแบบฐานข้อมูลเชิงสัมพันธ์ด้วยวิธีการของ NIAM Conceptual Schema
4. ทำการสร้างอัลกอริทึมในการแปลงจากแบบจำลองที่ได้สร้างขึ้นให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ศึกษาในเรื่องของกฎข้อบังคับความถูกต้องของข้อมูลที่มีอยู่ เพื่อทำการแปลงให้อยู่ในรูปที่สามารถใช้งานได้กับฐานข้อมูลเชิงวัตถุ
6. ทำการสร้างอัลกอริทึมในการแปลงกฎข้อบังคับความถูกต้องของข้อมูลต่าง ๆ
7. จัดทำระบบอัตโนมัติเพื่อสนับสนุนการใช้งานในการแปลงฐานข้อมูลดังกล่าว

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้พัฒนาระบบสามารถที่จะแปลงฐานข้อมูลที่อยู่ในรูปของฐานข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุได้ เพื่อนำไปใช้กับระบบจัดการฐานข้อมูลเชิงวัตถุต่อไป
2. ทำให้การออกแบบและพัฒนาระบบงานสารสนเทศสามารถทำได้อย่างต่อเนื่อง ไม่ต้องทำการออกแบบระบบใหม่เมื่อมีการเปลี่ยนแปลงเกิดขึ้น
3. เพิ่มความสะดวกในการแก้ไขกฎข้อบังคับความถูกต้องของข้อมูล รวมไปถึงการเปลี่ยนแปลงขั้นตอนการทำงานต่าง ๆ ของระบบได้สะดวกมากขึ้น โดยอาศัยหลักการออกแบบเชิงวัตถุ
4. ทำให้ผู้ที่เข้าใจและคุ้นเคยกับวิธีการออกแบบฐานข้อมูลเชิงสัมพันธ์สามารถทำการออกแบบฐานข้อมูลเชิงวัตถุได้สะดวกมากยิ่งขึ้น ไม่ต้องทำการศึกษาก่อนการออกแบบฐานข้อมูลเชิงวัตถุใหม่ทั้งหมด
5. เพื่อรองรับกับวิธีการออกแบบระบบงานสารสนเทศในแนวทางการออกแบบเชิงวัตถุที่ได้มีบทบาทเพิ่มมากขึ้นในปัจจุบัน

บทที่ 2

ทฤษฎีพื้นฐานที่เกี่ยวข้อง

2.1 แบบจำลองระดับแนวคิดในแอม (NIAM Conceptual Schema)

แบบจำลองระดับแนวคิดในแอมนั้น คำว่า NIAM ย่อมาจาก Nijssen's Information Analysis Methodology 1 เป็นรูปแบบหนึ่งของแบบจำลองระดับแนวคิดที่ได้มีการคิดค้นมาตั้งแต่ราวปลายปี ค.ศ. 1977 โดย ศาสตราจารย์ จี เอ็ม ไนเซน ซึ่งเป็นแบบจำลองที่มีพื้นฐานมาจากโครงสร้างภาษารวมชาติ ที่ใช้รูปประโยคมี ประธาน กริยา และกรรม เป็นแบบจำลองระดับแนวคิดที่มีความหมายและเครื่องหมายแสดงความสัมพันธ์ของข้อมูล และข้อจำกัดของข้อมูลได้อย่างชัดเจน นอกจากนี้แล้วยังสามารถแปลงแบบจำลองดังกล่าวเป็นโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ (Relational Database Schema) ซึ่งจะได้ผลลัพธ์ที่อยู่ในรูปของ Fifth Normal Form จึงมีการนำแบบจำลองระดับแนวคิดในแอม มาใช้ช่วยในการออกแบบฐานข้อมูลเชิงสัมพันธ์กันอย่างแพร่หลาย เพราะมีการใช้สัญลักษณ์ที่แสดงความสัมพันธ์ของข้อมูลอย่างชัดเจนและง่ายต่อการเข้าใจ

2.1.1 ส่วนประกอบพื้นฐานของแบบจำลองระดับแนวคิดในแอม

จะประกอบไปด้วย

1. ชนิดเอนติตี้ (Entity Type) หมายถึง เซตของสิ่งที่สนใจทั้งที่อยู่ในรูปของนามธรรม หรือรูปธรรม ซึ่งอาจเป็นสิ่งที่จับต้องได้หรือไม่ได้ เช่น คน, ภาควิชา, บริษัท, รถยนต์ เป็นต้น
2. ชนิดเลเบล (Label Type, Value Type) หมายถึง เซตของสิ่งที่ใช้บ่งบอกความแตกต่างหรือชื่อของแต่ละเอนติตี้ที่กำหนด เช่น ชื่อ, นามสกุล, รหัสประจำตัว, ทะเบียนรถ เป็นต้น
3. บทบาท (Role) หมายถึง ความสัมพันธ์ที่เกี่ยวข้องกับชนิดเอนติตี้ที่สัมผัสอยู่ เช่น ชนิดเอนติตตินักศึกษา แสดงบทบาท เป็นผู้ลงทะเบียนเรียนในวิชานั้น ๆ เป็นต้น
4. ประโยคความจริงมูลฐาน (Elementary Fact Type) หรืออาจเรียกว่าชนิดความจริง (Fact Type) หมายถึง เซตของความสัมพันธ์ระหว่างสมาชิกของชนิดเอนติตี้ตั้งแต่ 2 เอนติตี้ขึ้นไป โดยขนาดของชนิดความจริงจะขึ้นอยู่กับจำนวนบทบาทที่เกี่ยวข้อง โดยที่ชนิดความจริงที่มีจำนวน 2 บทบาท จะเรียกว่า binary fact type ส่วนชนิดความจริงที่มี 3 บทบาท จะเรียกว่า

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ขออนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ternary fact type สำหรับชนิดความจริงที่มีมากกว่า 3 บทบาทขึ้นไปจะรวมเรียกว่า n-ary fact type
- 5. ชนิดอ้างอิง (Reference Type) หมายถึง เซตของความสัมพันธ์ระหว่างสมาชิกของชนิดเอนทิตีกับสมาชิกของชนิดเลเบลที่มีอยู่
 - 6. ชนิดความจริงแบบเนส (Nested Fact Type) หมายถึง ชนิดเอนทิตีที่แสดงความสัมพันธ์ในการกำหนดกลุ่มของชนิดความจริงที่มีตั้งแต่ 2 บทบาทขึ้นไป
 - 7. กฎข้อบังคับความถูกต้องของข้อมูล (Integrity Constraints) หมายถึง สิ่งที่ใช้แสดงกฎที่ใช้กับควบคุมความถูกต้องของข้อมูล

2.1.2 สัญลักษณ์พื้นฐานที่ใช้ในแบบจำลองระดับแนวคิดในแอม



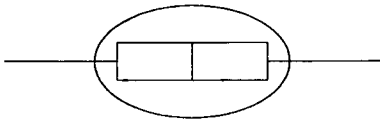
รูปที่ 2.1 แสดงสัญลักษณ์ของชนิดเอนทิตี A และตัวอย่างของชนิดเอนทิตีคน



รูปที่ 2.2 แสดงสัญลักษณ์ของชนิดเลเบล B และตัวอย่างของชนิดเลเบลชื่อคน

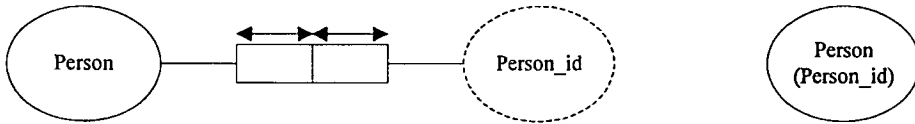


รูปที่ 2.3 แสดงสัญลักษณ์ชนิดความจริงแบบ 2 บทบาท และ 3 บทบาท

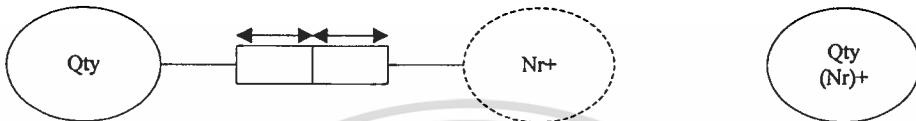


รูปที่ 2.4 แสดงสัญลักษณ์ของชนิดความจริงแบบเนส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงสัญลักษณ์ของชนิดอ้างอิงที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วย



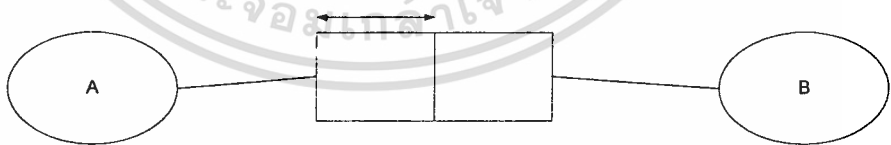
รูปที่ 2.6 แสดงสัญลักษณ์ของชนิดอ้างอิงที่มีความสัมพันธ์หนึ่งต่อหนึ่งกับชนิดเลเบลแบบตัวเลข

2.1.3 กฎข้อบังคับความถูกต้องของข้อมูลที่ใช้ในแบบจำลองระดับแนวคิดในแอม

1. Intra fact type uniqueness constraints (internal uniqueness constraints)

เป็นกฎข้อบังคับความถูกต้อง เพื่อทำการกำหนดบทบาทที่ใช้แสดงเอกลักษณ์ของความสัมพัทธ์ใด ๆ โดยจะกำหนดความสัมพันธ์ระหว่างสมาชิกของชนิดเอนติตี้หนึ่งกับสมาชิกของชนิดเอนติตี้อื่น หรือกับสมาชิกของชนิดเลเบล โดยสามารถกำหนดออกเป็นรูปแบบต่าง ๆ ได้ดังต่อไปนี้

- การกำหนดความสัมพันธ์แบบหนึ่งต่อหลายหน่วย (one to many relationship) ซึ่งสามารถแสดงบนแผนภาพได้ดังรูป

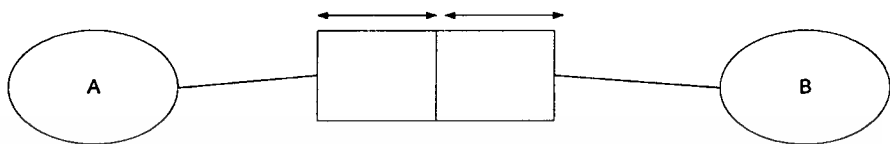


รูปที่ 2.7 แสดงความสัมพันธ์แบบหนึ่งต่อหลายหน่วย

ลักษณะเช่นนี้สามารถแสดงได้ว่า ชนิดเอนติตี้ A จะแสดงความสัมพันธ์กับ ชนิดเอนติตี้ หรือ ชนิดเลเบล B ได้อย่างมากที่สุดเพียงหนึ่งความสัมพันธ์เท่านั้น แต่ในทางกลับกัน ชนิดเอนติตี้ หรือ ชนิดเลเบล B จะแสดงความสัมพันธ์กับ ชนิดเอนติตี้ A ได้หลายความสัมพันธ์ โดยกฎข้อบังคับความถูกต้องจะต้องทำการควบคุมไม่ให้เกิดการซ้ำซ้อนของข้อมูลในคอลัมน์ A ขึ้นได้ เช่น คนหนึ่งคนจะมี

มารดาได้เพียงคนเดียวเท่านั้น แต่ในทางกลับกัน คนเพียงคนเดียวอาจเป็นมารดาของหลายคนได้

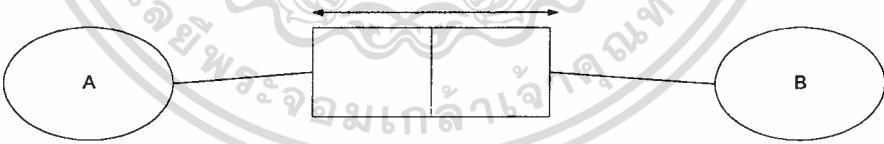
- การกำหนดความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วย (one to one relationship) ซึ่งสามารถแสดงบนแผนภาพได้ดังรูป



รูปที่ 2.8 แสดงความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วย

ลักษณะเช่นนี้สามารถแสดงได้ว่า ชนิดเอนิตี A จะแสดงความสัมพันธ์กับ ชนิดเอนิตี หรือ ชนิดเลเบล B ได้เพียงหนึ่งความสัมพันธ์เท่านั้น โดยกฎข้อบังคับจะทำการควบคุมไม่ให้เกิดความสัมพันธ์ของข้อมูลมากกว่าหนึ่งความสัมพันธ์ เช่น คนหนึ่งคนจะมีเลขประจำตัวได้เพียงหมายเลขเดียวเท่านั้น และในทางกลับกัน หมายเลขประจำตัวหนึ่งหมายเลขจะต้องหมายถึงคนเพียงคนเดียว

- การกำหนดความสัมพันธ์แบบหลายต่อหลายหน่วย (many to many relationship) ซึ่งสามารถแสดงบนแผนภาพได้ดังรูป



รูปที่ 2.9 แสดงความสัมพันธ์แบบหลายต่อหลายหน่วย

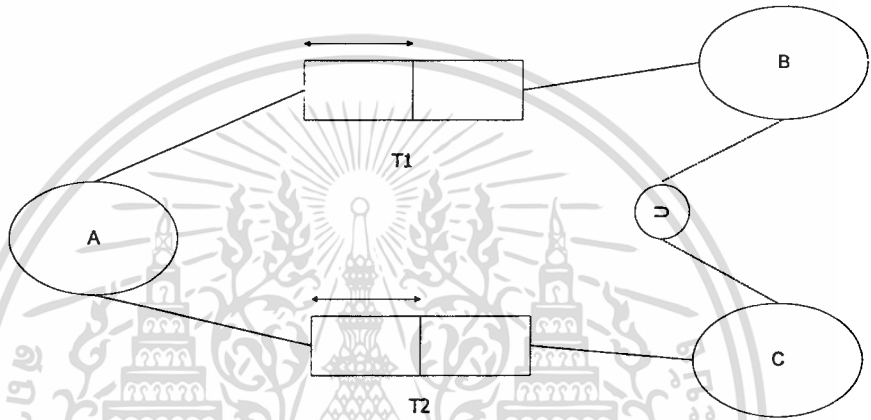
ลักษณะเช่นนี้สามารถแสดงได้ว่า ชนิดเอนิตี A จะแสดงความสัมพันธ์กับ ชนิดเอนิตี B ได้หลายความสัมพันธ์ และในทางกลับกัน ชนิดเอนิตี B ก็จะแสดงความสัมพันธ์กับ ชนิดเอนิตี A ได้หลายความสัมพันธ์เช่นกัน โดยกฎข้อบังคับความถูกต้องจะต้องทำการควบคุมความสัมพันธ์ A และ B ไม่ให้เกิดความซ้ำซ้อนเกิดขึ้นได้ เช่น นักศึกษาหนึ่งคนอาจลงทะเบียนเรียนได้หลายวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และวิชาใด ๆ ก็สามารถรองรับนักศึกษาได้หลายคน แต่นักศึกษาหนึ่งคนจะไม่สามารถลงทะเบียนวิชาใด ๆ ได้มากกว่า 1 ครั้งของการลงทะเบียน

2. Inter fact type uniqueness constraints (external uniqueness constraints)

เป็นกฎข้อบังคับความถูกต้องที่แสดงให้เห็นว่า ชนิดเอนติตี้ใด ๆ มีความสัมพันธ์กับชนิดเลเบิ้ล หรือ ชนิดเอนติตี้ ได้มากกว่าหนึ่ง โดยในทางกลับกัน ชนิดเลเบิ้ล หรือ ชนิดเอนติตี้ เหล่านั้น สามารถบ่งบอกถึงลักษณะเฉพาะของชนิดเอนติตตินั้นได้ ดังแสดงในแผนภาพดังนี้

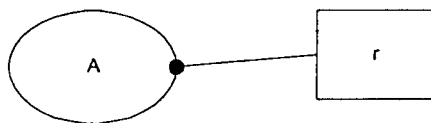


รูปที่ 2.10 ภาพแสดง Inter fact type uniqueness constraints

ลักษณะเช่นนี้สามารถแสดงได้ว่ากฎข้อบังคับความถูกต้องจะทำการควบคุม หากนำ $T1 \text{ join } T2$ แล้วผลที่ได้ BC จะไม่เกิดความซ้ำซ้อนกันเกิดขึ้น เช่น คนหนึ่งคนอาจมีชื่อหรือนามสกุลซ้ำกันได้ แต่ถ้า รวมทั้งชื่อและนามสกุลแล้วจะไม่เกิดความซ้ำซ้อน ดังนั้นจะสามารถบ่งบอกได้ว่าเป็นการระบุถึงคนใด

3. Mandatory role constraints

เป็นกฎข้อบังคับความถูกต้องที่ใช้ในการควบคุมเพื่อแสดงให้เห็นถึงการมีอยู่ของข้อมูลว่าต้องมีการบันทึกข้อมูลทุกครั้งที่เกิดมีความสัมพันธ์เกิดขึ้น สามารถแสดงได้ในแผนภาพดังนี้

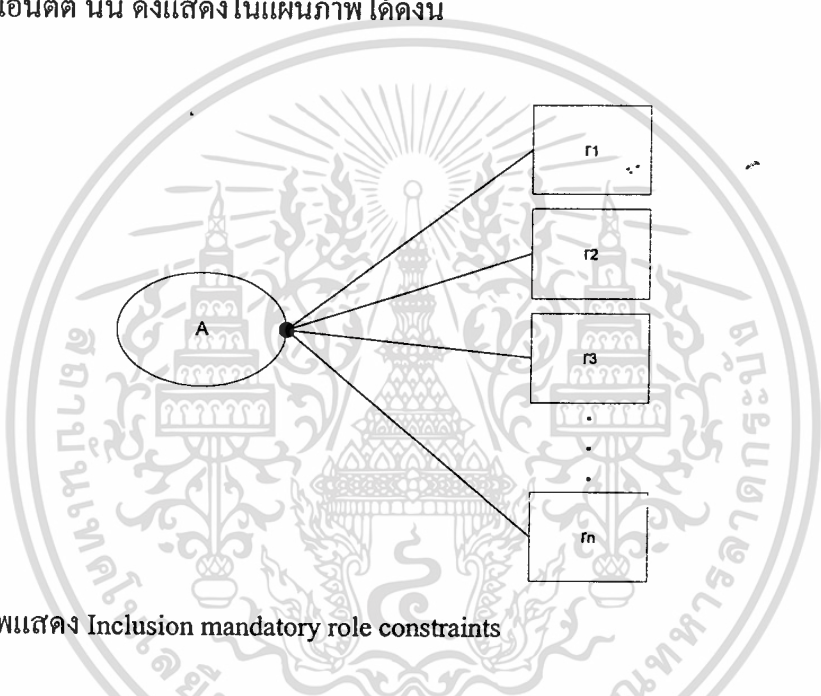


รูปที่ 2.11 ภาพแสดง Mandatory role constraints

จากภาพจุดทึบที่เชื่อมต่อระหว่าง ชนิดเอนติตี้ กับ Role นั้น แสดงให้เห็นว่าสมาชิก
ทุกตัวใน ชนิดเอนติตี้ A จะต้องถูกบันทึกข้อมูลเมื่อมีบทบาท r เกิดขึ้น โดยแสดง
ให้เห็นว่า $pop(A) = pop(r)$ เช่น นักศึกษาทุกคนต้องมีการบันทึกชื่อและนามสกุล
เป็นต้น

4. Inclusion mandatory role constraints

เป็นกฎข้อบังคับความถูกต้องที่แสดงให้เห็นถึงทางเลือกของบทบาทในกลุ่มของความ
สัมพันธ์ ที่มีอยู่ว่าต้องมีการบันทึกข้อมูลอย่างน้อยบทบาทใดบทบาทหนึ่งของ ชนิด
เอนติตี้ นั้น ดังแสดงในแผนภาพได้ดังนี้

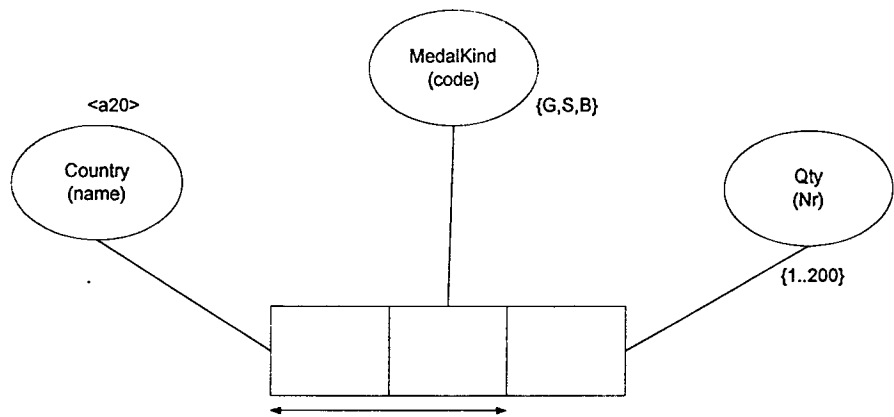


รูปที่ 2.12 ภาพแสดง Inclusion mandatory role constraints

จากภาพสามารถแสดงกฎข้อบังคับความถูกต้องของข้อมูล คือสมาชิกของชนิดเอนติตี้
A ใด ๆ ต้องมีการบันทึกความสัมพันธ์เกิดขึ้นความสัมพันธ์ใดความสัมพันธ์หนึ่ง ซึ่ง
แสดงได้ว่า $pop(A) = pop(r_1) \cup pop(r_2) \cup pop(r_3) \cup \dots \cup pop(r_n)$ เช่น
บุคคลใด ๆ จะต้องมีการระบุข้อมูลของบุตร หรือข้อมูลของบิดามารดาของบุคคล
นั้น ๆ อย่างน้อยที่สุดคนหนึ่งข้อมูล

5. Entity type constraints (Value constraints)

เป็นกฎข้อบังคับความถูกต้อง ที่ใช้ในการกำหนดค่าของสมาชิกภายในเซตของข้อมูล
ที่เป็นไปได้ของชนิดเลเบล หรือ ชนิดเอนติตี้ หนึ่ง ๆ รวมไปถึงการกำหนดชนิดของ
ข้อมูลในเซตด้วย ดังแสดงได้ในแผนภาพดังนี้

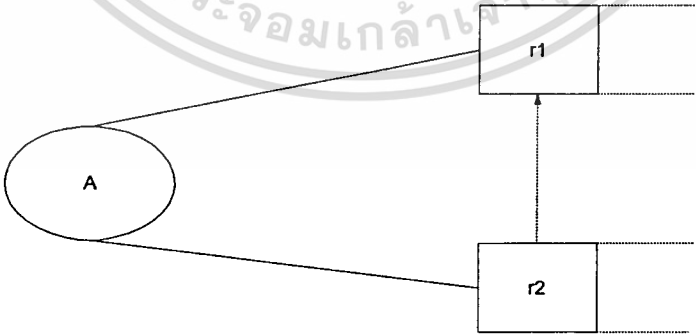


รูปที่ 2.13 ภาพแสดง Entity type constraints

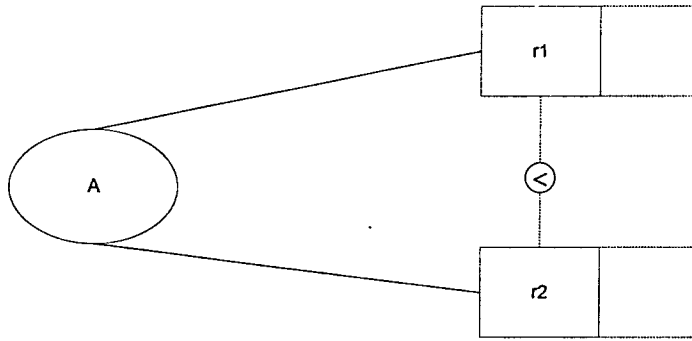
จากภาพนั้นจะมีกฎข้อบังคับความถูกต้องของข้อมูลเพื่อทำการระบุชนิดของเหรียญรางวัลในการแข่งขันกีฬาสามารถแยกออกได้เป็น เหรียญทองแดง, เหรียญเงิน, เหรียญทอง และระบุถึงจำนวนของเหรียญรางวัลที่ได้ว่าต้องอยู่ในช่วง 1 ถึง 200 เหรียญ รวมทั้งยังสามารถระบุชนิดของข้อมูลได้ด้วย ดังที่แสดงให้เห็นว่าชื่อประเทศนั้นกำหนดให้จัดเก็บได้มากที่สุด 20 ตัวอักษร

6. Subset constraints

เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่แสดงความสัมพันธ์ที่เป็นส่วนหนึ่งของความสัมพันธ์ที่มีอยู่ แต่จะมีลักษณะความสัมพันธ์ไปในทางเดียวดังแสดงความสัมพันธ์ได้โดยใช้สัญลักษณ์คือ $A \rightarrow B$ ซึ่งสามารถแสดงในแผนภาพได้ดังนี้



รูปที่ 2.14 ภาพแสดง Subset constraints



รูปที่ 2.14 (ต่อ) ภาพแสดง Subset constraints

ลักษณะดังกล่าวนี้แสดงให้เห็นกฎข้อบังคับความถูกต้องของข้อมูลว่า สมาชิกแต่ละตัวของ ชนิดเอนทิตี A หากมีการบันทึกความสัมพันธ์ r2 แล้ว ต้องมีการบันทึกความสัมพันธ์ r1 ด้วย แต่ในทางกลับกัน สมาชิกแต่ละตัวของ ชนิดเอนทิตี A หากมีการบันทึกความสัมพันธ์ r1 แล้วไม่จำเป็นต้องมีการบันทึกความสัมพันธ์ r2 ก็ได้ ดังแสดงได้ว่า $\text{pop}(r2) \subseteq \text{pop}(r1)$ เช่น บุคคลที่ชนะการแข่งขันกีฬา แสดงว่าต้องเป็นนักกีฬา แต่ผู้ที่เป็่นนักกีฬาไม่จำเป็นต้องเป็นผู้ชนะการแข่งขันทุกคน

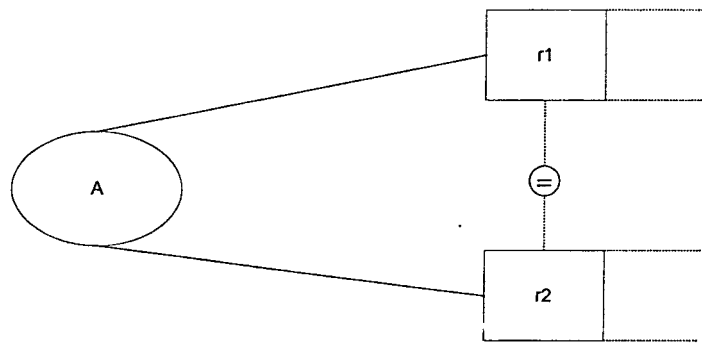
7. Equality constraints

เป็นกฎข้อบังคับความถูกต้องที่แสดงให้เห็นว่า ชนิดเอนทิตี เหล่านั้นจะต้องมีการถูกบันทึกข้อมูลควบคู่กันเสมอไป ใช้สัญลักษณ์แสดงความสัมพันธ์ได้คือ $A \leftrightarrow B$ ซึ่งสามารถแสดงในแผนภาพได้ดังนี้



รูปที่ 2.15 ภาพแสดง Equality constraints

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

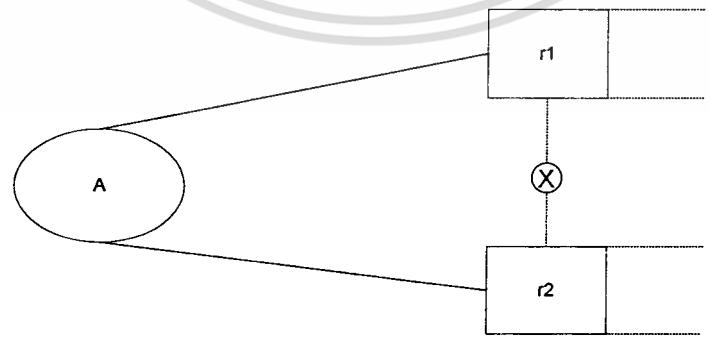


รูปที่ 2.15 (ต่อ) ภาพแสดง Equality constraints

ลักษณะดังกล่าวนี้ สามารถแสดงถึงกฎข้อบังคับความต้องการของข้อมูลว่า หากมีการบันทึกข้อมูลความสัมพันธ์ r1 ก็ต้องมีการบันทึกข้อมูลความสัมพันธ์ r2 ของสมาชิกของ ชนิดเอนติตี้ A ด้วย แสดงได้ว่า $pop(r1) = pop(r2)$ เช่น หากบุคคลใดจะทำการบันทึกระยะเวลาของการออกกำลังกาย ก็จะต้องทำการบันทึกข้อมูลของอัตราการเต้นของหัวใจด้วย และในทางกลับกัน หากมีการบันทึกข้อมูลอัตราการเต้นของหัวใจ ก็จะต้องทำการบันทึกข้อมูลระยะเวลาการออกกำลังกายด้วยเช่นกัน

8. Exclusion constraints

เป็นกฎข้อบังคับความต้องการที่มีลักษณะตรงข้ามกับ Equality constraints คือ แสดงความสัมพันธ์ที่ระบุว่าหากมีความสัมพันธ์แบบหนึ่งเกิดขึ้นจะต้องไม่มีความสัมพันธ์อีกแบบหนึ่งเกิดขึ้น โดยเด็ดขาด ซึ่งสามารถแสดงในแผนภาพได้ดังนี้

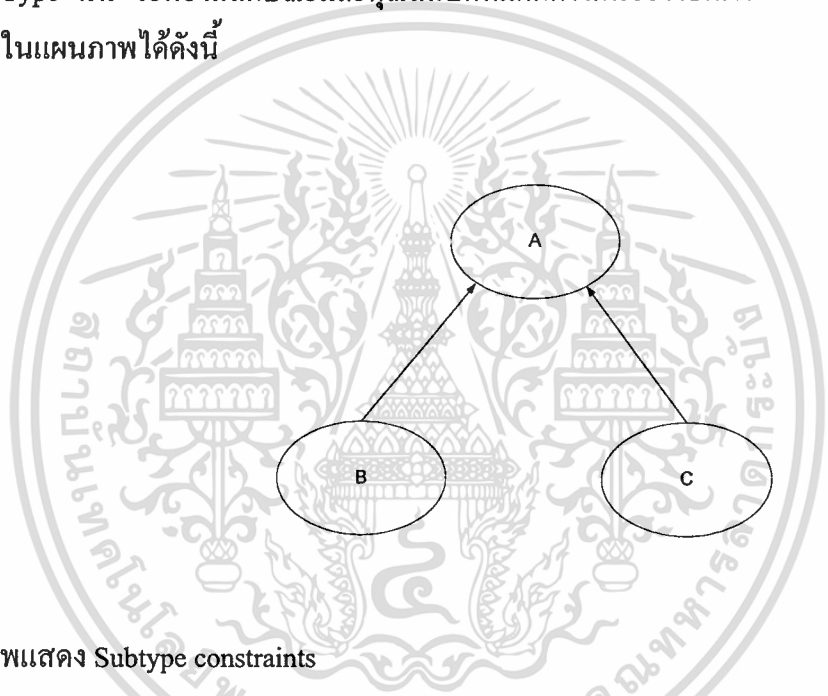


รูปที่ 2.16 แผนภาพแสดง Exclusion constraints

ลักษณะดังกล่าวนี้ แสดงให้เห็นกฎข้อบังคับความถูกต้องว่า หากมีการบันทึกข้อมูลความสัมพันธ์ r1 ของสมาชิกของ ชนิดเอนิตี A ใด จะต้องไม่มีการบันทึกข้อมูลความสัมพันธ์ r2 ของสมาชิกของ ชนิดเอนิตี A นั้นโดยเด็ดขาด ดังแสดงได้ว่า $pop(r1) \cap pop(r2) = \{ \}$ เช่น ถ้าบุคคลใดถูกเลือกให้เป็นกรรมการในการตัดสินใจตัดสินเกมส้นั้น บุคคลนั้นจะไม่มีสิทธิ์เป็นผู้แข่งขันในเกมส้อย่างเด็ดขาด

9. Subtype constraints

เป็นกฎข้อบังคับความถูกต้องที่ระบุถึงการแบ่งกลุ่มของสมาชิกของ ชนิดเอนิตี ที่มีอยู่อย่างชัดเจน ซึ่ง สมาชิกของ ชนิดเอนิตี ที่แบ่งแยกออกจาก ชนิดเอนิตี ที่เป็น Super Type นั้น จะต้องมิลักษณะและคุณสมบัติที่แตกต่างกันอย่างชัดเจน ดังสามารถแสดงในแผนภาพได้ดังนี้



รูปที่ 2.17 ภาพแสดง Subtype constraints

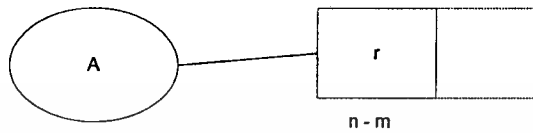
ลักษณะดังกล่าวนี้ แสดงให้เห็นว่า สมาชิกของ ชนิดเอนิตี A โดยจะเรียกว่า Super Type นั้นสามารถแบ่งออกเป็น 2 กลุ่มได้คือ กลุ่มของ ชนิดเอนิตี B และ กลุ่มของ ชนิดเอนิตี C ซึ่งเรียกว่า Subtype เช่น ชนิดเอนิตี ของบุคคล สามารถแบ่งออกเป็น Subtype ผู้ชาย และ ผู้หญิง ได้

10. Occurrence frequency constraints

เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่ใช้ในการระบุจำนวนครั้งที่สมาชิกของ ชนิดเอนิตีใด ๆ จะสามารถแสดงบทบาทใดบทบาทหนึ่งได้ ซึ่งสามารถแสดงในแผนภาพได้ดังนี้



(2.18a)



(2.18b)



(2.18c)

รูปที่ 2.18 ภาพแสดง Occurrence frequency constraints

จากรูปที่ 2.18(a) เป็นการแสดงกฎข้อบังคับความถูกต้องของข้อมูลโดยที่แต่ละ ชนิดเอนิตี A จะมีการแสดงบทบาทในคอลัมน์ r เป็นจำนวน n ครั้ง จากรูปที่ 2.18(b) เป็นการแสดงกฎข้อบังคับความถูกต้องของข้อมูลโดยที่แต่ละ ชนิดเอนิตี A ในการแสดงบทบาทในคอลัมน์ r ได้อย่างน้อยที่สุด n ครั้ง และมากที่สุด m ครั้ง และจากรูปที่ 2.18 (c) เป็นการแสดงกฎข้อบังคับความถูกต้องของข้อมูลโดยที่แต่ละ ชนิดเอนิตี A ในการแสดงบทบาทในคอลัมน์ r ได้อย่างน้อยที่สุด n ครั้ง เช่น ชมรมไคชมรมหนึ่งจะต้องมีสมาชิกอย่างน้อย 20 คน แต่จำนวนสมาชิกสูงสุดที่รับได้ต้องไม่เกิน 200 คน เป็นต้น

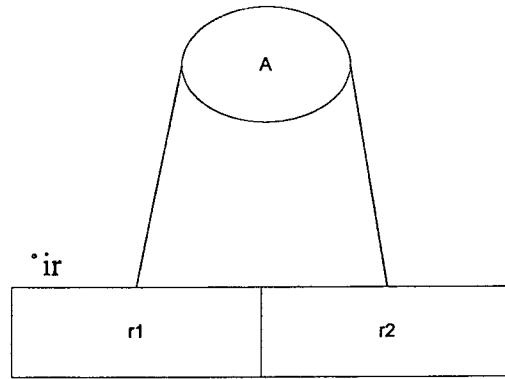
11. Irreflexive constraints

เป็นกฎข้อบังคับความถูกต้องที่แสดงให้เห็นว่าสมาชิกใด ๆ ของ ชนิดเอนิตี จะแสดงบทบาทเป็นทั้งประธานและกรรมของประโยคไม่ได้ กล่าวคือ ไม่มีคุณสมบัติการสะท้อนกลับของข้อมูล ดังเขียนในรูปประโยคสัญลักษณ์ได้คือ

$$R \text{ is irreflexive iff for all } x \sim xRx$$

ซึ่งสามารถแสดงในแผนภาพได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 ภาพแสดง Irreflexive constraints

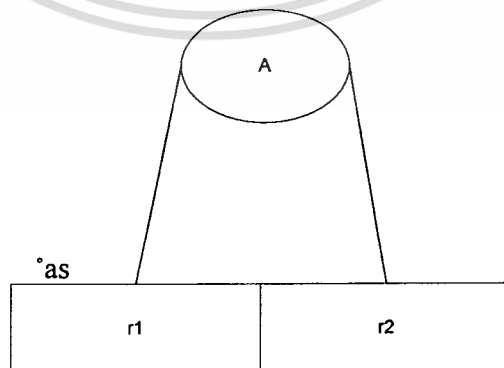
ลักษณะดังกล่าวแสดงให้เห็นว่า กฎข้อบังคับความถูกต้องของข้อมูล แต่ละสมาชิกของชนิดเอนทิตี A หากมีความสัมพันธ์ r1 เกิดขึ้นแล้ว จะมีความสัมพันธ์ r2 เกิดขึ้นด้วย ไม่ได้บนสมาชิกตัวเดียวกัน สรุปได้ว่า สมาชิกของ ชนิดเอนทิตี A ไม่มีคุณสมบัติการสะท้อนกลับ เช่น เด็กชายแดงจะเป็นพ่อของเด็กชายแดงไม่ได้

12. Asymmetric constraints

เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่เป็นสมาชิกใดๆของชนิดเอนทิตี ที่ถูกจับคู่กันในการแสดงบทบาทแล้ว สมาชิกคู่นั้นจะไม่สามารถแสดงบทบาทในทิศทางที่กลับกันได้ กล่าวคือไม่มีคุณสมบัติการสมมาตรของข้อมูล ดังสามารถเขียนอยู่ในรูปประโยคสัญลักษณ์ได้คือ

$$R \text{ is asymmetric iff for all } x,y \quad xRy \rightarrow \sim yRx$$

ซึ่งสามารถแสดงในแผนภาพได้ดังนี้



รูปที่ 2.20 ภาพแสดง Asymmetric constraints

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

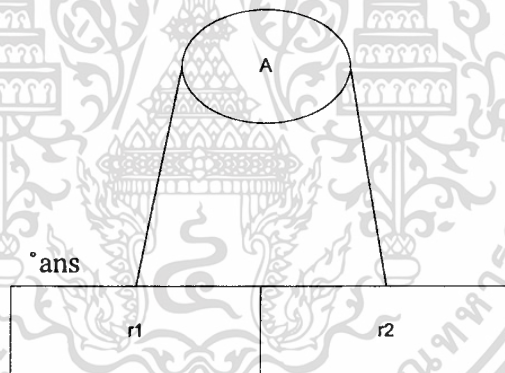
ลักษณะดังกล่าวแสดงให้เห็นว่า กฎข้อบังคับความถูกต้องของข้อมูล แต่ละสมาชิกของชนิดเอนทิตี A หากมีความสัมพันธ์ r1 เกิดกับสมาชิกตัวใดแล้ว จะต้องไม่มีความสัมพันธ์ r2 เกิดขึ้นย้อนกลับไปที่สมาชิกตัวเดิม สรุปได้ว่า สมาชิกของ ชนิดเอนทิตี A ไม่มีคุณสมบัติการสลับที่ เช่น หากเด็กชายแดงเป็นลูกของนายดำแล้ว นายดำจะเป็นลูกของเด็กชายแดงอีกไม่ได้

13. Antisymmetric constraints

เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่เป็นสมาชิกใด ๆ ของ ชนิดเอนทิตี ที่ถูกจับคู่กันในการแสดงบทบาทแล้ว สมาชิกคู่หนึ่งจะไม่สามารถแสดงบทบาทในทิศทางที่กลับกันได้ และจะไม่สามารถแสดงบทบาทกับชนิดเอนทิตีเดิมได้ด้วย กล่าวคือไม่มีคุณสมบัติการสมมาตรของข้อมูลและคุณสมบัติการสะท้อนกลับ ดังสามารถเขียนอยู่ในรูปประโยคสัญลักษณ์ได้คือ

R is antisymmetric iff for all x, y $x \neq y$ & $xRy \rightarrow \sim yRx$

ซึ่งสามารถแสดงในแผนภาพได้ดังนี้



รูปที่ 2.21 ภาพแสดง Antisymmetric constraints

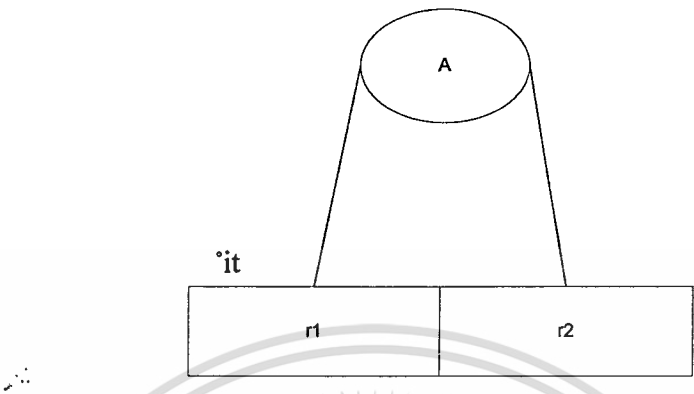
ลักษณะดังกล่าวแสดงให้เห็นว่า กฎข้อบังคับความถูกต้องของข้อมูล แต่ละสมาชิกของชนิดเอนทิตี A หากมีความสัมพันธ์ r1 เกิดกับสมาชิกตัวใดแล้ว จะต้องไม่มีความสัมพันธ์ r2 เกิดขึ้นย้อนกลับไปที่สมาชิกตัวเดิม สรุปได้ว่า สมาชิกของ ชนิดเอนทิตี A ไม่มีคุณสมบัติการสลับที่ เช่น หากเด็กชายแดงเป็นลูกของนายดำแล้ว นายดำจะเป็นลูกของเด็กชายแดงอีกไม่ได้

14. Intransitive constraints

เป็นกฎข้อบังคับความถูกต้องที่แสดงให้เห็นว่าความสัมพันธ์ที่ปรากฏอยู่บน fact type จะไม่มีคุณสมบัติการถ่ายทอด ดังสามารถเขียนอยู่ในรูปประโยคสัญลักษณ์ได้คือ

R is intransitive iff for all x,y,z $xRy \ \& \ yRz \rightarrow \sim xRz$

ซึ่งสามารถแสดงในแผนภาพได้ดังนี้



รูปที่ 2.22 ภาพแสดง Intransitive constraints

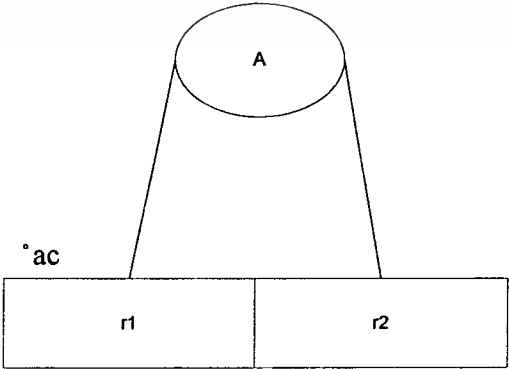
ลักษณะดังกล่าวแสดงให้เห็นว่า กฎข้อบังคับความถูกต้องของข้อมูลนั้นทำการควบคุมสมาชิกของ ชนิดเอนติตี้ A ไม่ให้เกิดคุณสมบัติการถ่ายทอด เช่น ถ้าเด็กชายแดงเป็นลูกของนายดำ และนายดำเป็นลูกของนายฟ้า จะไม่สามารถบันทึกข้อมูลที่ว่า เด็กชายแดงเป็นลูกของนายฟ้าได้

15. Acyclic Ring Constraints

เป็นกฎข้อบังคับความถูกต้องที่ควบคุมความสัมพันธ์ไม่ให้มีลักษณะของความสัมพันธ์การย้อนกลับได้ ดังสามารถเขียนอยู่ในรูปประโยคสัญลักษณ์ได้คือ

R is Acyclic Ring iff for all x,y,z $xRy \ \& \ yRz \rightarrow \sim zRx$

ซึ่งสามารถแสดงในแผนภาพได้ดังนี้



รูปที่ 2.23 ภาพแสดง Acyclic Ring constraints

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะดังกล่าวแสดงให้เห็นว่า กฎข้อบังคับความถูกต้องของข้อมูลนั้นทำการควบคุมสมาชิกของ ชนิดเอนติตี้ A เกิดความสัมพันธ์ย้อนกลับได้ เช่น ถ้าเด็กชายแดงเป็นลูกของนายดำ และนายดำเป็นลูกของนายฟ้า จะไม่สามารถบันทึกข้อมูลที่ว่า นายฟ้าเป็นลูกของเด็กชายแดงได้

16. Cardinality constraints

เป็นกฎข้อบังคับความถูกต้องที่ใช้ควบคุมจำนวนสมาชิกของ ชนิดเอนติตี้ ที่เป็นไปได้ ดังแสดงได้ในแผนภาพดังนี้

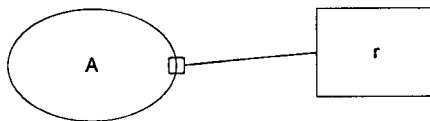


รูปที่ 2.24 ภาพแสดง Cardinality constraints

จากภาพแสดงให้เห็นกฎข้อบังคับความถูกต้องของข้อมูลที่ทำกรควบคุมจำนวนสมาชิกของ ชนิดเอนติตี้ A ให้มีการเกิดขึ้นได้อย่างมากที่สุดไม่เกิน n สมาชิก ในเวลาใดเวลาหนึ่ง เช่น คนที่เป็นคนปกติมีได้เพียงคนเดียวเท่านั้นในเวลาใดเวลาหนึ่ง

17. Relative closure constraints

เป็นกฎข้อบังคับความถูกต้องที่แสดงให้เห็นถึงการมีความสัมพันธ์ของ ชนิดเอนติตี้หนึ่งว่ามีการเกิดบทบาทของ ชนิดเอนติตี้ ดังกล่าวจริงหรือไม่ ดังนั้นถ้าไม่มีการเกิดบทบาทดังกล่าวถือว่า ชนิดเอนติตี้ นั้นไม่มีความสัมพันธ์นั้นอยู่ในข้อมูลความจริง ดังแสดงในแผนภาพได้ดังนี้



รูปที่ 2.25 ภาพแสดง Relative closure constraints

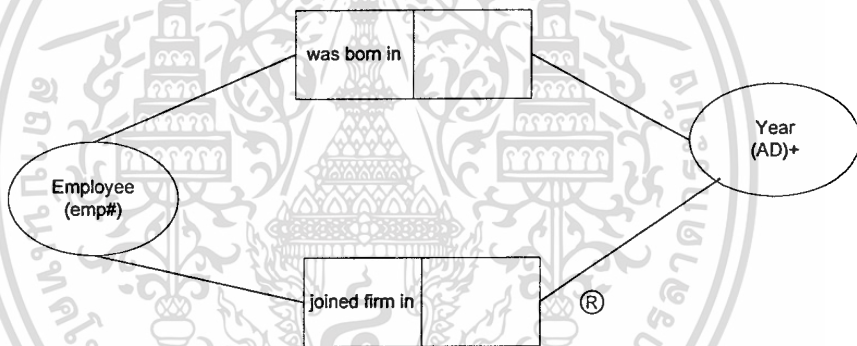
จากภาพกล่องสี่เหลี่ยมที่เชื่อมต่อระหว่าง ชนิดเอนติตี้ กับบทบาทนั้น แสดงให้เห็นว่า

เอกสารนี้เป็นเอกสารลิขสิทธิ์ใน ชนิดเอนติตี้ A ที่มีการบันทึกข้อมูลเมื่อมีบทบาท r เกิดขึ้นนั้นแสดงว่ามี ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ดังกล่าวเกิดขึ้นจริง หากสมาชิกใน ชนิดเอนิตี A ไ่ ใด ๆ ที่ไม่มีการบันทึกข้อมูลบทบาท r กล่าวได้ว่าสมาชิกนั้นไม่มีความสัมพันธ์ดังกล่าวเลย ถ้าไม่มีการใส่สัญลักษณ์ดังกล่าวนั้น ไม่อาจสรุปได้ว่ามีความสัมพันธ์นั้นหรือไม่ อาจมีแต่ข้อมูลที่ทราบมาไม่ครบถ้วนก็เป็นได้ เช่น พนักงานคนใดที่มีบทบาทในการจัดการแสดงว่ามีสถานะเป็นผู้จัดการหากไม่มีการบันทึกบทบาทดังกล่าว ถือได้ว่าไม่มีสถานะเป็นผู้จัดการ เป็นต้น

18. Textual constraints

เป็นกฎข้อบังคับความถูกต้องในกลุ่มของ Static constraints ที่ไม่สามารถอธิบายได้โดยใช้สัญลักษณ์แสดงความหมายของกฎข้อบังคับดังกล่าวบนแบบจำลอง ดังนั้นจึงมีการกำหนดสัญลักษณ์เพื่อระบุว่ามีกฎข้อบังคับความถูกต้องอยู่ และทำการแสดงกฎข้อบังคับดังกล่าวโดยรูปประโยค ดังแสดงในแผนภาพดังนี้



TC1 If Employee was born in Year y1 and Employee joined firm in Year y2 then y1 < y2

รูปที่ 2.26 ภาพแสดง Textual constraints

จากภาพเป็นการแสดงให้เห็นว่า ความสัมพันธ์ระหว่าง ชนิดเอนิตี Employee กับ ชนิดเอนิตี Year มีกฎข้อบังคับเกิดขึ้นโดยใช้สัญลักษณ์ดังที่ปรากฏในภาพ และมีการอธิบายความหมายของกฎข้อบังคับดังกล่าวโดยใช้ประโยคที่ว่า ถ้าพนักงานมีการระบุปีเกิด และปีที่เข้าทำงาน แล้ว ปีเกิดจะต้องมีค่าน้อยกว่าปีที่เข้าทำงานเสมอ

2.1.4 อัลกอริธึมในการแปลงแบบจำลองระดับแนวคิดในแอมโมโยนีในรูปแบบของโครงสร้าง

ฐานข้อมูลเชิงสัมพันธ์

1. เปลี่ยนทุก ๆ ชนิดความจริงที่มี 1 บทบาท (unary elementary fact type) ให้อยู่ในรูปแบบของชนิดความจริงแบบ 2 บทบาท (binary fact type)
2. ทำการย้ายชนิดความจริงทั้งหมดที่เชื่อมต่อกับ Super Type มาเชื่อมต่อกับ Sub Type

ของ Super Type นั้น ๆ แทน

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สร้างรีเลชันสำหรับแต่ละประโยชน์มูลฐานซึ่งมี intra uniqueness constraints กำกับอยู่บนทุกบทบาท โดยนำแต่ละชนิดเอนติตี้ที่เกี่ยวข้องกับชนิดความจริงนั้น ๆ มาสร้างเป็นแอตทริบิวต์ของรีเลชันนั้น ๆ ทุกแอตทริบิวต์ที่สร้างขึ้นจะถูกกำหนดให้เป็น mandatory และแอตทริบิวต์ทั้งหมดที่ถูก intra uniqueness constraints กำกับอยู่นั้นให้ทำการกำหนดเป็นคีย์หลัก (primary key) ของรีเลชัน
4. สร้างรีเลชันสำหรับแต่ละ n-ary fact type ที่มีมากกว่า 2 บทบาทขึ้นไปซึ่งมี intra uniqueness constraints กำกับอยู่ n-1 บทบาท โดยนำแต่ละชนิดเอนติตี้ที่เกี่ยวข้องกับชนิดความจริงนั้น ๆ มาสร้างเป็นแอตทริบิวต์ของรีเลชันนั้น ๆ ทุกแอตทริบิวต์ที่สร้างขึ้นจะถูกกำหนดให้เป็น mandatory และแอตทริบิวต์ทั้งหมดที่ถูก intra uniqueness constraints กำกับอยู่นั้นให้ทำการกำหนดเป็นคีย์หลัก (primary key) ของรีเลชัน
5. ทำการจัดกลุ่มชนิดความจริงที่มี 2 บทบาทที่เหลือ โดยมี intra uniqueness constraints กำกับอยู่ 1 บทบาทอยู่ฝั่งเดียวกัน สำหรับกรณีที่มี intra uniqueness constraints กำกับอยู่ 1 บทบาททั้งสองฝั่ง ให้ทำการจัดตามฝั่งที่มี mandatory constraints ที่ปรากฏอยู่ หากไม่ปรากฏอยู่ทั้งสองฝั่งให้ทำการจัดตามลำดับของตัวอักษร
6. สร้างรีเลชันตามกลุ่มที่ได้จัดไว้ในขั้นตอนที่ 5 โดยทำการสร้างแอตทริบิวต์ขึ้นจากชนิดเอนติตี้ที่มีอยู่ และทำการกำหนด mandatory constraints ตามชนิดเอนติตี้นั้น ๆ

2.2 แบบจำลองฐานข้อมูลเชิงวัตถุ (Object Oriented Database Model)

แบบจำลองฐานข้อมูลเชิงวัตถุเป็นวิธีการใหม่ที่ใช้ในการออกแบบฐานข้อมูล เพื่อให้สามารถจัดเก็บข้อมูลที่มีลักษณะซับซ้อน (Complex Data Type) ได้ เช่น ข้อมูลภาพและเสียง รวมไปถึงข้อมูลที่เป็น Abstract Data Type อีกด้วย ซึ่งรองรับกับการออกแบบระบบงานเชิงวัตถุ (Object Oriented Design) โดยในการออกแบบระบบงานใด ๆ ในลักษณะเชิงวัตถุนี้จะใช้เทคนิคของแบบจำลองเชิงวัตถุ (Object Oriented Modeling Technique: OMT) ตั้งแต่ การวิเคราะห์ระบบ, การออกแบบระบบงาน, การออกแบบฐานข้อมูล รวมไปถึงการพัฒนาแบบด้วย ซึ่งสามารถบ่งบอกถึงขั้นตอนการทำงานทั้งหมดภายในระบบได้ตามแผนภาพที่นำมาใช้ในการออกแบบ

แบบจำลองฐานข้อมูลเชิงวัตถุที่นำมาทำการศึกษาสำหรับงานวิจัยนี้คือ UML (Unified Modeling Language) [6, 7, 8, 9, 10] ซึ่งเป็นแบบจำลองที่มีการรวบรวมมาจากหลักการของ Booch, Rumbaugh และ Jacobson โดยมีการกำหนดเป็นมาตรฐานในการใช้งานจาก OMG (Object Management Group) ตั้งแต่ราวต้นปี ค.ศ. 1997 และเป็นที่ยอมรับและมีการใช้งานกันอย่างแพร่หลาย

ของแบบจำลองเชิงวัตถุ (Object Oriented Modeling Technique : OMT) ตั้งแต่ การวิเคราะห์ระบบ การออกแบบระบบงาน การออกแบบฐานข้อมูล รวมไปถึงการพัฒนาารบบด้วย

แบบจำลองฐานข้อมูลเชิงวัตถุที่นำมาทำการศึกษาสำหรับงานวิจัยนี้คือ UML (Unified Modeling Language) [6, 7, 8, 9, 10] ซึ่งเป็นแบบจำลองที่มีการรวบรวมมาจากหลักการของ Booch, Rumbaugh และ Jacobson โดยมีการกำหนดเป็นมาตรฐานในการใช้งานจาก OMG (Object Management Group) ตั้งแต่วาดต้นปี ค.ศ. 1997 และเป็นที่ยอมรับและมีการใช้งานกันอย่างแพร่หลาย

2.2.1 ส่วนประกอบพื้นฐานและสัญลักษณ์ของแบบจำลองฐานข้อมูลเชิงวัตถุ

1. คลาส (Class)

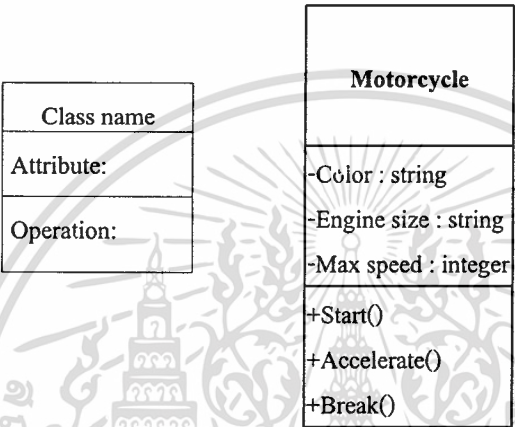
คลาส คือ การบอกลักษณะกลุ่มของออบเจกต์ (Objects) ซึ่งมีคุณสมบัติ, พฤติกรรม, ลักษณะของความสัมพันธ์กับออบเจกต์อื่น ๆ และโครงสร้างที่เหมือนกัน ประเภทของคลาสอาจจะมีการกำหนดได้หลายลักษณะดังนี้ คลาสที่เป็นสิ่งของหรือวัตถุที่มีอยู่จริง เช่น คน, บริษัท, รถยนต์ เป็นต้น หรือ คลาสที่อยู่ในลักษณะของเอนติตี้ระดับแนวคิด (Conceptual Entities) เช่น ตารางสอน, ตารางการบิน, สมการ เป็นต้น นอกจากนี้ยังมีคลาสที่กำหนดไว้เพื่อใช้เป็นเครื่องมือ เช่น แอเรย์ (Array), ไบนารีทรี (Binary Tree) เป็นต้น องค์ประกอบของคลาสสามารถแบ่งออกได้ 3 ส่วนคือ

- ชื่อคลาส (Class Name) เป็นการระบุชื่อของคลาส โดยทุก ๆ คลาสต้องมีการกำหนดชื่อ
- แอตทริบิวต์ (Attributes) เป็นการระบุคุณสมบัติต่าง ๆ ของคลาส โดยที่แต่ละคลาสอาจจะมีแอตทริบิวต์หรือไม่ก็ได้
- โอเปอเรชัน (Operations) หรือ เมธอด (Method) เป็นการกำหนดฟังก์ชันหรือขั้นตอน (Procedure) ซึ่งจะถูกระบุจากออบเจกต์ที่เป็นสมาชิกของคลาสนั้น ๆ

จากการที่คลาสนี้มีคุณสมบัติของการเอนแคปซูเลชัน (Encapsulation) ทำให้คลาสสามารถแบ่งระดับของการเข้าถึงองค์ประกอบต่าง ๆ ภายในคลาสที่กล่าวมาแล้วได้โดยจะมีการกำหนดสัญลักษณ์เพื่อแสดงคุณสมบัติดังกล่าวอยู่หน้าองค์ประกอบต่าง ๆ ของคลาส การเอนแคปซูเลชันสามารถแบ่งออกได้ 3 ระดับคือ

- สัญลักษณ์ + (Public) หมายถึงองค์ประกอบที่มีสัญลักษณ์ดังกล่าวจะสามารถให้คลาสอื่น ๆ เข้าถึงและเรียกใช้งาน

- สัญลักษณ์ – (Private) หมายถึงองค์ประกอบที่มีสัญลักษณ์ดังกล่าวจะสามารถเรียกใช้หรืออ้างถึงได้เพียงคลาสเดียวเท่านั้น และไม่สามารถอ้างถึงหรือเรียกใช้จากคลาสลูกได้
- สัญลักษณ์ # (Protected) หมายถึงองค์ประกอบที่มีสัญลักษณ์ดังกล่าวจะสามารถเรียกใช้หรืออ้างถึงได้เพียงคลาสเดียว แต่อนุญาตให้คลาสลูกสามารถใช้งานได้ด้วย



รูปที่ 2.27 ภาพแสดงสัญลักษณ์และตัวอย่างของคลาส

จากรูปดังกล่าวจะแสดงโดยใช้ช่องสี่เหลี่ยมดังกล่าว โดยช่องแรกจะระบุชื่อคลาส โดยจะต้องมีการกำหนดทุกครั้ง ช่องที่สองจะเป็นการกำหนดแอตทริบิวต์ของคลาสโดยรูปแบบของแอตทริบิวต์จะใช้รูปแบบดังนี้

Attribute_Name : Attribute_Type = Initial_Value
และช่องสุดท้ายเป็นการกำหนดโอเปอเรชั่นของคลาส ซึ่งจะใช้รูปแบบในการเขียนดังนี้

Operation_Name (Argument_Name : Argument_Type = Default_Value,...) : Return_Type

2. ความสัมพันธ์แบบแอสโซซิเอชัน (Association Relationships)
- เป็นการแสดงความสัมพันธ์ระหว่างคลาสของออบเจกต์ โดยความสัมพันธ์ดังกล่าวใช้เชื่อมออบเจกต์ที่เป็นออบเจกต์อิสระของคลาส 2 คลาส สัญลักษณ์ที่ใช้คือเส้นตรงเชื่อมระหว่างคลาส 2 คลาส ดังรูปที่ 2.28



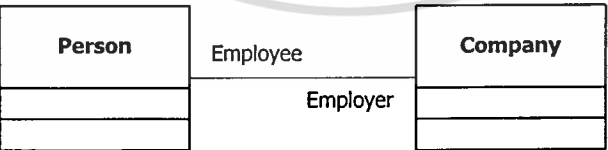
รูปที่ 2.28 ภาพแสดงสัญลักษณ์ความสัมพันธ์แบบแอโซซิเอชัน

3. ชื่อแอโซซิเอชัน (Association Naming)
- เป็นการบอกถึงรายละเอียดของความสัมพันธ์ระหว่างคลาส โดยจะกำหนดชื่อแอโซซิเอชันอยู่ตรงกลางของเส้น โดยชื่อดังกล่าวมักจะอยู่ในรูปของคำกริยา



รูปที่ 2.29 ภาพแสดงชื่อของแอโซซิเอชัน

4. ชื่อบทบาท (Role Naming)
- ในแต่ละปลายของแอโซซิเอชันอาจมีการกำหนดชื่อของบทบาท (Role Naming) ได้ โดยชื่อดังกล่าวนั้นจะอธิบายลักษณะของคลาสหนึ่ง ๆ มองเห็นคลาสฝั่งตรงข้ามอย่างไร ชื่อบทบาทนี้มักจะแสดงอยู่ในรูปของ noun ดังรูปที่ 2.30



รูปที่ 2.30 ภาพแสดงสัญลักษณ์ของชื่อบทบาท

5. Multiplicity of Associations
- ในแต่ละบทบาทของแอโซซิเอชันนั้นมันมีค่าได้หลายค่า เพื่อระบุจำนวนออบเจ็กต์ที่อยู่ในคลาสที่กำหนดคนั้น ๆ ว่า มีการเชื่อมต่อกับออบเจ็กต์ของอีกคลาสหนึ่ง โดยการกำหนดขอบเขตสามารถใช้สัญลักษณ์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงสัญลักษณ์ที่ใช้แทนความหมายของ Multiplicity of Associations

สัญลักษณ์	ความหมาย
1	มีค่าได้เพียงหนึ่งออบเจกต์เท่านั้น
0..1	มีค่าได้เพียงศูนย์หรือหนึ่งออบเจกต์
M..N	มีค่าจาก M ถึง N ออบเจกต์ (M และ N คือจำนวนนับ)
*	มีค่าจากศูนย์ถึงจำนวนเต็มบวกใด ๆ
0..*	มีค่าจากศูนย์ถึงจำนวนเต็มบวกใด ๆ
1..*	มีค่าจากหนึ่งถึงจำนวนเต็มบวกใด ๆ

การระบุ Multiplicity of Associations เปรียบเสมือนการกำหนดกฎข้อบังคับ ความถูกต้องของข้อมูลในลักษณะที่กล่าวถึงการมีอยู่ของข้อมูล ดังภาพแสดงตัวอย่างของ Multiplicity values ในรูปที่ 2.31

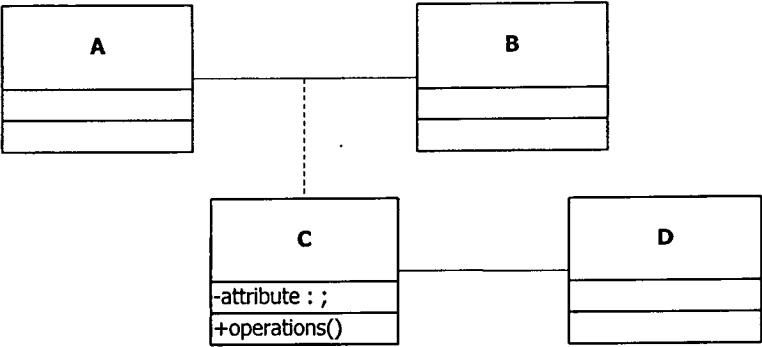


รูปที่ 2.31 ภาพแสดงตัวอย่างของ Multiplicity of Associations

จากภาพจะเห็นว่านักเรียน 1 คนจะอยู่ในห้องได้ห้องเดียว แต่ห้องหนึ่งห้องสามารถบรรจุนักเรียนได้หลายคน เป็นต้น

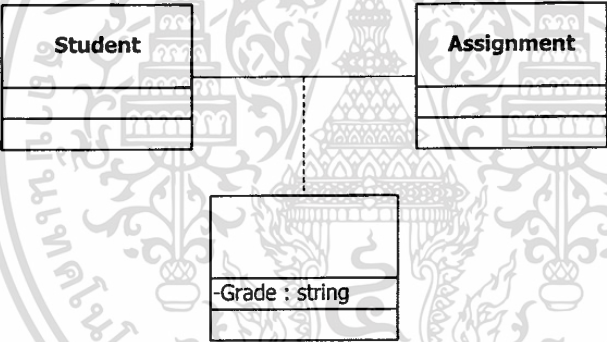
6. คลาสแอสโซซิเอชัน (Association classes)

ความสัมพันธ์แบบแอสโซซิเอชันนั้นสามารถที่จะนำเสนอเป็นคลาสได้ เพื่อที่จะสามารถเพิ่มแอตทริบิวต์และโอเปอเรชันที่เกี่ยวข้องกับความสัมพันธ์นั้นได้ คลาสประเภทนี้จะมีคุณสมบัติเหมือนกันคลาสดอื่น ๆ ทั่วไปโดยสามารถที่จะสร้างความสัมพันธ์กับคลาสดอื่น ๆ ได้ด้วย สัญลักษณ์ที่ใช้ในการแสดงคลาสแอสโซซิเอชันจะใช้เป็นลักษณะของเส้นประเพื่อเชื่อมต่อไปยังคลาสดดังกล่าว สามารถแสดงได้ดังรูปที่ 2.32



รูปที่ 2.32 ภาพแสดงคลาสแอตโซซิเอชัน

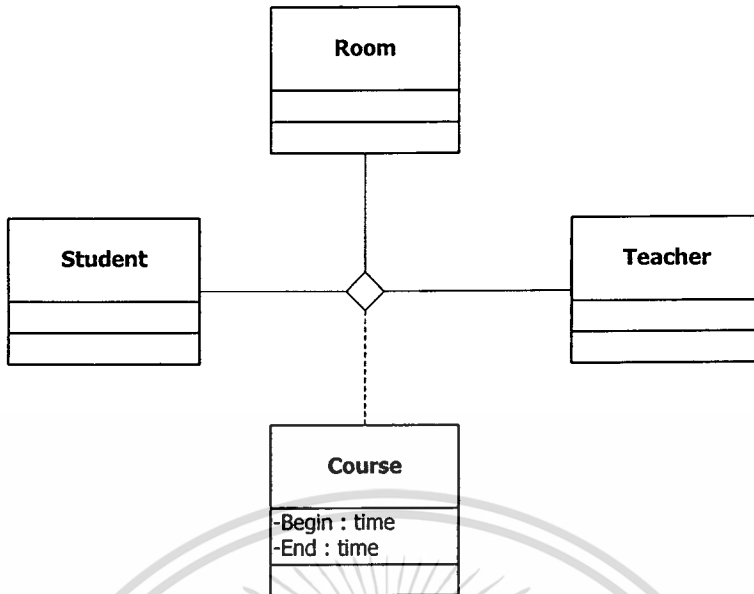
หากคลาสแอตโซซิเอชันที่ไม่มีความสัมพันธ์เชื่อมต่อกับคลาสอื่น ๆ เราอาจเรียกว่า แอตทริบิวต์แอตโซซิเอชัน (attributed association) ได้ ดังแสดงในรูปที่ 2.33



รูปที่ 2.33 ภาพแสดงตัวอย่างของแอตทริบิวต์แอตโซซิเอชัน

7. N-ary Associations

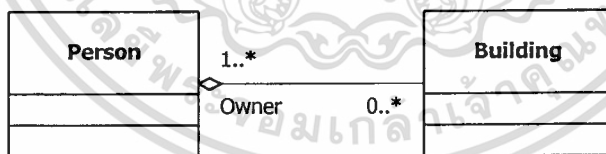
เป็นการแสดงความสัมพันธ์แบบแอตโซซิเอชันระหว่างคลาสที่มีมากกว่า 2 คลาสขึ้นไป โดยจะใช้สัญลักษณ์ diamond เป็นตัวเชื่อมความสัมพันธ์ ดังแสดงได้ในรูปที่ 2.34



รูปที่ 2.34 ภาพแสดงตัวอย่างความสัมพันธ์แบบ N-ary Associations

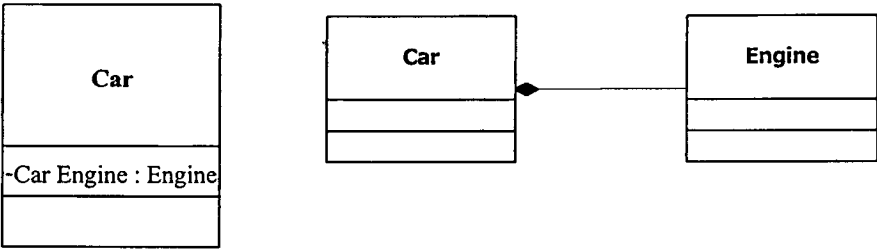
8. ความสัมพันธ์แบบเอกกรีเกชั่น (Aggregation Relationships)

เป็นการแสดงความสัมพันธ์ในลักษณะที่ไม่สมมาตรกล่าวคือ ปลายด้านหนึ่งของความสัมพันธ์จะมีบทบาทมากกว่าปลายอีกด้านหนึ่ง ดังนั้นจึงมักมีการกำหนดบทบาทเพียงด้านเดียวเท่านั้น สัญลักษณ์ที่ใช้จะเพิ่มสี่เหลี่ยมขนมเปียกปูนที่ปลายด้านหนึ่งของเส้น ดังแสดงได้ในรูปที่ 2.35



รูปที่ 2.35 ภาพแสดงความสัมพันธ์แบบเอกกรีเกชั่น

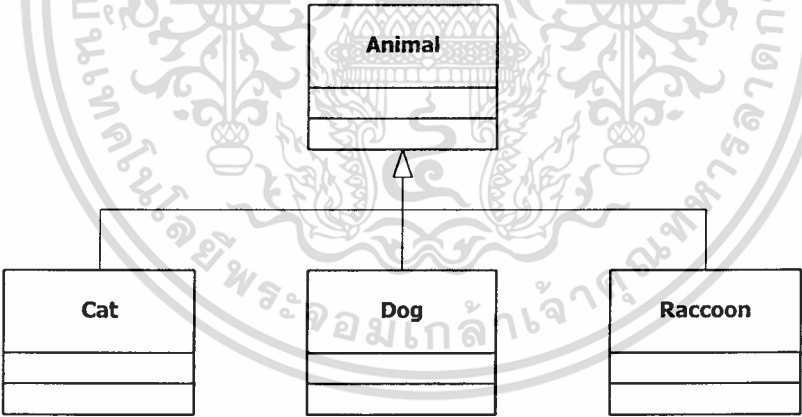
โดยทั่วไปมักใช้ความสัมพันธ์แบบเอกกรีเกชั่นในการบ่งบอกถึงการเป็นส่วนประกอบ (Composition) หรือใช้แทนความหมายของคำว่า “is a part of” หรือ “is made up of” โดยแสดงให้เห็นว่าออบเจกต์ของคลาสที่ประกอบขึ้นมาจากออบเจกต์อื่น ๆ ที่อยู่ในคลาสอื่น ซึ่งจะใช้สัญลักษณ์เป็นขนมเปียกปูนสี่เหลี่ยม แทน ดังแสดงได้ในรูปที่ 2.36



รูปที่ 2.36 ภาพแสดงความสัมพันธ์แบบแอกกรีเกชันในลักษณะของการเป็นส่วนประกอบ

9. ความสัมพันธ์แบบเจนเนอรัลไลเซชัน (Generalization Relationships)

เป็นความสัมพันธ์ที่ใช้ในการแสดงว่าคลาสสามารถที่จะบ่งบอกประเภทของสมาชิกของคลาสอีกคลาสหนึ่งได้ ความสัมพันธ์แบบนี้จะใช้ในการแทนความหมายของ “is a” หรือ “is a kind of” ซึ่งจะใช้สัญลักษณ์ในการแทนความสัมพันธ์โดยเป็นลูกศรชี้จากคลาสที่มีรายละเอียดเฉพาะมากกว่าไปยังคลาสทั่วไป โดยจะเรียกคลาสทั่วไปว่า ซุปเปอร์คลาส (Superclass) และคลาสที่มีรายละเอียดมากกว่าว่า ซับคลาส (Subclasses) ดังแสดงได้ในรูปที่ 2.37

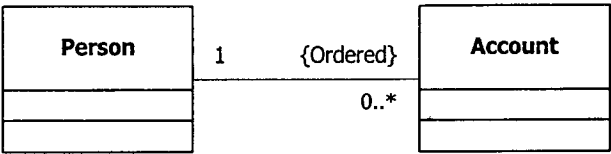


รูปที่ 2.37 ภาพแสดงความสัมพันธ์แบบเจนเนอรัลไลเซชัน

10. กฎข้อบังคับความถูกต้องของข้อมูล (Integrity Constraints)

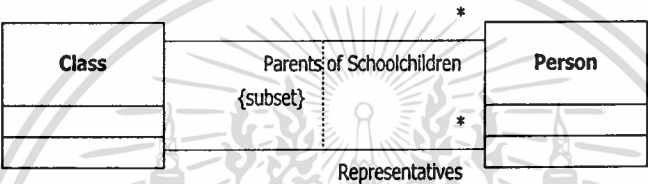
แบบจำลองฐานข้อมูลเชิงวัตถุนั้นจะมีการกำหนดสิ่งที่ใช้ในการควบคุมความถูกต้องของข้อมูลบนแบบจำลองได้โดยจะกำหนดกฎข้อบังคับความถูกต้องของข้อมูลไว้ภายในเครื่องหมาย { } และมักจะนำมาวางแทนที่ชื่อของบทบาท ดังแสดงได้ในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



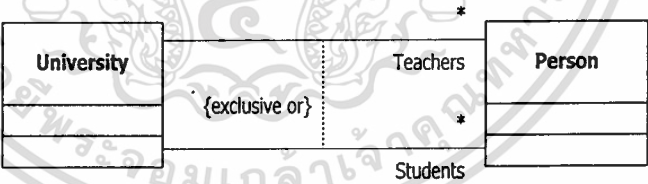
รูปที่ 2.38 ภาพแสดงกฎข้อบังคับความถูกต้องของข้อมูลแบบ Ordered

แสดงกฎข้อบังคับความถูกต้องของข้อมูล {Ordered} โดยเป็นการบอกลักษณะของออบเจกต์ที่อยู่ภายในคลาส Account ซึ่งเป็นเซตของออบเจกต์เหล่านั้นจะมีการเรียงลำดับของออบเจกต์ด้วย



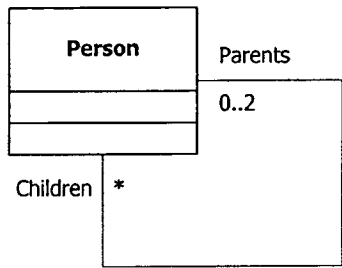
รูปที่ 2.39 ภาพแสดงกฎข้อบังคับความถูกต้องของข้อมูลแบบ subset

จากภาพแสดงกฎข้อบังคับความถูกต้องเพื่อระบุว่ากลุ่มของออบเจกต์ในคลาสเป็นซับเซตกับกลุ่มของออบเจกต์อีกกลุ่มหนึ่งในคลาสเดียวกัน



รูปที่ 2.40 ภาพแสดงกฎข้อบังคับความถูกต้องของข้อมูลแบบ exclusive or

จากภาพแสดงให้เห็นว่ามีการกำหนดกฎข้อบังคับความถูกต้อง เพื่อระบุว่าออบเจกต์ที่ทำการกำหนดนั้น สามารถมีความสัมพันธ์ได้เพียงความสัมพันธ์เดียวเท่านั้นจากความสัมพันธ์ที่มีทั้งหมด



รูปที่ 2.41 ภาพแสดงกฎข้อบังคับความถูกต้องแบบ reflexive association

จากภาพแสดงให้เห็นถึงความสำคัญในการระบุขอบเขตสำหรับความสัมพันธ์แบบ reflexive association ซึ่งใช้ในการควบคุมความถูกต้องของข้อมูลที่ปรากฏอยู่

2.2.2 อัลกอริธึมที่ใช้ในการแปลงแบบจำลองฐานข้อมูลเชิงวัตถุให้อยู่ในรูปของโครงสร้างฐานข้อมูลเชิงวัตถุ

1. กำหนดโดเมนของออบเจกต์ต่าง ๆ ดังนี้ โดเมนปกติ (Simple Domain) นำมาสร้างเป็นชนิดของข้อมูลทั่วไป หากมีการกำหนดโดเมนของออบเจกต์เป็นคลาส ให้ทำการสร้างเป็นโครงสร้างของ Embed Object Class หากโดเมนของออบเจกต์มีหลายค่า ให้ทำการสร้างเป็นเซตของโดเมนดังกล่าว
2. สร้างคลาสจากแบบจำลองโดยทำการสร้างแอตทริบิวต์ของคลาสตามที่กำหนดในแบบจำลองและทำการกำหนดโดเมนของแต่ละแอตทริบิวต์ดังที่ได้กล่าวมาแล้ว ในขั้นตอนที่ 1 และทำการสร้างโอเปอร์เรชันของคลาสตามที่กำหนดในแบบจำลอง
3. จากความสัมพันธ์แบบเจนเนอรัลไลเซชัน นำซูเปอร์คลาสและซับคลาสมาทำการสร้างเป็นคลาส แต่กำหนดให้ซับคลาสเป็นคลาสที่ inherit มาจากซูเปอร์คลาส
4. จากความสัมพันธ์แบบเอกกรีเกชันมาทำการสร้างเป็นนำ Composite Class มาทำการสร้างเป็นคลาสย่อย และกำหนดให้เป็นโดเมนของแอตทริบิวต์ในคลาสหลัก หรือนำคลาสย่อยมาทำการสร้างเป็นโครงสร้างแบบ struct ในคลาสหลักเพื่อให้อยู่ในรูปของ Embed Object Class
5. จากความสัมพันธ์แบบแอสโซซิเอชัน ที่อยู่ระหว่าง 2 คลาส ให้ทำการสร้างเป็น relationship และ inverse relationship ในลักษณะตาม Multiplicity of Association เช่น หากมีการกำหนดเป็น * ให้ทำการสร้างเป็น relationship แบบเซต และหากมี Ordered association ให้ทำการเปลี่ยน relationship จาก set เป็น list
6. ทำการสร้างคลาสใหม่จาก association class และ ternary association และสร้าง relationship ไปยังคลาสที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่เผยแพร่เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การเปรียบเทียบแบบจำลองฐานข้อมูลเชิงวัตถุกับแบบจำลองฐานข้อมูลเชิงสัมพันธ์

การเปรียบเทียบในที่นี้จะนำเสนอเกี่ยวกับการจัดการข้อมูลในลักษณะที่บ่งบอกถึงความสามารถของแบบจำลองข้อมูลที่รองรับกับความต้องการที่มีการเปลี่ยนแปลงอยู่ตลอดเวลา ซึ่งสามารถสรุปได้ดังนี้คือ

2.3.1 ชนิดของข้อมูล (Data Types)

ในระบบจัดการฐานข้อมูลเชิงสัมพันธ์จะทำการจัดการเกี่ยวกับโครงสร้างของข้อมูลในรูปแบบรีเลชันเท่านั้น โดยคำจำกัดความของแอตทริบิวต์ของรีเลชันที่ผ่านการทำนอร์มอลไรซ์มาแล้วนั้นจะไม่สามารถแบ่งแยกออกเป็นส่วนตัวย่อย ๆ ได้อีก และมีรูปแบบชนิดของข้อมูลอยู่ในลักษณะอย่างง่าย เช่น Integer, Real, String, Date เป็นต้น โอเปอเรชันบนรีเลชันได้ถูกจำกัดเพื่อที่จะทำการสืบค้น หรือปรับปรุงทUPLE (Tuples) ที่ถูกระบุโดยค่าของแอตทริบิวต์เท่านั้น เหตุผลหลักในงานลักษณะนี้เพื่อที่จะขยายความสามารถของรูปแบบของแบบจำลองเชิงสัมพันธ์ขณะที่ยังคงลักษณะที่ให้ง่ายต่อการใช้งานอยู่

สำหรับฐานข้อมูลเชิงวัตถุนั้นจะทำการจัดเก็บคำจำกัดความของคลาสและตัวอย่างของคลาสเหล่านั้น คลาสนั้นก็เปรียบเสมือนกับรีเลชันที่ใช้งานบนฐานข้อมูลเชิงสัมพันธ์ แต่มีลักษณะพิเศษที่สำคัญซึ่งเพิ่มขึ้นมาคือ คลาสจะทำการเอนแคปซูเลชันพฤติกรรมทั้งหมดของออบเจกต์โดยการรวมโอเปอเรชันและโครงสร้างของข้อมูลไว้ด้วยกัน คุณสมบัติของคลาสนั้นไม่จำเป็นต้องเป็นชนิดของข้อมูลในรูปแบบอย่างง่าย แต่สามารถอ้างอิงถึงคลาสอื่น ๆ ที่มีความซับซ้อนใด ๆ ได้ในระบบเชิงวัตถุสามารถที่จะทำการกำหนดขอบเขตของคลาสตามความถี่ของชนิดข้อมูลที่มีการใช้งาน ผู้ใช้สามารถปรับปรุงคลาสเหล่านี้ได้ตามความต้องการของโปรแกรมประยุกต์ที่ใช้งานนั้น ๆ โดยใช้คุณสมบัติที่มีอยู่อย่างง่ายเช่น การจัดกลุ่ม (Generality) และ การสืบทอด (Inheritance)

2.3.2 ความแน่นอนของข้อมูล (Data Integrity)

ฐานข้อมูลเชิงสัมพันธ์ มีความต้องการในการที่จะควบคุม referential integrity ของข้อมูล แต่บ่อยครั้งที่จุดมุ่งหมายนี้มักจะเป็นหน้าที่ของผู้พัฒนาโปรแกรมประยุกต์ ในบางระบบนั้นกำหนดเงื่อนไขสำหรับกฎข้อบังคับความถูกต้องของข้อมูลในเรื่องของ referential integrity แยกออกมาเฉพาะ ในส่วนนี้นั้นเป็นความรับผิดชอบของผู้พัฒนาฐานข้อมูลที่จะแน่ใจได้ว่ากฎข้อบังคับความถูกต้องทั้งหมดได้ถูกกำหนดรวมอยู่เรียบร้อยแล้ว โดยที่ระบบจะทำการควบคุมกฎข้อบังคับความถูกต้องเหล่านั้นในช่วงที่ระบบทำงาน แต่การทำลักษณะนี้เป็นการเสียเวลาและสิ้นเปลืองทรัพยากรค่อนข้างมาก

ตัวอย่างของปัญหาที่เกิดขึ้น กับฐานข้อมูลเชิงสัมพันธ์เช่น ฐานข้อมูลเชิงสัมพันธ์ไม่สามารถที่จะแสดงความสัมพันธ์ซึ่งกันเป็นความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วย (one-to-one relationships) หรือ ความสัมพันธ์แบบหนึ่งต่อหลายหน่วย (one-to-many relationships) ได้ กฎข้อบังคับดังกล่าวจะถูกสร้างและจัดเก็บไว้ในโปรแกรมประยุกต์ (Application Code) ที่ใช้งานร่วมกัน ไม่ว่าจะเป็นกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับฐานข้อมูลเชิงสัมพันธ์ แต่โดยทั่วไปแล้วโค้ดที่สร้างขึ้นนั้นไม่ได้ถูกใช้ร่วมกันระหว่างหลาย ๆ โปรแกรมประยุกต์ ดังนั้นจึงเป็นการยากที่จะแน่ใจได้ว่าข้อมูลดังกล่าวจะถูกปรับปรุงให้คงสภาพความถูกต้องได้ตลอดเวลา

ในทางตรงกันข้ามแบบจำลองเชิงวัตถุ นั้น คลาสจะมีการกำหนดในลักษณะของ data abstraction และ abstraction นี้จะทำการรวบรวมรายละเอียดของโอเปอเรชัน (methods) ซึ่งสามารถนำไปใช้งานกับตัวอย่างของคลาสได้ โดยนิยามของฐานข้อมูลในความหมายของ Abstraction ดังกล่าวมีความเป็นอิสระของข้อมูลสูงมาก นั่นคือเป็นไปได้ที่จะทำการเปลี่ยนวิธีดำเนินการในคลาสโดยที่ปราศจากผลกระทบต่อคลาส หรือ ทรานแซกชัน (Transactions) ที่มีการใช้ของ abstraction นั้น ๆ ดังนั้น โอเปอเรชันซึ่งถูกกำหนดขึ้นโดยผู้ใช้สามารถสร้างและจัดเก็บไว้ในฐานข้อมูลโอเปอเรชันหนึ่ง ๆ บนออบเจกต์สามารถเกิดขึ้นได้กับหลาย ๆ ฐานข้อมูล โดยที่โอเปอเรชันและทุก ๆ ทรานแซกชันที่ต้องขอโอเปอเรชันจะใช้โค้ดเหมือนกัน ซึ่งโค้ดดังกล่าวจะถูกเก็บอยู่ในรูปของกระบวนการ (Procedure) โดยจัดเก็บไว้ในคลาสจึงถือว่าเป็นข้อได้เปรียบอย่างมากเกี่ยวกับการดูแลความถูกต้องของข้อมูล เนื่องจากการตรวจสอบความสอดคล้องอาจถูกรวมเข้าไว้ในกระบวนการและจัดเป็น โอเปอเรชันหนึ่งเหมือนกับเป็นส่วนหนึ่งของฐานข้อมูล ดังนั้นส่วนใหญ่โค้ดของโปรแกรมประยุกต์จึงอยู่ภายใต้การควบคุมของผู้ดูแลฐานข้อมูล (Database Administrator) และสามารถถูกจัดการโดยยูทิลิตี้ของการจัดการฐานข้อมูลทั้งหมด

ความแน่นอนของเอนทิตี (Entity Integrity) จะถูกจัดการด้วยวิธีการที่แตกต่างกันในระบบเชิงวัตถุ ทุก ๆ ออบเจกต์มี Unique identity และออกเจกต์อื่น ๆ สามารถอ้างถึง identity นั้นได้ ออบเจกต์ยังคงรักษา identity นั้นไว้ได้ตลอดไม่ว่าออบเจกต์จะมีการเปลี่ยนสถานะไปอย่างไรก็ตาม ซึ่งจะตรงข้ามกับแบบจำลองเชิงสัมพันธ์ ซึ่งคุณสมบัติของเอนทิตีจะต้องมีอย่างเพียงพอในการที่จะแยกแยะความแตกต่างของเอนทิตีหนึ่ง ๆ ออกจากเอนทิตีอื่น ๆ ได้ ซึ่งก็คือจะต้องมีอย่างน้อยหนึ่งแอตทริบิวต์ขึ้นไปซึ่งค่าของแอตทริบิวต์นั้นต้องมีเพียงหนึ่งค่าและต้องไม่มีการเปลี่ยนแปลงคุณสมบัติใด ๆ ซึ่งมักจะ ไม่ตรงกับสภาพความเป็นจริง และมักต้องทำการกำหนด Artificial identifiers สำหรับเอนทิตีในฐานข้อมูลเชิงสัมพันธ์เสมอ

2.3.3 วิวัฒนาการของโครงสร้าง (Schema Evolution)

ระบบจัดการฐานข้อมูลมีแนวโน้มเพื่อที่จะวิวัฒนาการจำนวนชนิดของออบเจกต์พื้นฐานโดยรอบซึ่งเป็นรูปแบบของหลักการของระบบ ขณะที่วิวัฒนาการของระบบนั้น มุมมองของข้อมูลแนวใหม่ถูกเพิ่มโดยการควบคุมหรือการขยายหลักการของออบเจกต์ที่มีอยู่ และโปรแกรมประยุกต์แนวใหม่ในมุมมองเหล่านี้จะถูกสร้างโดยใช้โมดูลของโค้ดที่มีอยู่แล้วเป็นพื้นฐาน ส่วนระบบจัดการฐานข้อมูลเชิงสัมพันธ์มีแนวโน้มกำหนดขอบเขตอย่างมากสำหรับการขยายหรือการเปลี่ยนแปลงของโครงสร้างของข้อมูลที่มีอยู่ ตัวอย่างเช่น ระบบธุรกิจเชิงสัมพันธ์อนุญาตให้มีการเปลี่ยนแปลงของการสร้างที่ส่งวน การลบของรีเลชันอย่างแบบธรรมดา และการเพิ่มคีย์หลักใหม่หรือรีเลชันไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนชนิดของโดเมนมักจะก่อให้เกิดการสร้างใหม่ของรีเลชันที่ต้องการอยู่เสมอ และจะต้องทำการเปลี่ยนแปลงโปรแกรมประยุกต์ที่จะทำการใช้งานกับรีเลชันนั้นๆด้วย สิ่งเหล่านี้แสดงให้เห็นความจริงที่ว่าโครงสร้างของข้อมูลประกอบขึ้นมาจากรูปแบบของฐานข้อมูล (Database Schema) และโปรแกรมประยุกต์ซึ่งมีความเกี่ยวข้องกันน้อยมาก

ความเกี่ยวข้องกันอย่างมากระหว่างโปรแกรมและข้อมูลในแบบจำลองเชิงวัตถุแสดงถึงขอบเขตที่กว้างขวางสำหรับวิวัฒนาการของโครงสร้างตลอด โดยการขยายและการกำหนดใหม่ของโครงสร้างข้อมูลที่มีอยู่แล้วและยังแสดงให้เห็นถึงประสิทธิภาพของการนำมาใช้ใหม่ของโค้ดของโปรแกรมประยุกต์อีกด้วย ความสมบูรณ์ของแบบจำลองเชิงวัตถุนี้อนุญาตรูปแบบของการวิวัฒนาการโครงสร้างได้มีการกำหนดขึ้นอย่างตายตัวและมีเหตุผล



OONIAM Conceptual Schema

OONIAM Conceptual Schema นั้นเป็นแบบจำลองระดับแนวคิดที่ได้พัฒนาจากแบบจำลองระดับแนวคิดในแอมที่เป็นที่นิยมใช้เพราะง่ายและสะดวกในการนำมาใช้งาน อีกทั้งยังสามารถแสดงสัญลักษณ์ของกฎข้อบังคับความต้องการของข้อมูลบนแบบจำลองได้จำนวนมาก ซึ่งเป็นส่วนที่สำคัญที่จะทำให้การออกแบบฐานข้อมูลเป็นไปอย่างสมบูรณ์ สำหรับคำว่า OONIAM นั้นย่อมาจาก Object Oriented Nijssen's Information Analysis Methodology แนวความคิดที่ได้ทำการสร้างแบบจำลองระดับแนวคิด OONIAM ขึ้นมานั้น เพื่อให้แบบจำลองระดับแนวคิดในแอมมีความสามารถที่จะรองรับคุณสมบัติเชิงวัตถุ ทำให้ผู้ออกแบบที่มีความเข้าใจในการออกแบบฐานข้อมูลเชิงสัมพันธ์สามารถทำการออกแบบฐานข้อมูลเชิงวัตถุได้สะดวกมากยิ่งขึ้น โดยไม่ต้องทำการศึกษาเพิ่มเติมมากนัก รวมไปถึงการปรับปรุงฐานข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุจะทำให้สะดวกมากขึ้นด้วย

3.1 ส่วนประกอบและสัญลักษณ์ที่ใช้ใน OONIAM Conceptual Schema

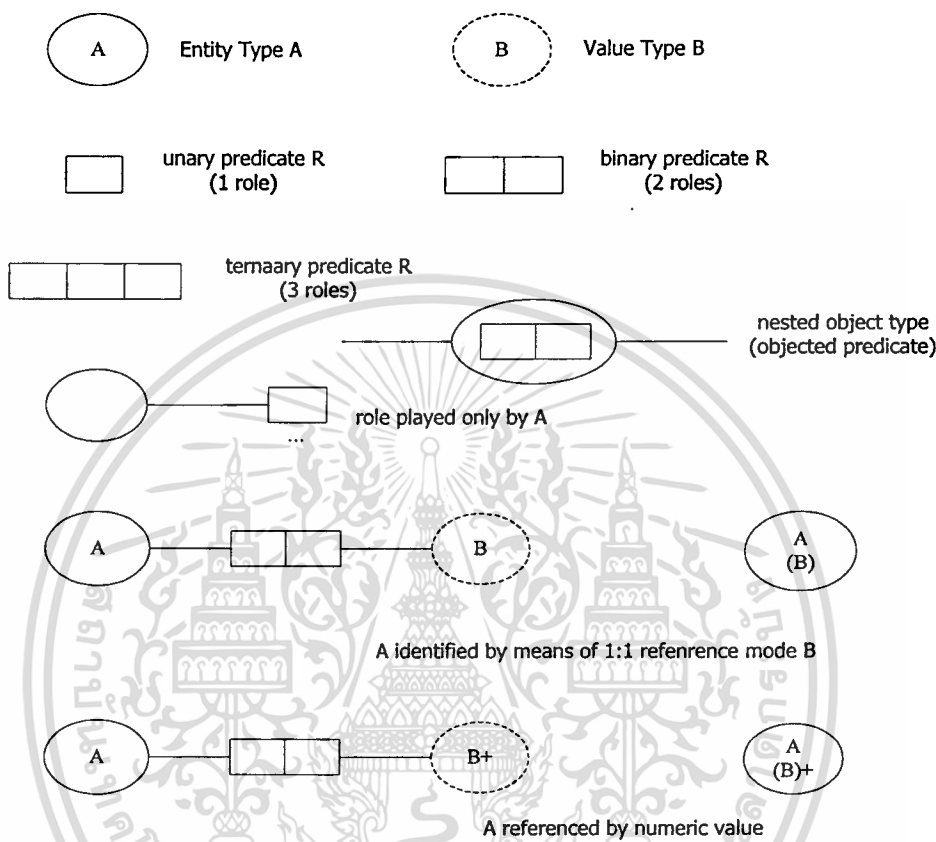
ส่วนประกอบและสัญลักษณ์ที่ใช้ในแบบจำลองระดับแนวคิด OONIAM นี้ แบ่งออกเป็น 2 กลุ่ม คือ

3.1.1 ส่วนประกอบและสัญลักษณ์ที่นำมาจากแบบจำลองระดับแนวคิดในแอม

ประกอบไปด้วยส่วนประกอบที่มีใช้ในแบบจำลองระดับแนวคิดในแอมทั้งหมด โดยส่วนประกอบมีดังนี้

1. ชนิดเอนทิตี (Entity Type)
2. ชนิดเลเบล (Label Type, Value Type)
3. บทบาท (Role)
4. ประโยคความจริงมูลฐาน (Elementary Fact Type) หรืออาจเรียกว่าชนิดความจริง (Fact Type)
5. ชนิดอ้างอิง (Reference Type)
6. ชนิดความจริงแบบเนส (Nested Fact Type)
7. กฎข้อบังคับความต้องการของข้อมูล (Integrity Constraints)

สัญลักษณ์ที่ใช้นำมาจากแบบจำลองระดับแนวคิดในแอมรวมไปถึงสัญลักษณ์ที่แสดงถึงกฎข้อบังคับความต้องการของข้อมูล ที่สามารถปรากฏอยู่บนแบบจำลองระดับแนวคิดในแอมด้วย สำหรับสัญลักษณ์พื้นฐานที่ใช้งานนั้นสามารถแสดงได้ในรูปที่ 3.1



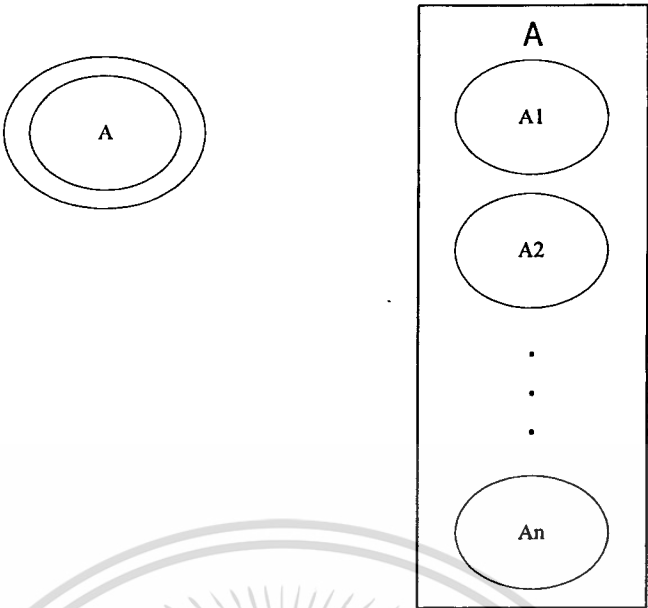
รูปที่ 3.1 ภาพแสดงสัญลักษณ์พื้นฐานที่นำมาจากแบบจำลองระดับแนวคิดในแอม

3.1.2 ส่วนประกอบและสัญลักษณ์เพิ่มเติมที่ใช้ในแบบจำลองระดับแนวคิด OONIAM

สำหรับส่วนประกอบและสัญลักษณ์ที่ทำการเพิ่มเติม เพื่อรองรับในการแสดงคุณสมบัติเชิงวัตถุในที่นี้ได้มีการกำหนดรูปแบบเพิ่มเติมขึ้น 3 รูปแบบดังนี้

1. ชนิดคอมเพลกซ์ออบเจกต์ (Complex Objects Type)

หมายถึงการรวมกันของชนิดเอนติตี้ย่อย ๆ จำนวนมากกว่า 1 เอนติตี้ขึ้นไปที่มีความสัมพันธ์ในลักษณะเดียวกัน โดยประกอบขึ้นเป็นออบเจกต์คลาสหนึ่งคลาส และมีคุณสมบัติของคลาสในแบบจำลองเชิงวัตถุ ดังนั้นอาจจะมีการกำหนดเมธอด (Method) ของออบเจกต์คลาสนั้น ๆ หรือไม่ก็ได้ เช่น ที่อยู่, หน้าคน, ลายนิ้วมือ เป็นต้น สัญลักษณ์ที่ใช้ในการแสดงชนิดคอมเพลกซ์ออบเจกต์สามารถ

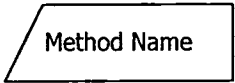


รูปที่ 3.2 แสดงสัญลักษณ์ชนิดคอมเพลกซ์ออบเจกต์

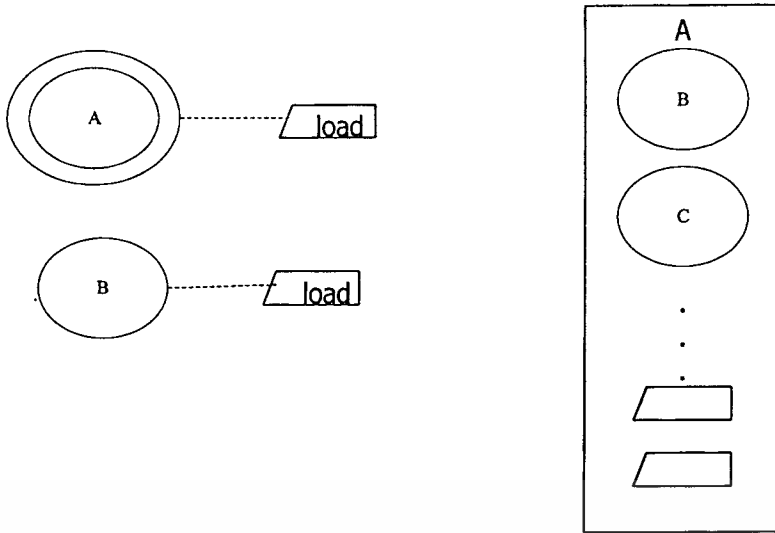
จากภาพข้างต้นเป็นสัญลักษณ์ที่ได้กำหนดเพิ่มเติม เพื่อรองรับกับชนิดคอมเพลกซ์ออบเจกต์ A โดยจะทำการกำหนดรูปแบบของสัญลักษณ์ไว้ 2 ลักษณะทางด้านซ้ายมือเป็นชนิดคอมเพลกซ์ออบเจกต์ที่แสดงในรูปแบบย่อ และภาพทางด้านขวามือเป็นชนิดคอมเพลกซ์ออบเจกต์ในรูปแบบเต็มที่สามารถแสดงชนิดเอนติตี้ภายในทั้งหมดได้

2. เมธอด (Method)

หมายถึงการกระทำใด ๆ ที่เกิดขึ้นกับชนิดเอนติตี้หรือชนิดคอมเพลกซ์ออบเจกต์โดยรวมไปถึง Derived rules ที่ปรากฏอยู่บนแบบจำลองระดับแนวคิดในแอมด้วย ในที่นี้สามารถรองรับกับกฎหรือขั้นตอนต่าง ๆ ที่เกิดขึ้นได้ด้วย โดยสัญลักษณ์ที่ทำการกำหนดไว้ในแบบจำลองจะเป็นกล่องสี่เหลี่ยมที่มีปลายตัดเฉียง ด้านซ้ายมือลักษณะดังในรูปที่ 3.3 และรูปที่ 3.4 จะแสดงให้เห็นถึงการเชื่อมโยงเมธอดกับชนิดเอนติตี้ที่เกี่ยวข้องหรือชนิดคอมเพลกซ์ออบเจกต์โดยใช้เส้นประ



รูปที่ 3.3 ภาพแสดงสัญลักษณ์ของเมธอด



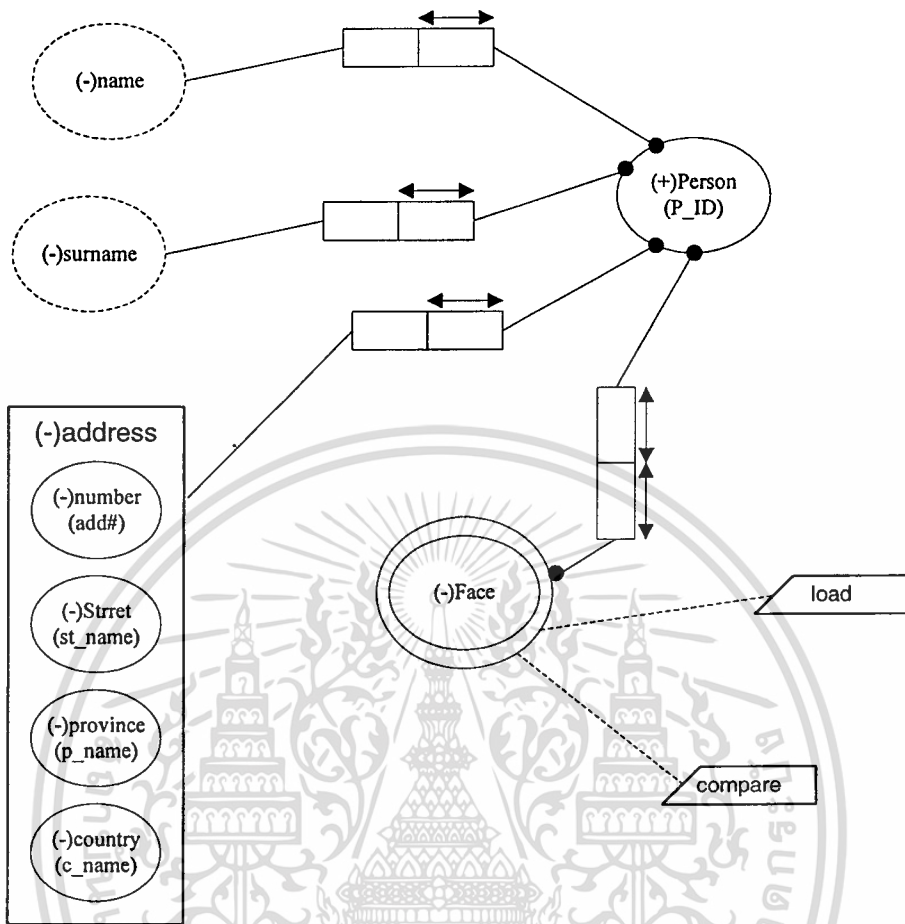
รูปที่ 3.4 แสดงสัญลักษณ์เมธอดที่ใช้ในแบบจำลอง

3. คุณสมบัติของการเอนแคปซูเลชัน (Encapsulation)

ซึ่งในที่นี้จะทำการกำหนดเพิ่มเติมอยู่หน้าชนิดเอนคิตี , ชนิดเลเบล, ชนิดคอมเพลกซ์ออบเจกต์ รวมไปถึงชนิดความจริงแบบเนส ด้วยสัญลักษณ์ดังกล่าวยังปรากฏอยู่หน้าเมธอด เพื่อแสดงคุณสมบัติดังนี้

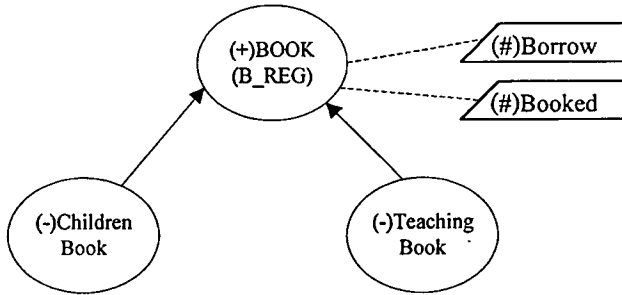
- สัญลักษณ์ (+) (Public) หมายถึงองค์ประกอบที่มีสัญลักษณ์ดังกล่าวจะสามารถให้คลาสอื่น ๆ เข้าถึงและเรียกใช้งาน
- สัญลักษณ์ (-) (Private) หมายถึงองค์ประกอบที่มีสัญลักษณ์ดังกล่าวจะสามารถเรียกใช้หรืออ้างถึงได้เพียงคลาสเดียวเท่านั้น และไม่สามารถอ้างถึงหรือเรียกใช้จากคลาสลูกได้
- สัญลักษณ์ (#) (Protected) หมายถึงองค์ประกอบที่มีสัญลักษณ์ดังกล่าวจะสามารถเรียกใช้หรืออ้างถึงได้เพียงคลาสเดียว แต่อนุญาตให้คลาสลูกสามารถใช้งานได้ด้วย

3.2 ตัวอย่างการใช้สัญลักษณ์ต่าง ๆ ในแบบจำลองระดับแนวคิด OONIAM



รูปที่ 3.5 แสดงตัวอย่างการใช้สัญลักษณ์ในแบบจำลองระดับแนวคิด OONIAM

จากภาพแสดงให้เห็นการใช้สัญลักษณ์ชนิดคอมเพล็กซ์ออบเจกต์ทั้ง 2 ลักษณะอย่างที่ได้กล่าวมาแล้วข้างต้น โดยแบบจำลองดังกล่าวจะสื่อความหมายได้ชัดเจนมากขึ้นเพราะสามารถกำหนดได้ว่า address นั้นเป็นชนิดคอมเพล็กซ์ออบเจกต์โดยประกอบขึ้นจากชนิดเอนตี้ย่อย ๆ เช่น number, street, province และ country เป็นต้น หรืออาจจะใช้สัญลักษณ์ของชนิดคอมเพล็กซ์ออบเจกต์ในลักษณะเดียวกับ Face ก็ได้ ซึ่งสามารถที่จะแสดงเมธอดที่เกี่ยวข้องกับ Face ทั้งหมดบนแบบจำลอง อีกทั้งยังสามารถใช้สัญลักษณ์ของกฎข้อบังคับความถูกต้องของข้อมูลที่เกี่ยวข้องกับข้อมูลชนิดคอมเพล็กซ์ออบเจกต์บนแบบจำลองได้ จากตัวอย่างจะเห็นว่า Face นั้นก็จำเป็นต้องมีการระบุว่าเป็น Face ของใคร และ คนทุกคนก็ต้องมี Face เช่นกัน จึงมีการกำหนด Mandatory constraints กำกับอยู่บนแบบจำลอง



รูปที่ 3.6 ภาพตัวอย่างแสดงการใช้เมรอกกับชนิดเอนติตี้

จากภาพแสดงให้เห็นถึงการใช้เมรอกกับชนิดเอนติตี้ที่มีอยู่ ซึ่งจะช่วยให้ทราบถึงการกระทำที่จะเกิดขึ้นกับชนิดเอนติตี้นั้น ๆ อีกทั้งยังสามารถสืบทอดเมรอกที่มีอยู่ให้กับชั้นคลาสสามารถเรียกใช้งานได้อีกด้วยตามคุณสมบัติเชิงวัตถุ

3.3 อัลกอริทึมในการแปลงแบบจำลองระดับแนวคิด OONIAM ให้อยู่ในรูปของ Object Oriented Definitions Language

จากรูปแบบของแบบจำลองระดับแนวคิด OONIAM ที่ได้กล่าวมาแล้วนั้น จะนำมาทำการแปลงให้เป็น Object Oriented Schemas ได้ สามารถแบ่งขั้นตอนได้ ดังนี้

- การปรับโครงสร้างของ OONIAM Conceptual Schema

1. เปลี่ยนทุก ๆ ความสัมพันธ์แบบ 1 บทบาท (unary elementary fact type) ที่มีทั้งหมดให้อยู่ในลักษณะของความสัมพันธ์แบบ 2 บทบาท (binary fact type)
2. เปลี่ยนทุก ๆ ความสัมพันธ์ที่มี n บทบาท (n -ary fact type) โดยที่ $n > 2$ และมี intra fact type uniqueness constraint กำกับอยู่ $n-1$ บทบาทให้อยู่ในรูปของชนิดความจริงแบบเนส (Nested Fact Type) รวมทั้งกำหนดชื่อของชนิดความจริงดังกล่าวที่สร้างขึ้นด้วย
3. เปลี่ยนทุก ๆ ความสัมพันธ์ที่มี n บทบาท (n -ary fact type) โดยที่ $n > 2$ และมี intra fact type uniqueness constraint กำกับทั้งหมด n บทบาท ให้อยู่ในลักษณะของชนิดความจริงแบบเนส รวมทั้งกำหนดชื่อของชนิดความจริงดังกล่าวที่สร้างขึ้นด้วย

- การสร้างออบเจกต์คลาส (Object Classes)

4. เลือกทุก ๆ ชนิดเอนติตี้ที่มีความสัมพันธ์แบบ 2 บทบาท มากกว่า 1 ความสัมพันธ์ขึ้นไป และมี intra fact type uniqueness constraint แบบหนึ่งต่อหลายหน่วย (one to many) หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบหนึ่งต่อหนึ่งหน่วย (one to one) กำกับอยู่ฝั่งเดียวกับชนิดเอนติตี้ นั้น ๆ มาทำการสร้างเป็นออบเจกต์คลาส โดยนำชื่อของชนิดเอนติตี้ ทำการกำหนดเป็นชื่อของออบเจกต์คลาส

5. นำทุก ๆ ชนิดความจริงแบบเนสมาสร้างเป็นออบเจกต์คลาส โดยใช้ชื่อของชนิดความจริงแบบเนส ที่ได้เตรียมไว้ก่อนหน้านี้มากำหนดเป็นชื่อของออบเจกต์คลาสดังกล่าว
6. นำทุก ๆ กฎข้อบังคับความต้องการของข้อมูลแบบซับไทป์ (SubType Constraints) มาสร้างเป็นออบเจกต์คลาสโดยนำชื่อของชนิดเอนติตี้ ที่เป็นซับไทป์มาสร้างเป็นชื่อของออบเจกต์คลาส และกำหนดให้ออบเจกต์คลาสดังกล่าวเป็นซับคลาส (SubClass) ของซูเปอร์ไทป์ (SuperType) ที่ได้สร้างเป็นออบเจกต์คลาสก่อนหน้านี้แล้ว
7. นำทุก ๆ ชนิดคอมเพลกซ์ออบเจกต์มาทำการสร้างเป็นออบเจกต์คลาสโดยนำชื่อของชนิดคอมเพลกซ์ออบเจกต์มาทำการกำหนดเป็นชื่อของออบเจกต์คลาส
8. เลือกทุก ๆ ชนิดเอนติตี้ที่มีความสัมพันธ์แบบ 2 บทบาทกับชนิดคอมเพลกซ์ออบเจกต์และมี intra fact type uniqueness constraint แบบหนึ่งต่อหลายหน่วย หรือแบบหนึ่งต่อหนึ่งหน่วย กำกับอยู่ฝั่งเดียวกับชนิดเอนติตี้ นั้น ๆ มาทำการสร้างเป็นออบเจกต์คลาส โดยนำชื่อของชนิดเอนติตี้ทำการกำหนดเป็นชื่อของออบเจกต์คลาส

- การสร้างแอตทริบิวต์และความสัมพันธ์ (Relationship) ของออบเจกต์คลาส

9. ทำการรวบรวมชนิดอ้างอิงของแต่ละชนิดเอนติตี้ที่ได้กำหนดเป็นออบเจกต์คลาสไว้ในขั้นตอนที่ 4 มาทำการสร้างแอตทริบิวต์โดยนำชื่อของชนิดเอนติตี้ที่อยู่อีกด้านหนึ่งของความสัมพันธ์มาทำการสร้างเป็นชื่อของแอตทริบิวต์ของออบเจกต์คลาสนั้น ๆ หากชนิดอ้างอิงใด ๆ มี intra fact type uniqueness constraints แบบหนึ่งต่อหนึ่งหน่วยกำกับอยู่ ให้ทำการกำหนดแอตทริบิวต์นั้นเป็นคีย์ (key) ของออบเจกต์คลาสนั้น ด้วย
10. จากทุก ๆ ชนิดเอนติตี้ ที่กำหนดเป็นออบเจกต์คลาสในขั้นตอนที่ 4 ไว้แล้วนั้น ในที่นี้จะนำ unique identifier ของชนิดเอนติตี้ นั้นมาทำการกำหนดเป็นแอตทริบิวต์หรือไม่ก็ได้ หากต้องการทำการกำหนดให้นำ unique identifier ของชนิดเอนติตี้ นั้น ๆ มาสร้างเป็นแอตทริบิวต์ของออบเจกต์คลาส และกำหนดให้แอตทริบิวต์ที่สร้างขึ้นเป็นคีย์ของออบเจกต์คลาสดังกล่าวด้วย
11. ทำการรวบรวมทุก ๆ ชนิดเอนติตี้ ที่มีความสัมพันธ์แบบ 2 บทบาทกับชนิดเอนติตี้ ที่มีการสร้างเป็นออบเจกต์คลาสไว้แล้ว และมี intra fact type uniqueness constraints แบบหนึ่งต่อหลายหน่วย หรือแบบหนึ่งต่อหนึ่งหน่วยกำกับอยู่ฝั่งเดียวกับออบเจกต์คลาสนั้น
12. นำทุก ๆ ชนิดเอนติตี้ ที่รวบรวมได้ในขั้นตอนที่ 10 มาทำการสร้างเป็นแอตทริบิวต์ โดยถ้าชนิดเอนติตี้ใด ไม่ได้ถูกสร้างเป็นออบเจกต์คลาสไว้ ให้ใช้ชื่อของชนิดเอนติตี้ นั้นมาสร้างเป็นชื่อแอตทริบิวต์ของออบเจกต์คลาส ในกรณีที่ชนิดเอนติตี้ใด ได้ทำการสร้างเป็น

ออบเจกต์คลาสไว้แล้ว ให้สร้าง relationship ซึ่งไปยังออบเจกต์คลาสดังกล่าว และถ้าหากมี

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วยให้นำแอตทริบิวต์หรือ relationship ที่ได้สร้างขึ้นไว้มาทำการกำหนดเป็นคีย์ของออบเจกต์คลาสนั้นด้วย

13. ทำการรวบรวมทุก ๆ ชนิดเอนติตี้ที่มีความสัมพันธ์แบบ 2 บทบาทและมี intra fact type uniqueness constraints แบบหลายต่อหลายหน่วย (many to many)
14. นำทุก ๆ ชนิดเอนติตี้ ที่รวบรวมได้ในขั้นตอนที่ 13 มาทำการสร้างเป็นแอตทริบิวต์แบบเซต ถ้าชนิดเอนติตี้ นั้นไม่ได้ถูกสร้างเป็นออบเจกต์คลาสไว้แล้วให้ใช้ชื่อของชนิดเอนติตี้ มาสร้างเป็นชื่อแอตทริบิวต์ของออบเจกต์คลาส หรือในอีกลักษณะหนึ่ง ถ้าชนิดเอนติตี้ นั้นได้ทำการสร้างเป็นออบเจกต์คลาสไว้แล้ว ให้สร้าง relationship แบบเซตชี้ไปยังออบเจกต์คลาส ดังกล่าว
15. นำชนิดความจริงแบบเนสที่ได้ทำการสร้างเป็นออบเจกต์คลาสไว้แล้วในขั้นตอนที่ 5 มาทำการพิจารณาชนิดเอนติตี้ที่อยู่ภายในชนิดความจริงแบบเนส ถ้าชนิดเอนติตี้ใด ๆ ไม่ได้ถูกสร้างเป็นออบเจกต์คลาสไว้ให้นำมาสร้างเป็นแอตทริบิวต์ โดยใช้ชื่อของชนิดเอนติตี้ นั้นมาสร้างเป็นชื่อแอตทริบิวต์ของออบเจกต์คลาส และถ้าชนิดเอนติตี้ใด ๆ ได้ทำการสร้างเป็นออบเจกต์คลาสไว้แล้วให้นำมาสร้าง relationship ชี้ไปยังออบเจกต์คลาสนั้น และทำการรวมทุก ๆ ชนิดเอนติตี้ที่อยู่ภายในชนิดความจริงแบบเนสมาทำการกำหนด เป็นคีย์ของออบเจกต์คลาสดังกล่าวด้วย
16. นำทุก ๆ ชนิดเอนติตี้ ที่เป็นส่วนประกอบอยู่ในชนิดคอมเพลกซ์ออบเจกต์มาทำการสร้างเป็นแอตทริบิวต์ของออบเจกต์คลาสที่ได้สร้างไว้ในขั้นตอนที่ 7 โดยนำชื่อของชนิดเอนติตี้ นั้นมาสร้างเป็นชื่อแอตทริบิวต์ของออบเจกต์คลาส
17. นำทุก ๆ ชนิดเอนติตี้ที่ได้ทำการสร้างเป็นออบเจกต์คลาสไว้แล้วในขั้นตอนที่ 8 โดยนำชื่อของชนิดคอมเพลกซ์ออบเจกต์ มาสร้างเป็นชื่อแอตทริบิวต์ของออบเจกต์คลาสนั้น และถ้าหากมีความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วยให้นำแอตทริบิวต์ที่ได้สร้างขึ้นไว้มาทำการกำหนดเป็นคีย์ของออบเจกต์คลาสนั้นด้วย
18. รวบรวมทุก ๆ ชนิดเอนติตี้ หรือชนิดเลบิลที่มี inter fact type uniqueness constraints เชื่อมโยงอยู่มาทำการสร้างเป็นคีย์ ของออบเจกต์คลาส ที่อยู่ฝั่งตรงข้ามของความสัมพันธ์

- การกำหนดชนิดของแอตทริบิวต์ (Type of attributes)

19. การพิจารณาชนิดของแอตทริบิวต์ที่ได้สร้างไว้แล้ว จะทำการพิจารณาตาม Entity Type constraints ที่ได้กำหนดไว้บนชนิดเอนติตี้ดังกล่าว ถ้า Entity Type constraints เป็นการกำหนดแบบเซต ให้ทำการกำหนดชนิดของแอตทริบิวต์ในลักษณะของ enum ของเซตนั้น ถ้า Entity Type constraints เป็นการกำหนดรูปแบบของชนิดเอนติตี้แบบใด ให้ทำการกำหนดรูปแบบของแอตทริบิวต์ ตามรูปแบบที่ได้กำหนดไว้ในกฎข้อบังคับความถูกต้องของข้อมูลเช่น <a20> ทำการกำหนดเป็นชนิดแอตทริบิวต์แบบ string เป็นต้น

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อใช้ในการอ้างอิงข้อมูลเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20. ถ้ามี Occurrence frequency constraints ปรากฏอยู่ให้ทำการระบุชนิดของแอตทริบิวต์ เป็นลักษณะของแอตทริบิวต์ตามจำนวนที่ระบุในกฎข้อบังคับความถูกต้อง
 21. หากแอตทริบิวต์ใด ๆ ทำการสร้างมาจากชนิดคอมเพลกซ์ออบเจกต์ ให้ทำการกำหนดชนิดของออบเจกต์นั้นเป็นออบเจกต์คลาสที่ได้ทำการสร้างมาจากชนิดคอมเพลกซ์ออบเจกต์ดังกล่าว แต่ถ้าหากชนิดคอมเพลกซ์ออบเจกต์ดังกล่าวไม่มีความสัมพันธ์กับออบเจกต์อื่นที่ได้ทำการสร้างไว้แล้วให้นำออบเจกต์คลาสนั้นมาทำการสร้างเป็น Embed Object Class ในออบเจกต์คลาสหลักหลักแทน
 22. ทำการกำหนดสิทธิการเข้าถึงแอตทริบิวต์ทุกแอตทริบิวต์ตามสัญลักษณ์แสดงคุณสมบัติการเอนแคปซูเลชันที่ปรากฏอยู่บนหน้าชนิดเอนติตี้หรือชนิดเลเวลนั้น ๆ
- การสร้างเมธอด (Method) ของออบเจกต์คลาส
23. นำกฎข้อบังคับความถูกต้องของข้อมูลที่เกี่ยวข้องกับชนิดเอนติตี้หรือชนิดคอมเพลกซ์ออบเจกต์ที่ได้ทำการสร้างเป็นออบเจกต์คลาส มาทำการสร้างเป็นเมธอดของออบเจกต์คลาสดังกล่าว หากกฎข้อบังคับความถูกต้องของข้อมูลนั้นเกี่ยวข้องกับชนิดเอนติตี้ที่ได้ทำการสร้างเป็นแอตทริบิวต์ให้นำกฎดังกล่าวมาทำการสร้างเป็นเมธอดของออบเจกต์คลาสที่มีแอตทริบิวต์นั้นเป็นส่วนประกอบ โดยการสร้างเมธอดสามารถแสดงได้ตามอัลกอริทึมที่จะกล่าวในหัวข้อต่อไป
 24. นำเมธอดที่ปรากฏอยู่บนแบบจำลองระดับแนวคิด OONIAM มาทำการสร้างเป็นเมธอดของออบเจกต์คลาสดังกล่าว และกำหนดสิทธิการเข้าถึงเมธอดนั้น ๆ ตามสัญลักษณ์แสดงคุณสมบัติการเอนแคปซูเลชันที่ปรากฏอยู่บนหน้าเมธอดนั้น ๆ ด้วย

3.4 อัลกอริทึมของเมธอดที่ทำการแปลงมาจากกฎข้อบังคับความถูกต้องของข้อมูล

ในการสร้างเมธอดที่ทำการแปลงมาจากกฎข้อบังคับความถูกต้องของข้อมูลนั้น สามารถทำการสร้างได้โดยใช้อัลกอริทึมในการแปลงดังนี้

กำหนดให้ A, B คือสิ่งที่ต้องการตรวจสอบ เช่น ออบเจกต์, relationship, แอตทริบิวต์ หรือคอมเพลกซ์ออบเจกต์

1. Mandatory role constraints

If (B is null or space) then

Return error

2. Inclusion mandatory role constraints

If (value of all of B are null or space) then

Return error

เอกสาร 3.5 Subset constraints สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ B is subset of A

If (the specific related attribute B is null) then

Return error

4. Equality constraints

ถ้าต้องการตรวจสอบ Equality constraints ระหว่าง A และ B

if (B is null and A is not null) or (B is not null and A is null) then

Return error

5. Exclusion constraints

ถ้าต้องการตรวจสอบ Exclusion constraints ระหว่าง A และ B

if (B is not null and A is not null) then

Return error

6. Occurrence frequency constraints

กำหนดให้ B ต้องการตรวจสอบ Occurrence frequency constraints ในลักษณะ n....m

If (number of B is less than n) or (number of B is more than m) then

Return error

กำหนดให้ B ต้องการตรวจสอบ Occurrence frequency constraints ในลักษณะมากกว่า n

If (number of B is less than n) then

Return error

7. Irreflexive constraints

ถ้าต้องการตรวจสอบ irreflexive constraints ของ A

if (A relate to A) then

Return error

8. Asymmetric constraints

ถ้าต้องการตรวจสอบ asymmetric constraints ของ A และ B

if (A relate to B) and (B relate to A) then

Return error

9. Antisymmetric constraints

ถ้าต้องการตรวจสอบ antisymmetric constraint ของ A และ B

if (A relate to B) and (B relate to A) and (A not equal B) then

Return error

10. Intransitive constraints

ถ้าต้องการตรวจสอบ Intransitive constraints ของ A และ B

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อใช้ในการศึกษาและวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

if (A relate to C) and (C relate to B) and (A relate to B) then

Return error

11. Acyclic ring constraints

ถ้าต้องการตรวจสอบ Acyclic ring constraints ของ A และ B

if (A relate to C) and (C relate to B) and (B relate to A) then

Return error

12. Cardinality constraints

if (number of objects is more than n) then

Return error

13. Relative closure constraints

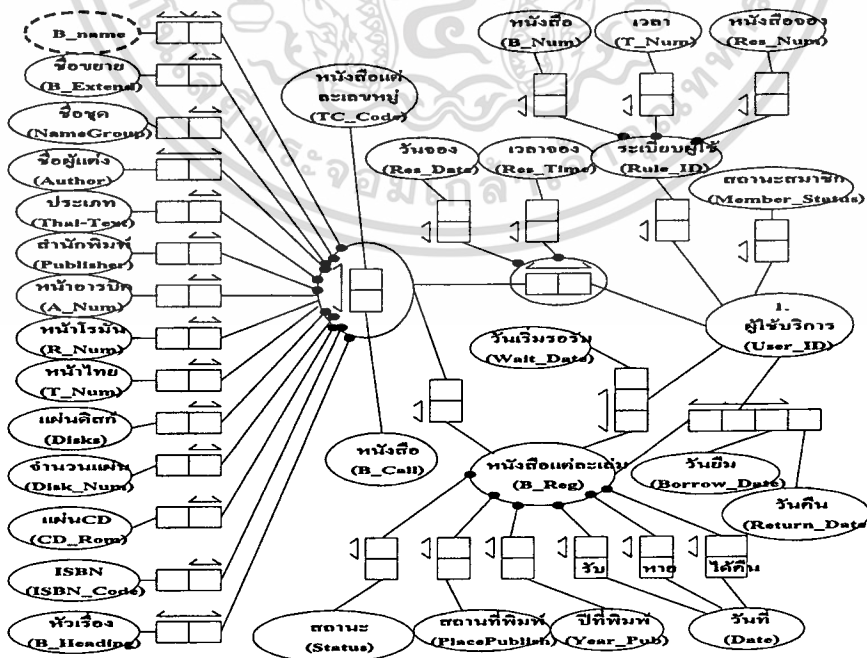
If (value of attribute is found) then

return true

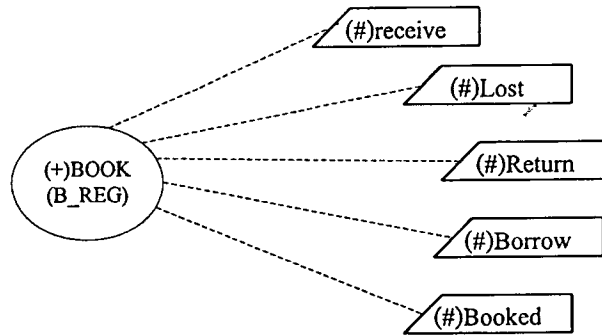
else

return false

3.5 ตัวอย่างการใช้แบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM ในการออกแบบฐานข้อมูลเชิงวัตถุ

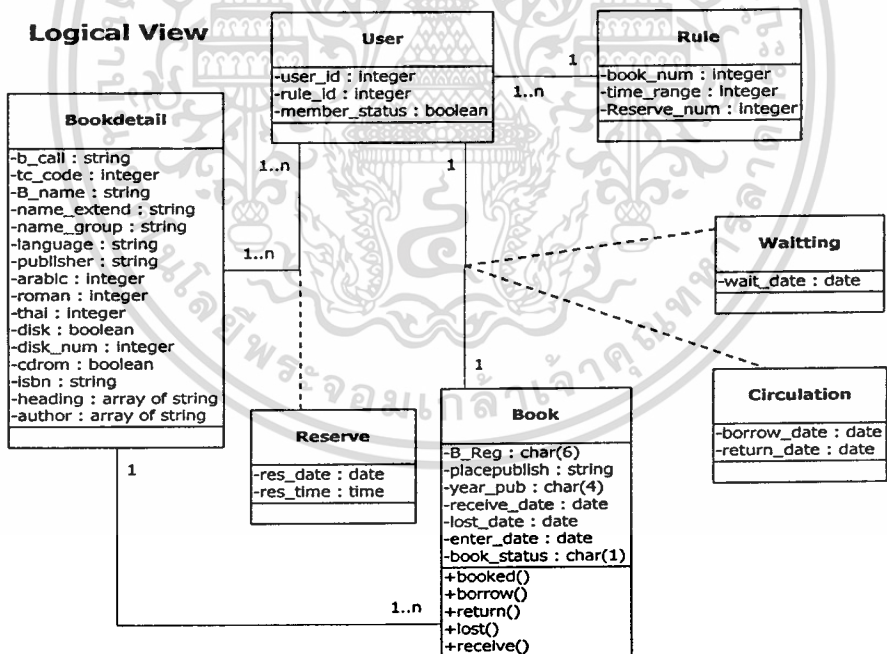


เอกสารที่ 3.7 ตัวอย่างแบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

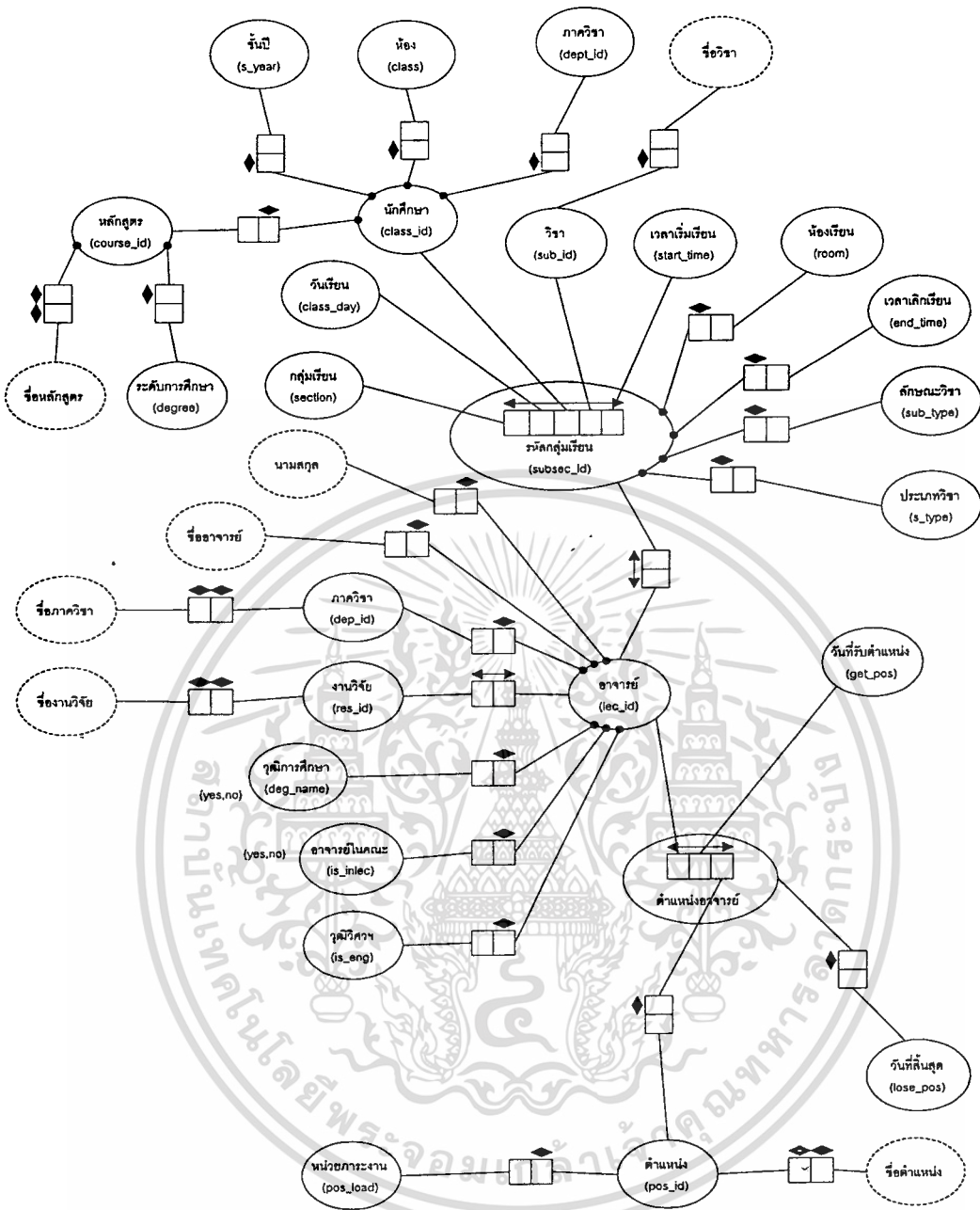


รูปที่ 3.8 ตัวอย่างการสร้างเมธอดใน OONIAM

จากรูปที่ 3.7 ตัวอย่างของแบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM ข้างต้นและรูปที่ 3.8 สามารถใช้อัลกอริทึมดังกล่าวมาทำการแปลงรูปให้เป็นฐานข้อมูลเชิงวัตถุ โดยสามารถเขียนอยู่ในลักษณะของ Class Diagram ได้ดังนี้



รูปที่ 3.9 แสดงผลลัพธ์ที่ได้จากการแปลงให้อยู่ในรูปของคลาสโคเอเกรม



รูปที่ 3.10 ตัวอย่างแบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM

จากรูปที่ 3. 10 ตัวอย่างของแบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM ข้างต้นสามารถนำอัลกอริธึมดังกล่าวมาทำการแปลงรูปให้เป็นฐานข้อมูลเชิงวัตถุ โดยสามารถเขียนอยู่ในลักษณะของ Object Definition Language ได้ดังนี้

```
interface Course {
    (key id, name)
```

```
attribute int id;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        attribute    string    name;

        attribute    string    degree;

};

interface    Subject {

    (key    id)

        attribute    char(8) id;

        attribute    string    name;

};

interface Department {

    (key    id, name)

        attribute    int    id;

        attribute    string    name;

};

interface Research {

    (key    id, name)

        attribute    int    id;

        attribute    string    name;

        relationship    Set(Lecturer)    memberin inverse Lecturer :: memberof;

};

interface Position {

    (key    id, name)

        attribute    int    id;

        attribute    string    name;

        attribute    int    load;

};

interface Student {

    (key    id)

        attribute    int    id;

        attribute    int    year;

        attribute    chat(2) class;

        relationship    Department depln;

        relationship    Course courseIn;

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

);

interface Lecturer {
    (key id, (name,surname))
    attribute      char(6) id;
    attribute      string  name;
    attribute      string  surname;
    attribute      string  deg_name;
    attribute      boolean is_inlec;
    attribute      boolean is_eng;
    relationship    Department  depin;
    relationship    Set(Research) memberof inverse Research :: memberin;
    relationship    Set(SubjectSection) teachin inverse SubjectSection ::
Lecin;
};

interface PositionDetail {
    (key (Lecin,get_pos,pos_in))
    relationship    Lecturer    Lecin;
    attribute      date          get_pos;
    relationship    Position     pos_in;
    attribute      date          lose_pos;
};

interface SubjectSection {
    (key (Std_in,Sub_in,section,class_date,start_time))
    relationship    Student      Std_in;
    relationship    Subject      Sub_in;
    attribute      int           section;
    attribute      date          class_date;
    attribute      string        start_time;
    attribute      string        room;
    attribute      string        end_time;
    attribute      string        sub_type;
    attribute      string        s_type;
};

```

```

relationship    Set(Lecturer)    lec_in inverse Lecturer :: teachin;
};

```



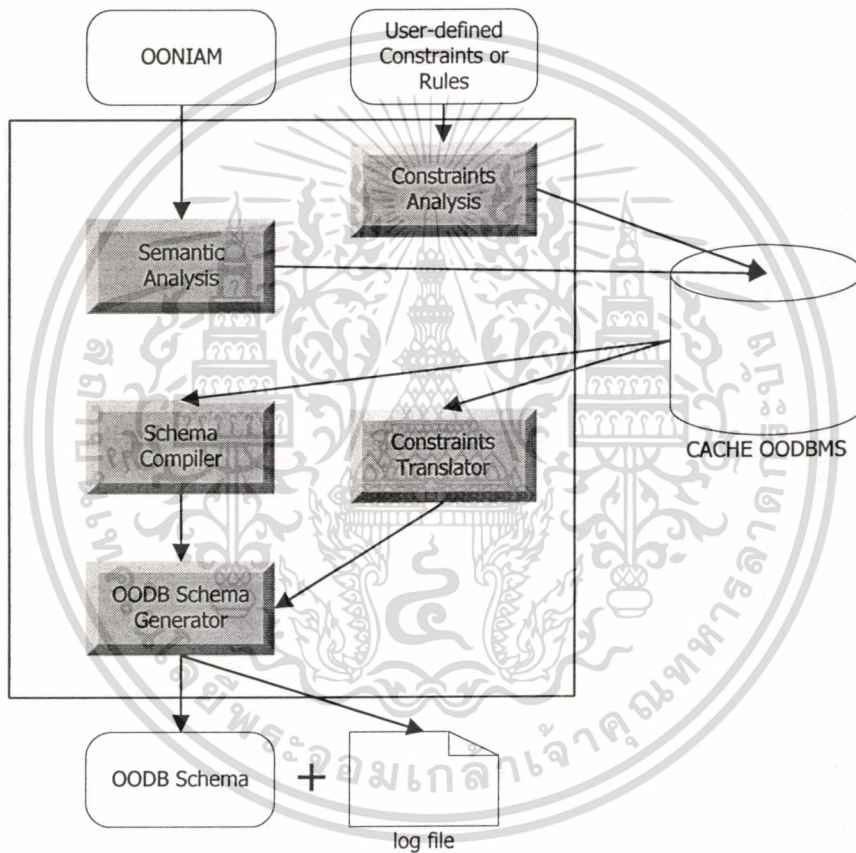
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

สถาปัตยกรรมของระบบ

4.1 สถาปัตยกรรมของระบบ

ในระบบที่ได้ทำการพัฒนาขึ้นมาเพื่อรองรับกับแบบจำลองระดับแนวคิด OONIAM นี้มีโครงสร้างของสถาปัตยกรรมของระบบดังภาพในรูปที่ 4.1



รูปที่ 4.1 สถาปัตยกรรมของระบบ OONIAM

จากภาพดังกล่าวสถาปัตยกรรมของระบบได้แบ่งออกเป็น 5 ขั้นตอนดังนี้

1. Semantic Analysis จะทำการรับข้อมูลจากแบบจำลองระดับแนวคิด OONIAM มาทำการแยกออกเป็น 2 ส่วนคือในส่วนที่เกี่ยวข้องกับโครงสร้างของแบบจำลอง (Constructs) เช่น ชนิดเอนติตี้, ชนิดเลเบล, ชนิดอ้างอิง เป็นต้น อีกส่วนหนึ่งเป็นส่วนของกฎข้อบังคับความถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของข้อมูล (Constraints) ที่ปรากฏอยู่บนแบบจำลอง และทำการจัดเก็บข้อมูลที่ได้ลงบนระบบจัดการฐานข้อมูลเชิงวัตถุ

2. Constraints Analysis จะทำการรับข้อมูลของกฎข้อบังคับความต้องการของข้อมูลที่ไม่สามารถแสดงรายละเอียดบนแบบจำลองได้ และรวมไปถึงขั้นตอนและเงื่อนไขต่าง ๆ ของการทำงานของระบบที่ได้รวบรวมไว้เป็นกฎ ซึ่งจะแสดงอยู่บนแบบจำลองระดับแนวคิดในลักษณะของ method มาทำการวิเคราะห์และจัดเก็บลงบนระบบจัดการฐานข้อมูลเชิงวัตถุ
3. Schema Compiler จะทำหน้าที่ในการนำข้อมูลที่เกี่ยวข้องกับโครงสร้างที่ได้ทำการจัดเก็บไว้ในระบบจัดการฐานข้อมูลเชิงวัตถุมาทำการแปลจากโครงสร้างดังกล่าวให้อยู่ในลักษณะของโครงสร้างของแบบจำลองเชิงวัตถุ
4. Constraints Translator จะทำหน้าที่ในการนำข้อมูลเกี่ยวกับกฎข้อบังคับความต้องการของข้อมูลที่สามารถแสดงอยู่บนแบบจำลองได้และทำการจัดเก็บในระบบจัดการฐานข้อมูลเชิงวัตถุ และข้อมูลของกฎข้อบังคับความต้องการหรือเงื่อนไขที่ผู้ใช้ทำการกำหนดขึ้น โดยมีได้แสดงอยู่บนแบบจำลองระดับแนวคิด OONIAM โดยข้อมูลดังกล่าวทั้งหมดจะนำมาจากระบบจัดการฐานข้อมูลเชิงวัตถุที่ได้จัดเก็บไว้แล้วมาทำการแปลงเป็นเมธอดของแบบจำลองเชิงวัตถุ
5. OODB Schema Generator จะทำหน้าที่ในการนำข้อมูลที่ได้จากขั้นตอน Schema Compiler และ Constraints Translator มาทำสร้างแบบจำลองระดับแนวคิดเชิงวัตถุในลักษณะของ Class Diagram หรือสร้างเป็นภาษาที่สามารถนำมาใช้ในการสร้างฐานข้อมูลเชิงวัตถุบนระบบจัดการเชิงวัตถุ (Object Oriented Definition Language : OODL) ได้ และจะทำการสร้าง log file เพื่อบันทึกผลที่เกิดขึ้นจากการทำงานดังกล่าวด้วย

4.2 เครื่องมือที่ใช้ในการพัฒนาระบบ

จากสถาปัตยกรรมของระบบที่ได้ทำการออกแบบไว้นั้น มีการใช้เครื่องมือที่ใช้ในการพัฒนาระบบโดยแบ่งออกเป็น 2 ส่วน คือ

1. ส่วนที่ใช้ในการพัฒนาโปรแกรมในส่วนติดต่อกับผู้ใช้คือ Visual Basic
2. ส่วนที่ใช้ในการจัดเก็บข้อมูลและพัฒนาโปรแกรมในส่วนที่เกี่ยวข้องกับฐานข้อมูล คือ ระบบจัดการฐานข้อมูลเชิงวัตถุ Cache

โดยในแต่ละส่วนจะมีรายละเอียดดังนี้

4.2.1 Visual Basic

โปรแกรม Visual Basic ที่ได้ทำการเลือกมาใช้ในส่วนของการพัฒนาโปรแกรมของระบบอัตโนมัติเพื่อออกแบบ OONIAM นี้ เนื่องจาก เป็นการเขียนโปรแกรมด้วยโครงภาษาเบสิกในเชิงวัตถุ และโปรแกรมหักกล่าวสามารถใช้งานติดต่อกับระบบจัดการฐานข้อมูลเชิงวัตถุที่ได้ทำการนำมาใช้ในส่วนของการจัดเก็บข้อมูลเชิงวัตถุได้สะดวกอีกด้วย

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

การนำโปรแกรมดังกล่าวมาช่วยให้การพัฒนาระบบสามารถสนับสนุนแนวคิดการเขียนโปรแกรมเชิงวัตถุได้เป็นอย่างดี

1. การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)

การเขียนโปรแกรมเชิงวัตถุเป็นวิธีการใหม่ที่ใช้ในการพัฒนาซอฟต์แวร์ ส่วนประกอบที่สำคัญของการเขียนโปรแกรมเชิงวัตถุคือ แอ็บสแตรกคดาต้าไทป์ (abstract data type), คลาส, การจัดลำดับของคลาส, อินเฮริเทนส์ (inheritance) และ โพลิมอร์ฟิซึม (polymorphism) ในการเขียนโปรแกรมเชิงวัตถุนั้นจะต้องทำการศึกษาถึงคุณสมบัติของออบเจกต์โอเรียลเตด (Object Oriented Prototies) 4 ประการดังนี้

- แอ็บสแตรกชัน (Abstraction)

คือการแทนความคิดที่ยุ่งยากซับซ้อนให้สั้นกระชับ กระทัดรัดขึ้น โดยใช้ ออบเจกต์ซึ่งจะทำการรวมทั้งในส่วนข้อมูล (data) และโอเปอเรชัน (operations) เข้าไว้ด้วยกัน หรืออาจกล่าวได้ว่าออบเจกต์เป็น encapsulation of abstractions การกำหนดแอ็บสแตรกชันใหม่เพื่อใช้ในการแก้ปัญหาเชิงวัตถุ นั้น จะต้องทำการกำหนดออบเจกต์ใหม่โดยแต่ละออบเจกต์จะมีการกำหนด ลักษณะของคุณสมบัติ (properties feature) และเมธอดที่จะทำการติดต่อ และจัดการเกี่ยวกับออบเจกต์นั้น ๆ

สำหรับในแนวทางการแก้ปัญหาหนึ่ง ๆ นั้น อาจจะประกอบไปด้วยหลาย ออบเจกต์ โดยที่บางออบเจกต์มีลักษณะที่คล้ายกัน ซึ่งเราจะรวมเรียกกลุ่มของออบเจกต์ที่มีลักษณะคล้ายกันนี้ว่า คลาส โดยคุณสมบัติพื้นฐานของ ออบเจกต์จะถูกกำหนดไว้ในคลาส โดยแต่ละออบเจกต์จะเรียกได้ว่าเป็น อินสแตนซ์ของคลาส และพฤติกรรมของออบเจกต์จะถูกกำหนดไว้ในกลุ่มของคุณสมบัติของคลาส

- เอ็นแคปซูลชัน (Encapsulation)

คือการกำหนดการเข้าถึงส่วนประกอบของคลาส ซึ่งรวมทั้งโครงสร้าง ข้อมูล (data structure) และฟังก์ชันที่เกี่ยวข้อง ลักษณะของออบเจกต์ เอ็นแคปซูลชันมีดังนี้

1. Private Data หมายถึงแต่ละออบเจกต์จะกำหนดพารามิเตอร์ของแต่ละอินสแตนซ์ในคลาสซึ่งออบเจกต์ทั้งหมดที่เป็นอินสแตนซ์ในคลาสจะมีพารามิเตอร์นี้แต่จะมีค่าแตกต่างกัน และออบเจกต์หนึ่ง ๆ จะสามารถเข้าถึง (access) ข้อมูลประเภทนี้ของออบเจกต์ได้เฉพาะข้อมูลของตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนรูปร่างของโครงสร้างข้อมูล ซึ่งการเปลี่ยนรูปร่างของโครงสร้างข้อมูลนั้นส่งผลทำให้เกิดความสับสนในการกำหนดความสัมพันธ์ของข้อมูลลงไปได้มาก อีกทั้งยังสามารถเพิ่มประสิทธิภาพในด้านความเร็วในการเข้าถึงข้อมูลได้

1. ส่วนประกอบของ Cache

- Cache Object Server มีหน้าที่ในการจัดการเกี่ยวกับความสามารถในการใช้งานเชิงวัตถุของ Cache ได้โดยตรงสามารถแยกออกเป็น Cache Object Server for Java และ Cache Object Server for ActiveX ซึ่งสามารถใช้ติดต่อผ่านทาง Java, ActiveX และ C++ และ เครื่องมือในการพัฒนา GUI อื่น ๆ
- Cache SQL Server เป็นส่วนประกอบที่ทำให้ประสิทธิภาพในการเข้าถึงฐานข้อมูลของ Cache ผ่านทาง SQL และ ODBC ที่ความเร็วสูงได้
- Cache Studio เป็นส่วนประกอบที่รองรับการพัฒนาการเชื่อมต่อกับผู้ใช้เชิงกราฟฟิค (Graphic User Interface :GUI) ให้มีประสิทธิภาพที่สูงในการกำหนดคลาสของออบเจกต์ และการสร้างส่วนประกอบของแอปพลิเคชัน (Applications)
- Cache Distributed Cache Protocol (DCP) เป็นส่วนประกอบที่จัดการเกี่ยวกับเทคโนโลยีประสิทธิภาพสูงทางด้านเครือข่ายที่สามารถกระจายฐานข้อมูลทั่วทั้งระบบเครือข่ายในการตอบสนองต่อความต้องการ
- Cache Object Script เป็นส่วนประกอบที่เกี่ยวข้องกับภาษาในการเขียนโปรแกรมที่มีมาเพื่อให้จุดประสงค์ในด้านตรรกทางธุรกิจและส่วนประกอบของฐานข้อมูลสำหรับแอปพลิเคชันด้านการประมวลผลทรานส์แอ็กชันในระดับองค์กรต่าง ๆ
- Cache Weblink เป็นส่วนประกอบที่รองรับการเชื่อมต่อระหว่าง Cache กับเว็บเซิร์ฟเวอร์ (Web Server) สำหรับแอปพลิเคชันในการประมวลผลทรานส์แอ็กชันผ่านทางเว็บ
- Visual Cache เป็นเครื่องมือที่ใช้ในการสร้างฟอร์มวิซวล เบสิก (Visual Basic Form) ของไมโครซอฟท์ได้อย่างอัตโนมัติและยังให้การเชื่อมต่อระหว่างออบเจกต์ของ Cache กับวิซวล เบสิกได้อย่างอัตโนมัติอีกด้วย

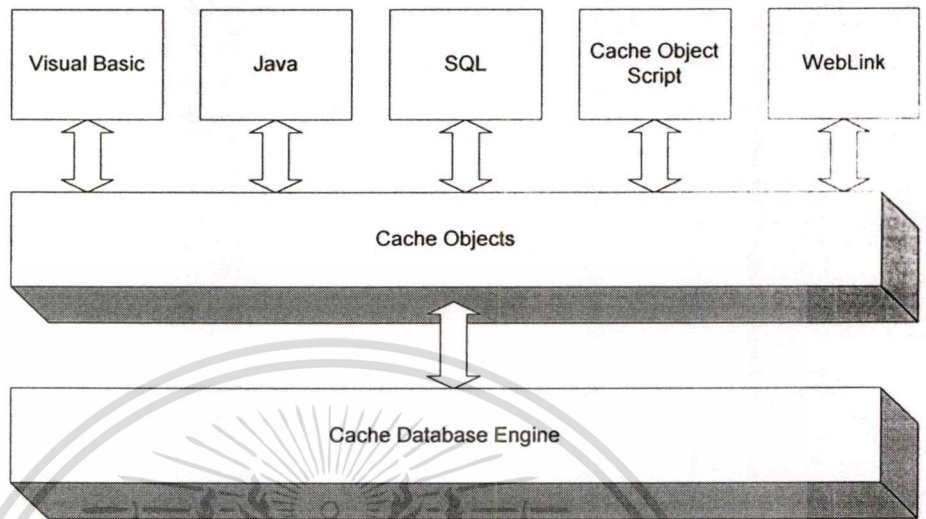
2. สถาปัตยกรรมของ Cache

โครงสร้างของสถาปัตยกรรมของในการติดต่อกับระบบจัดการฐานข้อมูลเชิงวัตถุ Cache ดังแสดงได้ในรูปที่ 4.2 ซึ่งจากภาพนั้นสถาปัตยกรรมในการติดต่อของ Cache แบ่งออกเป็น 3 ระดับ โดยระดับบนสุดจะหมายถึงเครื่องมือที่สามารถใช้ทำการติดต่อกับ Cache ได้ โดยเครื่องมือดังกล่าวมีประกอบไปด้วยเครื่องมือจากภายนอกและเครื่องมือที่มีการสร้างขึ้นเพื่อใช้งานกับ Cache แต่ในเครื่องมือทุกชนิด จะติดต่อกับระบบจัดการฐาน

เอกสารนี้ ข้อมูลโดยผ่านในระดับที่ 2 ซึ่งหมายถึง Cache Objects ที่จะทำให้มีการเข้าถึงข้อมูลได้ในการค้า

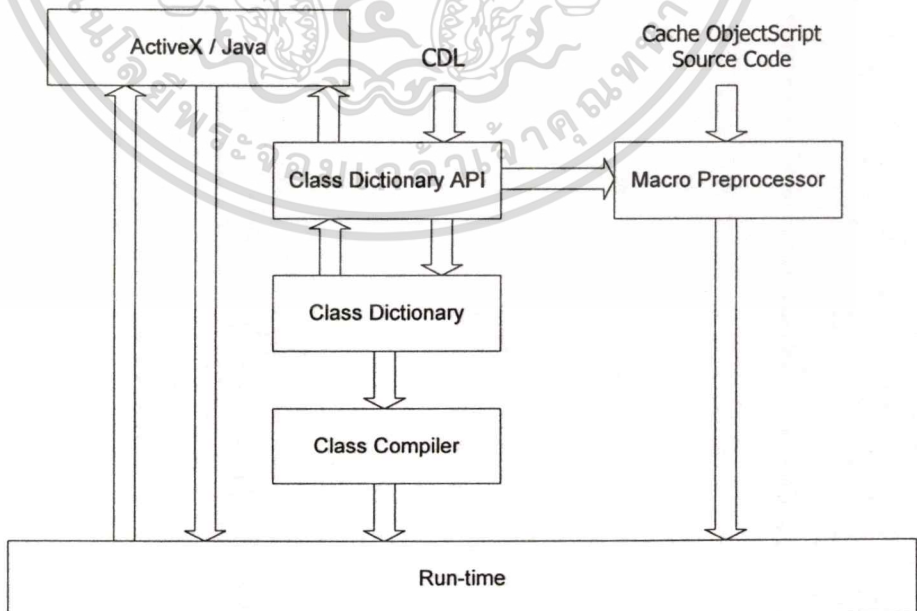
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะมุมมองของ Object Oriented Model ซึ่งสามารถติดต่อได้หลายทาง จากนั้น Cache Objects จะเป็นส่วนที่ติดต่อกับ Cache Database Engine ในระดับสุดท้าย



รูปที่ 4.2 ภาพแสดงสถาปัตยกรรมในการติดต่อของ Cache

สำหรับในส่วนของ Cache Objects นั้นสามารถแสดงรายละเอียดโครงสร้างสถาปัตยกรรมและการใช้งานดังในรูปที่ 4.3



รูปที่ 4.3 ภาพแสดงโครงสร้างและการใช้งานของ Cache Objects

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีทางด้านออบเจกต์ของ Cache จะประกอบไปด้วยระบบย่อย ๆ ดังนี้

- Cache Object Architect เป็นแอปพลิเคชันสำหรับการออกแบบคลาสเพื่อใช้ภายในแอปพลิเคชันและระหว่างแอปพลิเคชันด้วย
- Cache Object Server for ActiveX เป็น ActiveX automation server ซึ่งทำหน้าที่ได้ การเปิดโอกาสให้ Cache Objects เช่น ActiveX Objects ในการใช้เพื่อที่จะกำหนดค่าหรือสืบค้นคุณสมบัติต่าง ๆ ของออบเจกต์รวมไปถึงการใช้เมธอดจากเครื่องมือภายนอกเช่น Visual Basic, C++ และเครื่องมืออื่น ๆ ที่ใช้ในการสร้างส่วนติดต่อกับผู้ใช้
- Cache Object Server for Java เป็นส่วนที่ใช้สำหรับ Java Objects ในการติดต่อและทำการกำหนดค่าหรือสืบค้นคุณสมบัติต่าง ๆ ของออบเจกต์รวมไปถึงการใช้เมธอดจากจาวาแอปพลิเคชัน
- Class Dictionary เป็นส่วนที่จะทำการบรรจุคำจำกัดความของคลาสซึ่งได้ทำการสร้างมาจาก Cache Object application โดยมี Class Dictionary API เป็นตัวกลางในการสื่อสารกับ Class Dictionary ซึ่งผู้ใช้สามารถกำหนดคลาสที่จะใช้งานได้ 2 ทางคือ ทำการสร้างและแก้ไขผ่าน Cache Object Architect หรือใช้ Class Definition Language ได้
- Class Compiler เป็นตัวแปลคำจำกัดความของคลาสให้อยู่ในรูปของแอปพลิเคชันโค้ดที่ใช้งานจริง

3. ภาษาเชิงวัตถุที่ใช้กับ Cache

Cache ได้ทำการสร้างภาษาโปรแกรมมิ่งเชิงวัตถุขึ้นมาใช้งานเรียกว่า Cache Object Script ซึ่งสนับสนุนรูปแบบของการกำหนดแบบจำลองเชิงวัตถุ และทำการกำหนดกลุ่มของคำสั่ง, ฟังก์ชัน, ตัวดำเนินการ และตัวแปรพิเศษ เพื่อให้ผู้ใช้สามารถใช้ในการเขียนเมธอดของออบเจกต์ได้ Cache Object Script ยังประกอบด้วยลักษณะของคอตซินแทกซ์ (dot syntax) ซึ่งผู้ใช้สามารถที่จะกำหนดหรือรับค่าของคุณสมบัติต่าง ๆ และเพื่อจัดการกับเมธอด ยังรวมไปถึงการสร้างออบเจกต์ใหม่และการกระทำของเมธอดของคลาสอีกด้วย Cache Object Script ประกอบไปด้วยองค์ประกอบทางด้านประเภทของภาษาดังนี้

- ถ้อยคำ (Expressions) คือสิ่งซึ่งสามารถที่จะทำการหาค่าได้เพียงค่าเดียวขณะรันไทม์ (runtime) โดยอาจหมายถึง สตริง (string), ตัวแปร (variables), เมธอดซึ่งมีการส่งค่ากลับ, ฟังก์ชันที่มีการส่งค่ากลับ หรือ คุณสมบัติต่าง ๆ ลำดับของถ้อยคำนั้นจะมีการแบ่งออกจากกันโดยตัวดำเนินการ ตัวอย่างเช่น

เอกสารนี้เป็นเอกสาร “Woof, Woof” หรือเป็นการกำหนดค่าแบบสตริง นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Person.Name เป็นการหาค่าของคุณสมบัติ

Map.FindState(x,y) เป็นลักษณะของเมธอดที่มีการส่งค่ากลับ

\$\$SquareRoot(x) เป็นลักษณะของฟังก์ชันที่มีการส่งค่ากลับ

- คำสั่ง (Commands) เป็นคำสั่งพื้นฐานซึ่ง Cache ใช้ในการสร้างการกระทำที่ต้องการเฉพาะ โดยคำสั่งดังกล่าวคือ DO, SET, WRITE, READ, QUIT, NEW, IF, ELSE, FOR ซึ่งคำสั่งจะต้องตามด้วยถ้อยคำอย่างน้อย 1 ถ้อยคำ
- ฟังก์ชัน (Functions) ในที่นี้อาจจะมีการส่งค่ากลับหรือไม่มีการส่งค่ากลับก็ได้ ใน Cache Object Script นั้นสนับสนุนการใช้งานของฟังก์ชันอยู่ 2 ประเภทคือ
 - Intrinsic Functions คือฟังก์ชันที่สนับสนุนการทำงานกับระบบซึ่งผู้ใช้ไม่สามารถแก้ไขได้ การกำหนดฟังก์ชันประเภทนี้จะเริ่มต้นด้วยตัวอักษรคอลดาร์หนึ่งตัวนำหน้าชื่อของฟังก์ชัน (“\$”)
 - Extrinsic Functions คือฟังก์ชันที่ผู้ใช้สามารถกำหนดการทำงานได้ตามความต้องการ การกำหนดฟังก์ชันประเภทนี้จะเริ่มต้นด้วยตัวอักษรคอลดาร์สองตัวนำหน้าชื่อของฟังก์ชัน (“\$\$”)
- ตัวแปร (Variables) จะทำการอ้างอิงข้อมูลที่ถูกจัดเก็บทั้งในหน่วยความจำและดิสก์ โดยตัวแปรใน Cache Object Script นั้นแบ่งออกเป็น 3 ประเภทคือ
 - Special Variables เป็นตัวแปรที่อนุญาตให้ผู้ใช้สามารถเข้าถึงข้อมูลเฉพาะที่เก็บอยู่ใน Cache เช่น วันเวลาปัจจุบัน ตัวแปรชนิดนี้จะสามารถใช้อ่านค่าได้เพียงอย่างเดียว การอ้างอิงตัวแปรจะเริ่มด้วยตัวอักษรคอลดาร์เสมอ (“\$”)
 - Structured System Variables เป็นตัวแปรของระบบซึ่งจะสืบค้นข้อมูลของระบบและข้อมูลการจัดเก็บ การอ้างอิงตัวแปรชนิดนี้จะนำหน้าด้วยตัวอักษร (“^”) เสมอ
 - User-Defined Variables ใน Cache Object Script นั้นจะอนุญาตให้ผู้ใช้สร้างตัวแปรที่กำหนดขึ้นเองเพื่อนำตัวแปรมาใช้งานกับโปรแกรมเฉพาะเจาะจง โดยการกำหนดชื่อตัวแปรจะเริ่มด้วยตัวอักษรใด ๆ หรือนำตัวแปรที่กำหนดมาใช้งานสำหรับทุก ๆ โปรแกรมของ Cache โดยการกำหนดชื่อของตัวแปรจะเริ่มด้วยตัวอักษร (“^”)
- มาโคร(Macros) ผู้ใช้สามารถกำหนดมาโครไว้ใน include files (ไฟล์สกุล INC) และสามารถอ้างอิงมาโครในส่วนต่าง ๆ ได้โดยการอ้างอิงจะมีการใช้ตัวอักษรคอลดาร์ 3 ตัวนำหน้าชื่อของมาโครนั้น ๆ (“\$\$\$Macroname”)
- ตัวดำเนินการ (Operators) ใน Cache Object Script มีการแบ่งตัวดำเนินการออก

เอกสารนี้เป็นเอกสารที่เป็น 3 ประเภทดังนี้ ซึ่งงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)
- ตัวดำเนินการทางสตริง (String Operators)
- ตัวดำเนินการทางตรรกศาสตร์ (Logical Operators)

4.3 อุปกรณ์ที่ใช้ในการพัฒนาระบบ

อุปกรณ์ที่ใช้ในการพัฒนาระบบดังกล่าวมีดังต่อไปนี้

1. เครื่องไมโครคอมพิวเตอร์ที่มีหน่วยประมวลผลกลางสูงกว่า 486 50 mHz
2. หน่วยความจำหลักมากกว่า 32 Mb
3. เนื้อที่ในการจัดเก็บไม่น้อยกว่า 45 Mb
4. ระบบปฏิบัติการ Microsoft Windows 95/98/2000



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบอัตโนมัติสำหรับออกแบบ OONIAM Conceptual Schema

ระบบอัตโนมัติสำหรับออกแบบ OONIAM Conceptual Schema นี้ ได้ทำการพัฒนาขึ้น โดยใช้แนวทางในการออกแบบระบบเชิงวัตถุอาศัยหลักการของ UML (Unified Modeling Language) [6, 7, 8, 9, 10] ซึ่งเป็นมาตรฐานที่ใช้ในการออกแบบระบบเชิงวัตถุ วิธีการดังกล่าวง่าย และสะดวกในการที่จะมีการนำระบบอัตโนมัตินี้มาทำการพัฒนาเพิ่มเติมในส่วนอื่น ๆ ต่อไปเพื่อให้สามารถใช้งานได้ดียิ่งขึ้น ในหลักการของ UML นั้นจะใช้แผนภาพทั้งหมด 9 แบบเพื่อช่วยในการออกแบบ แผนภาพดังกล่าวมีดังนี้

1. Use case diagram
2. Class diagram
3. Object diagram
4. Sequence diagram
5. Collaboration diagram
6. Statechart diagram
7. Activity diagram
8. Component diagram
9. Deployment diagram

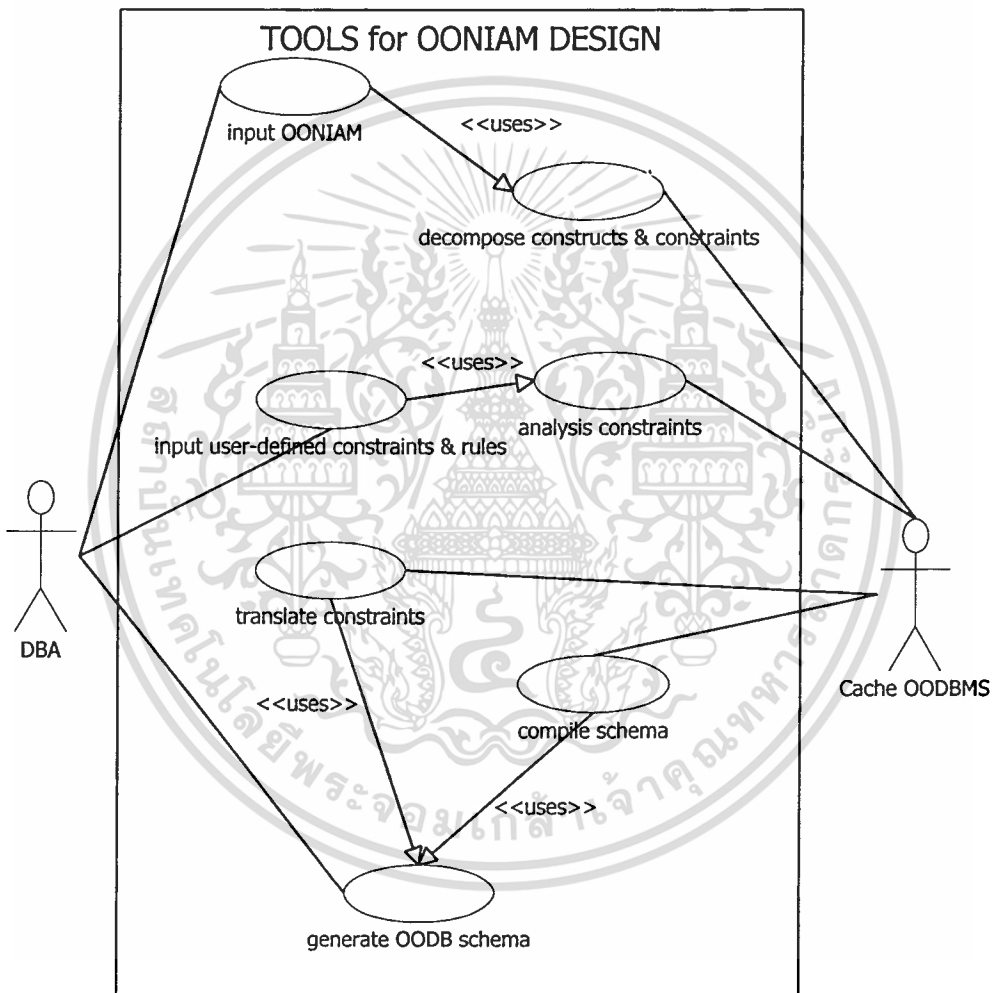
ซึ่งในการออกแบบระบบอัตโนมัตินี้ได้นำแผนภาพบางส่วนของ UML มาใช้ในการออกแบบระบบ โดยแผนภาพที่ได้นำมาใช้จะมีรายละเอียดดังที่จะกล่าวในหัวข้อต่อไป

5.1 Use case diagram

จากสถาปัตยกรรมของระบบที่ได้นำเสนอในบทที่ 4 มาแล้วนั้น สามารถนำมาเป็นแนวทางในการสร้าง Use case diagram เพื่อใช้ในการออกแบบระบบอัตโนมัติสำหรับออกแบบ แบบจำลองระดับแนวคิด OONIAM โดยใน Use case diagram นี้จะทำการแบ่งฟังก์ชันการทำงานทั้งหมดของระบบออกเป็น use case ย่อย ๆ ซึ่งสามารถแสดงฟังก์ชันต่าง ๆ ได้ดังรูปที่ 5.1 จากหลักการของการสร้าง Use case diagram นั้น สามารถแยกส่วนที่เกี่ยวข้องกับระบบออกเป็น 2 ส่วน คือส่วนของ Actors ซึ่งในส่วนนี้จะหมายถึงสิ่งที่กระทำกับระบบ ในการออกแบบนี้สามารถแยกได้ดังนี้

1. DBA คือผู้ใช้งานระบบอัตโนมัติดังกล่าว โดยเป็นผู้สร้างแบบจำลองระดับแนวคิด OONIAM หรือเป็นผู้มีความประสงค์ที่จะนำฐานข้อมูลเชิงสัมพันธ์ที่มีอยู่มาทำการ

1. DBA คือผู้ใช้งานระบบอัตโนมัติดังกล่าว โดยเป็นผู้สร้างแบบจำลองระดับแนวคิด OONIAM หรือเป็นผู้มีความประสงค์ที่จะนำฐานข้อมูลเชิงสัมพันธ์ที่มีอยู่มาทำการปรับปรุง โดยทำการแปลงให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุที่ต้องการ
2. Cache OODBMS เป็นระบบจัดการฐานข้อมูลเชิงวัตถุที่ทำหน้าที่ในการจัดเก็บข้อมูลที่ DBA ทำการป้อนเข้าสู่ระบบ รวมไปถึงการจัดเก็บข้อมูลผลลัพธ์ที่ได้จากการแปลง โดยใช้ระบบอัตโนมัติดังกล่าว เพื่อนำไปสร้างเป็นฐานข้อมูลเชิงวัตถุตามที่ต้องการ



รูปที่ 5.1 ภาพแสดง Use Case Diagram ของระบบ

และสำหรับอีกส่วนหนึ่งคือส่วนของ Use Cases โดยในแต่ละ Use Cases นั้นสามารถอธิบายการทำงานได้ดังนี้

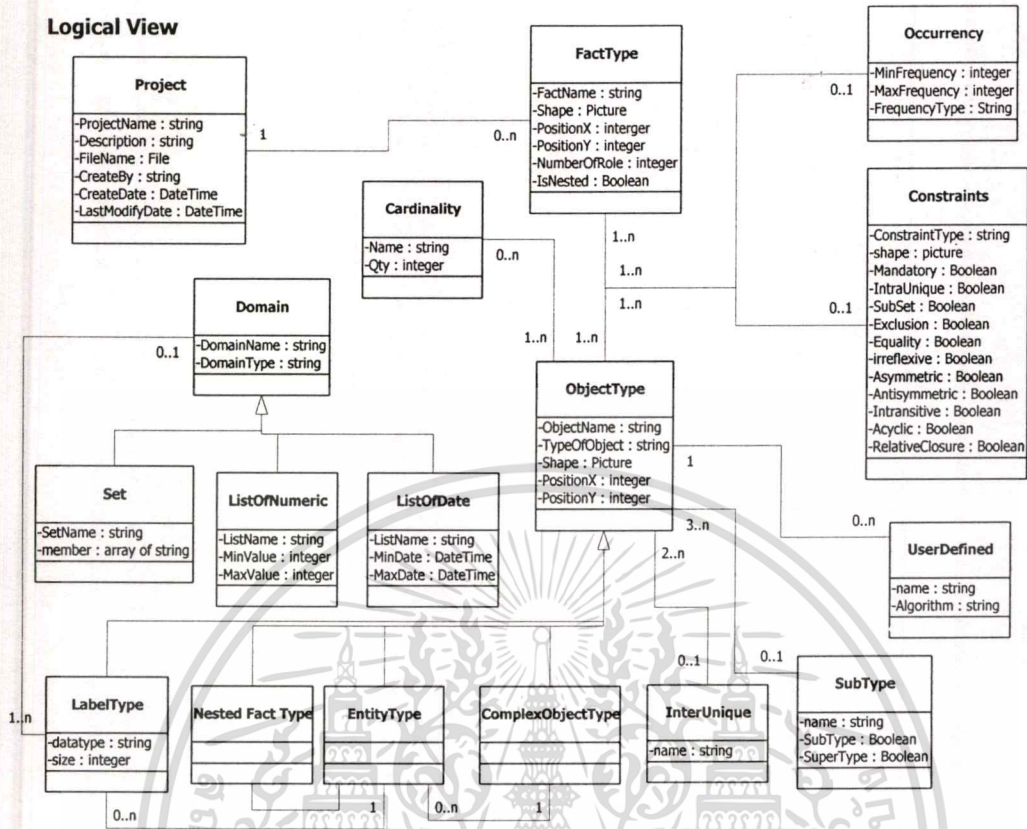
1. input OONIAM มีหน้าที่ในการรับข้อมูลแบบจำลองระดับแนวคิด OONIAM ที่สร้างขึ้นจาก DBA รวมไปถึงกฎข้อบังคับความถูกต้องต่าง ๆ ของข้อมูล ในระบบที่สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. input user-defined constraints and rules มีหน้าที่ในการรับข้อมูลของกฎข้อบังคับ ความถูกต้องของข้อมูลที่ใช้กำหนดขึ้นนอกเหนือจากที่ปรากฏในแบบจำลอง รวมไปถึงถึงกฎและขั้นตอนต่าง ๆ ที่ใช้ในระบบในรูปแบบของอัลกอริทึมที่กำหนดขึ้น
3. decompose constructs & constraints มีหน้าที่ในการวิเคราะห์และแยกประเภทของข้อมูลจากแบบจำลองระดับแนวคิดออกเป็น 2 ส่วนคือส่วนของโครงสร้างข้อมูลกับส่วนของกฎข้อบังคับความถูกต้องของข้อมูลที่แสดงอยู่ในแบบจำลอง แล้วทำการจัดเก็บลงในระบบจัดการฐานข้อมูลเชิงวัตถุ
4. analysis constraints มีหน้าที่ในการนำอัลกอริทึมของกฎข้อบังคับความถูกต้องของข้อมูลที่นอกเหนือจากที่ปรากฏอยู่ในแบบจำลอง รวมไปถึงถึงกฎและเงื่อนไขต่าง ๆ มาทำการวิเคราะห์และจัดเก็บลงบนระบบจัดการฐานข้อมูลเชิงวัตถุ
5. translate constraints มีหน้าที่ในการนำกฎข้อบังคับความถูกต้องของข้อมูลต่าง ๆ ของระบบที่ได้ทำการจัดเก็บไว้ในระบบจัดการฐานข้อมูลเชิงวัตถุ มาทำการแปลงให้อยู่ในรูปของเมธอดของฐานข้อมูลเชิงวัตถุ และทำการสร้างอัลกอริทึมสำหรับเมธอดต่าง ๆ ที่ได้กำหนดไว้ด้วย
6. compile schema มีหน้าที่ในการนำโครงสร้างของข้อมูลของแบบจำลองระดับแนวคิด OONIAM ที่ทำการจัดเก็บไว้ในระบบจัดการฐานข้อมูลเชิงวัตถุมาทำการแปลงให้อยู่ในรูปโครงสร้างของฐานข้อมูลเชิงวัตถุตามต้องการ
7. generate OODB Schemas มีหน้าที่ในการนำข้อมูลต่าง ๆ ที่ได้มาทำการสร้างเป็น Object Oriented Schema ในรูปแบบของ Class Diagram หรือ Object Definition Language ตามที่ต้องการ เพื่อให้ DBA สามารถนำมาใช้งานในการปรับปรุงระบบและทำการสร้างฐานข้อมูลที่ได้ลงบนระบบจัดการฐานข้อมูลเชิงวัตถุ

5.2 MetaModel

เมต้ามอดেলเป็นแบบจำลองที่ทำการสร้างขึ้นเพื่อใช้ในการจัดเก็บข้อมูลทั้งหมดที่เกี่ยวข้องกับระบบอัตโนมัติสำหรับออกแบบแบบจำลองระดับแนวคิด OONIAM โดยเมต้ามอดেলที่ได้สร้างขึ้นมานั้นจะสนับสนุนการออกแบบระบบเชิงวัตถุ ดังนั้นจึงนำเสนอเมต้ามอดัลในรูปแบบของแบบ Class Diagram เพื่อที่สามารถนำข้อมูลดังกล่าวมาทำการจัดเก็บลงบนระบบจัดการฐานข้อมูลเชิงวัตถุได้ต่อไป



รูปที่ 5.2 เมตาดาโมเดลที่ใช้ในการจัดเก็บแบบจำลองระดับแนวคิด OONIAM

ในส่วนของเมตาดาโมเดลที่สร้างขึ้นนั้น สามารถทำการแบ่งออกเป็น 2 ส่วนได้ โดยในส่วนแรกของเมตาดาโมเดลที่แสดงไว้ในรูปที่ 5.2 นั้น ได้ทำการออกแบบไว้สำหรับจัดเก็บข้อมูลรายละเอียดต่าง ๆ ของแบบจำลองระดับแนวคิด OONIAM ไว้ในระบบจัดการฐานข้อมูลเชิงวัตถุเพื่อนำมาใช้ในการแปลงฐานข้อมูลที่ออกแบบโดยใช้แบบจำลองระดับแนวคิด OONIAM ให้อยู่ในลักษณะของฐานข้อมูลเชิงวัตถุตามที่ต้องการ โดยได้ทำการแบ่งออกเป็นข้อมูลดังกล่าวออกเป็นออบเจ็กต์คลาสต่าง ๆ ดังนี้

1. Project Object Class แสดงรายละเอียดต่าง ๆ ของแต่ละ Project ที่นำมาใช้งานกับระบบอัตโนมัติที่สร้างขึ้น
2. Fact Type Object Class แสดงรายละเอียดต่าง ๆ ของแต่ละ Fact Type ที่มีอยู่ในแบบจำลองระดับแนวคิด OONIAM ที่สร้างขึ้น
3. Object Type Object Class แสดงรายละเอียดต่าง ๆ ของแต่ละ Object Type ที่มีอยู่ในแบบจำลอง โดยได้ทำการออกแบบให้มีความสัมพันธ์ลักษณะของเจนเนอรัลไลเซชัน

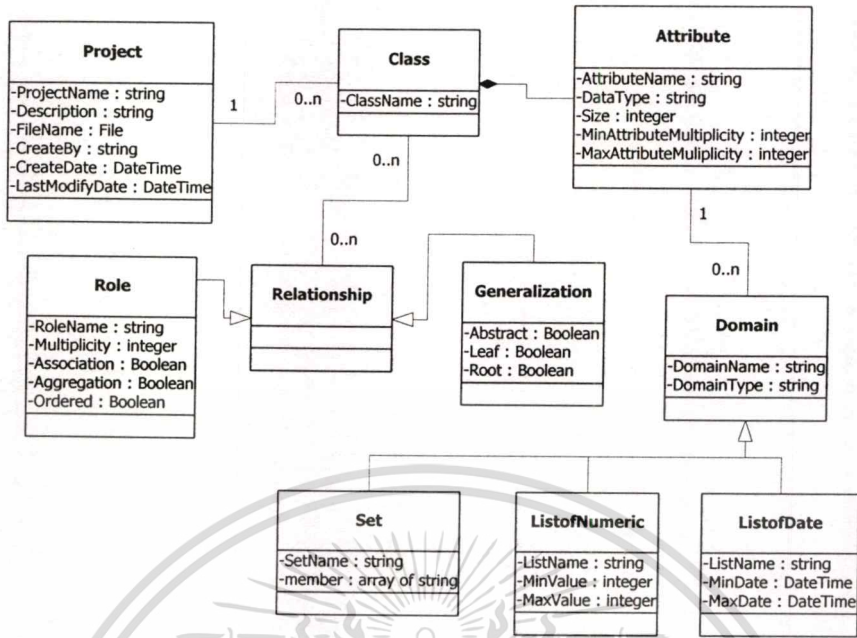
เอกสารนี้เป็นเอกสารที่เผยแพร่ภายใต้เงื่อนไขการใช้งานที่ออกโดยสำนักงานส่งเสริมการค้าในต่างประเทศ ณ นครเชียงใหม่ โดยทำการกำหนดให้ Object Type เป็นซูเปอร์คลาส และประกอบด้วยซับคลาสดังนี้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Label Type
 - Entity Type
 - Nested Fact Type
 - Complex Object Type
4. Domain Object Class แสดงรายละเอียดต่าง ๆ ของแต่ละโดเมนที่มีการกำหนดขึ้นเพื่อใช้งานในระบบ โดยได้ทำการออกแบบให้มีความสัมพันธ์ลักษณะของเจนเนอรัลไลเซชันโดยทำการกำหนดให้ Domain เป็นซูเปอร์คลาส และประกอบด้วยซับคลาสต่าง ๆ ดังนี้
 - Set แสดงรายละเอียดต่าง ๆ ของเซตจำกัดใด ๆ
 - ListofNumeric แสดงรายละเอียดต่าง ๆ ของเซตของจำนวนใด ๆ
 - ListofDate แสดงรายละเอียดต่าง ๆ ของเซตของวันเวลาใด ๆ
 5. Cardinality Object Class แสดงรายละเอียดต่าง ๆ ของกฎข้อบังคับความถูกต้องประเภท Cardinality
 6. Occurrency Object Class แสดงรายละเอียดต่าง ๆ ของกฎข้อบังคับความถูกต้องประเภท Occurrency
 7. InterUnique Object Class แสดงรายละเอียดต่าง ๆ ของกฎข้อบังคับความถูกต้องประเภท Inter Uniqueness
 8. SubType Object Class แสดงรายละเอียดต่าง ๆ ของกฎข้อบังคับความถูกต้องประเภท SubType
 9. Constraints Object Class แสดงรายละเอียดต่าง ๆ ของกฎข้อบังคับความถูกต้อง โดยจะรวบรวมกฎต่าง ๆ ไว้ดังนี้ Mandatory constraints, Intra Uniqueness constraints, SubSet constraints, Exclusion constraints, Equality constraints, Irreflexive constraints, Asymmetric constraints, Antisymmetric constraints, Intransitive constraints, Acyclic constraints, Relative Closure constraints
 10. UserDefined Object Class แสดงรายละเอียดต่าง ๆ ของกฎข้อบังคับความถูกต้อง, ขึ้นตอนต่าง ๆ และ เงื่อนไขของระบบ ที่แสดงอยู่ในรูปของเมรกดของแบบจำลองระดับแนวคิด OONIAM

สำหรับในส่วนที่สองของเมต้าโมเดลดังแสดงในรูปที่ 5.3 นั้นได้ทำการออกแบบไว้สำหรับทำการจัดเก็บข้อมูลรายละเอียดต่าง ๆ ของผลลัพธ์ที่ได้จากการแปลงแบบจำลองระดับแนวคิด OONIAM ให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุ โดยในการจัดเก็บข้อมูลดังกล่าวได้ทำการแบ่งออกเป็นออบเจ็กต์คลาสต่าง ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 เมตาดาโมเดลที่ใช้ในการจัดเก็บผลลัพธ์ที่ได้จากการแปลง

1. Project Object Class แสดงรายละเอียดต่าง ๆ ของแต่ละ Project ที่นำมาใช้งานกับระบบอัตโนมัติที่ทำงานเป็นออบเจกต์คลาสเดียวกับส่วนแรก
2. Domain Object Class แสดงรายละเอียดต่าง ๆ ของแต่ละโดเมนที่มีการกำหนดขึ้นเพื่อใช้งานในระบบ โดยได้ทำการออกแบบให้มีความสัมพันธ์ลักษณะของเจนเนอรัลไลเซชัน โดยทำการกำหนดให้ Domain เป็นซูเปอร์คลาส และประกอบด้วยซับคลาสต่าง ๆ ดังนี้
 - Set แสดงรายละเอียดต่าง ๆ ของเซตจำกัดใด ๆ
 - ListofNumeric แสดงรายละเอียดต่าง ๆ ของเซตของจำนวนใด ๆ
 - ListofDate แสดงรายละเอียดต่าง ๆ ของเซตของวันเวลาใด ๆ
 เป็นออบเจกต์คลาสเดียวกับที่แสดงไว้ในส่วนแรก
3. Class Object Class แสดงรายละเอียดต่าง ๆ ของแต่ละคลาสที่สร้างมาจากการแปลงของระบบอัตโนมัติดังกล่าว
4. Attribute Object Class แสดงรายละเอียดต่าง ๆ ของแต่ละแอตทริบิวต์ซึ่งเป็นส่วนประกอบของแต่ละคลาสที่ได้จากการแปลง
5. Relationship Object Class แสดงรายละเอียดต่าง ๆ ของความสัมพันธ์ที่ได้จากการแปลง โดยได้ทำการออกแบบให้มีความสัมพันธ์ลักษณะของเจนเนอรัลไลเซชัน โดย

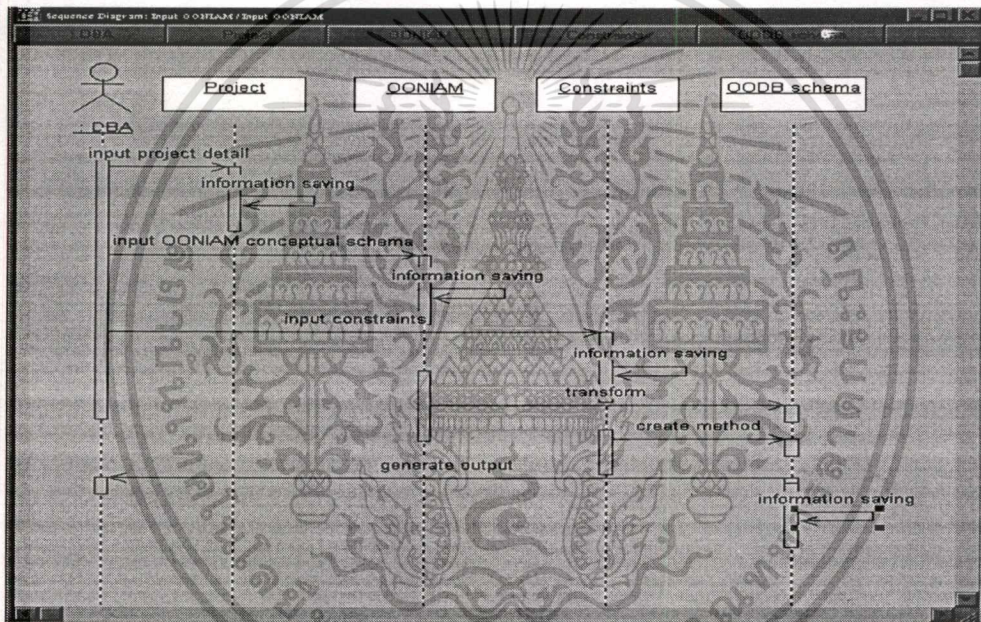
เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการกำหนดให้ Relationship เป็นซูเปอร์คลาส และประกอบไปด้วยซับคลาสดังนี้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Role แสดงรายละเอียดของต่าง ๆ ของความสัมพันธ์ที่เกิดขึ้นระหว่างคลาสโดยจะแสดงถึงความสัมพันธ์แบบแอโซซิเอชัน หรือความสัมพันธ์แบบแอกกรีเกรชัน
- Generalization แสดงรายละเอียดต่าง ๆ ของความสัมพันธ์แบบเจนเนอรัลไลเซชัน

5.3 Sequence Diagram และ Collaboration Diagram

เป็นแผนภาพที่แสดงขั้นตอนการทำงานที่เกิดขึ้นกับออบเจกต์ต่าง ๆ ของระบบ เพื่อให้ทราบถึงขั้นตอนการทำงานของระบบ ซึ่งสามารถแสดงโดยใช้ Sequence diagram ในการออกแบบขั้นตอนที่เกิดขึ้น ดังรูปที่ 5.4 และแสดงโดยใช้ Collaboration diagram ดังในรูปที่ 5.5



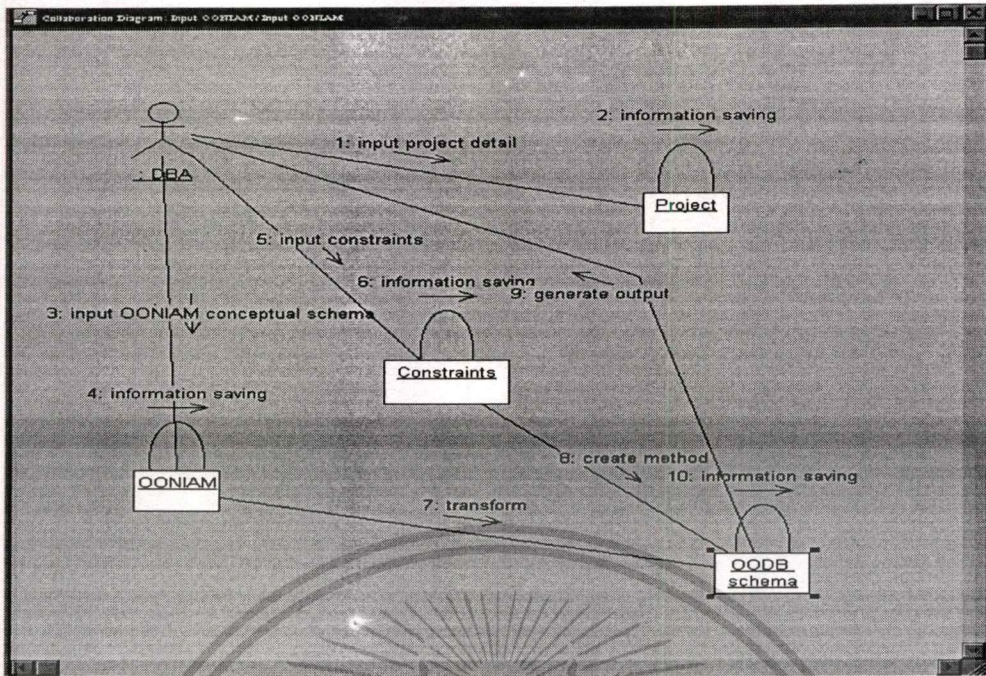
รูปที่ 5.4 ภาพแสดง Sequence diagram ของระบบ

จากภาพทั้งสองภาพจะแสดงให้เห็นว่าได้มีการแบ่งออบเจกต์ที่เกี่ยวข้องแบ่งออกเป็นออบเจกต์ต่าง ๆ ดังนี้

- Project
- OONIAM
- Constraints
- OODB schema

โดยออบเจกต์ดังกล่าวในบางออบเจกต์จะเป็นการรวมกลุ่มกันของออบเจกต์คลาสที่ได้ออกแบบไว้แล้วในเมตาโมเดลที่ได้กล่าวมาแล้วข้างต้นนั้น

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 ภาพแสดง Collaboration diagram ของระบบ

โดยขั้นตอนการทำงานจะมีดังนี้

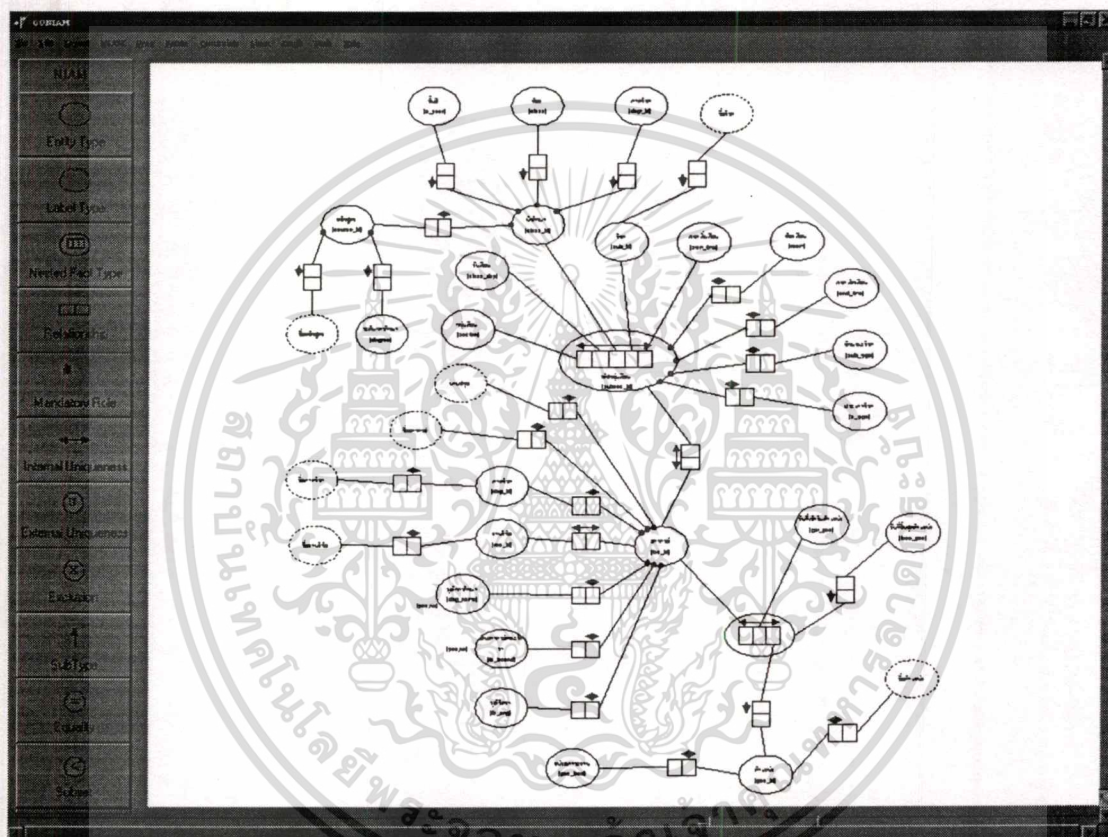
1. ป้อนข้อมูลรายละเอียดของออบเจกต์ Project
2. บันทึกข้อมูลรายละเอียดลงบนระบบจัดการฐานข้อมูล
3. ป้อนข้อมูลรายละเอียดของแบบจำลอง OONIAM ในที่นี้รวมไปถึงข้อมูลที่ได้ทำการแบ่งออกเป็นออบเจกต์คลาสต่าง ๆ ที่ได้อธิบายไว้ในส่วนของเมต้าโมเดลดังนั้นการป้อนข้อมูลในขั้นตอนนี้จะหมายถึงการป้อนรายละเอียดทั้งหมดของแบบจำลอง เพื่อนำมาใช้ในการสร้างฐานข้อมูลหรือปรับปรุงฐานข้อมูลที่ต้องการ
4. บันทึกข้อมูลรายละเอียดทั้งหมดของแบบจำลอง OONIAM
5. ป้อนข้อมูลรายละเอียดของกฎข้อบังคับความถูกต้องในที่นี้จะหมายถึงกฎข้อบังคับความถูกต้องที่ไม่สามารถแสดงบนแผนภาพได้ ซึ่งจะรวมถึงขั้นตอนและเงื่อนไขต่าง ๆ ของระบบด้วย โดยในแบบจำลองระดับแนวคิด OONIAM ที่ใช้นั้นจะรวมเรียกว่าเมรูด
6. บันทึกข้อมูลรายละเอียดของกฎข้อบังคับความถูกต้องของข้อมูลทั้งหมด
7. นำข้อมูลของ OONIAM ทั้งหมดมาทำการแปลงให้อยู่ในรูปแบบของ Object Oriented Database Schema
8. นำกฎข้อบังคับความถูกต้องทั้งหมดที่มีมาทำการสร้างเป็นเมรูดของออบเจกต์ที่เกี่ยวข้อง
9. แสดงผลลัพธ์ที่อยู่ในรูปของฐานข้อมูลเชิงวัตถุที่ได้จากการแปลง
10. บันทึกข้อมูลที่ได้จากแปลงลงบนฐานข้อมูลเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ระบบอัตโนมัติสำหรับออกแบบ OONIAM Conceptual Schema

ระบบอัตโนมัติสำหรับออกแบบ OONIAM Conceptual Schema เป็นเครื่องมือที่ได้พัฒนาขึ้นสำหรับใช้ในการแปลงข้อมูลจากแบบจำลองระดับแนวคิด OONIAM ที่ได้ทำการสร้างขึ้นใหม่ โดยใช้พื้นฐานมาจากแบบจำลองระดับแนวคิดในแอมมาทำการแปลงให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุ โดยสามารถนำมาใช้ในการออกแบบแผนภาพ NIAM และแผนภาพ OONIAM ได้ โดยแผนภาพดังกล่าวสามารถจัดทำขึ้นดังภาพที่แสดงไว้ในรูปที่ 5.6

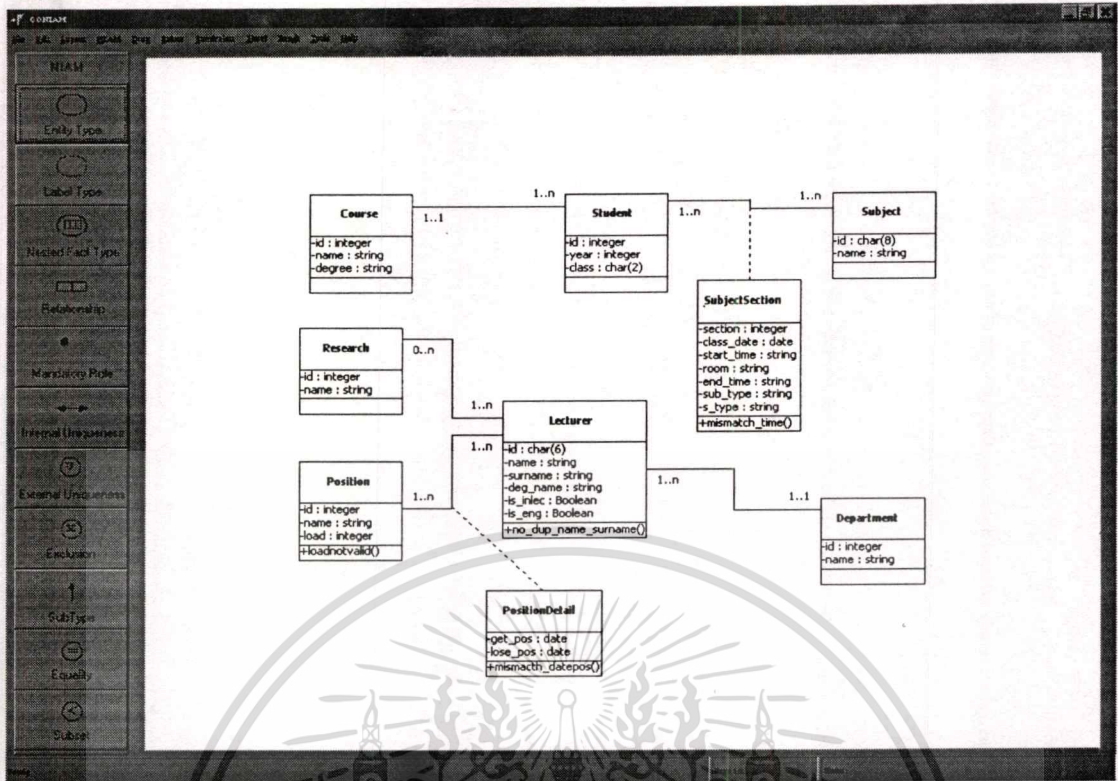


รูปที่ 5.6 ภาพแสดงตัวอย่างการสร้าง OONIAM Conceptual Schema

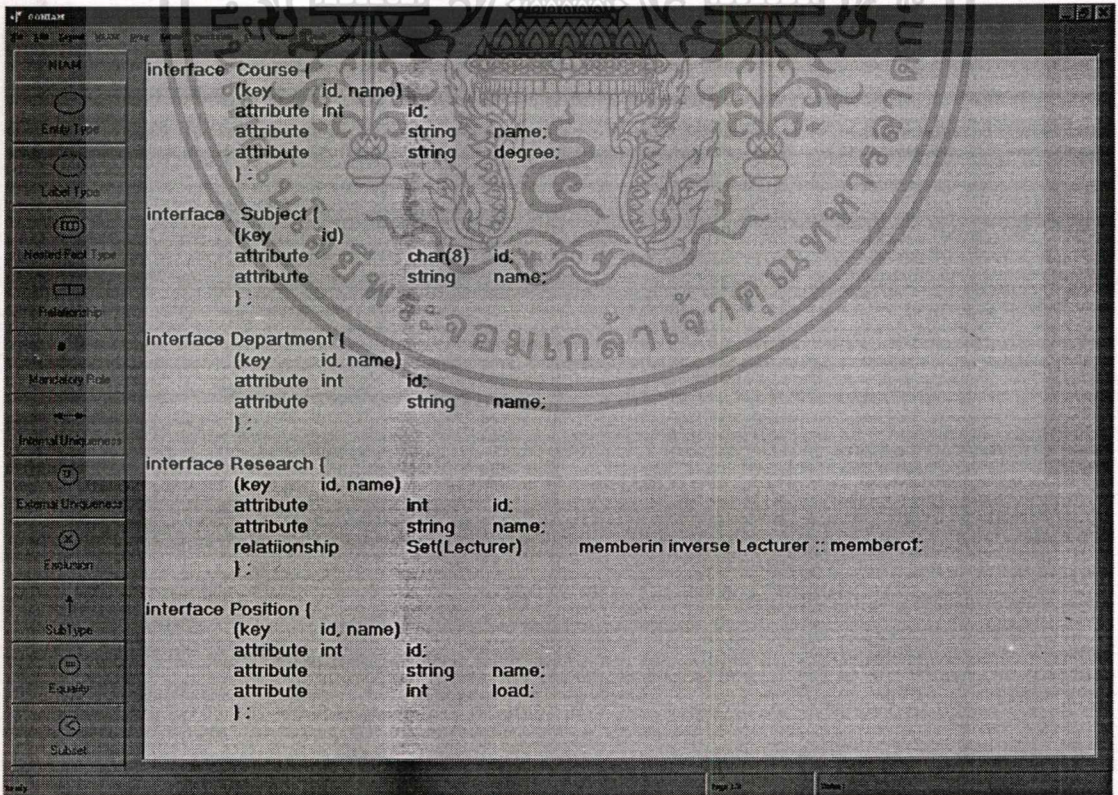
ในการสร้างแผนภาพนั้น ผู้ใช้สามารถกำหนดกฎข้อบังคับกับความถูกต้องต่าง ๆ ที่แสดงได้บนแผนภาพและที่ไม่อาจแสดงได้ในแผนภาพผู้ใช้สามารถกำหนดในลักษณะของเมธอดได้ เพื่อจะนำมาใช้ในการสร้างกฎข้อบังคับกับความถูกต้องบนระบบจัดการฐานข้อมูลเชิงวัตถุต่อไป

ผู้ใช้สามารถที่จะใช้ระบบอัตโนมัติสำหรับออกแบบ OONIAM นี้ทำการแปลงจากแผนภาพ OONIAM โดยสามารถเลือกผลลัพธ์ที่ต้องการว่าจะให้อยู่ในรูปของ Class Diagram หรือ Object Definition Language ได้ดังภาพที่ปรากฏในรูปที่ 5.7 และ 5.8 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 ภาพแสดงผลพัทธ์ที่อยู่ในรูปของ Class Diagram



รูปที่ 5.8 ภาพแสดงผลพัทธ์ที่ได้ในรูปของ Object Definition Language

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเปล่งนั้นจะทำการเปล่งกฎข้อบังคับความถูกต้องที่ปรากฏอยู่บนแผนภาพให้อยู่ในรูปของเมธอดของออบเจกต์คลาสที่เกี่ยวข้อง และยังสามารถให้ผู้ใช้งานกำหนดเมธอดของออบเจกต์คลาสนั้น ๆ เพิ่มเติมได้อีก เพื่อให้ทำให้ออบเจกต์คลาสที่สร้างขึ้นมีความสมบูรณ์สามารถนำมาใช้งานได้จริงบนระบบจัดการฐานข้อมูลเชิงวัตถุต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การเปรียบเทียบวิธีการออกแบบฐานข้อมูลเชิงวัตถุ

6.1 การออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ Rational Rose

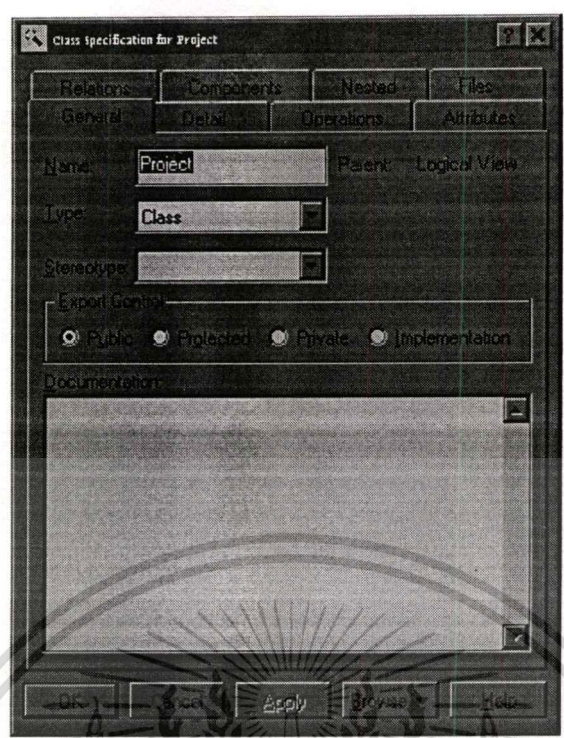
ในการออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ Rational Rose 98i [7] ซึ่งผลิตโดยบริษัท Rational Software Corporation เป็นโปรแกรมที่ใช้ในการออกแบบระบบเชิงวัตถุ ซึ่งในโปรแกรมดังกล่าวมีส่วนที่ไว้สำหรับใช้ในการออกแบบส่วนของฐานข้อมูลด้วย โดยในที่นี้เรียกว่าการออกแบบคลาสไดอะแกรม ซึ่งวิธีการออกแบบโดยใช้เครื่องมือดังกล่าวจะมีลักษณะดังต่อไปนี้



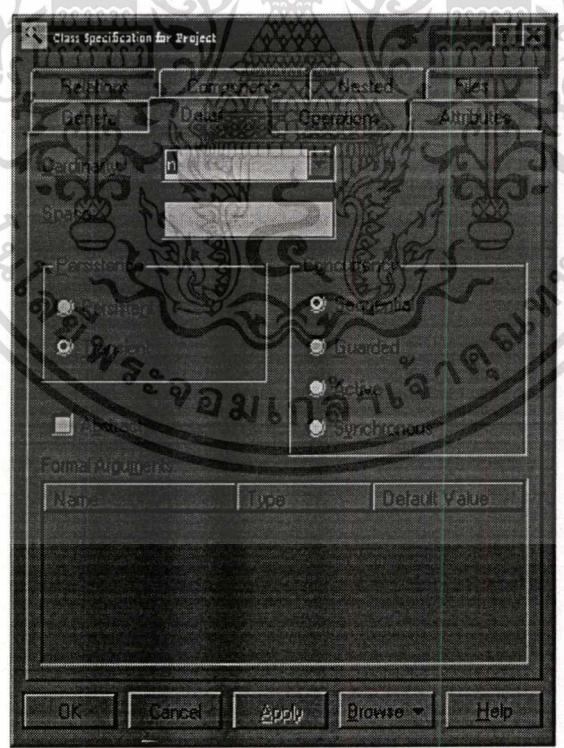
รูปที่ 6.1 ภาพแสดงการออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ Rational Rose

จากรูปที่ 6.1 สามารถทำการออกแบบคลาสไดอะแกรมได้โดยเริ่มทำการสร้างคลาสโดยทำการคลิกเมาส์ขวาที่ Logical View จะมีตัวเลือกในการสร้างคลาส หลังจากนั้นจะเกิดคลาสขึ้นมาดังภาพ NewClass ผู้ใช้สามารถใส่ชื่อคลาสที่ต้องการได้ สำหรับในการกำหนดลักษณะภายในคลาสนั้น ๆ ทำได้โดยทำการเลือกคลาสที่ต้องการกำหนดรายละเอียด แล้วทำการคลิกเมาส์ขวาที่คลาสนั้น โดยเลือกตัวเลือก Open Specification จะปรากฏหน้าจอในการกำหนดรายละเอียดดังรูปข้างล่าง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

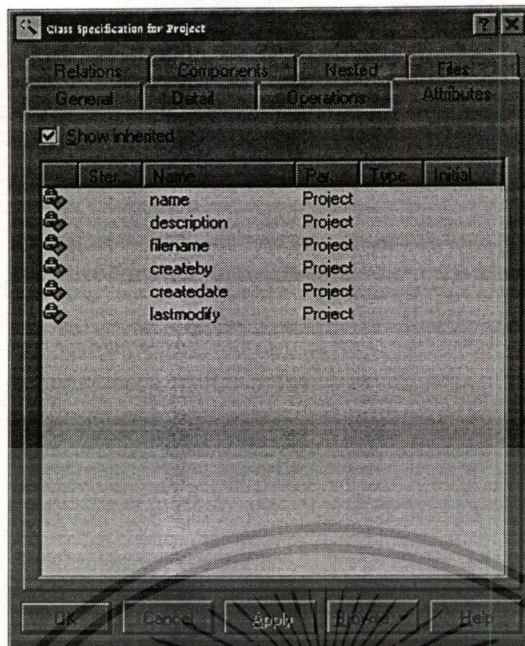


รูปที่ 6.2 ภาพแสดงตัวอย่างในการกำหนดรายละเอียดของชื่อคลาส



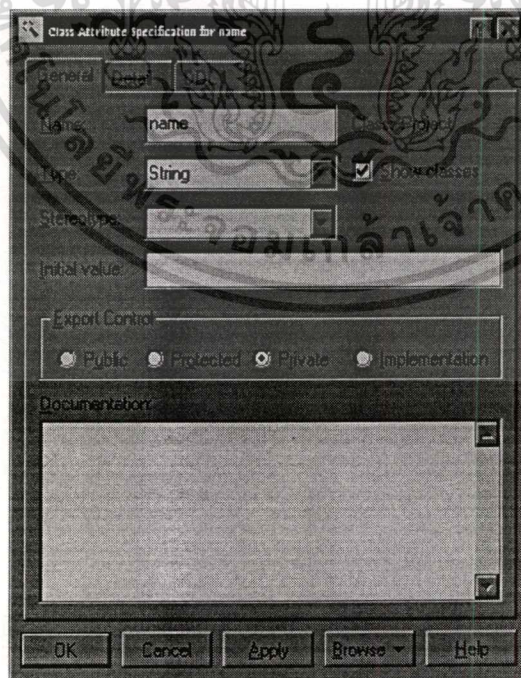
รูปที่ 6.3 ภาพแสดงตัวอย่างในการกำหนดรายละเอียดต่าง ๆ ของคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 ภาพแสดงตัวอย่างในการกำหนดแอตทริบิวต์ของคลาส

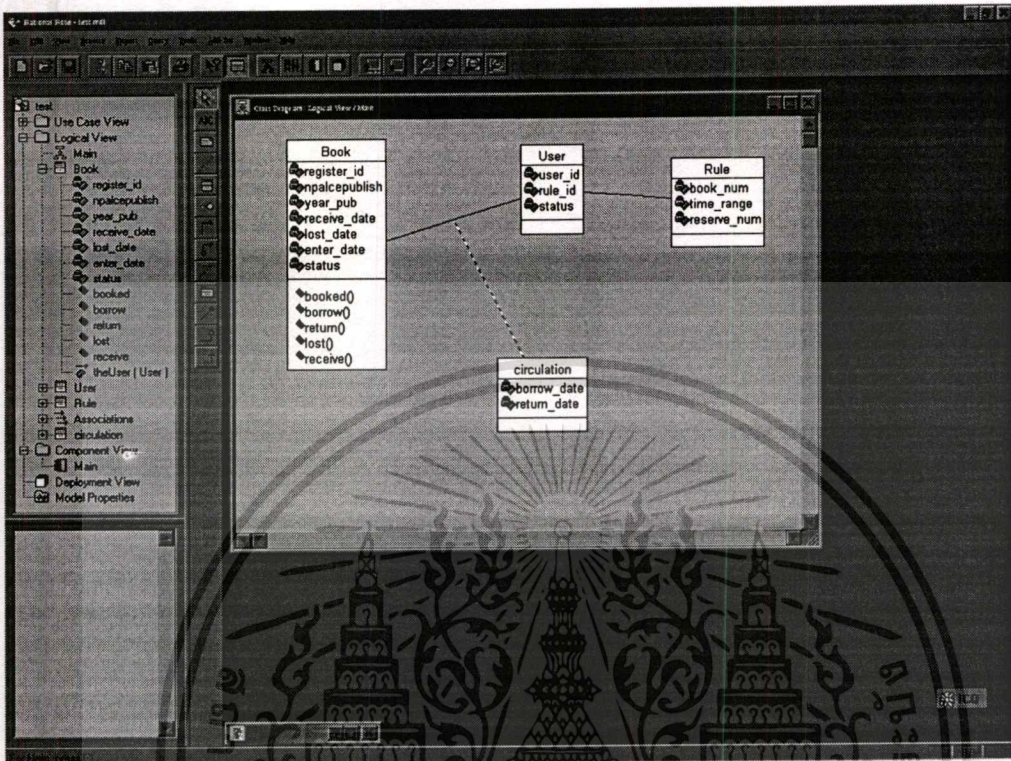
หลังจากกำหนดรายละเอียดของคลาสต่าง ๆ แล้ว ผู้ใช้สามารถทำการกำหนดรายละเอียดของแต่ละแอตทริบิวต์ได้จากหน้าจอที่ปรากฏอยู่ในรูปที่ 6.4 ทำการคลิกที่กุญแจของแต่ละแอตทริบิวต์จะปรากฏหน้าจอสำหรับกำหนดรายละเอียดของแอตทริบิวต์ดังนี้



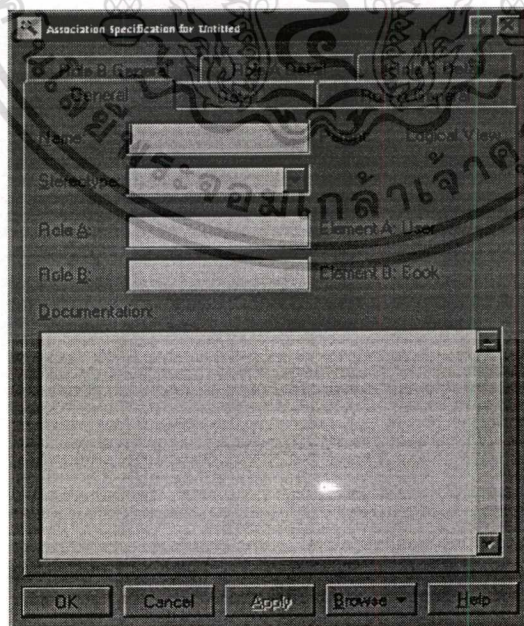
รูปที่ 6.5 ภาพแสดงตัวอย่างการกำหนดรายละเอียดของแอตทริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นผู้ใช้สามารถดึงคลาสดังกล่าวมาปรากฏอยู่บนเฟรมคลาสไดอะแกรมแล้ว
สามารถทำการกำหนดความสัมพันธ์ระหว่างคลาสได้ดังตัวอย่างในรูปที่ 6.6



รูปที่ 6.6 ภาพแสดงตัวอย่างของคลาสไดอะแกรม



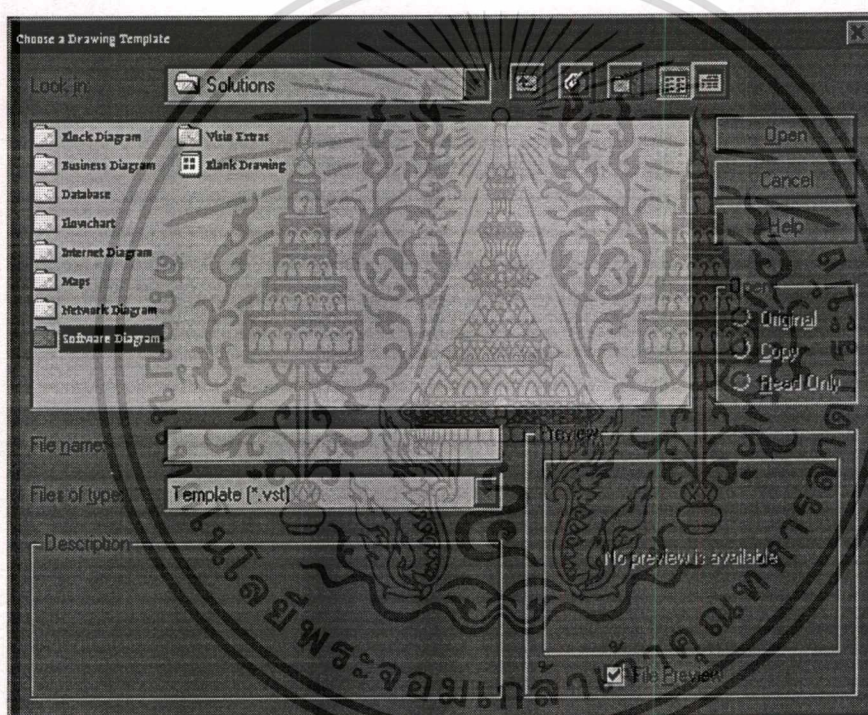
รูปที่ 6.7 ภาพแสดงตัวอย่างการกำหนดรายละเอียดของความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้สามารถกำหนดรายละเอียดของความสัมพันธ์ระหว่างคลาสได้ดังหน้าจอที่ปรากฏอยู่ในรูปที่ 6.7 เพื่อให้การออกแบบคลาสไดอะแกรมนั้นมีความสมบูรณ์มากขึ้น

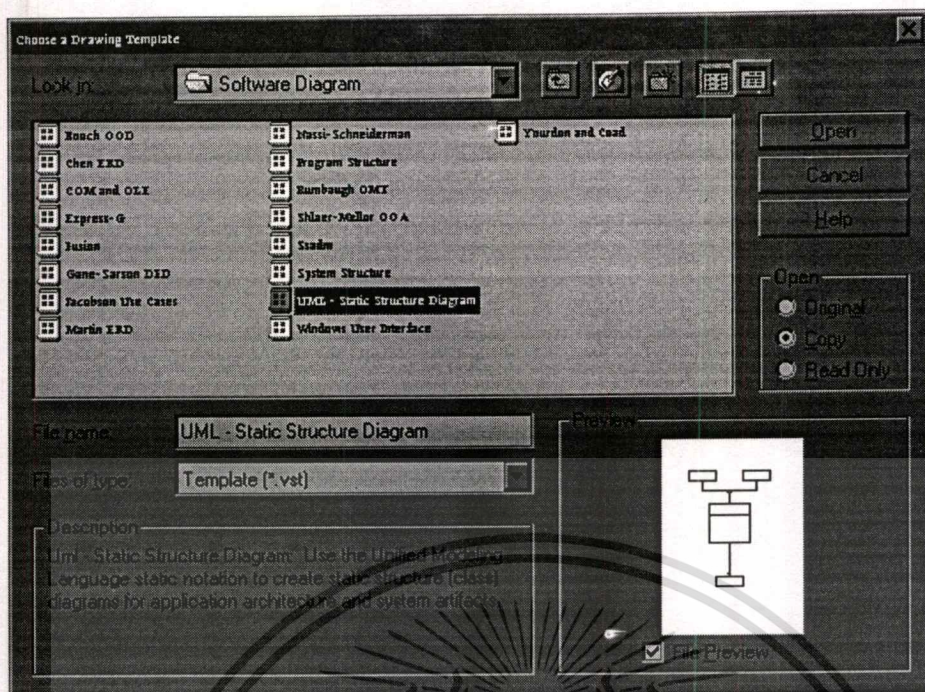
6.2 การออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ Visio Professional

ในการออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ Visio Professional version 5.0a for Microsoft Windows ซึ่งผลิตโดยบริษัท Visio Corporation เป็นโปรแกรมที่นิยมใช้ในการออกแบบแบบจำลองต่าง ๆ รวมไปถึงสามารถใช้ในการออกแบบแบบจำลองสำหรับฐานข้อมูลเชิงวัตถุได้อีกด้วย โดยในการออกแบบนั้นจะต้องมีการเลือกอุปกรณ์ในการสร้างแบบจำลองที่เหมาะสม โดยสามารถเลือกตามขั้นตอนดังต่อไปนี้

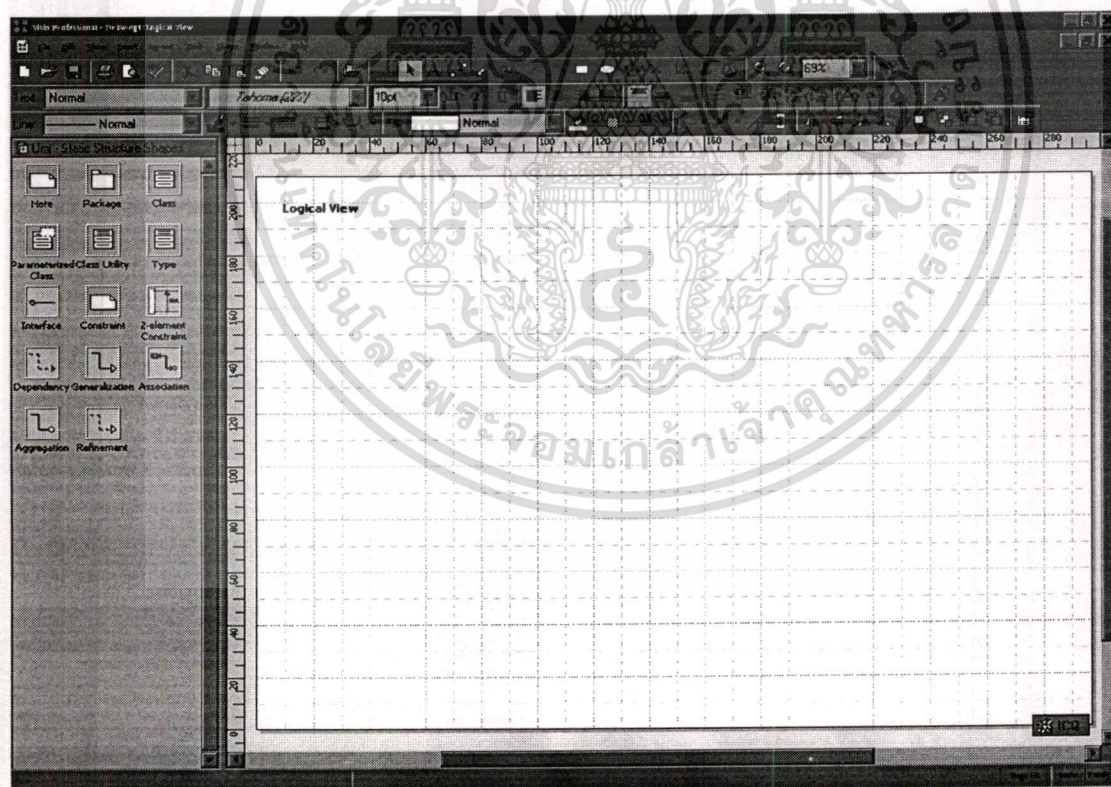


รูปที่ 6.8 ภาพแสดงการเลือก Template ที่ใช้ในการสร้างแบบจำลอง

หลังจากผู้ใช้เรียกใช้งานโปรแกรม Visio แล้ว เมื่อปรากฏหน้าจอในรูปที่ 6.8 ให้ผู้ใช้ทำการเลือกกลุ่มของเครื่องมือเป็นกลุ่ม Software Diagram เพื่อทำการเลือกเครื่องมือต่อไปจะปรากฏหน้าจอในรูปที่ 6.9 ให้ผู้ใช้ทำการเลือกเครื่องมือ UML- Static Structure Diagram เพื่อใช้ในการออกแบบฐานข้อมูลเชิงวัตถุต่อไป



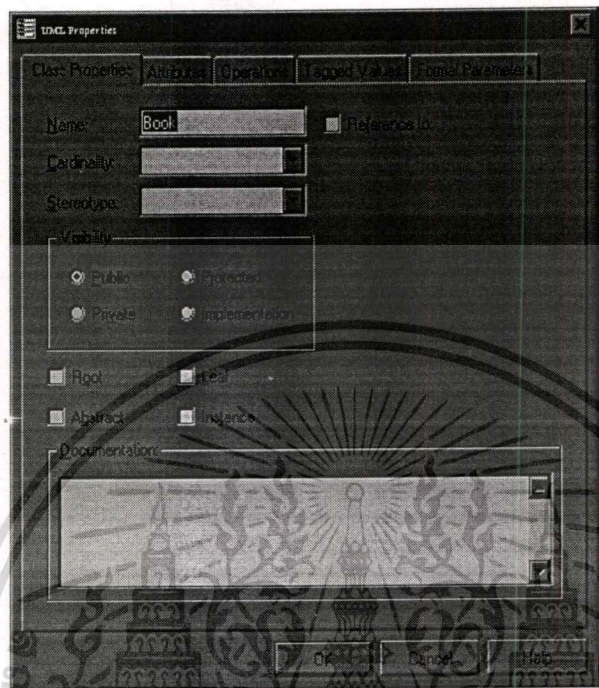
รูปที่ 6.9 ภาพแสดงรายละเอียดของกลุ่มของเครื่องมือ



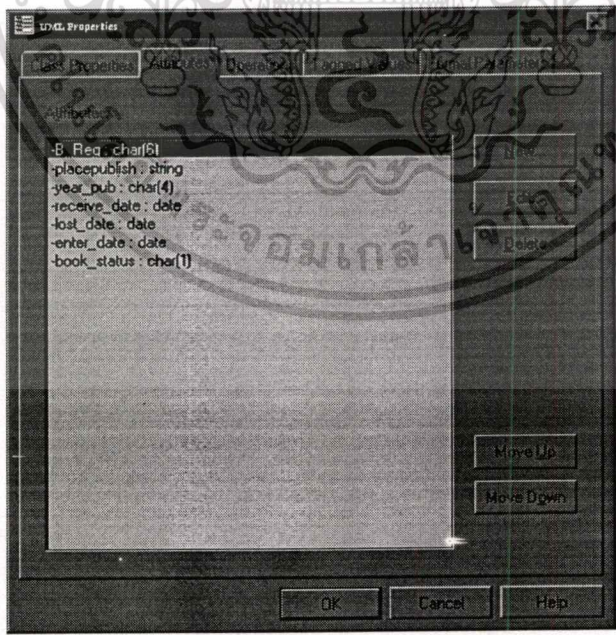
รูปที่ 6.10 ภาพแสดงหน้าจอสำหรับการออกแบบแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ผู้ใช้เลือกเครื่องมือเรียบร้อยแล้วนั้น จะปรากฏหน้าต่างดังรูปที่ 6.10 เพื่อให้ผู้ใช้สามารถออกแบบฐานข้อมูลเชิงวัตถุอย่างที่ต้องการได้ ในการออกแบบนั้นผู้ใช้สามารถที่จะทำการดึงเครื่องมือทางซ้ายมือมาวางบนเฟรมที่วางไว้ทางขวามือได้



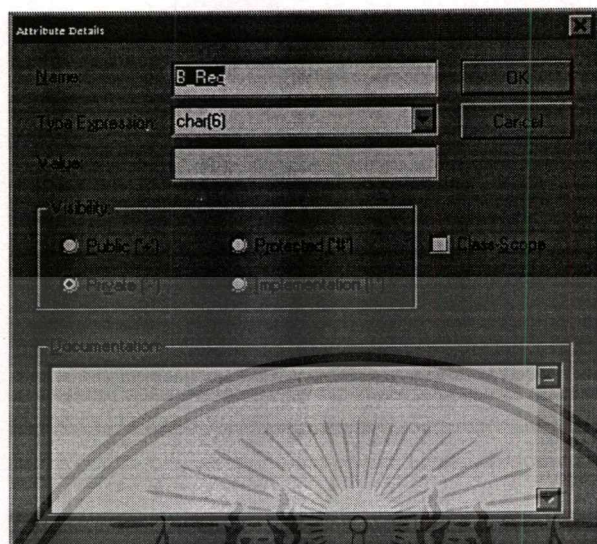
รูปที่ 6.11 ภาพแสดงตัวอย่างการกำหนดรายละเอียดของคลาส



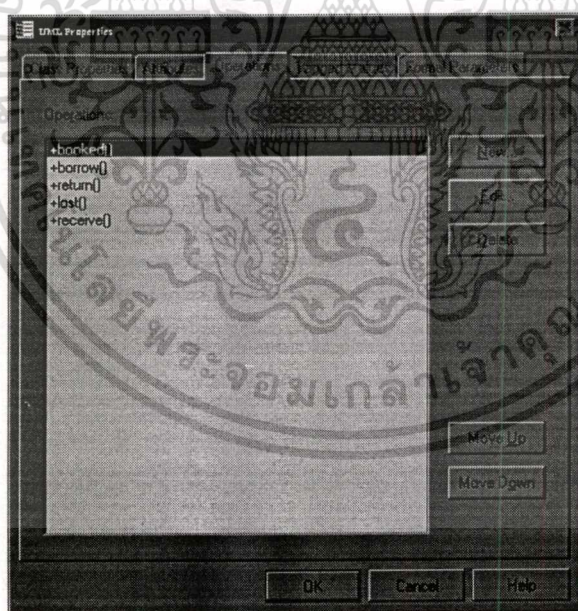
รูปที่ 6.12 ภาพแสดงการกำหนดแอตทริบิวต์ของคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการกำหนดแต่ละแอตทริบิวต์ของคลาสนั้นสามารถทำได้โดยการกดปุ่ม New จะแสดงหน้าจอสำหรับการบันทึกรายละเอียดของแอตทริบิวต์ดังนี้



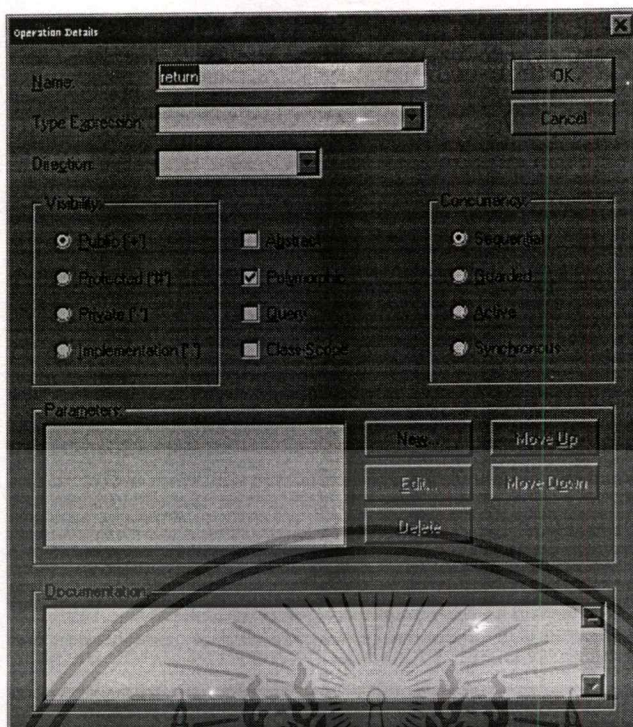
รูปที่ 6.13 ภาพแสดงตัวอย่างในการกำหนดรายละเอียดของแอตทริบิวต์



รูปที่ 6.14 ภาพแสดงตัวอย่างในการกำหนดเมธอดของคลาส

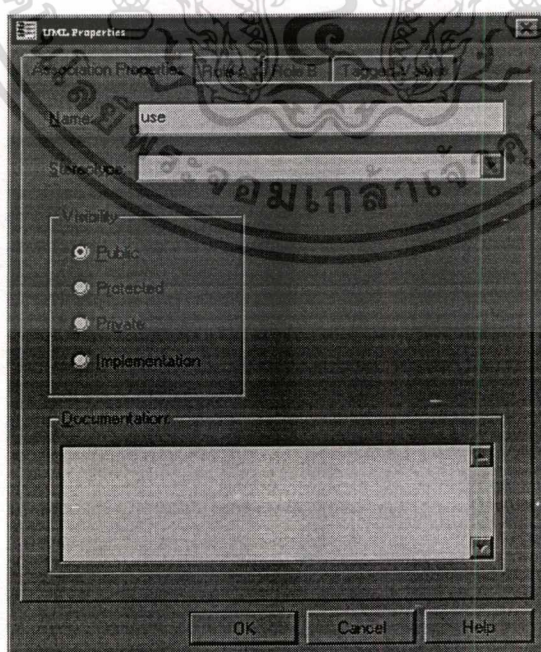
หลังจากที่ผู้ใช้ทำการกำหนดแอตทริบิวต์ของคลาสนั้นแล้วนั้น ผู้ใช้สามารถทำการกำหนดเมธอดของคลาสนี้ได้โดยทำการกดปุ่ม New ในหน้าจอที่ปรากฏอยู่ในรูปที่ 6.14 ซึ่งจะมีหน้าจอเพื่อให้ผู้ใช้ทำการกำหนดรายละเอียดของเมธอดดังที่ปรากฏอยู่ในรูปที่ 6.15 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.15 ภาพแสดงการกำหนดรายละเอียดของแต่ละเมธอด

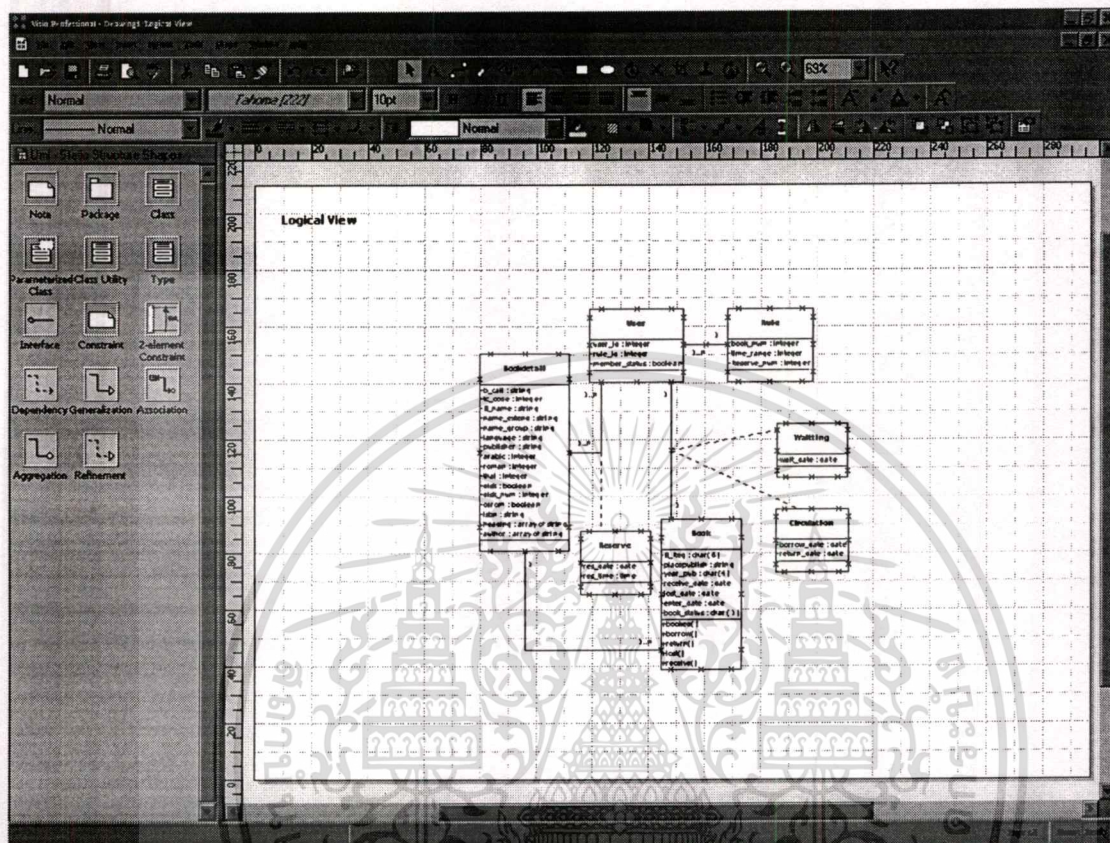
เมื่อผู้ใช้ทำการออกแบบรายละเอียดของแต่ละคลาสเสร็จแล้ว ผู้ใช้สามารถทำการกำหนดความสัมพันธ์ของระหว่างคลาสได้และกำหนดรายละเอียดของแต่ละความสัมพันธ์ได้จากหน้าจอ ดังรูปที่ 6.16



รูปที่ 6.16 ภาพแสดงหน้าจอสำหรับกำหนดรายละเอียดของความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

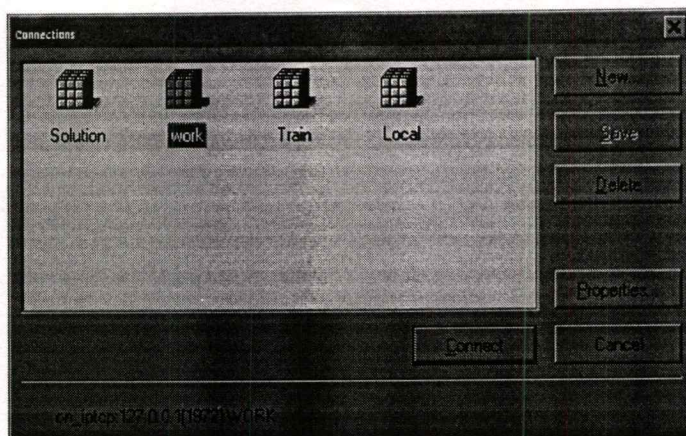
หลังจากที่ผู้ใช้ทำการออกแบบฐานข้อมูลเชิงวัตถุโดยกำหนดรายละเอียดต่าง ๆ เรียบร้อยแล้ว จะปรากฏแบบจำลองที่ได้ดังรูปที่ 6.17



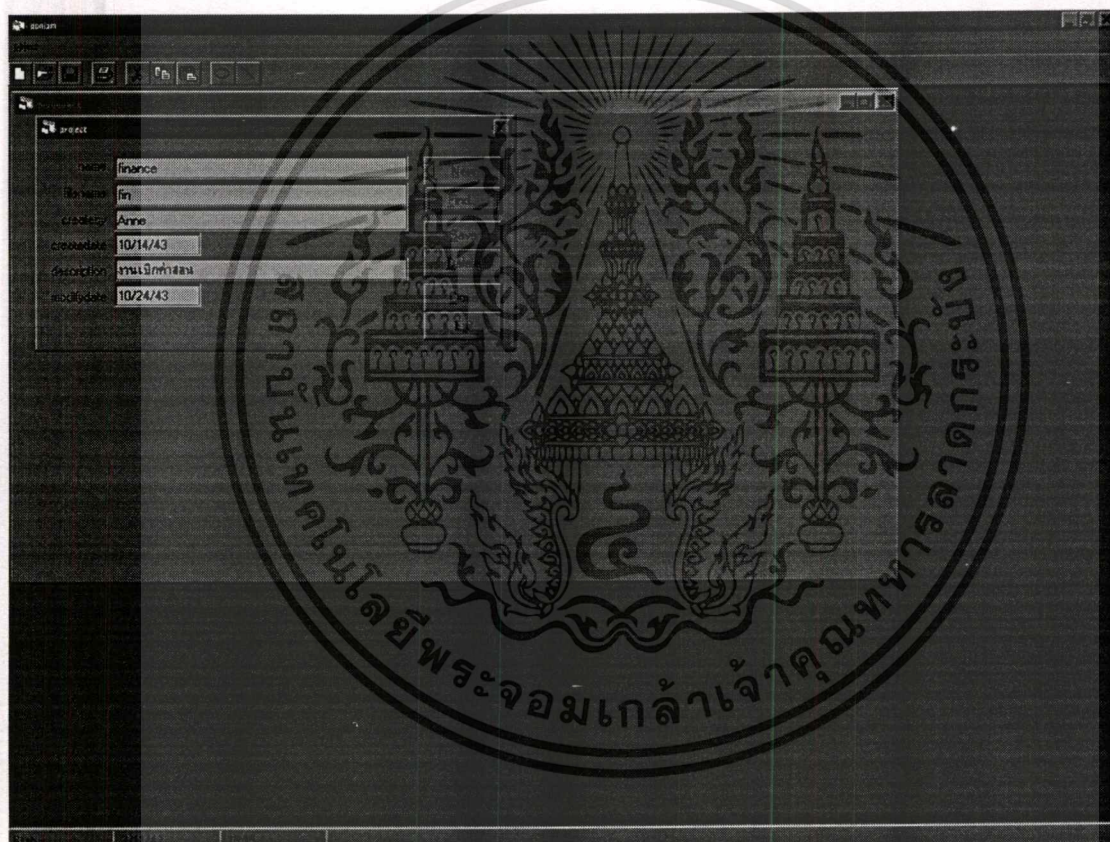
รูปที่ 6.17 ภาพแสดงผลลัพธ์ที่ได้จากการออกแบบ

6.3 การออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ OONIAM

จากเครื่องมือที่ได้ทำการพัฒนาเพื่อช่วยในการออกแบบแบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM นั้นได้ทำการพัฒนาโดยใช้ระบบจัดการฐานข้อมูลเชิงวัตถุทำการจัดเก็บข้อมูลทั้งหมดในระบบ ดังนั้นการที่ผู้ใช้จะทำการใช้เครื่องมือดังกล่าว ต้องทำการติดต่อกับระบบจัดการฐานข้อมูลเชิงวัตถุหลังจากเรียกใช้เครื่องมือดังกล่าว โดยจะปรากฏหน้าจอดังรูปที่ 6.18 หลังจากติดต่อกับฐานข้อมูลได้เรียบร้อยแล้วจะปรากฏหน้าจอดังรูปที่ 6.19 เพื่อให้ผู้ใช้ทำการเลือกโครงการที่ผู้ใช้ต้องการออกแบบใหม่หรือเรียกโครงการเก่ามาแก้ไข



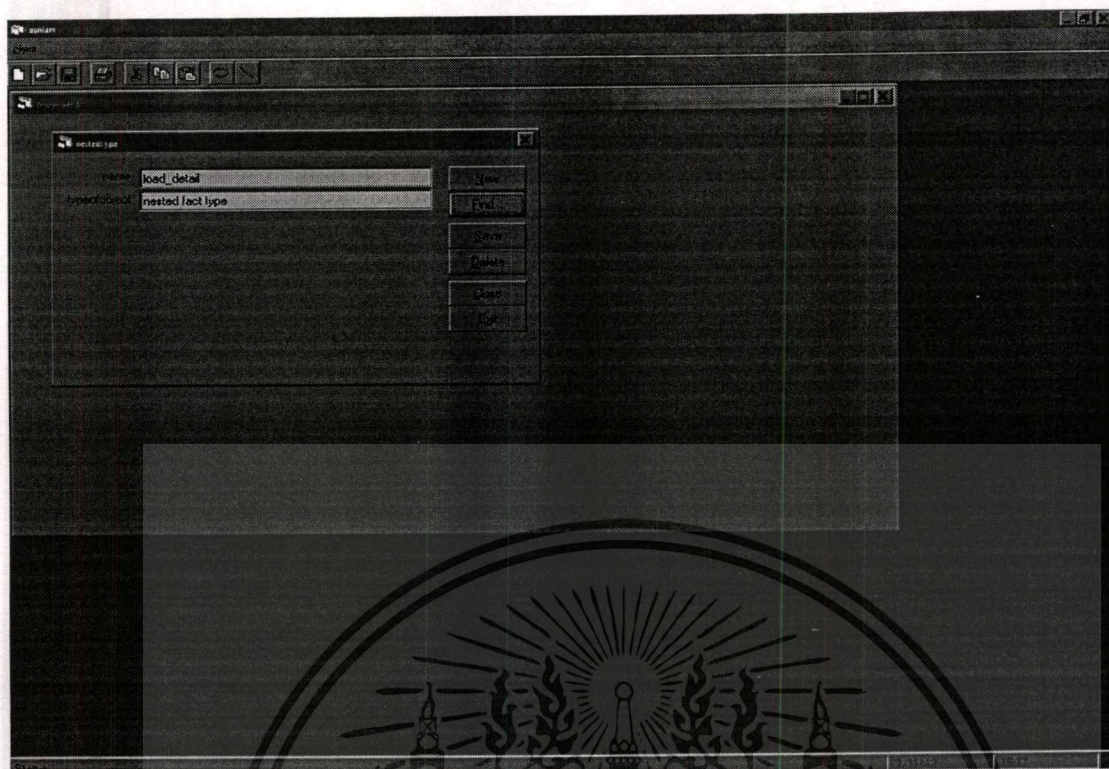
รูปที่ 6.18 ภาพแสดงการติดต่อกับฐานข้อมูลเชิงวัตถุ Cache



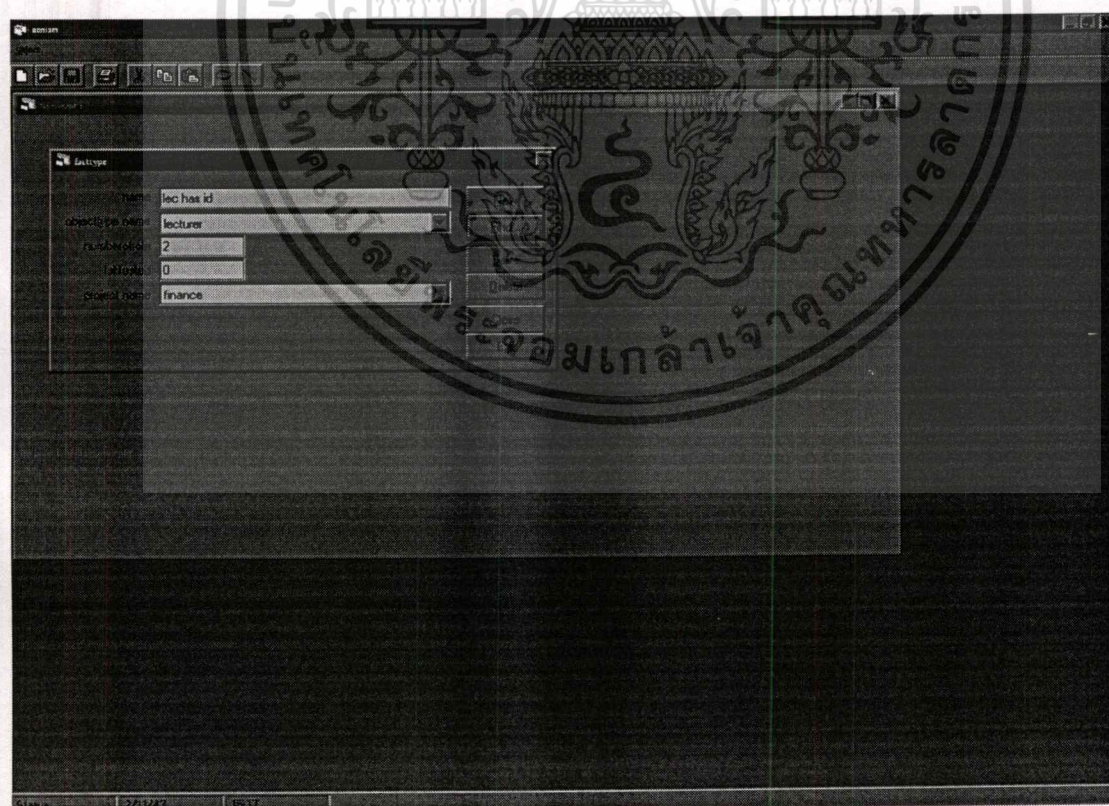
รูปที่ 6.19 ภาพแสดงการป้อนข้อมูลโครงการที่จะทำการออกแบบ

ในเครื่องมือที่ได้ทำการออกแบบมานั้นสามารถป้อนข้อมูลโดยแยกตามรายละเอียดต่าง ๆ โดยได้ทำการแบ่งออกเป็นหัวข้อตามเมนูที่ปรากฏในรูปที่ 6.20 โดยจะแสดงตัวอย่างในการป้อนข้อมูลรายละเอียดต่าง ๆ ในการออกแบบดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.22 ภาพแสดงตัวอย่างการป้อนข้อมูลของชนิดความจริงแบบเนส



รูปที่ 6.23 ภาพแสดงตัวอย่างการป้อนข้อมูลของชนิดความจริงเพื่อกำหนดความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 เปรียบเทียบวิธีการออกแบบฐานข้อมูลเชิงวัตถุ

การเปรียบเทียบวิธีการออกแบบฐานข้อมูลเชิงวัตถุจากเครื่องมือที่ใช้ในการออกแบบดังที่ได้กล่าวมาแล้วนั้น สามารถที่จะแยกพิจารณาในรายละเอียดต่าง ๆ ได้ดังนี้

1. ลักษณะของการออกแบบคลาส

จากเครื่องมือที่ใช้ในการออกแบบฐานข้อมูล 2 แบบแรกคือ Rational Rose และ Visio Professional นั้นจะพบว่าลักษณะในการออกแบบคลาสนั้นจะต้องทำการกำหนดตามลักษณะของคลาส ดังนั้นหากคลาสใด ๆ ที่สามารถกำหนดในลักษณะของรูปธรรม ซึ่งผู้ออกแบบสามารถทราบถึงรายละเอียดทั้งหมดของคลาสได้ จะสามารถทำการกำหนดเป็นคลาสได้อย่างง่าย เช่น คลาส Person, Book, User เป็นต้น โดยผู้ออกแบบสามารถกำหนดส่วนประกอบต่าง ๆ ของคลาสตามที่ต้องการ แต่ถ้าหากคลาสใดเป็นลักษณะของการกระทำหรือในลักษณะของนามธรรม การกำหนดลักษณะของคลาสจะทำการกำหนดได้ค่อนข้างยาก และอาจมีข้อผิดพลาดในการออกแบบได้ เช่น คลาส circulation, transaction banking เป็นต้น แต่หากผู้ออกแบบนำแบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM มาทำการออกแบบ จะไม่พบปัญหาดังกล่าวเพราะจะเป็นพิจารณารายละเอียดของสิ่งที่ต้องการออกแบบในระดับแอตทริบิวต์ และเป็นการกำหนดความสัมพันธ์ในแต่ละแอตทริบิวต์แล้วจึงใช้อัลกอริทึมในการแปลงแอตทริบิวต์และความสัมพันธ์ทั้งหมด รวมทั้งทำการจัดกลุ่มให้อยู่ในรูปแบบของคลาสที่เหมาะสม จะทำให้ผู้ใช้สามารถสร้างคลาสได้ถูกต้อง และมีรายละเอียดที่สมบูรณ์

2. ลักษณะของการกำหนดกฎข้อบังคับความถูกต้อง

3. ในเครื่องมือที่ใช้ในการออกแบบฐานข้อมูล 2 แบบแรกดังที่ได้กล่าวมาแล้วนั้น จะต้องทำการกำหนดรูปแบบของกฎข้อบังคับความถูกต้องของข้อมูลในลักษณะของเมธอดของแต่ละคลาส ดังนั้นการสร้างกฎข้อบังคับเพื่อดูแลความถูกต้องของข้อมูลทั้งหมดที่มีอยู่ในระบบครบถ้วนนั้น ต้องใช้เวลาในการออกแบบค่อนข้างมาก แต่ในการใช้แบบจำลองฐานข้อมูลเชิงแนวคิด OONIAM นั้น สามารถทำให้ผู้ออกแบบมองเห็นรูปแบบของกฎข้อบังคับความถูกต้องของข้อมูลได้อย่างละเอียด เพราะสามารถกำหนดกฎดังกล่าวที่เกี่ยวข้องกับทุก ๆ แอตทริบิวต์ในการออกแบบได้โดยตรง ทำให้การสร้างคลาสและการสร้างเมธอดของคลาสนั้นเป็นไปอย่างถูกต้อง

สรุปผลการวิจัยและข้อเสนอแนะ

7.1 สรุปผลการวิจัย

ในวิทยานิพนธ์ฉบับนี้ ได้มีการนำเสนอแนวทางในการจัดการกับระบบสารสนเทศที่มีการเปลี่ยนแปลงสูง โดยส่วนมากการเปลี่ยนแปลงดังกล่าวมักจะเกิดขึ้นกับกฎข้อบังคับความต้องการของข้อมูล อาจทำให้มีผลกระทบต่อโครงสร้างของข้อมูลด้วยและในการดำเนินการแก้ไขเพื่อให้ระบบสารสนเทศนั้นๆ สามารถใช้งานได้อย่างถูกต้องตลอดเวลานั้นเป็นเรื่องที่ค่อนข้างยุ่งยากและซับซ้อนมาก เนื่องจากมักจะมีการเก็บความสัมพันธ์ของโครงสร้างของข้อมูลที่ซับซ้อนไว้ในรูปแบบของตารางที่ใช้ในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ และความสัมพันธ์ดังกล่าวอันก่อให้เกิดกฎข้อบังคับความต้องการของข้อมูลนั้นมักจะทำการจัดเก็บในรูปของโค้ดของโปรแกรมต่าง ๆ จึงเป็นการยากที่จะทำการแก้ไข

การแก้ปัญหาสิ่งที่เกิดขึ้น นั้นได้นำเสนอโดยทำการแก้ไขที่การออกแบบฐานข้อมูลเนื่องจากในปัจจุบันได้มีการนำการออกแบบเชิงวัตถุมาใช้งานกันอย่างแพร่หลาย ซึ่งสามารถรองรับข้อมูลที่มีความซับซ้อนมากๆ ได้เป็นอย่างดีรวมถึงการออกแบบเชิงวัตถุนั้นมีประโยชน์อย่างมากในการใช้งานที่ต้องมีการนำมาปรับปรุงหรือแก้ไขโดยจะไม่กระทบถึงโครงสร้างของข้อมูลและสามารถแก้ไขได้อย่างง่ายเพราะจะจัดการโครงสร้างของข้อมูลในลักษณะของวัตถุซึ่งรวมทั้งข้อมูลและพฤติกรรมที่อาจเกิดขึ้นเข้าไว้ด้วยกัน ดังนั้นหากมีการแก้ไขกฎข้อบังคับความต้องการของข้อมูลใด ๆ ก็สามารที่จะแก้ไขจากพฤติกรรมของวัตถุนั้น ๆ ได้

ในวิทยานิพนธ์ได้มีการนำแบบจำลองเชิงวัตถุระดับแนวคิดในแอม ซึ่งใช้งานกันอย่างแพร่หลายในการออกแบบฐานข้อมูลเชิงวัตถุมาทำการปรับปรุง โดยเพิ่มส่วนประกอบต่าง ๆ เพื่อให้แบบจำลองดังกล่าวสามารถอธิบายแนวคิดในการออกแบบฐานข้อมูลเชิงวัตถุได้ และทำการเรียกแบบจำลองนั้นว่า แบบจำลองระดับแนวคิด OONIAM และได้ทำการสร้างอัลกอริทึมในการแปลงจากฐานข้อมูลดังกล่าวขึ้นมา โดยการแปลงนั้นสามารถนำมาใช้ในการแปลงแบบจำลองระดับแนวคิดในแอมเดิม ซึ่งเป็นการออกแบบฐานข้อมูลเชิงสัมพันธ์ให้อยู่ในรูปของฐานข้อมูลเชิงวัตถุได้ และยังสามารถสร้างฐานข้อมูลเชิงวัตถุได้โดยใช้แบบจำลองระดับแนวคิด OONIAM ที่สร้างขึ้นใหม่ได้ โดยไม่ต้องศึกษาความรู้ความเข้าใจเพิ่มเติมในส่วนของการออกแบบเชิงวัตถุมากนัก

นอกจากนี้ได้ทำการสร้างระบบอัตโนมัติ เพื่อรองรับกับการออกแบบแบบจำลองระดับแนวคิด OONIAM ที่สร้างขึ้นใหม่ด้วย เพื่อนำมาใช้เป็นเครื่องมือในการแปลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เชิงสัมพันธ์เดิมที่ใช้งานอยู่ในระบบสารสนเทศส่วนมากที่มีอยู่ ให้อยู่ในรูปแบบของฐานข้อมูลเชิงวัตถุ เพื่อให้การจัดการระบบสารสนเทศดังกล่าวสามารถทำงานได้สะดวกมากยิ่งขึ้น

7.2 ข้อเสนอแนะ

ในส่วนของการวิทยานิพนธ์ฉบับนี้ได้ทำการศึกษาไว้นั้น เป็นเพียงส่วนที่เกี่ยวข้องกับฐานข้อมูลของระบบสารสนเทศที่ใช้งานกันอยู่ในปัจจุบัน ดังนั้นหากต้องการให้ระบบอัตโนมัติที่ใช้งานนี้สามารถนำมาใช้งานได้สะดวกมากยิ่งขึ้นและในทุกด้าน ควรมีการศึกษาเพิ่มเติมในส่วนของการจัดการกับโค้ดที่มีอยู่ในระบบสารสนเทศให้สามารถแปลงให้อยู่ในลักษณะเชิงวัตถุได้อย่างอัตโนมัติ และควรมีการปรับปรุงระบบในส่วนของการติดต่อกับผู้ใช้ให้มีส่วนของการดึงข้อมูลแบบจำลองที่ได้เคยสร้างไว้แล้วมาทำการจัดเก็บบนระบบอัตโนมัติได้ทันที โดยที่ไม่ต้องทำการป้อนข้อมูลเข้าสู่ระบบใหม่ เพื่อให้มีความสะดวกสบายต่อผู้ใช้และสามารถใช้เป็นงานการซ้ำได้อีกทางหนึ่งด้วย



เอกสารอ้างอิง

- [1] Nijssen G.M., Halpin, T.A. **Conceptual Schema and Relational Database Design**. New York : Prentice Hall. 1989.
- [2] Jeffery D. Ullman, Jennifer Widom. **A First Course in Database Systems**. New Jersey : Prentice-Hall, Inc. 1997.
- [3] Ramez Elmasri, Shamkant B. Navathe. **Fundamentals of Database Systems**. 2nd Ed. Redwood City, CA : Benjamin/Cummings. 1994.
- [4] Loucopoulos P., Zicari, R. **Conceptual Model, Databases, and Case**. New York : Wiley Professional Computing. 1992.
- [5] Dillon G. T. **Object-Oriented Conceptual Model**. New York : Prentice-Hall. 1993.
- [6] Pierre Muller, Alain Muller. **Instant UML**. Paris : Wrox Press Ltd. 1997.
- [7] Terry Quatrani. **Visual Modeling with Rational Rose and UML**. Addison Wesley Longman, Inc. 1998.
- [8] Larman C. **Applying UML and patterns : an introduction to object-oriented analysis and design**. Upper Saddle River, NJ : Prentice Hall PTR. 1998.
- [9] Schach R.S. **Classical and object-oriented software engineering with UML and Java**. 4th Ed. Boston : McGraw-Hill. 1999.
- [10] Cantor M. **Object-oriented project management with UML**. New York : John Wiley. 1998.
- [11] Michael B., William Premerlani. **Object-Oriented Modeling and Design for Database Applications**. New Jersey : Prentice-Hall. 1998.
- [12] Setrag Khoshafian. **Object Oriented Databases**. New York : John Wiley & Sons, Inc. 1993.
- [13] Elisa Bertino, Lorenzo Martina. **Object-Oriented Database Systems Concepts and Architectures**. Wokingham : Addison-Wesley Publishing Company. 1993.
- [14] Michael Stonebraker, Paul Brown. **Object-Relational DBMSs**. California : Morgan Kaufmann Publishers, Inc. 1999.
- [15] Alfons Kemper, Guido Moerkotte. **Object-Oriented Database Management**. New Jersey : Prentice Hall, Inc. 1994.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [16] John G. Hughes. **Object-Oriented Databases**. New Jersey : Prentice-Hall, Inc. 1991.
- [17] Alan Dennis, Barbara H. Wixom. **Systems Analysis and Design**. New York : John Wiley & Sons, Inc. 1990.
- [18] Joseph Schmuller. **UML in 24 Hours**. New Delhi : Techmedia. 1999.
- [19] มยุรี เลิศเวชกุล. “ระบบการสร้างกฎข้อบังคับความต้องการของข้อมูลโดยใช้ภาษาธรรมชาติ สำหรับฐานข้อมูลเชิงสัมพันธ์” วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2540.
- [20] วีระนุช จันทรไพฑูรย์. “การแยกประเภทการบังคับความต้องการของข้อมูลสำหรับระบบฐานข้อมูลรีเลชันแนล” วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2540.
- [21] พันตรีหญิงรัชนิพร ผดุงผล. “เครื่องมือออกแบบฐานข้อมูลเชิงวัตถุโดยใช้ NIAM++” วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2538.

ประวัติผู้เขียน

ชื่อผู้เขียน	น.ส.สุธีรา พันธุ์ธีรานุรักษ์
วันเดือนปีเกิด	วันพฤหัสบดีที่ 22 มกราคม พ.ศ. 2513
สถานที่เกิด	วชิรพยาบาล กรุงเทพฯ
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตร์บัณฑิต เกียรตินิยมอันดับสอง สาขาวิชาคณิตศาสตร์ประยุกต์
สถานที่สำเร็จการศึกษา	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีที่สำเร็จการศึกษา	2534
ผลงานทางวิชาการที่ได้รับการตีพิมพ์	NIAM2OO สำหรับแปลงฐานข้อมูล ดีพิมพ์ในวารสาร “วิศวกรรมสารมก.” ฉบับที่ 41
ประสบการณ์การทำงาน	เจ้าหน้าที่แผนกระบบสารสนเทศและ หัวหน้าแผนกระบบ สารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อาชีพปัจจุบัน	รับราชการตำแหน่งอาจารย์ระดับ 3 ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง
บทความวิชาการที่ได้รับการตีพิมพ์	
1. สุธีรา พันธุ์ธีรานุรักษ์, ศุภมิตร จิตตะยโสธร. “NIAM2OO สำหรับแปลงฐานข้อมูล”. วิศวกรรม สาร มก. ฉบับที่ 41 ปีที่ 14 เดือน สิงหาคม – พฤศจิกายน 2543	