การประยุกต์วิธี SSADM ภายใต้ FOUNTAIN MODEL

APPLIED SSADM WITHIN FOUNTAIN MODEL

พัฒนพงษ์ ฉันทมิตรโอภาส

PATTANAPONG CHANTAMIT-O-PAS

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2545

ISBN 974-324-075-6

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์วิธี SSADM ภายใต้ FOUNTAIN MODEL

APPLIED SSADM WITHIN FOUNTAIN MODEL

พัฒนพงษ์ ฉันทมิตรโอภาส

PATTANAPONG CHANTAMIT-O-PAS

เลขหมู่.....
เลขทะเบียน 44089
วัน, เดือน, ปี 28 ต.ค. 2545

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2545

ISBN 974-324-075-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLIED SSADM WITHIN FOUNTAIN MODEL

PATTANAPONG CHANTAMIT-O-PAS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2002

ISBN 974-324-075-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2002

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การประยุกต์วิธี SSADM ภายใต้ FOUNTAIN MODEL

APPLIED SSADM METHODOLOGY WITHIN FOUNTAIN MODEL

ชื่อนักศึกษา นายพัฒนพงษ์ ฉันทมิตรโอภาส

รหัสประจำตัว 38626056

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.เอื้อน ปิ่นเงิน

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.เอื้อน	ปิ่นเงิน	
รศ.ดร.ศุภมิตร	จิตตะยโศธร	
รศ.ดร.วิเชียร	เปรมชัยสวัสดิ์	
ผศ.ดร.ประจวบ	วานิชชัชวาล	
ดร.พรฤดี	เนติโสภากุล	

วัน/เดือน/ปี ที่สอบ 30 กันยายน 2545 เวลา 10.00 น. เป็นต้นไป

สถานที่สอบ ณ ห้อง M21 (ชั้นลอย) คณะเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

คณบดีบัณฑิตวิทยาลัย

วันที่.....14.....**เดือน**....ตุลาคม.....**พ.ศ**.....2545....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การประยุกต์วิธี SSADM ภายใต้ FOUNTAIN MODEL
นักศึกษา	นาย พัฒนพงษ์ ฉันทมิตรโอภาส
รหัสประจำตัว	38626056
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2545
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. เอื้อน ปิ่นเงิน

บทคัดย่อ

การพัฒนาระบบสารสนเทศส่วนใหญ่ที่ปฏิบัติอยู่ในปัจจุบันเป็นแบบโครงสร้าง ปัญหามักเกิดขึ้นจากการเปลี่ยนแปลงความต้องการของผู้ใช้ ทำให้ระบบที่มีอยู่ไม่สามารถพัฒนาและเปลี่ยนแปลงได้อย่างมีประสิทธิภาพ ดังนั้นจึงเกิดแนวคิดใหม่กับการพัฒนาระบบที่เรียกว่า การพัฒนาระบบสารสนเทศเชิงวัตถุ แต่ระบบเชิงวัตถุในปัจจุบันยังไม่มีรูปแบบแน่นอน การวิเคราะห์และออกแบบยังไม่มีขั้นตอนที่ชัดเจน ทำให้ผู้ที่พัฒนาด้วยวิธีดังกล่าวไม่สามารถเห็นถึงขั้นตอนของการพัฒนาได้เหมือนกับการพัฒนาแบบใช้โครงสร้างที่ปฏิบัติอยู่ งานวิจัยนี้เป็นการนำเสนอขั้นตอนการพัฒนาระบบเชิงวัตถุโดยจัดหารูปแบบที่เหมาะสม ด้วยการนำขั้นตอนของการออกแบบแบบโครงสร้างและแบบเชิงวัตถุ มาผสมผสานกันภายใต้แบบจำลองน้ำพุเพื่อให้ได้ขั้นตอนการพัฒนาระบบสารสนเทศเชิงวัตถุที่ชัดเจนและมีประสิทธิภาพยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Applied SSADM within Fountain Model
Student	Mr. Pattanapong Chantamit-o-pas
Student ID.	38626056
Degree	Master of Science
Programme	Information Technology
Year	2002
Thesis Advisor	Assoc. Prof. Dr. Ouen Pin-ngern

ABSTRACT

At the present, most system developers conducted system development using structured approaches. The problem of using such approaches usually caused by changing in user requirements, resulting in inefficient system modification. New approach in system development called "Object-Oriented" is therefore introduced. However, this approach is somewhat "unmature" due to an unsystematic and unclear analysis and design steps. System analysts who are interested in this approach are unable to view each step of the development. This research aims to present a clearer view of system development steps by combining "Object-Oriented" system development approach and "Structured" approaches under fountain model for more precise and more efficient steps of system development.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดีจากความกรุณาของ รองศาสตราจารย์ ดร.เอื้อน ปิ่นเงิน ซึ่งเป็นอาจารย์ควบคุมวิทยานิพนธ์ฉบับนี้ ที่คอยให้คำปรึกษา คำแนะนำและแนะแนวทางแก้ปัญหาของงานวิจัยรวมทั้งวิชาการต่าง ๆ ที่เกี่ยวข้อง ซึ่งข้าพเจ้าซาบซึ้งในความอนุเคราะห์ของท่านและกราบขอบพระคุณเป็นอย่างสูง

กราบขอบพระคุณบิดา-มารดา และญาติพี่น้องของข้าพเจ้า ซึ่งให้กำลังใจและสนับสนุนทุนทรัพย์สำหรับการทำวิทยานิพนธ์มาโดยตลอด

ขอขอบคุณ Prof. Dr. Brain Henderson-Sellers แห่งมหาวิทยาลัย Technology Sydney (University of Technology, Sydney : UTS) ประเทศออสเตรเลีย ที่คอยให้คำแนะนำเกี่ยวกับแบบจำลองน้ำพุและแนวคิดต่าง ๆ ที่เกี่ยวข้อง

ขอขอบคุณ คุณเยาวภา บุญทรงฤทธิ ที่คอยสนับสนุนข้อมูลและให้กำลังใจสำหรับการทำวิทยานิพนธ์ให้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณ คุณวรางคณา เงินแก้ว คุณปิยะวรรณ กระแสร์สุนทร คุณนิภาพร ประภาศิริ คุณพิบูล เทพบุตร และคุณวศิน เสงี่ยมกุล ที่คอยให้คำปรึกษาเกี่ยวกับโปรแกรมของระบบคลังข้อสอบ ให้สามารถทำงานได้อย่างถูกต้องและตรงตามวัตถุประสงค์

ขอขอบคุณ คุณธีระณัฐ เริ่มคิดการ และเจ้าหน้าที่สำนักวิจัยและบริการคอมพิวเตอร์ที่เกี่ยวข้องที่เอื้อเฟื้อสถานที่ทำงานวิจัยของข้าพเจ้า

ขอขอบคุณ เพื่อน ๆ พี่ ๆ วิทยาการสารสนเทศ รุ่นที่ 1 รวมทั้งน้อง ๆ ทุกคนในคณะเทคโนโลยีสารสนเทศ ที่ให้กำลังใจและคำปรึกษาตลอดมา

ขอขอบคุณเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศที่คอยอำนวยความสะดวกเกี่ยวกับอุปกรณ์ตำราเรียนและงานวิจัยที่เกี่ยวข้อง

ขอขอบคุณบุคคลต่าง ๆ ที่มิได้เอ่ยนามทุกท่าน ซึ่งคอยให้กำลังใจและแนวคิดกับวิทยานิพนธ์ฉบับนี้

สุดท้ายขอขอบคุณบัณฑิตวิทยาลัย ที่ให้ทุนสนับสนุนการทำวิทยานิพนธ์ครั้งนี้

คุณค่าและประโยชน์ที่พึงได้จากวิทยานิพนธ์นี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

พัฒนพงษ์ ฉันทมิตรโอภาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VIII
สารบัญรูป XIII
บทที่
1. บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ 5
1.3 ขอบเขตงานวิจัย 5
1.4 ขั้นตอนการดำเนินงานวิจัย 8
1.5 สรุปงานวิจัยที่เกี่ยวข้อง 8
1.6 ข้อจำกัดของงานวิจัย 11
1.7 ส่วนประกอบวิทยานิพนธ์ 12

2. วิธีเชิงโครงสร้าง 13
2.1 วิธีการเชิงโครงสร้าง 13
2.1.1 หลักการของวิธี SSADM 14
2.2 โครงสร้างของวิธี SSADM 15
2.3 ขั้นตอนของ SSADM 15
2.4 สรุป 40

3. วิธีเชิงวัตถุ 41
3.1 แนวคิดของวิธีเชิงวัตถุ 41
3.2 การวิเคราะห์เชิงวัตถุ 41
3.3 แบบจำลองวัตถุ 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3.1 แบบจำลองไดนามิค 46
3.3.2 แบบจำลองฟังก์ชัน 63
3.4 การออกแบบเชิงวัตถุ 66
3.4.1 ขั้นตอนการออกแบบเชิงวัตถุ 66
3.5 สรุป 77

4. แบบจำลองน้ำพุ 78
4.1 ลักษณะของแบบจำลองน้ำพุ 80
4.1.1 วงจรการผลิต 80
4.1.2 วงจรการปฏิบัติการ 81
4.1.3 โมเสสกับวงจรการผลิตและวงจรการปฏิบัติงาน 82
4.2 สัญลักษณ์ที่ใช้ในแบบจำลองน้ำพุ 84
4.2.1 คลาสและแบบจำลองวัตถุ 84
4.2.2 การกำหนดรายละเอียดของคลาส 92
4.2.3 การกำหนดรายละเอียดของเซอร์วิส 95
4.2.4 อีเวนท์โมเดล 96
4.2.5 ออบเจ็กต์ชาร์ต 97
4.2.6 โมเดลการสืบทอดคุณสมบัติ 100
4.2.7 เทคนิคการจัดการความซับซ้อน 102
4.3 กิจกรรมในแบบจำลองน้ำพุ 108
4.4 สรุป 132

5. SSADM ภายใต้แบบจำลองน้ำพุ 133
5.1 หลักการของแบบจำลองประยุกต์ 134
5.1.1 ระดับของระบบ 134
5.1.2 ระดับของระบบย่อย 135
5.2 ลักษณะของแบบจำลองประยุกต์ 137

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

5.3 วิธีการของแบบจำลองประยุกต์ 137
5.4 โครงสร้างของแบบจำลองประยุกต์ 137
5.5 ขั้นตอนและกิจกรรมของการพัฒนาระบบภายใต้แบบจำลองประยุกต์ 139
5.5.1 โมดูลศึกษาความเหมาะสมและความต้องการของผู้ใช้ 140
5.5.2 โมดูลการวิเคราะห์ 169
5.5.3 โมดูลการออกแบบ 190
5.5.4 โมดูลอิมพลีเมนต์และการทดสอบ 209
5.5.5 System Support Module 221
5.6 สรุป 227

6. กรณีศึกษาระบบคลังข้อสอบ 228
6.1 ระยะที่ 0 228
6.2 ระยะที่ 1 259
6.3 ระยะที่ 2 264
6.4 ระยะที่ 3 270
6.5 ระยะที่ 4 332
6.6 ระยะที่ 5 337
6.7 ระยะที่ 6 349
6.8 ระยะที่ 7 356
6.9 ระยะที่ 8 358
6.10 ระยะที่ 9 361
6.11 ระยะที่ 10 362

7. บทสรุปและข้อเสนอแนะ 363
7.1 บทสรุป 363
7.2 ข้อเสนอแนะ 365
7.3 แนวทางการวิจัยต่อ 366

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

เอกสารอ้างอิง 367

ภาคผนวก 371

ภาคผนวก ก. ระบบคลังข้อสอบ (ต้นแบบ) ตามกรณีศึกษา 372

ภาคผนวก ข. บทความที่ตีพิมพ์ในวารสาร 386

ภาคผนวก ค. สรุปขั้นตอนและกิจกรรมของแบบจำลองประยุกต์ 398

ประวัติผู้เขียน 419

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

1.1 แสดงการเปรียบเทียบวงชีวิตของโมเดล 3
1.2 แสดงการเปรียบเทียบแนวคิดของวิธีเชิงวัตถุ 4
1.3 แสดงการเปรียบเทียบเทคนิคของวิธีเชิงวัตถุ 5
5.2 แสดงความต้องการของผู้ใช้ 146
5.3 แสดงตารางการทำงานของผู้ใช้ (User Catalogue) 148
5.4 แสดงรายละเอียดของ Use Case 149
5.5 แสดงทางเลือกทางธุรกิจ (BSO Matrix) 153
5.6 แสดงตารางรายละเอียดของการบวนการย่อย 159
5.7 แสดงความสัมพันธ์ระหว่างเอนติตีและกระบวนการ 161
5.8 แสดงรายละเอียดของเอนติตี 162
5.9 แสดงรายละเอียดแอตทริบิวต์ของแต่ละเอนติตี 163
5.10 แสดงรายละเอียดของฟังก์ชันเก็บตัวข้อสอบ 172
5.11 แสดงความสัมพันธ์ระหว่างฟังก์ชันกับการทำงานของผู้ใช้ 173
5.12 แสดงรายละเอียดของคลาส/วัตถุ 184
5.13 แสดงรายละเอียดการตรวจสอบการออกแบบเชิงวัตถุ 207
5.14 แสดงรายละเอียดการวัดความสัมพันธ์ระหว่างวัตถุ 208
5.15 แสดงรายละเอียดของการติดตั้งโปรแกรม 212
5.16 แสดงการวัดความซับซ้อนของฟังก์ชันภายในวัตถุ 215
6.1 แสดงรายละเอียด Context Diagram ของระบบเดิมทางกายภาพ 231
6.2 แสดงรายละเอียดการไหลเวียนของเอกสารของระบบเดิม 231
6.3 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 1 234
6.4 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 2 235
6.5 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 3 236
6.6 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 4 237
6.7 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 5 238
6.8 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 6 239
6.9 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 7 240

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

6.10 แสดงรายละเอียดของ Context Diagram ส่วนความต้องการของระบบเดิม 241
6.11 แสดงการไหลเวียนของเอกสารของระบบที่ต้องการ 242
6.12 แสดงรายละเอียดของ User Catalogue 243
6.13 แสดงรายละเอียดของกระบวนการที่ 1.1 245
6.14 แสดงรายละเอียดของกระบวนการที่ 1.2 245
6.15 แสดงรายละเอียดของกระบวนการที่ 1.3 245
6.16 แสดงรายละเอียดของกระบวนการที่ 2.1 245
6.17 แสดงรายละเอียดของกระบวนการที่ 2.2 246
6.18 แสดงรายละเอียดของแผนงานลำดับที่ 1 247
6.19 แสดงรายละเอียดของแผนงานลำดับที่ 2 248
6.20 แสดงรายละเอียดของแผนงานลำดับที่ 3 249
6.21 แสดงรายละเอียดของแผนงานลำดับที่ 4 250
6.22 แสดงตารางระยะเวลาดำเนินงานการพัฒนาระบบคลังข้อสอบ 251
6.23 แสดงความสัมพันธ์ของ BSO 254
6.24 แสดงตารางงบประมาณ แบบที่ 1 256
6.25 แสดงตารางงบประมาณ แบบที่ 2 257
6.26 แสดงตารางงบประมาณ แบบที่ 3 258
6.27 แสดงความสัมพันธ์ของกระบวนการกับเอนติตีในระบบเดิม 262
6.28 แสดงรายละเอียดของกระบวนการที่ 1.1 274
6.29 แสดงรายละเอียดของกระบวนการที่ 1.2 274
6.30 แสดงรายละเอียดของกระบวนการที่ 1.3 274
6.31 แสดงรายละเอียดของกระบวนการที่ 2.1 275
6.32 แสดงรายละเอียดของกระบวนการที่ 2.2 275
6.33 แสดงรายละเอียดของกระบวนการที่ 3.1 275
6.34 แสดงรายละเอียดของกระบวนการที่ 3.2 275
6.35 แสดงรายละเอียดของกระบวนการที่ 3.3 276
6.36 แสดงรายละเอียดของเอนติตีอาจารย์ 278

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

6.37 แสดงรายละเอียดของเอนติตีประวัติการออกข้อสอบ 279
6.38 แสดงรายละเอียดของเอนติตีข้อสอบ 280
6.39 แสดงรายละเอียดของแอตทริบิวต์รหัสข้อสอบ 281
6.40 แสดงรายละเอียดของแอตทริบิวต์รหัสวิชา 282
6.41 แสดงรายละเอียดของแอตทริบิวต์บทที่ 283
6.42 แสดงรายละเอียดของแอตทริบิวต์เนื้อหา 284
6.43 แสดงรายละเอียดของแอตทริบิวต์โจทย์ 285
6.44 แสดงรายละเอียดของแอตทริบิวต์ตัวเลือก 286
6.45 แสดงรายละเอียดของแอตทริบิวต์ความยาก 287
6.46 แสดงรายละเอียดของแอตทริบิวต์ความตรง 288
6.47 แสดงรายละเอียดของแอตทริบิวต์ความน่าเชื่อถือ 289
6.48 แสดงรายละเอียดของแอตทริบิวต์เวลาสอบ 290
6.49 แสดงรายละเอียดของแอตทริบิวต์เฉลย 291
6.50 แสดงรายละเอียดของแอตทริบิวต์วันที่ออกข้อสอบ 292
6.51 แสดงรายละเอียดของแอตทริบิวต์วันที่ให้เกณฑ์ข้อสอบ 293
6.52 แสดงรายละเอียดของแอตทริบิวต์วันที่ใช้สอบครั้งสุดท้าย 294
6.53 แสดงรายละเอียดของแอตทริบิวต์ภาคการศึกษา 295
6.54 แสดงรายละเอียดของแอตทริบิวต์ปีการศึกษา 296
6.55 แสดงรายละเอียดของแอตทริบิวต์จำนวนครั้งที่ออกข้อสอบ 297
6.56 แสดงรายละเอียดของแอตทริบิวต์รหัสอาจารย์ 298
6.57 แสดงรายละเอียดของแอตทริบิวต์ชื่ออาจารย์ 299
6.58 แสดงรายละเอียดของแอตทริบิวต์ที่อยู่อาจารย์ 300
6.59 แสดงรายละเอียดของการทำบรรทัดฐานฐานข้อมูล 301
6.60 แสดงรายละเอียดของฟังก์ชันเก็บตัวข้อสอบ 304
6.61 แสดงรายละเอียดของฟังก์ชันเก็บเกณฑ์และรวบรม 305
6.62 แสดงรายละเอียดของฟังก์ชันจัดทำชุดข้อสอบ 306
6.63 แสดงรายละเอียดของฟังก์ชันลบข้อสอบที่ไม่ต้องการ 307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

6.64 แสดงรายละเอียดของฟังก์ชันสอบถามข้อมูล 308
6.65 แสดงรายละเอียดของฟังก์ชันเพิ่มอาจารย์และผู้ให้เกณฑ์ 309
6.66 แสดงรายละเอียดของฟังก์ชันแก้ไขข้อมูลอาจารย์และผู้ให้เกณฑ์ 310
6.67 แสดงรายละเอียดของอินพุต-เอาต์พุต 311
6.68 แสดงรายละเอียดของอินพุต-เอาต์พุต (ต่อ) 312
6.69 แสดงรายละเอียดของแผนภูมิ I/O ส่วนบันทึกข้อสอบ 315
6.70 แสดงรายละเอียดของแผนภูมิ I/O ส่วนบันทึกเกณฑ์และรวบรวมข้อสอบ 316
6.71 แสดงรายละเอียดของแผนภูมิ I/O ส่วนจัดทำข้อสอบ 317
6.72 แสดงรายละเอียดของแผนภูมิ I/O ส่วนการลบข้อสอบ 318
6.73 แสดงรายละเอียดของตารางความสัมพันธ์ระหว่างฟังก์ชันกับผู้ใช้ 319
6.74 แสดงรายละเอียดเส้นทางที่ 1 ของแบบจำลอง 322
6.75 แสดงรายละเอียดของความสัมพันธ์ระหว่างเหตุการณ์กับเอนติตี 323
6.76 แสดงโอเปอร์เรชั่นของแต่ละเอนติตี 326
6.77 แสดงรายละเอียดของวัตถุอาจารย์ 335
6.78 แสดงรายละเอียดของวัตถุข้อสอบ 335
6.79 แสดงรายละเอียดของวัตถุประวัติการออกข้อสอบ 336
6.80 แสดงรายละเอียดของวัตถุชุดข้อสอบ 336
6.81 แสดงการเปรียบเทียบราคาของผู้เสนอขายสินค้า 348
6.82 แสดงรายละเอียดของฟังก์ชันเพิ่มข้อมูลของวัตถุข้อสอบ 350
6.83 แสดงรายละเอียดของฟังก์ชันปรับปรุงข้อมูลของวัตถุข้อสอบ 351
6.84 แสดงรายละเอียดของฟังก์ชันลบข้อมูลของวัตถุข้อสอบ 352
6.85 แสดงรายละเอียดของฟังก์ชันค้นหาข้อมูลของวัตถุข้อสอบ 353
6.86 แสดงรายละเอียดของการวัดความซับซ้อนของวัตถุ 354
6.87 แสดงรายละเอียดการวัดความสัมพันธ์ระหว่างวัตถุ 355
6.88 แสดงการวัดความซับซ้อนของฟังก์ชันภายในวัตถุ 356
6.89 แสดงรายละเอียดของการติดตั้งโปรแกรม 357
6.90 แสดงรายละเอียดของการตรวจสอบซอฟต์แวร์ 358

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

6.91 แสดงรายละเอียดของวิธีการติดตั้งระบบ 359
6.92 แสดงรายละเอียดของการติดตั้งซอฟต์แวร์ฐานข้อมูล 360
6.93 แสดงรายละเอียดของปัญหาที่เกิดขึ้นของผู้ใช้ 361
6.94 แสดงรายละเอียดความต้องการการปรับปรุงระบบจากผู้ใช้ 362

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

1.1 ก แสดงส่วนหลักของวงจรชีวิตซอฟต์แวร์ 7
1.1 ข แสดงรายละเอียดของวงจรชีวิตซอฟต์แวร์ 7
2.1 แสดงโครงสร้างของวิธี SSADM 15
2.2 แสดงระยะของ SSADM 16
2.3 แสดงขั้นตอนการทำงานของระยะที่ 0 17
2.4 แสดงขั้นตอนการทำงานของระยะที่ 1 20
2.5 แสดงขั้นตอนการทำงานของระยะที่ 2 24
2.6 แสดงขั้นตอนการทำงานของระยะที่ 3 25
2.7 แสดงขั้นตอนการทำงานของระยะที่ 4 30
2.8 แสดงขั้นตอนการทำงานของระยะที่ 5 32
2.9 แสดงขั้นตอนการทำงานของระยะที่ 6 35
3.1 ก สัญลักษณ์ของคลาส 42
3.1 ข ตัวอย่างของคลาส Person 42
3.2 สัญลักษณ์ที่ใช้แทนมัลติพลิซิตีของแอสโซซิเอชั่น 42
3.3 ก สัญลักษณ์ในการระบุบทบาทของคลาสในความสัมพันธ์ระหว่างคลาส 43
3.3 ข ตัวอย่าง ความสัมพันธ์ระหว่างคลาสที่มีการระบุบทบาท 43
3.4 ก สัญลักษณ์ที่ใช้แทนลิงค์แอตทริบิวต์ 43
3.4 ข ตัวอย่างความสัมพันธ์ระหว่างคลาสที่ทำให้เกิดลิงค์แอตทริบิวต์ 43
3.5 ก สัญลักษณ์ที่ใช้แทนแอสโซสิเอชั่นคลาส 44
3.5 ข ตัวอย่างความสัมพันธ์ระหว่างคลาสที่ทำให้เกิดแอสโซสิเอชั่นคลาส 44
3.6 ก สัญลักษณ์ที่ใช้แทนควอลิฟายด์แอสโซสิเอชั่น 44
3.6 ข ตัวอย่างของควอลิฟายด์แอสโซสิเอชั่น 44
3.7 ก สัญลักษณ์ที่ใช้แทนการถ่ายทอดคุณสมบัติ 45
3.7 ข ตัวอย่างการถ่ายทอดคุณสมบัติระหว่างคลาส 45
3.8 ก สัญลักษณ์ที่ใช้แทนการประกอบกันของคลาส 46
3.8 ข ตัวอย่างการประกอบกันของคลาส 46
3.9 แสดงเหตุการณ์ของคลาสและแอตทริบิวต์ 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.10 แสดงเหตุการณ์ของสายโทรศัพท์ 49
3.11 แสดงสถานะของสายโทรศัพท์ 52
3.12 แสดงสเตทไดอะแกรมทางเดียวของเกมหมากรุก 52
3.13 แสดงการส่งผ่านค่าของสเตทไดอะแกรมพร้อมเงื่อนไข 53
3.14 แสดงการกระทำของเมนูแบบ POP-UP 54
3.15 แสดงโครงสร้างของสเตทไดอะแกรม 55
3.16 แสดงสเตทไดอะแกรมของเครื่องขายสินค้าประเภทหยอดเหรียญ 56
3.17 แสดงกิจกรรมภายในสถานะ dispense item ของเครื่องหยอดเหรียญ 56
3.18 แสดงสเตทไดอะแกรมของ Transition Select (item) ในเครื่องหยอดเหรียญ 57
3.19 แสดงสเตทไดอะแกรมของการเข้าเกียร์รถยนต์ในรูปของ Generalization 58
3.20 แสดงสเตทไดอะแกรมของส่วนประกอบรถยนต์ 59
3.21 แสดงสเตทไดอะแกรมแบบขนานของเกมข้ามสะพาน 60
3.22 แสดงการส่งผ่านการกระทำ 60
3.23 แสดงลำดับการทำงานของการกระทำภายในแต่ละสถานะ 61
3.24 แสดงการเขียนการกระทำภายในสเตทไดอะแกรม 62
3.25 แสดงการเรียกข้อมูลจากบัญชีลูกค้า 65
3.26 แสดงองค์ประกอบของคลาสที่ประกอบด้วยฟังก์ชันและแอตทริบิวต์ 69
3.27 แสดงความสัมพันธ์แบบลูกโซ่ของคลาสบริษัท พนักงานและความสามารถพิเศษ 70
3.28 แสดงดรรชนีความสามารถพิเศษให้กับคลาสพนักงาน 70
4.1 แสดงรูปทั่วไปของแบบจำลองน้ำพุ 78
4.2 แสดงขั้นตอนการพัฒนาของแต่ละระบบย่อยของแบบจำลองน้ำพุ 79
4.3 แสดงภาพรวมของโมเสสที่มีความสัมพันธ์ระหว่าง Product Lifecycle และ Process Lifecycle..... 80
4.4 ก แสดงการพัฒนาแบบจำลองน้ำพุของ Process และ Product Lifecycle 82
4.4 ข แสดงการพัฒนาในสภาพแวดล้อมแบบทำซ้ำของ Process และ Product Lifecycle . 82
4.5 แสดงการเชื่อมต่อระหว่างขั้นตอน กิจกรรมและวงจรชีวิตของโมเสส 83
4.6 ก แสดงสัญลักษณ์ของคลาส 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.6 ข แสดงสัญลักษณ์ของคลาสโดยแบ่งระหว่าง Properties และ Operation 85
4.7 แสดงสัญลักษณ์ของ Abstract Class 85
4.8 แสดงการทำงานของเซอร์วิสภายในคลาสเมื่อมีการเรียกใช้ 86
4.9 แสดงสัญลักษณ์ของคลาสไลบารี 86
4.10 แสดงสัญลักษณ์ของคลาสในลักษณ์ต่าง ๆ 86
4.11 แสดงสัญลักษณ์ของวัตถุ 87
4.12 ก แสดงสัญลักษณ์ของการสืบทอดแบบ Single Generalization 88
4.12 ข แสดงสัญลักษณ์ของการสืบทอดแบบหลายระดับ 88
4.12 ค แสดงสัญลักษณ์ของการสืบทอดแบบมีทางเลือก 88
4.12 ง แสดงสัญลักษณ์ของการสืบทอดแบบมาจากคลาสย่อย 88
4.12 จ แสดงสัญลักษณ์ของการสืบทอดที่มาจากคลาสย่อยแต่ระบุ Property 88
4.13 ก แสดงสัญลักษณ์ของการสืบทอดรูปแบบของการอิมพลีเมนต์แบบเดียว 89
4.13 ข แสดงสัญลักษณ์ของการสืบทอดรูปแบบของการอิมพลีเมนต์แบบหลายระดับ 89
4.13 ค แสดงสัญลักษณ์ของการสืบทอดรูปแบบของการอิมพลีเมนต์ในระดับคลาสย่อย 89
4.14 แสดงความสัมพันธ์ระหว่างคลาส 90
4.15 แสดงความสัมพันธ์แบบ Ternary ที่เกิดขึ้นในคลาส Playing-record 90
4.16 ก แสดงความสัมพันธ์แบบเดี่ยว 91
4.16 ข แสดงความสัมพันธ์แบบหลายคลาส 91
4.16 ค แสดงความสัมพันธ์แบบหลายคลาสและหลายทางเลือก 91
4.16 ง แสดงความสัมพันธ์ที่ระบุความสัมพันธ์ 91
4.17 ก แสดงสัญลักษณ์ของความสัมพันธ์แบบ Cardinalities แบบมีเงื่อนไข 1:M 92
4.17 ข แสดงสัญลักษณ์ของความสัมพันธ์แบบ Cardinalities แบบจำเป็นต้องมี 1:M 92
4.17 ค แสดงสัญลักษณ์ของความสัมพันธ์แบบ Cardinalities แบบ Association .....
Cardinality 92
4.18 แสดงรูปแบบเอกสารของ Class Specification โดยใช้โครงสร้างทางไวยากรณ์ ... 94
4.19 แสดงการทำงานของเซอร์วิสในคลาส Bank_Account 95
4.20 แสดงรายละเอียดของเซอร์วิส 96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.21 แสดงรูปแบบของการวาดอีเวนท์โมเดลพร้อมกับคำอธิบาย 97
4.22 ก แสดงออบเจ็กต์ชาร์ตของสเตทที่มีเงื่อนไข OR 99
4.22 ข แสดงออบเจ็กต์ชาร์ตของสเตทที่มีเงื่อนไข AND 99
4.23 แสดงรายละเอียดของทรานเซกชั่นของสเตทไดอะแกรม 99
4.24 แสดงการสืบทอดคุณสมบัติของโมเดลสืบทอดคุณสมบัติ 101
4.25 แสดงการแบ่งคลาสในรูปแบบของ Sheet (ส่วนแรก) 102
4.26 แสดงการแบ่งคลาสในรูปแบบของ Sheet (ส่วนหลัง) 103
4.27 แสดงเอกสารของระบบย่อย 104
4.28 แสดงการกระจายระบบย่อยใน Layer ที่ต่างกันของระบบ 105
4.29 แสดงการแบ่งเลเยอร์ของ O/C Model 106
4.30 แสดง O/C Model ที่มีสัญลักษณ์แบบ Selective Visibility 107
4.31 แสดงคลาสของรถไฟ รถจักรยาน และรถแทรกเตอร์ 111
4.32 แสดงคลาสทั้งสามที่ย้ายเซอร์วิสที่เหมือนกันไปสู่คลาสยานพาหนะ 112
4.33 แสดงคลาส CABLE_SHEATH ที่มีลักษณะเป็น Genericity 113
4.34 แสดงการคำนวณ fan-in และ fan-out 125
4.35 ก แสดงการกระจายคลาสของระบบย่อยไปสู่ระบบย่อยอื่น 129
4.35 ข แสดงการกระจาย Abstract ของคลาสในระบบย่อยไปสู่ระบบย่อยอื่น 129
5.1 แสดงการประยุกต์ของโมเดลในระดับของระบบ 135
5.2 แสดงการประยุกต์ของโมเดลในระดับระบบย่อย 136
5.3 แสดงโครงสร้างของ SSADM ภายใต้แบบจำลองน้ำพุ 138
5.4 แสดงภาพรวมของการประยุกต์แบบจำลองน้ำพุกับวิธี SSADM 139
5.5 แสดงเอนติตีภายนอกที่เกี่ยวข้องกับระบบ 142
5.6 แสดงกระบวนการทำงานของระบบ 142
5.7 แสดงสัญลักษณ์ของดาต้าสโตร์ 142
5.8 แสดงเส้นทิศทางการไหลของข้อมูล 143
5.9 แสดงเอนติตีของ LDS 143
5.10 แสดงความสัมพันธ์ของเอนติตี 144

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

5.11 แสดง Gantt Chart ของระบบ ITEM BANK 155
5.12 แสดงกระบวนการสุดท้าย (Elementary Process) 158
5.13 แสดงการกระจายกระบวนการของแผนภูมิการไหลของข้อมูลระดับที่ 1 158
5.14 แสดงความสัมพันธ์ระหว่างดาต้าสโตร์ทางตรรกะกับเอนติตี 165
5.15 แสดงเอนติตีไลฟ์ฮิสทรีของผู้ออกข้อสอบ 175
5.16 แสดงการทำงานของเอนติตีไลฟ์ฮิสทรีของผู้ออกข้อสอบที่มีลักษณะการทำงานซ้ำ 176
5.17 แสดงรายละเอียดเอนติตีไลฟ์ฮิสทรีของผู้ออกข้อสอบ 176
5.18 แสดงเหตุการณ์ของเอนติตี X 177
5.19 แสดงการใส่ลำดับสเตทของเอนติตีไลฟ์ฮิสทรี 177
5.20 แสดงการใส่ลำดับสเตทให้กับเอนติตีไลฟ์ฮิสทรี 178
5.21 แสดงการกำหนดทางเลือกให้กับเอนติตีไลฟ์ฮิสทรีพร้อมกับระบุลำดับของสเตท. 178
5.22 แสดงการระบุสเตทที่มี 2 สเตทในครั้งเดียวของเอนติตีไลฟ์ฮิสทรี 179
5.23 แสดง Effect Correspondent Diagram 179
5.24 แสดงสัญลักษณ์ของคลาส 183
5.25 แสดงสัญลักษณ์ของวัตถุ 183
5.26 ก แสดงการเปลี่ยนความสัมพันธ์แบบเอนติตีหลักและรองขึ้นตรงต่อกัน 186
5.26 ข แสดงการเปลี่ยนความสัมพันธ์แบบเอนติตีหลักไม่ขึ้นตรงกับเอนติตีรอง 186
5.27 ก แสดงการเปลี่ยนความสัมพันธ์แบบเอนติตีหลักและรองขึ้นตรงต่อกัน 187
5.27 ข แสดงการเปลี่ยนความสัมพันธ์แบบเอนติตีหลักไม่ขึ้นตรงกับเอนติตีรอง 187
5.28 ก แสดงการแปลงความสัมพันธ์ของเอนติตีหลักและรองกรณีที่ถ่ายทอดถึงกันแบบ ไม่มีเงื่อนไข 188
5.28 ข แสดงการแปลงความสัมพันธ์ของเอนติตีหลักและรองกรณีที่ถ่ายทอดถึงกันแบบ มีเงื่อนไข 188
5.29 แสดงองค์ประกอบของคลาสที่ประกอบด้วยโอเปเรชันและแอตทริบิวต์ 196
5.30 แสดงความสัมพันธ์แบบลูกโซ่ของคลาสบริษัท พนักงานและความสามารถพิเศษ 201
5.31 แสดงอินเด็กซ์ความสามารถพิเศษให้กับคลาสพนักงาน 201
5.32 แสดงการคำนวณ fan-in และ fan-out 205

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

6.1 แสดง Context Diagram ของระบบคลังข้อสอบ 231
6.2 แสดงรายละเอียดไดอะแกรมของการทำงานภายในระบบ 232
6.3 แสดง LDS ไดอะแกรมของระบบเดิม 232
6.4 แสดงความสัมพันธ์ระหว่างเอนติตีของ LDS ไดอะแกรม 233
6.5 แสดง Context Diagram ของระบบเดิม 241
6.6 แสดงการทำงานของเอนติตีภายนอกกับงานภายในระบบ 242
6.7 แสดงแผนภูมิการไหลของข้อมูลทางกายภาพระดับที่ 1 ของระบบเดิม 243
6.8 แสดงแผนภูมิการไหลของข้อมูลทางกายภาพระดับที่ 2 ของระบบเดิม 244
6.9 แสดงแผนภูมิการไหลของข้อมูลทางกายภาพระดับที่ 2 ของระบบเดิม (ต่อ) 244
6.10 แสดง USE CASE ของระบบคลังข้อสอบ 246
6.11 แสดงหน้าจอสำหรับการป้อนข้อมูลข้อสอบ 252
6.12 แสดงหน้าจอสำหรับการเลือกข้อสอบ 252
6.13 แสดงหน้าจอสำหรับพิมพ์ข้อสอบ 253
6.14 แสดงหน้าจอสำหรับการค้นหาข้อสอบตามบท 253
6.15 แสดงแผนภูมิการไหลของข้อมูลทางกายภาพระดับที่ 1 ของระบบเดิม 259
6.16 แสดงแผนภูมิการไหลของข้อมูลทางกายภาพระดับที่ 2 ของระบบเดิม 259
6.17 แสดงแผนภูมิการไหลของข้อมูลทางกายภาพระดับที่ 2 ของระบบเดิม (ต่อ) 260
6.18 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะระดับที่ 1 ของระบบเดิม 260
6.19 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะระดับที่ 2 ของระบบเดิม 261
6.20 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะระดับที่ 2 ของระบบเดิม (ต่อ) 261
6.21 แสดงแผนภูมิของโครงสร้างข้อมูลทางตรรกะของระบบเดิม 262
6.22 แสดงความสัมพันธ์ระหว่าง LDS กับเอนติตี 263
6.23 แสดงแผนภูมิการไหลของข้อมูลตามความต้องการของผู้ใช้ระดับที่ 1 270
6.24 แสดงแผนภูมิการไหลของข้อมูลตามความต้องการของผู้ใช้ระดับที่ 2 270
6.25 แสดงแผนภูมิการไหลของข้อมูลตามความต้องการของผู้ใช้ระดับที่ 2 (ต่อ) 271
6.26 แสดงแผนภูมิการไหลของข้อมูลตามความต้องการของผู้ใช้ระดับที่ 2 (ต่อ) 271
6.27 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะตามความต้องการของผู้ใช้ระดับที่ 1 272

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

6.28 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะตามความต้องการของผู้ใช้ระดับที่ 2 ของกระบวนการที่ 1 272
6.29 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะตามความต้องการของผู้ใช้ระดับที่ 2 ของกระบวนการที่ 2 273
6.30 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะตามความต้องการของผู้ใช้ระดับที่ 2 ของกระบวนการที่ 3 273
6.31 แสดงแผนภูมิ LDS ทางตรรกะตามความต้องการของผู้ใช้ 276
6.32 แสดงการกำหนดกลุ่มของแผนภูมิ LDS 277
6.33 แสดงการกำหนดกลุ่มฟังก์ชันของแผนภูมิการไหลของข้อมูลระดับที่ 1 302
6.34 แสดงการกำหนดกลุ่มฟังก์ชันของแผนภูมิการไหลของข้อมูลระดับที่ 2 302
6.35 แสดงการกำหนดกลุ่มฟังก์ชันของแผนภูมิการไหลของข้อมูลระดับที่ 3 303
6.36 แสดงแผนภูมิโครงสร้าง I/O ของการบันทึกข้อสอบ 313
6.37 แสดงแผนภูมิโครงสร้าง I/O ของการบันทึกเกณฑ์และรวบรวมข้อสอบ 313
6.38 แสดงแผนภูมิโครงสร้าง I/O ของการจัดทำข้อสอบ 314
6.39 แสดงแผนภูมิโครงสร้าง I/O ของการลบข้อสอบที่ไม่ต้องการ 314
6.40 แสดงความสัมพันธ์ของ LDS พร้อมกับคีย์หลัก 319
6.41 แสดงเส้นทางเมนูของแบบจำลอง 320
6.42 แสดงเส้นทางเมนูที่ 1 ของแบบจำลอง 321
6.43 แสดงเอนติตีไลฟ์ฮีสทรีไดอะแกรมของอาจารย์ 323
6.44 แสดงเอนติตีไลฟ์ฮีสทรีไดอะแกรมของรายละเอียดข้อสอบ 324
6.45 แสดงเอนติตีไลฟ์ฮีสทรีไดอะแกรมของประวัติการออกข้อสอบ 324
6.46 แสดงเอนติตีไลฟ์ฮีสทรีไดอะแกรมของข้อสอบ 325
6.47 แสดงแผนภูมิ ECD การบันทึกข้อสอบกับเอนติตีอาจารย์ 327
6.48 แสดงแผนภูมิ ECD การบันทึกเกณฑ์ให้กับข้อสอบ 327
6.49 แสดงแผนภูมิ ECD การบันทึกประวัติการใช้ข้อสอบ 327
6.50 แสดงแผนภูมิ ECD การลบข้อสอบ 328
6.51 แสดงแผนภูมิ ECD การบันทึกข้อมูลอาจารย์ 328

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

6.52 แสดงแผนภูมิ ECD การแก้ไขข้อมูลอาจารย์ 328
6.53 ก แสดงผลกระทบของเอนติตีข้อสอบ 329
6.53 ข แสดงแผนภูมิ EAP สำหรับการสร้างข้อสอบ 329
6.54 ก แสดงผลกระทบของเอนติตีประวัติข้อสอบ 330
6.54 ข แสดงแผนภูมิ EAP สำหรับการสอบถามประวัติข้อสอบ 330
6.55 ก แสดงผลกระทบของเอนติตีอาจารย์ 331
6.55 ข แสดงแผนภูมิ EAP สำหรับการสอบถามอาจารย์ (ผู้ออกข้อสอบ) 331
6.56 แสดงไดอะแกรมของวัตถุหลังการเปลี่ยนแปลงจากวิธีเชิงโครงสร้าง 332
6.57 แสดงไดอะแกรมของวัตถุพร้อมกับแสดงแอตทริบิวต์ที่ปราศจากคีย์หลัก 333
6.58 แสดงไดอะแกรมของวัตถุพร้อมกับแอตทริบิวต์และฟังก์ชันภายใน 334
6.59 แสดงใบขอให้ซื้อ (Procurement Requesition) 337
6.60 แสดงใบเปรียบเทียบการสืบราคา 338
6.61 แสดงใบเสนอราคาของบริษัท นิ เทเลคอม จำกัด แบบที่ 1 339
6.62 แสดงใบเสนอราคาของบริษัท นิ เทเลคอม จำกัด แบบที่ 2 340
6.63 แสดงใบเสนอราคาของบริษัท CD SYSTEMS จำกัด แบบที่ 1 341
6.64 แสดงใบเสนอราคาของบริษัท CD SYSTEMS จำกัด แบบที่ 2 342
6.65 แสดงใบเสนอราคาของบริษัท CD SYSTEMS จำกัด แบบที่ 2 (ต่อ) 343
6.66 แสดงใบเสนอราคาของบริษัท คอมแพ็คคอมพิวเตอร์ (ประเทศไทย) จำกัด 344
6.67 แสดงใบเสนอราคาของซอฟต์แวร์ฐานข้อมูล 345
6.68 แสดงรายละเอียดของฮาร์ดแวร์ 346
6.69 แสดงรายละเอียดของฮาร์ดแวร์ (ต่อ) 347
6.70 แสดงการปรับปรุงไดอะแกรมของวัตถุครั้งที่ 1 349
6.71 แสดงการปรับปรุงไดอะแกรมของวัตถุครั้งที่ 2 349
ก.1 แสดงหน้าจอสำหรับเข้าสู่ระบบคลังข้อสอบ 372
ก.2 แสดงหน้าจอหลักของระบบคลังข้อสอบ 373
ก.3 แสดงหน้าจอการสร้างข้อสอบรายวิชา 374
ก.4 แสดงหน้าจอการเลือกข้อสอบรายข้อ 375

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

ก.5 แสดงหน้าจอเตรียมพิมพ์ข้อสอบ 376
ก.6 แสดงหน้าจอข้อมูลข้อสอบของระบบคลังข้อสอบ 377
ก.7 แสดงหน้าจอการเพิ่มข้อมูลข้อสอบ 378
ก.8 แสดงหน้าจอการกำหนดตัวเลือกให้กับข้อสอบประเภทปรนัย 379
ก.9 แสดงหน้าจอเพิ่มตัวเลือกให้กับข้อสอบประเภทปรนัย 379
ก.10 แสดงหน้าจอการบันทึกเกณฑ์ข้อคำถาม 379
ก.11 แสดงหน้าจอฐานข้อมูลอาจารย์ในระบบคลังข้อสอบ 380
ก.12 แสดงหน้าจอการเพิ่มข้อมูลอาจารย์ 381
ก.13 แสดงหน้าจอการแก้ไขข้อมูลอาจารย์ 381
ก.14 แสดงหน้าจอฐานข้อมูลวิชาของคณะเทคโนโลยีสารสนเทศในระบบคลังข้อสอบ 382
ก.15 แสดงหน้าจอเพิ่มข้อมูลวิชา 383
ก.16 แสดงหน้าจอการแก้ไขข้อมูลวิชา 383
ก.17 แสดงหน้าจอฐานข้อมูลคณะภายในสถาบันการศึกษา 384
ก.18 แสดงหน้าจอเพิ่มข้อมูลคณะ 384
ก.19 แสดงหน้าจอการแก้ไขข้อมูลคณะ 385

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ตลอดระยะเวลา 40 กว่าปีที่ผ่านมาระบบงานธุรกิจเริ่มนำคอมพิวเตอร์เข้ามามีบทบาทกับองค์กรมากขึ้น จึงทำให้เกิดการพัฒนาระบบต่าง ๆ ซึ่งเรียกว่า "ระบบสารสนเทศ" ขึ้น และการพัฒนาระบบสารสนเทศตั้งแต่อดีตจนถึงปัจจุบันมีอยู่หลายรูปแบบ แต่รูปแบบที่เป็นที่นิยมและแพร่หลายมากที่สุดคือ การพัฒนาแบบเชิงโครงสร้าง (Structured Method) เหตุที่รูปแบบนี้เป็นที่นิยมกันมากก็เพราะว่าเป็นวิธีที่มีการพัฒนาเป็นขั้นตอนสามารถตรวจสอบและติดตามผลได้ง่าย โดยใช้แผนภูมิหรือสัญลักษณ์ที่เรียกว่าดาต้าโฟล์วไดอะแกรม (Data Flow Diagram) ซึ่งแทนกระบวนการทำงานและอี อาร์ ไดอะแกรม (Entity Relationship Diagram) แทนข้อมูลไว้สำหรับสื่อสารกับผู้ใช้ เพราะแผนภูมิทั้งสองมีความง่ายต่อการทำความเข้าใจ แต่วิธีนี้มีข้อจำกัดตรงที่ต้องใช้กับการพัฒนาระบบที่มีความชัดเจนและความต้องการที่ไม่ค่อยมีการเปลี่ยนแปลงบ่อยครั้ง ลักษณะการทำงานของวิธีนี้เป็นลักษณะบนลงล่าง (Top-Down) คือ การวิเคราะห์และออกแบบจะรวบรวมข้อมูลและความต้องการทั้งหมดของระบบอยู่ในภาพรวม หลังจากนั้นจะค่อย ๆ กระจายสู่ระดับย่อยของระบบจนกระทั่งถึงการทำงานที่ย่อยที่สุด ซึ่งการกระจายนี้เรียกว่า "Functional Decomposition" แต่ในโลกของความเป็นจริงข้อมูลกับความต้องการของผู้ใช้นั้นมีความสลับซับซ้อนและสามารถเกิดความเปลี่ยนแปลงขึ้นได้ตลอดเวลาจึงทำให้การพัฒนาระบบสารสนเทศด้วยวิธีนี้มีข้อจำกัดอยู่หลายประการ คือ บำรุงรักษายาก การปรับปรุงแก้ไขแต่ละครั้งทำให้กระทบกระเทือนต่อระบบหลัก ไม่สามารถนำโพรเซสของระบบเดิมกลับมาใช้ใหม่ได้โดยสะดวก ไม่สามารถพัฒนาระบบแบบขนาน (Parallel) การเวียนบังเกิด (Recursive) และทำซ้ำ (Iterative) ได้ เป็นต้น[31]

ต่อมาปลายปี ค.ศ. 1980 มีนักคิดหลายท่านได้พยายามคิดแก้ไขข้อบกพร่องและข้อจำกัดที่ปรากฏในวิธีการพัฒนาระบบสารสนเทศแบบเดิม โดยเสนอวิธีที่ชื่อว่า การพัฒนาระบบเชิงวัตถุ (Object-Oriented System Development) การพัฒนาระบบแบบใหม่นี้ ผู้วิเคราะห์และออกแบบจะมองระบบในลักษณะความเป็นจริงมากขึ้น โดยจะมองทุก ๆ สิ่งที่ต้องการให้อยู่ในรูปของวัตถุ วิธีเชิงวัตถุประกอบด้วย วัตถุ (Object) การห่อหุ้ม (Encapsulation) เมสเซจ (Message) การสืบทอด (Inheritance) และโพลีมอร์ฟิซึม (Polymorphism)และในแต่ละออบเจ็กต์ประกอบด้วย ชื่อ (Name) แอตทริบิวต์ (Attributes) และ ฟังก์ชันหรือเมธอด (Function or Method) ที่วัตถุนั้นเป็นผู้กระทำ การพัฒนาระบบสารสนเทศตามวิธีการเชิงวัตถุนี้ ทำให้ระบบที่ได้มีลักษณะและคุณสมบัติที่แตกต่างไปจากเดิม เช่น แก้ปัญหาได้ถูกต้องมากขึ้น มีความยืดหยุ่น สามารถขยายหรือเพิ่มเติมได้ สามารถนำกลับมาใช้ใหม่และบำรุงรักษาง่าย เป็นต้น[22] ถึงแม้ว่าระบบที่ได้จะมีคุณสมบัติและลักษณะตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามที่ผู้ใช้และผู้พัฒนาต้องการก็ตาม แต่ขั้นตอนการพัฒนาระบบเชิงวัตถุก็ยังไม่ชัดเจน กล่าวคือ การพัฒนาระบบสารสนเทศเชิงวัตถุนี้ มีแบบจำลองจำนวนมากที่พยายามนำเสนอวงจรชีวิตและขั้นตอนของการพัฒนาเชิงวัตถุ เช่น แบบจำลอง Build-and-Fix[50] แบบจำลองน้ำตก (Waterfall Model)[45] แบบจำลอง Rapid Prototyping[50] แบบจำลอง Incremental[45] แบบจำลอง Spiral[6] และแบบจำลองน้ำพุ (Fountain Model)[2],[15],[16],[17],[18] เป็นต้น นอกจากแบบจำลองที่นำเสนอวิธีการพัฒนาแล้วยังมีวิธีเชิงวัตถุที่ผู้พัฒนาต้องพิจารณาเพื่อนำไปใช้พัฒนา เช่น OMT[28], OODA[13], OOSE[25], OOAD[27], The Wirfs-Brock method[54], The Coad/Yourdon method [39],[40], SOMA[23] และ HOOD[47] เป็นต้น แต่ละแบบจำลองที่ได้กล่าวมานี้ล้วนแต่เป็นสิ่งที่นำเสนอวงจรชีวิตของการพัฒนาระบบเชิงวัตถุทั้งสิ้น ซึ่งแต่ละแบบจำลองไม่ได้กล่าวถึงลำดับและขั้นตอนของกิจกรรมการพัฒนาที่ชัดเจน

สำหรับงานวิจัยนี้เลือกแบบจำลองน้ำพุและโมเดลของ James Rumbaugh โดยถ้าสังเกตจากตารางที่ 1.1, 1.2 และ 1.3 จะพบว่าแต่ละโมเดลและแบบจำลองมีเทคนิคและแนวคิดแตกต่างกัน ตารางที่ 1.1 เปรียบเทียบแบบจำลองประเภทต่าง ๆ เช่น แบบจำลองน้ำพุ แบบจำลองน้ำตก และแบบจำลองสไพเริล เป็นต้น ซึ่งแสดงถึงข้อดีและข้อเสียของแต่ละแบบจำลอง จากตารางนี้ทำให้ทราบว่าแบบจำลองน้ำพุ สนับสนุนการพัฒนาแบบทำซ้ำและการเวียนบังเกิดพร้อมกับสามารถพัฒนาแบบขนานกันระหว่างขั้นตอนได้ ตารางที่ 1.2 เปรียบเทียบวิธีเชิงวัตถุที่นิยมใช้อยู่ในปัจจุบันซึ่งแสดงให้เห็นถึงแนวคิดของแต่ละวิธีโดยเริ่มจากแนวคิดหลัก เช่น แนวคิดคลาส ออบเจ็กต์ แอตทริบิวต์และเมธอด เป็นต้น ความสัมพันธ์ของออบเจ็กต์และอื่น ๆ และตารางที่ 1.3 แสดงเทคนิคที่แต่ละวิธีเลือกใช้ ที่กล่าวถึงไดอะแกรมที่เลือกใช้สื่อสารกับผู้ใช้ระบบ ส่วนวิธี OMT มีวิธีการวิเคราะห์และออกแบบที่แบ่งออกเป็นโมเดล ซึ่งแต่ละโมเดลมีหน้าที่และแสดงความรับผิดชอบที่แตกต่างกันออกไปทำให้โมเดลที่จากการพัฒนาที่แบ่งเป็นสัดส่วนและสามารถนำไปประยุกต์กับขั้นตอนการอิมพลีเมนต์ได้ค่อนข้างง่าย

จากที่กล่าวมานี้ สรุปได้ว่าการพัฒนาระบบสารสนเทศมีวิวัฒนาการขึ้นตามลำดับ และวิธีการก็มีอยู่หลายวิธีที่ให้เลือกใช้มาพัฒนาระบบสารสนเทศตามต้องการ แต่ในปัจจุบันวิธีการวิเคราะห์และพัฒนาระบบสารสนเทศมีแนวโน้มไปสู่การวิเคราะห์ออกแบบเชิงวัตถุมากขึ้น ซึ่งวิธีต่าง ๆ ที่กล่าวมาข้างต้นไม่มีวิธีใดที่แสดงให้เห็นถึงขั้นตอนการทำงานที่ชัดเจนโดยเฉพาะพัฒนาระบบสารสนเทศเชิงวัตถุ เพราะถ้าผู้พัฒนาเลือกวิธีใดวิธีหนึ่งก็ไม่สามารถที่จะใช้อีกวิธีหนึ่งรวมกันได้ ดังนั้นงานวิจัยชิ้นนี้มุ่งเน้นไปที่ขั้นตอนการวิเคราะห์และออกแบบในแต่ละขั้นตอนของวงจรการพัฒนาระบบสารสนเทศโดยละเอียด พร้อมกับสามารถนำวิธีการออกแบบเดิมให้สามารถนำมาปรับปรุงให้เข้ากับวิธีเชิงวัตถุได้ ซึ่งในการปรับปรุงนี้จะต้องมีการเพิ่มเติมรายละเอียดบางอย่างลงไปเพื่อให้ได้ซึ่งวิธีการพัฒนาระบบเชิงวัตถุที่สมบูรณ์ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.1 แสดงการเปรียบเทียบวงจรชีวิตของโมเดล[50]

โมเดลวงจรชีวิต	ข้อดี	ข้อเสีย
Build-and-fix model	• เหมาะสำหรับโปรแกรมขนาดเล็กที่มีการบำรุงรักษาน้อย	• โปรแกรมมีลักษณะพื้น ๆ ไม่เพียงพอที่จะนำไปใช้งานต่อไปได้
แบบจำลองน้ำตก (Waterfall Model)	• เป็นวิธีที่มีระเบียบและขั้นตอน • มีเอกสารประกอบการพัฒนา	• ผลงานที่ออกมาไม่ตรงกับความต้องการของผู้ใช้เท่าที่ควร
Rapid Prototyping Model	• ระบบที่ได้ตรงตามความต้องการของผู้ใช้	• ข้อจำกัดของเครื่องมือที่นำมาใช้งาน
Incremental Model	• ให้ผลตอบแทนของการลงทุนสูง • บำรุงรักษาง่าย	• เป็นสถาปัตยกรรมแบบเปิด • ประยุกต์ไปสู่ Build-and-Fix Model
สไพเริลโมเดล (Spiral Model)	• คุณสมบัติทั้งหมดของโมเดลถูกรวบรวมไว้ในที่เดียวกัน	• ใช้ได้กับการพัฒนาระบบขนาดใหญ่ และเหมาะสำหรับพัฒนาเอง • ผู้พัฒนามีความเสี่ยงสำหรับการวิเคราะห์และการแก้ปัญหาที่เกิดขึ้น
แบบจำลองน้ำพุ (Fountain Model)	• สนับสนุนการทำซ้ำในแต่ละขั้นตอน • สามารถพัฒนาแบบขนานกันได้ระหว่างขั้นตอนการพัฒนา	• สามารถปรับเปลี่ยนไปสู่ CABTAB (code a bit, test a bit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.2 แสดงการเปรียบเทียบแนวคิดของวิธีเชิงวัตถุ

Concepts of Supermethodology	OOA/OOD	DOOS	OMT	OODA	OOSA	OOAD
MAIN CONCEPTS						
Class & Objects	=					
Class	=					
Abstract Class		=	=	=	Object	Object Type
Meta Class			=	=		
Object	=	=	=	=		
Passive Object				=		
Active Object				=		
Attribute	=		=	Field	=	
Derived Attribute			=			
Attribute Contraint	=		=	=	=	
Method	Service	Responsibility	Operation	Operation		Operation
Method Signature	Parameter	=	=	Operation Parameter		
Subject	=	Subsystem	Module	Class Category		
…						
RELATIONSHIPS						
Inheritance	Gen-Spec	Super/subclass	Super/subclass	Super/subclass	Super/subtype(1)	Super/subtype
Multiple Inheritance	=	=	=	=		=
Whole-Part relationship	=	(2)	=	=		Composition
Association	Instance connection		=	Using relationship	Relationship	Relation
Derived association			=			Computed function
Message connection	=	Collaboration		Message relationship		
Instantiation relationship				=		
…	…	…	…	…	…	…
OPERATIONS	…	…	…	…	…	…
COMMUNICATION	…	…	…	…	…	…
CONCURRENCY	…	…	…	…	…	…

(1) OOSA only supports inheritance of Attributes.

(2) DOOS does not really have a whole-part concept

ที่มา S. Hong, G. Goor, S. Brinkkemper, "A Formal Approach to the Comparison of Object-Oriented Analysis and Design Methodologies", IEEE 1993, p. 689-698

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.3 แสดงการเปรียบเทียบเทคนิคของวิธีการเชิงวัตถุ

Technique to capture	OOA/OOD	DOOS	OMT	OODA	OOSA	OOAD
1. Object	Object layer OOA model			Object Diagram		
2. Classes/Class Structures	Class layer OOA model	CRC cards; Venn diagram; Hierarchy Graphs	Object Diagram	Class Diagram	Information model	Object schema
3. Partitioning of the Class and Class Structure	Subject layer OOA model	Subsystem Cards	Modules	Class Category Diagrams		Object Flow Diagram
4. Object Dynamics	Object state Tables	(1)	State Diagrams	State Diagrams	State Transition diagram(2)	Event Schema
5. System Dynamics				Timing Diagrams		
6. Functional behavior	Service charts		Data Flow diagrams		Data Flow diagrams(2)	
7. Implementation			Subsystems	Module/Process Diagrams		
8. Communication between classes/objects	Message connection OOA model	Collaborations graphs	Event Flow Diagrams	Synchronization on Object Diagram		

(1) The dynamic behavior of object is not explicitly specified.

(2) This technique is provided but it is described very briefly.

ที่มา S. Hong, G. Goor, S. Brinkkemper, "A Formal Approach to the Comparison of Object-Oriented Analysis and Design Methodologies", IEEE 1993, p. 689-698

1.2 วัตถุประสงค์

วัตถุประสงค์ของงานวิจัยนี้เพื่อนำเสนอขั้นตอนการวิเคราะห์และออกแบบของการพัฒนาระบบสารสนเทศเชิงวัตถุ โดยนำวิธีเชิงโครงสร้างมาผสมผสานกับวิธีเชิงวัตถุภายใต้แบบจำลองน้ำพุ เพื่อให้เกิดประโยชน์ในด้านการพัฒนา คือ ทำให้วิธีการเชิงวัตถุมีขั้นตอนของกิจกรรมที่ชัดเจนและสามารถตรวจสอบขั้นตอนการพัฒนาได้ง่าย

1.3 ขอบเขตงานวิจัย

จากวัตถุประสงค์หลักของงานวิจัยนี้ คือ การนำเสนอขั้นตอนการพัฒนาระบบสารสนเทศให้สามารถตรวจสอบขั้นตอนการพัฒนาได้ง่ายจึงทำให้การประยุกต์โมเดลมีขอบเขตงานวิจัยดังนี้

1. ครอบคลุมรูปแบบการพัฒนาระบบสารสนเทศทั่วไป ปกติวงจรชีวิตของการพัฒนาระบบสารสนเทศประกอบด้วย 3 ขั้นตอนหลัก คือ วิเคราะห์ ออกแบบและลงมือปฏิบัติ (ดังแสดงในรูปที่ 1.1 (ก)) ซึ่งแต่ละขั้นตอนหลักสามารถแบ่งออกเป็นขั้นตอนย่อย ๆ ได้ (ดังแสดงในรูปที่ 1.1 (ข))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

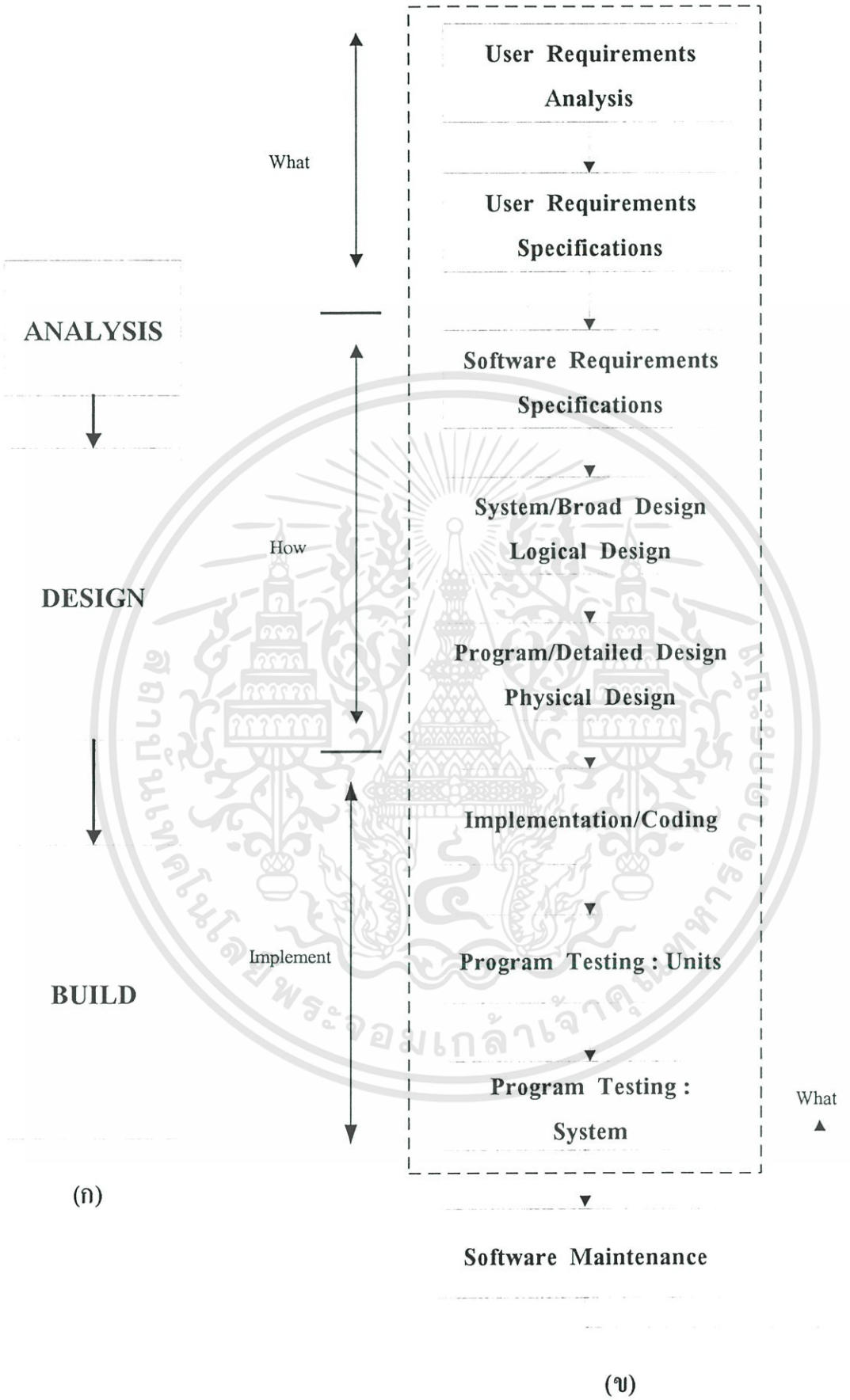
2. วิธีเชิงโครงสร้างจะใช้วิธี Structured System Analysis and Design Method (SSADM) รุ่นที่ 4.2 เพราะเป็นวิธีเชิงโครงสร้างที่มีขั้นตอน กิจกรรมและงานที่สามารถติดตามได้ง่าย (จะกล่าวถึงในบทที่ 2)

3. วิธีการเชิงวัตถุใช้วิธีการจำลองแบบ OMT เป็น วิธีการพัฒนาระบบสารสนเทศแนวใหม่ที่ถูกแบ่งออกเป็น 3 ส่วน คือ แบบจำลองเชิงวัตถุ (Object Model) แบบจำลองไดนามิก (Dynamic Model) และแบบจำลองฟังก์ชันนัล (Functional Model) (จะกล่าวถึงในบทที่ 3)

4. แบบจำลองใช้แบบจำลองน้ำพุ (Fountain Model) เป็น แบบจำลองที่สนับสนุนทั้งวิธีการพัฒนาแบบน้ำตกและแบบเชิงวัตถุ (จะกล่าวถึงในบทที่ 4)

โมเดลประยุกต์ของงานวิจัยนี้ ประกอบด้วย 5 โมดูลคือ Requirements and Feasibility Study Module, Analysis Module, Design Module, Implementation and Testing Module และ System Support Module ซึ่งทั้ง 5 โมดูลประกอบด้วยขั้นตอน 0-10 และในแต่ละขั้นตอนประกอบด้วยกิจกรรมและงานย่อย ๆ (ซึ่งจะกล่าวรายละเอียดต่อไปในบทที่ 5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 (ก) แสดงส่วนหลักของวงจรชีวิตซอฟต์แวร์
(ข) แสดงรายละเอียดของวงจรชีวิตซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการดำเนินงานวิจัย

1) ศึกษาผลงานวิจัยที่เกี่ยวข้อง
2) ศึกษาทฤษฎีและหลักการที่เกี่ยวข้อง
3) วิเคราะห์และออกแบบโมเดล ด้วยการผสมผสานวิธีเอส เอส เอ ดี เอ็ม กับแบบจำลองน้ำพุ
4) ทดลองโมเดลด้วยกรณีศึกษากับโมเดลประยุกต์
5) ประเมินผลการวิจัย
6) จัดทำเอกสาร

1.5 สรุปงานวิจัยที่เกี่ยวข้อง

1. งานวิจัยเรื่อง "การพัฒนาระบบซอฟต์แวร์ด้วยวิธีเชิงวัตถุ" (Object-Oriented approach to software development, a walk through a number of topics) จัดทำโดย J. Lewi, E. Steegmans และ J. De Man [32] กล่าวถึงการพัฒนาระบบซอฟต์แวร์ที่มีความซับซ้อนและมีขนาดใหญ่ ซึ่งวิธีการพัฒนาแบบเก่านั้นมีข้อจำกัด พร้อมกับเปรียบเทียบข้อดีและข้อเสียวิธีการพัฒนาซอฟต์แวร์แบบเชิงโครงสร้างและเชิงวัตถุ โดยเริ่มตั้งแต่โมเดล วิธีการ การแก้ไขปัญหาและการนำเสนอ เป็นต้น เพื่อสรุปให้เห็นถึงประโยชน์ของการพัฒนาซอฟต์แวร์เชิงวัตถุที่กำลังเป็นที่นิยมในปัจจุบัน
2. งานวิจัยเรื่อง "วิธีเชิงวัตถุกับการวิเคราะห์และออกแบบ" (Object-Oriented and Conventional analysis and Design Methodologies) โดย Robert G. Fichman และ Chris F. Kemerer [44] วิธีเชิงวัตถุที่ในปัจจุบันมีอยู่ด้วยกันหลายวิธี ในบทความนี้กล่าวถึงการเปรียบเทียบโมเดลของวิธีเชิงวัตถุในแบบต่าง ๆ ที่มีอยู่ โดยเปรียบเทียบตั้งแต่ขั้นตอนการวิเคราะห์ออกแบบและสิ่งที่โมเดลเหล่านั้นสนับสนุนในการนำไปใช้งาน เช่น วิธี OMT ของ James Rumbaugh วิธี OOA/OOD ของ Peter Coad กับ Edward Yourdon วิธี OOD ของ Gredy Booch และวิธีของ Wirfs-Brock และคณะ เป็นต้น
3. งานวิจัยเรื่อง "วงจรชีวิตของวิธีการเชิงวัตถุ" (The Object-Oriented System Life Cycle) โดย B. Henderson-Sellers และ J.M. Edwards [16] กล่าวถึงวงจรชีวิตของการพัฒนาระบบสารสนเทศ ส่วนใหญ่การพัฒนาเริ่มจากการวิเคราะห์ ออกแบบและอิมพลีเมนต์ สำหรับการพัฒนาระบบวิธีเชิงวัตถุก็มีวิธีการเช่นเดียวกัน แต่ในบทความนี้กล่าวถึงการพัฒนาที่ตอนเริ่มการวิเคราะห์ระบบอาจไม่จำเป็นต้องเป็นแบบเชิงวัตถุก็ได้ กล่าวคือ สามารถเริ่มต้นพัฒนาได้ทั้งแบบเชิงโครงสร้างหรือวัตถุได้แต่สุดท้ายของการกระบวนการต้องสิ้นสุดที่การออกแบบเชิงวัตถุ แต่ข้อเสนอแนะของผู้วิจัยกล่าวว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสารสนเทศที่ดีควรมีการพัฒนาที่เป็นวิธีการเดียวกันทั้งระบบ แต่ถ้ามีการเริ่มต้นที่แตกต่างกันก็ไม่ได้หมายความว่าระบบนั้นจะไม่ดีพอ ซึ่งถ้าเริ่มต้นแบบเชิงโครงสร้างนั้น ผู้ออกแบบต้องนำวิธีการต่าง ๆ เหล่านั้นมาเปลี่ยนแปลงไปสู่วิธีเชิงวัตถุอีกครั้งหนึ่งเพื่อให้ได้ระบบสารสนเทศที่เป็นวิธีการเชิงวัตถุทั้งระบบ

4. งานวิจัยเรื่อง "การพัฒนาระบบสารสนเทศด้วยสไพเรลโมเดล" (A Spiral Model of Software Development and Enhancement) โดย Barry W. Boehm [6] กล่าวถึง วิธีการพัฒนาระบบสารสนเทศที่มีวงจรชีวิตที่สามารถย้อนกลับไปสู่ขั้นตอนแรกได้ใหม่ กล่าวคือ เป็นโมเดลที่นำเสนอเพื่อแก้ไขข้อจำกัดของแบบจำลองน้ำตกแบบเดิม ที่เมื่อพัฒนาจนครบทั้ง 5 ขั้นตอนแล้วไม่สามารถย้อนกลับไปทำได้อีก ส่วนวิธีการนี้จะมีการพัฒนาแบบล่างขึ้นบนในลักษณะของก้นหอย ซึ่งเริ่มจากการพัฒนาในโพรเซสที่เล็ก ๆ ก่อนจากนั้นค่อยพัฒนาไปสู่โพรเซสที่มีขนาดใหญ่ขึ้น ระบบสารสนเทศที่เหมาะสมกับแบบจำลองนี้มักจะเป็นระบบที่ความต้องการของผู้ใช้นั้นเปลี่ยนแปลงบ่อย
5. งานวิจัยเรื่อง "เทคโนโลยีเชิงวัตถุ : ก้าวไปในทิศทางใดในอนาคต?" (Object technology: w(h)ither the future?) โดย B. Henderson-Sellers [15] กล่าวถึงวิธีการพัฒนาระบบสารสนเทศเชิงวัตถุในรูปแบบใหม่ภายใต้ MOSES ซึ่งภายในประกอบด้วย 3 ส่วนหลักคือ Product Lifecycle, Process Lifecycle และ Activities ทั้ง 3 ส่วนนำเสนอในรูปของแบบจำลองน้ำพุ ซึ่งมีลักษณะการทำซ้ำและเวียนบังเกิดในทุก ๆ ขั้นตอนของการพัฒนา พร้อมกับแบ่งการวิเคราะห์และออกแบบออกเป็น 2 ระดับ คือ ระดับระบบ (System Level) และระดับระบบย่อย (Sub-System Level) ซึ่งทั้ง 2 ระดับมีความเกี่ยวข้องกัน ระดับระบบ คือ การวิเคราะห์และออกแบบระดับบนของระบบสารสนเทศโดยกล่าวถึงขอบเขตโดยรวมของระบบทั้งหมด ส่วนระดับระบบย่อยจะเพิ่มเติมรายละเอียดของการวิเคราะห์และออกแบบให้มากขึ้น ซึ่งในระดับนี้ผู้พัฒนาสามารถพัฒนาแบบขนานกันได้ นอกจากนี้แบบจำลองน้ำพุยังสนับสนุนวิธีการพัฒนาแบบเชิงวัตถุและสามารถนำไปใช้กับการพัฒนาแบบเชิงโครงสร้างได้อีกด้วย
6. งานวิจัยเรื่อง "การรวมฟังก์ชันกับวิธีเชิงวัตถุสำหรับการออกแบบซอฟต์แวร์" (A Combined Functional and Object-Oriented Approach to Software Design) โดย Haifeng Qian, Eduardo B. Fernandez และ Jie Wu [14] กล่าวถึงซอฟต์แวร์ในปัจจุบันมีขนาดใหญ่และซับซ้อน ซึ่งมักประกอบด้วยออบเจ็กต์และฟังก์ชันการทำงานของระบบ การออกแบบระบบที่มีลักษณะเชิงวัตถุหรือเชิงโครงสร้างจะอยู่แยกออกจากกันโดยสิ้นเชิง ในบทความนี้นำเสนอวิธีการนำฟังก์ชันของวิธีเชิงโครงสร้างมารวมกับวิธีเชิงวัตถุที่เรียกว่า "CFOOD – Combined Functional and Object-Oriented Design

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Approach" ซึ่งนำเสนอวิธีการรวมทั้งสองวิธีเข้าด้วยกัน โดยนำไปประยุกต์ใช้กับระบบตรวจคนไข้ของโรงพยาบาลเป็นกรณีศึกษา

7. งานวิจัยเรื่อง "การประยุกต์วิธีเชิงวัตถุสำหรับการวิเคราะห์และออกแบบ" (Applying Object-Oriented Analysis and Design) โดย Jean-Marc Nerson [29] กล่าวถึงการนำวิธีเชิงวัตถุมาประยุกต์ใช้กับการพัฒนาระบบสารสนเทศ โดยนำเสนอเทคนิคของการพัฒนาในระดับโพรเซสของวิธีเชิงวัตถุเพราะเชื่อว่าทำให้การค้นหาเอกภพของปัญหาและแนวทางการแก้ไขมีความสอดคล้องกัน ซึ่งนำเสนอในรูปแบบของ "BON (Better Object Noation)"
8. งานวิจัยเรื่อง "การรวมตัวและความสอดคล้องของการวิเคราะห์เชิงวัตถุ" (Unification and Consistency Verification of Object-Oriented Analysis Models) โดย Toshiaki Aoki และ Takuya Katayama [52] กล่าวถึงขนาดของซอฟต์แวร์ที่เกิดจากการพัฒนาจากวิธีเชิงวัตถุนั้นมีขนาดใหญ่ขึ้นเรื่อย ๆ และวิธีการที่ใช้มักจะมีมุมมองที่แตกต่างกันออกไปเช่น วิธีของ OMT ซึ่งมีการแยกออกเป็น วิเคราะห์และออกแบบ โดยเฉพาะส่วนของการวิเคราะห์ที่มีโมเดลประกอบกันหลาย ๆ โมเดล คือ ออบเจ็กต์โมเดล ไดนามิกโมเดลและฟังก์ชันนัลโมเดล ซึ่งทั้งหมดจะถูกรวมในขั้นตอนนี้ ทำให้ยากสำหรับการตรวจสอบความถูกต้องโดยเฉพาะดาต้าโฟลว์ในฟังก์ชันนัลโมเดลที่อธิบายถึงฟังก์ชันการทำงานของออบเจ็กต์นั้นกระทำได้ยาก จึงนำเสนอโมเดลสำหรับการตรวจสอบขึ้น เรียกว่า "Unified Model" ซึ่งประกอบด้วย Basic Model และ Unification Mapping ก่อนที่จะนำไปสู่ Unified Model ต้องผ่านขั้นตอนของ Basic Model ก่อน ที่ประกอบด้วย Basic Object Model, Basic Dynamic Model และ Basic Functional Model ทั้งหมดเป็นอิสระต่อกัน จากนั้นขั้นตอนเหล่านั้นจะถูกส่งมาให้กับ Unification Mapping ทำงานอีกครั้งเพื่อนำไปสู่โมเดลดังกล่าว และสุดท้ายเราสามารถที่จะอธิบายถึงการทำงานของดาต้าโฟลว์ที่เกิดขึ้นกับออบเจ็กต์ได้ชัดเจนมากยิ่งขึ้น
9. งานวิจัยเรื่อง "การใช้ฟอร์มอลโมเดลสำหรับการออกแบบระบบเชิงวัตถุที่น่าเชื่อถือ" (Formal Model Approach for Reliable Object-Oriented Information System Design) โดย Toshiaki Aoki และ Takuya Katayama [51] กล่าวถึงการพัฒนาระบบสารสนเทศเชิงวัตถุในปัจจุบันระบบมีขนาดใหญ่และซับซ้อนมาก ซึ่งวิธีการที่มีอยู่ก็ไม่ได้สนับสนุนหรือรองรับสิ่งต่าง ๆ เหล่านี้ทำให้นักพัฒนาระบบมีความไม่มั่นใจกับระบบที่พัฒนา ดังนั้นจึงได้คิดวิธีการพัฒนาขึ้นใหม่ เรียกกว่า "Unified Model" ขึ้นมา ซึ่งคือการรวมกันระหว่างฟอร์มอลโมเดลกับการพัฒนาเชิงวัตถุ เพื่อเป็นตัวกลางสำหรับนักพัฒนาระบบที่ต้องการพัฒนาระบบสารสนเทศเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. งานวิจัยเรื่อง "การจัดการพัฒนาซอฟต์แวร์โดยใช้เครื่องมือ RMT" (Managing the Software Development by Using the Recursive Multi-Treaded(RMT) Tool) โดย Arturo I. Concepcion, Sunny Lin และ Scott J. Simon [5] กล่าวถึง การพัฒนาซอฟต์แวร์ที่มีลักษณะการเวียนบังเกิด ซึ่งในปัจจุบันการพัฒนาซอฟต์แวร์มีแนวโน้มการพัฒนาแบบขนานมากขึ้น โดยโมเดลที่นำเสนอในปัจจุบันนั้นมีข้อจำกัด เช่น การยืดหยุ่นของกิจกรรมระหว่างวงจรชีวิตเพราะทำให้ยากต่อการกำหนดสเตจปัจจุบัน กับขนาดของวงจรชีวิตที่มีขนาดเล็กเกินไปทำให้ยากต่อการประเมินคุณภาพและติดตามผล เป็นต้น ดังนั้นจึงพัฒนาโมเดลขึ้นมาใหม่เรียกว่า "RMT" เพื่อนำมาแก้ไขข้อจำกัดดังกล่าว หลักการคือ แบ่งการพัฒนาออกเป็นส่วนย่อยหรือ "Thread" ภายในมีขั้นตอนและกระบวนพัฒนาเหมือนกับวงจรชีวิตของการพัฒนาระบบทั่วไป คือ วิเคราะห์ ออกแบบ อิมพลีเมนต์และทดสอบ และสามารถพัฒนาแบบขนานและมีความแตกต่างกันได้
11. งานวิจัยเรื่อง "การเปรียบเทียบวิธีการวิเคราะห์และออกแบบเชิงวัตถุ" (A Formal Approach to the Comparison of Object-Oriented Analysis and Design Methodologies) โดย Shuguang Hong, Geert van den Goor และ Sjaak Brinkkemper [47] กล่าวถึงการเปรียบเทียบวิธีเชิงวัตถุที่มีอยู่ในปัจจุบันทั้งแนวคิดและเทคนิคที่ใช้ ซึ่งวิธีที่นำมาเปรียบเทียบ ได้แก่ OMT, OODA, OOAD, OOA/OOD และ OOSA การเปรียบเทียบจะทำกับระดับ Meta-Process Model และ Meta-Data Model

จากงานวิจัยข้างต้นแต่ละบทความมุ่งเน้นเรื่องการนำวิธีการเชิงวัตถุไปประยุกต์ใช้กับการพัฒนาระบบสารสนเทศและการวิเคราะห์เปรียบเทียบโมเดลเชิงวัตถุของแต่ละวิธีมากกว่าการจัดการหรือจัดทำขั้นตอนการพัฒนาระบบสารสนเทศให้มีความชัดเจน ถ้าพิจารณาแต่ละบทความโดยเฉพาะบทความที่ 3, 5 และ 6 นั้น พยายามนำวิธีเชิงโครงสร้างและวิธีเชิงวัตถุมาผสมผสานกันเพื่อให้ได้ระบบสารสนเทศที่เหมาะสมจึงนับว่าเป็นแนวคิดเริ่มต้นของการประยุกต์ของทั้ง 2 วิธีเข้าด้วยกันแต่การประยุกต์นั้นไม่ได้มุ่งเน้นไปทางด้านการจัดการขั้นตอนการพัฒนาระบบสารสนเทศ

1.6 ข้อจำกัดของงานวิจัย

เนื่องจากงานวิจัยนี้เน้นไปทางด้านขั้นตอนการจัดการและวิธีการพัฒนาระบบสารสนเทศ ดังนั้นจึงมีข้อจำกัดทางด้านการอิมพลีเมนต์ของระบบ เพราะเครื่องมือที่ใช้สำหรับการทดลองในปัจจุบันยังมีการแบ่งแยกหน้าที่การทำงานกันอยู่ กล่าวคือ เครื่องมือสำหรับพัฒนาโปรแกรมจะแสดงวิธีการทำงานของฟังก์ชัน ส่วนเครื่องมือสำหรับการจัดการฐานข้อมูลจะมีหน้าที่จัดเก็บและจัดการฐานข้อมูลเพียงอย่างเดียว ถึงแม้ว่าการวิเคราะห์และออกแบบจะเป็นแบบเชิงวัตถุทั้งหมดก็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาม ดังนั้นข้อจำกัดเหล่านี้ยังคงเป็นสิ่งที่ผู้พัฒนาระบบต้องตัดสินใจในการเลือกใช้เครื่องมือสำหรับการพัฒนาระบบสารสนเทศต่อไปในอนาคต

1.7 ส่วนประกอบวิทยานิพนธ์

วิทยานิพนธ์นี้ถูกแบ่งออกเป็น 4 ส่วนประกอบด้วย 7 บท มีรายละเอียดดังนี้

บทที่ 1 บทนำ กล่าวถึงที่มาของปัญหา ทฤษฎีที่เกี่ยวข้อง งานวิจัยที่เกี่ยวข้อง ขอบเขตงานวิจัย และขั้นตอนการวิจัย

บทที่ 2 กล่าวทฤษฎีของวิธีการออกแบบสารสนเทศเชิงโครงสร้างที่ใช้วิธีการของเอส เอส เอ ดี เอ็ม

บทที่ 3 กล่าวถึงทฤษฎีของวิธีเชิงวัตถุซึ่งนำเสนอวิธีของ James Rumbaugh เป็นพื้นฐานของงานวิจัยชิ้นนี้

บทที่ 4 กล่าวถึงทฤษฎีของแบบจำลองน้ำพุโดยกล่าวถึงหลักการ วิธีการพัฒนา และกิจกรรมที่เกิดขึ้นในแบบจำลองน้ำพุ

บทที่ 5 กล่าวถึงโมเดลประยุกต์ซึ่งเป็นการผสมผสานระหว่างวิธีเชิงโครงสร้าง วิธีเชิงวัตถุภายใต้แบบจำลองน้ำพุ

บทที่ 6 กรณีศึกษาของการนำโมเดลประยุกต์มาใช้งานโดยนำเสนอระบบคลังข้อสอบเป็นต้นแบบในงานวิจัยนี้

บทที่ 7 สรุปผลและเสนอแนะงานวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

วิธีเชิงโครงสร้าง

(Structured Model)

เอส เอส เอ ดี เอ็ม (SSADM) [42],[43] เป็นวิธีการหนึ่งที่ใช้สำหรับการวิเคราะห์และออกแบบระบบสารสนเทศ ซึ่งในช่วงปี 1980 รัฐบาลอังกฤษได้มอบหมายให้หน่วยงานของรัฐที่ชื่อว่า CCTA (Central Computing and Telecommunications Agency) พัฒนาขึ้นเพื่อเป็นแบบแผนสำหรับการพัฒนาระบบสารสนเทศให้กับองค์กรต่าง ๆ ของรัฐและเอกชน ภายในประเทศ เพื่อให้เป็นไปในทิศทางเดียวกัน ปัจจุบันมีการพัฒนาอย่างต่อเนื่องจากรุ่นที่แรกจนถึงรุ่นล่าสุดมีทั้งหมด 5 รุ่น ได้แก่ รุ่นที่ 1 2 3 4 และ 4.2

วัตถุประสงค์หลักของวิธีนี้คือช่วยให้ทีมพัฒนาสามารถวางแผนการวิเคราะห์และออกแบบระบบได้อย่างถูกต้องและตรงตามความต้องการขององค์กรมากที่สุด จึงนับได้ว่า SSADM นั้นเป็นวงจรชีวิตของการพัฒนาระบบที่มีความละเอียดและครบถ้วน เพราะแต่ละระยะประกอบด้วยขั้นตอน กิจกรรมและเอกสารต่าง ๆ มากมาย แต่ในทางปฏิบัติยังขาดอีกหลายขั้นตอนเช่น การวางแผน การทดสอบ และ การจัดการโครงงาน เป็นต้น

โดยพื้นฐานของการพัฒนาด้วยวิธีนี้ต้องการให้ระบบเป็นไปตามความต้องของผู้ใช้จึงทำให้การพัฒนาของแต่ละกระบวนการมีความจำเป็นมาก เพราะในแต่ละขั้นตอนที่เกิดขึ้นจะอ้างอิงถึงกันเพื่อตรวจสอบความถูกต้องและความสมบูรณ์ของระบบจากผู้ใช้ทุกครั้ง กล่าวได้ว่าผู้พัฒนากับผู้ใช้ระบบจะต้องมีการติดต่อสื่อสารกันตลอดเวลาเพื่อให้แต่ละขั้นตอนสมบูรณ์ที่สุด เพราะผลลัพธ์ของแต่ละขั้นตอนจะนำไปเป็นข้อมูลเข้าของขั้นตอนถัดไป ดังนั้นถ้าขั้นตอนแรกไม่สมบูรณ์หรือไม่ถูกต้องก็ไม่สามารถที่จะข้ามไปสู่ขั้นตอนถัดไปได้

ในบทนี้จะกล่าวถึงรายละเอียดของ SSADM ทั้งหลักการ โครงสร้าง และขั้นตอนทั้ง 7 ใน 5 โมดูลที่ประกอบด้วย Feasibility Stage, Investigation of Current Environment Stage, Business System Option Stage, Definition of Requirements Stage, Technical System Option Stage, Logical Design Stage และ Physical Design Stage

2.1 วิธีการเชิงโครงสร้าง (Structured Method)

โดยทั่วไปการพัฒนาระบบสารสนเทศมีขั้นตอนที่สำคัญดังนี้

- การศึกษาความเหมาะสม (Feasibility)
- การวิเคราะห์ (Analysis)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การออกแบบ (Design)
- การเขียนและทดสอบโปรแกรม (Implementation)
- การบำรุงรักษา (Maintenance)

แต่วิธี SSADM (Version 4.0) จะครอบคลุมเพียง Feasibility, Analysis, และ Design ซึ่งใน 3 ส่วนของวิธีนี้จะแบ่งออกเป็น Stage ซึ่งมีทั้งหมด 6 Stage แต่ละ Stage มี Step การทำงานที่เกี่ยวข้องกันและต่อเนื่องกัน

2.1.1 หลักการของวิธี SSADM

SSADM แบ่งการพัฒนาออกเป็น 3 ส่วนคือ

- กระบวนการ (Processes)
- ข้อมูล (Data)
- เหตุการณ์ (Event)

ในแต่ละส่วนมีความสำคัญและมีความเกี่ยวข้องกัน ส่วนของกระบวนการเป็นการทำงานในแต่ละฟังก์ชัน (Function) หรือ โพรซีเจอร์ (Procedure) ที่มีข้อมูลผ่านเข้ามา ส่วนข้อมูลเป็นส่วนสำคัญของระบบคือเป็นฐานข้อมูล (Database) ของระบบและสุดท้ายคือเหตุการณ์เป็นการมองในรูปแบบของสิ่งที่จะเกิดขึ้นในระบบและอาจเปลี่ยนแปลงข้อมูลที่มีอยู่ในระบบได้ ส่วนรายละเอียดในหลักการของงาน 3 ส่วนคือ กระบวนการ ข้อมูลและเหตุการณ์มีรายละเอียดดังนี้

- กระบวนการ ใช้เทคนิคในการออกแบบที่เรียกว่า DFM (Data Flow Modeling) เป็นส่วนที่ออกแบบกระบวนการที่จะทำงานเมื่อได้รับข้อมูลจากฐานข้อมูล โดยภายในของ DFMประกอบไปด้วย DFDs ย่อย ๆ การสร้าง DFDs ใช้วิธีแบบบนลงล่าง (Top-Down) DFDs ชั้นบนสุดเรียกว่า คอนเท็กซ์ไดอะแกรม (Context Diagram) ที่แสดงถึงภาพรวมทั้งหมดของความสัมพันธ์ระหว่าง DFDs ย่อย ๆ ในขณะที่ DFDs ย่อยจะแสดงรายละเอียดของโพรเซส เอนติตี้ (Entity) และดาต้าสโตร์ (Data Store)

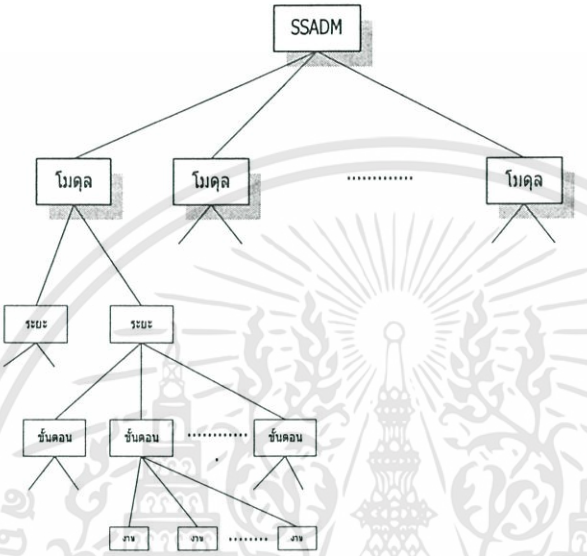
- ข้อมูล ใช้เทคนิคการออกแบบที่เรียกว่า LDM (Logical Data Modeling) เป็นส่วนของการออกแบบฐานข้อมูลและความสัมพันธ์กันระหว่างแฟ้มข้อมูล ภายในของ LDM ประกอบไปด้วย LDS (Logical Data Store) และรายละเอียดของ LDS ซึ่งในรายละเอียดของขั้นตอนนี้จะเหมือนกับการพจนานุกรมข้อมูลในแต่ละแฟ้มข้อมูล

- เหตุการณ์ คือการกระทำใด ๆ ที่ก่อให้เกิดการเปลี่ยนแปลงของดาต้าที่อยู่ในระบบ (write,store) ของแต่ละเอนติตีมีลักษณะเป็นการรายงานเหตุการณ์ที่เกิดขึ้นในแต่ละขั้นตอนตั้งแต่เริ่มจนถึงสิ้นสุดของการกระทำ โดยในส่วนนี้จะให้การวาดแผนภูมิที่เรียกว่า ELHs (Entity Life History) ซึ่งในแต่ละเอนติตีจะมี ELHs อยู่ด้วยเสมอเพื่อแสดงให้เห็นว่าในแต่ละเอนติตีมีการกระทำและการเปลี่ยนแปลงอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างของวิธี SSADM

โครงสร้างของ SSADM แบ่งกิจกรรมการทำงานเป็นแบบ Top-Down คือ ระดับแรกแบ่งเป็นโมดูล (Module) จากโมดูลแบ่งเป็นระยะ (Stage) จากระยะเป็นขั้นตอน (Step) และจากขั้นตอนเป็นงาน (Task) ซึ่งงานเป็นกิจกรรมย่อยที่สุด ที่สามารถปฏิบัติ ตรวจสอบ และประเมินผลได้ (ดังรูปที่ 2.1)



รูปที่ 2.1 แสดงโครงสร้างของวิธี SSADM

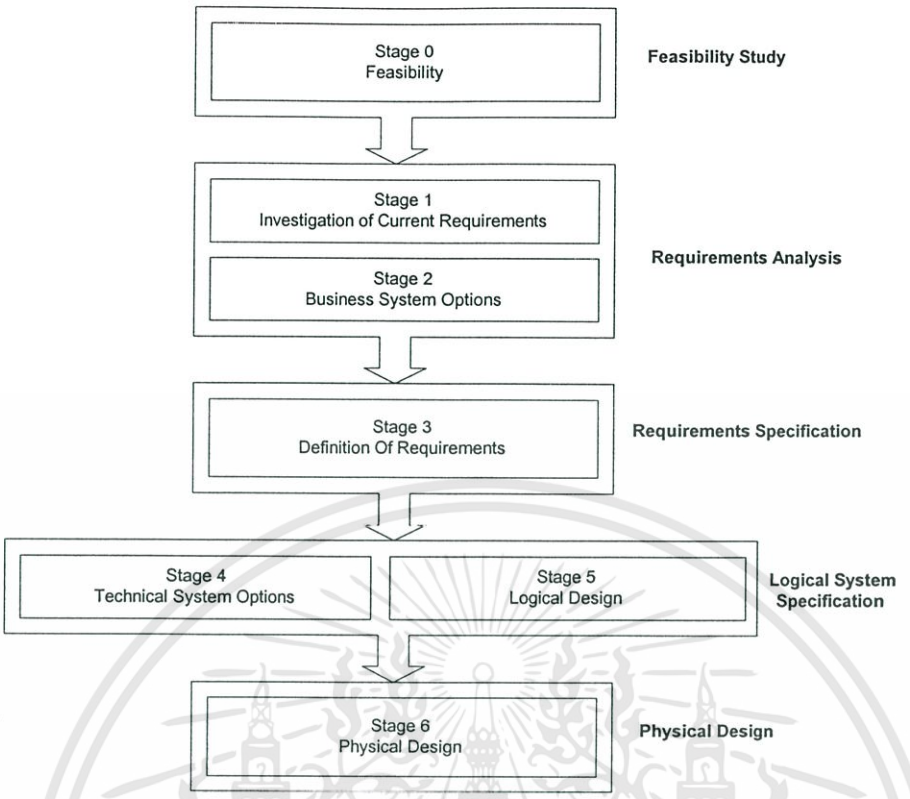
2.3 ขั้นตอนของ SSADM

ขั้นตอนของ SSADM เวอร์ชั่น 4.2 แบ่งการทำงานออกเป็น 5 โมดูลคือ

- Feasibility Study
- Requirements Analysis
- Requirements Specification
- Logical System Specification
- Physical Design

แต่ละโมดูลแบ่งออกเป็นขั้นตอนที่ดังนี้ (ดังรูปที่ 2.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงระยะของ SSADM

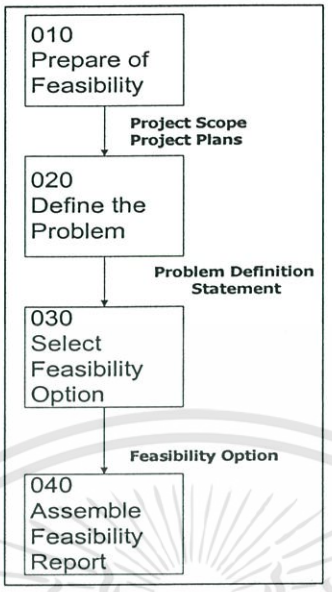
<u>ระยะที่ 0 การศึกษาความเหมาะสม (Stage 0 : Feasibility)</u>

การศึกษาความเหมาะสม คือ ระยะเริ่มต้นของการพัฒนาระบบสารนสนเทศ ซึ่งเป็นขั้นตอนที่มีความสำคัญกับระบบ เพราะการพัฒนาระบบสารสนเทศระบบใหม่จากความต้องการของผู้ใช้นั้น ผู้วิเคราะห์จำเป็นต้องทราบถึงขอบเขตของปัญหา สภาพแวดล้อมการทำงานของระบบเดิม และความต้องการของผู้ใช้ทั้งหมดก่อนตัดสินพัฒนาระบบใหม่ ซึ่งระยะที่ 0 เป็นระยะเริ่มแรกแผนภูมิที่ใช้คือ DFM และ LDM โดยมีขั้นตอนการทำงานดังต่อไปนี้ (ดังแสดงรูปที่ 2.3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระยะที่ 0 : การศึกษาความเหมาะสม



รูปที่ 2.3 แสดงขั้นตอนการทำงานของระยะที่ 0

ขั้นตอนที่ 010 เตรียมการเพื่อทำการศึกษาความเหมาะสมของโครงงาน (Prepare for the Feasibility Study)

การศึกษาความเหมาะสมของโครงงานเป็นสิ่งเริ่มต้นของการพัฒนาระบบเพื่อหาขอบเขตของปัญหาและวางแผนงานสำหรับศึกษาสภาพแวดล้อมของระบบเดิม การศึกษาระบบเดิมของขั้นตอนนี้จะมองภาพรวมของระบบทั้งหมด โดยใช้เทคนิคของ DFM และ LDM พร้อมกับรายละเอียดความต้องการต่าง ๆ ของผู้ใช้ ซึ่งทำให้ทั้งผู้วิเคราะห์เข้าใจปัญหาและความต้องการของผู้ใช้มากขึ้น การทำงานของขั้นตอนนี้มีดังนี้

งาน 10 กำหนดขอบเขตโครงงานโดยสร้าง Model ของระบบโดยรวม พร้อมกับออกแบบความต้องการของระบบใหม่แบบกว้าง ๆ

งาน 20 ศึกษาขอบเขตและความต้องการของผู้ใช้

งาน 30 วางแผนในการศึกษาความเหมาะสม

ขั้นตอนที่ 020 กำหนดขอบเขตของปัญหาที่เกิดขึ้น (Define the Problem)

การกำหนดของเขตของปัญหา คือ ขั้นตอนที่ปฏิบัติต่อเนื่องจากขั้นตอนที่ 010 ที่ได้ศึกษาความเหมาะสม แต่ขั้นตอนนี้คือพิจารณาถึงความเป็นไปได้ของความต้องการที่จะเกิดขึ้นกับระบบใหม่และพิจารณาความสามารถเดิมของระบบไปพร้อมกัน เช่น ความสามารถใดที่มีอยู่นั้นมีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

44089

ล้าสมัยนั้นและสามารถปรับปรุงแก้ไขได้หรือไม่ หรือสร้างใหม่ เป็นต้น พร้อมกับบันทึกรายละเอียดต่าง ๆ ลงในเอกสาร และขั้นตอนที่ 020 มีการทำงานดังต่อไปนี้

งาน 10 สร้างโครงร่างรายละเอียดของความต้องการโดยใช้ LDS และ DFM

งาน 20 ขยายขอบเขตรายละเอียดของ LDS และ DFM ที่ได้จาก ขั้นตอนที่ 010 ในรายละเอียดของโครงสร้างปัจจุบัน, บันทึกปัญหากับการทำงานในปัจจุบันลงในตารางความต้องการ (Requirement Catalogue)

งาน 30 บันทึกความต้องในระบบลงใน User Catalogue

งาน 40 กำหนด Function Requirement และ Non-Function Requirement ของระบบใหม่ และ บันทึกลงใน Requirement Catalogue

งาน 50 สรุปปัญหาของระบบตามความต้องการของผู้ใช้

งาน 60 ตกลงปัญหาของระบบกับผู้บริหารโครงการ

ขั้นตอนที่ 030 จัดทำแนวทางเลือกของการพัฒนาระบบให้ผู้ใช้ (Select Feasibility Option)

ขั้นตอนที่ 030 คือ ขั้นตอนนำเสนอทางเลือกทางธุรกิจและเทคนิคของการพัฒนาระบบ ทางเลือกทางธุรกิจ คือ การนำเสนอแนวทางเลือกทางธุรกิจเพื่อแก้ปัญหาที่เกิดขึ้นกับระบบที่กำลังพัฒนา รายละเอียดของแต่ละทางเลือกต้องขึ้นกับความต้องการที่ผู้ใช้ต้องการ ซึ่งทางเลือกที่นำเสนอผู้วิเคราะห์ควรนำเสนอทางเลือกหลายทางเลือก ภายในรายละเอียดของแต่ละทางเลือกควรประกอบด้วย แนวทางการแก้ไขปัญหา การตอบสนองของแต่ละทางเลือก และ อื่น ๆ ส่วนทางเลือกทางเทคนิค คือ รายละเอียดทางทางเทคนิคที่จะนำเสนอในระบบใหม่ ซึ่งประกอบด้วย สภาพแวดล้อมของระบบ เทคนิคสำหรับการนำเสนอ ซอฟต์แวร์และฮาร์ดแวร์ สำหรับขั้นตอนที่ 030 ผู้วิเคราะห์นำเสนอรายละเอียดของทางเลือกทางธุรกิจและทางเลือกทางเทคนิคแบบย่อให้ผู้ใช้เลือกเพื่อเป็นแนวทาง ส่วนรายละเอียดของแต่ละทางเลือกนั้นจะกล่าวในระยะที่ 2 และระยะที่ 4 การนำเสนอทางเลือกทั้งสองของขั้นตอนนี้มีการทำงานดังนี้

งาน 10 กำหนดความต้องการของระบบใหม่

งาน 20 เสนอโครงร่างของ BSOs อย่างน้อย 6 ทางเลือก

งาน 30 กำหนดโครงร่างของ TSO ที่สนับสนุนโครงร่างของ BSO

งาน 40 รวม TSO กับ BSO เข้าด้วยกันในทางเลือกที่เป็นไปได้โดยเลือกออกมาที่ดีที่สุด 3 ทางเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน 50 กำหนดรายละเอียดของแต่ละทางเลือกจาก งาน 40

งาน 60 กำหนดโครงร่างของโครงงานในแต่ละทางเลือกที่ทำการศึกษาความเหมาะสม

งาน 70 แสดงทางเลือกที่ถูกเลือกจากผู้บริหาร

งาน 80 พัฒนาทางเลือกที่เลือก ถ้าทางเลือกนั้นไม่สามารถดำเนินการได้ก็จะทำให้โครงงานนั้นชะงัก

ขั้นตอนที่ 040 รวบรวมและจัดทำรายงานของการศึกษาความเหมาะสม

ขั้นตอนที่ 040 เป็นขั้นตอนสุดท้ายของระยะที่ 0 ซึ่งจะรวบรมเอกสารและแผนภูมิของแต่ละขั้นตอนรวมทั้งตรวจสอบความถูกต้องทั้งหมดก่อนข้ามไปปฏิบัติงานระยะต่อไป

ผลลัพธ์ของระยะที่ 0 คือ

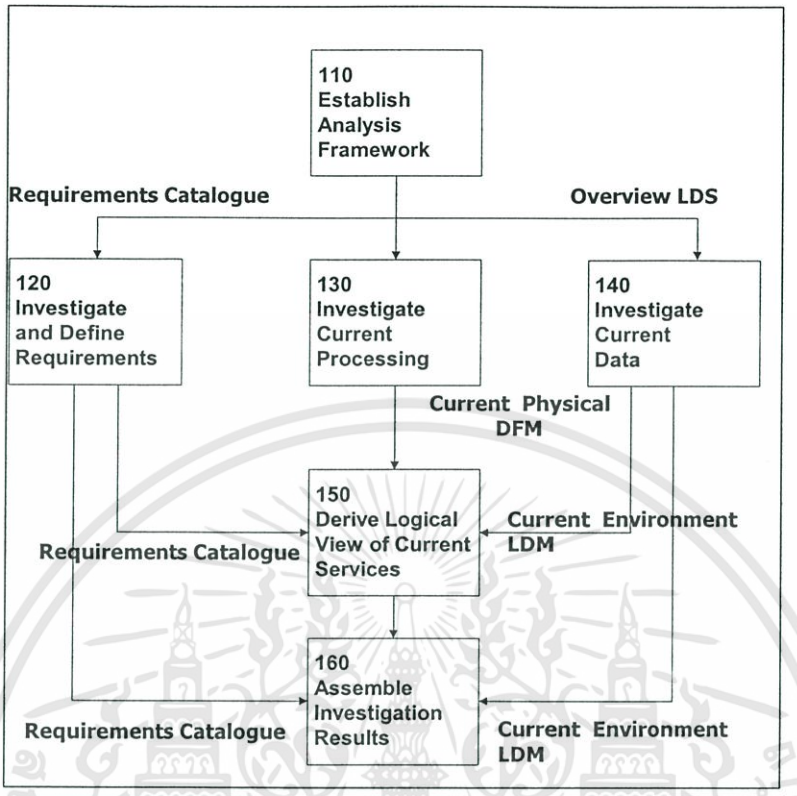
1. แผนภูมิการไหลของข้อมูล (Data Flow Diagram) และแผนภูมิของข้อมูล (Logical Data Store) ซึ่งทั้ง 2 แผนภูมิเป็นภาพรวมของระบบเดิม
2. รายละเอียดความต้องการของผู้ใช้ที่ปรากฏกับระบบใหม่ (User Requirements)

<u>ระยะที่ 1 การศึกษาระบบเดิม (Stage 1 : Investigation of Current Environment)</u>

ระยะที่ 1 คือ การนำระบบเดิมที่ได้จากระยะที่ 0 มากระจายรายละเอียดในแต่ละส่วน เช่น แผนภูมิการไหลของข้อมูลต้องกระจายไปสู่ขั้นตอนการทำงานสุดท้ายหรือที่เรียกว่า EP (Elementary Process) และแผนภูมิของข้อมูลจะพิจารณาถึงความสัมพันธ์ในแต่ละส่วนที่เชื่อมโยงกับแผนภูมิการไหลของข้อมูลว่ามีความสัมพันธ์มากน้อยเพียงใด เป็นต้น ซึ่งขั้นตอนนี้ผู้พัฒนาจะต้องเข้าไปศึกษาถึงพฤติกรรมของระบบทั้งในด้านกายภาพและตรรกะเพื่อบันทึกสิ่งที่เกิดกับระบบแล้วนำสิ่งเหล่านี้มาประกอบในการสร้างระบบใหม่ โดยมีขั้นตอนการทำงานดังนี้ต่อไปนี้ (แสดงดังรูปที่ 2.4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะที่ 1 : การศึกษาระบบเดิม



รูปที่ 2.4 แสดงขั้นตอนการทำงานของระยะที่ 1

ขั้นตอนที่ 110 วิเคราะห์ระบบเดิม

ขั้นตอนที่ 110 นี้จะนำสิ่งที่ได้จากการศึกษาความเป็นไปได้มาปฏิบัติต่อ โดยนำแผนภูมิทางกายภาพของระบบเดิมพร้อมกับเอกสารต่าง ๆ ที่ได้มาจาก ขั้นตอนที่ 010 และ 020 มาตรวจสอบใหม่อีกครั้ง เริ่มจาก แผนภูมิการไหลของข้อมูล LDS และ Requirements Catalogue ตามลำดับ แผนภูมิการไหลของข้อมูล และ LDS จะแสดงถึงการทำงานของระบบปัจจุบัน เช่น การติดต่อกับผู้ใช้หรือการจัดการฐานข้อมูลให้เข้าใจตรงกับผู้ใช้อีกครั้ง เป็นต้น ถ้ายังไม่เป็นสิ่งที่ถูกต้อง ควรกลับไปเก็บข้อมูลใหม่กับผู้ใช้อีกครั้ง ส่วน Requirements Catalogue กล่าวถึงส่วนของการศึกษาความเหมาะสมในเรื่องของเวลาและต้นทุนสำหรับการพัฒนาถึงความเหมาะสมของระบบที่ศึกษามานั้นมีความจำเป็นมากน้อยเพียงใดที่จะต้องพัฒนาต่อไป ซึ่งความต้องการของผู้ใช้บางครั้งจำเป็นต้องคัดออกจากระบบเนื่องจากความต้องการบางส่วนไม่สามารถเกิดขึ้นกับระบบได้ผู้วิเคราะห์จึงต้องคัดเฉพาะความต้องการที่สามารถเกิดขึ้นจริงในระบบเท่านั้น หลังจากตรวจสอบแผนภูมิและเอกสารต่าง ๆ เสร็จสิ้นถัดมา คือ การนำเครื่องมือต่าง ๆ เข้ามาช่วยในการวางแผนงานสำหรับพัฒนาระบบ เช่น PERT/CPM หรือ Grant Chart [12],[26]เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 120 สำรวจปัญหาของระบบเดิม

ขั้นตอนที่ 120 เป็นการนำ Requirements Catalogue ที่ได้จาก ขั้นตอนที่ 110 มาปรับปรุงและเพิ่มรายละเอียดของ Users' Requirements เพื่อประเมินถึงความเป็นไปได้ของความต้องการที่จะเกิดขึ้นในระบบ จากนั้นนำรายละเอียดที่สมบูรณ์ไปพัฒนาต่อในขั้นตอนที่ 130 และ ขั้นตอนที่ 140 ถ้าผู้พัฒนาละเลยการทำงานในส่วนนี้ย่อมทำให้ไม่สามารถนำสิ่งที่จะเกิดขึ้นใหม่ของผู้ใช้มาปรับปรุงหรือสร้างระบบใหม่ให้เกิดขึ้นจริงได้ แต่ถ้ามีการศึกษาความเหมาะสมที่เพียงพอย่อมทำให้ระบบที่กำลังจะพัฒนามีข้อมูลอ้างอิงสามารถพัฒนาต่อไปได้

ขั้นตอนที่ 130 สำรวจกระบวนการทำงานของระบบเดิม

ขั้นตอนที่ 130 คือ การขยายงานต่อจากขั้นตอนที่ 110 ที่แสดงถึงรายละเอียดของกระบวนการทำงานของระบบเดิมโดยแผนภูมิการไหลของข้อมูล เพื่อแสดงถึงการทำงานของระบบเดิมให้ชัดเจนมากขึ้นและมีการทำงานดังต่อไปนี้

งาน 10 สร้าง แผนภูมิการไหลของข้อมูลของระบบงานเดิม

งาน 20 ทบทวนและขยาย แผนภูมิการไหลของข้อมูล ลงไปในระดับชั้นที่ 1

งาน 30 แสดงรายละเอียดของ แผนภูมิการไหลของข้อมูล ระดับชั้นที่ 1 ไป 2 และ 3 ตามลำดับ

งาน 40 สร้าง EPDs แสดงรายละเอียดของ External Entity และ แสดงรายละเอียดของข้อมูลเข้าออก

งาน 50 ปรับปรุง Requirements Catalogue ให้สอดคล้องกับความต้องการในระบบใหม่หรือปัญหาที่เกิดขึ้น

ขั้นตอนที่ 140 สำรวจข้อมูลของระบบเดิม (Investigation of Current Data)

การสำรวจข้อมูลของระบบเดิม คือ การนำ LDM ที่เกิดขึ้นจากการศึกษาความเหมาะสมมาปรับปรุงใหม่และทำการจัดกลุ่มใหม่ เพื่อให้ LDM ที่เกิดขึ้นให้มีความกระชับและชัดเจนมากขึ้น วิธีการทำงาน คือ ทำตามขั้นตอนและแนวคิดทางตรรกะ ที่ถูกต้อง นอกจากนี้เราจะต้องนำ Data ที่อยู่ใน Requirements Catalogue เข้ามาเพิ่มเติมในการพิจารณาด้วยเพื่อที่จะทำให้ Current Data มีความสมบูรณ์มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน 10 ทบทวนและปรับปรุง LDM กรณีที่ความสัมพันธ์ระหว่างเอนติตีมีความสัมพันธ์แบบ M:N ให้เปลี่ยนเป็นแบบ 1:M เพื่อทำให้การทำงานระหว่างเอนติตีดีขึ้น (ในด้านการค้นหาและปรับปรุงในตัวของ Entity เอง)

งาน 20 กำหนดคีย์หลัก (Primary Key) ในแต่ละเอนติตี

งาน 30 ตรวจสอบ LDM ที่จะสนับสนุนกระบวนการสุดท้าย (Elementary Process) ในแผนภูมิการไหลของข้อมูล พร้อมกับบันทึกลงใน Process/Entity Matrix เพื่อที่จะทราบว่าใน Entity ใดมีการกระทำสิ่งใน LDM และตัว เช่น เปลี่ยนแปลง , สร้าง , ลบ เป็นต้น

งาน 40 บันทึกข้อมูลใหม่ตามความต้องการ หรือ ปัญหาของระบบเดิมใน Requirement Catalogue

ขั้นตอนที่ 150 : Derive Logical View of Current Service

จากขั้นตอนที่ 130 แผนภูมิการไหลของข้อมูลเป็นแผนภูมิที่กล่าวถึงแผนภูมิทางกายภาพซึ่งเป็นการแสดงถึงการทำงานต่าง ๆ ของระบบเดิมที่นักวิเคราะห์ได้จากการสอบถามผู้ใช้ระบบ ในขั้นตอนนี้คือการนำแผนภูมิทางกายภาพทั้งหมดที่มีอยู่มาเปลี่ยนไปสู่ระดับตรรกะเพื่อแสดงให้เห็นถึงการทำงานต่าง ๆ ที่เกิดขึ้นในระบบ ซึ่งมีลำดับการทำงานดังนี้

งาน 10 จากแผนภูมิการไหลของข้อมูลระดับกายภาพของระบบปัจจุบัน (Current Physical DFM) เดิมที่ได้จากขั้นตอนที่ 130 นำมาปรับปรุงโดยเปลี่ยนจากระดับกายภาพของแต่ละกระบวนการไปสู่เซอร์วิสทางตรรกะ (Logical Service) เพื่อทำให้เห็นการทำงานชัดขึ้นว่าในระบบมีฟังก์ชันอะไรที่เป็นงานหลักและสนับสนุนของระบบเดิม ซึ่งบางฟังก์ชันอาจต้องนำกลับมาใช้กับระบบใหม่ได้ นอกจากนี้ควรนำแผนภูมิการไหลของข้อมูลที่ได้มาปรับเปลี่ยน โดยนำส่วนที่เป็นกระบวนการสุดท้ายของกระบวนการหลัก (EPDs) มาทำการปรับเปลี่ยนให้เหมาะสมและต้องคำนึงถึง Requirement Catalogue ที่เกิดขึ้นด้วย สิ่งที่ได้จาก งาน นี้เราจะสามารถนำไปกระทำต่อได้ในระยะที่ 2 และ 3 ได้อีก

งาน 20 ตรวจสอบความถูกต้องของดาต้าสโตร์ให้เหมาะสมโดยจะต้องทำการ Group LDS ที่มีความเกี่ยวเนื่องกัน เช่น Product , Product Type, Product Type Zone Allocation เป็นต้น มาไว้ด้วยกันเพื่อทำการสร้างฐานข้อมูลของระบบเดิม ให้เห็นได้ชัดเจนมากขึ้น นอกจากทำการตรวจสอบ LDS แล้วยังต้องตรวจสอบกระบวนการต่างที่เกิดขึ้นไปพร้อมกันโดยเฉพาะกระบวนการสุดท้าย (Elementary Processes)ให้มีความเหมาะสม คือ ถ้ากระบวนการใดมีความเกี่ยวข้องกันแต่มีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานแยกจากกันควรนำมารวมเป็นกระบวนการเดียวกัน หรือการส่งข้อมูลกระหว่างกระบวนการมีลักษณะในรูปของการดึงข้อมูลมาจากฐานข้อมูลควรให้มีลักษณะการเชื่อมต่อในรูปของการเชื่อมต่อผ่านฐานข้อมูลแทน เป็นต้น

งาน 30 เมื่อผ่านการทำงานใน งาน ที่ 10 และ 20 มาแล้ว ให้นำสิ่งที่ได้มาปรับปรุงและสร้างแผนภูมิการไหลของข้อมูลใหม่เพื่อให้สอดคล้องกับขั้นตอนที่ทำมาก่อนหน้านี้ โดยเริ่มทำจากท้ายขึ้นมาก่อน คือ ทำจากการบวนการสุดท้ายไปสู่กระบวนการหลัก

งาน 40 ตรวจสอบความถูกต้องของแผนภูมิการไหลของข้อมูลทางตรรกะ นอกจากนี้ยังต้องทำในส่วนของ LDS ด้วยโดยจะต้องทำการตรวจสอบกับ Process/Entity Matrix (ในเอกสารชุดนี้จะบอกถึงว่าในแต่ละ Entity หรือ Process มีความเกี่ยวข้องอย่างไรกับฐานข้อมูลในขั้นตอนที่การทำงานที่เกิดขึ้น)

งาน 50 ปรับปรุง Requirement Catalogue และข้อสงสัยต่าง ๆ ใน Physical ของระบบเดิมให้ชัดเจนก่อนที่เราจะปรับเปลี่ยนไปสู่ระบบใหม่

ขั้นตอนที่ 160 : จัดทำรายงานจากการศึกษาระบบเดิม

ขั้นตอนที่ 160 เป็นขั้นตอนสุดท้ายของระยะที่ 1 ซึ่งจะรวบรมเอกสารและแผนภูมิของแต่ละขั้นตอนรวมทั้งตรวจสอบความถูกต้องทั้งหมดก่อนข้ามไปปฏิบัติงานระยะต่อไป

ผลลัพธ์ของระยะที่ 1 คือ

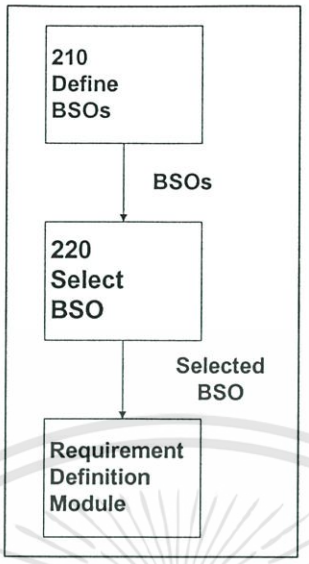
1. แผนภูมิการไหลของข้อมูลระบบเดิมในทางตรรกะและกายภาพได้ชัดเจนยิ่งขึ้น
2. แผนภูมิข้อมูลกับตารางความสัมพันธ์กับเอนติตี
3. รายละเอียดความต้องการของผู้ใช้ (User Requirements / User Catalogue)

<u>ระยะที่ 2 เสนอทางเลือกทางธุรกิจ (Stage 2 : Business System Options)</u>

ระยะที่ 2 คือ การนำเสนอแนวทางของการพัฒนาระบบ เช่น ประเมินค่าใช้จ่ายต่าง ๆ ที่เกิดขึ้นกับระบบ ระยะเวลาของการพัฒนา ประโยชน์ที่จะได้รับและผลกระทบกับระบบเดิม เป็นต้น ซึ่งแนวทางการพัฒนาระบบผู้พัฒนาต้องเสนอทางเลือกอย่างน้อย 3 ทางเลือกให้กับผู้ใช้ได้พิจารณา ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้ (แสดงดังรูปที่ 2.5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะที่ 2 : เสนอทางเลือกทางธุรกิจ



รูปที่ 2.5 แสดงขั้นตอนการทำงานของระยะที่ 2

ขั้นตอนที่ 210 กำหนดทางเลือก

งาน 10 กำหนดความต้องในระบบใหม่

งาน 20 กำหนดทางเลือกไว้ให้เลือกอย่างน้อย 6 ทางเลือก

งาน 30 ตรวจทานโครงร่างของทางเลือกให้กับผู้ใช้ และเลือกทางเลือกที่มีโอกาสเป็นไปได้ออกมา 2 - 3 ทางเลือก

งาน 40 เพิ่มรายละเอียดในทางเลือกที่เลือก (เพื่อเป็นส่วนเสริมและแสดงถึงความสอดคล้องระหว่าง DFM กับ LDM ที่เกี่ยวข้องมากขึ้น)

งาน 50 จัดทำต้นทุนของการวิเคราะห์ในแต่ละทางเลือก

ขั้นตอนที่ 220 เลือกแนวทางที่เหมาะสมกับการพัฒนาระบบ (ขั้นตอนนี้ผู้ใช้เป็นผู้เลือก)

ผลลัพธ์ของระยะที่ 2

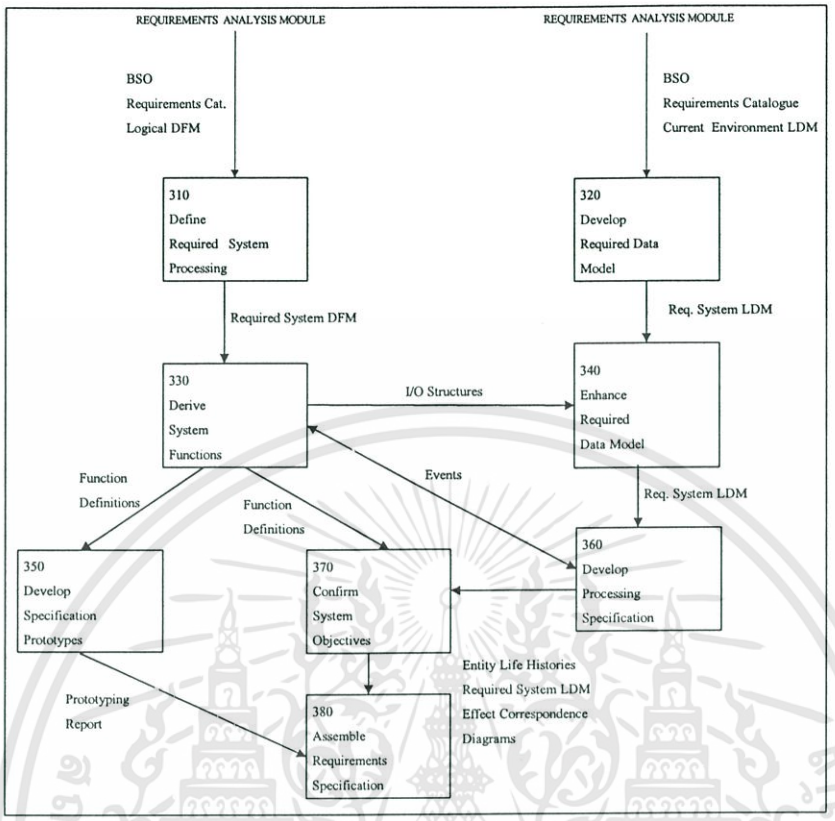
รายงานทางเลือกของระบบที่เจ้าของโครงการได้เลือกไว้ จากนั้นนำทางเลือกดังกล่าวไปรวมกับความต้องการอื่น ๆ เพื่อนำไปประกอบการวิเคราะห์และออกแบบในระยะต่อไป

ระยะที่ 3 กำหนดความต้องการของระบบใหม่ (Stage 3 : Definition of Requirements)

ขั้นตอนนี้เป็นส่วนสำคัญที่สุดของการพัฒนาระบบสารสนเทศ เพราะผู้พัฒนานำรายละเอียดของขั้นตอนก่อนหน้ามารวบรวมเพื่อพัฒนาเป็นระบบใหม่ ซึ่งแนวทางปฏิบัติของขั้นตอนนี้มีลำดับการทำงานดังนี้ (แสดงในรูปที่ 2.6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะที่ 3 : กำหนดความต้องการของระบบใหม่



รูปที่ 2.6 แสดงขั้นตอนการทำงานของระยะที่ 3

ขั้นตอนที่ 310 กำหนดความต้องการของกระบวนการภายในระบบ

การกำหนดความต้องการของระบบ คือ การนำแผนภาพการไหลของข้อมูลทางตรรกะจากระยะที่ 1 กับความต้องการของผู้ใช้และแนวทางการพัฒนา (BSO) มาปรับปรุงให้กับตรงกับระบบที่พัฒนาและมีการทำงานดังนี้

งาน 10 ปรับปรุง Requirement Catalogue ให้สอดคล้องกับ BSO ที่เลือก ใน ขั้นตอนที่ 220

งาน 20 ปรับปรุงและเพิ่มเติมกระบวนการในแผนภูมิการไหลของข้อมูลระดับที่ 1 ตามความต้องการของระบบใหม่ และแยกกระบวนการภายนอกที่มีผลกับ BSO ที่ถูกเลือก

งาน 30 ปรับปรุงกระบวนการระดับล่างและปรับปรุง Cross-reference ที่สัมพันธ์กับ Requirement Catalogue

งาน 40 ปรับปรุงกระบวนการพื้นฐาน (Elementary Process) อินพุต-เอาต์พุตและรายละเอียดของเอนติตีภายนอก (External Entity)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน 50 ตรวจสอบองค์ประกอบของ LDS กับเอนติตีที่มีอยู่ใน Requirement System LDM และ LDS กับ Entity Cross-Reference นั้นจะต้องสามารถสนับสนุน แผนภูมิการไหลของข้อมูลทั้งหมดที่มีอยู่

งาน 60 สร้างกฎเกณฑ์ของผู้ใช้และตรวจสอบองค์ประกอบของ External Entity กับดาต้าโฟลว์ (Data flow)

ขั้นตอนที่ 320 สร้างแบบจำลองข้อมูลของระบบที่ต้องการ (Develop Required Data Model)

ขั้นตอนที่ 320 คือ ขั้นตอนที่สำคัญของการจัดการฐานข้อมูลก่อนเข้าสู่ขั้นตอนการทำบรรทัดฐานฐานข้อมูล ซึ่งก่อนทำขั้นตอนดังกล่าวนั้น ผู้วิเคราะห์จะต้องนำ LDM ที่ได้มาพิจารณาถึงความถูกต้องในระดับของความสัมพันธ์ที่เกิดขึ้นระหว่างเอนติตีก่อน จากนั้นควรเพิ่มเติมรายละเอียดและข้อบังคับต่าง ๆ ของแต่ละเอนติตี เช่น รายละเอียดของแอตทริบิวต์ รายละเอียดของเอนติตีและข้อบังคับต่าง เป็นต้น ซึ่งขั้นตอนนี้มีการทำงานย่อยดังต่อไปนี้

งาน 10 ปรับปรุง LDM พร้อมกับเพิ่ม Data Requirement ในระบบใหม่ และแยกองค์ประกอบที่สำคัญภายนอกของ LDM ซึ่งมีผลกับ BSO ที่เลือก พร้อมกับทำเอกสารของเอนติตีให้สมบูรณ์รวมทั้งกำหนดแอตทริบิวต์และเปลี่ยน Cross-Reference ที่สัมพันธ์กับ Requirement Catalogue ทั้งหมด

งาน 20 ตรวจสอบ LDM ที่สนับสนุนกระบวนการพื้นฐานใน DFM ของระบบ

งาน 30 ปรับปรุง LDM จาก Requirement Catalogue ให้เหมาะสมกับ Non-Function Requirements

ขั้นตอนที่ 330 สร้างฟังก์ชันของระบบ (Derive System Functions)

การออกแบบฟังก์ชันให้กับระบบใหม่นั้นให้ผู้วิเคราะห์ระบบนำแผนภาพการไหลของข้อมูลกับความต้องของผู้ใช้มาประกอบการออกแบบ เพื่อออกแบบโครงสร้างของข้อมูลเข้าและออก (I/O Structure) พร้อมกับกำหนดกฎเกณฑ์การทำงานของแต่ละฟังก์ชัน ลำดับการงานของงานย่อยมีดังนี้

งาน 10 กำหนดและเปลี่ยนแปลงฟังก์ชันจาก DFM และคำแนะนำของผู้ใช้ ตรวจสอบฟังก์ชันแต่ละตัวให้เกี่ยวข้องกับกระบวนการพื้นฐานอย่างน้อยหนึ่งฟังก์ชันเป็นอย่างน้อย พร้อมกับกำหนดเหตุการณ์และองค์ประกอบของแต่ละฟังก์ชัน

งาน 20 กำหนดฟังก์ชันจาก Requirement Catalogue, DFM และคำแนะนำของผู้ใช้

งาน 30 สร้างอินพุต-เอาต์พุตของแต่ละฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน 40 สร้าง User Role/Function Matrix พร้อมกับกำหนด Dialogue ที่เกี่ยวกับการวิเคราะห์

งาน 50 ระบุรายละเอียดเซอร์วิส (Service) ของแต่ละฟังก์ชัน

ขั้นตอนที่ 340 ขยายแบบจำลองข้อมูลของระบบที่ต้องการ (Enhance Required Data Model)

การจัดทำบรรทัดฐานฐานข้อมูล เป็นขั้นตอนที่สำคัญสำหรับการจัดการฐานข้อมูลที่ผู้วิเคราะห์ได้ออกแบบไว้ เพราะฐานข้อมูลที่ปรากฏนั้นบางครั้งยังมีความซ้ำซ้อนกันอยู่ภายในเอนติตีเดียวกันหรือระหว่างเอนติตี ดังนั้นขั้นตอนที่ 340 นั้นจึงมีความจำเป็นที่ผู้วิเคราะห์ต้องนำมาจัดทำเพื่อลดความซ้ำซ้อนเหล่านั้น ซึ่งผู้วิเคราะห์ควรเริ่มจาก LDS ไดอะแกรมที่ได้จากขั้นตอนที่ 320 ที่ได้ปรับปรุงความสัมพันธ์เรียบร้อยแล้วโดยแปลงจากไดอะแกรมไปสู่รูปของ Normal Form และนำไปสู่การทำบรรทัดฐานฐานข้อมูล ซึ่งมีลำดับการทำงานดังนี้

งาน 10 กำหนดโครงสร้างอินพุต-เอาต์พุตให้อยู่ในรูป Normal Form

งาน 20 ทำบรรทัดฐานฐานข้อมูล (Normalization) ตารางให้เป็น 3NF

งาน 30 สร้าง LDM จากตารางที่เป็น 3NF

งาน 40 เปรียบเทียบ LDM ที่สร้างจากงาน 30 กับ Require System LDM และหาข้อแตกต่างที่เกิดขึ้นหลังจากทำบรรทัดฐานฐานข้อมูล

ขั้นตอนที่ 350 สร้างแบบจำลองของระบบใหม่

การสร้างแบบจำลอง (Prototype) เป็นการจำลองระบบก่อนสร้างระบบจริงให้กับผู้ใช้ซึ่งแบบจำลองผู้วิเคราะห์นำรายละเอียดและความต้องการต่าง ๆ ที่ศึกษาไว้ข้างต้นนำมาสร้าง การสร้างแบบจำลองนั้นควรสร้างให้คล้ายกับระบบจริงที่จะต้องนำเสนอผู้ใช้ตอนสิ้นสุดโครงการ ระหว่างการนำเสนอต้นแบบของระบบควรบันทึกรายการที่บกพร่องและที่ต้องเพิ่มเติมตามที่ผู้ใช้ต้องการเพื่อที่จะนำรายละเอียดเหล่านั้นมาปรับปรุงให้กับระบบต่อไป ซึ่งการสร้างแบบจำลองให้กับระบบนั้นมีลำดับการทำงานย่อยดังต่อไปนี้

งาน 10 กำหนดส่วนของการติดต่อหลักของแบบจำลอง (Prototype)

งาน 20 สร้างโครงสร้างเมนูให้กับแบบจำลอง

งาน 30 สร้างเส้นทาง (Pathways) การติดต่อให้กับแบบจำลองแต่ละส่วน

ทำซ้ำในงานที่ 40-70 จนกระทั่งได้แบบจำลองที่ต้องการ

งาน 40 สร้างแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน 50 เตรียมแบบจำลองแต่ละส่วนเพื่อนำเสนอให้กับผู้ใช้

งา น 60 สาธิตแบบจำลองให้ผู้ใช้

งา น 70 นำแบบจำลองในแต่ละส่วนที่สาธิตไปมาพิจารณาใหม่ว่าคุ้มค่าหรือไม่

งาน 80 ปรับปรุงแบบจำลองให้ตรงกับความต้องการของผู้ใช้และจัดทำรายงานทั้งหมด

ขั้นตอนที่ 360 กำหนดลักษณะเฉพาะของกระบวนการ (Develop Processing Specification)

ขั้นตอนที่ 360 เป็นขั้นตอนต่อเนื่องจากการวิเคราะห์เพื่อหาระบบใหม่ ซึ่งการวิเคราะห์ที่ผ่านมานั้น คือ การออกแบบและกำหนดฟังก์ชันต่าง ๆ โดยใช้แผนภูมิการไหลของข้อมูลของระดับต่าง ๆ พร้อมกับรายละเอียดเป็นสิ่งกำหนดกระบวนการทำงานและลักษณะของฟังก์ชันที่เกิดขึ้นในระบบใหม่ แต่ขั้นตอนนี้การออกแบบฟังก์ชันจะแตกต่างออกไป โดยใช้ "Entity-Event Modeling" เป็นแผนภาพสำหรับอธิบายเหตุการณ์และผลกระทบที่เกิดขึ้นข้อมูลในแต่ละเอนติตี เพราะ Entity-Event Modeling เป็นโมเดลที่รวมระหว่างกระบวนการและข้อมูลที่ต้องการไว้ในที่เดียวกัน ใน Entity-Event Modeling ใช้เทคนิคดังนี้ คือ วิเคราะห์เอนติตีไลฟ์ฮีสทรีไดอะแกรม (Entity Life History – ELH) กับสร้าง Effect Correspondence Diagram

เอนติตีไลฟ์ฮีสทรีไดอะแกรม คือ การนำเอนติตีต่าง ๆ ที่มีผลกระทบจากเหตุการณ์มาสร้างไดอะแกรมซึ่งภายในไดอะแกรอนุญาตให้เหตุการณ์ที่เข้ามากระทบกับเอนติตีสามารถเกิดขึ้นได้หลายรูปแบบ เช่น การทำซ้ำ (Iteration) การเลือก (Option) และ การทำงานแบบเรียงลำดับ (Sequence) เป็นต้น ซึ่งรูปแบบของเหตุการณ์ที่เกิดขึ้นกับเอนติตีนั้นขึ้นกับกฎเกณฑ์ทางธุรกิจของระบบ บางครั้งการวิเคราะห์กระบวนการที่ผ่านมานั้นบางกระบวนการอาจถูกละเลย มีความคลุมเคลือหรือความไม่ชัดเจนในการทำงานที่เกิดขึ้นได้ ดังนั้นการนำเอนติตีไลฟ์ฮีสทรีทำให้การวิเคราะห์จะทำให้กระบวนการต่าง ๆ ที่เกิดขึ้นกับระบบมีความสมบูรณ์มากขึ้น ซึ่งเอนติตีไลฟ์ฮีสทรีไดอะแรมเป็นรายละเอียดพื้นฐานสำหรับการออกแบบกระบวนการภายในระบบของขั้นตอนถัดไป

Effect Correspondence Diagram เป็นขั้นตอนที่ทำให้โครงงานในส่วนของการออกแบบนั้นมีความชัดเจนมากขึ้น โดยจะพิจารณาจาก LDM ที่ถูกสร้างจากขั้นตอนก่อนหน้า ดังนั้นการสร้าง ECD พิจารณาจากเหตุการณ์ที่เข้ามากระทบเอนติตีแต่ละตัวที่อยู่ใน LDM ทั้งหมดของระบบ ECD ที่ถูกสร้างขึ้นจะเป็นรายละเอียดพื้นฐานเพื่อนำไปปรุงกระบวนการของฟังก์ชันและสามารถนำไปใช้ในระยะที่ 5 ต่อได้

ใน Entity-Event Modelling นั้นจะพิจารณาแต่เพียงกระบวนการที่มีเหตุการณ์เข้ามากระทบเท่านั้น ซึ่งรายละเอียดต่าง ๆ ที่แสดงถึงการสืบค้นข้อมูลที่เกิดขึ้นตามเหตุการณ์ที่เข้ามากระทบนั้นยังขาดหายไป ดังนั้นขั้นตอนที่ 360 จึงได้แสดงรายลายละเอียดเหล่านี้ไว้ในส่วนของ Enquiry Access Path (EAPs) แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถูกต้องและรายละเอียดที่เกิดขึ้นของขั้นตอนนี้ทำให้ความต้องการของระบบทั้ง LDM, DFM และการกำหนดฟังก์ชันของระบบสามารถขยายรายละเอียดและมีความสมบูรณ์มากขึ้น ซึ่งต่างจากขั้นตอนที่ 330 ที่แสดงเพียงอินพุต-เอาต์พุตให้กับฟังก์ชั่นเท่านั้น โดยไม่แสดงถึงเหตุการณ์ที่เข้ามากระทบกับฟังก์ชั่นรวมถึงข้อมูลต่างที่เกิดขึ้นจากผลกระทบที่เกิดขึ้น จึงเป็นเหตุให้ไดแกรมต่าง ๆ ของขั้นตอนที่ 330 นั้นยังไม่สมบูรณ์ ดังนั้นขั้นตอนที่ 360 จึงเป็นขั้นตอนที่แสดงถึงสิ่งที่ขาดหายไปของขั้นตอนที่ 330 การสร้าง ELH, ECD และ EAP ของขั้นตอนที่ 360 มีการทำงานย่อย ๆ ดังนี้

งาน 10 กำหนดเหตุการณ์ที่มีผลต่อเอนติตีโดยใช้ Event/Entity Matrix

งาน 20 สร้างเอนติตีไลฟ์ฮีสทรีไดอะแกรมจาก LDS ถ้าสร้างจากรายละเอียดของเอนติตี (Detail Entity)ให้สร้างจากล่างขึ้นบนพร้อมเอกสาร ถ้าสร้างจากเอนติตีหลัก (Master Entity) ให้สร้างจากบนลงล่างพร้อมกับระบุเงื่อนไข กรณีพิเศษและความสัมพันธ์ระหว่างกัน จากนั้นควรระบุโอเปอร์เรชันให้กับเอนติตีไลฟ์ฮีสทรีทั้งสองกรณี

งาน 30 สร้าง ECD ของแต่ละเหตุการณ์ที่เกิดขึ้นระหว่างที่วิเคราะห์เอนติตีไลฟ์ฮีสทรีไดอะแกรม

งาน 40 ปรับปรุง Requirement Catalogue และ LDM ให้ตรงกับความต้องการใหม่ ๆ ระหว่างกำหนด entity-event Model จากนั้นกำหนดฟังก์ชันให้กับเหตุการณ์ที่เกิดขึ้นใหม่ทุกเหตุการณ์ที่เกี่ยวข้อง (ซึ่งจะต้องย้อนกลับไปทำขั้นตอนที่ 330)

งาน 50 สร้าง Enquiry Access Path สำหรับแต่ละ Function

งาน 60 ระบุปริมาณของข้อมูลที่ต้องการบรรจุลงในแต่ละเอนติตีให้กับ LDM

ขั้นตอนที่ 370 ยืนยันจุดมุ่งหมายกับผู้ใช้ (Confirm System Objective)

ตรวจสอบรายละเอียดของฟังก์ชันและ Non-Function ที่ออกแบบไว้ให้เรียบร้อยก่อนทำขั้นตอนต่อไป ซึ่งมีงานย่อยดังนี้

งาน 10 ตรวจสอบ Function Requirement และ Non-Function Requirements ให้ตรงกับ Requirement Catalogue และ Cross-Reference

งาน 20 กำหนด Non-Function ให้กับสิ่งที่เกิดขึ้นใหม่หรือทำความเข้าใจกับสิ่งที่เกิดขึ้นใน Requirement Catalogue , Function Definition หรือ Required System LDM

งาน 30 ตรวจสอบแต่ละฟังก์ชันที่สร้างขึ้นให้ถูกต้อง

งาน 40 รับรอง Required System LDM ที่สร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 380 รวบรวมเอกสาร

ขั้นตอนที่ 380 เป็นขั้นตอนสุดท้ายของระยะที่ 3 ซึ่งจะรวบรวมเอกสารและแผนภูมิของแต่ละขั้นตอนรวมทั้งตรวจสอบความถูกต้องทั้งหมดก่อนข้ามไปปฏิบัติงานระยะต่อไป

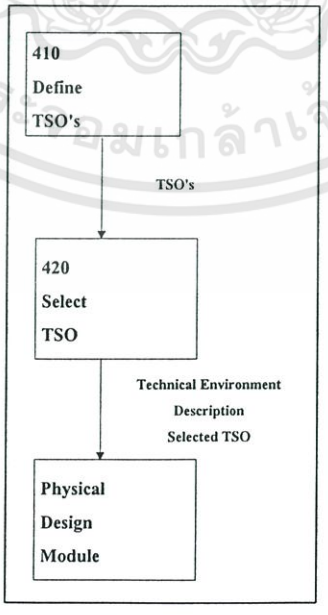
ผลลัพธ์ของระยะที่ 3 คือ

1. Requirement System LDM
2. Function Definitions
3. I/O Structure
4. ELHs (Entity Life History)
5. ECD (Effect Correspondences Diagram)
6. EAP (Enquiry Access Paths)
7. Requirements Catalogue
8. User Role/Function Matrix

ระยะที่ 4 กำหนดทางเลือกทางเทคนิค (Stage 4 : Technical System Options)

ระยะที่ 4 คือ เลือกสภาวะแวดล้อมของระบบ เช่น อุปกรณ์ทางด้านฮาร์ดแวร์ ซอฟต์แวร์ ภาษาที่ใช้สำหรับการพัฒนา วิธีการพัฒนา (ผลิตเองหรือซื้อ) เป็นต้น ดังนั้นผู้พัฒนาควรเสนอทางเลือกให้กับผู้ใช้ 6 ทางเลือกเพื่อตัดสินใจหาทางเลือกที่ดีที่สุดได้ มีขั้นตอนการทำงานดังนี้ (แสดงในรูปที่ 2.7)

ระยะที่ 4 : กำหนดทางเลือกทางเทคนิค



รูปที่ 2.7 แสดงขั้นตอนการทำงานของระยะที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 410 กำหนดทางเลือกของการสร้างระบบ (TSOs)

งาน 10 กำหนดข้อจำกัดในทางด้านเทคนิค

งาน 20 กำหนดทางเลือกทางด้านเทคนิคไว้อย่างน้อย 6 ทางเลือก

งาน 30 ให้ผู้ใช้พิจารณาเลือกทางเลือกอย่างน้อย 2-3 ทางเลือก

งาน 40 เพิ่มรายละเอียดในแต่ละทางเลือกที่ผู้ใช้ได้เลือกไว้แล้ว พร้อมกับกำหนดรายละเอียดของสภาวะแวดล้อมของเทคนิค (TED) และรายละเอียดของระบบ

งาน 50 ตรวจสอบระดับการให้บริการและความสามารถของข้อมูลที่ใช้

งาน 60 ตรวจสอบสภาพแวดล้อมของเทคนิคและวิเคราะห์ผลกระทบที่มีต่อระบบและโครงร่างของแผนงานรวมทั้งวิเคราะห์ต้นทุนและค่าใช้จ่ายในเทคนิคนั้น ๆ

ขั้นตอนที่ 420 ผู้ใช้เลือกวิธีการพัฒนาระบบจากทางเลือกทางด้านเทคนิคที่เหมาะสมและควรให้มีความสอดคล้องกับทางเลือกในระยะที่ 2

ผลลัพธ์ของระยะที่ 4 คือ

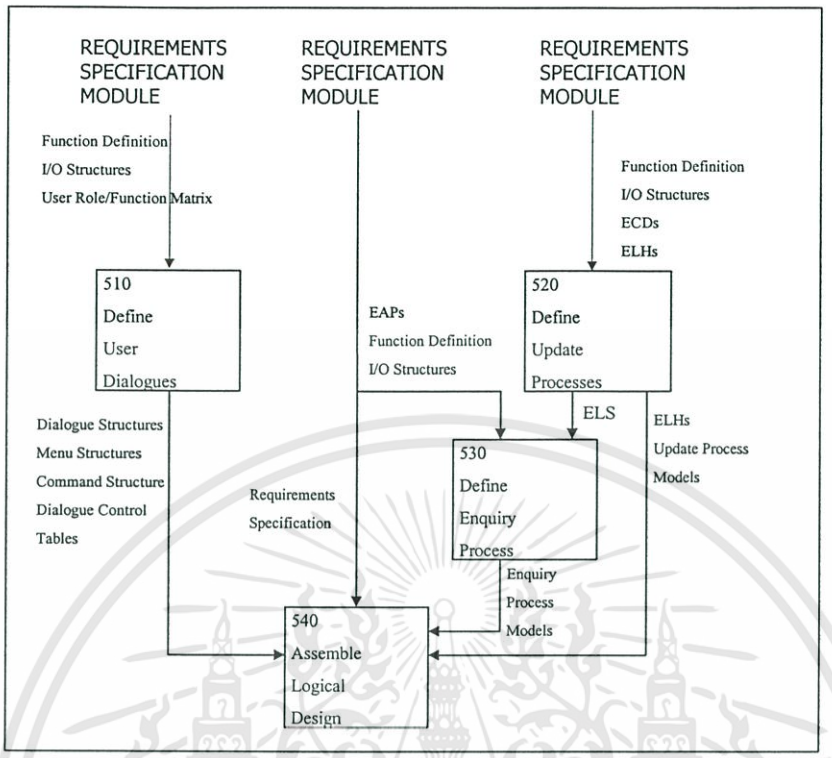
1. นำเสนอข้อสรุปของการสร้างระบบ เช่น ฮาร์ดแวร์ ซอฟต์แวร์ ภาษาที่ใช้พัฒนา แนวทางของการพัฒนาระบบ (ผลิตเองหรือจ้างบริษัทภายนอก) เป็นต้น
2. เลือกข้อสรุปและรายละเอียดต่าง ๆ ไปใช้ในขั้นตอนที่ 6

ระยะที่ 5 การออกแบบทางตรรกะ (Stage 5 : Logical Design)

นำผลลัพธ์ของระยะที่ 3 มาเพิ่มเติมรายละเอียดให้มากขึ้น เช่น นำเส้นทางของการเข้าสู่ระบบมาจัดกลุ่มให้เหมาะสม (User Dialogue) ปรับปรุง ELHs และ ECDs ให้เหมาะสมกับผลกระทบกับเหตุการณ์ที่เกิดขึ้นเพื่อนำไปพัฒนาต่อในส่วนของการออกแบบเชิงกายภาพ ปรับปรุง EAP ให้สอดคล้องกับความเป็นจริง เป็นต้น โดยมีขั้นตอนการทำงานดังนี้ (แสดงในรูปที่ 2.8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะที่ 5 : การออกแบบทางตรรกะ



รูปที่ 2.8 แสดงขั้นตอนการทำงานของระยะที่ 5

ขั้นตอนที่ 510 กำหนดส่วนติดต่อกับผู้ใช้ (Define User Dialogues)

การกำหนดส่วนติดต่อกับผู้ใช้ คือ การกำหนดการติดต่อระหว่างฟังก์ชันกับผู้ใช้ให้เป็นหมวดหมู่ ปกติฟังก์ชันที่เกิดขึ้นจากระยะที่ 3 บางครั้งการทำงานอาจไม่สอดคล้องหรือไม่สามารถแสดงได้ชันเจนกับระบบได้ โดยเฉพาะอินพุตและเอาต์พุตที่เกิดขึ้นในระบบ การกำหนดส่วนติดต่อกับผู้ใช้นั้นเป็นอีกวิธีหนึ่งที่แสดงถึงความสอดคล้องของฟังก์ชันกับผู้ใช้ได้เป็นอย่างดี ซึ่งการกำหนดส่วนที่ติดต่อกับผู้ใช้เปรียบเสมือนกับ "Front-end" ของระบบและแต่ละส่วนการติดต่อนั้นยังต้องสอดคล้องกับกฎเกณฑ์ต่าง ๆ (User Role/ Function Matrix) ที่เคยถูกสร้างขึ้นจากระยะที่ 3 หลังจากนั้นผู้วิเคราะห์ต้องนำส่วนดังกล่าวมาออกแบบเพื่อให้ได้ลำดับรายการ (Menu) ของระบบที่เหมือนกันอยู่ในกลุ่มเดียวกันและสุดท้ายการทำงานของขั้นตอนนี้ระบบจะได้กลุ่มของฟังก์ชั่นที่สัมพันธ์กับส่วนของการติดต่อที่สามารถนำไปสร้างกระบวนการต่อไปในขั้นตอนที่ 520 ต่อไป ซึ่งมีงานย่อย ๆ ดังนี้

งาน 10 เปลี่ยนโครงสร้างอินพุต-เอาต์พุต (I/O Structure) ให้เป็น Dialogue Structure สำหรับทุก ๆ Online Function

งาน 20 แสดงรายละเอียดของแต่ละ Dialogue ลงในตารางของ Dialogue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน 30 สร้างของเมนู (Menu Structure) ในแต่ละ User Role และทำเอกสารประกอบในแต่ละ Dialogue

งาน 40 สร้างสิ่งที่จะช่วยให้เข้าถึง Dialogue นั้น ๆ

ขั้นตอนที่ 520 กำหนดกระบวนการที่ทำหน้าที่เปลี่ยนแปลง (Define Update Processes)

จากขั้นตอนที่ 510 ที่กล่าวถึงส่วนติดต่อกับผู้ใช้โดยแสดงในรูปของ User Dialogue แต่ในขั้นตอนนี้ คือ การแสดงรายละเอียดภายในของกระบวนการในระบบ โดยนำเอนติตีไลฟ์ฮีสทรีไดอะแกรมและ ECD เพื่อมากำหนดความต้องการตามเหตุการณ์ที่เกิดขึ้นในระบบกับโมเดลของข้อมูลทางตรรกะ (Logical Data Model) ซึ่งผู้ออกแบบต้องนำทั้ง 2 ส่วนมาจัดกลุ่มรวมเพื่อให้เป็นรายละเอียดของกระบวนการที่ติดต่อกับผู้ใช้ หรือเรียกว่า UPM (Update Process Model) ซึ่งมีงานย่อย ๆ ดังนี้

งาน 10 เพิ่มลำดับขั้นตอนที่การทำงานในเอนติตีไลฟ์ฮีสทรีไดอะแกรม
(ทำซ้ำงานที่ 20-50 จนกว่าเหตุการณ์นั้นจะสิ้นสุด)

งาน 20 สร้างและปรับปรุงโครงสร้างของ กระบวนการในแต่ละเหตุการณ์ รวมทั้งใช้ ECD เข้ามาช่วยสร้างเอาต์พุตในโครงสร้างอินพุต-เอาต์พุต

งาน 30 แสดงขั้นตอนการทำงานของแต่ละเหตุการณ์

งาน 40 กำหนดโครงสร้างของกระบวนการปฏิบัติงานของแต่ละเหตุการณ์รวมทั้งกำหนดเงื่อนไขการเลือก และ การทำซ้ำของโครงสร้างที่สร้างขึ้น

งาน 50 อธิบายความผิดพลาดที่เกิดขึ้นกับข้อมูล

ขั้นตอนที่ 530 ปรับปรุงส่วนของโครงสร้างข้อมูลเข้าและออก

ขั้นตอนนี้คือการนำ UPM ที่เกิดจากขั้นตอนที่ 520 มากำหนดการเข้าถึงฐานข้อมูลโดยใช้เทคนิค LDPD (Logical Database Process Design) เข้ามาช่วย ซึ่ง LDPD นี้นำ EAPs ที่เกิดจากขั้นตอนที่ 360 มาใช้ประกอบสำหรับการสร้างโครงสร้างของกระบวนการ (Process Structure) กับโครงสร้างข้อมูลเข้าและออก เพื่อให้ได้ EPMs (Enquiry Process Model) และมีงานย่อย ๆ ดังนี้

(ทำซ้ำในงานที่ 10-50 ของแต่ละ Enquiry จนกว่าจะหมด)

งาน 10 แยกโครงสร้างของกระบวนการที่เป็นอินพุตไปสู่ EAP

งาน 20 แยกโครงสร้างของกระบวนการที่เป็นเอาต์พุตออกจากโครงสร้างอินพุต-เอาต์พุต

งาน 30 รวมโครงสร้างทั้งสองเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน 40 แสดงการปฏิบัติงานที่เกิดกับแต่ละ enquiry process รวมทั้งกำหนด เงื่อนไขการเลือก และ การทำซ้ำของโครงสร้างที่สร้างขึ้น

งาน 50 แสดงข้อผิดพลาดที่เกิดขึ้นกับเอาต์พุตที่มักเกิดขึ้นกับโครงสร้าง

ขั้นตอนที่ 540 รวมงานของการออกแบบทางตรรกะ (Assemble Logical Design)

ขั้นตอนที่ 540 เป็นขั้นตอนสุดท้ายของระยะที่ 5 ซึ่งจะรวบรมเอกสารและแผนภูมิของแต่ละขั้นตอนรวมทั้งตรวจสอบความถูกต้องทั้งหมดก่อนข้ามไปปฏิบัติงานระยะต่อไป

ผลลัพธ์ของระยะที่ 5 คือ

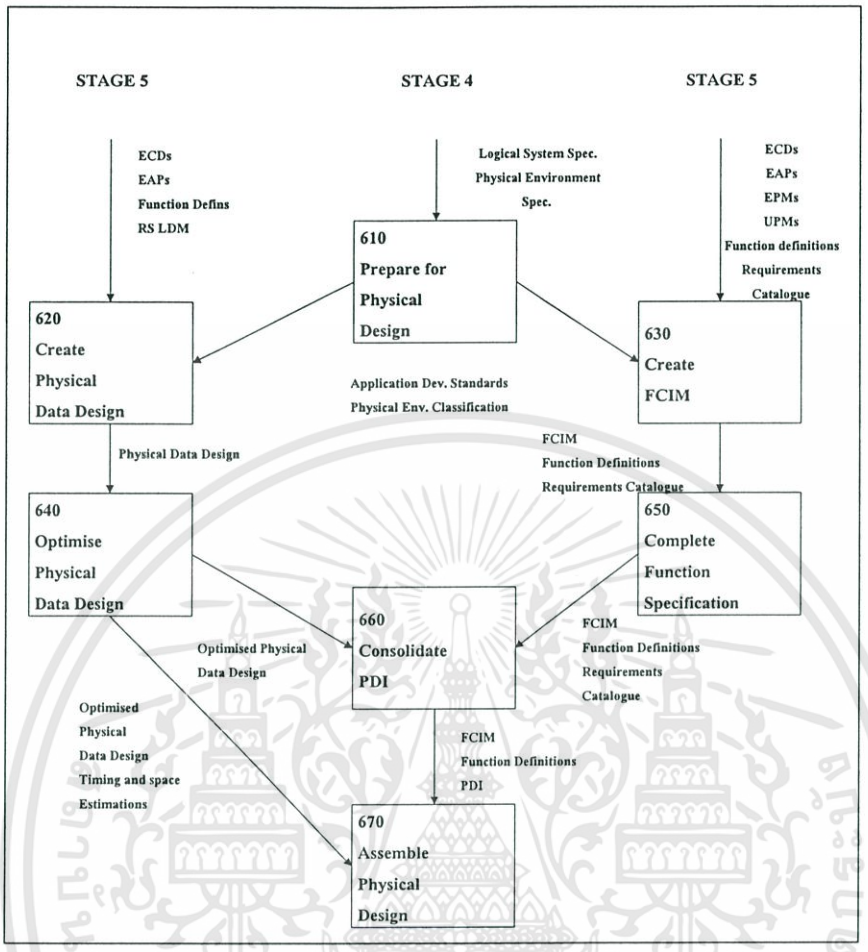
1. โครงสร้างของ Dialogue
2. ตารางควบคุม Dialogue
3. โครงสร้างของรายการ (Menu Structure)
4. โครงสร้างคำสั่งต่าง ๆ (Command Structure)
5. EPM (Enquiry Process Models)
6. UPM (Update Process Models)

<u>ระยะที่ 6 การออกแบบทางกายภาพ (Stage 6 : Physical Design)</u>

การออกแบบทางกายภาพ คือ การย้ายจากการออกแบบเชิงตรรกะไปสู่การออกแบบเชิงกายภาพที่นำไปสู่ระบบที่สามารถนำไปใช้งานได้ โดยมีขั้นตอนการทำงานดังนี้ (แสดงในรูปที่ 2.9)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะที่ 6 : การออกแบบทางกายภาพ



รูปที่ 2.9 แสดงขั้นตอนการทำงานของระยะที่ 6

ขั้นตอนที่ 610 เตรียมการย้ายไปสู่การออกแบบทางกายภาพ (Physical Design)

การย้ายจากการออกแบบทางตรรกะไปสู่การออกแบบทางกายภาพนั้น ก่อนการย้ายผู้วิเคราะห์ระบบต้องตรวจสอบเอกสารและแผนภูมิที่เกิดขึ้นจากระยะต่าง ๆ ที่ผ่านมา ซึ่งเอกสารและแผนภูมิของทุกระยะสามารถแสดงถึงขั้นตอนการทำงานและความต้องการของระบบได้เป็นอย่างดี การย้ายจากตรรกะไปสู่กายภาพนั้น สิ่งที่ต้องคำนึงถึงคือ การย้ายฐานข้อมูลเชิงตรรกะไปสู่กายภาพ การคำนวณหาเนื้อในการเก็บฐานข้อมูล การทำงานของฟังก์ชันและการตั้งชื่อสำหรับฟังก์ชันในระบบ สิ่งเหล่านี้เป็นข้อกำหนดก่อนที่จะเริ่มย้ายในขั้นตอนที่ 620 ซึ่งก่อนที่จะข้ามไปขั้นตอนดังกล่าวนั้น ผู้วิเคราะห์ควรทำงานย่อยในขั้นตอนที่ 610 ดังต่อไปนี้

งาน 10 เลือกสภาพแวดล้อมที่จะออกแบบทางกายภาพ (Physical Environment) เช่น ระบบการทำงาน ความสามารถในการปรับแต่งฐานข้อมูลและคุณสมบัติของฐานข้อมูล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน 20 คำนวณเนื้อที่ของฐานข้อมูลที่ออกแบบพร้อมกับประมาณเวลาสำหรับการเรียกใช้แต่ละครั้ง

งาน 30 กำหนดมาตรฐานของสภาพแวดล้อมทางกายภาพของกระบวนการและข้อมูลให้กับผู้ใช้

งาน 40 กำหนดกฎในการออกแบบของข้อมูล

งาน 50 กำหนดมาตรฐานสำหรับการตั้งชื่อให้กับโปรแกรม ฐานข้อมูล เรคอร์ด และอื่น ๆ

งาน 60 วางแผนในการออกแบบทางกายภาพและกำหนดเกณฑ์ต่าง ๆ ของระบบ ซึ่งขั้นตอนของ SSADM นี้จะขึ้นอยู่กับสภาพแวดล้อมทางกายภาพเป็นส่วนใหญ่

งาน 70 จัดเตรียมการฝึกสอนและทำคู่มือการใช้งานสำหรับผู้ใช้

งาน 80 ตกลงขั้นตอนการออกแบบทางกายภาพกับผู้บริหาร

ขั้นตอนที่ 620 การออกแบบฐานข้อมูลทางกายภาพ (Create Physical Data Design)

แผนภูมิข้อมูล LDM ที่เกิดขึ้นจากขั้นตอนก่อนหน้านั้น เป็นเพียงแผนภูมิที่แสดงถึงความต้องการฐานข้อมูลและนโยบายทางธุรกิจเท่านั้น ซึ่งไม่ได้กล่าวถึงระบบฐานข้อมูลทางกายภาพและการจัดเก็บข้อมูลทางกายภาพ ดังนั้นขั้นตอนที่ 620 จึงเป็นขั้นตอนที่แสดงถึงการเปลี่ยนฐานข้อมูลทางตรรกะไปสู่การออกแบบฐานข้อมูลทางกายภาพ การแปลฐานข้อมูลไปสู่ระดับกายภาพนั้น ในขั้นตอนนี้แบ่งออกเป็น 2 ลักษณะ กล่าวคือ ผู้วิเคราะห์ระบบต้องนำแผนภูมิข้อมูลเดิมที่มีอยู่นั้นมาจัดทำกระบวนการซึ่งเรียกว่า “First-Cut Data Design” เป็นกระบวนการสำหรับการแปลงแผนภูมิทางตรรกะไปสู่กายภาพเพื่อให้ฐานข้อมูลที่ออกแบบไว้นั้นถูกต้องและสามารถนำไปใช้งานจริงได้ จากนั้นนำฐานข้อมูลที่ถูกแปลงมาคำนวณขนาดของฐานข้อมูลต้องบันทึกลงในระบบ พร้อมกับกำหนดวิธีการเข้าถึงฐานข้อมูลและวางแผนสำหรับการบำรุงรักษาฐานข้อมูลเมื่อเกิดข้อผิดพลาด ซึ่งในขั้นตอนที่ 620 มีลำดับการทำงานดังนี้

งาน 10 แปลง LDM ไปสู่รูปแบบของฐานข้อมูลทางกายภาพ (Physical Data Model)

งาน 20 กำหนดการเข้าถึงฐานข้อมูลทางกายภาพของแต่ละเอนติตี

งาน 30 กำหนด Root ของ Physical Hierarchies

งาน 40 จับกลุ่มของข้อมูลที่ไม่ขึ้นอยู่กับ Root

งาน 50 ประยุกต์ข้อมูลให้ขึ้นไปอยู่ตามกฎเกณฑ์ของข้อมูลในกรณีที่เป็นอิสระต่อกัน

งาน 60 กำหนดขนาดของข้อมูล

งาน 70 คำนวณขนาดข้อมูลแต่ละเอนติตีไปสู่ขนาดของข้อมูลที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน 80 ประยุกต์กลุ่มของข้อมูลโดยทำ First-cut เพื่อให้เป็นไปตามกฎเกณฑ์ของการออกแบบ

ขั้นตอนที่ 630 สร้างองค์ประกอบฟังก์ชันสำหรับการอิมพลีเมนต์ (Create Function component Implementation Map)

การสร้างองค์ประกอบของฟังก์ชันในขั้นตอนที่ 630 นี้ผู้ออกแบบต้องนำ UPM, EAP, ECD และ EPM มาประกอบเพื่อจัดทำ FCIM (Function Complete Implementation Map) พร้อมกับแสดงรายละเอียดแต่ตารางของแต่ละฟังก์ชันเพื่อให้ทราบถึงความสัมพันธ์ที่เกิดขึ้น รายละเอียดที่แสดงนั้นจะแบ่งออกเป็น 2 ส่วน ส่วนแรกเรียกว่า "PART A" แสดงถึงการแปลงจากตรรกะไปสู่กายภาพที่กล่าวถึงซุปเปอร์ฟังก์ชันที่ครอบคุมทั้ง Procedural, Non-Procedural อินพุต-เอาต์พุตและรายละเอียดย่อย ๆ ที่เกี่ยวข้อง ส่วนสุดท้ายเรียกว่า "PART B" แสดงถึงย้อนกลับจากกายภาพไปสู่ตรรกะเพื่อแสดงให้ทราบถึงความถูกต้องของการสร้าง FCIM โดยภายในประกอบด้วย Procedure, Non-Procedural, Screen, input-output และอื่น ๆ ที่เกี่ยวข้อง ทั้ง 2 ส่วนที่กล่าวมานั้นในแต่ละส่วนยังประกอบด้วยข้อความต่าง ๆ ของระบบที่จำเป็นต้องแสดงให้ผู้ใช้ระบบทราบ เช่น ข้อความแสดงข้อผิดพลาดของโปรแกรม เครื่องมือช่วยเหลือ และ ข้อความอื่น ๆ เป็นต้น ซึ่งหลังจากสร้าง FCIM เสร็จสิ้น FCIM เหล่านี้จะถูกเรียกว่า UFM (Universal Function Model) ที่แสดงรายละเอียดต่าง ๆ ของฟังก์ชันในระบบได้ครบถ้วนและสามารถนำไปปฏิบัติกับขั้นตอนต่อไปได้ การสร้าง FCIM และ UFM นั้นมีลำดับการทำงานดังต่อไปนี้

งาน 10 เปรียบเทียบฟังก์ชันทั้งหมดอีกครั้งก่อนทำการออกแบบกระบวนการ

งาน 20 เปรียบเทียบฟังก์ชันทั้งหมดก่อนที่จะกำหนดว่าสามารถจะให้ร่วมกันได้มากน้อยเพียงใด

(ทำซ้ำงานที่ 30-80 ของแต่ละฟังก์ชันจนกว่าจะครอบคลุมทุกฟังก์ชันที่ต้องการ)

งาน 30 อธิบายรายละเอียดของแต่ละฟังก์ชัน

งาน 40 อธิบายรายละเอียดเกี่ยวกับรูปแบบและขั้นตอนในการทำงาน

งาน 50 อธิบายรายละเอียดของวิธีการควบคุมและการตรวจสอบข้อผิดพลาด

งาน 60 อธิบายรายละเอียดของรูปแบบในการติดต่อกับระบบทั้งหมด เช่น ระบบรายงานและข้อมูล เป็นต้น

งาน 70 ออกแบบส่วนประกอบของ Dialogue ในแต่ละอัน รวมทั้งองค์ประกอบของรายงาน หน้าจอและเมนูการใช้งาน เป็นต้น

งาน 80 นำเสนอองค์ประกอบของ Non-Procedural Function ที่ใช้กับ Physical

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Environment ในระบบให้มีความสะดวกมากขึ้น

ขั้นตอนที่ 640 ออกแบบข้อมูลทางกายภาพให้เหมาะสม (Optimize Physical Data Design)

จากขั้นตอนที่ 620 ที่ทำ first-cut กับกำหนดขนาดของฐานข้อมูลไปสู่กายภาพนั้น ในขั้นตอนนี้ผู้ออกแบบต้องนำสิ่งแผนภูมิที่ถูกแปลงมาทำการตรวจสอบอีกครั้ง เพื่อให้มั่นใจว่าสิ่งที่ทำมานั้นถูกต้องตรงตามความต้องการเพราะหลังจากขั้นตอนนี้แล้วจะต้องถูกส่งไปให้โปรแกรมเมอร์สร้างฐานข้อมูลจริงให้กับระบบซึ่งถ้าแผนภูมิมีความผิดพลาดย่อมทำให้ข้อมูลที่ถูกเก็บในระบบผิดพลาดตามไปด้วย ดังนั้นผู้ออแบบจึงต้องปฏิบัติดังนี้ กำหนดวัตถุประสงค์ของแต่ละฐานข้อมูลคำถึงขนาดและเวลาที่เสียไปในการเข้าถึงฐานข้อมูลรวมทั้งการรักษาความปลอดภัยให้กับข้อมูล สุดท้ายตรวจสอบโครงสร้างของแผนภูมิทั้งหมดก่อนสร้างฐานข้อมูลไปสู่ฐานข้อมูลจริง ซึ่งการออกแบบฐานข้อมูลให้กับระบบจากระยะที่ 3 นั้นมีข้อกำหนดหลายประการ เช่น ความสัมพันธ์นั้นต้องเป็นแบบ 1:1 , 1:M หรือ M:N เป็นต้น แต่ในขั้นตอนนี้ให้ต้องเปลี่ยนความสัมพันธ์ที่มีอยู่ให้เป็นแบบ 1:1 ทั้งแผนภูมิเพราะในระยะที่ 3 ผู้วิเคราะห์และออกแบบมองระบบทางตรรกะเท่านั้น เมื่อมาถึงขั้นตอนการแปลงไปสู่กายภาพนั้นควรทำให้ความสัมพันธ์นั้นเป็นแบบ 1:1 ให้มากที่สุดเท่าที่นทำได้ เพราะการออกแบบทางกายภายมองที่การสร้างฐานข้อมูลลงบนสื่อมากกว่าแนวคิด การทำลักษณะเช่นนี้เพราะต้องการฐานข้อมูลที่เกิดขึ้นจริงตรงกับความต้องการของผู้ใช้มากที่สุด ซึ่งการทำงานของขั้นตอนนี้มีดังนี้

งาน 10 ตรวจสอบ LDM ที่ออกแบบในส่วนของกายภาพที่มีลักษณะเป็นแบบ 1:1 ให้กระชับมากขึ้นในส่วนของเนื้อที่

งาน 20 ตรวจสอบ LDM ที่ออกแบบในส่วนของกายภาพที่มีลักษณะเป็นแบบ 1:1 ให้กระชับมากขึ้นในส่วนของจำนวนครั้งการใช้งาน

ขั้นตอนที่ 650 Complete Function Specification

ขั้นตอนที่ 650 คือ การตรวจสอบฟังก์ชันที่ผู้ออกแบบย้ายจากการออกแบบทางตรรกะไปสู่กายภาพ ซึ่งเมื่อปฏิบัติถึงขั้นตอนนี้ฟังก์ชันและข้อมูลต่าง ๆ ที่เกิดขึ้นนั้นมีความสมบูรณ์และพร้อมที่จะนำไปสู่ขั้นตอนการอิมพลีเมนต์ได้ ฟังก์ชันที่เกิดขึ้นตั้งแต่ระยะที่ 5 นั้นเป็นจุดเริ่มต้นที่แสดงถึงการทำงานของกระบวนการ อินพุต-เอาต์พุตที่เกี่ยวข้อง เมื่อปฏิบัติเช่นนี้ต่อเนื่องมาทำให้ฟังก์ชันนั้นเปลี่ยนรูปแบบไปสู่กระบวนการของฟังก์ชันที่เรียกว่า UFM ส่วนของข้อมูลที่เกิดขึ้นจากระยะที่ 3 เช่น LDS และ EAP ที่แสดงถึงความสัมพันธ์ของฐานข้อมูลในระบบจะถูกแปลงไปสู่ฐานข้อมูลทางกายภาพพร้อมกับคำนวณหาพื้นที่ที่ต้องใช้ในระบบ กล่าวได้ว่าขั้นตอนนี้เป็นที่คอยตรวจสอบก่อนนำไปสู่การอิมพลีเมนต์จริงและนำไปสู่การทำงานในขั้นตอนถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 660 Consolidate Process Data Interface- PDI

ตั้งแต่ขั้นตอนที่ 610 ถึง 650 ที่ผู้พัฒนาได้แปลงการออกแบบทางตรรกะไปสู่ระดับกายภาพนั้น การแปลงรูปแบบของฟังก์ชันและฐานข้อมูลให้อยู่ในรูปของ UFM FCIM และเปลี่ยนแปลงข้อมูลทางตรรกะไปสู่ระดับกายภาพนั้น ในขั้นตอนที่ 660 นี้สามารถสร้างกระบวนการติดต่อระหว่างกระบวนการทางตรรกะกับกระบวนการทางกายภาพ กล่าวคือ การออกแบบทางตรรกะที่เริ่มต้นพัฒนาตั้งแต่เริ่มอนุมัติโครงการจะถูกพัฒนาอย่างต่อเนื่อง ซึ่งทำให้เกิดแผนผังต่าง ๆ เช่น อินพุต-เอาต์พุต กระบวนการ และข้อมูล เป็นต้น สิ่งเหล่านี้เป็นเพียงแนวคิดทางตรรกะของระบบส่วนการออกแบบทางกายภาพ คือ การนำสิ่งที่เกิดขึ้นทางตรรกะมานำเสนอให้เป็นรูปธรรมมากขึ้น การออกแบบทั้งสองส่วนมีความเกี่ยวข้องกันก็ตามแต่ในทางปฏิบัติงานนั้นจำเป็นต้องมีกระบวนการคอยเชื่อมโยงความสัมพันธ์กัน ซึ่งเรียกว่า “Process Data Interface(PDI)”

PDI คือ กระบวนการกลางที่คอยเชื่อมโยงการเข้าถึงข้อมูลทางกายภาพกับกระบวนการทางตรรกะที่ได้ออกแบบไว้และคอยอำนวยความสะดวก เช่น คำสั่ง SQL หรือ แบตช์ไฟล์ เป็นต้น การออกแบบ PDI ควรออกแบบให้มีความยืดหยุ่นสำหรับการนำไปใช้งานและปรับปรุงภายหลังได้ง่าย เนื่องจาก PDI เป็นกระบวนการที่ทำหน้าที่ที่คอยติดต่อกับฐานข้อมูลทางกายภาย ถ้า PDI ที่ออกแบบนั้นไม่มีความยืดหยุ่นเพียงพอย่อมมีผลเมื่อมีการเปลี่ยนแปลงการทำงานของกระบวนการ เช่น การปรับปรุง แก้ไขหรือเพิ่มเติม เป็นต้น การเปลี่ยนแปลงการทำงานของกระบวนการมักเกิดจากความต้องการของผู้ใช้

ดังนั้นในขั้นตอนที่ 660 จึงเป็นขั้นตอนที่คอยเชื่อมโยงระหว่างกระบวนการกับข้อมูลทางกายภาพของระบบ ซึ่งมีการทำงานดังต่อไปนี้

งาน 10 กำหนดสิ่งที่ไม่เข้ากันระหว่าง Logical กับ Physical ในส่วนของการใช้ database

งาน 20 กำหนดคีย์ทั้งหมดของข้อมูลที่ไม่เข้ากัน

งาน 30 อธิบายรายละเอียดของวิธีใช้ในส่วนต่าง ๆ และบอกถึงผลลัพธ์ที่ทำได้และทำไม่ได้

งาน 40 ตรวจสอบความถูกต้องของ PDI อีกครั้ง

งาน 50 จัดทำเอกสารประกอบของ PDI Process ในส่วนของ FCIM

งาน 60 ปรับปรุง Requirement Catalogue ใน Data Requirement ให้สอดคล้องกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 670 รวบรวมเอกสาร

ขั้นตอนที่ 670 เป็นขั้นตอนสุดท้ายของระยะที่ 6 ซึ่งจะรวบรมเอกสารและแผนภูมิของแต่ละขั้นตอนรวมทั้งตรวจสอบความถูกต้องทั้งหมดก่อนนำรายละเอียดต่าง ๆ ส่งให้กับโปรแกรมเมอร์เพื่อเขียนโปรแกรมให้กับระบบใหม่

ผลลัพธ์ของระยะที่ 6 คือ

1. FCIM (Function Component Implementation Map)
2. Function Definition
3. Optimized Physical Data Design
4. Required System LDM
5. Requirements Catalogue
6. PDI Specification
7. Space and Timing Estimations.

2.4 สรุป

การพัฒนาระบบวิธีเชิงโครงสร้างในปัจจุบันมีอยู่ด้วยกันหลายวิธี ซึ่งแต่ละวิธีจะมีขั้นตอนและวิธีการพัฒนาแตกต่างกันออกไป ซึ่งขึ้นอยู่กับนักวิเคราะห์และนักพัฒนาระบบจะเลือกใช้ วิธี SSADM นี้เป็นอีกวิธีหนึ่งที่มีความละเอียดทุกขั้นตอนและสามารถติดตามได้และมีเอกสารประกอบอย่างชัดเจน ซึ่งเอกสารเหล่านี้สามารถที่จะลดหรือเพิ่มได้ตามความจำเป็น การพัฒนาระบบสารสนเทศวิธีของ SSADM แบ่งออกเป็น 7 ระยะ ตั้งแต่ระยะที่ 0-6 แต่ละระยะสามารถแยกออกเป็นขั้นตอน และ งานย่อย ๆ ได้ในขั้นตอนการพัฒนาระบบ ในปัจจุบันวิธี SSADM ได้พัฒนามาถึงเวอร์ชั่น 4.2 ซึ่งมีเครื่องมือแบบใหม่ ๆ เพิ่มขึ้นเพื่อให้สอดคล้องและยืดหยุ่นกับการออกแบบสารสนเทศในปัจจุบันมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีเชิงวัตถุ
(Object-Oriented Model)

ในบทนี้กล่าวถึงวิธีการพัฒนาระบบสารสนเทศเชิงวัตถุ ซึ่งวิธีการพัฒนาระบบเชิงวัตถุมีอยู่ด้วยกันหลายวิธี เช่น วิธีของ Peter Coad / Edward Yourdon, James Rumbaugh หรือ Greedy Booch เป็นต้น

วิธีการเชิงวัตถุ คือ การวิเคราะห์และการออกแบบ ระบบเชิงวัตถุที่มีลักษณะเป็นไปตามความจริงของปัญหาเพื่อที่ทำให้การนำเสนอเป็นไปได้ง่ายและถูกต้อง

3.1 แนวคิดของวิธีเชิงวัตถุ [22],[27],[28],[37],[39],[48]

แนวคิดพื้นฐานของวิธีเชิงวัตถุที่สำคัญได้แก่

1. **วัตถุ (Object)** คือ สิ่งที่เป็นรูปธรรมหรือนามธรรม ที่มีความหมายกับระบบและเป็นสิ่งที่ระบบสนใจเพื่อใช้เป็นตัวเก็บข้อมูล
2. **คลาส (Class)** คือ กลุ่มของวัตถุที่มีคุณสมบัติและพฤติกรรมที่เหมือนกัน
3. **การห่อหุ้ม (Encapsulation)** คือ โครงสร้างของข้อมูลและรายละเอียดต่าง ๆ ของวัตถุแต่ละตัวโดยวัตถุจะปกปิดไว้ ไม่ต้องการให้ผู้อื่นเห็น หรือที่เรียกว่า Information Hiding
4. **เมสเซจ (Message)** คือ การติดต่อกันระหว่างวัตถุด้วยกันเพื่อให้เกิดการเปลี่ยนแปลงภายในระบบ
5. **การสืบทอดคุณสมบัติ (Inheritance)** คือ การถ่ายทอดคุณสมบัติจากคลาสหลักมายังวัตถุภายในคลาสเดียวกัน
6. **โพลีมอร์ฟิซึม (Polymorphism)** คือ การทำงานที่มีลักษณะคล้ายกันแต่ความหมายของผลลัพธ์ไม่เหมือนกัน

3.2 การวิเคราะห์เชิงวัตถุ (Object-Oriented Analysis)

การวิเคราะห์เชิงวัตถุมีหลายวิธีโดยทั่วไปมีขั้นตอนดังนี้

3.2.1 แบบจำลองวัตถุ (Object Model)

แบบจำลองวัตถุ เป็นแบบจำลองที่อธิบายถึงโครงสร้างและความพันธ์ระหว่าง วัตถุในระบบ แบบจำลองวัตถุ ประกอบด้วยคลาสไดอะแกรม (Class Diagram) คือกราฟที่โหนด แทนคลาส (Class) และเส้นระหว่างโหนดแทนความสัมพันธ์ระหว่างคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 สัญลักษณ์ของวัตถุและคลาส

สัญลักษณ์ของคลาสเป็นรูปสี่เหลี่ยมผืนผ้าโดยแบ่งออกเป็น 3 ส่วน คือ ส่วนแรกแสดงชื่อของวัตถุหรือคลาส ส่วนที่สองแสดงแอตทริบิวต์ภายใน และส่วนสุดท้ายแสดงฟังก์ชัน (Function) ที่ใช้ในวัตถุหรือคลาส ดังแสดงดังรูปที่ 3.1

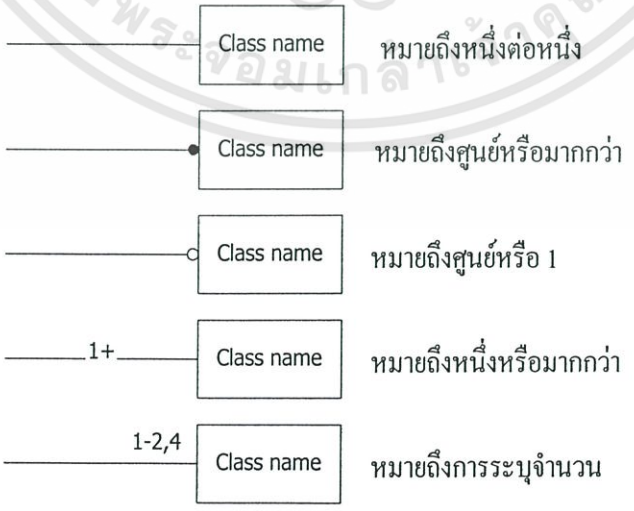
(ก) (ข)

รูปที่ 3.1 (ก) สัญลักษณ์ของคลาส (ข) ตัวอย่างของคลาส Person

2 ลิงค์และแอสโซซิเอชั่น (Link and Association)

ลิงค์ คือ ความสัมพันธ์ที่เป็นรูปธรรมหรือนามธรรมระหว่างวัตถุ ส่วนแอสโซซิเอชั่นจะอธิบายกลุ่มของลิงค์ที่มีโครงสร้างและความสัมพันธ์เหมือนกัน ระหว่างวัตถุของคลาสหนึ่งไปยังวัตถุของอีกคลาสหนึ่ง

- มัลติพลิซิตี้ (Multiplecity) คือ การระบุจำนวนสมาชิกของคลาสหนึ่ง ๆ ที่มีความสัมพันธ์กับจำนวนสมาชิกของอีกคลาสหนึ่ง สัญลักษณ์ที่ใช้ระบุมัลติพลิซิตี้ของแอสโซซิเอชั่นแสดงดังรูปที่ 3.2



รูปที่ 3.2 สัญลักษณ์ที่ใช้แทนมัลติพลิซิตีของแอสโซซิเอชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

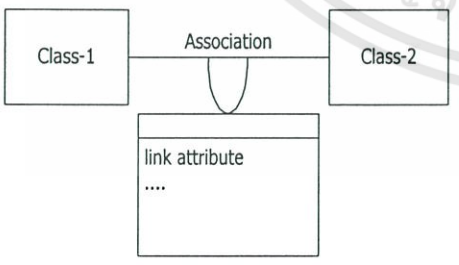
- บทบาท ในความสัมพันธ์ระหว่างคลาสตั้งแต่ 2 คลาสขึ้นไป คลาสแต่ละคลาสจะมีบทบาทแตกต่างกันไปในความสัมพันธ์นั้น หากต้องการระบุบทบาทของคลาสในความสัมพันธ์ ทำได้โดยเขียนชื่อของบทบาทที่ใต้เส้นแอสโซซิเอชั่นติดกับคลาสที่แสดงบทบาทนั้น ๆ สัญลักษณ์ของการระบุบทบาทแสดงดังรูปที่ 3.3 โดย role – 1 และ role – 2 เป็นบทบาทของ Class – 1 และ Class – 2 ในความสัมพันธ์ระหว่าง Class–1 และ Class– 2

(ก)

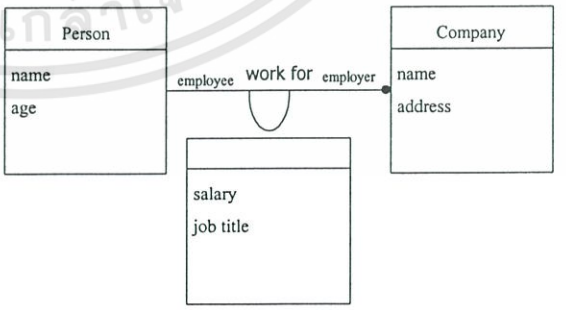
(ข)

รูปที่ 3.3 (ก) สัญลักษณ์ในการระบุบทบาทของคลาสในความสัมพันธ์ระหว่างคลาส
(ข) ตัวอย่าง ความสัมพันธ์ระหว่างคลาสที่มีการระบุบทบาท

- ลิงค์แอตทริบิวต์ คือคุณสมบัติที่เกิดจากความสัมพันธ์ของคลาสตั้งแต่ 2 คลาสขึ้นไป สัญลักษณ์ที่ใช้แทนลิงค์แอตทริบิวต์แสดงดังรูปที่ 3.4 โดยลิงค์แอตริบิวต์ดังกล่าว เป็นคุณสมบัติที่เกิดขึ้นจากความสัมพันธ์ระหว่าง Class–1 กับ Class-2



(ก)

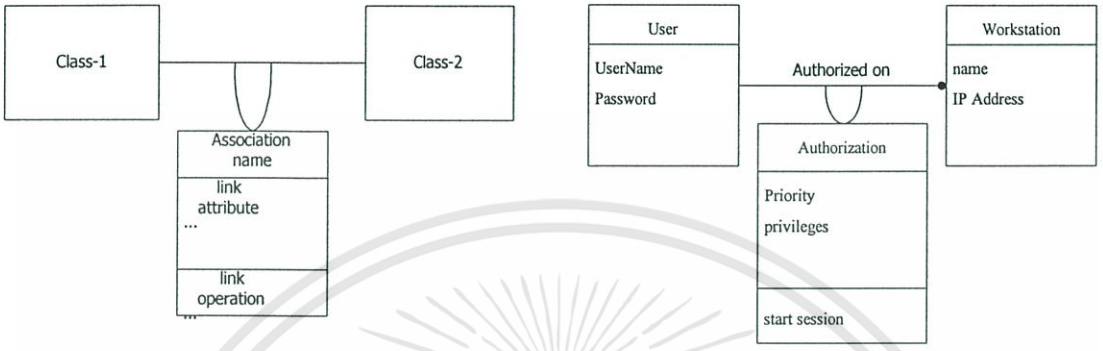


(ข)

รูปที่ 3.4 (ก) สัญลักษณ์ที่ใช้แทนลิงค์แอตทริบิวต์
(ข) ตัวอย่างความสัมพันธ์ระหว่างคลาสที่ทำให้เกิดลิงค์แอตทริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

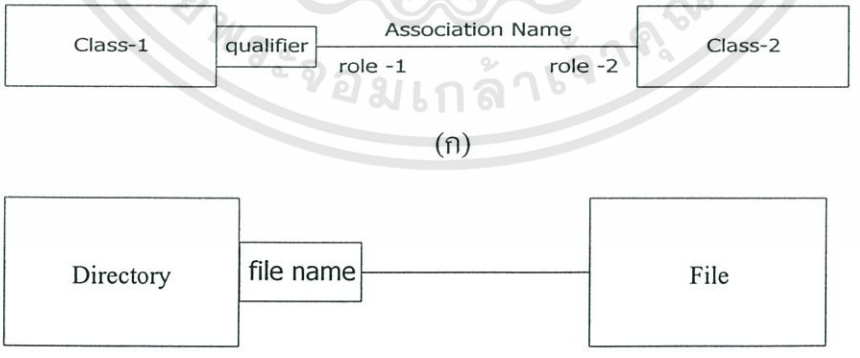
- แอสโซซิเอชั่นคลาส คือคลาสที่เกิดจากความสัมพันธ์ระหว่างคลาสตั้งแต่ 2 คลาสขึ้นไป สัญลักษณ์ที่แทนแอสโซซิเอชั่นคลาสแสดงดังรูปที่ 3.5 โดยความสัมพันธ์ระหว่าง Class–1 กับ Class–2 ทำให้เกิดแอสโซซิเอชั่นคลาสดังรูปดังกล่าว



(ก) (ข)

รูปที่ 3.5 (ก) สัญลักษณ์ที่ใช้แทนแอสโซสิเอชั่นคลาส
(ข) ตัวอย่างความสัมพันธ์ระหว่างคลาสที่ทำให้เกิดแอสโซสิเอชั่นคลาส

- ควอลิฟายด์แอสโซซิเอชั่น คือแอตทริบิวต์ที่เรียกว่าควอลิฟายเออร์ ซึ่งจะช่วยลดมัลติพลิซิตี้ของความสัมพันธ์ สัญลักษณ์ที่ใช้แทนควอลิฟายด์แอสโซซิเอชั่นคือรูปกล่องสี่เหลี่ยมเขียนชื่อของควอลิฟายเออร์ไว้ภายใน โดยเกาะติดอยู่กับคลาสที่ถูกกำหนดควอลิฟายเออร์ การกำหนดควอลิฟายเออร์แสดงดังรูปที่ 3.6



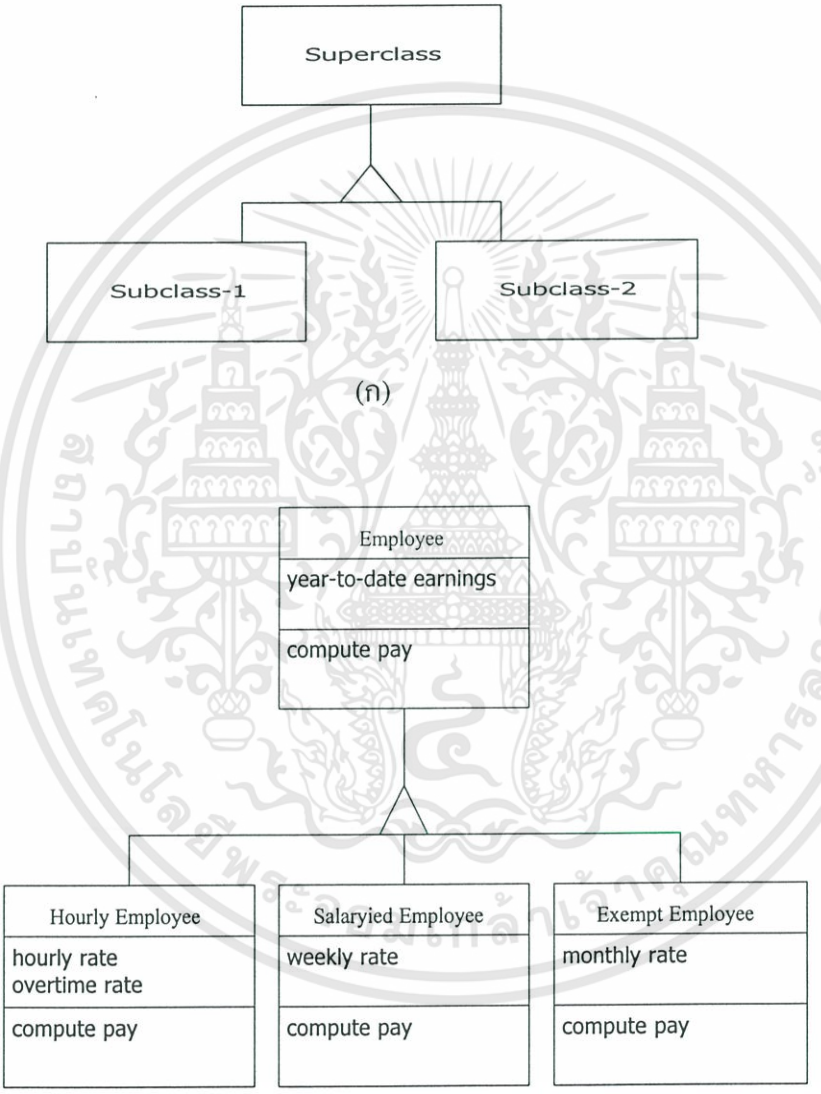
(ก)

(ข)

รูปที่ 3.6 (ก) สัญลักษณ์ที่ใช้แทนควอลิฟายด์แอสโซสิเอชั่น
(ข) ตัวอย่างของควอลิฟาย์แอสโซสิเอชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

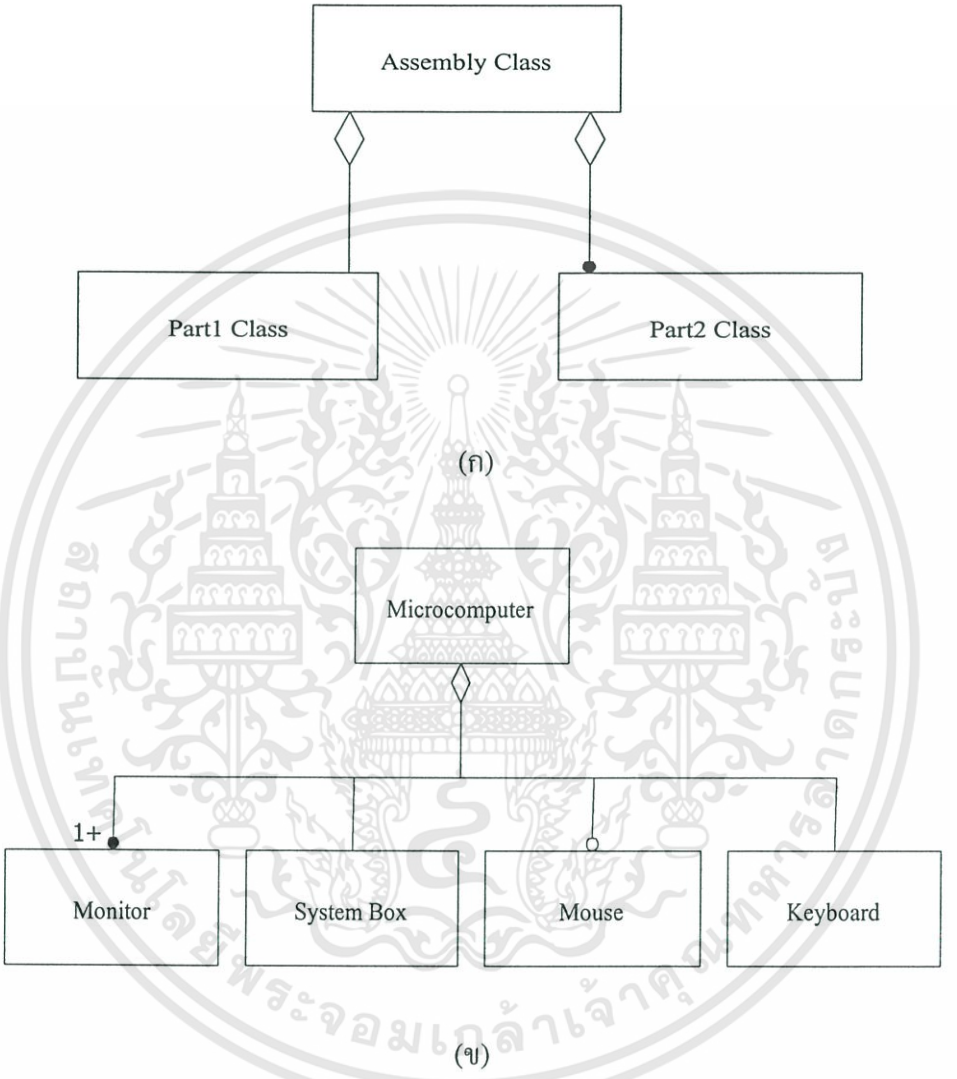
- การสืบทอดคุณสมบัติ (Inheritance) เป็นการแสดงความสัมพันธ์ตามลำดับชั้นที่มีการถ่ายทอดคุณสมบัติและพฤติกรรมจากคลาสหนึ่งไปยังอีกคลาสหนึ่ง เรียกคลาสที่เป็นผู้ถ่ายทอดคุณสมบัติและพฤติกรรมว่า "ซุปเปอร์คลาส (Superclass)" และคลาสที่ได้รับการถ่ายทอดคุณสมบัติและพฤติกรรมว่า "ซับคลาส (Subclass)" สัญลักษณ์ที่ใช้แสดงการถ่ายทอดคุณสมบัติและพฤติกรรมระหว่างคลาสแสดงดังรูปที่ 3.7



รูปที่ 3.7 (ก) สัญลักษณ์ที่ใช้แทนการถ่ายทอดคุณสมบัติ
(ข) ตัวอย่างการถ่ายทอดคุณสมบัติระหว่างคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การประกอบกันของคลาส (Aggregation) เป็นการแสดงการรวมกันของคลาส เพื่อให้เกิดคลาสอีกคลาสหนึ่ง มีสัญลักษณ์ดังรูปที่ 3.8 กล่าวคือ คลาส Assembly ประกอบไปด้วยคลาส Part1 และคลาส Part2 เป็นแอตทริบิวต์



รูปที่ 3.8 (ก) สัญลักษณ์ที่ใช้แทนการประกอบกันของคลาส
(ข) ตัวอย่างแสดงการประกอบกันของคลาส

3.2.2 แบบจำลองไดนามิค (Dynamic Model)

1. <u>เหตุการณ์และสถานะ (events and states)</u>

จากที่กล่าวมาแบบจำลองเชิงวัตถุได้อธิบายถึงรูปแบบของวัตถุ แอตทริบิวต์และการเชื่อมต่อระหว่างคลาส/วัตถุ ค่าของแอตทริบิวต์และการเชื่อมต่อของวัตถุเรียกว่า "สถานะ" ซึ่งจะเลียนแบบวัตถุแต่ละอัน ผลลัพธ์ที่ได้คือการเปลี่ยนสถานะ สิ่งที่เข้ามากระทบกับแต่ละวัตถุ เรียกว่า "เหตุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การณ์" ซึ่งสถานะขึ้นอยู่กับเหตุการณ์ของสิ่งที่วัตถุได้รับมาหรือส่งเหตุการณ์ไปยังวัตถุอื่นในรูปของเหตุการณ์ สถานะและการเปลี่ยนสถานะของคลาสแสดงอยู่ในรูปของไดอะแกรมที่เรียกว่า "สเตทไดอะแกรม (State diagram)" สเตทไดอะแกรม คือ การแสดงของเหตุการณ์ที่เกิดขึ้น ณ เวลาใดเวลาหนึ่ง ซึ่งต่างจากวัตถุไดอะแกรมที่แสดงการเชื่อมต่อระหว่างคลาสและแสดงความสัมพันธ์ของคลาส แบบจำลองไดนามิค ประกอบด้วย สเตทไดอะแกรมมากกว่าหนึ่งไดอะแกรม ในหนึ่งไดอะแกรมของแต่ละกราฟมีความสำคัญกับพฤติกรรมและแสดงถึงกิจกรรมของระบบ โดยแต่ละสถานะจะทำงานตามลำดับและสามารถเปลี่ยนได้อิสระ

1.1 <u>เหตุการณ์(event)</u>

เหตุการณ์ คือ การกระทำใด ๆ ที่เข้ามากระทบกับวัตถุ ณ เวลาใดเวลาหนึ่งและไม่มีช่วงเวลาที่แน่นอนบางครั้งเหตุการณ์หนึ่งอาจเกิดขึ้นก่อนหรือเกิดหลังอีกเหตุการณ์หนึ่ง หรืออาจเกิดขึ้นพร้อมกันแต่ไม่มีความเกี่ยวข้องกันเลย เช่น เที่ยวบิน 123 ออกจากดอนเมืองก่อนที่จะไปถึงสนามบินเชียงใหม่ ซึ่งทั้งสองเหตุการณ์มีความเกี่ยวข้องกัน หรือเที่ยวบิน 123 ออกไปก่อนหรือหลังเที่ยวบิน 456 ที่ขึ้นจากสนามบินเชียงใหม่สังเกตว่าทั้งสองเหตุการณ์ไม่มีความเกี่ยวข้องกัน กล่าวได้ว่าทั้งสองเหตุการณ์เกิดขึ้นพร้อมกันแต่ไม่มีผลกระทบซึ่งกันและกัน ถ้าการติดต่อสื่อสารของทั้ง 2 สนามบินถูกเลื่อนออกไปทำให้เหตุการณ์ที่เกิดขึ้นมีความแตกต่างกัน แต่เหตุการณ์ที่เกิดขึ้นนั้นเกิดขึ้นพร้อมกันจะไม่มีผลกระทบต่อกันถึงแม้ว่าเหตุการณ์ทั้ง 2 จะไม่ต่อเนื่องกันก็ตาม (สิ่งที่เกิดขึ้นพร้อมกันไม่ควรที่จะนำไปใส่ในแบบจำลองของระบบเพราะว่าเหตุการณ์นั้นๆจะไปเกิดกับทุกๆระดับว่าในความเป็นจริงการกระจายข้อมูลประกอบไปด้วยเหตุการณ์และกิจกรรมที่เกิดขึ้นพร้อมกันก็ตาม)

การส่งเหตุการณ์ไปยังวัตถุอื่นซึ่งเป็นการส่งข้อมูลแบบทางเดียว และไม่เหมือนกับซับรูทีน (Subroutine) ที่มีการส่งค่าไปมาระหว่างโมดูล ถ้าวัตถุมีการทำงานพร้อมกันหลายวัตถุ การส่งเหตุการณ์ระหว่าง 2 วัตถุ บางครั้งอาจมีบางวัตถุไม่มีการส่งค่ากลับออกมาก็ได้ ทั้งนี้ขึ้นอยู่กับตัวควบคุมการทำงานของวัตถุแต่ละตัว เช่น วัตถุ A ส่งเหตุการณ์ไปยังวัตถุ B เพื่อให้เกิดการเปลี่ยนแปลงแอตทริบิวต์ให้กับวัตถุที่ 2 เมื่อสิ้นสุดทำงาน ก็ไม่จำเป็นต้องส่งค่ากลับออกมา เป็นต้น

เหตุการณ์ที่แสดงมีความเป็นอิสระต่อกัน แต่ในที่นี้จะกำหนดให้เหตุการณ์อยู่ในกลุ่มของคลาสเพื่อนำมากำหนดโครงสร้างและพฤติกรรมของคลาส โดยมีลักษณะเป็นโครงสร้างแบบลำดับชั้นทั้งวัตถุและคลาส เช่น เที่ยวบิน 123 ออกจากเชียงใหม่กับเที่ยวบิน 450 ออกจากกรุงเทพ ซึ่งทั้ง 2 เป็นเหตุการณ์ของคลาสเที่ยวบิน เป็นต้น จากตัวอย่างกล่าวได้ว่าเหตุการณ์นั้นมีลักษณะโครงสร้างและพฤติกรรมเหมือนกันทั้งคลาสและมีประเภทเหมือนกัน บางครั้งเหตุการณ์ที่มีเป็นเพียงการส่งสัญญาณหรือบอกอาการเท่านั้น ซึ่งส่วนมากเหตุการณ์ของคลาสจะมีแอตทริบิวต์เป็นตัวกำหนด เช่น คลาสของเที่ยวบินมีแอตทริบิวต์ เที่ยวบิน เลขที่เที่ยวบิน และเมือง เป็นต้น โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุการณ์ที่เข้ามากระทบคลาสหรือวัตถุ ทำให้แอตทริบิวต์ซึ่งอยู่ภายในวัตถุถูกกระทบไปตลอดจนกระทั่งสิ้นสุดเหตุการณ์นั้น

วิธีการส่งเหตุการณ์จากคลาสหนึ่งไปยังอีกคลาสหนึ่งนั้น ส่งโดยระบุค่าตามหลังคลาสซึ่งประกอบด้วย แอตทริบิวต์ที่อยู่ในแต่ละคลาสหรือวัตถุ สังเกตจากรูปที่ 3.9 แสดงถึงวิธีการส่งค่าจากคลาสให้กับเหตุการณ์ต่าง ๆ เช่น คลาสเที่ยวบินส่งหมายเลขเที่ยวบิน ชื่อเมืองและสายการบิน เป็นต้น การส่งค่าของเหตุการณ์นั้น บางครั้งอาจไม่จำเป็นที่จะต้องส่งทุกครั้ง ซึ่งบางเหตุการณ์ต้องการเพียงรับอย่างเดียว

บางครั้งเหตุการณ์อาจมีการทำงานที่คลุมเครืออยู่บ้าง แต่ในทางปฏิบัติไม่มีปัญหา เพราะทุก ๆ เหตุการณ์จะแสดงคำอธิบายของทุกเหตุการณ์ไว้อย่างชัดเจน

Airplane flight departs (airline, flight number, city)
Mouse button pushed (button, location)
Input string entered (text)
Phone receiver lifted
Digit dialed (digit)
Engine speed enters danger zone

รูปที่ 3.9 แสดงเหตุการณ์ของคลาสและแอตทริบิวต์

เหตุการณ์ที่นำไปใช้อาจมีลักษณะที่ไม่ชัดเจน บางครั้งเหตุการณ์ที่เกิดขึ้นอ้างถึงลักษณะเฉพาะเจาะจงลงไปแต่ ณ เวลาเดียวกันเหตุการณ์ที่อ้างถึงเป็นเหตุการณ์ตามความเป็นจริง เหตุการณ์ลักษณะเช่นนี้ไม่มีผลกับการนำไปใช้งาน ข้อผิดพลาดที่เกิดขึ้นกับเหตุการณ์จะถูกรวบรวมไว้ในที่นี้ด้วย เช่น เครื่องยนต์หยุดทำงาน การขนส่งถูกยับยั้งหรือหมดเวลา เป็นต้น ซึ่งสิ่งเหล่านี้เป็นเพียงข้อผิดพลาดบางครั้งเท่านั้น

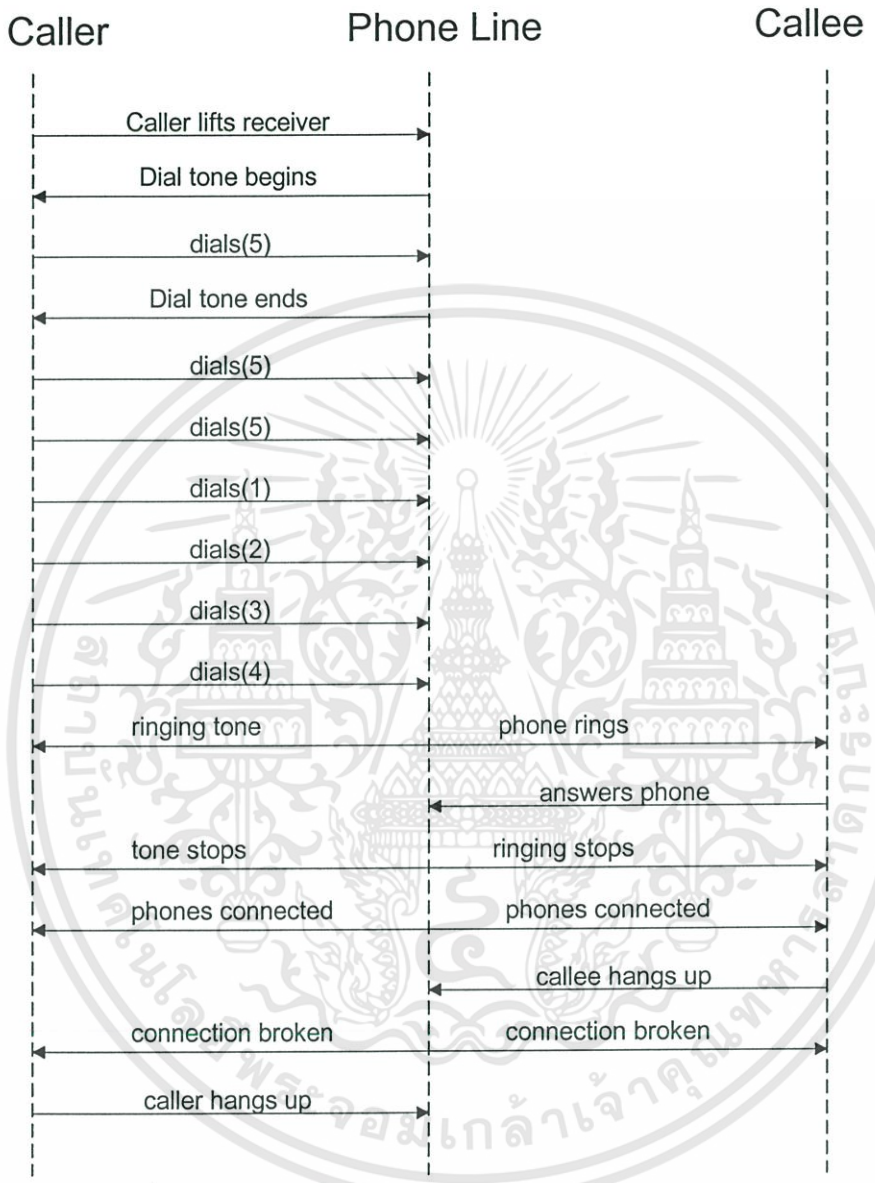
1.2 <u>แผนงานและอีเวนท์เทรส (Scenarios and Event Traces)</u>

แผนงานและอีเวนท์เทรส คือ แบบแผนของการลำดับของเหตุการณ์ที่เกิดขึ้นเฉพาะในระบบ ขอบเขตของแผนงานมีลักษณะไม่ตายตัว อาจรวมเหตุการณ์ทั้งหมดของระบบหรือบางส่วนเท่านั้น นอกจากนี้แบบแผน คือ ส่วนที่คอยบันทึกสิ่งต่างๆที่เกิดขึ้นกับแผนงานด้วย

เหตุการณ์จะส่งข้อมูลจากวัตถุหนึ่งไปยังวัตถุอื่น เช่น การหมุนโทรศัพท์จะส่งสัญญาณจากสายโทรศัพท์ไปยังผู้รับ เป็นต้น หลังจากเขียนแบบแผนแล้วขั้นตอนถัดมา คือ การกำหนดเหตุการณ์ที่เกิดขึ้นในแต่ละวัตถุของผู้รับกับผู้ส่งซึ่งจะลำดับการเปลี่ยนแปลงของเหตุการณ์ที่เกิดขึ้นในรูปของ “Event trace diagram” โดยไดอะแกรมนี้จะแสดงวัตถุในแนวดิ่งและแสดงเหตุการณ์ในแนวขวาง (ดังแสดงในรูปที่ 3.10) จากนั้นใช้ลูกศรเป็นเครื่องหมายแสดงทิศทางของเหตุการณ์ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระทบกับวัตถุผู้รับและผู้ส่ง ระยะเวลาที่เกิดขึ้นจะถูกรวบรวมตั้งแต่เริ่มเหตุการณ์จนถึงสิ้นสุด แต่ถ้ามีช่องว่างเกิดขึ้น หมายถึงไม่มีเหตุการณ์ใด ๆ เข้ามากระทบให้เกิดการเปลี่ยนแปลงและจะไม่นำมารวมกับระยะเวลาที่เกิดขึ้นจริง



รูปที่ 3.10 แสดงเหตุการณ์ของสายโทรศัพท์
ที่มา[28]

1.3 <u>สถานะ(states)</u>

สถานะ คือ การเปลี่ยนแปลงค่าแอตทริบิวต์ภายในวัตถุเมื่อมีเหตุการณ์ใด ๆ เข้ามากระทบกับวัตถุนั้น ภายในสถานะประกอบด้วย กลุ่มของแอตทริบิวต์ที่มีผลกับพฤติกรรมของวัตถุ เช่น สถานะทางด้านเร่งรัดหนี้สินของธนาคารมี 2 สถานะคือ ชำระหนี้ได้กับชำระหนี้ไม่ได้ ซึ่งทั้ง 2 สถานะก็ขึ้นอยู่กับทรัพย์สินและหนี้สินของธนาคารที่มีอยู่ การตอบสนองของสถานะในวัตถุขึ้นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับจำนวนของเหตุการณ์ที่เข้ามากระทบและความถูกต้องของเหตุการณ์ที่รับเข้ามา เช่น เหตุการณ์ที่รับเข้ามามีจำนวนมากแต่ค่าที่รับเข้ามามีค่าที่เหมือนกันจึงทำที่สถานะเดียวกัน หรือจำนวนของเหตุการณ์มีค่าที่แตกต่างกันอาจกระทำกับสถานะที่ต่างกัน เป็นต้น ผลกระทบของเหตุการณ์ต่างๆเหล่านี้สามารถทำให้กิจกรรมหรือสถานะภายในวัตถุมีการเปลี่ยนแปลง เช่น ถ้าตัวเลข 0 เป็นการหมุนในสถานะของการหมุนโทรศัพท์ กล่าวคือ ถ้ามีการหมุนโทรศัพท์เกิดขึ้นก็จะทำให้สถานะของการหมุนโทรศัพท์เปลี่ยนแปลงจากเตรียมพร้อมไปสู่สถานะกำลังหมุนแทนที่และในทางกลับกัน ถ้าวางสายโทรศัพท์ลงสถานะของการหมุนโทรศัพท์กลับไปสู่สถานะเตรียมพร้อมอีกครั้งหนึ่ง เป็นต้น

ภายในหนึ่งสถานะของแต่ละวัตถุสามารถรับเหตุการณ์ได้ 2 เหตุการณ์ซึ่งเหตุการณ์แทนหน่วยของเวลาและสถานะแทนช่วงของเวลา เช่น หลังจากที่ผู้รับยกหูโทรศัพท์ซึ่งก่อนหน้านั้นเป็นการหมุนหมายเลขโทรศัพท์ของผู้ส่งซึ่งทำให้สายโทรศัพท์อยู่ในสถานะของการเรียก เป็นต้น สถานะของวัตถุขึ้นอยู่กับลำดับเหตุการณ์ที่ผ่านมาแล้ว แต่ส่วนมากเหตุการณ์เหล่านี้จะถูกป้อนโดยเหตุการณ์ต่อมา เช่น เหตุการณ์ที่เกิดขึ้นก่อนการโทรศัพท์ คือการที่หูโทรศัพท์ถูกวางอยู่ทำให้ไม่มีผลกระทบกับพฤติกรรมใด ๆ และทำให้สถานะเตรียมพร้อมไม่อยู่ในความสนใจเพราะเหตุการณ์ที่เกิดขึ้นนั้นคือวางหูโทรศัพท์

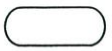
ณ ช่วงเวลาหนึ่งของสถานะอาจมีเหตุการณ์ที่เกิดขึ้นและมีความเกี่ยวเนื่องกับกิจกรรมที่ดำเนินอยู่ ทำให้กิจกรรมต่าง ๆ ที่เกิดขึ้นเสร็จไปพร้อมกัน เช่น เที่ยวบินจากดอนเมืองไปเชียงใหม่และขณะเดียวกันเครื่องบินจากเชียงใหม่ไปดอนเมือง เป็นต้น สังเกตได้ว่าเหตุการณ์และสถานะเป็นสิ่งที่คู่กัน ดังนั้น กล่าวได้ว่า เหตุการณ์หนึ่งสามารถแบ่งออกเป็น 2 สถานะ และแต่ละสถานะสามารถแบ่งออกเป็น 2 เหตุการณ์

บางกรณีค่าของสถานะในวัตถุมีความเกี่ยวข้องกับเงื่อนไข เช่น น้ำเป็นของเหลว จึงสรุปได้ว่า “น้ำควรมีอุณหภูมิมากกว่า 0 องศาเซลเซียสและน้อยกว่า 100 องศาเซลเซียส” เพราะว่าถ้ามากกว่าหรือน้อยกว่านั้น สามารถทำให้น้ำเปลี่ยนสภาพได้ เป็นต้น แต่บางกรณีถ้ามองคุณลักษณะก็สามารถที่บ่งบอกถึงการเปลี่ยนแปลงได้เช่นกัน เช่น เกียร์ของรถยนต์อาจมีสถานะเป็นเกียร์ถอยหลัง เกียร์ว่าง หนึ่ง สอง หรือสาม เป็นต้น

1.4 <u>สเตทไดอะแกรม (State Diagram)</u>

สเตทไดอะแกรมประกอบด้วยเหตุการณ์และสถานะ กล่าวคือ สถานะที่มีอยู่เกิดการเปลี่ยนแปลงไป เนื่องจากการกระทบของเหตุการณ์ที่ได้รับมา ดังนั้นสถานะที่ถูกเปลี่ยนแปลงจากเหตุการณ์ที่เข้ามากระทบเรียกว่า “Transition” สเตทไดอะแกรมแสดงในรูปของแผนภูมิภาพซึ่งประกอบด้วยโหนดหรือสถานะและเส้นทางของการเปลี่ยนแปลงหรือ Transition สัญลักษณ์ของสถานะถูกแทนด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 และภายในประกอบด้วยชื่อของสถานะนั้น ๆ ส่วน Transition ถูกแทนด้วยลูกศรที่ออกจากสถานะผู้ส่งไปยังสถานะของผู้รับและแสดงชื่อของเหตุการณ์ที่เกิดขึ้นไว้บน Transition ซึ่ง Transition ทั้งหมดจะออกจากสถานะไปตามเหตุการณ์ที่แตกต่างกันออกไป

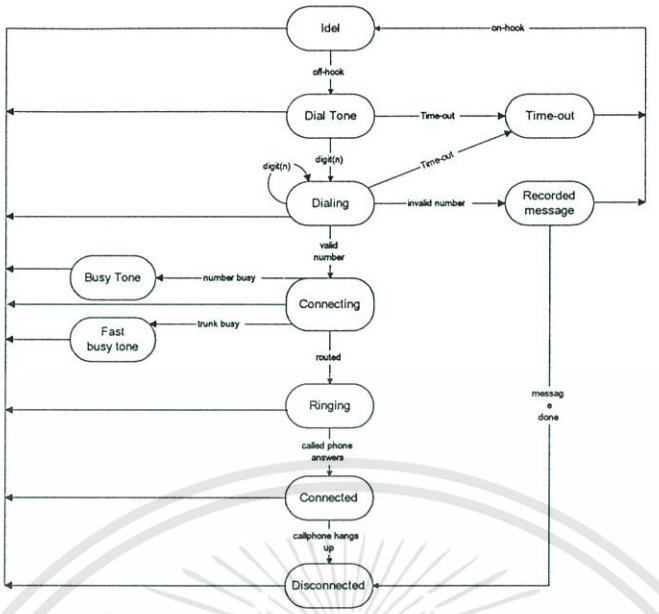
สเตทไดอะแกรมอธิบายถึงลำดับการทำงานที่ขึ้นตามลำดับของเหตุการณ์ที่เข้ามากระทบ ถ้าสถานะและเหตุการณ์ของวัตถุ คือ การส่งผ่านค่า ดังนั้นการส่งผ่านค่าสิ้นสุดเมื่อสถานะของวัตถุนั้นมีการทำงานเกิดขึ้น ดังนั้นการส่งผ่านค่าเรียกว่า “fire” ถ้าการส่งผ่านของสถานะมีมากกว่าหนึ่งครั้งเหตุการณ์แรกที่เข้ามากระทบจะเข้าไปกระตุ้นเหตุการณ์ที่ต้องการทำงาน แต่ถ้าสถานะปัจจุบันไม่มีเหตุการณ์ใด ๆ เกิดขึ้นทำให้เหตุการณ์นั้นถูกมองข้ามไป ซึ่งลำดับการทำงานของเหตุการณ์จะทำตามแผนภูมิที่แสดงไว้

จากรูปที่ 3.11 แสดงถึงพฤติกรรมของสายโทรศัพท์ที่ไม่มีทั้งผู้รับและผู้ส่ง แผนภูมินี้ประกอบด้วยลำดับการทำงานที่ประกอบด้วยเหตุการณ์แบบปกติและไม่ปกติ เช่น กรณียกเลิกการใช้สายโทรศัพท์ สายโทรศัพท์ไม่ว่าง เป็นต้น จากรูปเหตุการณ์ทุกเหตุการณ์ของแต่ละสถานะ มีเส้นทางกลับสู่สถานะ Idle ที่เป็นสถานะของการรอรับเหตุการณ์ใหม่เข้ามากระทบให้เกิดการทำงานขึ้นอีกครั้ง

สังเกตว่าสถานะของวัตถุไม่ใช่ขั้นตอนของการกำหนดค่าให้กับวัตถุ เช่น สถานะของการหมุนเป็นเพียงการรวบรวมเหตุการณ์บางส่วนของหมายเลขโทรศัพท์เท่านั้น แต่ผลที่ได้มีความแตกต่างกันเพราะการกดตัวเลขต่างกัน แม้ว่าสถานะต่าง ๆ เหมือนกันก็ตาม แต่ตัวเลขที่ถูกกดจะถูกนำไปเก็บไว้ในแอตทริบิวต์ เป็นต้น

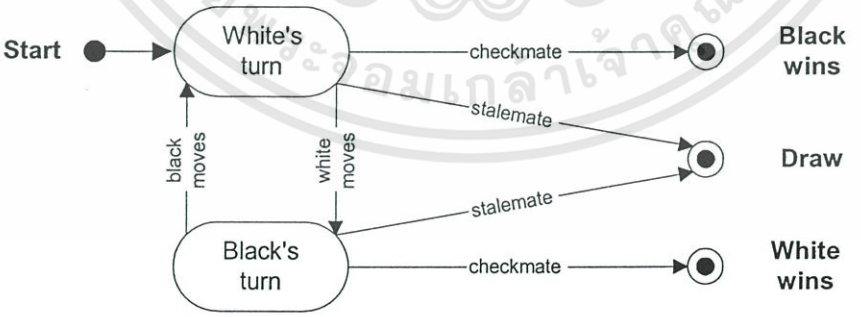
สเตทไดอะแกรมเป็นการอธิบายถึงพฤติกรรมของวัตถุทุกตัวภายในคลาส ดังนั้นทุก Instance ของคลาสจะมีพฤติกรรมที่เหมือนกันอยู่ จึงทำให้มีส่วนที่เหมือนกันและสามารถนำไปใช้ร่วมกันได้ และแต่ละวัตถุจะมีค่าของแอตทริบิวต์ สถานะ และผลลัพธ์ ที่รับเข้ามาเป็นของตนเอง ส่วนการกระทำของวัตถุนั้น ๆ ขึ้นอยู่กับวัตถุอื่นที่เข้ามากระทบ ส่วนการปฏิบัติงานภายในขึ้นอยู่กับกระบวนการของตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงสถานะของสายโทรศัพท์
ที่มา [28]

ในสเตทไดอะแกรมสามารถอธิบายการทำงานได้ทั้งแบบต่อเนื่องหรือแบบรอบเดียว ไดอะแกรมของการใช้สายโทรศัพท์ (ดังแสดงในรูปที่ 3.11) เป็นการทำงานแบบต่อเนื่องที่อธิบายถึงการใช้โทรศัพท์ ซึ่งไม่จำเป็นต้องสนใจว่าการทำงานแบบต่อเนื่องนั้นถูกเริ่มต้นที่ไหน ส่วนไดอะแกรมแสดงการทำงานรอบเดียว มักเป็นการทำงานที่มีจุดเริ่มต้นและจุดสิ้นสุดที่แน่นอน ดังนั้นจุดเริ่มต้นจะเริ่มจากวัตถุที่สนใจและดำเนินการไปเรื่อยจนถึงจุดสิ้นสุดของวัตถุนั้น จุดเริ่มต้นของไดอะแกรมจะถูกแทนด้วยจุดทึบสีดำพร้อมกับทิศทางและเงื่อนไขที่เกี่ยวข้องส่วนจุดสิ้นสุดถูกแทนด้วยจุดทึบสีดำเช่นกันพร้อมด้วยวงกลมล้อมรอบและมีลูกศรที่ชี้เข้ามากับชื่อหรือเงื่อนไขของเหตุการณ์นั้น ดังแสดงในรูปที่ 3.12 อธิบายถึงขั้นตอนของการเล่นเกมหมากรุก



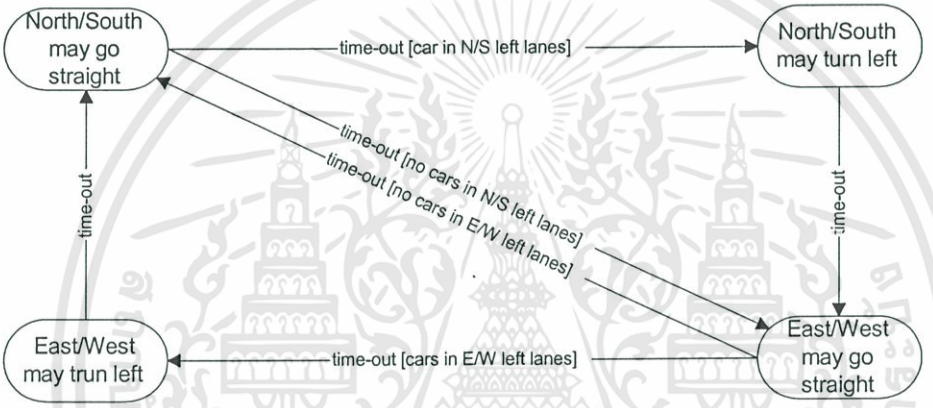
รูปที่ 3.12 แสดงสเตทไดอะแกรมทางเดียวของเกมหมากรุก
ที่มา [28]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 เงื่อนไข (Condition)

เงื่อนไข คือ การเปรียบเทียบค่าของวัตถุในลักษณะบูลลีนและคอยตรวจสอบความถูกต้อง ณ เวลาใดเวลาหนึ่งเท่านั้น เช่น อากาศหนาวมีอุณหภูมิต่ำกว่าปกติหรืออากาศหนาวอยู่ระหว่างวันที่ 15 พ.ย. ถึง 3 มี.ค.ของทุกปี เป็นต้น เงื่อนไขเป็นสิ่งสำคัญมากเพราะทำให้เหตุการณ์ที่เกิดขึ้นมีความแตกต่างกันออกไป และเงื่อนไขสามารถกำหนดได้จากตอนเริ่มต้นสถานะ

เงื่อนไขแต่ละเงื่อนไขที่ถูกส่งผ่านสามารถใช้ตัวขยายหรือถ้อยแถลงคอยกำกับให้เงื่อนไขนั้นทำตามได้อย่างถูกต้อง เช่น พรุ่งนี้ฉันจะไปธุระนอกบ้านตอนเช้า (เหตุการณ์) ถ้าเกิดอุณหภูมิหนาวเย็น (เงื่อนไข) ให้หยิบถุงมือไปด้วย (สถานะถัดไป) เป็นต้น ตัวขยายหรือตัวที่คอยกำกับถูกแสดงเหมือนกับบูลลีนที่แสดงอยู่ภายในเครื่องหมายก้ามปูและอยู่หลังชื่อเหตุการณ์



รูปที่ 3.13 แสดงการส่งผ่านค่าของสเตทไดอะแกรมพร้อมเงื่อนไข
ที่มา [28]

จากรูปที่ 3.13 แสดงสเตทไดอะแกรมที่ขยายสถานะของสัญญาณไฟจราจรตามสี่แยก คือด้านหนึ่งของสัญญาณไฟคอยควบคุมไฟทางทิศเหนือและทิศใต้ส่วนอีกด้านคอยควบคุมสัญญาณทางทิศตะวันออกและตะวันตกเงื่อนไข คือ ถ้าไม่มีรถผ่านทางด้านหนึ่งด้านใดสัญญาณไฟจะสลับสัญญาณไปอีกทางด้านหนึ่งทันทีและปฏิบัติลักษณะเช่นนี้ไปเรื่อย ๆ

2. ตัวดำเนินการ (Operation)

จากหัวข้อที่ 1 ได้กล่าวถึงสเตทไดอะแกรมที่อธิบายถึงสถานะและเหตุการณ์ของคลาส/วัตถุเพียงอย่างเดียวแต่หัวข้อนี้อธิบายถึงตัวกระตุ้นที่ทำให้เหตุการณ์ที่เกิดขึ้นสามารถดำเนินงานได้

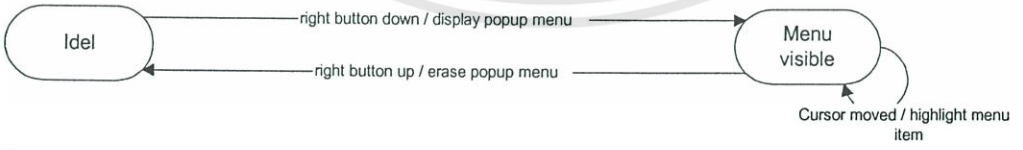
2.1 ตัวควบคุมการดำเนินงาน (Controlling Operations)

สเตทไดอะแกรมเป็นเพียงการอธิบายถึงรูปแบบของเหตุการณ์และพฤติกรรมที่เกิดขึ้นกับวัตถุและยังอธิบายถึงความสัมพันธ์ของวัตถุที่มีต่อเหตุการณ์อีกด้วย ส่วนการปฏิบัติเป็นเพียงการติดต่อกับสถานะหรือคอยส่งเหตุการณ์ที่เกิดขึ้นผ่านไปยังสถานะเพื่อให้เหตุการณ์ตรงกันเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในสถานะหนึ่งประกอบด้วยกิจกรรมต่างๆ ซึ่งกิจกรรมเหล่านี้ทำให้สถานะสามารถดำเนินงานได้สมบูรณ์มากขึ้น และส่วนใหญ่กิจกรรมมักจะมีการทำงานที่ต่อเนื่องกัน ตัวอย่าง การแพร่ภาพทางโทรทัศน์จะแสดงไปเรื่อย ๆ จนกระทั่งจะถูกขัดจังหวะการดำเนินงาน เช่น การเปิดหรือปิดโทรทัศน์ เป็นต้น บางครั้งสถานะอาจเข้ามามีส่วนร่วมโดยคอยควบคุมการปฏิบัติงานของกิจกรรม เช่น กริ่งโทรศัพท์ดังขึ้นอย่างต่อเนื่องจนกระทั่งจะมีเหตุการณ์เข้ามากระทบหรือขัดจังหวะให้สิ้นสุดลงซึ่งเหตุการณ์ที่เข้ามากระทบนี้อาจถูกส่งมาจากสถานะอื่น เป็นต้น ซึ่งสัญลักษณ์ของสถานะจะปรากฏข้อความของกิจกรรมอยู่ภายในคือ “do : A” (do คือ คำสั่ง A คือ กิจกรรมที่ต้องกระทำ) โดยข้อความดังกล่าวคือการกำหนดลำดับการทำงานตั้งแต่จุดเริ่มต้นจนถึงจุดสิ้นสุดของสถานะ เช่น หุ่นยนต์กำลังเคลื่อนย้ายสิ่งของไปข้างหน้าเพื่อให้เสร็จสมบูรณ์หรืออาจถูกขัดจังหวะจากเหตุการณ์อื่นเพื่อไม่ให้การทำงานนั้นสมบูรณ์สังเกตว่า “do : A” คือการกำหนดลำดับการทำงานของกิจกรรมภายในสถานะเพื่อให้สามารถดำเนินงานได้สมบูรณ์ตั้งแต่ต้นจนจบสถานะ ถ้าเหตุการณ์ถูกส่งมาก่อนที่สถานะจะทำกิจกรรมเสร็จ ย่อมทำให้กิจกรรมนั้นถูกขัดจังหวะก่อนเวลาอันควร เช่น หุ่นยนต์อาจเกิดการปะทะกับเหตุการณ์หนึ่งเหตุการณ์ใดทำให้การเคลื่อนไหวยุติลง เป็นต้น ตามความจริงแล้วคำว่า “กิจกรรมดำเนินอย่างต่อเนื่อง” กับ “ลำดับของกิจกรรม”ไม่มีความแตกต่างกันเพราะขึ้นอยู่กับการสังเกต

เหตุการณ์แต่ละเหตุการณ์ประกอบด้วย การกระทำ(Action) และการทำงานของตัวกระทำจะเกิดขึ้นทันทีที่เหตุการณ์นั้นเข้ามากระทบ การกระทำถูกแทนด้วยการปฏิบัติถ้าเปรียบเทียบกับสเตทไดอะแกรมแล้วนับว่ามีความสำคัญน้อยมาก เช่น รูปที่ 3.11 แสดงถึงโทรศัพท์ถูกตัดการติดต่อหรือกรณีสายหลุดอาจเป็นเพียงการกระทำเพื่อตอบสนองเหตุการณ์ On-hook ของสายโทรศัพท์ เป็นต้น ตามความเป็นจริงการกระทำไม่ได้เกิดขึ้นทันทีแต่แบบจำลองการกระทำมีส่วนทำให้ผู้ออกแบบไม่ต้องกังวลถึงการควบคุมโครงสร้างภายในไดอะแกรมของสถานะแต่ถ้าไม่มีการกระทำเข้ามาช่วย ทำให้ต้องคอยดูแลกิจกรรมตั้งแต่เริ่มต้นจนกระทั่งสิ้นสุดของเหตุการณ์และเหตุการณ์ที่เกิดขึ้นกระทันหันอีกด้วย



รูปที่ 3.14 แสดงการกระทำของเมนูแบบ POP-UP
ที่มา [28]

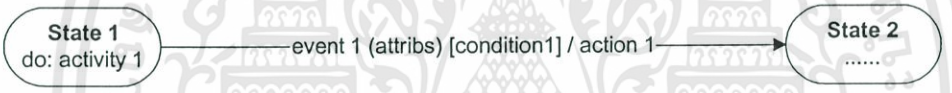
การกระทำสามารถแทนการควบคุมของการดำเนินงานภายใน เช่น กำหนดคุณลักษณะหรือทำให้เกิดเหตุการณ์อื่น เป็นต้น และการกระทำยังเป็นวิธีการสำหรับการควบคุมโครงสร้างกับการนำเสนอ เช่น การติดต่อที่เกิดขึ้นภายในอาจนับเหตุการณ์ทุกครั้งที่เข้ามากระทบ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์ที่ใช้แทนการกระทำอยู่หลังเครื่องหมาย "/" และชื่อของการกระทำตามลำดับ (ซึ่งจะอยู่หลังชื่อเหตุการณ์) จากรูปที่ 3.14 เป็นการแสดงสเตทไดอะแกรมเมนูของเวอร์สเตชั่น (Workstation) เมื่อปุ่มทางด้านขวาถูกกดลงเมนูจะแสดงออกมาถ้าปุ่มไม่ได้ถูกกดเมนูจะหายไปหรือถูกลบไปจากจอภาพและขณะที่เมนูแสดงอยู่นั้นจะมีแถบแสงสว่างอยู่ที่เมนูที่คอยเลื่อนไปเรื่อย ๆ ตามเคอร์เซอร์ที่เคลื่อนไหวอยู่

2.2 สรุปสัญลักษณ์ของสเตทไดอะแกรมกับตัวดำเนินการ

จากรูปที่ 3.15 แสดงโครงสร้างของสเตทไดอะแกรม ซึ่งเป็นการสรุปสัญลักษณ์ที่กล่าวมาข้างต้นของหัวข้อที่ (1) และ (2) สังเกตว่าชื่อของสถานะถูกพิมพ์ด้วยตัวหนาภายในสัญลักษณ์ ชื่อของเหตุการณ์ถูกแสดงอยู่บนเส้นของการส่งผ่านจากนั้นตามด้วยแอตทริบิวต์ บางครั้งอาจมีมากกว่าหนึ่งแอตทริบิวต์และตามด้วยเงื่อนไขที่แสดงอยู่ในเครื่องหมายก้ามปูและการกระทำจะอยู่หลังเครื่องหมาย "/" ส่วนกิจกรรมต่าง ๆ ถูกกำหนดภายในสถานะพร้อมกับชื่อของกิจกรรมหรือคำอธิบายกิจกรรม ทั้งหมดเป็นสัญลักษณ์สเตทไดอะแกรมที่แสดงอยู่ในแบบจำลองไดนามิค



รูปที่ 3.15 แสดงโครงสร้างของสเตทไดอะแกรม

3. กลุ่มของสเตทไดอะแกรม (nested state diagrams)

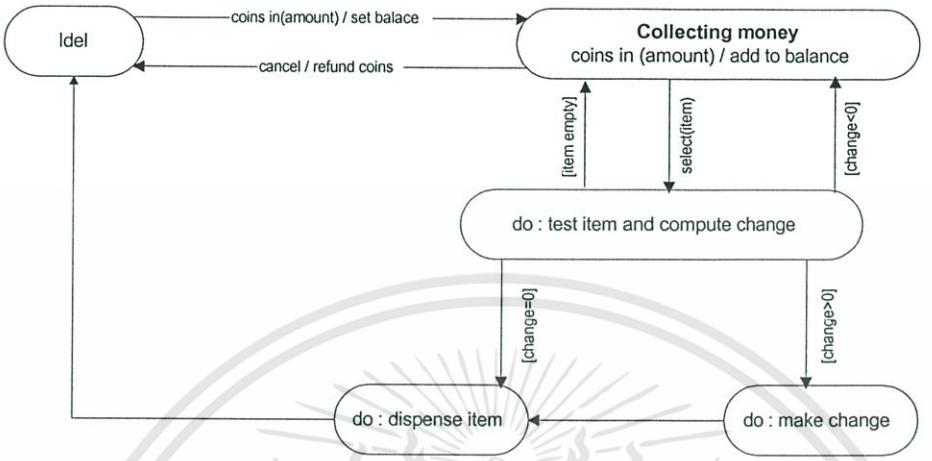
สเตทไดอะแกรมเป็นวิธีหนึ่งที่คอยอธิบายถึงโครงสร้างของระบบให้กับระบบที่มีความซับซ้อนซึ่งคล้ายกับโครงสร้างของวัตถุที่มีทั้งแบบทั่วไป(Generalization) และการประกอบกันของสถานะ (Aggregation / Generalization) คือ ส่วนที่บ่งบอกถึงกลุ่มของกิจกรรมดังนั้นกิจกรรมที่บรรยายในส่วนนี้ จึงอยู่ในระดับบนและถ้าต้องการรายละเอียดเพิ่มเติมจะถูกบรรยายในระดับล่างคล้ายกับการเรียกของกลุ่มการทำงานนั่นเอง การจัดการสถานะและเหตุการณ์ที่เกิดขึ้นแบบทั่วไปมีลักษณะคล้ายกับการสืบทอดคุณสมบัติและฟังก์ชัน ส่วนการประกอบการของสถานะ กล่าวถึงสถานะที่แยกออกจากองค์ประกอบพื้นฐานและมีข้อจำกัดของการติดต่อระหว่างกัน ซึ่งการดำเนินงานจะกระทำพร้อมกับสถานะที่เกี่ยวข้องเท่านั้น เช่น ถ้าสถานะแบบทั่วไปมีการเคลื่อนไหวและมีความเกี่ยวข้องกับสถานะย่อยจะทำให้สถานะย่อยนั้น ๆ ทำงานพร้อมกันไปด้วย เป็นต้น

3.1 Nesting State Diagrams

ภายในสเตทไดอะแกรมประกอบด้วย สถานะ เหตุการณ์และกิจกรรมต่าง ๆ ซึ่งสิ่งเหล่านี้สามารถนำมาเพิ่มเติมรายละเอียดได้ โดยเฉพาะกิจกรรมที่แสดงอยู่ในสเตทไดอะแกรมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มเติมรายละเอียดจะอยู่ที่ระดับล่างของสเตทไดอะแกรม กลุ่มของกิจกรรมมีลักษณะคล้ายกับกระบวนการทำงานย่อย (Subroutine)ที่มีทั้งการส่งข้อมูลเข้าและออก



รูปที่ 3.16 แสดงสเตทไดอะแกรมของเครื่องขายสินค้าประเภทหยอดเหรียญ
ที่มา[28]

จากรูปที่ 3.16 แสดงสเตทไดอะแกรมระดับบนของเครื่องหยอดเหรียญ ซึ่งภายในไดอะแกรมประกอบด้วยกิจกรรม “dispense item” และเหตุการณ์ “select (item)” โดยจะนำสิ่งที่แสดงอยู่ในสเตทไดอะแกรมมาเพิ่มเติมรายละเอียด ในเหตุการณ์หยอดเหรียญถูกแสดงในสถานะ “Collection state” สถานะที่แสดงตามรูปที่ 3.17 เป็นการกำหนดเพียงสถานะเดียว ดังนั้น การส่งผ่านค่าจากสถานะที่ไม่มีชื่อของสถานะที่แสดงกิจกรรม “do : dispense item” ไปสู่สถานะ Idle จะไม่แสดงข้อความของเหตุการณ์แต่สิ่งนี้ไม่จำเป็นเพราะหลังจากเสร็จกิจกรรมดังกล่าวการส่งผ่านจะทราบทันทีว่าต้องมีสถานะ Idle เพื่อให้ทำให้กิจกรรมสมบูรณ์

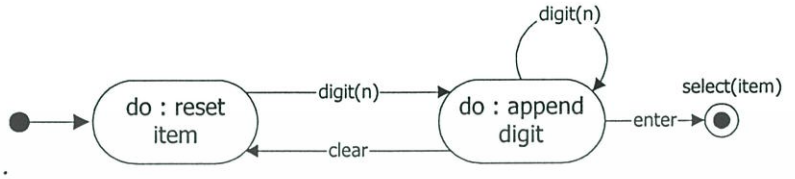
จากรูปที่ 3.17 แสดงไดอะแกรมย่อยของกิจกรรม “dispense item”ที่ต่อเนื่องมาจาก รูปที่ 3.16 ซึ่งลำดับการทำงานที่แสดงนั้นคือการทำงานระดับล่างที่ถูกกระจายออกมา เพราะจากรูปที่ 3.16 นั้นเป็นเพียงการแสดงในระดับบนเท่านั้น

รูปที่ 3.17 แสดงกิจกรรมภายในสถานะ dispense item ของเครื่องหยอดเหรียญ
ที่มา[28]

การขยายเหตุการณ์ของสเตทไดอะแกรมคือการเพิ่มเติมรายละเอียดให้กับ สถานะไดอะแกรม รูปที่ 3.18 แสดงเหตุการณ์ “Select item” ที่ต่อจากรูปที่ 3.16 ซึ่งภายในเหตุการณ์สามารถขยายไปสู่สเตทไดอะแกรมย่อยได้และเป็นเพียงการแสดงไดอะแกรมในระดับล่างของการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กดเลือกรายการของผู้ใช้ ณ เวลาหนึ่งเพื่อเปลี่ยนไปสถานะถัดไป สถานะนี้สามารถถูกเปลี่ยนบนแปลงได้คืออาจยกเลิกรายการแล้วเริ่มต้นใหม่หรือกดหมายเลขรายการซ้ำอีกครั้งหรือกดตอบตกลงเพื่อให้กระบวนการสิ้นสุด (เครื่องหมายแสดงจุดสิ้นสุด) และทั้งหมดที่กล่าวมาคือกระจายเหตุการณ์ Select item ที่แสดงอยู่ในระดับบน



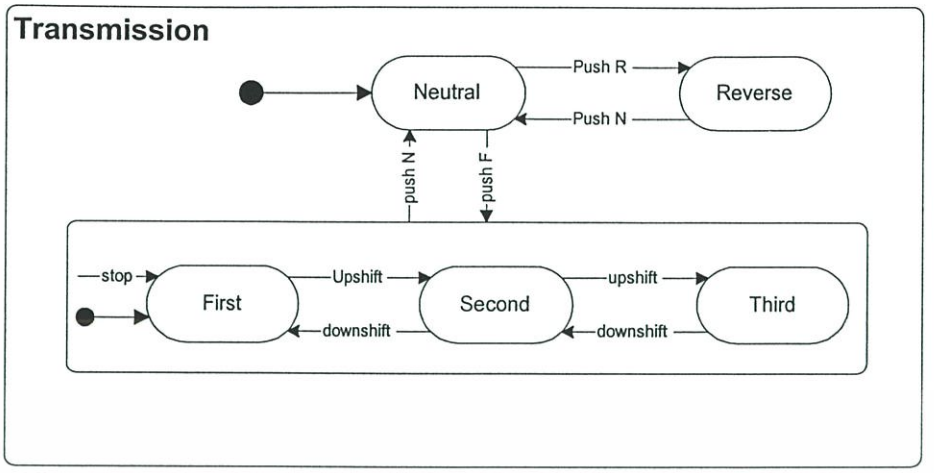
รูปที่ 3.18 แสดงสเตทไดอะแกรมของ Transition Select (item) ในเครื่องหยอดเหรียญ
ที่มา[28]

3.2 สถานะแบบทั่วไป (State Generalization)

จากที่กล่าวมาสังเกตว่าสเตทไดอะแกรมที่แสดงนั้นเป็นเพียงการแสดงสถานะของวัตถุซึ่งเป็นเพียงกลุ่มของสถานะที่แสดงในวัตถุหนึ่งวัตถุใดเท่านั้น ดังนั้นสเตทไดอะแกรมโดยทั่วไปที่ปรากฏอยู่เป็นเพียงการแสดงระดับบนของสถานะเท่านั้นและไม่ได้บ่งบอกถึงความเกี่ยวข้องของสถานะที่เกิดขึ้นภายในวัตถุนั้น

บางครั้งสถานะอาจมีสถานะย่อยซ่อนอยู่ภายในซึ่งคอยรับการกระทำ (Transaction) ที่ส่งมาจากสถานะหลัก (superstates) เหมือนกับคลาสที่มีคลาสย่อยคอยสืบทอดคุณสมบัติต่าง ๆ เช่น แอตทริบิวต์และฟังก์ชันที่สืบทอดมาจากคลาสหลัก เป็นต้น ซึ่งทุกการติดต่อหรือทุกการกระทำ (action) ของสถานะจะถูกส่งผ่านจากสถานะหลักไปสู่สถานะย่อยจนครบทุก ๆ สถานะย่อยที่เกี่ยวข้อง จากรูปที่ 3.19 แสดงสเตทไดอะแกรมของเกียร์รถยนต์ การติดต่อหรือการกระทำที่เกิดขึ้นภายในสถานะของเกียร์รถยนต์มีอยู่ 3 สถานะ คือ เดินหน้า (Forward) ถอยหลัง (Reverse) และเกียร์ว่าง (Neutral) ถ้าอยู่ที่สถานะเดินหน้า ซึ่งภายในประกอบด้วย 3 สถานะย่อย คือ เกียร์หนึ่ง สอง และสาม สัญลักษณ์ที่ใช้แสดงเป็นกรอบขนาดใหญ่และสถานะย่อยแสดงอยู่ภายในกรอบของสถานะหลักพร้อมกับพฤติกรรมต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แสดงสเตทไดอะแกรมของการเข้าเกียร์รถยนต์ในรูปของ Generalization **ที่มา[28]**

การทำงานของสถานะย่อยเป็นสิ่งที่ได้รับมาจากสถานะหลัก กล่าวคือ เมื่อสถานะของเกียร์ถูกเลือกไปยังสถานะเดินหน้ามีผลทำให้สถานะย่อยเกิดการกระตุ้นและทำงานตามลำดับของสถานะย่อยนั้น ซึ่งสถานะย่อยแรก (เกียร์หนึ่ง) เป็นสถานะที่ถูกเริ่มต้นเสมอและค่อย ๆ เปลี่ยนสถานะไปเรื่อย ๆ เมื่อมีความเร็วเพิ่มขึ้น ในทางตรงกันข้ามถ้าความเร็วลดลง สถานะจะถูกเปลี่ยนให้ค่อย ๆ ลดลงเช่นกัน

4. Concurrency

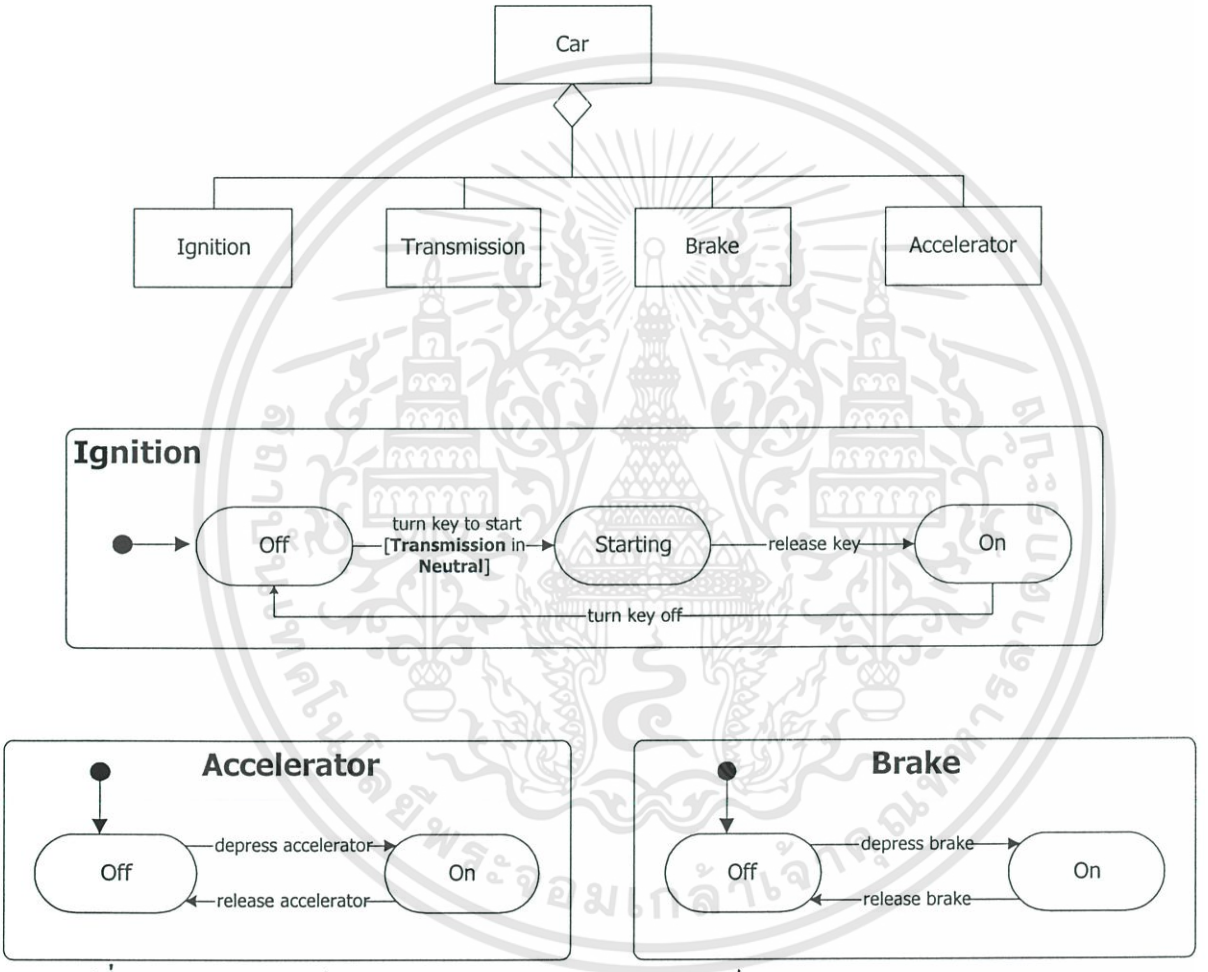
4.1 Aggregation Concurrency

แบบจำลองไดนามิคอธิบายถึงกลุ่มของสเตทไดอะแกรมของวัตถุที่เกิดขึ้น ซึ่งระบบที่เกิดขึ้นมีวัตถุที่เกี่ยวข้องกันมากจึงไม่สามารถแยกออกจากกันได้ และสถานะที่เกิดกับวัตถุแต่ละตัวย่อมสามารถเปลี่ยนสถานะได้ตามอิสระของตัวมันเอง จึงทำให้สเตทไดอะแกรมของสถานะหนึ่ง ๆ ไม่สามารถแทนสถานะทั้งหมดของวัตถุที่มีอยู่ได้ ดังนั้นควรมีสเตทไดอะแกรมให้กับทุกวัตถุที่เกี่ยวข้อง

กลุ่มของสเตทไดอะแกรมที่ประกอบเข้าด้วยกันจะทำงานไปพร้อม ๆ กัน ซึ่งกลุ่มเหล่านี้มีความสัมพันธ์ร่วมกัน คือ การทำงานที่เกิดขึ้นของแต่ละสเตทไดอะแกรมมีการติดต่อระหว่างกัน เช่น กลุ่มแรกส่งเหตุการณ์เข้าไปยังกลุ่มที่ 2 หรือกลุ่มอื่น ๆ ที่เกี่ยวข้องส่งเข้ามากระตุ้นให้กลุ่มถัดไปดำเนินงาน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.20 วัตถุของรถยนต์ประกอบด้วยส่วนประกอบต่าง ๆ เช่น วัตถุของกระบวนการเผาไหม้ของเครื่องยนต์ วัตถุของการเคลื่อนที่หรือเกียร์ วัตถุของคันเร่ง และวัตถุของเบรก เป็นต้น ซึ่งสิ่งเหล่านี้ล้วนเป็นส่วนประกอบและมีความสัมพันธ์ต่อกัน และกลุ่มของสเตทไดอะแกรมเหล่านี้มีการทำงานขนานกันและอิสระต่อกัน กล่าวคือ ถ้าสถานะของวัตถุการเผาไหม้ของห้องเครื่องทำงาน สถานะที่อยู่ในวัตถุของการเคลื่อนที่ต้องอยู่ ณ สถานะของเกียร์ว่างแต่ถ้าสถานะภายในวัตถุการเผาไหม้ของห้องเครื่องหยุดการทำงานมีผลทำให้ทุกสเตทไดอะแกรมหยุดการทำงานด้วย



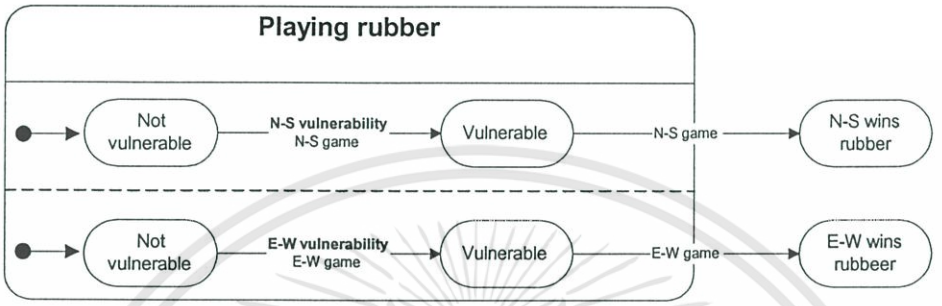
รูปที่ 3.20 แสดงสเตทไดอะแกรมของส่วนประกอบรถยนต์
ที่มา[28]

4.2 Concurrency with an Object

สเตทไดอะแกรมเป็นสิ่งที่แสดงถึงสถานะของวัตถุแต่ละวัตถุที่มีอยู่ในระบบ และบางครั้งวัตถุนั้นอาจมีกลุ่มของสถานะย่อยมากกว่าหนึ่งกลุ่ม จึงทำให้ภายในสเตทไดอะแกรมของวัตถุนั้นเกิดการทำงานแบบขนานขึ้น ซึ่งกลุ่มของสถานะนี้จะไม่มีอิสระต่อกัน กล่าวคือลักษณะการทำงานของเหตุการณ์ที่เข้ามากระทบนี้เป็นเหตุการณ์เกี่ยวข้องกันและมีผลให้กลุ่มของสถานะมีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานพร้อมกันภายในวัตถุเดียวกัน สัญลักษณ์ที่ใช้แสดงมีลักษณะเป็นเส้นประอยู่ระหว่างวัตถุ แสดงดังรูปที่ 3.21 จากรูปวัตถุของการเล่นเกมกลิ้งล้อยางซึ่งภายในวัตถุนี้มีกลุ่มของสถานะอยู่ 2 กลุ่มคือ ขาว กับ ดำ และทั้งสองกลุ่มรับเหตุการณ์มาพร้อมกัน เพื่อทำการแข่งขันกันว่าฝ่ายใดจะเป็นผู้ชนะ

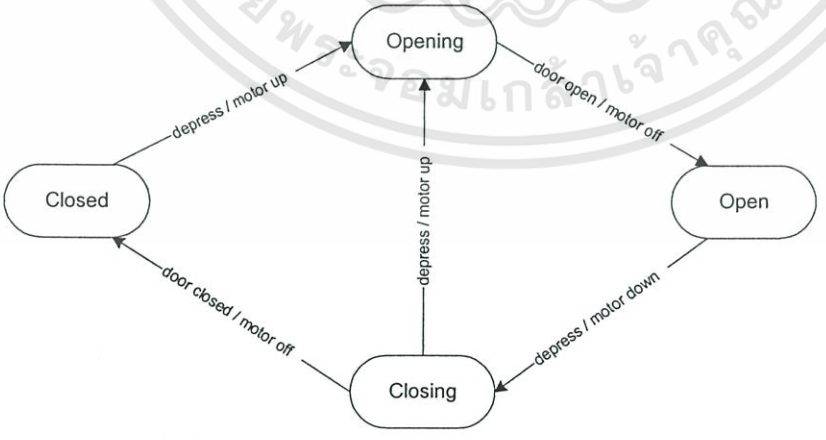


รูปที่ 3.21 แสดงสเตทไดอะแกรมแบบขนานของเกมข้ามสะพาน

5. Advanced dynamic modeling concept

5.1 Entry and Exit Action

การกระทำที่ถูกส่งผ่านสามารถแสดงออกมาอยู่ในรูปของไดอะแกรมได้ เพื่อให้การกระทำที่เกิดขึ้นกับสถานะเห็นได้ชัดเจนขึ้น ส่วนสัญลักษณ์ที่ใช้มีลักษณะคล้ายกับสเตทไดอะแกรม ดังตัวอย่างในรูปที่ 3.22 ไดอะแกรมแสดงการเปิดปิดประตูโรงรถ ซึ่งใช้การกดปุ่มควบคุมการเปิดปิดของประตู ดังนั้นก่อนที่ประตูจะเปิดออกควรอยู่ในตำแหน่งของการปิดก่อน ส่วนการกระทำจะมีผลกับการเปิดปิดมอเตอร์ให้ทำงานตามปุ่มที่กดเพื่อให้ประตูเปลี่ยนสถานะของการทำงาน

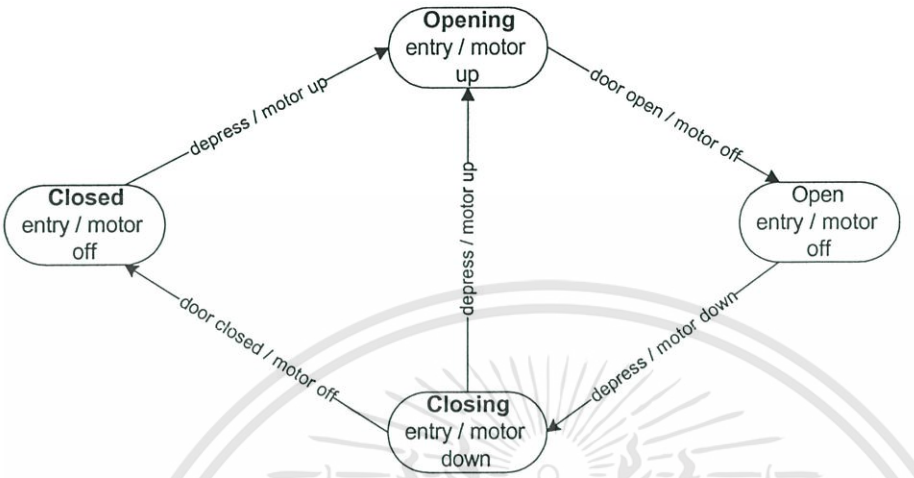


รูปที่ 3.22 แสดงการส่งผ่านการกระทำ

จากรูป 3.23 คล้ายกับรูป 3.22 ต่างกันที่เพิ่มรายละเอียดของการกระทำเข้าและออก ซึ่งการกระทำเข้าใช้คำว่า “entry” และตามด้วยเครื่องหมาย “/” จากนั้นเป็นการทำงานของการกระทำที่ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าไป ส่วนการกระทำออกใช้คำว่า “exit” แทน ส่วนขยายต่าง ๆ มีลักษณะคล้ายกับการกระทำเข้าทุกประการ การทำงานของการกระทำจะรอให้การกระทำก่อนหน้าเสร็จก่อนและอยู่ในสภาวะพร้อมจึงค่อยทำการกระทำถัดไป



รูปที่ 3.23 แสดงลำดับการทำงานของการกระทำภายในแต่ละสถานะ

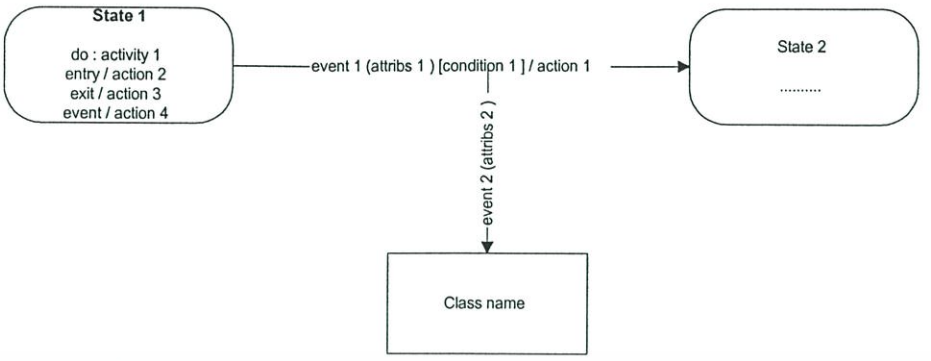
ถ้ามีการปฏิบัติงานหลายอย่างการทำงานจะทำเรียงลำดับกัน โดยเริ่มต้นจากการกระทำเข้าและเริ่มทำกิจกรรมต่าง ๆ จนกระทั่งออกจากการกระทำนั้น จึงถือว่าเป็นอันเสร็จสิ้นกิจกรรมนั้นโดยสมบูรณ์ ซึ่งระหว่างทำกิจกรรมอยู่ถ้ามีเหตุการณ์ใดมาขัดจังหวะก็สามารถทำให้ออกจากการกระทำนั้นได้ทันที

การกระทำเข้าและออกเป็นไดอะแกรมที่ใช้เฉพาะสเตทไดอะแกรมเพราะสามารถอธิบายถึงเหตุการณ์และกิจกรรมที่เกิดขึ้นกับสถานะหนึ่งสถานะใดเท่านั้น จึงทำให้สามารถติดตามการเปลี่ยนแปลงก่อนหลังที่เกิดกิจกรรมได้อย่างชัดเจน

5.2 <u>การกระทำภายใน (Internal Actions)</u>

เหตุการณ์ที่ทำให้การกระทำดำเนินการภายในสถานะไม่ได้เป็นตัวที่ทำให้สถานะนั้น ๆ เปลี่ยนไป แต่เป็นเพียงสิ่งที่คอยบอกเหตุการณ์ที่เกิดขึ้นภายในสถานะเท่านั้น ซึ่งการเขียนรายละเอียดของการกระทำภายในสถานะ มีลักษณะดังนี้ การกระทำเข้าใช้คำว่า “entry” และตามด้วยเครื่องหมาย “/” จากนั้นจบด้วยรายละเอียดของกิจกรรม ส่วนการกระทำออกใช้คำว่า “exit” และถ้อยแถลงจากนั้นเหมือนกับการกระทำเข้าทุกประการ จากรูปที่ 3.24 แสดงรายละเอียดของการเขียนการกระทำที่เกิดขึ้นภายในสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 แสดงการเขียนการกระทำภายในสเตทไดอะแกรม

5.3 Automatic Transition

บ่อยครั้งการทำงานของสถานะถูกกระทำในลักษณะเรียงตามลำดับเมื่อกิจกรรมเสร็จสิ้นก็จะเปลี่ยนไปสถานะถัดไปเพื่อกระตุ้นให้ทำงานทันที ซึ่งลักษณะ Automatic Transition นี้เป็นกิจกรรมที่เกิดจากภายใน โดยมีเงื่อนไขคอยตรวจสอบสังเกตจากรูปที่ 3.16 พบว่า กิจกรรมของ "do: test item and compute change" มีเงื่อนไขคอยควบคุมถึง 4 เงื่อนไข คือ ถ้าเงื่อนไขใดถูกต้องก็ไปทำกับเงื่อนไขนั้นทันที จึงไม่จำเป็นต้องมีชื่อของเหตุการณ์คอยกำกับอยู่บนลูกศร เพราะการทำงานที่เรียงลำดับนี้เป็นเหมือนกับเป็นการบังคับทิศทางการทำงานไว้แล้ว ซึ่งเรียกการส่งผ่านนี้ว่า "Lambda Transitions"

5.4 การส่งเหตุการณ์ (Sending Events)

การส่งผ่านเหตุการณ์ที่เกิดจากสถานะก่อนหน้าไปยังสถานะถัดไปจะถูกแสดงอยู่บนเส้นลูกศร ซึ่งชื่อของเหตุการณ์จะถูกแสดงด้วยตัวอักษรเอียงและแอตทริบิวต์ถูกแสดงอยู่ภายในเครื่องหมายวงเล็บดังแสดงในรูปที่ 3.24

5.5 Synchronization of Concurrent Activities

การทำงานของกิจกรรมที่เกิดขึ้นภายในวัตถุบางครั้งอาจมีกิจกรรมที่มีความสำคัญพร้อมกันอยู่ในวัตถุเดียวกันและกิจกรรมทั้งสองต้องทำให้เสร็จสมบูรณ์ก่อนที่จะเปลี่ยนสถานะไปยังสถานะถัดไปของวัตถุใหม่ ดังนั้นภายในวัตถุนั้นจึงจำเป็นต้องแสดงกิจกรรมเหล่านี้ระหว่างเส้นปะภายในวัตถุเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 แบบจำลองฟังก์ชัน (Functional Model)

แบบจำลองฟังก์ชันกล่าวถึงผลลัพธ์ของการทำงานมากกว่าที่จะกล่าวถึงเรื่องของวิธีการและเวลาในการคำนวณผลลัพธ์ที่เกิดขึ้น ซึ่งหมายความว่าฟังก์ชันและการกระทำที่แสดงอยู่ในแบบจำลองฟังก์ชันเป็นผลต่อเนื่องมาจากการวิเคราะห์ของแบบจำลองวัตถุและแบบจำลองไดนามิก ลักษณะการทำงานของแบบจำลองฟังก์ชันมีผลกระทบโดยตรงกับโปรแกรม ถึงแม้ว่าแบบจำลองวัตถุจะให้ความสนใจกับปัญหาของระบบก็ตาม ส่วนฐานข้อมูลของโมเดลจะแสดงเพียงแค่การจัดการข้อมูลและจัดเก็บเท่านั้น

1. แผนภูมิการไหลของข้อมูล (Data Flow Diagrams)

แผนภูมิการไหลของข้อมูลเป็นแผนภูมิที่แสดงถึงการไหลของข้อมูลที่เกิดขึ้นภายในระบบและภายในประกอบด้วย ข้อมูลเข้า ข้อมูลออก และคลังข้อมูล ซึ่งสิ่งต่าง ๆ เหล่านี้จะประกอบกันอยู่ภายในแผนภูมิของระบบ การไหลของข้อมูลเริ่มต้นจากวัตถุต้นทางไปยังกระบวนการปลายทางของวัตถุอื่น ๆ ที่เชื่อมระหว่างกันและมีเส้นทางของข้อมูลเป็นตัวเชื่อมระหว่างกระบวนการกับคลังข้อมูล สัญลักษณ์ที่แสดงในแผนภูมิการไหลของข้อมูลประกอบไปด้วยสัญลักษณ์ต่าง ๆ เช่น กระบวนการแสดงการส่งผ่านข้อมูล เส้นของข้อมูลแสดงถึงการไหลของข้อมูล ออบเจ็กต์แอคเตอร์ (Object Actor) เป็นส่วนที่คอยบริโภคหรือใช้ข้อมูล และคลังข้อมูลเป็นส่วนที่คอยเก็บข้อมูล ซึ่งรายละเอียดของสัญลักษณ์ต่าง ๆ มีดังต่อไปนี้

1.1 กระบวนการ(Processes)

การทำงานของกระบวนการ คือ การส่งผ่านข้อมูลหรือคำนวณข้อมูลที่ได้มาจากแหล่งข้อมูลภายนอก เช่น คลังข้อมูล กระบวนการ หรือ วัตถุอื่น ๆ ที่สัมพันธ์กัน เป็นต้น ซึ่งแหล่งเหล่านี้จะเป็นส่วนที่คอยกระตุ้นให้กระบวนการเกิดกิจกรรมหรือการทำงานต่าง ๆ ขึ้น ลักษณะของการทำงานที่แสดงในแผนภูมิการไหลของข้อมูลสามารถแสดงออกเป็น 2 ระดับ คือ ระดับสูงและระดับล่าง ซึ่งกระบวนการระดับสูงที่แสดงนั้นคล้ายกับเป็นการแสดงภาพรวมของกระบวนการภายในระบบ ถ้าย้อนกลับไปพิจารณาแผนภูมิการไหลของข้อมูลลักษณะเชิงโครงสร้างพบว่ามีส่วนเหมือนกับ Context Diagram ที่แสดงกระบวนการระดับสูงรวมไว้ที่เดียวกันก่อนแล้วค่อยกระจายกระบวนการที่เกี่ยวข้องในระดับถัดไป ส่วนกระบวนการระดับล่าง คือ การกระจายกระบวนการจากกระบวนการระดับสูงไปสู่กระบวนการย่อยที่เกี่ยวข้อง การกระจายนี้จะกระทำจนกระทั่งกระบวนการย่อยเหล่านั้นไม่สามารถกระจายต่อไปได้อีก สัญลักษณของกระบวนการมีลักษณะเป็นรูปวงรีและภายในประกอบด้วยคำอธิบายของกระบวนการ แต่ละกระบวนการมีลูกศรเข้าและออกที่แสดงถึงอินพุตและเอาต์พุตของข้อมูลเป็นจำนวนแน่นอนและแต่ละลูกศรมีค่าของข้อมูลตามชนิดที่กำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ดาต้าโฟลว์ (Data Flows)

ดาต้าโฟลว์เชื่อมต่อระหว่างผลลัพธ์ของวัตถุหรือกระบวนการกับอินพุตจาก วัตถุหรือกระบวนการอื่น ดาต้าโฟลว์แทนค่าของข้อมูลที่เกิดจากการคำนวณ ค่าของข้อมูลไม่ได้เปลี่ยนแปลงไปเพราะดาต้าโฟลว์ ซึ่งถูกแทนด้วยลูกศรที่อยู่ระหว่างผู้สร้างและผู้ใช้ค่าของข้อมูล บนลูกศรจะมีคำอธิบายเกี่ยวกับค่าของข้อมูลซึ่งปกติจะเขียนเป็นชื่อหรือชนิดของข้อมูล ค่าข้อมูลที่เหมือนกันสามารถส่งไปได้หลายที่โดยใช้สัญลักษณ์ลักษณะคล้ายส้อมที่มีหลายหัวลูกศรรวมกัน และลูกศรผลลัพธ์ไม่แสดงเขียนชื่อกำกับเพราะใช้แทนค่าอินพุตอันเดียวกัน ดังแสดงในรูปที่ 3.26 บางครั้งค่าของข้อมูลแบบอะกรีเกชันแตกออกเป็นส่วนประกอบซึ่งแต่ละส่วนสามารถถูกส่งไปยังกระบวนการที่แตกต่างกัน

ในกรณีนี้จะแสดงด้วยสัญลักษณ์ส้อมซึ่งแต่ละลูกศรจะเขียนชื่อของส่วนประกอบกำกับส่วนประกอบที่รวมกันเป็นค่าอะกรีเกชัน คือ ค่าที่อยู่ด้านตรงข้าม (ดังแสดงในรูปที่ 3.26) แต่ละดาต้าโฟลว์จะแทนค่าซึ่งได้จากการคำนวณที่จุดใดจุดหนึ่ง ซึ่งในความเป็นจริงค่านี้ไม่จำเป็นต้องมีความสำคัญ โฟลว์ที่อยู่ตรงส่วนขอบของแผนภูมิการไหลของข้อมูล คือ อินพุตและเอาต์พุตโฟลว์เหล่านี้อาจไม่ถูกเชื่อมต่อ (ถ้าไดอะแกรมเป็นเพียงส่วนหนึ่งของระบบ) หรืออาจจะเชื่อมต่อกับวัตถุก็ได้

1.3 ผู้กระทำ (Actors)

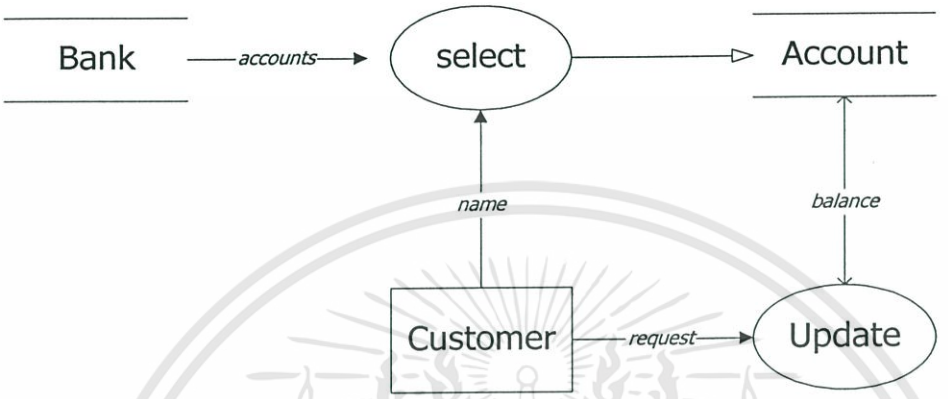
ผู้กระทำ คือ วัตถุที่ทำหน้าที่ขับเคลื่อนดาต้าโฟลว์โดยสร้างและใช้ค่าข้อมูล ผู้กระทำติดต่อกับอินพุตและเอาต์พุตของดาต้าโฟลว์ บางครั้งจึงเรียกผู้กระทำนี่ว่าเทอร์มิเนเตอร์ (Terminators) การกระทำของผู้กระทำอยู่นอกขอบเขตของแผนภูมิการไหลของข้อมูลแต่เป็นส่วนหนึ่งของแบบจำลองไดนามิค ผู้กระทำถูกแทนด้วยรูปสามเหลี่ยมเพื่อแสดงถึงวัตถุ ลูกศรระหว่างผู้กระทำและไดอะแกรม คือ อินพุตและเอาต์พุตของไดอะแกรม

1.4 ดาต้าสโตร์ (Data Stores)

ดาต้าสโตร์ คือ วัตถุที่ไม่มีการกระทำภายในแผนภูมิการไหลของข้อมูลแต่เก็บข้อมูลสำหรับการกระทำล่าสุด แหล่งเก็บข้อมูลไม่เหมือนกับผู้กระทำเพราะแหล่งเก็บข้อมูลไม่ได้สร้างฟังก์ชันใด ๆ บนแหล่งเก็บข้อมูลเพียงแต่ตอบรับที่จะเก็บข้อมูล แหล่งเก็บข้อมูลเขียนแทนด้วยเส้นคู่ขนานที่เขียนชื่อแหล่งเก็บข้อมูลไว้ภายใน ลูกศรเข้าไปที่แหล่งเก็บข้อมูลระบุข้อมูลข่าวสารหรือโอเปอเรชันซึ่งปรับปรุงแก้ไขข้อมูลที่เก็บไว้ในแหล่งข้อมูล ได้แก่การเพิ่ม การเปลี่ยนแปลงค่า หรือการลบค่าออก ลูกศรออกจากแหล่งเก็บข้อมูลระบุข้อมูลข่าวสารที่ถูกดึงออกมาจากแหล่งเก็บข้อมูลซึ่งอาจจะเป็นค่าทั้งหมดหรือค่าเพียงบางส่วน โครงสร้างของวัตถุจะต้องอธิบายในแบบจำลองวัตถุรวมทั้งต้องอธิบายฟังก์ชันการแก้ไขและการกระทำทั้งผู้กระทำและแหล่งเก็บข้อมูลคือวัตถุ แต่ที่กำหนดให้แตกต่างกันก็เพราะพฤติกรรมและการใช้มีความแตกต่างกันถึงแม้ว่าในภาษาเชิงวัตถุจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างเป็นวัตถุเหมือนกันหรือแหล่งเก็บข้อมูลอาจจะสร้างเป็นแฟ้มข้อมูลและผู้กระทำเป็นอุปกรณ์ภายนอกดาต้าโฟลว์ ดาต้าโฟลว์จึงเป็นวัตถุด้วย ถึงแม้ว่าในหลายกรณีที่ดาต้าโฟลว์เป็นค่าของข้อมูล เช่น เลขจำนวนเต็ม (ในภาษาเชิงวัตถุและค่าแท้ ๆ มักจะสร้างเหมือนกัน) มุมมองที่แตกต่างกันระหว่างวัตถุ คือ วัตถุมีข้อมูลได้ค่าเดียวแต่แหล่งเก็บข้อมูลมีข้อมูลได้มากกว่าหนึ่ง



รูปที่ 3.25 แสดงการเลือกข้อมูลจากบัญชีลูกค้า

จากรูปที่ 3.25 แสดงการเลือกข้อมูลจากบัญชีลูกค้าโดยนำชื่อของลูกค้าไปเลือกจากบัญชีของธนาคารทำให้ได้ผลลัพธ์เป็นวัตถุบัญชีที่ใช้เป็นแหล่งเก็บข้อมูลสำหรับฟังก์ชันแก้ไขข้อมูล ดาต้าโฟลว์ซึ่งสร้างวัตถุและจะใช้เป็นเป้าหมายของฟังก์ชันอื่นระบุโดยใช้รูปสามเหลี่ยมกลวงที่ปลายของดาต้าโฟลว์ตรงข้ามฟังก์ชันการแก้ไขซึ่งปรับยอดบัญชีในวัตถุบัญชีระบุโดยใช้หัวลูกศรเล็ก ๆ รูปสามเหลี่ยมกลวงที่ระบุค่าผลลัพธ์ของดาต้าโฟลว์เป็นวัตถุโดยปกติแล้วคือแหล่งเก็บข้อมูล

1.5 คอนโทรลโฟลว์ (Control Flows)

แผนภูมิการไหลของข้อมูลแสดงเส้นทางการคำนวณที่เป็นไปได้สำหรับค่าข้อมูล โดยไม่แสดงว่าเส้นทางใดถูกกระทำและลำดับของการกระทำเป็นอย่างไร การตัดสินใจและการเรียงลำดับคือ การควบคุมซึ่งเป็นส่วนหนึ่งของแบบจำลองไดนามิค การตัดสินใจนั้นมีผลกับหนึ่งฟังก์ชันหรือมากกว่าหนึ่งฟังก์ชันที่ถูกกระทำจึงไม่ใช่เป็นเพียงการให้ค่ากับฟังก์ชันเท่านั้น ถึงแม้ว่าการตัดสินใจเลือกฟังก์ชันนี้จะไม่ได้มีค่าอินพุตแต่ในบางครั้งก็มีประโยชน์ที่จะรวมการตัดสินใจเลือกฟังก์ชันนี้ไว้ในแบบจำลองฟังก์ชันเพื่อว่าได้ไม่ลืมและยังเป็นการแสดงข้อมูลที่ขึ้นต่อกันอีกด้วย วิธีทำคือรวมการควบคุมดาต้าโฟลว์นี้เข้าไปไว้ในดาต้าโฟลว์ไดอะแกรมด้วย การควบคุมดาต้าโฟลว์คือค่าบูลีน (Boolean) ซึ่งจะมีผลกับกระบวนการว่าว่ากระบวนการใดจะถูกกระทำ ตัวของการควบคุมดาต้าโฟลว์ไม่ใช่เป็นค่าอินพุตของกระบวนการ การควบคุมดาต้าโฟลว์แทนด้วยเส้นลูกศรประจากกระบวนการที่สร้างค่าบูลีนไปยังกระบวนการที่ถูกควบคุม การควบคุมดาต้าโฟลว์นั้นมีประโยชน์แต่จะซ้ำกับข้อมูลข่าวสารในแบบจำลองไดนามิค ดังนั้นจึงควรใช้เท่าที่จำเป็นเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบวัตถุ (Object Design) [28],[40]

การออกแบบวัตถุ คือ ขั้นตอนที่ทำต่อจากการวิเคราะห์ เพราะ การวิเคราะห์นั้นนักวิเคราะห์จะนำเสนอระบบสารสนเทศในเชิงตรรกะมากกว่าเชิงกายภาพ ซึ่งจะต้องนำแนวคิดเหล่านั้นไปสู่ขั้นตอนของการออกแบบเพื่อให้ได้ระบบสารสนเทศที่สามารถใช้ได้จริง วิธีการออกแบบเชิงวัตถุนี้ มีอยู่ด้วยกันหลายวิธี เช่น OMT[28], COAD/YOURDON[40], Jacobson[25] และ Gready Booch[13] เป็นต้น โดยแต่ละวิธีจะมีเทคนิคและวิธีการแตกต่างกันออกไปแต่สุดท้ายปลายทางของกระบวนการทั้งหลายนั้น ย่อมทำให้ระบบสารสนเทศที่ผ่านกระบวนการต่าง ๆ เหล่านี้ได้ผลลัพธ์ที่เหมือนกัน ในหัวข้อนี้จะนำเสนอวิธีการออกแบบของ OMT กล่าวโดยสรุปดังต่อไปนี้

วิธีการอออกแบบของ OMT นั้นมีขั้นตอนการออกแบบอยู่หลายขั้นตอน ก่อนที่จะย้ายขั้นตอนต่าง ๆ ที่เกิดขึ้นในแบบจำลองวัตถุ แบบจำลองไดนามิค และแบบจำลองฟังก์ชัน ไปสู่การนำเสนอนั้น จะต้องทำให้ระบบอยู่บนพื้นฐานของความเป็นจริงที่ขึ้นกับคอมพิวเตอร์ก่อน การย้ายโมเดลต่าง ๆ ที่เกิดขึ้นจากการวิเคราะห์นั้นเราจะต้องนำโมเดลทั้ง 3 มาพิจารณา กล่าวคือ

แบบจำลองวัตถุที่อยู่ในขั้นตอนการวิเคราะห์นั้น ได้อธิบายสิ่งต่าง ๆ ในรูปของแผนภาพ ที่ประกอบด้วย วัตถุของคลาสระบบ พร้อมกับแอตทริบิวต์และฟังก์ชันต่าง ๆ ที่เกี่ยวข้องในวัตถุนั้น สามารถย้ายแบบจำลองนี้ไปสู่การออกแบบได้โดยตรง แต่บางครั้งการย้ายอาจจะไม่สามารถย้ายได้ครบถ้วนสมบูรณ์ตามที่ต้องการนัก เพราะว่าอาจจะมีวัตถุบางตัวไม่มีความชัดเจนพอที่จะนำไปสู่ขั้นตอนการออกแบบ และบางครั้งอาจมีวัตถุที่เกิดขึ้นใหม่ในระบบก็ได้

แบบจำลองไดนามิค อธิบายถึงสิ่งที่เกิดขึ้นกับระบบ ซึ่งทำให้ทราบว่า มีเหตุการณ์อะไรมากระทบกับระบบเพื่อให้ระบบมีการกระทำบ้าง สิ่งต่าง ๆ ที่เกิดขึ้นในแบบจำลองไดนามิคนี้จะทำให้ทราบถึงโครงสร้างของโปรแกรม เพราะสามารถนำเอาสเตทไดอะแกรมที่เกิดขึ้นมาคอยควบคุมการทำงานของโปรแกรมเพื่อทำให้เกิดความชัดเจนและความแน่นอน

แบบจำลองฟังก์ชัน อธิบายถึงรายละเอียดที่ต้องนำมาเสนอในระบบ โดยแสดงในรูปของดาต้าโฟลว์ซึ่งระหว่างการออกแบบ เราจะต้องเลือกอัลกอริทึมที่เหมาะสมเพื่อมานำเสนอแต่ละโอเปอเรชันที่อยู่ในวัตถุ การเลือกอัลกอริทึมอาจจำเป็นต้องทำแบบกระจายในลักษณะการทำซ้ำของแต่ละกระบวนการ ซึ่งอาจต้องทำถึงระดับของ ADT (Abstract Data type) เหตุที่ต้องทำเช่นนี้เพราะต้องการให้ได้อัลกอริทึมที่มีประสิทธิภาพ ตรงตามความต้องการ และง่ายต่อการทำความเข้าใจ

ขั้นตอนการออกแบบเชิงวัตถุที่กล่าวมาข้างต้น เป็นเพียงแนวคิดเบื้องต้นเท่านั้น ซึ่งมีขั้นตอนการออกแบบดังต่อไปนี้

3.3.1 ขั้นตอนการออกแบบเชิงวัตถุ

ขั้นตอนการออกแบบที่นักออกแบบต้องปฏิบัติมีดังต่อไปนี้

- การรวมโมเดลทั้ง 3 (Combine the three models)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การออกแบบอัลกอริทึมและการอิมพลีเมนต์ (Design algorithms to implement operations)
- การจัดการการเข้าถึงข้อมูลให้มีประสิทธิภาพ (Optimize access paths to data)
- การควบคุมการสำหรับการอิมพลีเมนต์ (Implement of control)
- ปรับปรุงการสืบทอดคุณสมบัติ (Adjust of inheritance)
- ออกแบบความสัมพันธ์ (Design of Associations)
- การแทนวัตถุ (Object Representation)
- การบรรจุคลาสและความสัมพันธ์ไปสู่ระดับกายภาพ (Package classes and associations into Physical)

การออกแบบเชิงวัตถุมีลักษณะการทำซ้ำของกระบวนการ กล่าวคือ ถ้าคุณคิดว่าการออกแบบคือการทำเพียงแค่ระดับของ ADT (Abstract Data Type) กับเพิ่มรายละเอียดและกระจายรายละเอียดไปสู่การออกแบบเพียงอย่างเดียวนั้นไม่ใช่ทั้งหมด เพราะคลาสหรือวัตถุ แอตทริบิวต์ และฟังก์ชันต่าง ๆ ที่ปรากฏในขั้นตอนการวิเคราะห์นั้นอาจจะมี คลาสหรือวัตถุ แอตทริบิวต์ หรือ โอเปอเรชันใหม่เพิ่มเติมขึ้นมาได้ จึงกล่าวได้ว่าเปลี่ยนระบบจากการวิเคราะห์ไปสู่การออกแบบนั้นไม่สามารถที่จะทำให้เสร็จสิ้นในรอบเดียวได้

1. การรวมแบบจำลองทั้ง 3 (Combine the Three Models)

หลังจากเสร็จจากขั้นตอนวิเคราะห์ระบบจะได้ แบบจำลองวัตถุ แบบจำลองไดนามิค แบบจำลองฟังก์ชัน ซึ่งแบบจำลองทั้ง 3 เป็นเพียงเค้าโครงของการออกแบบเท่านั้น ในแบบจำลองวัตถุที่เกิดจากการวิเคราะห์นั้นไม่ได้แสดงฟังก์ชัน ซึ่งสิ่งที่นักออกแบบต้องการคือการกระทำและกิจกรรมที่เกิดขึ้นจากแบบจำลองไดนามิค และกระบวนการของแบบจำลองฟังก์ชันที่แสดงถึงการทำงานของแต่ละคลาสในแบบจำลองวัตถุ สิ่งที่เราต้องทำคือ ควรย้ายโครงสร้างทางตรรกะที่เกิดขึ้นจากโมเดลวิเคราะห์ไปสู่โปรแกรมซึ่งเป็นระดับกายภาพเสียก่อน

ในแต่ละสถานะจะแสดงประวัติชีวิตของแต่ละวัตถุ และ Transition คือส่วนที่คอยเปลี่ยนแปลงสถานะของวัตถุ ซึ่งเราต้องนำสิ่งที่เกิดขึ้นเปลี่ยนไปสู่ฟังก์ชันบนวัตถุนั้น ซึ่งฟังก์ชันที่เกิดขึ้นจะนำไปสู่การเลือกอัลกอริทึม การนำเสนออัลกอริทึมขึ้นอยู่กับสถานะของวัตถุ บางครั้งอาจมี 2 อีเวนท์ที่เหมือนกันทำงานอยู่ในสถานะเดียวกันการนำเสนออัลกอริทึมอาจจะอิมพลีเมนต์มากกว่าหนึ่งสถานะในอัลกอริทึมนั้น ทั้งนี้ขึ้นอยู่กับความสามารถของภาษาที่ใช้ในการอิมพลีเมนต์ด้วย

การส่งเหตุการณ์ของวัตถุนั้นมักจะส่งเป็นคู่ ตัวแรก คือ เหตุการณ์ที่เป็นตัวกระตุ้นของการกระทำ ตัวที่สองเป็นเหตุการณ์ของการส่งค่ากลับหรือคอยแสดงถึงการสิ้นสุดของการกระทำที่เกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นกับวัตถุ ซึ่งในการกระทำที่ส่งมาเป็นคู่สามารถนำไปเป็นตัวควบคุมการทำงานของอัลกอริทึมที่เขียนขึ้นได้ แต่การกระทำที่เกิดขึ้นจะทำกับการส่งผ่านค่าระหว่างวัตถุเท่านั้น

ภายในการกระทำหรือกิจกรรมที่เกิดขึ้นโดย Transition ในแต่ละสเตทไดอะแกรมจะแสดงในรูปของแผนภูมิการไหลของข้อมูลที่อยู่ในแบบจำลองฟังก์ชัน ซึ่งจะอธิบายถึงการทำงานของกระบวนการในแต่ละฟังก์ชันที่เกิดขึ้นกับค่าของข้อมูลที่อยู่ภายในฟังก์ชันซึ่งจะถูกเปลี่ยนแปลงไปตามโฟล์วที่วิ่งผ่าน ดังนั้นนักออกแบบควรจะนำโฟลว์เหล่านี้แปลงไปสู่การทำงานในอัลกอริทึมของโปรแกรมและกระบวนการในดาต้าโฟล์วจะประกอบด้วยฟังก์ชันย่อย ซึ่งในแต่ละฟังก์ชันที่แสดงในดาต้าโฟล์วนั้นเราไม่สามารถที่จะทราบได้ว่าฟังก์ชันนั้นเป็นของหรืออยู่ที่วัตถุใด ดังนั้นการพิจารณาฟังก์ชันที่เกิดขึ้นในวัตถุมีหลักการพิจารณาดังนี้

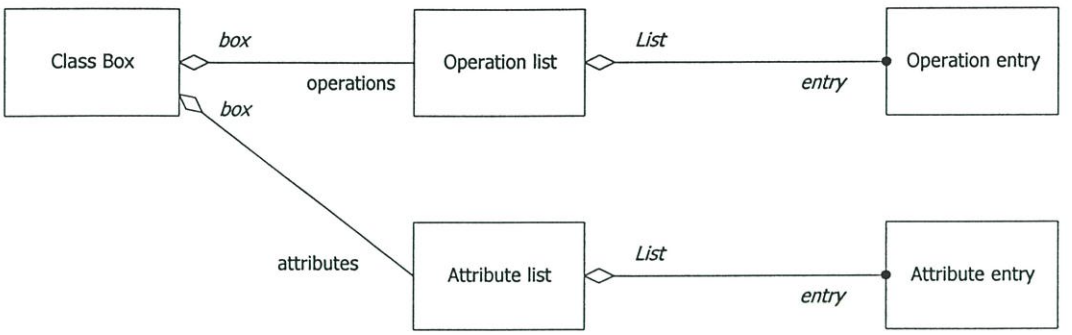
- ถ้ากระบวนการใดส่งค่ามาจากอินพุตโพลว์ อินพุตนั้นจะเป็นของกระบวนการต้นทางที่ส่งออกมา
- ถ้ากระบวนการใดมีทั้งอินพุตและเอาต์พุตโพลว์ที่เหมือนกัน และค่าของเอาต์พุตเป็นการปรับปรุงค่าอินพุตที่อยู่ในกระบวนการ อินพุต-เอาต์พุตนั้นจะเป็นของต้นทาง
- ถ้ากระบวนการใดส่งค่าเอาต์พุตที่เกิดจากหลาย ๆ อินพุต ฟังก์ชันที่เกิดขึ้นจะเป็นคลาสฟังก์ชันบนคลาสเอาต์พุต
- ถ้ากระบวนการใดมีอินพุตที่มาจากหรือเอาต์พุตไปสู่ ดาต้าสโตร์หรือการกระทำ ดังนั้นดาต้าสโตร์หรือการกระทำนั้นจะเป็นกระบวนการต้นทาง

ดังนั้นการนำทั้ง 3 โมเดลมาเขียนอัลกอริทึมควรที่จะพิจารณาในทุก ๆ โมเดลเพื่อให้ได้อัลกอริทึมที่ถูกต้องและมีประสิทธิภาพกับระบบด้วย

2. การออกแบบอัลกอริทึมและการอิมพลีเมนต์ (Design algorithms to implement operations)

การออกแบบอัลกอริทึมของ OMT นั้น ในแบบจำลองฟังก์ชันจะแสดงรายละเอียดคร่าว ๆ ไว้แล้ว เปรียบเสมือนกับการนำเสนออัลกอริทึมเป็นที่เรียบร้อยแล้ว แต่การออกแบบอัลกอริทึมนั้นควรคำนึงถึง ค่าใช้จ่ายของการอิมพลีเมนต์อัลกอริทึม ความเหมาะสมของโครงสร้างข้อมูลที่นำมาใช้ การทำงานของฟังก์ชันภายในคลาสและความเหมาะสมของฟังก์ชันของคลาส เป็นต้น
ดังนั้นการเลือกอัลริทึมผู้ออกแบบจะต้องกระจายฟังก์ชันและแอตทริบิวต์ที่มีอยู่ในแต่ละคลาสออกเพื่อสามารถเห็นการทำงานได้ชัดเจนขึ้น ซึ่งเป็นสิ่งสำคัญ เพื่อแสดงว่าอัลกอริทึมที่อิมพลีเมนต์กับฟังก์ชันนั้นตรงกันและควรกำหนดค่าเริ่มต้นให้กับฟังก์ชันให้ถูกต้องก่อนใช้งาน การกระจายฟังก์ชันและแอตทริบิวต์ของคลาสสามารถแสดงได้ดังรูปที่ 3.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 แสดงองค์ประกอบของคลาสที่ประกอบด้วยฟังก์ชันและแอตทริบิวต์

การพิจารณาอัลกอริทึมในการอิมพลีเมนต์ควรคำนึงถึง ความซับซ้อนในการคำนวณของฟังก์ชัน ความง่ายต่อการทำความเข้าใจของการอิมพลีเมนต์ มีความยืดหยุ่นและสามารถปรับปรุงได้ง่ายกับแบบจำลองวัตถุ เป็นต้น ในระหว่างการออกแบบนั้นสิ่งที่เกิดขึ้นจะเป็นเพียงโครงสร้างทางตรรกะของระบบแต่ส่วนการออกแบบจะเป็นการเลือกโครงสร้างข้อมูลที่นำไปใส่ในอัลกอริทึมของโปรแกรม ซึ่งในการอิมพลีเมนต์โครงสร้างข้อมูลผู้ออกแบบอาจจะต้องเลือกวิธีการนำเสนอในโปรแกรม เช่น อาร์เรย์ ลิสต์ คิว หรือ สแต็ก เป็นต้น หลังจากที่เลือกวิธีการให้กับโปรแกรมแล้ว จำเป็นต้องกำหนดหน้าที่ของฟังก์ชันที่เกิดขึ้นกับแต่ละวัตถุให้ชัดเจน บางครั้งจำเป็นต้องที่จะทำลงไปถึงระดับล่าง (Low-level) ของคลาสนั้น การทำเช่นนี้อาจทำให้เกิดคลาสใหม่ให้กับระบบซึ่งไม่เคยเกิดขึ้นกับขั้นตอนการวิเคราะห์มาก่อนก็อาจเป็นได้ การกำหนดฟังก์ชันของคลาสผู้ออกแบบควรกำหนดกฎเกณฑ์สำหรับฟังก์ชันที่เกิดขึ้นในแต่ละวัตถุซึ่งอาจจะจัดทำในลักษณะของแบบสอบถามให้กับวัตถุแต่ละตัว เพราะในระหว่างการอิมพลีเมนต์แบบสอบถามเหล่านี้จะเป็นประโยชน์มากสำหรับการเขียนโปรแกรมให้กับระบบ

3. การจัดการการเข้าถึงข้อมูลให้มีประสิทธิภาพ (Optimization access paths to data)

พื้นฐานของการออกแบบจำลองมาจากแบบจำลองการวิเคราะห์เพื่อไปสู่การอิมพลีเมนต์ซึ่งแบบจำลองการวิเคราะห์เป็นเพียงการนำเสนอแนวคิดทางตรรกะของระบบ แต่ขณะที่แบบจำลองการออกแบบนำเสนอรายละเอียดของแบบจำลองเพื่อสนับสนุนทางการเข้าถึงข้อมูลในระดับกายภาพได้มากกว่า ดังนั้นระหว่างการออกแบบโดยเฉพาะขั้นตอนนี้นักออกแบบควรนำแบบจำลองที่ได้จากการวิเคราะห์มาปรับปรุงซึ่งขั้นตอนการปรับปรุงนั้นควรทำดังนี้

- ลดความซับซ้อนที่เกิดขึ้นกับการเข้าถึงข้อมูลของคลาสให้น้อยที่สุดเท่าที่จะทำได้
- ปรับปรุงประสิทธิภาพลำดับการทำงานของแบบจำลองฟังก์ชัน (อัลกอริทึม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จัดเก็บข้อมูลหรือแอตทริบิวต์ในแต่ละวัตถุที่มีการเปลี่ยนแปลงบ่อยให้มีความชัดเจนขึ้น

การลดความซับซ้อนของการเข้าถึงข้อมูลในแต่ละคลาสนั้นในบางครั้งความสัมพันธ์ของคลาสที่เกิดขึ้นอาจจะมีความสัมพันธ์ที่ซับซ้อน เช่น ต้องการหาความสามารถพิเศษทางภาษาของพนักงานในบริษัท ซึ่งประกอบด้วย คลาสบริษัท คลาสพนักงาน คลาสความสามารถพิเศษ โดยความสัมพันธ์ที่มีสามารถแสดงได้ดังรูปที่ 3.27 ซึ่งถ้าต้องการหาความสามารถพิเศษทางภาษาญี่ปุ่นของพนักงานในบริษัทจะต้องเกี่ยวข้องกันทั้งสามคลาส ซึ่งถ้ามีการค้นหาจากพนักงานจำนวนน้อยก็จะไม่เกิดปัญหาในเรื่องของเวลา แต่ถ้ามีพนักงานจำนวนมากประมาณ 1000 คนและทุกคนก็มีความสามารถพิเศษทางภาษาด้วยกันทั้งสิ้นอาจต้องทำการค้นหาจากพนักงานทั้งหมดและทำให้การค้นหานั้นมีความล่าช้าขึ้น

รูปที่ 3.27 แสดงความสัมพันธ์แบบลูกโซ่ของคลาสบริษัท พนักงานและความสามารถพิเศษ

ทางแก้ไข คือ ควรยุบคลาสความสามารถพิเศษเสีย แล้วเพิ่มดรรชนี (index) ให้กับคลาสบริษัทจะทำให้การเข้าถึงนั้นง่ายและรวดเร็วขึ้นเพราะถ้าผู้ออกแบบเพิ่มดรรชนีเข้าไปจะทำให้การเข้าถึงนั้นกระชับมากขึ้นดังแสดงในรูปที่ 3.28 การยุบคลาสและความสัมพันธ์ที่เกิดขึ้นนั้นผู้ออกแบบควรที่จะคำนึงถึงความจำเป็นและความเหมาะสมของระบบงานเป็นหลัก โดยอาจจะต้องตรวจสอบจากความต้องการของผู้ใช้และลักษณะการทำงานของฟังก์ชันของคลาสเป็นหลัก

รูปที่ 3.28 แสดงดรรชนีความสามารถพิเศษให้กับคลาสพนักงาน

การปรับปรุงประสิทธิภาพลำดับการทำงานของแบบจำลองฟังก์ชันหรืออัลกอริทึมนั้น คือ การปรับปรุงการทำงานของอัลกอริทึมให้มีลำดับการทำงานที่กระชับ รวดเร็วและเหมาะสมกับวัตถุต่าง ๆ ที่จะนำไปใช้ โดยอาจจะทำการทดสอบทั้งกรณีที่ดีที่สุดและกรณีที่แย่ที่สุด เพื่อหาความเหมาะสมกับงานที่จะเอาไปใช้ เช่น การหาความสามารถทางภาษาของพนักงานซึ่งถ้าต้องการหาพนักงานที่สามารถพูดภาษาญี่ปุ่นและฝรั่งเศสได้ จากพนักงานทั้งหมด 1000 คน แต่มีเพียง 5 คนสามารถพูดภาษาญี่ปุ่น และ 100 คนสามารถพูดฝรั่งเศสได้ ซึ่งการค้นหาอาจจะหาจากผู้ที่พูดภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ญี่ปุ่นได้ก่อน แล้วจึงย้อนหาคนที่สามารถพูดฝรั่งเศส จากนั้นนำผลลัพธ์ทั้งสองมารวมกันเพื่อหาคนที่พูดได้ทั้ง 2 ภาษา แต่ถ้าในทางที่แย่ที่สุดคือไม่มีพนักงานที่พูดได้ทั้ง 2 ภาษา เป็นต้น

การจัดการแอตทริบิวต์ที่อยู่ในแต่ละวัตถุ ซึ่งแอตทริบิวต์ถือเป็นข้อมูลอย่างหนึ่งในแต่ละวัตถุอาจมีความซับซ้อนเกิดขึ้นได้ ดังนั้นผู้ออกแบบควรที่จะจัดเก็บข้อมูลเหล่านี้ให้อยู่ในรูปของฐานข้อมูล โดยการจัดเก็บนั้นควรจะคำนึงถึงการนำไปใช้งาน เพราะการใช้งานแต่ละครั้งค่าที่อยู่ภายในสามารถที่จะเกิดการเปลี่ยนแปลงได้ตลอดเวลา ซึ่งวิธีการปรับปรุงค่าของข้อมูลควรคำนึงถึงหลักการดังนี้ ความถูกต้องในการปรับปรุง (by explicit code) ช่วงเวลาในการปรับปรุง (by periodic recomputation) และค่าที่ถูกใช้งาน (by using active values) วิธีการแรก พิจารณาถึงความถูกต้องในการปรับปรุงค่าของแอตทริบิวต์ในแต่ละวัตถุผู้ออกแบบควรที่จะเพิ่มโค้ดของโปรแกรมสำหรับฟังก์ชันในการอัพเดทในแต่ละวัตถุที่แอตทริบิวต์นั้นเกิดการเปลี่ยนแปลงให้ชัดเจนและตรงกับวัตถุที่รับผิดชอบ ต่อมาพิจารณาถึงความถี่ในการปรับปรุงค่าของแอตทริบิวต์ในแต่ละวัตถุ กลุ่มของค่าพื้นฐานที่มีการปรับปรุงอยู่บ่อย ๆ เป็นอีกตัวอย่างหนึ่งที่ควรที่จะคำนึงถึงเพราะบางครั้งกลุ่มของแอตทริบิวท์ที่ได้รับมาจากวัตถุต่าง ๆ ปราศจากปรับปรุงค่าก่อนส่งออกมาทั้ง ๆ ที่ค่าเหล่านี้ควรที่จะทำให้เป็นค่าปัจจุบันก่อน ที่เป็นเช่นนี้เพราะกลุ่มของแอตทริบิวต์เหล่านี้กระจายอยู่ในหลายวัตถุจึงทำให้การปรับปรุงค่านั้นมีผลกระทบมากขึ้นโดยจะต้องไล่ตามไปปรับปรุงทุก ๆ ที่ที่มีค่านั้นอยู่ ดังนั้นวิธีการปรับปรุงแบบช่วงเวลาจึงเป็นวิธีที่ชัดเจนและมีข้อบกพร่องน้อยที่สุด แต่วิธีนี้ยังมีข้อบกพร่องอยู่ในกรณีที่ ณ เวลาใด ๆ กลุ่มของดาต้าที่มีเกิดการเปลี่ยนแปลงน้อยครั้งวิธีนี้จะละเลยหน้าที่ไปเพราะจะสนับสนุนกับกลุ่มแอตทริบิวต์ที่มีขนาดใหญ่เท่านั้น วิธีการสุดท้ายพิจาณาถึงค่าที่ถูกใช้งาน ณ ปัจจุบัน กล่าวคือ ค่าที่ถูกนำไปใช้งานนั้นควรเป็นค่าที่มีมูลค่าของค่านั้น ณ การใช้งานจริงในแต่ละครั้ง ซึ่งการกำหนดค่าเริ่มต้นให้กับสเตทไดอะแกรมผู้ออกแบบควรที่จะพิจารณาถึงความเหมาะสมกับโปรแกรมที่นำไปใช้งานด้วย

4. การควบคุมการสำหรับการอิมพลีเมนต์ (Implementation of control)

แบบจำลองไดนามิคที่นำเสนอด้วยสเตทไดอะแกรม ผู้ออกแบบสามารถนำมาวางแผนสำหรับการอิมพลีเมนต์โดยใช้ สถานะ-อีเวนท์โมเดลเข้ามาประกอบการพิจารณา ในส่วนของการออกแบบระบบจำเป็นต้องเลือกจากแผนงานที่เกิดขึ้นจริงซึ่งระหว่างการออกแบบวัตถุเราอาจต้องขจัดบางแผนงานออกไป สเตทไดอะแกรมที่อยู่ในแบบจำลองไดนามิคเราสามารถนำมาเปลี่ยนแปลงไปสู่ขั้นตอนการทำงานของโปรแกรมในระบบหรือที่เรียกว่า “Procedure-driven System” โดยปกติการควบคุมโปรแกรมจะถูกนำเสนอในรูปแบบทั่วไป ซึ่งจะถูกกำหนดโดยสถานะของโปรแกรมในทุกสเตทไฟไนท์แมชีน (Stage Finite Machine) เพื่อแปลงไปสู่โค้ดของโปรแกรม และทุกสเตททรานส์ซิชั่นสามารถนำไปสู่สถานะของการอินพุตค่าของข้อมูลได้ หลังจากอ่านค่าอินพุตโปรแกรมจะดำเนินการต่อโดยจะขึ้นอยู่กับอินพุตที่ป้อนเข้ามา ซึ่งค่าที่รับเข้ามานั้นโปรแกรมจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทราบเพียงแต่ว่ามีค่าผ่านเข้าออกและรับรู้ในการรับและส่งค่าเท่านั้น โดยบางกรณีอาจเป็นเพียงการตรวจเช็คเงื่อนไขของโปรแกรมอีกครั้งหนึ่ง ซึ่งมีวิธีการเปลี่ยนแปลงสเตทไดอะแกรมไปสู่โค้ดของโปรแกรมโดยมีขั้นตอนดังต่อไปนี้

1. กำหนดเส้นทางเริ่มต้นของการควบคุมโปรแกรม เริ่มพิจารณาจากสถานะเริ่มต้นเพื่อกำหนดเส้นทางการทำงานของสเตทไดอะแกรมที่เหตุการณ์นั้นรับผิดชอบอยู่จากนั้นให้เขียนชื่อของสถานะตามลำดับการทำงานที่เกิดขึ้นจริง เพื่อนำไปสู่ขั้นตอนและลำดับการทำงานจริงของโปรแกรม
2. พิจารณาสถานะที่เส้นทางในสเตทไดอะแกรมถูกแยกออกจากสถานะหลัก เพื่อกำหนดเงื่อนไขให้กับโปรแกรม
3. พิจารณาสถานะที่มีลักษณะการทำงานแบบวนลูปในสเตทไดอะแกรม เพื่อนำไปเลือกวิธีการใช้ลูปในโปรแกรม โดยการนำสเตทไดอะแกรมมาพิจารณาเพื่อให้สามารถเลือกใช้ลูปได้ง่ายขึ้น เพราะถ้าการวนกลับของการทำงานในแต่ละสถานะที่มีลักษณะการทำงานแบบซ้ำซ้อนหลาย ๆ ชั้นสามารถที่จะพิจารณาได้ง่ายกว่าเนื่องจากผู้วิเคราะห์สามารถแยกลูปต่าง ๆ เหล่านั้นได้จากการสังเกตสถานะต้นทางและปลายทางตามเส้นทางของสถานะทรานส์ซิชั่นที่เกี่ยวข้องเพื่อแยกลูปแต่ละอันออกเป็นชั้น ๆ แทนที่จะเหมือนการเขียนแผนผังงานแบบเดิม เพราะในสถานะจะรวมคำสั่งหรือขั้นตอนต่าง ๆ ไว้ในที่เดียวกัน
4. สถานะและทรานส์ซิชั่นที่ไม่ใช่ส่วนของเงื่อนไข สามารถที่จะนำเทคนิคของภาษาต่าง ๆ ที่เลือกใช้อิมพลีเมนต์มาช่วยจัดการทางด้านการตรวจสอบความถูกต้องของโครงสร้างภาษานั้น ๆ ซึ่งส่วนใหญ่จะนำมาใช้กับการตรวจสอบเงื่อนไขที่มีการเปรียบเทียบและการวนลูปของโปรแกรม โดยการนำมาใช้นั้นควรจะคำนึงถึงความสามารถของภาษา

นอกจากวิธีที่กล่าวมาข้างต้น ผู้ออกแบบสามารถที่จะนำวิธีการของสถานะแมชชีนหรือ “Event-driven System” มาใช้กับโปรแกรมได้ ซึ่งส่วนใหญ่การอิมพลีเมนต์โปรแกรมด้วยสถานะแมชชีนมักจะมีวิธีการปฏิบัติเช่นนี้ คือ ผู้วิเคราะห์สามารถย้ายจากแบบจำลองการวิเคราะห์ไปสู่โครงสร้างของระบบได้รวดเร็วขึ้น โดยสามารถกำหนดคลาสได้จากแบบจำลองวัตถุ กำหนดสถานะแมชชีนจากแบบจำลองไดนามิคและสร้าง “stub” จากการทำงานของการกระทำได้ทันทีโดยไม่ต้องผ่านขั้นตอนมากมาย ซึ่ง “stub” คือ ส่วนของฟังก์ชันหรือกิจกรรมย่อยของระบบงานซึ่งเป็นเพียงโค้ดของโปรแกรมที่ปราศจากค่าภายใน (การส่งค่ากลับ) และอาจเป็นเพียงโฟล์วสำหรับตรวจสอบการทำงานและความถูกต้องของโปรแกรมก็ได้ การสร้างสเตทแมชชีนนั้นในภาษาเชิงวัตถุมันเป็นสิ่งที่ยุ่งยากสำหรับการกำหนดอยู่บ้าง เพราะต้องพิจารณาจากสเตทไดอะแกรมเพื่อให้ทราบว่าส่วนใดเป็นสเตทแมชชีนที่แท้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าสังเกตจากสเตทไดอะแกรมที่เกิดขึ้นกับบางวัตถุจะพบว่ามีบางสถานะที่มีการทำงานลักษณะขนานกันในวัตถุเดียวกัน ซึ่งการทำงานลักษณะเช่นนี้ ควรนำไปอิมพลีเมนต์กับภาษาที่สามารถสนับสนุนวิธีการนี้ได้ ภาษาที่สนับสนุน เช่น Concurrent Pascal และ Concurrent C++ เป็นต้น ซึ่งในปัจจุบันภาษาที่สามารถสนับสนุนการทำงานแบบขนานสำหรับวิธีการเชิงวัตถุนั้นค่อนข้างหายาก

5. ปรับปรุงการสืบทอดคุณสมบัติ (Adjustment of Inheritance)

การสืบทอดคุณสมบัติของคลาสที่เกิดขึ้นจากขั้นตอนการวิเคราะห์พอมาถึงขั้นตอนการออกแบบผู้ออกแบบควรที่นำไดอะแกรมเหล่านั้นกลับมาพิจารณาถึงความถูกต้องและเหมาะสมอีกครั้ง ซึ่งการสืบทอดคุณสมบัติของคลาสนั้นเป็นคุณสมบัติและความจำเป็นอย่างหนึ่งของวิธีการนี้ก็ตาม แต่ควรที่จะนำกลับมาทบทวนอีกครั้งเพื่อให้ไดอะแกรมมีความถูกต้องและเหมาะสมก่อนที่จะนำไปอิมพลีเมนต์ สิ่งแรกที่ควรนำมาพิจารณา คือ คลาสและฟังก์ชันที่เกิดจากการสืบทอด ซึ่งการเปลี่ยนแปลงคลาสและฟังก์ชันมีวิธีพิจารณาเพื่อให้คลาสที่มาจากการสืบทอดนั้นมีความเหมาะสม โดยพิจารณาถึงความเป็นไปได้ดังต่อไปนี้

- บางฟังก์ชันในคลาสมีอาร์กิวเมนต์ (Arguments) ที่ไม่เท่ากันเพราะเนื่องจากมีลักษณะการทำงานที่แตกต่างกันออกไปถูกต้องหรือไม่ (ซึ่งสามารถที่จะเพิ่มเข้าไปภายหลังได้แต่ในที่นี้เราเลือกไม่เพิ่มลงไป) เช่น ฟังก์ชันของการแสดงภาพบนจอภาพขาวดำกับภาพสี เป็นต้น พารามิเตอร์มีที่ต่างกันอยู่หนึ่ง คือ “Color” ซึ่งพารามิเตอร์นี้ไม่จำเป็นต้องมีก็ได้เพราะหน้าที่การทำงานเหมือนกันเพียงแต่ว่าแสดงผลกับจอภาพที่แตกต่างกัน
- บางฟังก์ชันอาจจมีอาร์กิวเมนต์น้อยเพราะเป็นกรณีเฉพาะที่แตกต่างไปจากอาร์กิวเมนต์ทั่วไปใช่หรือไม่
- สามารถกำหนดแอตทริบิวต์ที่คล้ายกันให้อยู่ในคลาสที่ต่างกันและอาจให้ชื่อที่ต่างกันได้เพื่อทำให้การทำงานของฟังก์ชันง่ายต่อการเข้าถึงแอตทริบิวต์นั้นและสนับสนุนเทคนิคการปกปิด (Information hiding) ใช่หรือไม่
- ฟังก์ชันหนึ่งอาจกำหนดได้ในหลายคลาสแต่เลือกที่จะกำหนดไว้ที่คลาสแม่เพียงคลาสเดียวตรงตามวัตถุประสงค์หรือไม่

จากหัวข้อที่กล่าวมานี้เป็นเพียงคำถามที่ผู้ออกแบบต้องคำนึงว่าคลาสและฟังก์ชันที่เกิดจากการสืบทอดนั้นตรงตามวัตถุประสงค์หรือไม่ สิ่งที่ต้องพิจารณาต่อไป คือ พฤติกรรมของคลาสซึ่งควรเป็นพฤติกรรมแบบทั่วไป (Common Behavior) เพื่อสามารถที่จะขนถ่ายพฤติกรรมนี้ให้กับคลาสลูกและนำไปใช้งานได้ ซึ่งการขนถ่ายของแบบจำลองวัตถุเรียกว่า “abstracting out” บางครั้งการกำหนดพฤติกรรมทั่วไปให้กับคลาสหลักผู้ออกแบบอาจมีความสับสนสำหรับการนำไปใช้งานของคลาสลูก ซึ่งในความเป็นจริงแล้วพฤติกรรมเหล่านี้จะสามารถถูกเรียกใช้งานได้ทันทีในทุก ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุที่สืบทอดคุณสมบัติมา สุดท้ายพิจารณาถึงความสัมพันธ์ระหว่างคลาส 2 คลาสที่มีลักษณะของการสืบทอดเพราะบางครั้งไดอะแกรมที่เกิดจากขั้นตอนวิเคราะห์อาจเหมาะสมที่จะทำเช่นนั้น แต่หลังจากนำมาพิจารณาใหม่พบว่าบางครั้งการสืบทอดไม่จำเป็นสำหรับคลาสนั้นอาจเปลี่ยนเป็นความสัมพันธ์แบบอื่นได้ เช่น กรณีที่คลาสสแต็กสืบทอดคุณสมบัติมาจากคลาสลิสต์ ซึ่ง ณ ขณะนั้นอาจเข้าใจว่าคลาสสแต็กเป็นคลาสที่เหมาะจะทำเช่นนี้ แต่นำกลับมาพิจารณาใหม่พบว่าการทำเช่นนั้นไม่เป็นผลดีเลย เพราะฟังก์ชันที่เกิดขึ้นในคลาสลิสต์ไม่ได้มีส่วนกระทบต่อคลาสสแต็กเลย เมื่อมีการเพิ่มค่าหรือลบค่าในคลาสสแต็ก ในคลาสลิสต์ไม่มีการเรียกฟังก์ชันใด ๆ ขึ้นมาใช้งาน ในทางกลับกันสิ่งที่อยู่ในคลาสลิสต์ก็ไม่ได้ใช้ฟังก์ชันในการ pop และ push เช่นกัน ดังนั้นควรเปลี่ยนความสัมพันธ์จากการสืบทอดคุณสมบัติไปเป็นความสัมพันธ์แบบ is-a-part-of แทนและทำให้ทั้ง 2 คลาสมีการเรียกใช้ฟังก์ชันระหว่างกันได้ เป็นต้น

จากที่กล่าวมาข้างต้นการพิจารณาการสืบทอดคุณสมบัติของคลาสเป็นสิ่งจำเป็นอย่างยิ่งที่ผู้ออกแบบควรนำไปพิจารณาให้กับไดอะแกรมนั้นมีความถูกต้องและเหมาะสมมากยิ่งขึ้น นอกจากนี้ยังทำให้การนำไปสู่ขั้นตอนการอิมพลีเมนต์มีความชัดเจนและไม่ซับซ้อนอีกด้วย

6. ออกแบบความสัมพันธ์ (Design of Associations)

ความสัมพันธ์ของวัตถุเปรียบเสมือน “กาว” ที่คอยเชื่อมระหว่างสองวัตถุ ซึ่งแนวคิดของความสัมพันธ์เป็นแบบเดียวกับการเชื่อมความสัมพันธ์ของเอนติตีที่มักพบได้จากโมเดลและการวิเคราะห์ ซึ่งระหว่างการออกแบบควรที่กำหนดแนวทางให้กับวัตถุในระบบโดยต้องเลือกวิธีการอิมพลีเมนต์ของความสัมพันธ์ ให้เหมาะสมที่สุด ส่วนใหญ่ความสัมพันธ์ของวัตถุมี 2 ชนิดคือ สัมพันธ์แบบทางเดียวกับสองทาง ดังกล่าวต่อไปนี้

ความสัมพันธ์แบบทางเดียว คือ การอ้างอิงความสัมพันธ์ของวัตถุเพียงข้างเดียว ซึ่งถูกอิมพลีเมนต์ในรูปของพอยน์เตอร์ กล่าวคือ ถ้ามีความสัมพันธ์แบบ 1:1 จะอยู่ในรูปของพอยน์เตอร์ แต่ถ้ามีความสัมพันธ์แบบ 1:M แอตทริบิวต์ฝั่ง “one” จะไปเกาะที่ฝั่ง “many” โดยอิมพลีเมนต์ในรูปแบบเซ็ตของพอยน์เตอร์ ซึ่งถ้ามีความสัมพันแบบ 1:M มากเท่าไหร่ จำนวนเซ็ตของพอยน์เตอร์ย่อมมีมากตามไปด้วย

ความสัมพันธ์แบบสองทาง คือ การอ้างอิงระหว่างวัตถุ ซึ่งมีทางเลือกสำหรับการอิมพลีเมนต์ 3 วิธีดังนี้

- การอิมพลีเมนต์อ้างอิงทางเดียวเหมือนกับความสัมพันธ์แบบทางเดียว โดยให้เหตุผลว่า การอ้างอิงทั้งสองทางนั้น ไม่มีโอกาสเกิดขึ้นบ่อยครั้งนัก ถ้ามีความต้องการจะย้อนกลับก็ให้ทำเป็นกรณีไป ซึ่งวิธีนี้มีข้อดี คือ ใช้เนื้อที่ของฮาร์ดดิสก์และค่าใช้จ่ายในการบำรุงรักษาน้อยแต่ถ้ามีการอ้างอิงทั้งสองทางบ่อยครั้งเท่าไหร่ย่อมจะทำให้ค่าใช้จ่ายเพิ่มขึ้นมากกว่าทุกวิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การอิมพลีเมนต์แบบสองทาง ใช้เทคนิคแบบ Outline คือวัตถุที่อ้างอิงจะมีเซ็ตของพอยน์เตอร์ชี้ถึงกัน ซึ่งทำให้การเข้าถึงข้อมูลนั้นมีความรวดเร็วขึ้น เพราะถ้ามีการปรับปรุงข้อมูลในแต่ละแอตทริบิวต์ในวัตถุจะเก็บลิงค์ของพอยน์เตอร์ที่มาทำการปรับปรุงไว้ด้วย เพื่อให้ทราบภายหลังว่าถูกปรับปรุงโดยใคร วิธีนี้มีข้อดี คือ การปรับปรุงสามารถทำได้สะดวกและเหมาะกับข้อมูลจำนวนมาก
- การอิมพลีเมนต์ในแบบแฮชชิ่งฟังก์ชันเป็นวิธีที่แตกต่างไปจากพอยน์เตอร์แบบที่กล่าวมาข้างต้นคือ ในลิสต์จะเก็บค่าของความสัมพันธ์แบบคู่ลำดับของเซ็ตซึ่งทำให้มีความสะดวกในการหาข้อมูลได้เร็วกว่าวิธีเดิม แต่วิธีนี้มีข้อจำกัด คือ ไม่สามารถที่จะเพิ่มเติมปรับปรุงหรือแก้ไข ข้อมูลได้สะดวก

การเชื่อมแอตทริบิวต์เป็นอีกแห่งหนึ่งที่จะต้องคำนึงถึงเพราะการอ้างอิงถึงกันของวัตถุยังต้องอาศัยหลักความสัมพันธ์ของอี-อาร์ไดอะแกรม คือ ถ้าวัตถุใดเป็นแบบ M:N ให้กระจายออกมาอีกหนึ่งวัตถุเสมอ ถ้าเป็นแบบ 1:1 หรือ 1:M สามารถที่จะผ่านขั้นตอนนี้ได้

7. การแทนวัตถุ (Object Representation)

การอิมพลีเมนต์ของวัตถุจะถูกดำเนินต่อเนื่องไปเรื่อย ๆ ฉะนั้นผู้ออกแบบควรต้องพิจารณาถึงชนิดของแอตทริบิวต์ของออกเจ็กบ้างเพราะถ้ากำหนดชนิดของแอตทริบิวต์ผิดไปจากจุดประสงค์อาจจะมีผลกระทบในระยะยาว ซึ่งชนิดของแอตทริบิวต์มีอยู่หลายประเภทคือ ชุดตัวอักษร (String) ตัวเลขจำนวนเต็ม (Number of Integer) และ ตัวเลขจำนวนจริง (Number of Real) เช่น วัตถุของพนักงานที่มีแอตทริบิวต์หมายเลขประกันสังคม ซึ่งสามารถกำหนดได้ทั้งแบบชุดตัวอักษรหรือตัวเลขจำนวนเต็ม ถ้ากำหนดเป็นแบบชุดตัวอักษรไม่สามารถนำไปคำนวณได้ แต่ถ้ากำหนดเป็นแบบตัวเลขจำนวนเต็มสามารถที่จะนำไปคำนวณได้ แต่ในความเป็นจริงหมายเลขประกันสังคมส่วนใหญ่มักเก็บในรูปของชุดตัวอักษรเท่านั้น เป็นต้น บางกรณีการกำหนดชนิดของแอตทริบิวต์ต้องคำนึงถึงชุดของแอตทริบิวต์ คือ ถ้าแอตทริบิวท์นั้นมีความสัมพันธ์กับแอตทริบิวต์หรือกลุ่มของแอตทริบิวต์ที่ต้องมีการทำงานร่วมกันควรกำหนดให้ไปในทางเดียวกัน

8. การบรรจุคลาสและความสัมพันธ์ไปสู่ระดับกายภาพ (Package classes and association into Physical)

การบรรจุคลาสและความสัมพันธ์ไปสู่ระดับกายภาพ คือ การจัดการขั้นตอนการวิเคราะห์และออกแบบเป็นครั้งสุดท้ายก่อนที่จะนำไปอิมพลีเมนต์ในรูปแบบภาษาต่าง ๆ เช่น C, FORTAN และ Eiffel เป็นต้น ซึ่งจะนำแบบจำลองการวิเคราะห์มาพิจารณาตั้งแต่ ข้อมูลแบบปกปิด (Information Hiding) ความสอดคล้องกันของเอนติตี (Coherence of Entity) และ Constructing Modules ซึ่งทั้งหมดนี้เป็นขั้นตอนที่ควรคำนึงถึงเพราะตามหลักของวิศวกรรมซอฟต์แวร์ระบบที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควรมีความชัดเจน เมื่อย้ายไปอยู่ในรูปของภาษาคอมพิวเตอร์ซึ่งเรียกว่า “แพ็กเกจ (package)” เพราะมันมีความจำเป็นอย่างยิ่ง เพื่อให้การปฏิบัติงานของแต่ละทีมงานสามารถทำงานรวมกันได้

ข้อมูลแบบปกปิด (Information Hiding) เปรียบเสมือนกับ “Black Box” ของคลาส โดยระหว่างการวิเคราะห์ไม่จำเป็นต้องระบุความชัดเจนลงไปในโมเดลแต่ในขั้นตอนการออกแบบนั้นควรที่จะระบุลงไปในคลาสทันที ซึ่งการที่จะระบุนั้นเป็นเรื่องที่ยากพอสมควรที่จะชี้ชัดลงไปว่าแอตทริบิวต์ใดมีข้อมูลแบบปกปิด นอกจากนี้ฟังก์ชันก็เป็นสิ่งสำคัญอีกอย่างหนึ่งที่ควรจะระบุให้ชัดเจน เพราะเป็นกระบวนที่ทำให้เกิดการเปลี่ยนแปลงแอตทริบิวต์ในวัตถุได้

ความสอดคล้องกันของเอนติตี (Coherence of Entity) เป็นอีกหลักการหนึ่งที่สำคัญสำหรับการออกแบบเพราะเป็นสิ่งที่คอยระบุว่าเอนติตี้ในระบบมีความสอดคล้องกันมากน้อยเพียงใด ซึ่งพิจารณาจาก คลาส ฟังก์ชันหรือโมดูล ที่อยู่ในแต่ละคลาส ฟังก์ชันเป็นอีกแนวคิดหนึ่งที่สามารถนำมาใช้ประกอบได้ เพราะหนึ่งฟังก์ชันประกอบด้วย นโยบาย (policy) และ การอิมพลีเมนต์ นโยบายคือ การสร้างกระบวนการตัดสินใจ เช่น อินพุต-เอาท์พุต เงื่อนไข และการเข้าถึงข้อมูลในแต่ละวัตถุเป็นต้น ส่วนการอิมพลีเมนต์ คือ การทำงานของอัลกอริทึมในแต่ละกระบวนการ ซึ่งทั้งสองส่วนจะทำงานพร้อมกันแต่แยกหน้าที่ทำงานกันของวัตถุที่ถูกอ้างถึง เช่น การปรับปรุงยอดบัญชีและการอิมพลีเมนต์จะทำการเพิ่ม/ลดยอดบัญชีที่เกิดขึ้นกับบัญชีของลูกค้า ส่วนนโยบายทำการพิจารณาถึงเงื่อนไขการใช้ค่าแอตทริบิวต์ภายในวัตถุ เป็นต้น

การสร้างแบบจำลอง (Constructing Modules) ระหว่างการออกแบบของวิธีการเชิงวัตถุจะมีสิ่งที่เหมือนกับการแบ่งแยก (Partition) ที่แบ่งความสัมพันธ์ของแบบจำลองวัตถุกับโมดูลของการอิมพลีเมนต์ ซึ่งผู้ออกแบบจำเป็นต้องทำการปรับปรุง หรือจัดทำให้อยู่ในรูป Packaging เพื่อให้สามารถนำไปอิมพลีเมนต์ได้ โดยอาจทำในรูปของ เลเยอร์หรือแยกโมดูลออกเป็นส่วน ๆ ระหว่างการออกแบบระบบ การกำหนดโมดูลส่วนใหญ่จะหมายถึงการทำหน้าจอ (Interface) สำหรับติดต่อระหว่างสองโมดูลที่เกี่ยวข้องกัน กล่าวคือ โมดูลใดในคลาสที่ความเกี่ยวข้องกับฟังก์ชันของคลาสอื่นซึ่งอยู่ในขอบเขตเดียวกันควรทำหน้าจอให้สัมพันธ์และต่อเนื่องกัน ส่วนการติดต่อกันระหว่างแบบจำลองวัตถุกับโมดูลควรใช้แนวทางการแบ่งแยกของโมดูลซึ่งคล้ายกับขั้นตอนการวิเคราะห์มาประกอบการพิจารณาในการจัดการโมดูลที่เกิดขึ้นกับระบบที่รับผิดชอบ

ดังนั้นวิธี Physical Packaging เป็นวิธีที่คอยรวบรวมโมเดลทั้งหมดของวิธีเชิงวัตถุให้อยู่ในรูปของ “Unit” เพื่อง่ายกับการนำไปอิมพลีเมนต์ของระบบต่อไป ซึ่งเป็นสิ่งที่ผู้ออกแบบจำเป็นต้องเข้าไปจัดการกับไดอะแกรมของแบบจำลองทั้งหมดใหม่ในบางกรณี กล่าวคือ ต้องกลับไปตรวจสอบวัตถุ คลาส และฟังก์ชันต่าง ๆ ที่มีอยู่ เพื่อให้ได้ความชัดเจนกับระบบมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 สรุป

การวิเคราะห์และออกแบบเชิงวัตถุเป็นวิธีใหม่สำหรับการพัฒนาระบบสารสนเทศในปัจจุบัน ซึ่งมีอยู่ด้วยกันหลายวิธีให้เลือกใช้ แต่วิธีที่นิยมใช้ คือ วิธีของ James Rumbaugh (OMT) เพราะมีความเข้าใจง่าย การวิเคราะห์แบ่งออกเป็นสัดส่วน ซึ่งมี 3 แบบจำลอง คือ แบบจำลองวัตถุ แบบจำลองไดนามิค และแบบจำลองฟังก์ชัน แบบจำลองทั้ง 3 มีหน้าที่และความรับผิดชอบที่ต่างกันออกไป แบบจำลองวัตถุ แสดงรายละเอียดของวัตถุ คลาส แอตทริบิวต์และความสัมพันธ์ในรูปของไดอะแกรม แบบจำลองไดนามิค แสดงพฤติกรรมของวัตถุและคลาสที่มีการเปลี่ยนแปลงเมื่อมีเหตุการณ์เข้ามากระทบโดยแสดงในรูปของ อีเวนท์เทรส สเตทไดอะแกรมและรายละเอียดของกิจกรรม ส่วนแบบจำลองฟังก์ชัน แสดงการทำงานของกระบวนการที่อยู่ภายในวัตถุและคลาสแต่ละตัว ซึ่งอยู่ในรูปของแผนภูมิการไหลของข้อมูล ทุกแบบจำลองแสดงรายละเอียดในรูปของเอกสารประกอบเพื่อให้สามารถนำไปแก้ไขในภายหลังได้

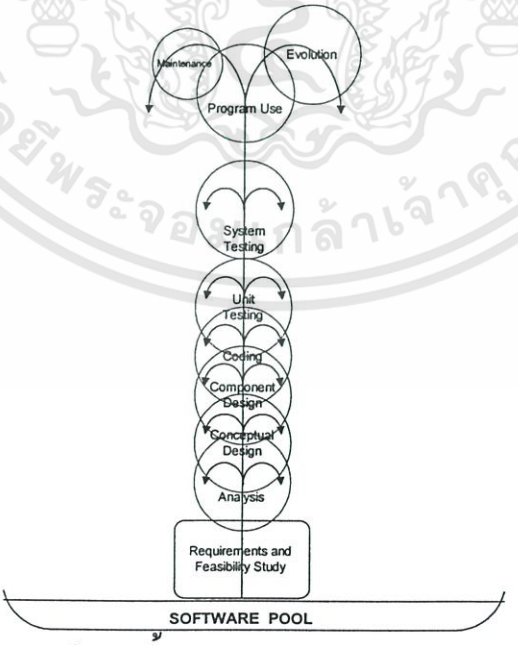
ส่วนการออกแบบ คือ ขั้นตอนที่เชื่อมต่อระหว่างการวิเคราะห์กับการออกแบบระบบ ซึ่งคอยตรวจสอบไดอะแกรมต่าง ๆ ที่เกิดพร้อมกับเพิ่มเติมรายละเอียดของบางขั้นตอนที่ขาดหายไประหว่างการวิเคราะห์และแปลงจากแนวคิดทางตรรกะไปสู่แนวคิดระดับกายภาพเพื่อนำไปสู่ขั้นตอนอิมพลีเมนต์ การเปลี่ยนแปลงเริ่มจาก แบบจำลองวัตถุ แบบจำลองไดนามิค และแบบจำลองฟังก์ชัน ตามลำดับ แบบจำลองวัตถุนำมาพิจารณาถึงความสัมพันธ์ของแอตทริบิวต์และลดความซ้ำซ้อนระหว่างคลาสที่เกี่ยวข้องในไดอะแกรม แบบจำลองไดนามิคสามารถนำไปกำหนดขั้นตอนการทำงานของกระบวนการในวัตถุ ส่วนแบบจำลองฟังก์ชันกับแบบจำลองไดนามิคนำไปสร้างเป็นฟังก์ชัน นอกจากนี้เหตุการณ์จากแบบจำลองไดนามิคยังสามารถนำมาเป็นตัวควบคุมฟังก์ชันของวัตถุได้อีก โดยแบบจำลองทั้งสามผู้ออกแบบจะต้องนำมารวมกันให้เป็นกลุ่มเดียวกันหรือที่เรียกว่า “Packaging” ซึ่งขั้นตอนนี้จะต้องเลือกข้อมูลที่จะนำมากำหนดเป็นข้อมูลแบบปกปิด ความสอดคล้องกันของเอนติตี และการเชื่อมต่อกันระหว่างการวิเคราะห์กับการออกแบบระบบเพื่อให้เป็นอันหนึ่งอันเดียวกันทั้งระบบและง่ายต่อการจัดการในขั้นตอนการอิมพลีเมนต์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
แบบจำลองน้ำพุ
(Fountain Model)

บทนี้จะกล่าวถึงแบบจำลองน้ำพุ [2],[15],[16],[17],[18] ซึ่งเหมาะสมกับการพัฒนาระบบสารสนเทศเชิงวัตถุ เนื่องจากมีลักษณะการทำซ้ำ (Iterative) และการเวียนบังเกิด (recursive) ของแบบจำลองน้ำพุเป็นรูปแบบการพัฒนาระบบสารสนเทศที่มีความยืดหยุ่นและครอบคลุมทั้งการวิเคราะห์ การออกแบบจึงทำให้มีลักษณะของการนำกลับมาใช้ใหม่ได้ แม้ว่ากิจกรรมในการพัฒนาระบบตามแบบจำลองนี้ไม่สามารถเริ่มก่อนกิจกรรมหนึ่งได้ เช่น ผู้พัฒนาไม่สามารถเขียนโปรแกรมก่อนการออกแบบหรือไม่สามารถออกแบบก่อนศึกษาถึงความต้องการของผู้ใช้ได้ เป็นต้น แต่การพัฒนาระบบ โดยใช้แบบจำลองน้ำพุจะยังมีกิจกรรมที่มีการคาบเกี่ยวกัน และรวมกันของกิจกรรมเกือบตลอดทั้งวงจรการพัฒนาในระบบแบบเชิงวัตถุ

แบบจำลองน้ำพุมีรูปแบบการพัฒนาที่เปรียบได้กับน้ำพุที่พุ่ง กล่าวคือเส้นทางการพัฒนาระบบจะพุ่งขึ้นจากส่วนกลางและไหลลงสู่สระเบื้องล่างเพื่อสามารถนำกลับมาใช้ใหม่ได้ (Reusability) โดยทั่วไปการพัฒนาระบบจะเริ่มจากการวิเคราะห์ การออกแบบและการอิมพลีเมนต์ (implementation) ซึ่งสอดคล้องกับระบบเชิงวัตถุคือจะครอบคลุมสองในสามส่วนของวงจรการพัฒนาระบบ (หรือทั้งหมด) ขณะที่สระน้ำคือคลังของคลาสหรือที่เก็บซอฟต์แวร์ส่วนกลาง (Software pool หรือ repository) ดังแสดงในรูป



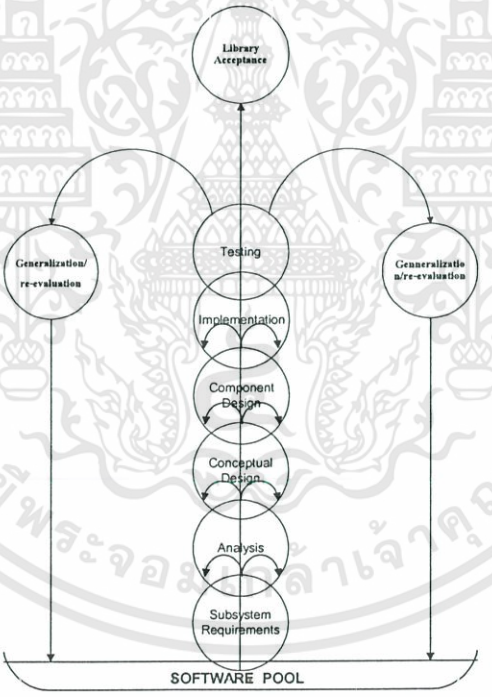
รูปที่ 4.1 แสดงรูปทั่วไปของแบบจำลองน้ำพุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่4.1 แสดงถึงลักษณะทั่วไปในระดับของระบบ (Systems-level) ของการพัฒนาเชิงวัตถุซึ่งมีการรวมกันในระดับสูงของการวิเคราะห์ ออกแบบ การอิมพลีเม้นต์และขั้นตอนการทดสอบ

การพัฒนาระบบสารสนเทศเชิงวัตถุสามารถแบ่งเป็นกลุ่มย่อย (Subsystem หรือ cluster) โดยแต่ละกลุ่มย่อยคือกลุ่มของงานที่ทำงานใกล้ชิดกันคือมีระดับความเกี่ยวเนื่องระหว่างกันสูงแต่มีความเกี่ยวเนื่องระหว่างกลุ่มย่อยนั้นต่ำกว่า แต่ละกลุ่มย่อยถูกกำหนดจากส่วนการวิเคราะห์และออกแบบในระดับของระบบ ส่วนขั้นตอนการพัฒนาในแต่ละคลาสของกลุ่มย่อยสามารถทำลักษณะขนานกันได้แม้ว่าวงจรชีวิตการพัฒนาในแต่ละกลุ่มย่อยมีความแตกต่างกัน

จากรูปที่ 4.2 แสดงรายละเอียดของแต่ละกลุ่มย่อยซึ่งกำหนดให้วงกลมแต่ละวงแทนขั้นตอนต่างๆของกลุ่มย่อย ส่วนนี้จะเน้นในเรื่องของกลุ่มย่อยมากกว่าระดับของระบบ (สังเกตจากรูปที่ 4.1) และจากรูปพิจารณาว่าการคาบเกี่ยวกันของขั้นตอนจะน้อยลง แต่ลักษณะการทำซ้ำยังคงเป็นส่วนที่สำคัญและเห็นได้ชัดเจนในวงจรชีวิต ดังนั้นกลุ่มย่อยที่พัฒนาควรทำให้อยู่ในรูปทั่วไป เพื่อสามารถนำกลับมาใช้ใหม่ได้



รูปที่ 4.2 แสดงขั้นตอนการพัฒนาของแต่ละระบบย่อยของแบบจำลองน้ำพุ

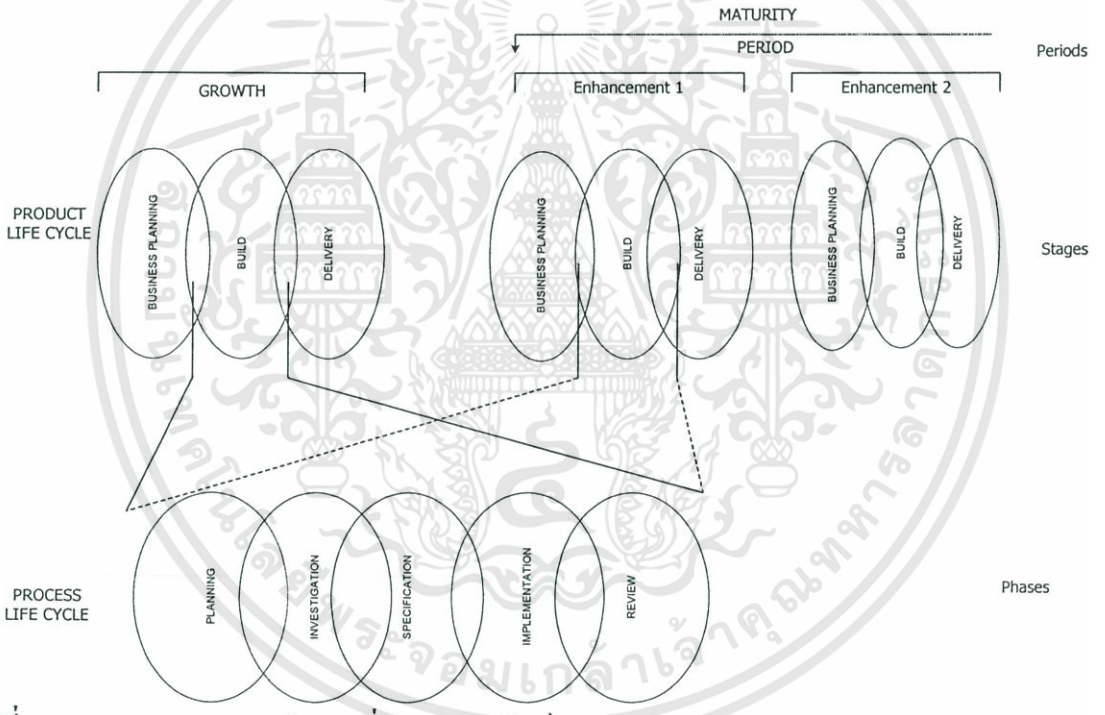
ในขั้นตอนการวิเคราะห์จะเป็นการแสดงความเป็นจริง ในขณะที่ขั้นตอนการออกแบบจะแสดงด้วยการออกแบบระบบสารสนเทศจุดประสงค์หลักของการออกแบบเชิงวัตถุคือ เพื่อช่วยสนับสนุนการออกแบบเชิงวิศวกรรมซอฟต์แวร์ที่ดีในแง่ ความถูกต้อง (Correctness) การเป็นโมดูล (modularity) ความสามารถในการการนำกลับมาใช้ใหม่ (reusability) และ Abstraction เป็นต้น และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลดข้อจำกัดทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่มี ดังนั้นการออกแบบจึงมุ่งประเด็นไปที่คำว่า "ทำอย่างไร" มากกว่าพยายามทำให้รอยต่อของทั้งการวิเคราะห์และการออกแบบเชิงวัตถุมีความเชื่อมกันอย่างสนิทมากยิ่งขึ้น ส่วนกิจกรรมและสัญลักษณ์ที่ใช้จะกล่าวในหัวข้อถัดไป

4.1 คุณลักษณะของแบบจำลองน้ำพุ

แบบจำลองน้ำพุที่กล่าวถึงอยู่ภายในวิธีการของวิศวกรรมซอฟต์แวร์เชิงวัตถุหรือ โมเสส (Methodology for object-Oriented Engineering of Systems หรือ MOSES) ซึ่งมีลักษณะของวงจรชีวิตเต็มรูปแบบและสนับสนุนเครื่องมือต่างๆ เช่น กิจกรรม ข้อเสนอแนะและการส่งมอบของโครงงาน เป็นต้น โดยแต่ละขั้นตอนยังประกอบด้วยสัญลักษณ์และเอกสารต่างๆ โดยแบ่งออกเป็น 2 ส่วนหลักคือ Product Lifecycle และ Process Lifecycle (แสดงดังรูปที่ 4.3) ซึ่งมีรายละเอียดดังต่อไปนี้



รูปที่ 4.3 แสดงภาพรวมของโมเสสที่มีความสัมพันธ์ระหว่าง Product Lifecycle และ Process Lifecycle

4.1.1 วงจรการผลิต (Product Lifecycle)

ภายใน Product Lifecycle แบ่งออกเป็น 2 ส่วนหลัก คือ ส่วนที่เป็นโครงงานระยะสั้น (Growth Period) กับโครงงานระยะยาว (Maturity Period) โครงการระยะสั้นมีระยะเวลาการพัฒนา 1 ปี และเน้นหนักทางด้านวิชาการพัฒนา ซึ่งต่างกับโครงงานระยะยาว คือเริ่มตั้งแต่การกำหนดตามต้องการเบื้องต้นจนถึงค่าใช้จ่ายตลอดทั้ง โครงการจนกระทั่งเสร็จสมบูรณ์ นอกจากนี้ยังครอบคลุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

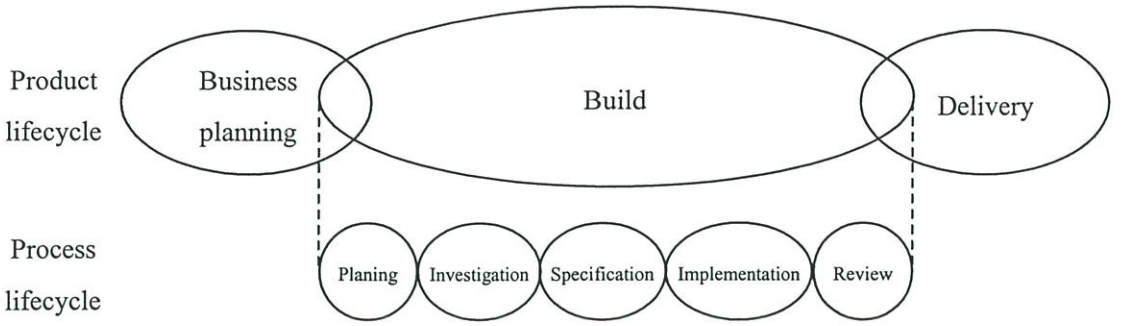
ถึงการบำรุงรักษาและการเพิ่มเติมความเหมาะสมด้วย ความสำคัญของขั้นตอนการบำรุงรักษา การเมเติมสิ่งใหม่ๆและการตรวจสอบนั้นเป็นสิ่งที่ Product Lifecycle ให้ความสำคัญน้อยกว่าขั้นตอนการพัฒนา โครงการระยะสั้นถูกแบ่งขั้นตอนออกเป็น 3 ขั้นตอน คือ ขั้นตอนการวางโครงงานทางธุรกิจ (Business planning Stage) ขั้นตอนการสร้าง (Build Stage) ขั้นตอนการส่งมอบ (Delivery Stage) ขั้นตอนการวางโครงงานทางธุรกิจเน้นในเรื่องของการศึกษาความเหมาะสมของโครงงาน การวางแผนระดับกว้างวิเคราะห์ความเสี่ยง ค่าใช้จ่ายและประโยชน์ที่จะได้รับจากการพัฒนาโครงงาน (หรือที่เรียกว่า CBA – Cost Benefit Analysis) สังเกตได้ว่าขั้นตอนนี้ไม่ได้เน้นไปทางด้านเทคนิค เช่น เทคโนโลยีเชิงวัตถุภาษาโมเดล เป็นต้น แต่เน้นหนักไปทางด้านการสร้างเอกสารและศึกษาความเหมาะสมของโครงงานที่จะพัฒนาเท่านั้น

4.1.2 วงจรการปฏิบัติการ (Process Lifecycle)

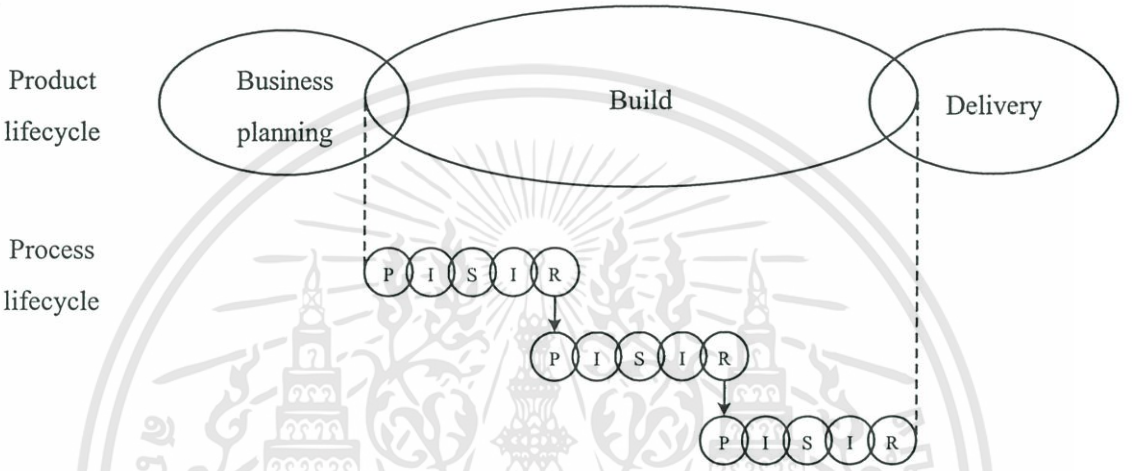
Process Lifecycle เป็นส่วนหนึ่งของ Product Lifecycle ที่อยู่ในส่วนของ MOSES ซึ่งประกอบด้วย 5 ขั้นตอน คือ การวางแผน (Planning) ศึกษาความเหมาะสม (Investigation) การวิเคราะห์และออกแบบ(Specification) การอิมพลีเมนต์ (Implementation) และการตรวจสอบ (Review) โดยแต่ละขั้นตอนจะกล่าวถึงรายละเอียดในบทต่อไป ขั้นตอนทั้ง 5 นี้ไม่มีข้อจำกัดในการพัฒนาก่อนหลังเหมือนกับแบบจำลองน้ำพุ หมายความว่าการทำงานของแต่ละขั้นตอนนั้นสามารถย้อนกลับมาทำในขั้นตอนก่อนหน้าได้ตลอดเวลาในกรณีที่ข้อมูลหรือรายงานจากขั้นตอนก่อนไม่สมบรูณ์ ซึ่งเรียกการทำลักษณะนี้ว่า “Iterative Development Process (IDP) ” แต่ถ้าขั้นตอนต่างๆไม่มีข้อบกพร่องและสมบรูณ์ การพัฒนาจะดำเนินการตามขั้นตอนที่คล้ายกับแบบจำลองน้ำพุ จากรูปที่ 4.4 (ก) แสดงถึงขั้นตอนการพัฒนาตามลำดับคล้ายกับแบบแบบจำลองน้ำพุ ส่วนรูป 4.4 (ข) แสดงถึงการพัฒนาในลักษณะของ IDP คือการทำซ้ำ ๆ กันหลาย ๆ ครั้งซึ่งคล้ายกับการพัฒนาแบบ Spiral Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก)



(ข)



รูปที่4.4 (ก) แสดงการพัฒนาแบบจำลองน้ำพุของ Process และ Product Lifecycle
(ข) แสดงการพัฒนาในสภาพแวดล้อมแบบทำซ้ำของ Process และ Product Lifecycle

การพัฒนาแบบ IDP ใน Process Lifecycle นั้นนับว่ามีความสำคัญอย่างมาก เพราะสามารถพัฒนาได้พร้อมๆกัน โดยใช้ข้อมูลจากแหล่งเดียวกัน จึงทำให้มีความถูกต้องและรวดเร็วมากขึ้นกว่าเดิม ข้อดี คือการทำลักษณะเช่นนี้ทำให้การทำงานระหว่างนักพัฒนากับผู้ใช้ใกล้ชิดกันมากยิ่งขึ้น เพราะสามารถเห็นสิ่งที่เกิดขึ้นแต่ละขั้นตอนได้อย่างใกล้ชิดมากขึ้นและสามารถเปลี่ยนแปลงและควบคุมคุณภาพได้ด้วย

4.1.3 โมเสสกับวงจรการผลิตและวงจรการปฏิบัติงาน (MOSES : both a product and process Lifecycle methodology)

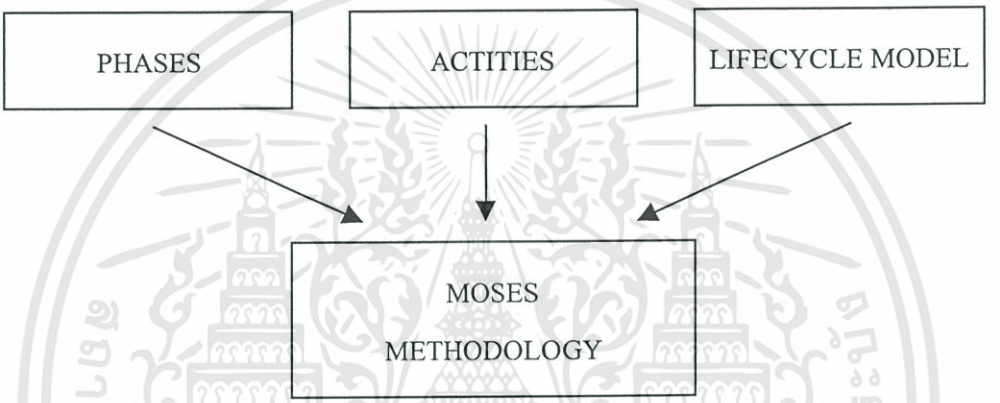
แม้ว่าบางครั้งการทำงานของ Process Lifecycle จะมีส่วนคาบเกี่ยวกับ Product Lifecycle อยู่บ้างแต่ทั้ง 2 วงจรชีวิตนั้นมีจุดประสงค์เดียวกันคือ

- สนับสนุนทางด้านวิศวกรรมซอฟต์แวร์ทั้งโครงงานระดับเล็กและใหญ่ของวิธีการเชิงวัตถุ
- การเคลื่อนย้ายระหว่างการพัฒนาไปสู่อิมพลีเมนต์มีความราบรื่นมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถขยายและยืดหยุ่นระบบได้ง่าย
- สนับสนุนการนำกลับมาใช้ใหม่

แม้ว่าการพัฒนาที่เกิดขึ้นภายในโมเสสที่อยู่ในส่วนของ Process Lifecycle จะมีการพัฒนาเหมาะสมกับโครงงานทุกๆขนาดก็ตาม แต่สิ่งหนึ่งที่ขาดไปคือ คำแนะนำแต่ละขั้นตอน ซึ่งสิ่งนี้ก็ไม่ได้เป็นปัญหาต่อการพัฒนา เพราะว่าภายในโมเสส ประกอบด้วย ขั้นตอน (Phase) กิจกรรม (Activities)และรูปแบบของวงจรชีวิตการพัฒนา (Lifecycle Model) ทั้ง 3ส่วนนี้จะคอยช่วยให้การพัฒนามีความยืดหย่นและควบคุมได้ง่ายและสมดุลย์มากขึ้น(ดังแสดงในรูปที่ 4.5)



รูปที่ 4.5 แสดงการเชื่อมต่อระหว่าง ขั้นตอน กิจกรรมและวงจรชีวิตของโมเสส

ภายในขั้นตอนประกอบด้วย 5 ขั้นตอน คือ การวางแผน (Planning) ศึกษาความเหมาะสม (Investigation) Specification การอิมพลีเมนต์ (Implement) และการตรวจสอบ (Review) นอกจากนี้ขั้นตอนทั้ง 5 ยังเป็นหัวใจหลักของ Process Lifecycle เพราะเป็นส่วนที่ทำให้เกิดการพัฒนาระบบขึ้น (ดังแสดงในรูปที่ 4.3) กิจกรรม (Activities) เป็นขั้นตอนการพัฒนาที่เกิดขึ้นในแต่ละโพรเซสของพัฒนาระบบ (ซึ่งจะกล่าวในหัวข้อที่ 4.3) ส่วนวงจรชีวิต (Lifecycle Model) เป็นกรอบของการพัฒนาที่แสดงถึงลำดับและความสัมพันธ์ระหว่างขั้นตอนของ Process Lifecycle ในส่วนของขั้นตอนการสร้าง ซึ่งทั้ง 5 ขั้นตอนจะมีความเกี่ยวข้องเนื่องกันตลอดทั้งวงจรถึงแม้ว่าอยู่ต่างวงจรชีวิตกัน (ดังแสดงในรูปที่ 4.4)

ทั้ง 3 ส่วนประกอบรวมกันเป็นโมเสสและจะสมบรูณ์ก็ต่อเมื่อทุกส่วนถูกรวบรวมไว้ใน Process Lifecycle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 สัญลักษณ์ที่ใช้ในแบบจำลองน้ำพุ

ในหัวข้อนี้นำเสนอสัญลักษณ์และเอกสารที่ใช้สำหรับออกแบบระบบสารสนเทศเชิงวัตถุที่ใช้ในโมเสสซึ่งมีอยู่ด้วยกันหลายตัวด้วยกัน สัญลักษณ์ต่างๆเหล่านี้ถูกนำมาประยุกต์ใหม่โดย Edwards กับ Henderson-sellers[18] ที่ประกอบด้วย UON (Uniform Object Notation) ถูกคิดขึ้นโดย Page-Jones[38] และ Object Chart ถูกคิดขึ้นโดย Coleman[10] เป็นต้น โดยทั่วไปสัญลักษณ์ที่ใช้อยู่นั้นจะไม่สนับสนุนทางด้านกายภาพมากกว่าภาษาซึ่งขัดแย้งกับสิ่งที่นำเสนอ แต่ในโมเสสสนับสนุนทั้ง 2 ด้าน คือ ทั้งคุณสมบัติพื้นฐานของแนวคิดเชิงวัตถุและคุณสมบัติพิเศษทางภาษาเช่น ฟังก์ชัน friend ในภาษา C++ เป็นต้น จึงทำให้ครอบคลุมในทุกขั้นตอนของวงจรชีวิตนอกจากนี้ยังมีคุณสมบัติต่างๆอีก เช่น สนับสนุนทั้งด้านสแตติกและไดนามิก เหมาะสำหรับการจัดการระบบขนาดใหญ่และซับซ้อน มีเอกสารประกอบทุกขั้นตอนโดยละเอียด เป็นต้น ส่วนใหญ่ในทุกๆการออกแบบให้กับระบบไม่ว่าจะเป็นระบบเชิงวัตถุหรือไม่ มักจะมีการออกแบบในลักษณะขั้นบันไดหรือมีระดับ (Level) การเข้าถึง เช่น การออกแบบเชิงโครงสร้างจะมีระดับ 1,2,3,...,n ตามลำดับเป็นต้น ซึ่งในโมเสสมีลักษณะเช่นเดียวกันโดยที่ระดับขั้นเหล่านี้จะเป็นตัวบ่งบอกถึงความละเอียดของการออกแบบที่แตกต่างกันออกไป เช่น O/C Model ในระดับแรกจะแสดงเพียงรายละเอียดคร่าวๆ เท่านั้น เปรียบได้กับ Context Diagram พอระดับถัดไปค่อยๆกระจายคลาสที่เกี่ยวข้องและเหมาะสม เป็นต้น

4.2.1 คลาสและแบบจำลองวัตถุ(O/C MODEL)

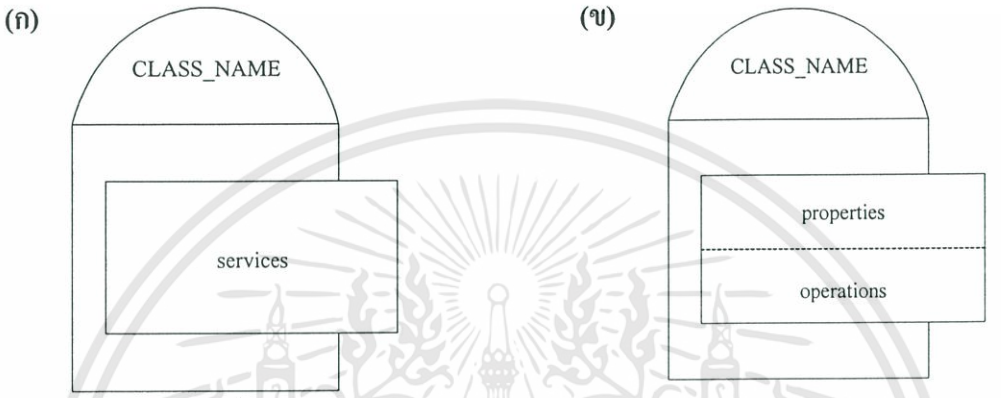
คลาสและวัตถุดีคือการนำเสนอโครงสร้างโดยรวมของระบบทั้งหมดไม่ว่าจะเป็นระบบหลักหรือระบบย่อยโดยจะแสดงในรูปของเชิงวัตถุที่สามารถเข้าใจง่ายและยังสามารถนำไปขยายหรือกระจายกับส่วนที่เกี่ยวข้องกันได้อีกด้วย สัญลักษณ์ของ O/C Model มีหลายลักษณะซึ่งจะนำเสนอดังต่อไปนี้

4.2.1.1 คลาส (Class)

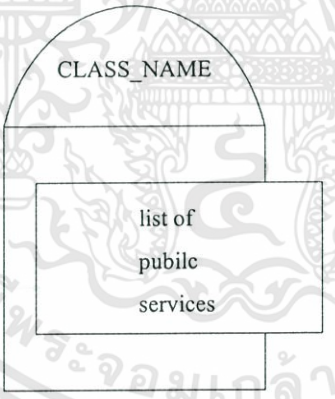
คลาสคือ กลุ่มที่มีลักษณะคล้ายกัน เหมือนกัน หรือพวกเดียวกัน มารวมไว้ที่เดียวกัน ซึ่งนิยามนี้มีความหมายเหมือนกันกับทุก ๆ วิธี แต่ส่วนที่ทำให้แตกต่างกันออกไปคือรูปร่างของสัญลักษณ์เหล่านี้จะไม่มีมิติ คือ มีลักษณะแบนราบ ในโมเสสจะมีสัญลักษณ์ของคลาสอยู่หลายลักษณะซึ่งแสดงดังต่อไปนี้ ในรูปที่ 4.6 (ก) คือลักษณะของคลาสโดยทั่วไป ซึ่งสังเกตจากรูปส่วนบนเป็นรูปโดมและภายในประกอบด้วยชื่อของคลาส ส่วนท่อนล่างเป็นรูปสี่เหลี่ยมจัตุรัส ภายในประกอบด้วยเซอร์วิสทั้ง private และ public เซอร์วิสจะอยู่ในสี่เหลี่ยมผืนผ้าที่ยื่นออกมาจากตัวคลาสเพื่อให้เห็นความแตกต่างระหว่างทั้ง 2 เซอร์วิส ภายใน public เซอร์วิสสามารถแบ่งออกเป็นแอตทิบิวต์กับ โอเปอร์เรชั่น และขั้นด้วยเส้นประ (ดังแสดงในรูปที่ 4.6 (ข)) ซึ่งเส้นนี้สามารถยกเลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ถ้าไม่ต้องการ ส่วนรูปที่ 4.7 แสดงคลาสที่มีลักษณะ abstrsct จะอยู่ในรูปของเส้นประทั้งคลาส การออกแบบคลาสบางครั้งมีความต้องการรายละเอียดของ แอตทิรบิวต์ กับ โอเปอร์เรชั่น ในส่วนของการแสดงชื่อเซอร์วิสประเภท Private โมเสสมีสัญลักษณ์ที่สนับสนุน ดังแสดงในรูปที่ 4.8 โดยที่สัญลักษณ์ เป็นส่วนที่บรรจุแอตทิบิวต์หรือข้อมูลของ Private เซอร์วิส หรือ เรียกว่า “data box” ส่วนสี่เหลี่ยมผืนผ้าจะบรรจุด้วยเซอร์วิส ของ Private รายละเอียดทั้งหมดจะอยู่ภายในคลาสทั้งหมด ไม่มีส่วนใดยื่นออกมานอกคลาสเหมือนกับ public เซอร์วิส

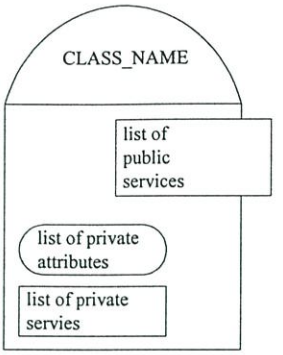


รูปที่ 4.6 (ก) แสดงสัญลักษณ์ของคลาส
(ข) แสดงสัญลักษณ์ของคลาสโดยแบ่งระหว่าง Properties และ Operation



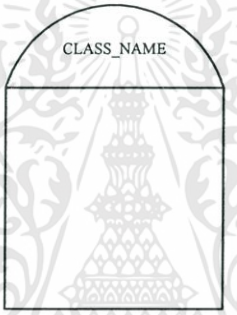
รูปที่ 4.7 แสดงสัญลักษณ์ของ Abstract Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



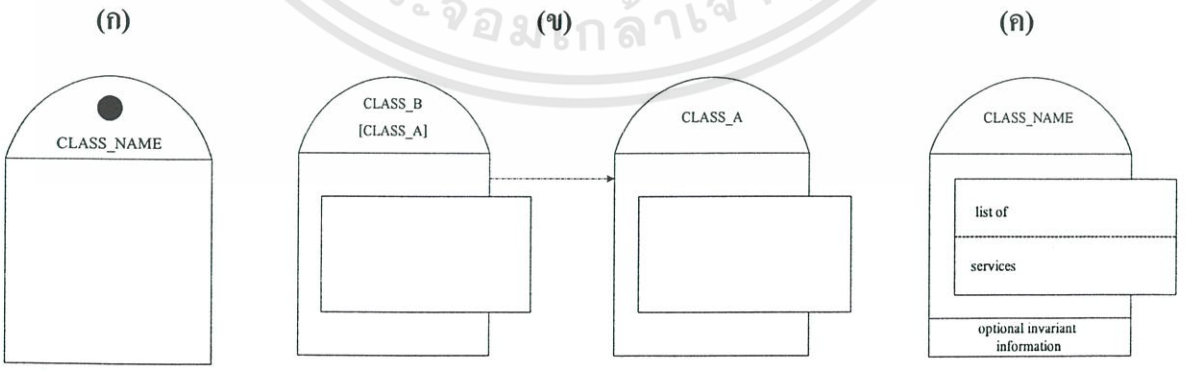
รูปที่ 4.8 แสดงการทำงานของเซอร์วิสภายในคลาสเมื่อมีการเรียกใช้

คลาสไลบรารี (Class library) แสดงอยู่ในรูปของเส้นหนาตลอดทั้งคลาส ดังแสดงในรูปที่ 4.9 ซึ่งคลาสในลักษณะนี้ไม่จำเป็นที่ต้องสร้างขึ้นมาใหม่แต่จะนำมาปรับปรุงหรือแก้ไขเพื่อจะนำมาเป็นส่วนขยายหรืออ้างอิงในการออกแบบโครงงานปัจจุบัน



รูปที่ 4.9 แสดงสัญลักษณ์ของคลาสไลบรารี

จากสัญลักษณ์ที่กล่าวมานั้นเป็นสัญลักษณ์ที่ใช้ในโมเสสทั้งสิ้น และยังมีอีกมากเช่น สัญลักษณ์ของ Persistence Class, Generic Class และคลาสที่มีลักษณะของ invariant ดังแสดงในรูปที่ 4.10 (ก) (ข) และ (ค) ตามลำดับ

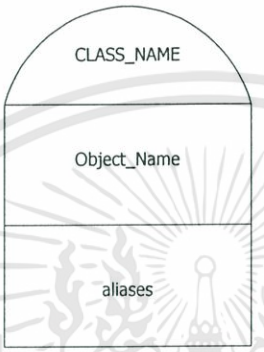


รูปที่ 4.10 แสดงสัญลักษณ์ของคลาสในลักษณะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.2 วัตถุ (Object)

วัตถุเป็นส่วนที่เล็กกว่าคลาส มีลักษณะรูปร่างคล้ายกันคือ มีส่วนบนเป็นรูปโดมที่ประกอบด้วยชื่อของคลาสและส่วนล่างเป็นรูปสี่เหลี่ยมจัตุรัส ภายในแบ่งออกเป็น 2 ส่วน คือชื่อวัตถุและพื้นที่หรือแหล่งที่อ้างอิง (Aliases) (ดังแสดงในรูปที่ (4.11))แหล่งที่อ้างอิง คือส่วนที่บ่งบอกถึงชื่อของวัตถุที่อ้างถึงกัน เช่น วัตถุ A อ้างอิง วัตถุ B ที่ชื่อ BB ในวัตถุ B จะต้องใส่ชื่อของวัตถุ A ลงในส่วนของแหล่งอ้างอิง เป็นต้น สาเหตุที่ต้องทำเช่นนี้ เพราะว่าในแต่ละวัตถุอาจมีชื่อซ้ำกันก็เป็นได้ ดังนั้นเพื่อความชัดเจนจึงควรระบุชี้ชัดลงไป เพื่อลดความซ้ำซ้อนปัญหาที่เกิดขึ้นกับแบบจำลองวัตถุ



รูปที่ 4.11 แสดงสัญลักษณ์ของวัตถุ

4.2.1.3 ความสัมพันธ์ (Relationships) สัญลักษณ์ของความสัมพันธ์ระหว่างวัตถุมีอยู่ด้วยกันหลายรูปแบบ เช่น ความสัมพันธ์แบบการสืบทอดคุณสมบัติ (Inheritance) แบบร่วมหรือเป็นส่วนหนึ่ง (Aggregation) และธรรมดา (Association) ซึ่งมีรายละเอียดดังต่อไปนี้

1) ความสัมพันธ์แบบการสืบทอดคุณสมบัติ (Inheritance Relationship)

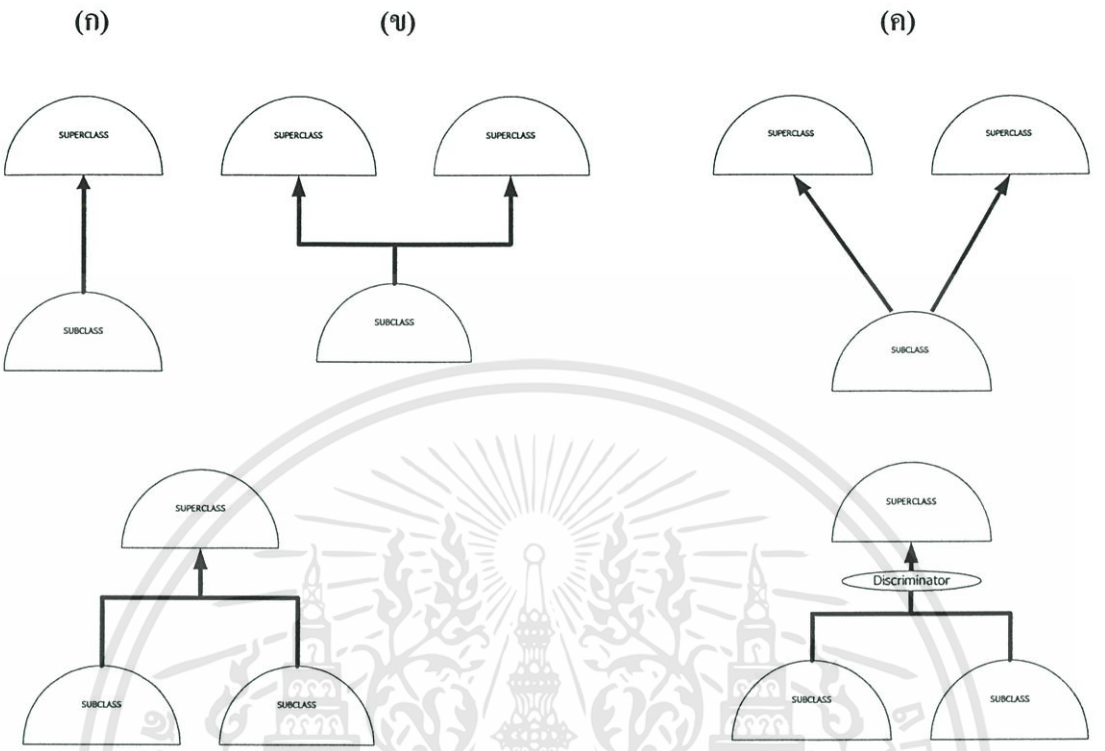
การสืบทอดคุณสมบัติแบ่งออกเป็น 2 ลักษณะ คือ แบบทั่วไป (Generalization) กับ แบบของการอิมพลีเมนต์(Implementation Inheritance)ซึ่งทั้งหมดมีลักษณะการใช้ในระดับที่แตกต่างกัน กล่าวคือแบบทั่วไปจะใช้ในระดับของการออกแบบหรือโมเดล ส่วนการอิมพลีเมนต์ใช้ในระดับของภาษาหรือระดับการอิมพลีเมนต์ สัญลักษณ์ที่ใช้มีลักษณะที่เหมือนกันทั้ง 2 แบบ คลาสย่อย (Subclass) และซุปเปอร์คลาส (Superclass) โดยมีลักษณะเป็นรูปโดม (ดังแสดงในรูปที่ 4.12)

1.1) การสืบทอดคุณสมบัติลักษณะทั่วไป (Generalization Inheritance)

การสืบทอดคุณสมบัติในลักษณะทั่วไปของคลาสแสดงด้วยเส้นหนาและมีลูกศรบอกทิศทางโดยเริ่มจากคลาสย่อยไปยังซุปเปอร์คลาส ดังแสดงในรูปที่ 4.12 (ก) ส่วนการสืบทอดแบบมากกว่าหนึ่ง ในรูปแบบของซุปเปอร์คลาสที่มีคลาสย่อย ๆ หลายตัวจะแสดงความสัมพันธ์ในลักษณะของลูกศรเป็นรูปป้อม หรือจะโยงลูกศรในลักษณะแยกกัน (ดังแดงในรูปที่ 4.12 (ข) และ (ค) แต่ถ้าการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สืบทอดของคลาสย่อยมีซุปเปอร์คลาสเดียวกันเส้นคลาสแสดงความสัมพันธ์จะลากจากคลาสย่อยทั้งหมดซึ่งปลายทางลูกศรจะไปบรรจบที่ซุปเปอร์คลาส (ดังแสดงในรูปที่ 4.12 (ง))



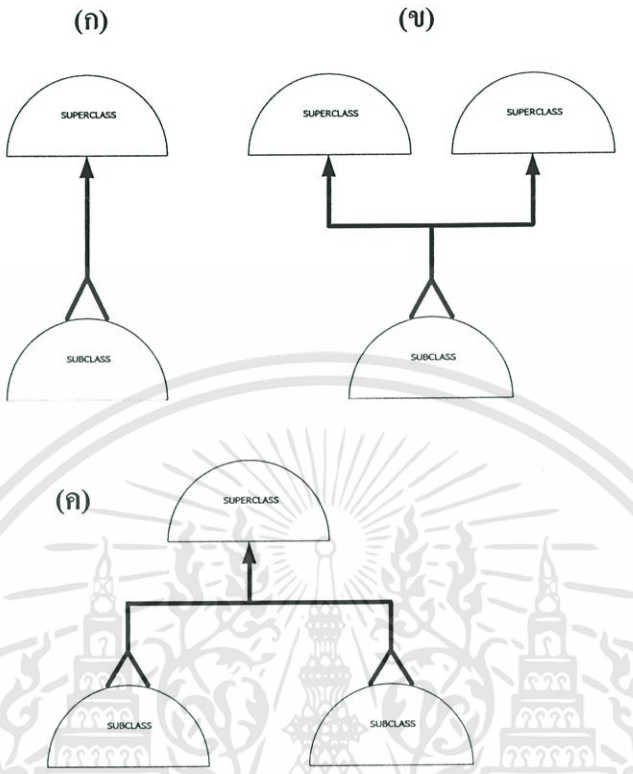
รูปที่ 4.12 ก) แสดงสัญลักษณ์ของการสืบทอดแบบ Single Generalization
ข) แสดงสัญลักษณ์ของการสืบทอดแบบหลายระดับ
ค) แสดงสัญลักษณ์ของการสืบทอดแบบมีทางเลือก
ง) แสดงสัญลักษณ์ของการสืบทอดแบบมาจากคลาสย่อย
จ) แสดงสัญลักษณ์ของการสืบทอดที่มาจากคลาสย่อยแต่ระบุ Property

1.2) อิมพลีเมนต์การสืบทอดคุณสมบัติ (Implementation Inheritance)

การนำเสนอรูปแบบของการสืบทอดในลักษณะของการอิมพลีเม้นต์จะแตกต่างจากเดิม กล่าวคือ ในรูปแบบเดิมที่เคยนำเสนอแสดงความสัมพันธ์ระหว่างคลาสย่อยกับซุปเปอร์คลาสซึ่งอยู่ในระดับของ Specification เท่านั้น ส่วนระดับภาษาหรืออิมพลีเม้นต์นั้นจะนำเอากระบวนการหรือวิธีการที่เลือกไว้มาเขียนเป็นโปรแกรม เช่น การทำสแต็ก (STACK) ซึ่งใน Specification จะเห็นเพียงคลาสเดียวแต่ในระดับภาษาคลาสของสแตกมีการสืบทอดไปยังคลาสใหม่ที่ชื่ออาร์เรย์ (ARRAY) ซึ่งคลาสนี้เป็นคลาสที่ถูกเลือกไว้แล้วว่าจะนำเสนอวิธีการของสแตกด้วยการใช้อาร์เรย์ เป็นต้น สัญลักษณ์ที่ใช้ประกอบด้วยเส้นหนาและบ่งบอกทิศทางด้วยลูกศรและมีสัญลักษณ์รูปหมวกอยู่ทางด้านท้าย (ดังแสดงในรูปที่ 4.13) การลากเส้นความสัมพันธ์ ลากจากคลาสย่อยไปยังซุปเปอร์คลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คล้ายกับการสืบทอดแบบทั่วไปประโยชน์ของการนำเสนอของรูปแบบนี้ทำให้เข้าถึงระดับของภาษาได้ง่ายและคอยตรวจสอบความถูกต้องของการสืบทอดได้

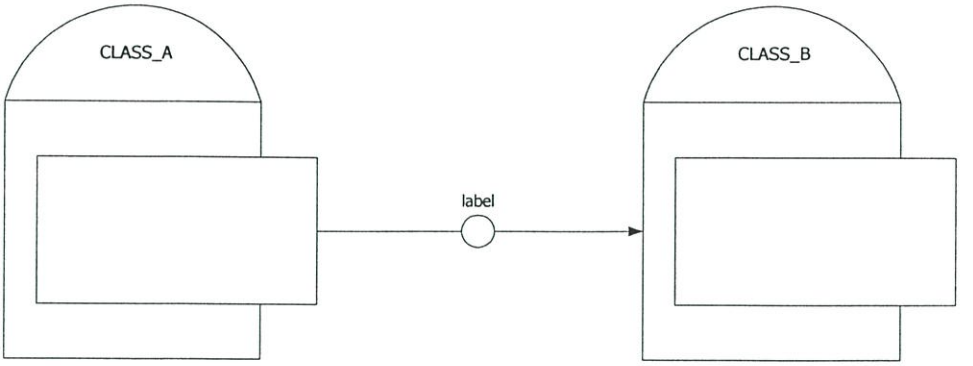


รูปที่ 4.13 ก) แสดงสัญลักษณ์ของการสืบทอดรูปแบบของการอิมพลีเมนต์แบบเดี่ยว
ข) แสดงสัญลักษณ์ของการสืบทอดรูปแบบของการอิมพลีเมนต์แบบหลายระดับ
ค) แสดงสัญลักษณ์ของการสืบทอดรูปแบบของการอิมพลีเมนต์ในระดับคลาสย่อย

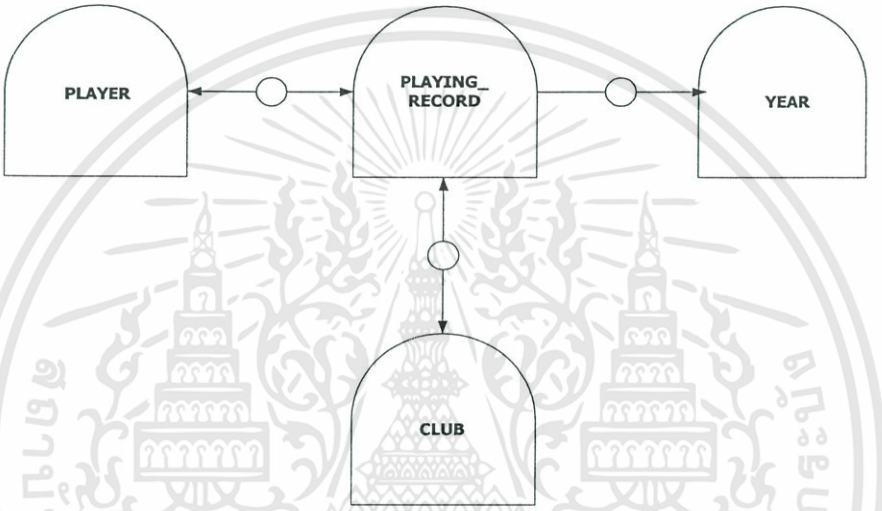
2) แอสโซซิเอชัน (Association)

การแสดงความสัมพันธ์ในระดับพื้นฐาน ซึ่งแสดงระหว่างคลาส หรือ วัตถุ ลักษณะคล้ายกับการแสดงความสัมพันธ์แบบทั่วๆไปที่พบเห็นกันในการออกแบบระดับตรรกะ เช่น ความสัมพันธ์ของ LDS หรือ อี-อาร์ เป็นต้น สัญลักษณ์ที่ใช้มีลักษณะเป็นเส้นประกอบด้วยลูกศรและมีวงกลมเล็กอยู่ตรงกลางเส้นพร้อมรายละเอียด (ดังแสดงในรูปที่ 4.14) ลักษณะของการแสดงความสัมพันธ์ลากจากจุดเริ่มต้นไปยังปลายทาง ซึ่งการลากนี้ต้องคำนึงถึงความสัมพันธ์ของกันและกัน เช่น การแสดงความสัมพันธ์ของคลาส YEAR คลาส CLUB คลาส PLAYER และคลาส PLAYING-RECORD (ดังแสดงในรูปที่ 4.15) เป็นต้น โดยความสัมพันธ์มิได้มีเพียงด้านเดียวเท่านั้น เช่น คลาสของ PLAYER สัมพันธ์กับคลาส PLAYING-RECORDและคลาส PLAYING-RECORD ก็มีความสัมพันธ์กับคลาส PLAYER เช่นเดียวกัน (สังเกตจากลูกศร)เป็นต้น แต่ถ้าให้ชัดเจนควรบอกระดับของความสัมพันธ์ เช่น 1:1 1:M หรือ M:N เป็นต้น ซึ่งจะแสดงอยู่บนเส้นใกล้กับคลาสหรือวัตถุที่สัมพันธ์ด้วย (ดังแสดงในรูปที่ 4.16 (ง))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงความสัมพันธ์ระหว่างคลาส

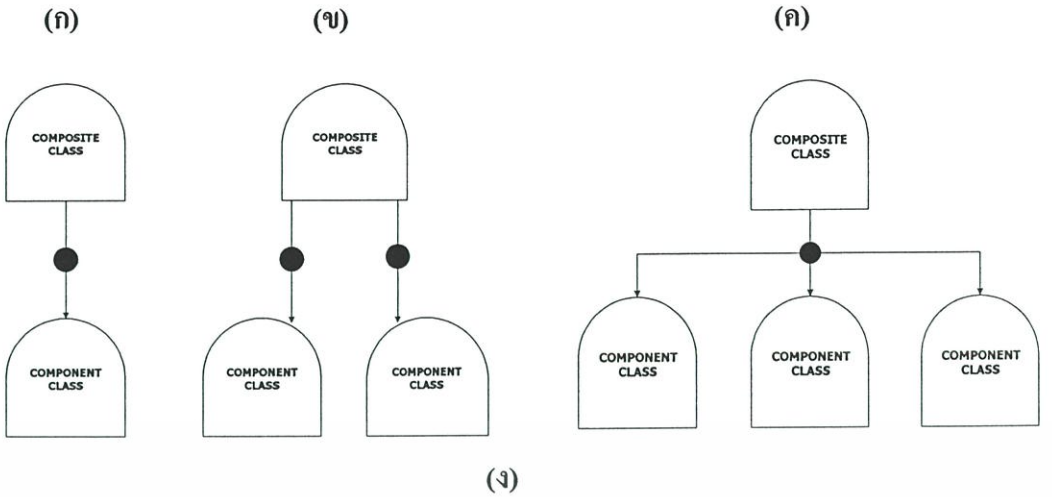


รูปที่ 4.15 แสดงความสัมพันธ์แบบ Ternary ที่เกิดขึ้นในคลาส Playing-record

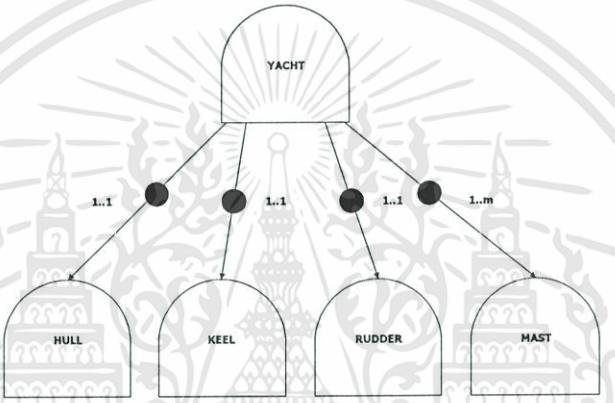
3) การประกอบกันของคลาส (Aggregation)

การประกอบกันของคลาสเป็นความสัมพันธ์อีกลักษณะหนึ่งซึ่งแตกต่างจากความสัมพันธ์โดยทั่วไป กล่าวคือ ความสัมพันธ์ลักษณะนี้จะอยู่ในรูปขององค์ประกอบหรือส่วนขยายที่มีต่อคลาสหรือวัตถุที่เกี่ยวข้อง แต่ละคลาสสามารถที่จะมีองค์ประกอบหรือส่วนขยายได้มากกว่าหนึ่งคลาส เช่น คลาส A อาจมีคลาส B, C, D เป็นส่วนขยายของคลาส A เป็นต้น ประโยชน์ของส่วนที่เพิ่มเติมสามารถทำให้คลาสหลักมีความสมบูรณ์มากขึ้น เช่น คลาสของ YACHT มีคลาส HULL คลาส KEEL คลาส RUDDER และคลาส MAST เป็นส่วนขยาย (ดังแสดงในรูปที่ 4.16 (ง)) เป็นต้น สัญลักษณ์จะมีลักษณะคล้ายกับแอสโซซิเอชั่น แต่จะแตกต่างกันตรงที่วงกลม ซึ่งจะเป็นวงกลมทึบ (ดังแสดงในรูปที่ 4.16) ถ้าสังเกตพบว่าสิ่งที่กล่าวมานี้จะมีความคล้ายคลึงกับสิ่งที่ได้กล่าวไปในบทที่ 3 ที่เป็นทฤษฎีของ James Rumbaugh

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ง)

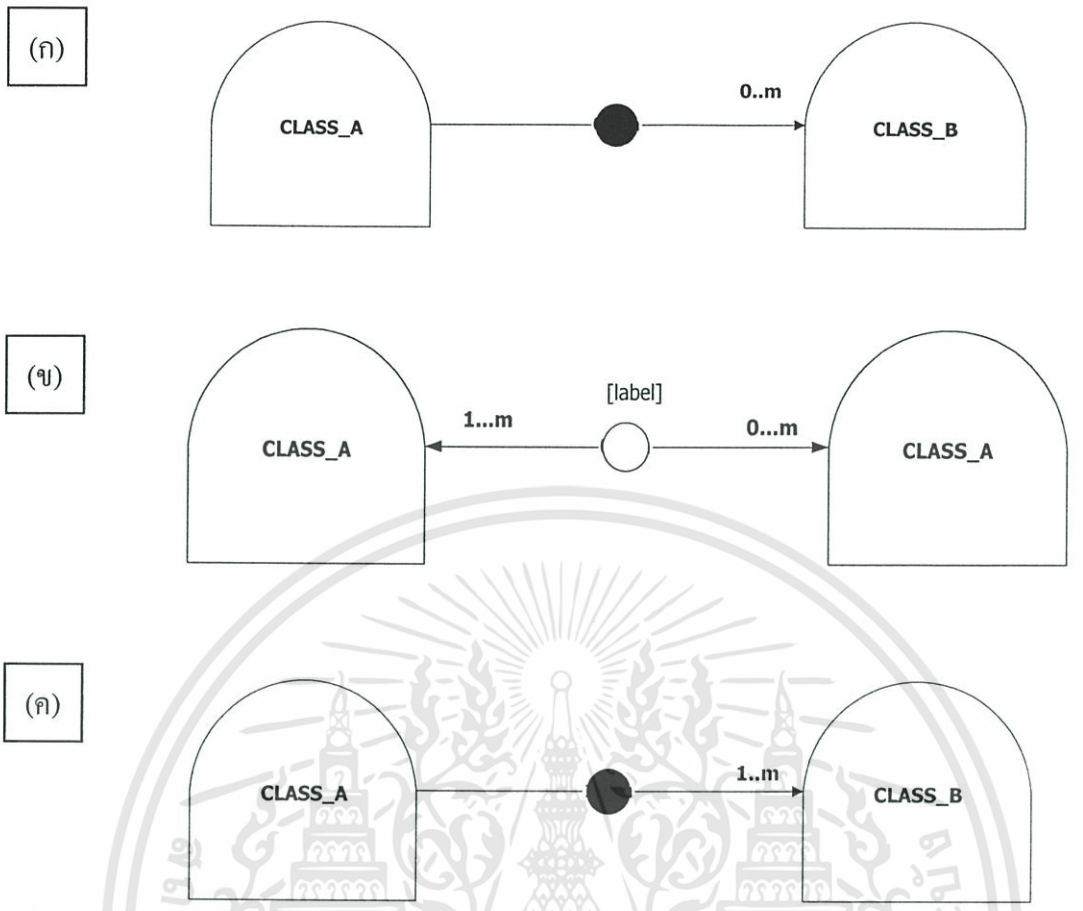


รูปที่ 4.16 ก) แสดงความสัมพันธ์แบบเดี่ยว
ข) แสดงความสัมพันธ์แบบหลายคลาส
ค) แสดงความสัมพันธ์แบบหลายคลาสและหลายทางเลือก
ง) แสดงความสัมพันธ์ที่ระบุความสัมพันธ์

4) Cardinality and existence constraints

Cardinality and existence constraints คือ สิ่งที่บ่งบอกถึงระดับของความสัมพันธ์ที่เกิดขึ้นระหว่างคลาสหรือวัตถุ ซึ่งสามารถพบเห็นได้ทั่วไป เช่น 1:1 , 1:M , 0:M หรือ M:N (บางครั้งอาจจะอยู่ในรูปของ 1..1 , 1..M, 1..N, 0..M ฯลฯ) เป็นต้น สิ่งต่าง ๆ เหล่านี้สามารถที่จะชี้เฉพาะลงไปได้ว่าความสัมพันธ์ที่เกิดขึ้นอยู่ในระดับใด และมีความสำคัญแค่ไหน เช่น ถ้าบนเส้นความสัมพันธ์แสดงในลักษณะของ 0..M มีความหมายว่า มีความสัมพันธ์ก็ได้หรือไม่มีก็ได้ (Optional) โดยระดับความสัมพันธ์เริ่มตั้งแต่ 0 จนถึง M (ดังแสดงในรูปที่ 4.17(ก)) แต่ถ้าเป็น 1..M หมายความว่า จำเป็นต้องมีอย่างน้อย 1 จนถึง M (ดังแสดงในรูปที่ 4.17(ข)) หรือเรียกว่า "Mandatory" เป็นต้น ในรูปที่ 4.17(ค) จะแสดงถึงความสัมพันธ์ของคลาส A กับคลาส B อย่างน้อย 0 จนถึง M และคลาส B สัมพันธ์กับคลาส A อย่างน้อย 0 จนถึง M เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 ก) แสดงสัญลักษณ์ของความสัมพันธ์แบบ Cardinalities แบบมีเงื่อนไข 1:M
ข) แสดงสัญลักษณ์ของความสัมพันธ์แบบ Cardinalities แบบจำเป็นต้องมี 1:M
ค) แสดงสัญลักษณ์ของความสัมพันธ์แบบ Cardinalities แบบ Association Cardinality

4.2.2 การกำหนดรายละเอียดของคลาส (Class Specification (CS))

คลาสที่ปรากฏในแผนภาพเป็นเพียงรูปแบบหรือโมเดลเท่านั้น ยังไม่สามารถนำมาใช้ประกอบการพัฒนาโครงงานได้ เนื่องจากยังขาดรายละเอียดต่าง ๆ ของคลาสแต่ละตัวและยังไม่สามารถนำไปสู่ขั้นตอนถัดไปได้ Class Specification หรือ CS เป็นรายละเอียดของคลาสที่ระบุถึงสิ่งต่าง ๆ ที่อยู่ภายในกับคลาส เช่น แอตทริบิวต์และเซอร์วิสที่อยู่ภายในความสัมพันธ์ที่เกิดขึ้นกับคลาส เป็นต้น ซึ่งรายละเอียดจะอยู่ในรูปของเอกสารและสามารถแสดงได้ 2 รูปแบบ คือ รูปแบบเต็ม (formal) กับรูปแบบย่อ (informal)

เอกสารรูปแบบเต็มแสดงตามหลักไวยากรณ์ทางภาษาที่มีลักษณะเป็นโครงสร้างทั้งหมด ซึ่งภายในประกอบด้วย ชื่อคลาส ลักษณะโดยทั่วไปของคลาส ความสัมพันธ์แบบ Association และ/หรือ Aggregation (ถ้ามี) และรวมถึงระดับของความสัมพันธ์ (เช่น optional, mandatory,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

cardinality) ของเซอร์วิสภายในที่เป็นแบบ private และ public พร้อมกับค่าคงที่ต่าง ๆ ภายในคลาส (ดังแสดงในรูปที่ 4.18)

เอกสารแบบย่อหรือไม่เต็มรูปแบบแสดงรายละเอียดภายในเอกสารไม่อยู่ในรูปของหลักไวยากรณ์เหมือนกับเอกสารรูปแบบเต็มแต่จะอยู่ในรูปของภาษาธรรมชาติหรือคล้ายกับการจดบันทึกมากกว่าและกล่าวถึงรายละเอียดต่าง ๆ แบบย่อแทน

Class Specification เป็นหนึ่งในหลายเอกสารที่กล่าวถึงรายละเอียดต่าง ๆ เช่น ObjectCharts มีรายละเอียดคล้ายกันแต่ต่างกันตรงที่ที่มีรายละเอียดมากกว่า เป็นต้น แต่เอกสารนี้นับเป็นเอกสารแรกที่เชื่อมการทำงานระหว่างนักออกแบบกับโปรแกรมเมอร์ให้เข้าใจกันและสามารถนำไปเขียนเป็นอัลกอริทึมต่อไปได้ จากที่กล่าวมาทั้งหมดของ CS ไม่มีการกล่าวถึงรายละเอียดการสืบทอดคุณสมบัติของคลาส ซึ่งจะนำไปแสดงในส่วนของโมเดลการสืบทอดคุณสมบัติแทน (Inheritance Model – IM) ส่วนรายละเอียดของอัลกอริทึมภายในคลาสสามารถขยายต่อไปได้อีก โดยใช้ Service Structure Model (SSM) ซึ่งรายละเอียดของ ObjectCharts Inheritance Mode และ SSM จะกล่าวในหัวข้อถัดไป

4.2.2.1 การกำหนดรายละเอียดของคลาสแบบไดอะแกรม (Class Specification Diagrams)

รายละเอียดต่าง ๆ ที่ปรากฏในเอกสารสามารถแสดงอยู่ในรูปของสัญลักษณ์ได้ ซึ่งเรียกว่า “pinout” ซึ่งจะแสดงในส่วนของเซอร์วิส โดยใช้วิธีของการส่งผ่านข้อความ (message-passing) เช่น การปรับปรุงยอดเงินฝากของลูกค้าที่ประกอบด้วย คลาส BANK_ACCOUNT คลาส DOLLAR คลาส DEPOSIT ฯลฯ ในที่นี้จะกล่าวถึงคลาส BANK_ACCOUNT ภายในคลาสประกอบด้วย 3 เซอร์วิส คือ new , make_deposit และ give_balance เมื่อมีการฝากเงินเซอร์วิส new จะส่งหมายเลขบัญชีไปยัง Client ที่ร้องขอ จากนั้นคลาส Deposit จะส่งยอดเงินฝากไปยังเซอร์วิส make_deposit เพื่อปรับปรุงยอดบัญชีของลูกค้า หลังจากนั้นจะส่งต่อไปยังเซอร์วิส give_balance เพื่อส่งยอดไปยังหน้าจอของ Client เพื่อให้พนักงานส่งยอดเงินฝากที่ปรับปรุงเรียบร้อยแล้วให้กับลูกค้าต่อไป (ดังแสดงในรูปที่ 4.19) เป็นต้น จากตัวอย่างข้างต้นสังเกตว่าการแสดงให้อยู่ในรูปของสัญลักษณ์นั้น ต้องอาศัยรายละเอียดของ CS จึงจะสามารถแสดงออกมาได้ ซึ่งข้อดีของการใช้สัญลักษณ์เข้ามามีส่วนร่วม คือ ทำให้เข้าใจได้ง่ายและรวดเร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

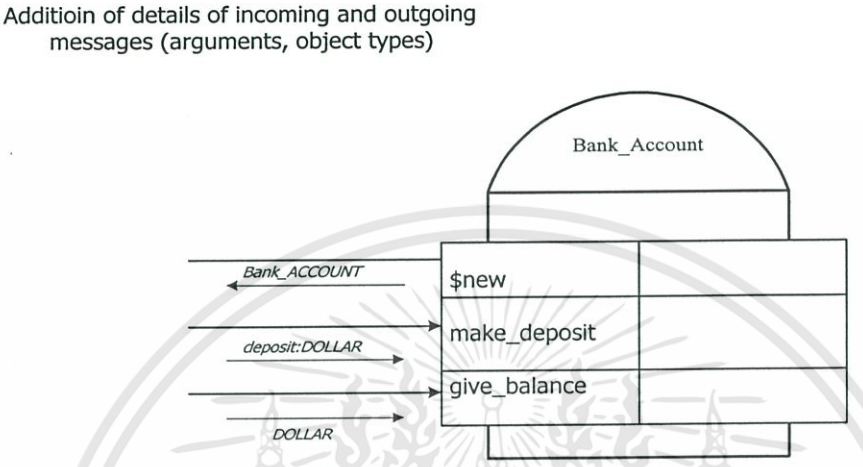
```
abstract class CLASS NAME
    -- description of class responsibility
specialize
    CNAME_A
visible
    service1: CNAME_B
        -- description of property
        -- (i.e. service which returns an object)
        precondition
        postcondition
        end
    service2: (a:CNAME_C)
        -- description of operation
        precondition
        postcondition
        end
secret
    service3: a:CNAME_D
        -- description of property
        precondition
        postcondition
        end
    service4: (a:CNAME_E)
        -- description of operation
        precondition
        postcondition
        end
    label_a: component CNAME (0..m)
    label_b: CNAME.symmetric (0..m)
invariant
    invariant clause
end
```

รูปที่ 4.18 แสดงรูปแบบเอกสารของ Class Specification โดยใช้โครงสร้างทางไวยากรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การกำหนดรายละเอียดของเซอร์วิส (Service Specification)

การกำหนดรายละเอียดของเซอร์วิส คือ ส่วนขยายที่ต่อจาก CS แต่จะเน้นไปทางเซอร์วิสแทน ดังนั้นรายละเอียดจะกล่าวถึงการทำงานของเซอร์วิสทุกขั้นตอนตั้งแต่ต้นจนถึงสุดท้าย โดยจะทำการบันทึกตั้นแต่ค่าอินพุตเอาต์พุตและการทำงานทั้งหมดของเซอร์วิสแต่ละตัวในแต่ละคลาส ซึ่งแทนด้วยรูปภาพดังปรากฏในรูปที่ 4.19



รูปที่ 4.19 แสดงการทำงานของเซอร์วิสในคลาส Bank_Account

รูปที่ 4.19 แสดงการทำงานของเซอร์วิส make_deposit ในคลาส Bank_Account ที่อธิบายถึงกระบวนการภายในที่มีการติดต่อกับค่าคงที่ภายในเพื่อเปลี่ยนแปลงค่าและส่งค่าที่ปรับปรุงแล้วไปยังเซอร์วิส give_balance เพื่อนำไปปรับปรุงค่าในบัญชีของลูกค้าและเซอร์วิสนี้ก็จะส่งค่าปัจจุบันไปยังคลาส DOLLAR ต่อไป สัญลักษณ์ที่แสดงนี้เรียกว่า "Service Structure Model (SSM)" ในบางครั้งการอธิบายเซอร์วิสภายในคลาสโดยอาศัย SSM ก็ไม่สามารถที่จะอธิบายได้หมดด้วยการแทนสัญลักษณ์ ดังนั้นในแต่ละครั้งจึงควรมีเอกสารแนบต่อท้าย ซึ่งเรียกว่า "Contracts" ที่ประกอบด้วย ชื่อเซอร์วิส รายละเอียดหรือลักษณะเฉพาะตัวของเซอร์วิสและเงื่อนไขก่อนหลังของการประมวลผล เป็นต้น การบันทึกจะอยู่ในรูปของข้อความเชิงบรรยายทั้งหมด (ดังแสดงในรูปที่ 4.20)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Service Specified by Contracts

Service_name
        --comments
        Precondition
                -- Boolean Expression or English Text
        Postcondition
                -- Boolean Expression or English Text
end
```

รูปที่ 4.20 แสดงรายละเอียดของเซอร์วิส

4.2.4 อีเวนท์โมเดล (Event Models (EMs))

ภายใน Contracts หรือ SSMs คือการแสดงพฤติกรรมของคลาสในรูปของเซอร์วิส แต่ในส่วนของ O/C Model นั้นทั้งสองส่วนไม่ถูกกล่าวถึงซึ่งต่างจากอีเวนท์โมเดลที่กล่าวถึงพฤติกรรมของคลาส โดยใช้ลูกศรพร้อมกับระบุหมายเลขกำกับและชื่อเหตุการณ์เป็นตัวกำหนดเหตุการณ์ที่เกิดขึ้นตามลำดับก่อนหลังกับคลาสที่สัมพันธ์ด้วย (ดังแสดงในรูปที่ 4.21) โดยส่งข้อความพร้อมกับรายละเอียดไปยังคลาสที่มีความสัมพันธ์ด้วย ถ้าคลาส A ส่งข้อความไปยังคลาส B เพื่อขอใช้ Service_a จากคลาส B จากนั้น B ส่งข้อความไปยังคลาส C เพื่อขอใช้ Service_b ที่อยู่ในคลาส C สังเกตว่าการทำงานของ Event Model จะทำตามลำดับหมายเลขที่กำกับไว้ให้ เป็นต้น ประโยชน์ของ Event Model คือสามารถนำไปกำหนดแผนงาน (Scenario) ของการพัฒนาได้ ซึ่งการเขียนแผนงานบางวิธีเรียกว่า "อีเวนท์เทร็ก (Event Track)" [28] โดยอีเวนท์เทร็กทำหน้าที่คล้ายกัน กล่าวคือ แผนงานจะบันทึกเหตุการณ์ต่าง ๆ ที่เกิดขึ้นระหว่างผู้ใช้กับระบบ และผู้พัฒนาจะนำแผนงานเหล่านี้ไปกำหนดหรือปรับปรุงเซอร์วิสให้กับคลาสแต่ละตัวอีกครั้งหนึ่ง ซึ่งการบันทึกจะอยู่ในรูปของภาษาแบบทั่วไป หรือภาษาธรรมชาติ ตัวอย่างการทำงานของแผนงาน เช่น คลาสแสดงผลลัพธ์ของกราฟฟิก โดยการแสดงผลจะต้องแสดงชื่อแฟ้มข้อมูลและรายละเอียดต่าง ๆ ซึ่งภายในคลาสไม่มีข้อมูลเกี่ยวข้องอยู่แต่อย่างใด มีเพียงแต่เชื่อมไปยังคลาสที่เกี่ยวข้องเท่านั้น ดังนั้นเพื่อให้เป็นไปตามเอาต์พุตที่ต้องการคลาสกราฟฟิกจึงต้องส่งข้อความไปยังคลาสต่าง ๆ เพื่อขอข้อมูลมาแสดงผล จากตัวอย่างสังเกตว่าเหตุการณ์ที่เข้ามากระทบกับคลาสใดคลาสหนึ่งนั้นจะทำให้คลาสที่ถูกกระทบถูกกระตุ้นให้ทำงานตามไปด้วย เปรียบได้ว่าแผนงานนั้นเหมือนกับ "Script" ที่คอยกระตุ้นการทำงานของคลาสต่าง ๆ ในระบบไม่ว่าจะเป็นระบบหลักหรือระบบย่อย

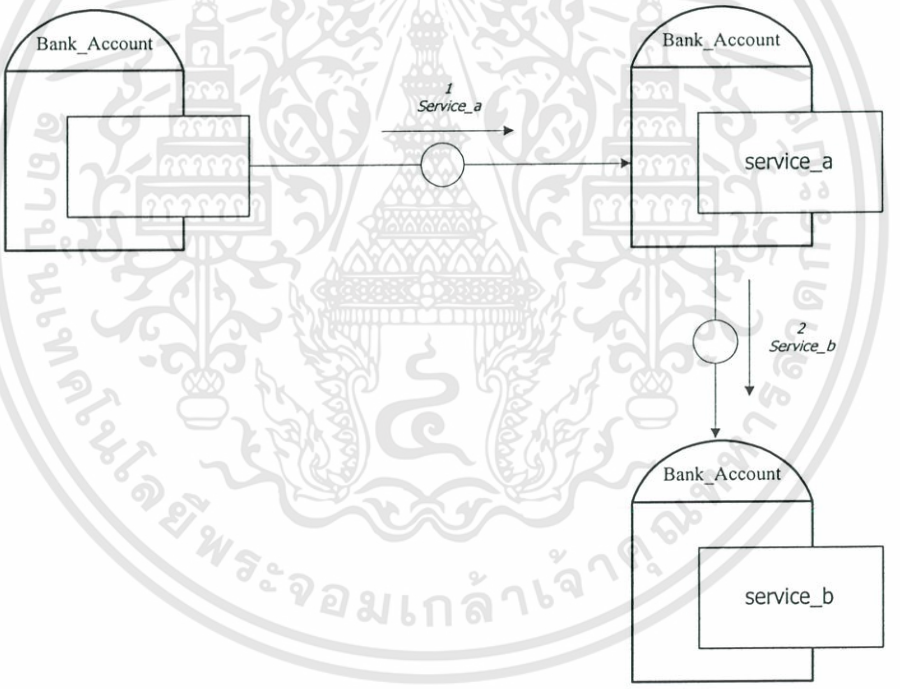
ความสามารถของแผนงานนอกจากจะบอกถึงเหตุการณ์ที่เข้ามากระทบกับคลาสในระบบแล้วยังสามารถนำมาใช้ประโยชน์ร่วมกับ EMs ได้อีกด้วย กล่าวคือ สามารถนำมากำหนดคลาสในระบบให้ทราบว่าควรจะเริ่มต้นที่คลาสใดเป็นคลาสแรกและเหตุการณ์แรกเริ่มขึ้นที่ใด นอกจากนี้ยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวมไปถึงการกำหนดเซอร์วิสต่าง ๆ ที่อยู่ภายใน O/C Models และยังทำให้ผู้พัฒนาสามารถตรวจสอบความต้องการของผู้ใช้อีกครั้งหนึ่งด้วย

แทร็ก (Trace) [28], [37] คือ การอธิบายการติดต่อกันระหว่างวัตถุ ในรูปของข้อความซึ่งมีรูปแบบดังนี้ <a, s, b> โดยอักษรตัวแรก a กล่าวถึงวัตถุต้นทางที่แจ้งความจำนงในการขอใช้เซอร์วิส s หมายถึงเซอร์วิสที่ต้องการใช้งาน สุดท้ายอักษร b คือวัตถุปลายทางที่ต้องการใช้เซอร์วิส เช่น <ps, fill(W1), t1> หมายถึง คลาส Primary_Settler มีความต้องการใช้เซอร์วิส fill ที่อยู่ในวัตถุ t1 ของคลาส Tank พร้อมกับส่งข้อมูลที่อยู่ในตัวแปร W1 เป็นต้น ประโยชน์ของ Trace คือ แสดงรายละเอียดของเหตุการณ์ที่เกิดขึ้นกับวัตถุให้สามารถอ่านได้ง่ายขึ้น

ทั้ง Scenario และ Trace เป็นการแทนการติดต่อระหว่างกันของฟังก์ชั่นที่อยู่ภายในคลาสให้สามารถเข้าใจง่ายและทำงานตามความต้องการได้ถูกต้อง ซึ่งทั้งสองส่วนนี้สามารถที่จะใช้แยกกันหรือรวมกันได้ภายในระบบเดียวกัน



รูปที่ 4.21 แสดงรูปแบบของการวาดอีเวนท์โมเดลพร้อมคำอธิบาย

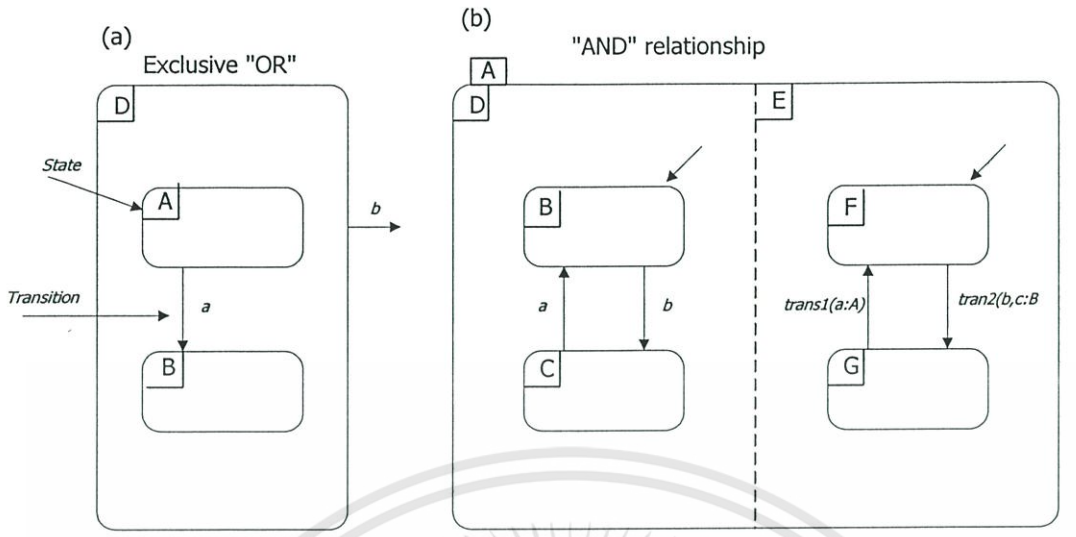
4.2.5 ออบเจ็กต์ชาร์ต (ObjectCharts)

ออบเจ็กต์ชาร์ต คือ การแสดงพฤติกรรมและความเปลี่ยนแปลงของวัตถุ หรือ เรียกว่า "สเตทไดอะแกรม (State Diagram)" สถานะที่แสดงในออบเจ็กต์ชาร์ตเป็นการแสดงในกลุ่มของออบเจกต์มากกว่าในกลุ่มของคลาสที่แสดงอยู่ใน O/C Model ซึ่งการแสดงสเตทมีอยู่หลายแบบที่แตกกต่างกันออกไปตามวิธีที่นำมาใช้ วิธีของ Rumbaugh เรียกว่า ไดนามิกโมเดล (Dynamic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Model) ส่วนวิธีของ Code/Yourdon เรียกว่า สเตทไดอะแกรม เป็นต้น แต่วิธีนี้เรียกว่าออบเจ็กต์ชาร์ต หรือ STDs สัญลักษณ์ที่ใช้มีลักษณะคล้ายกัน คือ สัญลักษณ์ของสเตทเป็นรูปสี่เหลี่ยมผืนผ้าปลายมนทั้งสี่ด้านและมุมบนทางด้านซ้ายแสดงชื่อสเตท ส่วนการเปลี่ยนสเตทจากที่หนึ่งไปยังอีกที่หนึ่งจะถูกกำหนดทิศทางด้วยลูกศรซึ่งเรียกว่า "Transition" พร้อมกับแสดงชื่อของ Transition และข้อมูลเข้า-ออกที่ส่งมาจากสเตทนั้น ๆ ในบางกรณีสเตทอาจมีสเตทย่อยซ้อนอยู่พร้อมกับเงื่อนไข เช่น "Exclusive", "OR" และ "AND" เป็นต้น กรณีที่เป็นเงื่อนไข AND ภายในสเตทหลักจะถูกแบ่งด้วยเส้นประระหว่างสเตทย่อยทั้งสอง (ดังแสดงในรูปที่ 4.22) สเตทหลักเรียกว่า "ซุปเปอร์สเตท (Superstate)" ส่วนรายละเอียดของ Transition นั้นมีรูปแบบดังนี้ Transition_name(in:INPUT) ซึ่งประกอบด้วยชื่อข้อมูลเข้า/ออกพร้อมชื่อของสเตทหรือTransition_name/FORMAL_NAME.service_name(in:INPUT) ซึ่งระบุรายละเอียดมากกว่าเดิม สิ่งที่เพิ่มเข้ามา คือ ชื่อของคลาสและเซอร์วิสที่แสดงถึงความชัดเจนในการเข้าถึงข้อมูล โดยแสดงหลังเครื่องหมาย "/" ส่วนข้อมูลเข้า/ออกแสดงเหมือนเดิม แต่ละ Transition ที่แสดงจะถูกบันทึกรายละเอียดต่าง ๆ ในรูปแบบของตาราง ซึ่งเรียกว่า "Transition Table" (ดังแสดงในรูปที่ 4.23) ซึ่งตารางนี้จะบันทึกรายละเอียดทุกอย่างที่เกิดขึ้นในแต่ละสเตทที่ Transition เข้าไปกระทบ ซึ่งรูปแบบประกอบด้วยชื่อ คลาส ชื่อระบบย่อย ประโยชน์ของการใช้ออบเจ็กต์ชาร์ต คือ ทำให้ผู้พัฒนาสามารถตรวจสอบความถูกต้องโดยทั่ว ๆ ไปของ Semantic ได้ง่ายและแม่นยำมากขึ้น นอกจากนี้ยังสามารถทราบถึงขั้นตอนการเปลี่ยนแปลงสเตทของวัตถุและทราบถึงแหล่งข้อมูลที่เข้ามากระทบกับสเตทได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 ก) แสดงออบเจ็กต์ชาร์ตของสเตทที่มีเงื่อนไข OR
ข) แสดงออบเจ็กต์ชาร์ตของสเตทที่มีเงื่อนไข AND

TRANSITION SPECIFICATION TABLE	
class CLASS_NAME	subsystem: SUBSYSTEM_NAME
strart state name →	end state name: {Precondition} Transition_name {Postcondition}

รูปที่ 4.23 แสดงรายละเอียดของทรานเซกชั่นของสเตทไดอะแกรม

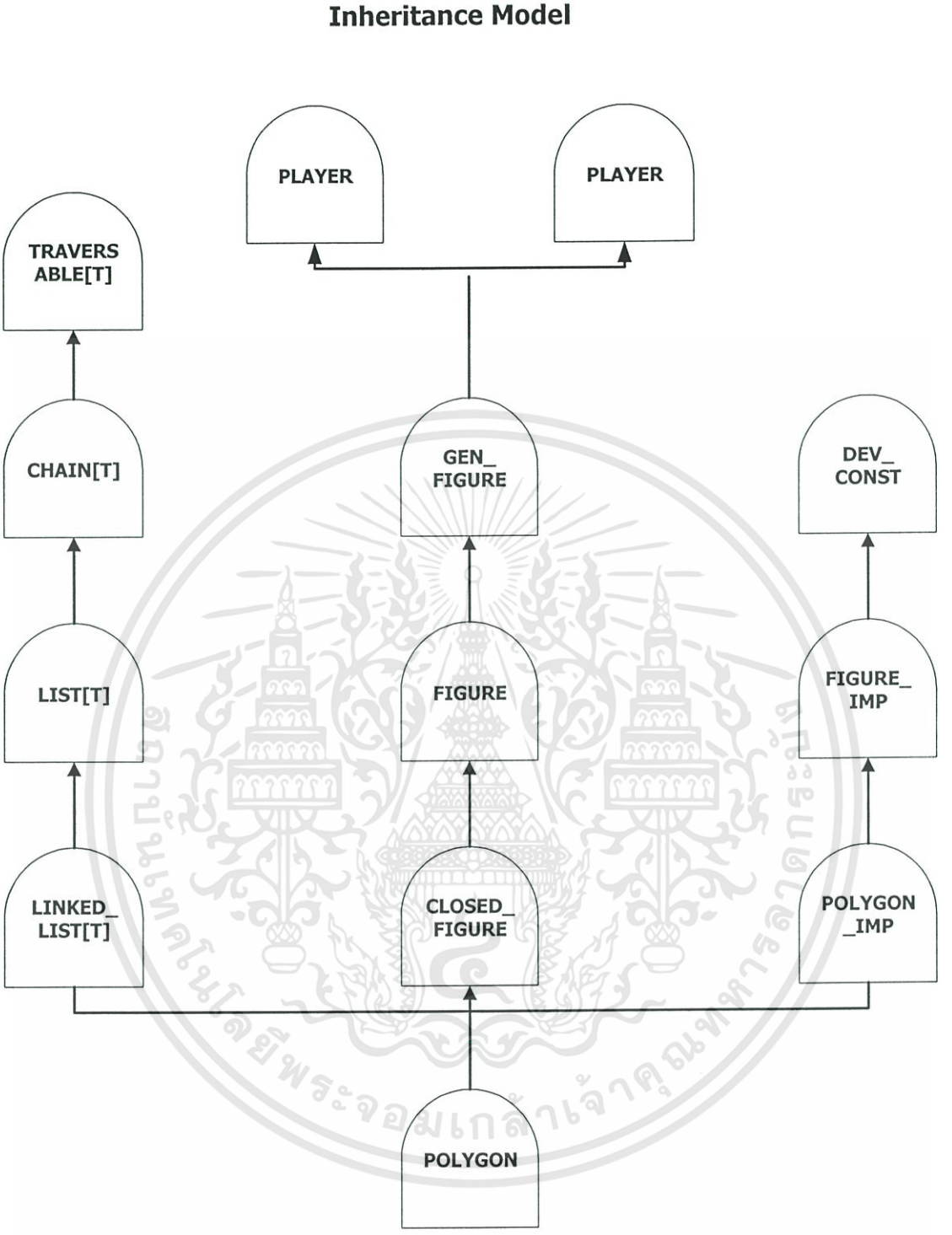
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6 โมเดลการ์สืบทอดคุณสมบัติ (Inheritance Model)

โมเดลของการสืบทอดคุณสมบัติจากคลาสหลักไปยังคลาสลูกที่เกี่ยวข้องกัน แสดงลักษณะจากล่างขึ้นบน คือ คลาสหลักจะอยู่ข้างล่างสุดและค่อย ๆ กระจายขั้นไปเรื่อย ๆ จนสุดปลายทาง โดยอาศัยลูกศรหนาเป็นตัวนำทาง (ดังแสดงในรูปที่ 4.24) ซึ่งการแสดงจะต่างจากที่ได้กล่าวมาแล้วในหัวข้อ 1.2 ใน IM แสดงในส่วนของโมเดลไม่สามารถนำไปใช้กับขั้นตอนการอิมพลีเมนต์ได้ โดยจะเน้นไปทางด้านการแสดงในรูปของการแสดงลำดับขั้นของคลาส (Class Hierarchies) เท่านั้น ซึ่งบางครั้งอาจจะมีบางส่วนไม่ตรงกับสิ่งที่มีอยู่ใน IM ประโยชน์ของ IM ทำให้การอ่านลำดับของคลาสที่มีความสัมพันธ์กันในลักษณะของการสืบทอดได้ง่ายขึ้น

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 แสดงการสืบทอดคุณสมบัติของโมเดลสืบทอดคุณสมบัติ

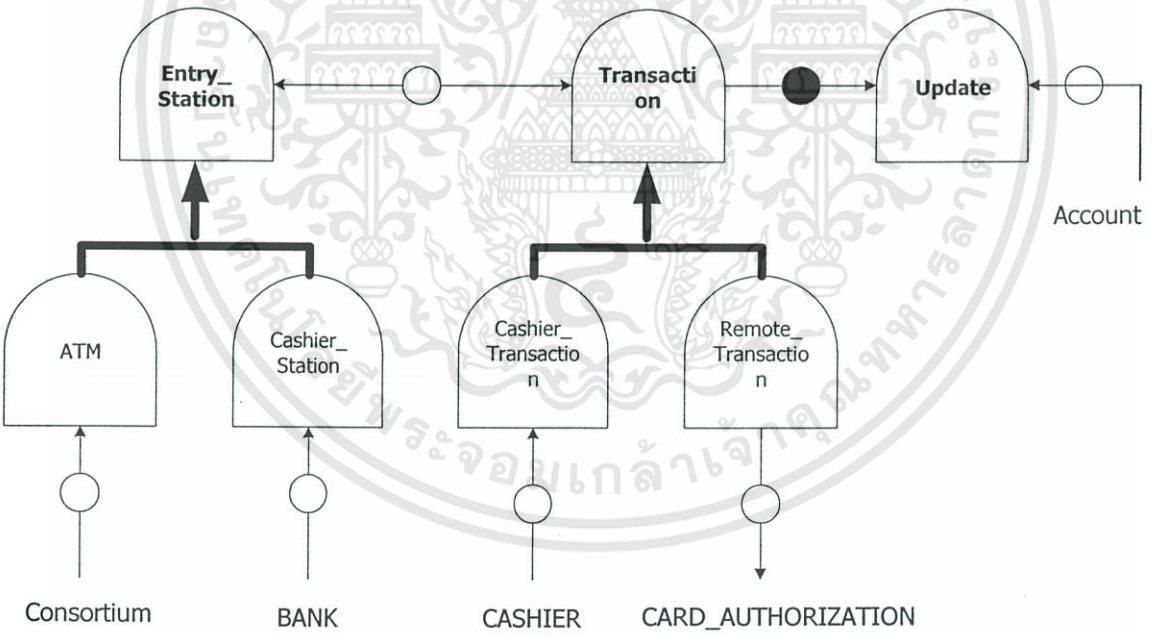
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.7 เทคนิคการจัดการความซับซ้อน (Complexity Management Techniques)

หัวข้อนี้จะกล่าวถึงเทคนิคทั่ว ๆ ไปของการวาดแผนภาพในโครงงานที่พัฒนา โดยเริ่มจาก Sheet, Subsystems, Layering และ Selective Visibility ตามลำดับ ซึ่งการแสดงแต่ละส่วนมีลักษณะคล้ายคลึงกันแต่จะแตกต่างกันที่เทคนิคที่นำมาใช้

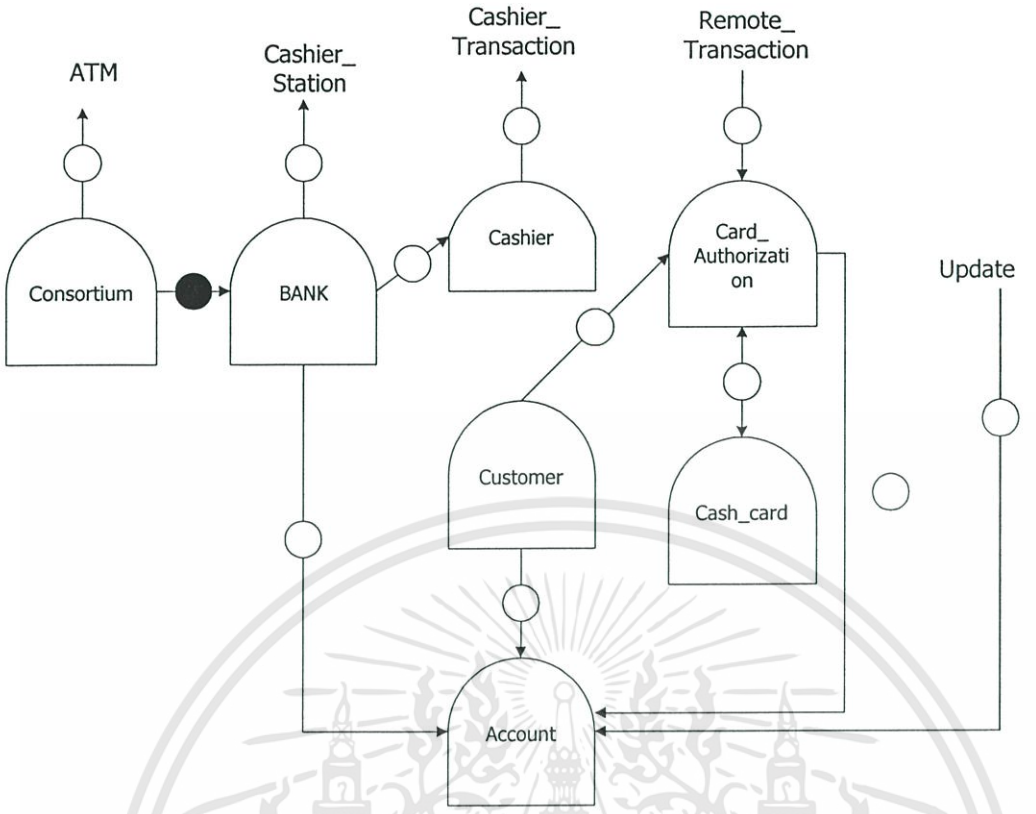
4.2.7.1 ชีท (Sheet)

ปัญหาอย่างหนึ่งที่ผู้พัฒนามักพบกับการออกแบบระบบ คือ ระบบมีขนาดใหญ่และมีคลาสที่เกี่ยวข้องมากเกินกว่าที่จะบันทึกลงได้หมดในหน้ากระดาษเดียว แต่ถ้าบันทึกหมดก็ไม่สามารถอ่านเข้าใจได้หมด ดังนั้นเทคนิคของการใช้ Sheet จึงนับว่ามีความจำเป็นอย่างมากเพราะ Sheet จะแบ่งคลาสใน O/C Model ออกเป็นส่วนย่อย ๆ และสามารถติดตามได้ง่าย วิธีทำเริ่มจากกำหนดขอบเขตของคลาสที่อยู่ใน O/C Model ที่จะแบ่งออกจากกัน จากนั้นนำคลาสที่ถูกแบ่งออกมาเขียนใน Sheet ที่เตรียมไว้พร้อมกับระบุความสัมพันธ์เดิมด้วย เช่น คลาส A เดิมสัมพันธ์กับคลาส B แบบ Association ใน Sheet ที่นำคลาส A ไปก็ยังคงต้องอ้างอิงถึงสิ่งที่เกิดขึ้นกับคลาสด้วย ดังแสดงในรูปที่ 4.25 และ 4.26 เป็นต้น



รูปที่ 4.25 แสดงการแบ่งคลาสในรูปแบบของ Sheet (ส่วนแรก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 แสดงการแบ่งคลาสในรูปแบบของ Sheet (ส่วนหลัง)

4.2.7.2 ระบบย่อย (Subsystem)

โดยทั่วไปการออกแบบระบบใหญ่ ๆ มักจะมีระบบย่อยประกอบอยู่จำนวนหนึ่ง ซึ่งระบบย่อยเหล่านี้เปรียบเสมือนกล่องที่เคยเก็บรายละเอียดต่าง ๆ ของระบบเหล่านั้นไว้ สัญลักษณ์ที่ใช้แสดงถูกแทนด้วยสี่เหลี่ยมผืนผ้าซึ่งประกอบด้วยชื่อของระบบย่อย การติดต่อกันระหว่างระบบย่อยกับคลาสหรือวัตถุที่เกี่ยวข้องกันจะใช้ลูกศรที่เป็นเส้นคู่บอกทิศทางของความสัมพันธ์ (ดังแสดงในรูปที่ 4.28) พร้อมกับมีเอกสารคอยแจกแจงรายละเอียดของแต่ละระบบซึ่งเรียกว่า "Subsystem Responsibility Specification (SRS)" [17] ซึ่งบรรยายถึงลักษณะของแต่ละระบบอย่างชัดเจน (ดังแสดงในรูปที่ 4.27) เช่น ชื่อของระบบย่อย หน้าที่หรือลักษณะงานในระบบและกลุ่มงานหรือระบบย่อยที่มีความเกี่ยวข้องกัน เป็นต้น

การกระจายระบบย่อยจะทำเป็นลำดับชั้นที่อ้างอิงถึงกัน เช่น ในรูปที่ 4.28 สังเกตว่าระดับแรกประกอบด้วย 4 คลาสกับ 1 ระบบย่อย ซึ่งคลาสแต่ละตัวจะมีความสัมพันธ์กัน โดยเฉพาะคลาส B, C และ D มีความสัมพันธ์กับระบบย่อย จากนั้นในระดับต่อมาจะทำการกระจายสิ่งที่อยู่ภายในระบบย่อย ซึ่งพบว่าภายในประกอบด้วยคลาส E, F, G และ H พร้อมกับระบบย่อย ตามลำดับ ถ้าย้อยกลับไปพิจารณาระบบย่อยที่อยู่ในระดับแรกพบว่าระบบมีการติดต่อกันระหว่างคลาส B, C และ D โดยคลาส B และ C ส่งอินพุตเข้า ส่วนคลาส D รับเอาต์พุตจากระบบย่อย จึงทำให้ระดับต่อ

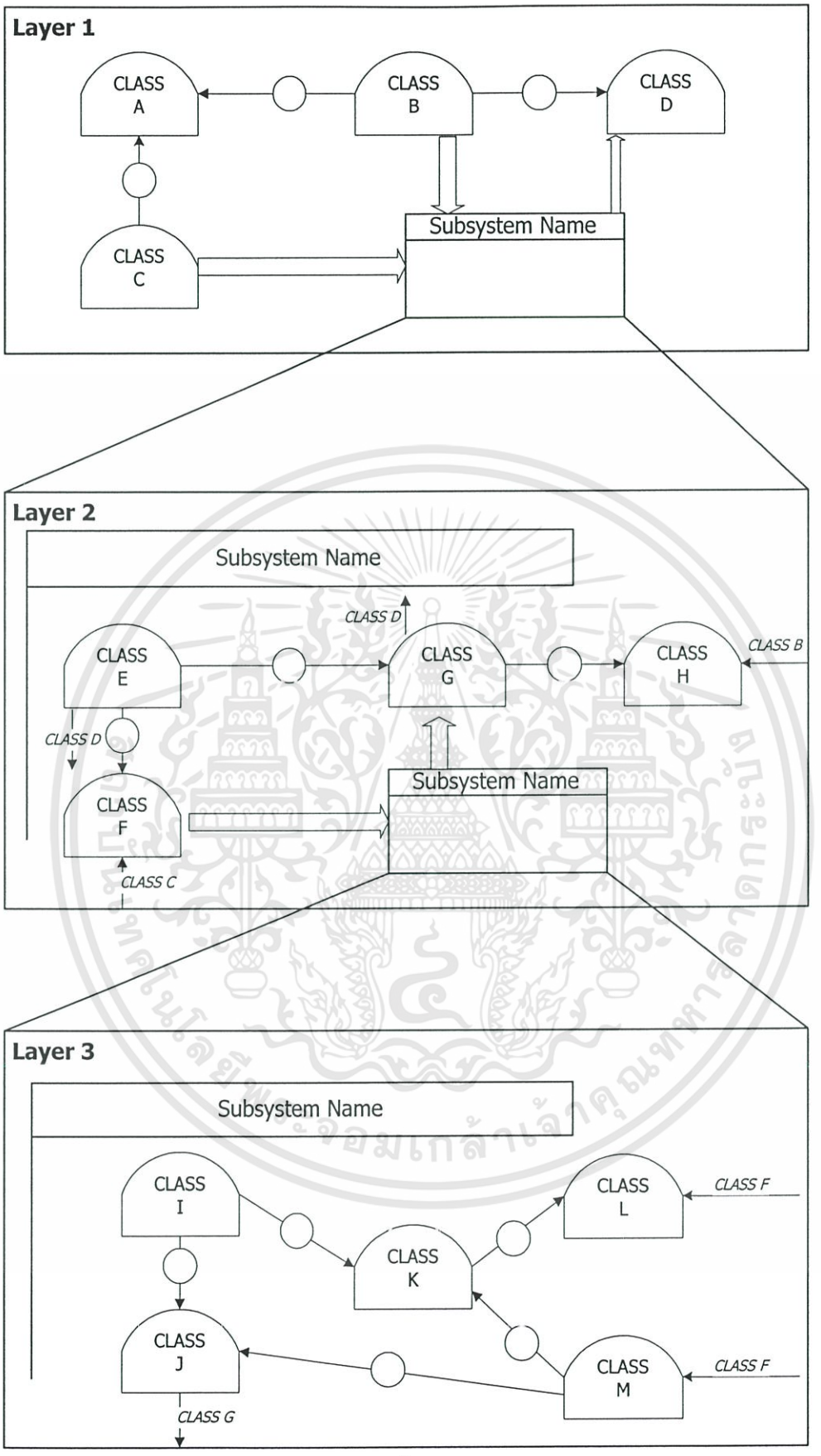
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาต้องแสดงเส้นทางความสัมพันธ์พร้อมกับระบุแหล่งที่มา ดังที่ปรากฏกับคลาส E, F, G และ H ตามลำดับ ซึ่งในระดับที่สองมีระบบย่อยประกอบอยู่ 1 ระบบจึงต้องกระจายไปสู่ระดับถัดไป

Subsystem: Subsystem Name	
Responsibilities	Collaborators
Paragraph explaining the responsibilities of the subsystem	Subsystem 1 Subsystem 2

รูปที่ 4.27 แสดงเอกสารของระบบย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

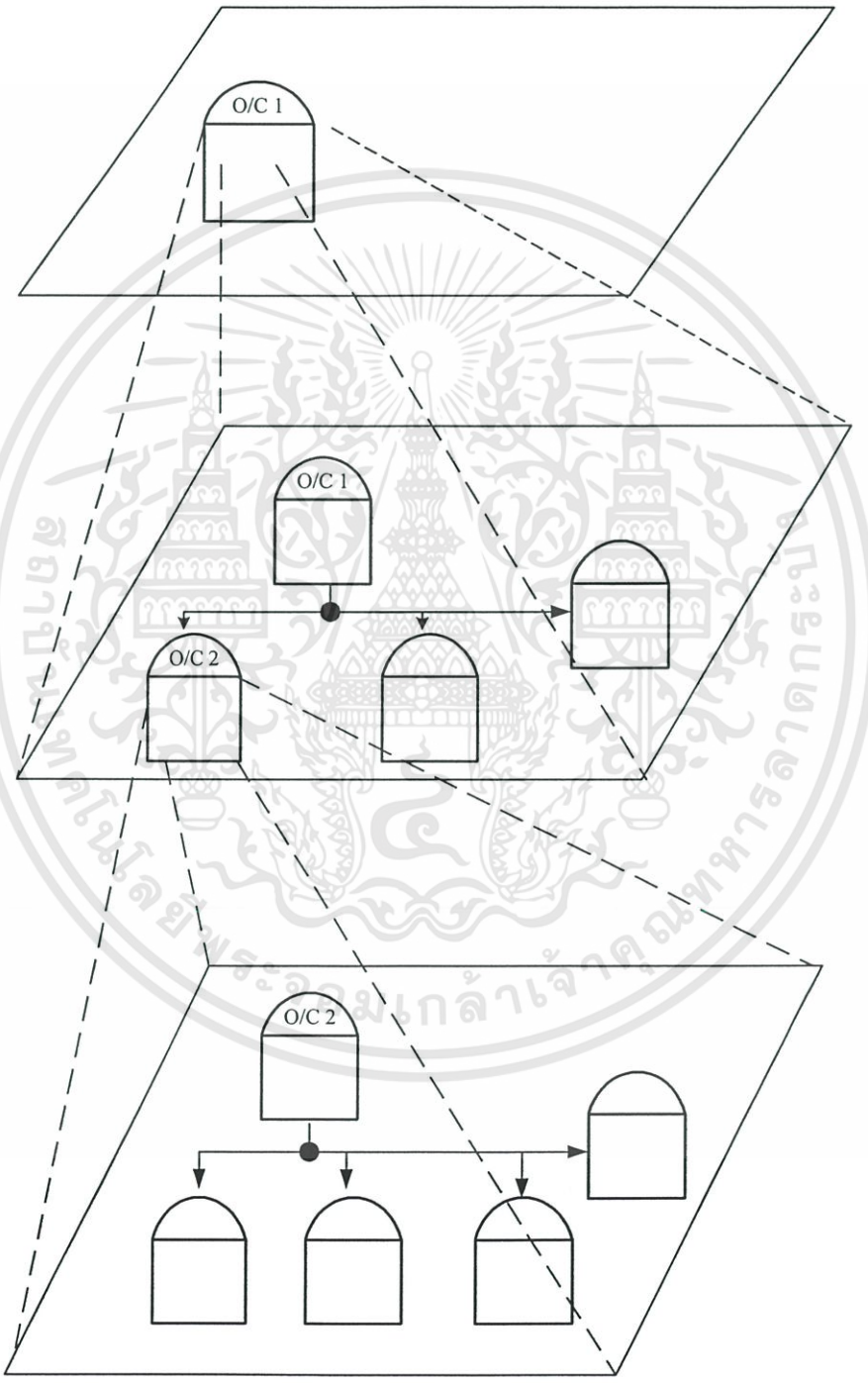


รูปที่ 4.28 แสดงการกระจายระบบย่อยในเลเยอร์ที่ต่างกันของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.7.3 เลเยอร์ (Layering)

เลเยอร์เป็นอีกเทคนิคหนึ่งที่เข้ามามีบทบาทสำหรับการออกแบบระบบ เพื่อคอยจัดการกับระบบที่มีความซับซ้อน โดยมีหลักการคล้ายกับการกระจายกระบวนการของวิธีเชิงโครงสร้าง แต่เป็นการกระจายวัตถุแทน วิธีการกระจายวัตถุเริ่มจากเลเยอร์บนสุดก่อนไปจนกระทั่งเลเยอร์สุดท้ายที่ไม่สามารถกระจายได้อีก (ดังแสดงในรูปที่ 4.29)

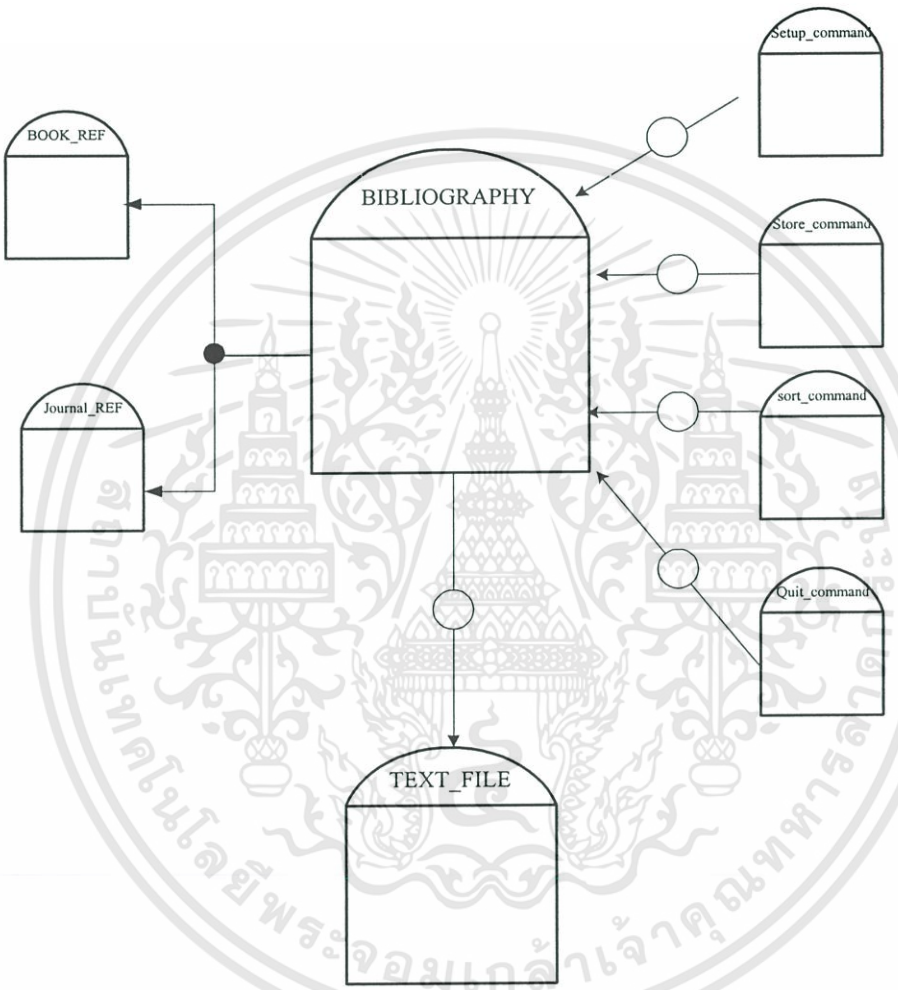


รูปที่ 4.29 แสดงการแบ่งเลเยอร์ของ O/C Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.7.4 Selective visibility

Selective Visibility คือ ไดอะแกรมที่แสดงเฉพาะชื่อและความสัมพันธ์ของวัตถุเท่านั้น โดยจะซ่อนแอตทริบิวต์และเซอร์วิสไว้ ซึ่งเป็นเทคนิคใหม่ที่เกิดขึ้นเฉพาะวิธีนี้เท่านั้น มีประโยชน์สำหรับการทำงานของระบบที่มีขนาดใหญ่ โดยเฉพาะระดับของระบบเพราะการตรวจสอบนั้นไม่จำเป็นต้องทราบถึงรายละเอียดภายในของวัตถุที่มีอยู่ในระบบเพียงแต่ต้องการทราบถึงความสัมพันธ์ของวัตถุเท่านั้น (ดังแสดงในรูปที่ 4.30)



รูปที่ 4.30 แสดง O/C Model ที่มีลักษณะแบบ Selective Visibility

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 กิจกรรมของแบบจำลองน้ำพุ

กิจกรรมของแบบจำลองน้ำพุแบ่งออกเป็น 20 กิจกรรมที่มีรายละเอียดแตกต่างกันออกไปตามความเหมาะสมของกิจกรรมนั้น ๆ ซึ่งจะกล่าวดังต่อไปนี้

กิจกรรมที่ 1 : การกำหนดรายละเอียดของเซอร์วิส (Contract Specification)

การกำหนดรายละเอียดของเซอร์วิส คือ กิจกรรมที่แสดงถึงการเปลี่ยนแปลงภายในเซอร์วิสในรูปของเอกสาร ซึ่งไม่เกี่ยวข้องกับการนำไปสู่ขั้นตอนการอิมพลีเมนต์ หลักการคือนำเซอร์วิสที่อยู่ใน O/C มาแสดงรายละเอียดในรูปของสัญลักษณ์หรือเงื่อนไขต่าง ๆ เช่น เงื่อนไขก่อนเข้าดำเนินการ (precondition) เงื่อนไขหลังเข้าดำเนินการ (post-condition) หรือ การเปลี่ยนแปลงปรับปรุงค่าของตัวแปรในคลาส (Class invariant) เป็นต้น เงื่อนไขเป็นสิ่งหนึ่งที่สามารถบอกถึงวิธีการใช้งานเพื่อให้ได้ผลลัพธ์ที่ต้องการและรายละเอียดที่แสดงในแต่ละเซอร์วิสอาจแสดงมากกว่าหนึ่งเงื่อนไขได้และรายละเอียดสามารถย้ายไปสู่ระดับ Abstraction ได้ เพื่อให้สะดวกต่อการพัฒนาในแต่ละกระบวนการ เพราะระหว่างการพัฒนามักมีการปรับปรุงและเปลี่ยนแปลงอยู่เสมอ ดังนั้นผู้ออกแบบจึงควรตรวจสอบพฤติกรรมต่าง ๆ ที่อยู่ใน O/C ให้ตรงกับแผนงาน ตรงกับความต้องการและปรับปรุงแผนภูมิวัตถุที่มีอยู่อย่างสม่ำเสมอ เมื่อได้รายละเอียดของเซอร์วิสเป็นที่เรียบร้อยแล้วจึงนำมากำหนดค่าของตัวแปรต่าง ๆ ในเซอร์วิส และกำหนดค่าเริ่มต้นหรือค่าก่อนเข้าเงื่อนไขให้กับตัวแปรที่มีอยู่ โดยกำหนดค่าเป็นจริงเสมอเพราะถ้าค่าของตัวแปรมีค่าเป็นเท็จก่อนการใช้งานจะทำให้ตัวแปรนั้น ๆ ไม่ผ่านเข้าไปสู่กระบวนการของเซอร์วิสที่ต้องการได้ ซึ่งการใช้ค่าเริ่มต้นจึงเป็นส่วนสำคัญในทุกสเตทของเซอร์วิสและยังทำให้ซอฟต์แวร์ที่พัฒนามีคุณภาพและถูกต้อง

โอเปเรชั่น คือ การปฏิบัติงานตามคำสั่งที่มีอยู่ในเซอร์วิส ซึ่งบางครั้งอาจมีการเปลี่ยนแปลงค่าของตัวแปรหรือค่าคงที่และอาจทำให้สเตทของวัตถุเกิดการเปลี่ยนแปลงได้ซึ่งการเปลี่ยนสเตทอาจเปลี่ยนทั้งหมดหรือบางส่วน ค่าที่ได้จากการปฏิบัติงานเรียกว่า “ค่าหลังการดำเนินการ” หรือ “Post-Condition” ซึ่งสามารถบอกถึงรายละเอียดของการปฏิบัติที่เกิดขึ้น เช่น ตัวแปร a ก่อนเข้าสู่เซอร์วิสมีค่าเริ่มต้นเป็น 0 แต่หลังจากออกจากเซอร์วิสกลับมีค่าเป็น 10 หรือสเตทของวัตถุ A เริ่มต้นสเตท b หลังจากเสร็จสิ้นการทำงานของเซอร์วิสสเตทเปลี่ยนไปเป็นสเตท c เป็นต้น

รายละเอียดของเซอร์วิสในแต่ละคลาสมีข้อดี คือ ทำให้ผู้พัฒนามีความมั่นใจมากยิ่งขึ้นกับคลาสที่ออกแบบ เพราะการตรวจสอบก่อนและหลังดำเนินการเป็นสิ่งที่ทำให้ผู้วิเคราะห์มีความเชื่อมั่นในระดับหนึ่ง ซึ่งแสดงอยู่ในรูปของโครงสร้างทางภาษาที่เข้าใจง่ายและเครื่องมือที่สนับสนุนได้แก่ Object-Z นอกจากนี้ยังช่วยให้การสืบทอดคุณสมบัติของคลาสมีระเบียบและมีความมั่นใจมากขึ้น เพราะบางครั้งซุปเปอร์คลาสอาจมีรายละเอียดของเซอร์วิสที่เหมือนกันซึ่งทำให้คลาสนั้นเกิดความสับสนในการเรียกใช้งานทั้งก่อนและหลังเรียกใช้งาน เช่น คลาส A ที่สืบทอดคุณสมบัติมาจากซุปเปอร์คลาส AA มีเซอร์วิส a ทั้งคู่ และคลาส C ต้องการเรียกใช้งานเซอร์วิส a จากคลาส A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงทำให้คลาส C ต้องระบุรายละเอียดที่ต้องการให้มีความชัดเจนสำหรับการเรียกใช้มากขึ้น เป็นต้น การพัฒนาระบบงานสารสนเทศที่ใช้วิธีของโมเลสนี้ กิจกรรมที่แสดงรายละเอียดของเซอร์วิสในแต่ละคลาสนับเป็นหนึ่งในกิจกรรมของวิธีนี้ เพราะทำให้การพัฒนามีประสิทธิภาพมากขึ้น

กิจกรรมที่ 2 : ตรวจสอบเอกสาร (Documentation Review)

วัตถุประสงค์ของกิจกรรมนี้ คือ การตรวจสอบเอกสารต่าง ๆ ของการออกแบบเพื่อทำให้แน่ใจก่อนที่จะส่งมอบต่อไป ซึ่งการตรวจสอบอาจทำทุกสัปดาห์ ทุกเดือนหรือทุกปี โดยใช้วิธีการกำหนด “Breakpoint” เข้ามาช่วยระหว่างการพัฒนากระบวนการ เช่น ขั้นตอนสุดท้ายของการอิมพลีเมนต์ในแต่ละรอบอาจมีขั้นตอนการตรวจสอบเสริมขึ้นมาอีกหนึ่งกิจกรรม เป็นต้น

การตรวจสอบเอกสารควรมีหลักเกณฑ์ดังนี้

- ถ้าเป็นเอกสารทั่วไปควรระบุและตรวจสอบชนิดของเอกสารให้ชัดเจน เช่น แยกเอกสารที่เป็นเอกสารหลักหรือเป็นส่วนประกอบให้ชัดเจน เป็นต้น ถ้าเอกสารนั้นเป็นเอกสารหลักหรือเอกสารสำคัญสำหรับองค์กรเอกสารนั้นสามารถนำมาใช้ได้ตลอดทั้งวงจรชีวิตของการพัฒนาระบบได้
- ถ้าเป็นเอกสารประเภทแผนภูมิของระบบ ควรตรวจให้ถูกต้องก่อนนำไปใช้งาน เช่น แผนภาพของ O/C Model ที่แสดงถึงการสืบทอดคุณสมบัติของวัตถุที่แสดงมาจากลำดับชั้นของคลาสควรแสดงคุณสมบัติของคลาสให้ครบถ้วน ก่อนที่จะนำไปใช้กับเครื่องมือช่วยพัฒนาระบบ (CASE Tools)

กล่าวโดยสรุป คือ หลักเกณฑ์ของกิจกรรมนี้จะตรวจสอบเอกสารทั้งหมดที่เกิดขึ้นในการพัฒนาระบบทั้งเอกสารทั่วไปและแผนภูมิ ให้มีข้อผิดผลาดน้อยที่สุด ก่อนที่จะส่งต่อไปยังขั้นตอนถัดไป

กิจกรรมที่ 3 : การสร้างอีเวนท์โมเดล (Event Model Construction)

การแสดงโมเดลของเหตุการณ์ที่เกิดขึ้นระหว่างวัตถุที่มีอยู่ในระบบ ซึ่งจะแสดงในรูปของข้อความ (message-passing) หรือคำอธิบายของเหตุการณ์ที่เกิดขึ้น การแสดงโมเดลจะแสดงในระบบได้จะต้องมีหลักเกณฑ์ ดังต่อไปนี้

- เหตุการณ์ทุกเหตุการณ์จะต้องสัมพันธ์กับเซอร์วิสใน O/C และต้องอยู่ในแผนงานที่กำหนดไว้
- อีเวนท์โมเดลต้องมาจากเซอร์วิสที่มีคุณสมบัติทางวิศวกรรมซอฟต์แวร์ เช่น Robustness, Stable และ Extensible เป็นต้น
- คลาสหรือวัตถุที่นำมาใช้ต้องถูกสร้างขึ้นมาจากกิจกรรมที่ 11 และต้องตรงตามแผนงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เซอร์วิสที่นำมาใช้งานจะต้องถูกกำหนดมาจากแผนงานที่กำหนดไว้ (กิจกรรมที่ 15)
- เหตุการณ์ต้องแสดงอยู่ในรูปของข้อความและตามด้วยหมายเลขของเหตุการณ์และต้องตรงตามความต้องการหรือแผนงานของระบบ

สัญลักษณ์ที่ใช้แสดงแบ่งเป็น 2 ลักษณะ คือ ใช้ลูกศรเพื่อแสดงทิศทางของเหตุการณ์ วงรีแสดงเซอร์วิสที่เกี่ยวข้อง การแสดงโมเดลของเหตุการณ์บางครั้งอาจมีเหตุการณ์เกิดขึ้นมากกว่าหนึ่งเหตุการณ์ในหนึ่งแผนงานในระบบได้ ดังนั้นผู้พัฒนาจะต้องบรรยายหรือแสดงเหตุการณ์ทั้งหมดที่มีหรือถ้ามีเซอร์วิสเกิดขึ้นใหม่ในระบบและมีเหตุการณ์เกี่ยวข้องควรนำมากล่าวถึง

สรุปอีเวนท์โมเดล คือ การอธิบายถึงการติดต่อสื่อสารระหว่างวัตถุอย่างละเอียดเมื่อมีเหตุการณ์เข้ามากระทบ ซึ่งจะทำให้ระบบสามารถดำเนินการได้ตรงตามความต้องการของผู้ใช้ได้สมบูรณ์มากขึ้น

กิจกรรมที่ 4 : การนำกลับมาใช้ใหม่ (Generalization for Reuse)

การนำคลาสที่มีอยู่ในคลังมาปรับปรุงเปลี่ยนแปลงให้เหมาะสมกับโครงงานที่จะนำไปใช้งาน ซึ่งคลาสที่นำมาใช้ต้องพิจารณาหลักการอยู่ 2 ประการ คือ คลาสที่นำมานั้นจะต้องมีคุณสมบัติของการนำกลับมาใช้ใหม่ (Potentially Reuse Components – PRC) และต้องอยู่ในคลังของคลาส (Library of Generalized Components – LGC) ประการที่สอง คลาสที่นำมาใช้ต้องอยู่บนพื้นฐานของ Cost-Center Model [19] เมื่อได้คลาสที่ต้องนำมาใช้กับโครงงานแล้ว สิ่งที่ควรคำนึงต่อไป คือ คุณภาพ ซึ่งจุดนี้มีความสำคัญอย่างยิ่ง คลาสที่เลือกขึ้นมานั้นต้องมีคุณสมบัติครบทั้ง 4 ประการกล่าวคือ สามารถที่จะเปลี่ยนการอิมพลีเมนต์ได้ สามารถเพิ่มคลาสใหม่ ๆ ได้ สามารถปรับปรุงแผนภาพของการสืบทอดคุณสมบัติได้และสามารถเปลี่ยนโครงสร้างความสัมพันธ์ของคลาสในระดับ Aggregation ได้[22] ซึ่งพิจารณาทางด้านการจัดการผู้พัฒนาระบบต้องการที่ให้คลาสถูกเลือกนั้นครอบคลุมคุณสมบัติทั้งคุณภาพและอยู่ในคลังของคลาสไปพร้อม ๆ กัน

ระหว่างการทำงานจริงของการหาแผนภาพของการสืบทอดใหม่บางครั้งอาจจัดการกับสิ่งเหล่านี้ได้ไม่ยุ่งยากนักหรือบางครั้งอาจต้องหาข้อมูลเพิ่มเติมสำหรับการสืบทอดของคลาสเหล่านั้นซึ่งทำให้คลาสนั้นตรงกับจุดประสงค์หรือเข้ากับโครงงานที่จะนำไปรวมได้เป็นอย่างดี โดยการกระทำนี้บางครั้งมีผลทำให้เกิดการทำซ้ำหรือกระบวนการทำซ้ำขึ้นอีกครั้งหนึ่งใน O/C อาจเป็นได้ และภายในกิจกรรมนี้ยังมีส่วนย่อย ๆ สามารถแบ่งได้ดังนี้

<u>กิจกรรมที่ 4.1 : การสร้างความสมบูรณ์ให้กับเซอร์วิส (Completion of abstractions)</u>

การหาเซอร์วิสที่มีอยู่ในคลาสให้เหมาะสมกับโครงการ คือ การเพิ่มหรือลดเซอร์วิสให้กับคลาสที่สนใจ เช่น คลาสของปากกา เดิมอาจมีความสามารถเพียงแค่การเขียนและบอกสีของหมึกเท่านั้นพอนำเข้ามาใช้ใหม่ควรเพิ่มความสามารถใหม่เข้าไป คือ สามารถเติมน้ำหมึกได้หรือระบุได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

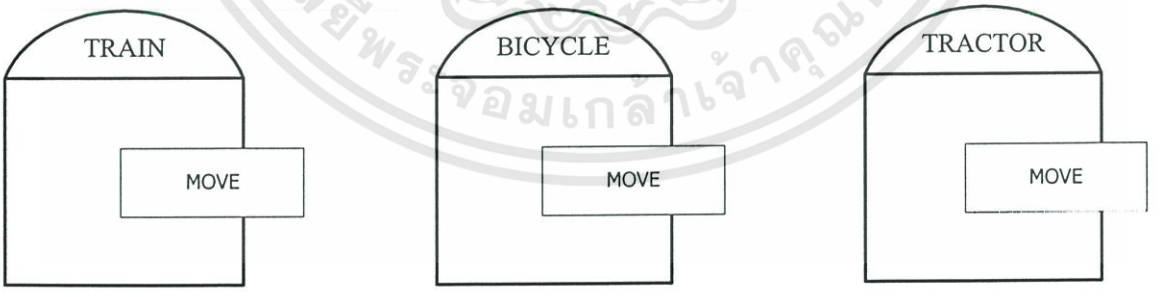
ว่าขณะนี้หมึกยังมีอยู่หรือหมดแล้ว หรือ คลาสของเครื่องคิดเลข เดิมมีเพียงแต่การบวก กับ ลบ เท่านั้น เมื่อนำมาใช้ใหม่ก็สามารถเพิ่มเซอร์วิสลงไปได้อีก (เช่น การคูณ การหาร การคิดเปอร์เซ็นต์ เป็นต้น) เป็นต้น

กิจกรรมที่ 4.2 : Optimization

กิจกรรมย่อยนี้จะเน้นไปทางด้านการอิมพลีเมนต์มากกว่าโครงสร้างโดยจะเข้าไปจัดการกับกระบวนการทำงานของเซอร์วิสให้กระชับและรวดเร็วมากยิ่งขึ้น เช่น การจัดการกับตัวแปรของการเก็บข้อมูล การเข้าถึงข้อมูลและ/หรือลดความซ้ำซ้อนของความสัมพันธ์ของคลาส เป็นต้น ซึ่งการทำของกิจกรรมนี้จะไม่มีผลกับการติดต่อทั้งภายในและภายนอกคลาส ถึงแม้ว่าการจัดการต่าง ๆ จะต้องนำข้อมูลมาจากขั้นตอน Class Specification ก็ตาม

กิจกรรมที่ 4.3 : การค้นหาลำดับชั้นของการสืบทอดคุณสมบัติ (Refinement of inheritance hierarchies)

การค้นหาหรือจัดการกับแผนภาพของการสืบทอดคุณสมบัติของ O/C ที่มีอยู่ให้เหมาะสมหรือเข้ากับคลาสที่เลือกมาจากคลังของคลาส ซึ่งบางครั้งอาจต้องสร้างคลาสขึ้นมาใหม่ เช่น อาจมีซุปเปอร์คลาส หรือ คลาสย่อย ๆ เพิ่มขึ้น เป็นต้น จากรูปที่ 4.31 แสดงคลาสของจักรยาน รถไฟและรถแทรกเตอร์ ซึ่งทั้ง 3 เป็นคลาสที่ได้มาจากการออกแบบและทั้งหมดมีคุณสมบัติที่เหมือนกัน คือ เคลื่อนที่ได้ จากการพิจารณาพบว่าคลาสทั้งหมดนี้สามารถนำมารวมอยู่ในประเภทเดียวกันได้ โดยเพิ่มซุปเปอร์คลาสที่ชื่อว่า ยานพาหนะ จากนั้นย้ายคุณสมบัติที่คล้าย ๆ กัน ไปอยู่ที่คลาสยานพาหนะ (ดังแสดงในรูปที่ 4.32) เหตุที่ต้องเป็นคลาสยานพาหนะเพราะว่าทำให้คลาส มีความคงทน มีความถาวรรวมไปถึงเข้าใจง่ายและมีคุณสมบัติที่เหมือนกัน

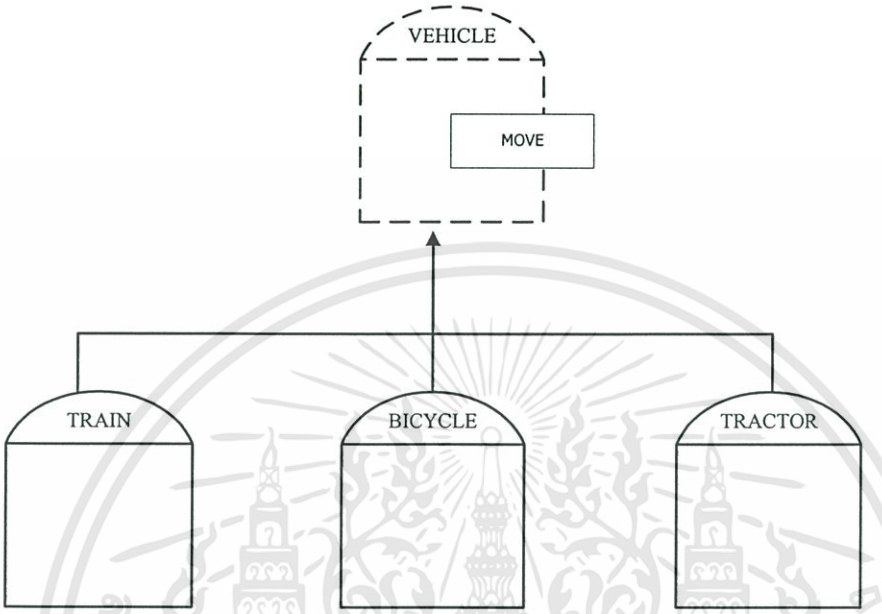


รูปที่ 4.31 แสดงคลาสของรถไฟ รถจักรยาน และรถแทรกเตอร์

จากตัวอย่างที่แสดงเป็นเพียงส่วนหนึ่งของการนำกลับมาใช้ใหม่ซึ่งรวมไปถึงการปรับปรุงและเปลี่ยนแปลง ซึ่งการที่จะนำเช่นนี้ได้นั้น คือ ต้องกำหนดคลาสที่ต้องการให้ตรงกับโครงงานที่จะนำไปใช้ กำหนดคุณสมบัติของคลาสให้ตรงกับโครงงาน คุณสมบัติของคลาสที่ปรากฏอยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานนั้นเป็นเพียงการใช้ภายใน ถ้าเป็นไปได้ควรจะทำให้อยู่ในรูปทั่วไปพยายามกำหนดคุณสมบัติให้ตรงกับชนิดของคลาส (Abstract Class) พร้อมกับเลือกคลาสย่อยให้เหมาะสมกับซุปเปอร์คลาสและระหว่างวงจรชีวิตของโครงงานควรพยายามนำคลาสต่าง ๆ มาปรับปรุงหรือเปลี่ยนแปลงอยู่เสมอ [38]



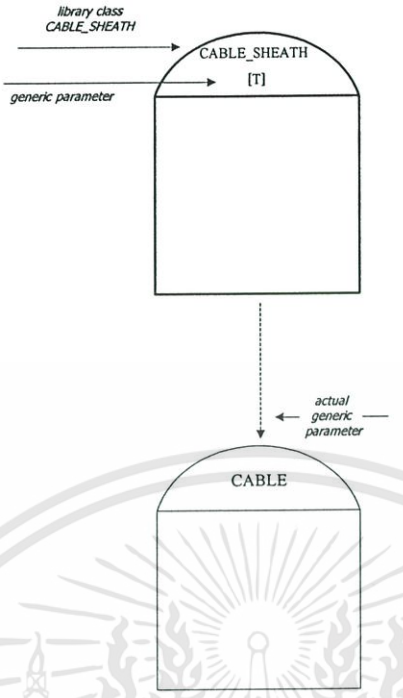
รูปที่ 4.32 แสดงคลาสทั้งสามที่ย้ายเซอร์วิสที่เหมือนกันไปสู่คลาสยานพาหนะ

จากที่กล่าวมาทั้งหมดนั้นพบว่าการนำคลาสที่มีอยู่ในคลังกลับมาใช้ใหม่ควรพิจารณาให้รอบครอบถึงความต้องการที่แท้จริงของโครงงาน ซึ่งถ้าทำได้ตามวัตถุประสงค์ย่อมทำให้คลาสที่เลือกขึ้นมานั้นมีความเหมาะสมทั้งคุณภาพและประสิทธิภาพ นอกจากนี้ยังทำให้การออกแบบคลาสไม่ยุ่งยากอีกด้วย

กิจกรรมที่ 5 : การระบุประเภทของคลาส (Genericity Specification)

กิจกรรมนี้จัดอยู่ในขั้นตอนอิมพลีเมนต์ซึ่งเป็นการระบุหรือชี้ชัดลงไปว่าคลาสที่นำมาอิมพลีเมนต์ควรใช้วิธีใด อัลกอริทึมใด เช่น คลาสของการนำเสนอโครงสร้างข้อมูล ซึ่งมีทั้ง Stack, Queue และ Linked_List ฯลฯ วิธีทำ คือ นำคลาสเหล่านั้นมาใส่เครื่องหมาย (Generic Parameter) และอธิบายรายละเอียดลงใน O/C กับClass Specification เช่น คลาสของสายเคบิลมีอยู่หลายชนิด ซึ่งการนำมาใช้งานควรระบุลงไปให้ชัดเจนว่าชนิดของสายเคบิลที่ใช้เป็นชนิดใด ดังนั้นในรูปที่ 4.33 ที่แสดงถึงคลาส CABLE_SHEATH หลังจากระบุประเภทของคลาสทำให้นักพัฒนาสามารถเห็นความชัดเจนของคลาสมากขึ้นในขั้นตอนการเขียนโปรแกรมเชิงวัตถุ แต่เครื่องมือที่ใช้พัฒนานั้นควรสนับสนุนคุณสมบัติดังกล่าวด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 แสดงคลาส CABLE_SHEATH ที่มีลักษณะเป็น Genericity

กิจกรรมที่ 6 : การกำหนดการสืบทอดคุณสมบัติ (Inheritance Identification)

จุดประสงค์ของกิจกรรมนี้มุ่งเน้นทางด้านการสืบทอดคุณสมบัติของวัตถุและการจัดลำดับชั้นของคลาส (Hierarchies of Classes) ซึ่งในโมเสสมีสัญลักษณ์ของกิจกรรมนี้อยู่ 2 ลักษณะคือ แบบทั่วไปที่ใช้กับขั้นตอนการวิเคราะห์และออกแบบ กับแบบของการนำเสนอในขั้นตอนการอิมพลีเมนต์ ซึ่งมีรายละเอียดดังนี้

<u>การสืบทอดคุณสมบัติแบบทั่วไป (Generalization Hierarchies)</u>

การนำเสนอคลาสใน O/C Model ที่ได้มักอยู่ในรูปทั่วไปซึ่งยังไม่สามารถระบุถึงความสัมพันธ์แบบ “is-a” ได้ ซึ่งคลาสที่มีลักษณะนี้จะมีอิสระต่อกันมากกว่า ทำให้กิจกรรมนี้กล่าวถึงขั้นตอนการพิจารณาคลาสเหล่านั้น คือ นำคลาสทั้งหมดมาพิจราณาโดยละเอียดถึงความเหมือนและความเข้ากันได้ให้มากขึ้น การจัดลำดับคลาสหรือการสืบทอดคุณสมบัติสามารถกระทำได้ 2 ลักษณะ คือ นำคลาสแบบธรรมดามากระทำให้อยู่ในรูปแบบทั่วไป เพื่อให้สามารถนำไปใช้กับทุกสถานการณ์ได้ ถัดมานำคลาสลักษณะเฉพาะที่มีความเหมือนกับคลาสแบบทั่วไปไปเป็นคลาสย่อยของคลาสทั่วไปซึ่งความสัมพันธ์ลักษณะนี้อาจมีมากกว่าหนึ่งได้ ซึ่งลักษณะทั้งสองมีความต่อเนื่องจากขั้นตอน Specification

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับของการอิมพลีเมนต์ในการสืบทอดคุณสมบัติ (Implementation Inheritance Hierarchies)

วัตถุประสงค์ของกิจกรรมย่อยนี้ คือ พิจารณาโปรแกรมที่มีอยู่เดิมในไลบรารีให้สามารถนำกลับมาใช้ใหม่ได้อีก เริ่มจากตรวจสอบไลบรารีและองค์ประกอบต่าง ๆ ของคลาสที่สามารถนำมาใช้งานได้ดีและทำการเพิ่มรายละเอียดลงไปในคลาสนั้น การนำกลับมาใช้ใหม่นั้น เปรียบเสมือนกับการกระตุ้นและ/หรือการจูงใจให้พยายามค้นหาไลบรารีและความซับซ้อนของลำดับขั้นของคลาส

การอิมพลีเมนต์ของการนำกลับมาใช้ใหม่ของโปรแกรมที่มาจากการสืบทอดคุณสมบัติของคลาสหลักสามารถนำมาใช้กับลำดับชั้นของคลาสได้เพียงลำดับเดียว กล่าวคือ เซอร์วิสในแต่ละคลาสจะอิมพลีเมนต์ตัวของมันเองที่อยู่ ณ ระดับเดียวกันหรือที่เรียกว่าความสัมพันธ์แบบ “is-a” เท่านั้น ซึ่งแต่ละวัตถุจะถูกแทนด้วยโค้ดของโปรแกรม ซึ่งโค้ดเหล่านี้สามารถนำกลับมาปรับปรุงและแก้ไขได้ โดยไม่ต้องจัดลำดับชั้นของการสืบทอดใหม่ทั้งหมด ทำนองเดียวกันโค้ดที่นำกลับมาใช้ใหม่ สามารถนำเทคนิคการผสม (MIX) เข้ามาช่วยสำหรับการออกแบบและพัฒนา เพื่อช่วยลดเวลาและผลกระทบที่เกิดขึ้นให้น้อยที่สุด ความสัมพันธ์ของโค้ดที่นำกลับมาใช้ใหม่จะอยู่ในรูปของ “is-implement-in-term-of” ซึ่งเป็นความสัมพันธ์ในระดับคลาส กล่าวคือ การอิมพลีเมนต์ที่เกิดขึ้นจะสัมพันธ์กับคลาสที่ตนสัมพันธ์เท่านั้น ซึ่งตามกฎของการอิมพลีเมนต์นั้นจะคำนึงถึงสิ่งที่อยู่ภายในคลาสและไลบรารีที่จะนำกลับมาใช้ใหม่เป็นหลักเพื่อให้สามารถนำกลับมาใช้ใหม่ได้เต็มที่

การสืบทอดคุณสมบัติแบบหลายลำดับ (Multiple Inheritance)

การสืบทอดคุณสมบัติแบบหลายชั้น ส่วนใหญ่มักจะขึ้นกับภาษาเชิงวัตถุ ซึ่งบางครั้งการทำลักษณะเช่นนี้ทำให้ ชื่อและคุณสมบัติต่าง ๆ มีความขัดแย้งกันและยังทำให้เซอร์วิสที่เกิดขึ้นนั้นไม่สามารถทำงานได้ทั้งกับคลาสและวัตถุที่สืบทอดคุณสมบัติลงมา Brachman กล่าวว่า “ลักษณะของความสัมพันธ์ที่เกิดขึ้นควรมีลักษณะที่หลากหลาย เช่น การสืบทอดคุณสมบัติของหนูที่มาจากช้างนั้นควรที่จะเปลี่ยนแปลได้เพราะหนูนั้นไม่มีงาเหมือนช้าง ดังนั้น แอตทริบิวต์ภายในวัตถุที่สืบทอดคุณสมบัติมากจากคลาสหลักควรมีความชัดเจนและไม่คลุมเคลือพร้อมกับสามารถเปลี่ยนแปลงได้เป็นต้น”[7]

กิจกรรมที่ 7 : กำหนดความสัมพันธ์ (Interaction Specification)

ความสัมพันธ์ที่เกิดขึ้นระหว่างวัตถุกับคลาส วัตถุกับวัตถุ หรือ คลาสกับคลาส มีอยู่หลายลักษณะ เช่น ความสัมพันธ์ในลักษณะของการสืบทอดคุณสมบัติ (Inheritance Relationship) ความสัมพันธ์ลักษณะเป็นส่วนหนึ่งของวัตถุหรือคลาสที่อ้างถึง (is-part-of หรือ Aggregation Relationship) และความสัมพันธ์ลักษณะพึ่งพาซึ่งกันและกัน (Association Relationship) เป็นต้น แต่ในกิจกรรมนี้กล่าวถึงความสัมพันธ์ 2 ลักษณะหลัง ซึ่งมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดความสัมพันธ์ลักษณะการประกอบกันของคลาส (Aggregation Specification)

การกำหนดความสัมพันธ์ลักษณะการประกอบกันของคลาส คือ ความสัมพันธ์ที่คลาสหรือวัตถุมีลักษณะเป็นส่วนหนึ่งของคลาสหรือวัตถุที่อ้างถึง โดยมีลักษณะของความสัมพันธ์แบบเฉพาะ (Private Part) มากกว่าสัมพันธ์แบบสาธารณะ (Public Part) ซึ่งอาจจะมีหรือไม่มีก็ได้ขึ้นกับความเหมาะสม เพราะบางกรณีอาจจำเป็นต้องมีเพื่อให้การทำงานโดยรวมมีผลลัพธ์ที่ดีขึ้น แต่ถ้าวัตถุหรือคลาสที่เกี่ยวข้องมีลักษณะของความสัมพันธ์เป็นแบบชั่วคราว กล่าวคือ วัตถุหรือคลาสที่สัมพันธ์ด้วยจะต้องถูกลบทิ้งหรือทำลายเมื่อเสร็จสิ้น ก็ควรใช้ความสัมพันธ์ในลักษณะเช่นนี้ได้และถ้าเป็นความสัมพันธ์แบบถาวรซึ่งไม่มีการทำลายวัตถุหรือคลาสทิ้งควรใช้ความสัมพันธ์แบบแอสโซซิเอชันมากกว่า

การกำหนดความสัมพันธ์แบบแอสโซซิเอชัน (Association Specification)

ความสัมพันธ์แบบ Association เป็นอีกรูปแบบหนึ่งที่ใช้กำหนดความสัมพันธ์ที่เกิดขึ้นระหว่างวัตถุหรือคลาส การกำหนดขอบเขตของปัญหาที่เกี่ยวกับความสัมพันธ์ของคลาสควรยึดหลักความต้องการของผู้ใช้ (URS-User Requirement Specification) และแผนงานเป็นสำคัญ ส่วนความสัมพันธ์ที่เกิดขึ้นกับคลาสต้องคำนึงถึงโมเดลเป็นอันดับแรก เพราะจะทำให้การกำหนดกลุ่มของความสัมพันธ์นั้นทำได้ง่ายขึ้น ซึ่งมีหลักเกณฑ์ต่าง ๆ ดังต่อไปนี้

- ถ้ามีความสัมพันธ์แบบสาธารณะ คือ ต้องสัมพันธ์กันทั้งโครงสร้างของวัตถุ ไม่ว่าจะเป็นแอตทริบิวต์และเซอร์วิส ซึ่งไม่ใช่เป็นแบบชั่วคราวที่เกิดตามการกระทำหรือเหตุการณ์ที่เข้ามากระทบกับวัตถุ
- ตรวจสอบความสัมพันธ์ที่เป็นแบบกลุ่ม (Ternary Association) คือ ต้องสามารถอธิบายรายละเอียดของเขตของปัญหาได้ชัดเจนที่สุด และถ้าต้องการกระจายไปสู่ความสัมพันธ์ระดับไบนารี ข้อมูลที่มีอยู่ต้องไม่สูญหาย
- ขจัดความสัมพันธ์ที่ซ้ำซ้อนออกไป ซึ่งควรจะทำในขั้นตอนของ Specification เพราะจะทำให้เข้าใจปัญหาที่ทำให้เกิดความซับซ้อนได้ง่ายขึ้น
- นำกฎของการใส่ชื่อความสัมพันธ์มาใช้กับ O/C Model เพื่อแสดงถึงความสัมพันธ์ให้ชัดเจนมากยิ่งขึ้น
- คลาสหรือวัตถุที่สัมพันธ์ด้วย ควรตรวจสอบเรื่องข้อจำกัดต่าง ๆ ก่อนออกจากกิจกรรม

ปัจจุบันการกำหนดความสัมพันธ์มีอยู่หลายวิธี เช่น วิธีของ Rumbaugh, Killian, Jacobson หรือ Coad/Yourdon เป็นต้น แต่จะขึ้นอยู่กับการเลือกนำไปประยุกต์ โดยมีหลักเกณฑ์ที่คล้ายกัน คือ ต้องการเชื่อมโยงความสัมพันธ์ระหว่างวัตถุตั้งแต่ 2 วัตถุขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป คือ การกำหนดความสัมพันธ์ทั้งแบบ Association และ Aggregation ที่กล่าวมาข้างต้น ส่วนใหญ่มีลักษณะคล้ายกับการกำหนดความสัมพันธ์ของดาต้าโมเดล ซึ่งเทคนิคของแต่ละแบบจะมีความหมายไม่เหมือนกันแต่หลักการคือต้องการจัดกลุ่มของดาต้าที่มีความเกี่ยวข้องกันตั้งแต่ 2 หรือมากกว่า 2 วัตถุขึ้นไปสามารถติดต่อสื่อสารแลกเปลี่ยนข้อมูลกัน

กิจกรรมที่ 8 : การวางแผนพัฒนาแบบทำซ้ำ (Iteration Plan Development)

การกำหนดแผนงานแบบทำซ้ำของโครงงานที่ต้องพัฒนามีความสำคัญลำดับต้น ๆ ของการพัฒนา เพราะทำให้ผู้พัฒนาได้ทราบถึง ระยะเวลา วัตถุประสงค์และทรัพยากรที่จะต้องใช้ในการพัฒนาได้อย่างถูกต้องและเหมาะสม ซึ่งทั้งหมดนี้อยู่ในขั้นตอนการวางแผน นอกจากนี้ยังทำให้ทราบถึงแผนงานถัดไปหรือแผนงานที่เกี่ยวข้องอีกด้วย ซึ่งภายในกิจกรรมประกอบด้วย

- กำหนดและจัดลำดับความต้องการของระบบ (Identify and Prioritize System Requirement) ซึ่งจะเกิดขึ้นกับระดับบนของโครงงาน โดยนำรายละเอียดต่าง ๆ ที่มาจากลูกค้าหรือผู้ใช้ระบบโดยตรง เช่น การสัมภาษณ์ การเข้าไปติดตามงานด้วยตนเอง หรือนโยบายขององค์กร เป็นต้น เพื่อนำมาเป็นส่วนประกอบในการากำหนดแผนงานทั้งหมดของโครงงาน
- กิจกรรมนี้จะจัดการเอกสารภายนอกที่เกี่ยวข้องทั้งหมดกับโครงงานที่ทำซ้ำทั้งหมด ไม่ว่าจะเป็นการใช้ทรัพยากร ความเกี่ยวข้องกับระบบอื่น ๆ เพื่อที่จะนำสิ่งเหล่านี้มาประกอบการตัดสินใจเพื่อลดความเสี่ยงที่จะเกิดขึ้นกับโครงงานที่กำลังทำอยู่
- กำหนดตารางการทำงานของโครงงาน เริ่มตั้งแต่ขั้นตอนแรกจนถึงขั้นตอนสุดท้ายโดยละเอียด ซึ่งกล่าวถึง วิธีการพัฒนา เทคนิค ระยะเวลาในการพัฒนา และการประเมินความผิดที่จะเกิดขึ้นในทุก ๆ ขั้นตอน

กิจกรรมย่อยที่เกิดขึ้นภายในกิจกรรมนี้เป็นส่วนหนึ่งของโครงงานเท่านั้น ซึ่งนับว่ามีความจำเป็นอย่างมากที่ทำให้โครงงานทั้งหมดบรรลุผลไปได้ด้วยดี ซึ่งถ้าสังเกตให้ดีแล้ว พบว่ากิจกรรมย่อยเหล่านี้มีความเหมือนกับ PERT/CPM หรือ Gantt Chart[12],[26] ที่นำมากำหนดการพัฒนาโครงงานที่พบโดยทั่วไป

กิจกรรมที่ 9 : การรวมคลาสไปสู่ไลบรารี (Library Class Incorporation)

วัตถุประสงค์หลักของกิจกรรมนี้คือการนำคลาสที่มีอยู่กลับมาใช้ใหม่เพื่อให้เกิดประโยชน์มากที่สุด ซึ่งการนำกลับมาใช้ใหม่นี้นับว่ามีส่วนสำคัญต่อการพัฒนาระบบสารสนเทศเชิงวัตถุและการนำมาใช้นี้ต้องขึ้นอยู่บนพื้นฐานที่เหมาะสม โดยเฉพาะโมเดลที่นำมาใช้ต้องสนับสนุนซึ่งกันและกัน เช่น แบบจำลองน้ำพุ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การค้นหาคลาสจากคลังเพื่อนำมาใช้นั้นมีความสำคัญต่อกระบวนการพัฒนาโครงงาน ดังนั้นจึงต้องเน้นถึงความสอดคล้องกับคลาสที่มีอยู่ ถ้าเป็นไปได้ควรมีคุณสมบัติที่ใกล้เคียงกับคลาสที่เคยสืบทอดกันมา เพราะจะต้องนำคลาสเหล่านั้นไปใช้กับขั้นตอน Specification กับการอิมพลีเมนต์ ซึ่งทั้ง 2 ขั้นตอนนี้มีวิธีการนำเสนอที่แตกต่างกัน คือ ขั้นตอนแรกนำเสนอแบบบนลงล่าง ส่วนการอิมพลีเมนต์นำเสนอแบบล่างขึ้นบน ซึ่งผู้พัฒนาจะต้องนำมาประยุกต์ให้เข้ากัน บางครั้งอาจต้องใช้วิธี "Fuzzy" เข้ามาช่วยเพื่อให้การพัฒนาทำได้ง่ายและสะดวกยิ่งขึ้น

คลาสที่อยู่ในคลังมีอยู่ 5 ลักษณะ คือ

1) คลาสที่ถูกพัฒนาขึ้นมาจากทีมเดียวกันและในโครงงานเดียวกัน คลาสลักษณะนี้สามารถนำมาใช้งานได้ทันที เพราะมักมีแนวทางแก้ไขปัยหาไปในทิศทางเดียวกัน
2) คลาสที่พัฒนาขึ้นเองแต่เป็นของโครงงานอื่น ซึ่งสามารถนำมาใช้กับโครงงานที่กำลังพัฒนาอยู่แต่อาจจะต้องนำมาปรับปรุงให้ถูกต้องและตรงกับปัญหาที่ต้องการใช้เสียก่อน (กิจกรรมที่ 4)
3) คลาสจากภายนอกซึ่งเป็นคลาสที่เกิดจากโครงงานหรือระบบงานที่องค์กรจ้างให้บริษัทภายนอกพัฒนาขึ้น เช่น ระบบบัญชี ระบบการเงิน ระบบประกันภัยและระบบอื่น ๆ เป็นต้น
4) คลาสที่ได้จากผู้ค้าร่วมของผู้ที่ถูกว่าจ้างสร้างระบบให้กับองค์กร ซึ่งจะถูกสร้างโดยภาษาที่แตกต่างกันไป และอยู่ในระดับล่างของแอพพลิเคชั่นที่เรียกใช้งาน
5) คลาสที่ได้จากเครื่องมือที่เลือกมาพัฒนาโครงงาน เช่น ภาษา C++, Eiffel[37], Visual Basic, Visual C++ และ Delphi เป็นต้น ซึ่งอยู่ในรูปทั่วไป เช่น ฟังก์ชันการบวก การลบ การจัดหน้าจอ การแสดงเวลาและอื่น ๆ เป็นต้น

ในกรณีที่เลือกคลาสจากภายนอกคือหัวข้อที่ 3-5 มาประกอบในการพัฒนาระบบ ควรคำนึงถึงคุณภาพของคลาสต่าง ๆ ดังนี้

- คลาสต้องมีความสมบูรณ์ เช่น สามารถเข้ากับระบบได้ ครอบคลุมปัญหาที่นำไปใช้เป็นต้น
- มีการออกแบบที่กระชับและเข้าใจง่าย
- มีมาตรฐาน
- สามารถใช้กฎการสืบทอดคุณสมบัติได้
- สามารถใช้ได้กับทุกรูปแบบ ไม่ว่าจะเป็นแบบเครือข่ายหรือแบบเดี่ยว
- สามารถเข้าถึงระดับล่างของคลาสที่นำไปใช้ได้เป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เรียกใช้ได้ง่ายและมีข้อความแสดงถึงการทำงาน เช่น เมื่อเกิดผิดพลาดต้องมีข้อความแสดง เป็นต้น
- มีฟังก์ชันคอยตรวจสอบเงื่อนไขก่อนหลัง (ถ้าจำเป็น)
- มีความยืดหยุ่น
- มีเอกสารประกอบโดยละเอียด เช่นโครงสร้างของคลาส บอกระดับความยากง่าย อธิบายจุดประสงค์โดยรวมของคลาสที่นำเสนอ เป็นต้น
- มีบริการหลังการขายจากผู้ขาย

จากที่กล่าวมาข้างต้นเป็นสิ่งจำเป็นสำหรับคลาสที่จะนำมาใช้ประกอบการพัฒนาในโครงงาน ลักษณะของการนำมาใช้งานโดยทั่วไปมักจะมีอยู่หลายลักษณะ เช่น นำมาใช้โดยตรง นำมาเป็นส่วนประกอบ หรือนำมาใช้เพียงบางส่วน เป็นต้น สิ่งที่สำคัญที่สุด คือ การนำมาใช้งานซึ่งไม่ใช่นำมาใช้เพียงชั่วคราวเท่านั้นแต่เป็นการนำมาใช้ทั้งวงจรการพัฒนาของโครงงาน ที่ไม่ได้เน้นหนักไปที่ขั้นตอนใดขั้นตอนหนึ่งระหว่าง Specification กับ อิมพลีเมนต์ เท่านั้น

สิ่งที่เกิดขึ้นอีกอย่างหนึ่งกับโครงงาน คือ คลาสที่นำเข้ามาเสริมหรือนำมาใช้งานนี้ทำให้เกิดความสัมพันธ์ใหม่ ๆ กับคลาสที่มีอยู่เดิม ทั้งแบบ Association, Aggregation หรือระดับไคลเอ็นต์/เซอร์เวอร์ที่มีความสัมพันธ์ทั้ง 2 แบบ การใช้คลาสมาประกอบในโครงงานอาจจะทราบผลก็ต่อเมื่อผ่านเข้าสู่ขั้นตอนรีวิว (Review) ซึ่งขั้นตอนนี้อาจทำให้เกิดการเปลี่ยนแปลงและทราบว่าคลาสที่นำมาใช้มีความเหมาะสมหรือไม่บางครั้งอาจต้องกลับขั้นตอนการออกแบบใหม่อีกครั้งเพื่อค้นหาหรือปรับแต่งคลาสให้สามารถเข้ากับระบบที่นำไปใช้งาน

จากที่กล่าวมาสามารถสรุปได้ว่า การนำคลาสจากคลังมาใช้งานนั้นต้องขึ้นอยู่กับประสบการณ์ของผู้ออกแบบ เช่น โครงสร้างของคลาส ปัญหาที่เกิดขึ้นและรายละเอียดของคลาสเหล่านั้น เป็นต้น ดังนั้นการนำคลาสมาใช้งานเปรียบเสมือนการนำอุปกรณ์ชิ้นหนึ่งมาประยุกต์ใช้ให้เข้ากับงานที่กำลังทำได้เป็นอย่างดีและเหมาะสม

กิจกรรมที่ 10 : การสร้างออบเจ็กต์ชาร์ต (ObjectChart Construction)

การสร้างออบเจ็กต์ชาร์ต คือ กิจกรรมที่นำเสนอพฤติกรรมของวัตถุในรูปของแผนภาพที่ Coleman[10] นำมาประยุกต์ใช้กับวงจรของการพัฒนาระบบสารสนเทศซึ่งเรียกว่า "StateCharts" แต่ในโมเสสเรียกว่า "ออบเจ็กต์ชาร์ต" (ObjectCharts) ภายในประกอบด้วย State Transition Diagram (STD) ของวัตถุ การสร้างออบเจ็กต์ชาร์ตไม่มีความแตกต่างกันกับวิธีของ Rumbaugh, Coad/Yourdon หรือ Jacobson กล่าวคือออบเจ็กต์ชาร์ตจะบันทึกพฤติกรรมของวัตถุที่สนใจอย่างละเอียดและมีหลักเกณฑ์ของการสร้างเสตทดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถอธิบายทุกแอตทริบิวต์ที่สเตทเข้าไปเกี่ยวข้องได้
- กำหนดค่าให้กับตัวแปรเมื่อเสร็จสิ้นสเตทได้
- กำหนดค่าเริ่มต้นให้กับการเริ่มต้นสเตททุกครั้งได้

ออบเจ็กต์ชาร์ตสามารถแสดงได้ดังรูปที่ 4.22 โดยแต่ละออบเจ็กต์ชาร์ตนอกจากสามารถอธิบายการทำงานของวัตถุที่สนใจด้วยไดอะแกรมแล้ว ยังต้องแสดงเอกสารประกอบด้วย (ดังแสดงในรูปที่ 4.23)

กิจกรรมที่ 11 : การกำหนดคลาสและ/หรือวัตถุ (O/C Identification)

การกำหนดและระบุวัตถุใน O/C ไดอะแกรม คือ การค้นหาวัตถุให้กับระบบ วัตถุที่ปรากฏอยู่ในแบบจำลองวัตถุเป็นเพียงโมเดลคร่าว ๆ ที่ยังไม่สมบูรณ์มากนัก ซึ่งต้องกลั่นกรองและทบทวนวัตถุที่มีอยู่ การทบทวนวัตถุแต่ละครั้งมักจะมีวัตถุที่ถูกยุบหรือเกิดขึ้นใหม่ใน O/C ไดอะแกรม ดังนั้นวัตถุที่เกิดขึ้นหรือเปลี่ยนแปลงควรบันทึกรายละเอียดต่าง ๆ ลงในเอกสารของคลาสซึ่งเรียกว่า "Class Specification" เพื่อใช้เป็นเอกสารประกอบสำหรับการพัฒนาระบบต่อไป

<u>กิจกรรมที่ 11.1 การกำหนดและค้นหาวัตถุ/คลาสในระบบ</u>

การกำหนดวัตถุที่มีอยู่ในระบบนั้น มีทั้งยากและง่าย Meyer กล่าวว่า "วัตถุเป็นเพียงการเลือก"[37] อย่างไรก็ตามสิ่งที่กล่าวนี้เป็นจริงได้ หากผู้วิเคราะห์และออกแบบเข้าใจขอบเขตของระบบดีพอ เช่น โครงสร้างข้อมูล เป็นต้น การกำหนดวัตถุให้ดีและเหมาะสมกับระบบทางธุรกิจได้นั้นทำได้ยาก

วิธีการกำหนดวัตถุสำหรับ O/C ไดอะแกรมมีหลายวิธี การกำหนดวัตถุส่วนมากกำหนดวัตถุเป็นคำนามที่มีลักษณะเป็นรูปธรรม เช่น คน สัตว์ สิ่งของ เป็นต้น หรือนามธรรม เช่น ข้อคิดเห็น เป็นต้น หรือนามธรรมรูปแบบอื่น ซึ่งการกำหนดแบบนี้ยังเกิดปัญหาเพราะมักจะทำให้เกิดคลาสมากเกินความจำเป็นและทำให้เกิดค่าใช้จ่ายของคลาสใหม่มากพร้อมกันสิ้นเปลืองเวลาสำหรับการวิเคราะห์ใหม่ ส่วน Wirfs-Brock และคณะ กล่าวว่า "ให้พิจารณาวัตถุ/คลาสทั้ง 2 แบบด้วยชื่อที่มีความจำเป็นเท่านั้นและให้รอบครอบสำหรับการใช้คำคุณศัพท์ สำหรับการบ่งชี้ถึงความแตกต่างของ O/C ในประโยคต่าง ๆ ให้ละเอียด เพื่อไม่ให้ผิดพลาดสำหรับการระบุสิ่งต่าง ๆ ที่จำเป็นใน O/C ไดอะแกรม" [54] ส่วน Coad/Yourdon กล่าวว่า "การหา O/C ในระบบนั้น ให้พิจารณาจากขอบเขตของระบบ" [39] ซึ่งอาจพิจารณาจากเหตุการณ์ กฎข้อบังคับ สถานที่และองค์ประกอบ ฯลฯ

คลาส คือ กลุ่มของวัตถุที่มีความหมายเหมือนกัน ถ้าแต่ละคลาสมีสมาชิกเพียงวัตถุเดียวถือว่าคลาสนั้นไม่ควรอยู่ในระบบที่พัฒนา วัตถุของคลาสต่าง ๆ นั้นจำเป็นต้องมีเอกลักษณ์ หมายถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุแต่ละตัวควรมีความชัดเจน ความสอดคล้องและมีพื้นฐานที่ดี เช่น คลาสของยานพาหนะประกอบด้วย วัตถุของรถยนต์ เครื่องบิน เรือ เป็นต้น ถ้าไม่มีเอกลักษณ์ที่ดีย่อมทำให้วัตถุและคลาสที่เกิดขึ้นในระบบเป็นคลาสที่ไม่เหมาะสม

การให้ชื่อคลาสควรกำหนดให้ชัดเจน ตั้งแต่เริ่มสร้างคลาสและชื่อที่ใช้ควรเป็นชื่อที่ไม่ซ้ำกับคลาสอื่น (Unique Name) จากนั้นบันทึกรายละเอียดของคลาสลงใน Class Specification ทันที รายละเอียดของคลาสที่เกิดขึ้นส่วนใหญ่มักเกิดจากแผนงาน (Scenario) ซึ่งแผนงาน คือ แหล่งที่แสดงถึงการได้มาของคลาส เช่น คำนามต่าง ๆ เป็นต้น และรายละเอียดอื่น ๆ ของการได้มาของคลาส เช่น ไลบรารีของคลาส ส่วนประกอบที่เกี่ยวข้องกันหรือดาต้าโมเดล เป็นต้น

<u>กิจกรรมที่ 11.2 การกำหนดความต่อเนื่องของคลาส</u>

การดำเนินการที่สำคัญของกิจกรรมย่อยที่ 11.2 คือ การทำให้คลาสที่ออกแบบสามารถนำไปใช้กับขั้นตอนอิมพลีเมนต์ได้ แม้ว่าบางครั้งอาจกำหนดระหว่างขั้นตอน Specification ได้ ส่วนใน Persistence O/C เป็นการแสดงผลการทำงานของคลาสกับโปรแกรม เช่น การจำลองผลการทำงานของโปรแกรม หรือ การนำข้อมูลที่เกิดจากการรันโปรแกรมขึ้นมาใช้งาน เช่น การกำหนดค่าของข้อมูล โดยข้อมูลต่าง ๆ เหล่านี้ อาจนำมาจากไฟล์หรือสิ่งที่เก็บไว้ใน Persistence O/C โดยธรรมชาติของข้อมูลจะถูกเก็บอยู่ในประเภทของฐานข้อมูลเชิงสัมพันธ์ ฐานข้อมูลเชิงวัตถุ หรือ แฟ้มข้อมูลระบบปฏิบัติการ เป็นต้น ส่วนใหญ่แอพพลิเคชั่นของ Persistance มักอยู่ในรูปของระบบฐานข้อมูล โดยเฉพาะ O/C จะถูกเก็บในรูปของหน่วยเก็บความจำสำรอง (Secondary Storage) เพื่อสามารถเก็บข้อมูลต่าง ๆ ได้คงทนตลอดระยะเวลาการพัฒนาระบบงานนั้น

กิจกรรมที่ 12 : Optimization

เมื่อความต้องการของผู้ใช้ระบบมีมากเกินความสามารถของระบบคอมพิวเตอร์ การออกแบบระบบอาจต้องนำมาปรับปรุงใหม่เพื่อให้ระบบที่ออกแบบออกมาดีที่สุด ซึ่งกิจกรรมนี้ เป็นกระบวนการที่เกิดขึ้นภายหลังจากสร้างแบบจำลองของระบบเรียบร้อยแล้ว วิธีการของขั้นตอนนี้จะแตกต่างกับกิจกรรมที่กล่าวมา เพราะเป็นหลักการของการนำกลับมาใช้ใหม่ โดยจะเข้าถึงระดับโครงสร้างของคลาสในส่วนของการออกแบบซึ่งการเปลี่ยนแปลงที่เกิดขึ้นอาจทำให้ต้องออกแบบใหม่เพื่อให้ดีขึ้นกว่าเดิม

การทำให้คลาสที่ออกแบบไว้ดีขึ้นนั้นจำเป็นต้องนำรายละเอียดของข้อมูลระดับกายภาพ (Physical Information Storage) และรหัสกระบวนการ (Process Code) มาปรับปรุง ซึ่งการปรับปรุงนั้นอาจทำเพื่อหลีกเลี่ยงไม่ให้มีการทำงานของกระบวนการมีลักษณะเวียนบังเกิดมากเกินไป หรือทำการปรับปรุงค่าของแอตทริบิวต์ที่กำหนดไว้มากเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของการปรับปรุงคลาสให้ดีขึ้นนั้น มีอยู่ 2 ระดับ ระดับแรก คือ การออกแบบโมเดลของ O/C ใหม่ ซึ่งระดับแรกจะถูกดำเนินการระหว่างขั้นตอนอิมพลีเมนต์และ Specification เพื่อลดเวลาและช่องว่างของพารามิเตอร์ในระบบ ซึ่งการออกแบบใหม่นั้นอาจต้องเคลื่อนย้ายคลาสและจัดความสัมพันธ์ของคลาสใหม่หรืออาจเพิ่มคลาสในส่วนของการออกแบบ Rumbaugh และคณะ กล่าวว่า "การปรับปรุงโมเดลของ O/C นั้น ควรมาจากความถี่ของคำถามที่ได้มาจากผู้ใช้ระบบ ซึ่งการค้นหาวัตถุนั้นควรใช้อินเด็กซ์เข้ามาช่วยเพื่อลดขั้นตอนการสืบค้นและความสัมพันธ์ที่ซับซ้อนของวัตถุ" [28] ซึ่งจากคำกล่าวนี้ทำให้คลาสที่ได้เกิดการปรับปรุงขึ้นใหม่มีความซับซ้อนของความสัมพันธ์ลดลงและมีผลทำให้ O/C โมเดลที่ได้นั้นมีความกระชับมากขึ้น ส่วนการนำอินเด็กซ์เข้ามาช่วยทำให้ล่นระยะเวลาในการส่งข้อความไปยังวัตถุได้ดีขึ้นและชัดเจนมากขึ้น

ระดับที่สอง คือ การแยก O/C โมเดลออกเพื่อหาทางแก้ไขปัญหาให้ดีที่สุด โดยบางครั้งอาจต้องลดวัตถุที่มีลักษณะการห่อหุ้มลง (Encapsulation) และเปลี่ยนไปใช้เทคนิคลักษณะ Friends Mechanism แทน ทั้งนี้เพราะการใช้เทคนิคการห่อหุ้มทำให้วัตถุมีความยืดหยุ่นและการนำกลับมาใช้ใหม่ได้มากกว่า ส่วนการออกแบบใหม่บางครั้งอาจเสียค่าใช้จ่ายเป็นต้นทุนของเวลาบ้างก็ตามแต่ผลที่ได้นั้นทำให้ผู้ออกแบบเกิดความมั่นใจกับสิ่งที่กำลังทำอยู่นั้นเป็นสิ่งที่ถูกต้อง โดยผู้ออกแบบอาจต้องสร้างแบบจำลองของระบบขึ้นมาเพื่อทดสอบแนวทางการแก้ปัญหาที่ได้เลือกไว้ ซึ่งการสร้างแบบจำลองนี้อาจเป็นเพียงการจำลองการทำงานในระยะสั้นเท่านั้น แต่ประโยชน์ของการสร้างแบบจำลองนี้สามารถชี้ให้เห็นถึงพื้นฐานของประสิทธิภาพในระบบได้เป็นอย่างดี

กิจกรรมที่ 13 : การประเมินคุณภาพ (Quality Evaluation (Metrics))

คำว่า "คุณภาพ" มีความหมายต่อการพัฒนาระบบเป็นอย่างมาก โดยทั่วไปมักวัดคุณภาพกันที่การเขียนโปรแกรมเป็นส่วนใหญ่ แต่สำหรับโมเสสนั้นจะครอบคลุมตั้งแต่ โครงสร้าง ความสัมพันธ์และการเขียนโปรแกรม (เซอร์วิส) ซึ่งการวัดนั้นไม่ได้สิ้นสุดแค่คำว่า "ดี" หรือ "ผ่านการทดสอบ" เท่านั้น หากกล่าวไปแล้ว การวัดคุณภาพและการประเมินคลาสนั้นเป็นเรื่องที่ทำความเข้าใจได้ยากและไม่สามารถระบุได้ชัดเจนนัก ดังนั้นจึงควรมีการป้องกันหรือมีมาตรฐานในการออกแบบคลาสเพื่อลดความซับซ้อน นักวิจัยหลายท่านได้แสดงความคิดเห็นไว้ เช่น McGregor และ Sykes[36], Johnson และ Foote[31] และ Wirfs-Brock กับ Wilkerson[53] เป็นต้น สามารถสรุปได้ดังนี้

- โมดูลรับการติดต่อจากภายนอกควรอยู่ในรูปแบบสาธารณะ
- ขั้นตอนการทำงานภายในคลาสไม่ควรเปิดเผยให้ทราบ ถ้ามีการเรียกใช้งานจากภายนอกควรติดต่อผ่านโอเปอร์เรเตอร์เท่านั้น
- การปรับปรุงเปลี่ยนแปลงข้อมูลควรกระทำอยู่ภายในคลาสเดียวกัน
- โอเปอร์เรเตอร์แต่ละตัวสามารถเข้าถึงและแก้ไขได้ตลอดเวลา (ภายในคลาสเดียวกัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- พยายามทำให้คลาสมีความเป็นอิสระจากคลาสอื่นให้มากที่สุด
- การส่งผ่านค่าระหว่างกันควรมีความชัดเจนและรวบรัด
- ในกรณีที่คลาสลูกสืบทอดคุณสมบัติมาจากคลาสแม่ โมดูลการติดต่อควรครอบคลุมทั้งคลาสแม่และลูก
- องค์ประกอบที่อยู่ภายในคลาสแม่อยู่ในลักษณะทั่วไป เช่น แอตทริบิวต์ เซอร์วิส/โอเปอร์เรชัน เป็นต้น
- ถ้านำคลาสกลับมาใช้ใหม่ควรนำวิธีการสืบทอดคุณสมบัติมาใช้ ในการแก้ไขปัญหาของระบบให้มากที่สุดเท่าที่จะทำได้

จากที่กล่าวมาข้างต้นเป็นเพียงมาตรฐานบางส่วนเท่านั้น การวัดคุณภาพของวิธีการเชิงวัตถุแบ่งออกเป็น ซอฟต์แวร์โปรดักต์ (Software Product) กับกระบวนการซอฟต์แวร์ (Software Process) [46] ซึ่งทั้ง 2 วิธีวัดในทิศทางที่แตกต่างกัน กล่าวคือ วิธีแรกวัดที่การเขียนโปรแกรมเป็นหลัก ซึ่งวัดตั้งแต่การผ่านค่าทางพารามิเตอร์ โครงสร้างภายใน ขนาดของโปรแกรมหรือความซับซ้อนของโครงสร้างทางตรรกะถัดมาคือการประมาณความสัมพันธ์ระหว่างโปรดักต์ที่เกิดขึ้น ซึ่งทำให้ทราบถึงความสัมพันธ์ที่เกิดขึ้นภายในวงจรชีวิตและสามารถประเมินค่าใช้จ่ายของผลิตภัณฑ์ได้ง่ายขึ้นอีกด้วย

การวัดคุณภาพของโมเสสสามารถแบ่งออกเป็น 3 ชนิด คือ การวัดความซับซ้อนภายในคลาส (Intraclass Metrics) การวัดความซับซ้อนระหว่างคลาส (Interclass Metrics) และโมเดลการวัดความซับซ้อน CCM (Congitive Complexity Model Metrics) [8],[20] ดังกล่าวต่อไปนี้

<u>การวัดความซับซ้อนภายในคลาส (Intraclass Metrics)</u>

การวัดขนาดและความซับซ้อนของโปรแกรมเชิงวัตถุมีอยู่หลายวิธี แต่ในโมเสสเลือกวิธีของ McCabe กับ Thomas และ Jacobson มาใช้ในการวัด โดยทั้ง 2 วิธีมีลักษณะการวัดที่แตกต่างกัน กล่าวคือ วิธีของ McCabe[35] วัดความซับซ้อนของโปรแกรมโดยวัดแบบ LOC (Line of Code) โดยใช้ค่าไซโคลเมติค (Cyclomatic Number) ที่วัดจากโฟลว์กราฟของโปรแกรม ซึ่งเริ่มวัดจากโหนดเริ่มต้นจนถึงโหนดสิ้นสุดและมีสูตรการคำนวณดังนี้

$$v(F) = e - n + 2$$

โดยที่ F มี arcs จำนวน e เส้น และมี n โหนด ค่าไซโคลเมติคเป็นการวัดจำนวนเส้นทางที่เป็นอิสระเชิงเส้น (linearly independent paths) ของ F การวัดนี้จะเป็นไปตามวัตถุประสงค์และเป็นประโยชน์เมื่อนับเส้นทางที่เป็นอิสระเชิงเส้นได้ถูกต้อง ค่าไซโคลเมติคสามารถคำนวณได้อีกวิธีหนึ่งดังนี้

$$v(F) = 1 + d$$

โดยที่ d คือจำนวนโหนดที่มีทิศทางออกจากตัวเอง (out degree) มากกว่า 1 ซึ่งอยู่ใน F นั้น จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีการวัดที่ McCabe แนะนำไว้ค่า V ของโมดุลควรมีค่าระหว่าง 3-7 แต่ไม่ควรเกิน 10 จึงจะเป็นค่าที่เหมาะสม นอกจากนี้การวัดความซับซ้อนของโปรแกรมยังมีอีกลักษณะหนึ่งคือ พิจารณาจากระดับโครงสร้างของโปรแกรมทั้งหมดจากโฟลว์กราฟ และกำหนดค่าปัจจัยความซับซ้อน (ev) ไว้ดังนี้

$$ev(F) = v(F) - m$$

โดยที่ m คือจำนวนของโฟลว์กราฟย่อยของ F ซึ่งอยู่ในรูปของ D-Structured prime หรือเรียกว่าโครงสร้าง D จากตัวอย่างโฟลว์กราฟ F มีโฟลว์กราฟย่อยที่เป็นโครงสร้าง D เพียงโครงสร้างเดี่ยวคือ โครงสร้างของ D_1 ซึ่งอยู่ทางขวาของโฟลว์กราฟ ดังนั้น m = 1 และ v(F) = 5 สามารถแทนค่าตามสูตรได้ดังนี้ ev(F) = 5-1 ฉะนั้นค่าปัจจัยความซับซ้อนของ ev(F) จึงมีค่าเท่ากับ 4 เป็นต้น ถ้าโฟลว์กราฟที่นำมาวัดความซับซ้อนมีโครงสร้างของโฟลว์กราฟลักษณะของโครงสร้าง D ประกอบอยู่สามารถลดค่าความซับซ้อนลงได้อีก [1] ส่วนการวัดขนาดของโปรแกรมของ Thomas กับ Jacobson [25] มีสูตรการคำนวณดังนี้

$$s = (AW_A + MW_M)$$

ให้ s แทนคลาส ส่วน A กับ M แทนแอตทริบิ้วและโอเปอร์เรชั่นตามลำดับ ค่าถ่วงน้ำหนักของ W_A และ W_M เป็นค่าคงที่ โดยที่ W_A มีค่าประมาณ 1 และ W_M มีค่าประมาณ 5-20 ซึ่งขึ้นอยู่กับค่าเฉลี่ยของโอเปอร์เรชั่นหรือเซอร์วิสของคลาสที่นำมาวัด แต่กรณีของ Lorenz[33] จะขึ้นอยู่กับขนาดของเซอร์วิสโดยเฉลี่ยและจำนวนของเซอร์วิสที่เฉลี่ยต่อคลาส ซึ่งทั้ง 2 วิธีที่นำเสนอไปนี้สามารถทำให้ผู้พัฒนาทราบถึงความสัมพันธ์และความซับซ้อนของโปรแกรมที่นำเสนอได้เป็นอย่างดี และสามารถนำไปเป็นแนวทางการปรับปรุงระบบ ประมาณค่าใช้จ่ายที่จะเกิดขึ้นต่อไปได้

<u>การวัดความซับซ้อนระหว่างคลาส (Interclass Metrics)</u>

การวัดโครงสร้างของวิธีการเชิงวัตถุนั้นแตกต่างจากการวัดแบบ LOC ซึ่งจะวัดจากความสัมพันธ์ระหว่างคลาส การวัดในลักษณะนี้กระทำกันที่การรับ-ส่งข้อมูลระหว่างกัน ซึ่งเรียกว่า "fan-in / fan-out" fan-in หมายถึง การรับข้อมูลเข้าจากภายนอกมาทำประมวลผลกับข้อมูลภายใน ส่วน fan-out หมายถึง ข้อมูลภายในและภายนอกที่ประมวลผลแล้วส่งออกสู่ภายนอก สูตรการคำนวณมีอยู่ด้วยกันหลายวิธี เช่น วิธีของ Henery กับ Kafura [21] จะมีวิธีการคำนวณดังนี้

1) $fan-in \times fan-out$

2) $(fan-in \times fan-out)^2$

3) $S_s \times (fan-in \times fan-out)^2$

ในวิธีการคำนวณแบบที่ 3 นำขนาดของ LOC ซึ่งแทนด้วย S_s ขณะที่วิธีของ Card กับ Glass [8] ให้ความคิดเห็นว่าขนาดของ LOC นั้นไม่มีความจำเป็นต่อการนำมาวัดค่าความซับซ้อนจึงนำค่า fan-out มาวัดเพียงอย่างเดียวซึ่งสามารถคำนวณได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

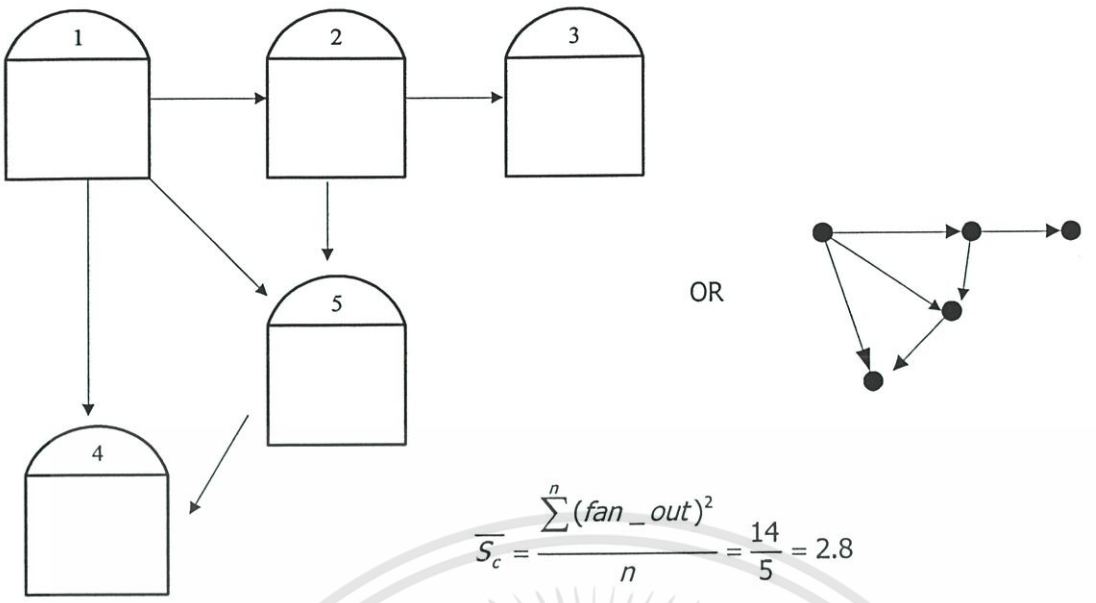
$$\overline{C}_I \equiv \frac{C_I}{n} = \frac{\sum^n (fan-out)^2}{n}$$

หรือ

$$\overline{S}_C = \frac{\sum^n (fan - out)^2}{n}$$

การวัดค่าความซับซ้อนเชิงโครงสร้างของโปรแกรมเชิงวัตถุด้วยการใช้ค่าจาก fan-in และ fan-out นั้น มีค่ามาตรฐานของการวัด กล่าวคือ ถ้าหากผลลัพธ์จากการคำนวณจากสูตรปรากฏว่า ค่า fan-out มีค่าในระดับต่ำ และค่า fan-in มีค่าในระดับสูง แสดงว่าการออกแบบคลาสและความสัมพันธ์กับคลาสอื่น ๆ อยู่ในเกณฑ์ที่ดี ในทางตรงกันข้าม หากค่า fan-out และ fan-in มีค่าในระดับสูงเหมือนกันแสดงว่าการออกแบบคลาสนั้นควรนำกลับไปสู่ขั้นตอนการออกแบบอีกครั้ง[8] วิธีการคำนวณสามารถแสดงดังในรูปที่ 4.34 สิ่งสำคัญอีกประการหนึ่งที่สามารถแสดงถึงความซับซ้อนของความสัมพันธ์ที่เกิดขึ้นระหว่างคลาส คือ ลำดับของการสืบทอดคุณสมบัติจากคลาสหลักไม่ควรเกิน 7 ± 2 ชั้น เพราะว่าความลึกและค่าเฉลี่ยนั้นเป็นประโยชน์ต่อการพัฒนาระบบได้และสามารถแสดงถึงอัตราส่วนของการนำกลับมาใช้ใหม่กับลักษณะเฉพาะของคลาสย่อยและคลาสหลักโดยอัตราส่วนของการนำกลับมาใช้ใหม่บอกถึงจำนวนของคลาสหลักทั้งหมดและจำนวนของการนำมาใช้ของระบบ ส่วนอัตราส่วนของลักษณะเฉพาะแสดงถึงจำนวนของคลาสย่อยที่มีต่อคลาสหลักพร้อมกับแสดงถึงลักษณะของ ADT ที่มีอยู่ในคลาสหลัก[20] อย่างไรก็ตามการวัดความซับซ้อนเชิงโครงสร้างของการสืบทอดคุณสมบัติยังมีความจำเป็นอยู่ระหว่างคุณลักษณะเฉพาะทางกับการอิมพลีเมนต์ แม้ว่าภายในสามารถแสดงถึงความเกี่ยวข้องกันระหว่างเซอร์วิสหรือการเข้าถึงโครงสร้างข้อมูลภายในคลาสก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.34 แสดงการคำนวณ fan-in และ fan-out

แบบจำลองการวัดความซับซ้อน CCM (Cognitive complexity model metrics)

แบบจำลองการวัดความซับซ้อนโดยใช้แบบจำลองของ CCM ที่พัฒนาโดย Henderson-Sellers กับ Cant และคณะ[20]ใช้หลักการวิเคราะห์จากโปรแกรมที่เขียนขึ้นโดยโปรแกรมเมอร์และนำกระบวนการของโปรแกรมมาประเมินผลตามแบบจำลอง ซึ่งสามารถอธิบายได้ 2 ลักษณะคือ อธิบายด้วยกราฟ หรือที่เรียกว่า "landscape Diagram" กับสูตรทางคณิตศาสตร์เข้ามาอธิบาย การวัดด้วยวิธีนี้มีเทคนิคอยู่ 2 วิธีคือ "Chunking" กับ "Tracking" เทคนิคแรกจับกลุ่มของโปรแกรมที่อยู่ในทิศทางเดียวกัน (ตรรกะที่มีความเกี่ยวข้องกัน) เช่น การทำซ้ำ (Loop) ในบล็อกเดียวกัน เป็นต้น ส่วนที่เหลือของโปรแกรมจะจัดให้อยู่ในกลุ่มเทคนิคของโปรแกรม โดยมีสูตรการคำนวณตามลำดับดังนี้

$$C_i = R_i + \sum_{j \in N} C_j + \sum_{j \in N} T_j$$

โดยให้ R_i แทน chunk ที่กำลังสนใจ C_j แทน chunk ก่อนหน้าหรือระดับก่อนหน้า T_j แทน track ก่อนหน้าหรือระดับก่อนหน้า N คือจำนวนของ chunk ทั้งหมดที่มีอยู่ในโปรแกรม

การวัดความซับซ้อนของระบบทั้งหมดที่กล่าวมาแล้วนั้น เป็นสิ่งที่มีความสำคัญและจำเป็นสำหรับโครงการทางธุรกิจเป็นอย่างยิ่ง เพราะเหมือนกับการตรวจสอบคุณภาพของซอฟต์แวร์ให้มีประสิทธิภาพก่อนนำไปใช้งานจริงซึ่งระหว่างการพัฒนาระบบควรทำกิจกรรมนี้ควบคู่กันไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิจกรรมที่ 14 : แผนงานของการพัฒนา (Scenario Development)

วัตถุประสงค์หลักของแผนงาน คือการอธิบายถึงลำดับของการทำงานที่เกิดขึ้นโดยเริ่มตั้งแต่ O/C โมเดลและเหตุการณ์ พร้อมกับกำหนดการติดต่อที่เกิดขึ้นภายในระบบ ซึ่งนับว่าเป็นรูปแบบของการเข้าถึงความต้องการของผู้ใช้อย่างแท้จริง

การสัมภาษณ์ผู้ใช้ระบบเป็นสิ่งหนึ่งที่มีความจำเป็นอย่างยิ่งสำหรับการพัฒนาแผนงาน เพราะทำให้ทราบถึงลำดับการทำงานของระบบงานทางธุรกิจและสามารถนำมากำหนดลำดับการทำงานภายในระบบได้ง่ายยิ่งขึ้น ซึ่งลักษณะของการทำงานที่เกิดขึ้นจะครอบคลุมตั้งแต่กรณีปกติจนถึงกรณีที่ผิดปกติที่มีเงื่อนไขและไม่มีเงื่อนไข บางกรณีแผนงานอาจมีลักษณะการดำเนินงานที่แตกต่างกันแต่อยู่ภายใต้การทำงานเดียวกันก็ย่อมได้

สิ่งแรกที่ต้องทำคือกำหนดผู้ใช้ระบบ (Actor) ที่เป็นทั้งผู้ใช้งาน (บุคคล) หรือระบบอื่น ๆ ที่เข้ามาติดต่อเพื่อขอใช้ข้อมูลภายในหรือส่งออกไปสู่ระบบภายนอกระบบที่กำลังพัฒนา ซึ่งผู้ใช้ระบบสามารถช่วยให้กำหนดขอบเขตของระบบได้ชัดเจนมากยิ่งขึ้น การระบุรายละเอียดของผู้ใช้ระบบมีดังนี้

Name: ชื่อผู้ใช้ระบบ
Description: วัตถุประสงค์และการทำงานที่เกิดขึ้นกับผู้ใช้ระบบ
Scenario: ชื่อของแผนงานที่ผู้ใช้ระบบนั้นต้องการใช้งาน
Contact: ชื่อบุคคลหรือระบบอื่น ๆ ที่ต้องทำการติดต่อด้วย

คำถามแรกที่ควรถามผู้ใช้ระบบ คือ "คุณต้องการใช้ข้อมูลจากใครหรือจากระบบใดบ้างสำหรับงานที่คุณรับผิดชอบ ?" ที่ถามเช่นนั้นเพราะต้องการทราบถึงความสัมพันธ์ที่เกิดขึ้นระหว่างการทำงาน และนำรายละเอียดเหล่านั้นมากำหนดอุปกรณ์และระบบที่เกี่ยวข้อง ซึ่งผู้ใช้ระบบมีผลต่อการพัฒนาระบบ เพราะผู้ใช้แต่ละคนมีความต้องการที่แตกต่างกัน ดังนั้นเพื่อกำหนดทิศทางและขอบเขตของระบบที่จะพัฒนาให้ตรงกับความต้องการจึงต้องนำรายละเอียดจากการสัมภาษณ์มาประกอบเพื่อที่จะทำให้ระบบออกมาถูกต้องและตรงกับความต้องการมากที่สุด ดังนั้นการตั้งคำถามจึงต้องตั้งให้กระชับและชัดเจนมากที่สุด เช่น

- อะไรคืองานหลักของผู้ใช้แต่ละคนที่ต้องทำในระบบ ?
- ผู้ใช้จำเป็นต้อง อ่าน เขียน เปลี่ยนแปลง ข้อมูลในระบบทุกครั้งหรือไม่ ?
- ข้อมูลจากภายนอกมีผลกระทบกับข้อมูลภายในหรือไม่ ?
- ข้อมูลที่มีอยู่ต้องการปรับปรุงหรือเปลี่ยนแปลงทันทีหรือไม่ ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรายละเอียดของแต่ละแผนงานกับความต้องการของผู้ใช้ของขั้นตอนก่อนหน้านี้ผู้พัฒนาระบบสามารถนำมาประกอบเข้าด้วยกันเพื่อกำหนดฟังก์ชั่นของระบบเพื่อให้เหมาะสมกับผู้ใช้แต่ละคน ซึ่งบางครั้งแผนงานอาจมีรายละเอียดที่คล้ายกันหรือเหมือนกัน แต่ฟังก์ชันที่ผู้ใช้ที่การใช้อาจไม่เหมือนกัน ดังนั้นแผนงานจึงแบ่งออกเป็น 2 กรณี คือกรณีทั่วไป (basic course) กับ กรณีผิดปกติ (Alternative course) กรณีทั่วไป คือ แผนงานที่มีฟังก์ชันตรงตามความต้องการของผู้ใช้มากที่สุด ส่วนกรณีที่ผิดปกติมีความแตกต่างจากกรณีแรก เพราะแผนงานงานที่เกิดขึ้นมีลักษณะของแผนงานผิดแปลกไปจากเดิม หรือมีข้อผิดพลาดทางเทคนิค (error condition) ที่ไม่สามารถนำมากำหนดฟังก์ชั่นให้กับผู้ใช้ได้เนื่องจากไม่ครอบคลุมและไม่ชัดเจน แต่ในบางครั้งแผนงานที่มีข้อผิดพลาดเหล่านี้มีส่วนสำคัญกับระบบธุรกิจเพราะทำให้ระบบงานมีความชัดเจนและเข้าใจมากยิ่งขึ้น ผู้พัฒนาจึงควรนำมาแผนงานนั้นมาสร้างเป็นฟังก์ชันขึ้นมา ซึ่งอาจทำขนานไปกับแผนงานปกติหรือทำภายหลังก็ได้

การสร้างแผนงานหลาย ๆ แผนงานควบคู่กันไป Jacobson แนะนำว่า "ควรใช้วิธีการดำเนินการแบบขยาย (extend) เข้ามาช่วย"[25] ซึ่งวิธีนี้จะพัฒนาเพียงแผนงานเดียวแล้วค่อยขยายไปสู่แผนงานที่เกี่ยวข้องหรืออยู่ในขอบเขตที่จะพัฒนา โดยจะต้องมีลักษณะดังนี้

- เป็นส่วนหนึ่งของแผนงานหลัก
- เป็นแผนงานที่ไม่มีความซับซ้อนหรือเป็นทางเลือก
- เป็นแผนงานมีลักษณะพิเศษและแตกต่างจากแผนงานปกติและต้องสามรถพัฒนาได้

ถ้าแผนงานมีลักษณะพิเศษหรือเฉพาะทางจำนวนหลายแผนงานควรนำมารวมไว้ในแผนงานเดียวกันเพื่อสะดวกต่อการเรียกใช้ การพัฒนาแผนงานใหม่ที่เพิ่มเติมจากงานหลักจะไม่มีผลกระทบแต่อย่างไร เพราะแผนงานหลักจะกระทำต่อเนื่องจนกว่าจะถึงแผนงานที่พัฒนาขึ้นมาใหม่และทำต่อเนื่องกันไปทันที ถ้างานหลักไม่ถูกระงับไปเสียก่อน

กิจกรรมที่ 15 : การกำหนดเซอร์วิส (Service Identification)

วัตถุประสงค์ของกิจกรรมนี้ คือ การกำหนดเซอร์วิสให้กับวัตถุโดยพัฒนาจากพฤติกรรมของวัตถุนั้น ๆ เซอร์วิสที่กำหนดจะอยู่ในระดับของคลาสมากกว่าระดับของวัตถุตามที่เข้าใจกัน เพราะการกำหนดที่ระดับนี้จะทำให้เซอร์วิสนั้นสามารถครอบคลุมได้มากกว่า เมื่อคลาสถูกเรียกใช้งานระดับวัตถุจะสร้างวัตถุขึ้นมาใหม่พร้อมกับรายละเอียดต่างให้เหมาะสมกับวัตถุนั้น ๆ ออกมาด้วย เช่น การกระทำ แอตทริบิวต์ เซอร์วิส หรือ สเตท เป็นต้น ซึ่งการกำหนดสามารถกระจายออกเป็นกิจกรรมย่อยดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โอเปอร์เรชัน (Operations) คือ การกำหนดการทำงานของวัตถุ ซึ่งสามารถกำหนดได้จากแผนงานระดับต้น ๆ เพื่อไม่ได้เกิดความผิดพลาดกับแผนงานที่ศึกษาไว้ ในบางครั้งโอเปอร์เรชันอาจเตรียมไว้สำหรับงานเพียงงานเดียวแต่ในบางครั้งอาจมีผลมากกว่าหนึ่งก็ได้
- คุณสมบัติ (Properties) คือ เซอร์วิสชนิดหนึ่งที่อยู่ในรูปของการส่งค่ากลับของวัตถุที่สัมพันธ์ด้วย การส่งค่ากลับของดาต้าอาจจะอยู่ในรูปของแบบฟอร์มหรือตาราง การกำหนดคุณสมบัติสามารถกำหนดใน O/C ได้ตามเอกสารความต้องการของผู้ใช้

กิจกรรมที่ 16 : การทำงานรวมกันของระบบย่อย (Subsystem Coordination)

การทำงานระหว่างระบบย่อยมีส่วนสำคัญมากสำหรับโครงงาน เพราะในโครงงานหนึ่งสามารถแบ่งการพัฒนาออกเป็นส่วน ๆ ให้กับแต่ละทีมงานนำไปพัฒนาตามแผนงานที่วางไว้ ซึ่งแต่ละระบบจะมีการแก้ไขปัญหาและวัตถุประสงค์ไม่เหมือนกัน บางครั้งทีมงานที่รับผิดชอบหรือระบบย่อยอื่นอาจเข้ามาเรียกใช้คลาสที่อยู่ในระบบไปใช้งาน สิ่งที่แต่ละทีมมีควรกระทำคือ ทำสำเนาของคลาสและรายเอียดต่าง ๆ ที่มีอยู่ในระบบของตน เพื่อให้ทีมงานอื่นสามารถนำไปใช้งานได้ง่ายและสะดวกยิ่งขึ้น

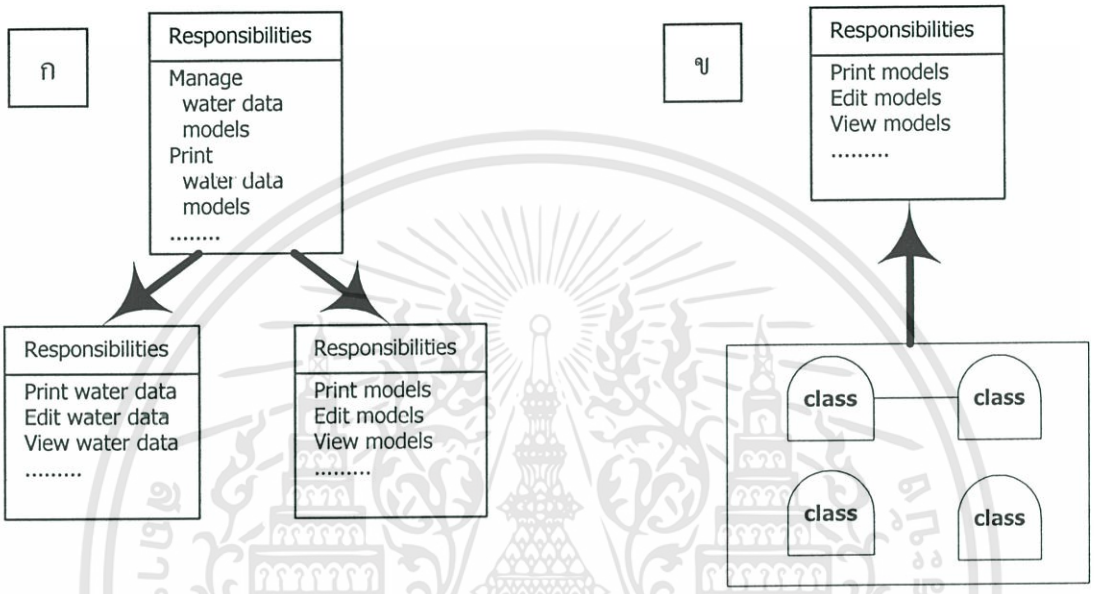
กิจกรรมที่ 17 : การกำหนดระบบย่อย (Subsystem Identification)

การแบ่งระบบย่อยนั้นมีความสำคัญมากสำหรับระบบขนาดใหญ่และมีความซับซ้อนมาก วิธีและเทคนิคที่ใช้สำหรับพัฒนาระบบเชิงวัตถุ คือ วิธีบนลงล่างกับเทคนิค “Divide and Conquer” ที่นำเสนอโดย de Champeaux[11] ระบบย่อยที่พัฒนาโดยนักออกแบบหรือผู้จัดการโครงงานจะอยู่ในความรับผิดชอบของการพัฒนาซอฟต์แวร์ทั้งสิ้นและแต่ละระบบจะแสดงรายละเอียดที่สำคัญต่าง ๆ เช่น ลักษณะของระบบ คุณสมบัติของระบบ งานที่รับผิดชอบ ความสัมพันธ์ของคลาสที่ซ่อนอยู่ภายใน เป็นต้น นอกจากนี้ระบบหลักสามารถมองเห็นภาพรวมและความเกี่ยวข้องทั้งหมดของระบบเพื่อสามารถนำไปใช้งานได้ เช่น ระบบช่วยตัดสินใจ (Decision Support System)[12] ที่ต้องการเพียงรายงานทางเลือกสำหรับการตัดสินใจ แต่ต้องมีส่วนประกอบจากระบบย่อยคอยส่งข้อมูลให้ (ระบบย่อยที่จัดการเกี่ยวกับฐานข้อมูล รูปแบบการตัดสินใจ (ModelBase) และระบบจัดการอินเทอร์เฟส) เป็นต้น

การแบ่งระบบขนาดใหญ่ออกเป็นส่วนย่อย ๆ ใช้วิธีบนลงล่างซึ่งคล้ายกับการกระจายกระบวนการในการพัฒนาเชิงโครงสร้าง โดยแบ่งความรับผิดชอบของแต่ละระบบตั้งแต่ระดับบนสุดไปจนถึงระดับสุดท้ายที่ไม่สามารถกระจายได้ ดังแสดงในรูปที่ 4.35(ก) แสดงระบบย่อยที่เกี่ยวข้องโดยแสดงข้อมูลและโมเดล เช่น การจัดการข้อมูลและโมเดล พิมพ์ข้อมูล พิมพ์เอกสารต่าง ๆ ของโมเดล เป็นต้น ซึ่งจากรูปสังเกตว่าระบบจะกระจายจากระดับบนสุดลงมายังระบบย่อยทั้งสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยแบ่งเป็นสองส่วน คือ ระบบทางซ้ายจัดการเกี่ยวกับข้อมูลทั้งหมด เช่น การพิมพ์ การแก้ไข การแสดงผล ส่วนระบบทางขวาจัดการเกี่ยวกับโมเดลทั้งหมดเช่น การพิมพ์ แก้ไข แสดงผล เป็นต้น การกระจายเซอร์วิสให้กับระบบย่อยใช้วิธีล่างขึ้นบนเข้ามาช่วยเหลือโดยใช้เพียง abstract ของคลาสที่มีอยู่ใน O/C เท่านั้น ซึ่งทำให้ทราบถึงอ๊อบเจ็คที่ต้องรับผิดชอบกับระบบดังกล่าว รูปที่ 4.35(ข) แสดงกลุ่มของคลาสที่มีหน้าที่จัดการเกี่ยวกับข้อมูลของโมเดลให้กับระบบย่อยที่ต้องการใช้สำหรับการพิมพ์ แสดงผล แก้ไข ของโมเดลทั้งหมด



รูปที่ 4.35 ก) แสดงการกระจายคลาสของระบบย่อยไปสู่ระบบย่อยอื่น
ข) แสดงการกระจาย Abstract ของคลาสในระบบย่อยไปสู่ระบบย่อยอื่น

ระบบย่อยในโมเสสเป็นเพียงการแสดงกลุ่มโครงสร้างของคลาสที่เกี่ยวข้องและสัมพันธ์กันเท่านั้น ไม่ใช่แสดงถึงระดับแนวคิดของเอนติตีและไม่สนับสนุนการนำไปสู่ขั้นตอนการเขียนโปรแกรมถึงแม้ว่าจะอยู่ในสภาวะแวดล้อมของซอฟต์แวร์ก็ตาม นอกจากนี้ขั้นตอนการแบ่งระบบย่อยยังสามารถเกิดขึ้นได้ในทุกขั้นตอนของ Specification เพราะการเกิดแต่ละครั้งทำให้มีการทำซ้ำของระบบและมีรายละเอียดเพิ่มมากขึ้น ส่วนการวัดความซับซ้อนของระบบย่อยจะวัดจากจำนวนความสัมพันธ์กับระบบย่อยอื่น ๆ ที่มีการเชื่อมต่อกัน

อย่างไรก็ตามการแบ่งระบบออกเป็นส่วนย่อยมักทำกับระบบที่มีขนาดใหญ่และมีความซับซ้อนเพื่อจัดการกับระบบได้สะดวกยิ่งขึ้น ซึ่งวิธีนี้เป็นอีกวิธีหนึ่งที่นักวิชาการหลายท่านให้คำแนะนำไว้ เช่น Wirfs-Brock[54], Coad and Yourdon[39],[40], Booch[13], Rumbaugh[28] เป็นต้น ข้อดีคือ ทำให้ผู้พัฒนาระบบมีความมั่นใจมากขึ้นกับขั้นตอนการออกแบบของตนโดยเฉพาะเซอร์วิสที่ได้ออกแบบไว้ตรงกับความต้องการหรือไม่ เพราะอาจมีเซอร์วิสที่นำมาใช้กับระบบย่อยจำนวนหนึ่งนำมาจำลองการใช้งาน ซึ่งถ้าไม่ตรงหรือขาดไปสามารถเพิ่มเข้าไปใหม่ได้ ถ้าจะทำให้เกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความมั่นใจมากยิ่งขึ้นนั้นในทุกวงจรชีวิตและการพัฒนาระบบสารสนเทศนั้นควรมีการทำต้นแบบ (prototype) เพื่อทดสอบด้วยทุกครั้ง ซึ่งในแบบจำลองน้ำพุก็ทำเช่นนี้

กิจกรรมที่ 18 : การทดสอบ (Testing)

การทดสอบเริ่มทำหลังจากผ่านขั้นตอนการออกแบบแล้ว ซึ่งเข้าไปทดสอบทั้งวงจรชีวิตในระยะเวลาอันสั้น เริ่มตั้งแต่พิสูจน์ภายในระบบ (Verification) และตรวจความถูกต้อง (Validation) ซึ่งงานทางวิศวกรรมซอฟต์แวร์ คำว่า "Verification" มักมีความสัมพันธ์กับคำถามที่ว่า "ระบบที่กำลังสร้างถูกต้องหรือไม่ ?" ส่วนคำว่า "Validation" สัมพันธ์กับคำถามที่ว่า "กำลังสร้างระบบถูกต้องหรือไม่ ?" การทดสอบสามารถแบ่งออกเป็น ทดสอบตามผลลัพธ์ที่ได้ (Specification-base หรือ blackbox) กับ ทดสอบการทำงานของโปรแกรม (program-base หรือ whitebox) ซึ่งต่างจากการทดสอบแบบสแตติกและไดนามิก แบบแรกประเมินการทำงานของกระบวนการและไม่คำนึงถึงการทำงานของกระบวนการภายใน แบบถัดมาจะเข้าไปตรวจสอบถึงการทำงานในแต่ละกระบวนการอย่างละเอียด

กิจกรรมที่ 19 : การย้ายไปสู่การเขียนโปรแกรมเชิงวัตถุ (Translation to the OOPL)

<u>กิจกรรมที่ 19.1 การอิมพลีเมนต์ทางภาษาเชิงวัตถุ</u>

ในขั้นตอนอิมพลีเมนต์ แต่ละเซอร์วิสจะถูกเขียนขึ้นด้วยโค้ดของโปรแกรมทำให้การออกแบบอัลกอริทึมเป็นสิ่งสำคัญที่สุด การเขียนโปรแกรม คือ การนำสิ่งที่ได้บันทึกไว้ใน PL อาจเป็นโครงสร้างของโปรแกรม pseudocode หรือ โครงสร้างภาษาธรรมชาติ เช่น ภาษาอังกฤษ เป็นต้น ระหว่างการอิมพลีเมนต์บางครั้งถ้ามีการเปลี่ยนแปลงขึ้นระหว่างการเขียนผู้พัฒนาไม่สามารถที่จะทราบได้ ผู้พัฒนาทำได้เพียงตรวจสอบรายละเอียดของโปรแกรมเพื่อหาความสัมพันธ์ของโปรแกรมที่เกิดจากการพัฒนาเท่านั้น ดังนั้นสิ่งที่ต้องคำนึงถึงระหว่างการเขียนโปรแกรมควรจะมีข้อตกลงที่ชัดเจนและสามารถเกิดขึ้นจริงได้สำหรับการอิมพลีเมนต์

<u>กิจกรรมที่ 19.2 การอิมพลีเมนต์โครงสร้าง</u>

ระหว่างการอิมพลีเมนต์โมเดลของซอฟต์แวร์ การกำหนดความสัมพันธ์ระหว่างคลาสผู้พัฒนาควรระบุให้ชัดเจน ซึ่งการระบุความสัมพันธ์สำหรับโครงสร้างของคลาสนั้น ควรใช้ตัวชี้ ที่มีลักษณะเป็น พอยเตอร์ (Pointer) ตัวอ้างอิง (Reference) หรือคลาส เป็นตัวระบุความสัมพันธ์ที่เกิดขึ้น การอ้างอิงความสัมพันธ์ระหว่างกันซึ่งทำกับฝั่งผู้อ้างและฝั่งผู้ถูกอ้าง เช่น เซอร์วิสของคลาส B อ้างถึงเซอร์วิสของคลาส A จะปรากฏคำร้องของของคลาส B ภายใต้คลาส A เป็นต้น โดยปกติของการอ้างอิงระหว่างโครงสร้างของคลาสมักจะทำในทิศทางเดียว เพื่อลดความซ้ำซ้อนของความสัมพันธ์ระหว่างคลาส ซึ่งสังเกตได้จากระบบไคล์เอ็นต์-เซิร์ฟเวอร์ กล่าวคือ ฝั่งใดทำหน้าที่ใดมักทำหน้าที่นั้นเพียงอย่างเดียว เช่น ฝั่งไคล์เอ็นต์มีหน้าที่ของบริการจะมีหน้าที่ขอบริการเพียงอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนเซิร์ฟเวอร์มีหน้าที่ให้บริการตามคำขอจะทำหน้าที่ให้บริการเพียงอย่างเดียวเท่านั้น จึงทำให้แต่ละฝั่งมีหน้าที่ของตนเองที่ชัดเจน ดังนั้นการกำหนดความสัมพันธ์ของคลาสควรทำภายใต้ O/C ไดอะแกรมเพื่อแสดงให้เห็นถึงโครงสร้างของความสัมพันธ์ในแต่ละคลาส ความเหมาะสมของการอ้างอิงขึ้นอยู่กับความเหมาะ ความจำเป็นและการนำกลับมาใช้ใหม่ เพราะฉะนั้น ระหว่างขั้นตอนอิมพลีเมนต์ควรระบุหน้าที่ของคลาสให้ชัดเจน เพื่อลดการสูญเสียของการเชื่อมต่อระหว่างคลาส ซึ่งการสูญเสียนี้ผู้พัฒนาสามารถนำเครื่องมือมาตรวจสอบความสัมพันธ์ได้ โดยใช้วิธีการวัดความซับซ้อนแบบเชิงวัตถุ (Object-Oriented Matrix) ซึ่งการวัดนี้สามารถทำให้ผู้พัฒนาทราบถึงความชัดเจน ความซับซ้อน และความถูกต้องของคลาสมากขึ้น

กิจกรรมที่ 20 : User Requirements Elicitation

วัตถุประสงค์ของกิจกรรมนี้ คือ การกลั่นกรองรูปแบบและกำหนดความต้องการของผู้ใช้ให้ชัดเจน ซึ่งเอกสารต่าง ๆ จะถูกเตรียมในรูปของภาษาที่ผู้ใช้เข้าใจง่าย ซึ่งกล่าวถึงรายละเอียดต่าง ๆ เช่น รายละเอียดของเวลา ฮาร์ดแวร์และค่าใช้จ่าย เป็นต้น ซึ่งเป็นที่สังเกตได้ว่าขั้นตอนนี้มีลักษณะคล้ายกับการกำหนดขอบเขตของปัญหาและการกำหนดฟังก์ชันของระบบที่ต้องการ

กระบวนการค้นหาความต้องการของผู้ใช้อาจได้จาก การสัมภาษณ์ ลักษณะแผนงานและผลกระทบของแบบจำลอง เป็นต้น การหาความต้องการของผู้ใช้ Graham กล่าวว่า “การค้นหาความต้องการสำหรับการพัฒนาระบบในสภาพแวดล้อมเชิงวัตถุนั้น ควรหาผู้ที่มีประสบการณ์ ในการสัมภาษณ์ ถ้าเป็นไปได้ควรมีถึง 2 คน คนแรกคอยฟังและสนทนากับผู้ใช้ อีกคนหนึ่งคอยจดบันทึกความต้องการลงในแบบฟอร์ม ซึ่งบางครั้งของการสัมภาษณ์อาจต้องมีเครื่องบันทึกเทปไว้สำหรับบันทึก เพื่อนำกลับมาพิจารณาภายหลัง และผู้ที่ทำหน้าที่สัมภาษณ์ควรมีความรู้ทางด้านเทคนิคและเทคโนโลยีเชิงวัตถุด้วย”[22] การสัมภาษณ์แต่ละครั้งควรคำนึงถึงความต้องการในเชิงธุรกิจของผู้ใช้เสมอ เพราะการบันทึกความต้องการนั้นผู้ใช้ไม่ต้องทราบถึงรายละเอียดทางเทคนิคของระบบ เช่น โปรแกรมที่ใช้ ฐานข้อมูลแบบใด เทคนิคหรือโมเดลในการพัฒนาระบบ เป็นต้น แต่สิ่งที่ต้องการกลับเป็นคุณภาพ ประสิทธิภาพและการบำรุงรักษาของระบบมากกว่า หลังจากการสัมภาษณ์กับผู้ใช้เสร็จสิ้นแล้ว ความต้องการต่าง ๆ จะถูกกลั่นกรองเพื่อนำมาวางแผนงานที่เป็นพื้นฐานของการพัฒนาแบบจำลอง ซึ่งทำให้ระบบมีความชัดเจนมากขึ้น ดังนั้นการกลั่นกรองความต้องการจากผู้ใช้ควรทำซ้ำ เพราะการทำซ้ำในเทคนิคเชิงวัตถุสามารถทำได้อย่างอิสระ แม้ว่าไดอะแกรมจะมีลักษณะคล้ายคลึงกันกับ Event Model ก็ตาม

สรุป คือ การสัมภาษณ์ความต้องการของผู้ใช้เป็นสิ่งที่จำเป็นอย่างยิ่ง เพราะสามารถนำมาวางแผนงานสำหรับการพัฒนาระบบและทำให้ระบบตรงความต้องการของผู้ใช้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 สรุป

แบบจำลองน้ำพุ คือ แบบจำลองที่เหมาะสมกับการพัฒนาระบบสารสนเทศเชิงวัตถุ เนื่องจากมีลักษณะการทำซ้ำและการเวียนบังเกิดของแบบจำลองน้ำพุ เป็นรูปแบบการพัฒนาระบบสารสนเทศที่มีความยืดหยุ่นและครอบคลุมทั้งการวิเคราะห์และการออกแบบ จึงทำให้มีลักษณะของการนำกลับมาใช้ใหม่ได้และแบบจำลองนี้อยู่ภายใต้วิธีการของวิศวกรรมซอฟต์แวร์ หรือโมเสส ซึ่งประกอบด้วย Product Lifecycle, Process Lifecycle และกิจกรรมต่าง ๆ ตลอดทั้งวงจรชีวิตของการพัฒนาระบบ

Product Lifecycle คือ วงจรชีวิตทางธุรกิจและการพัฒนาของวงจรชีวิต Product Lifecycle จะแบ่งเป็นโครงการระยะสั้นกับระยะยาว โครงการระยะสั้นมีระยะเวลาการพัฒนาประมาณ 1 ปี และถ้าเป็นโครงระยะยาวมีระยะเวลาการพัฒนามากกว่า 1 ปี โดยโครงการสามารถแบ่งออกเป็นส่วนย่อย ๆ ได้ เพราะทำให้สะดวกในการบริหารและจัดการโครงการนั้น ซึ่งในแต่ละโครงการจะประกอบด้วย Business Planning, Build และ Delivery ซึ่งเป็นขั้นตอนทางธุรกิจทั้งสิ้น ใน Business Planning และ Delivery เป็นขั้นตอนที่มีความสำคัญ เพราะเป็นส่วนเริ่มต้นและตัดสินใจของโครงงาน ส่วน Build กล่าวถึงเทคนิคการพัฒนาระบบสารสนเทศ ซึ่งทั้งหมดจะทำซ้ำ ๆ กันเช่นนี้จนกระทั่งสิ้นสุดโครงการและรวมไปถึงการบำรุงรักษาระบบ

Process Lifecycle คือ วงจรชีวิตของการพัฒนา โดยกล่าวถึงขั้นตอน Build ที่อยู่ใน Product Lifecycle ซึ่งครอบคลุมตั้งแต่ การวางแผน การศึกษาความเหมาะสม วิเคราะห์ออกแบบ อิมพลีเมนต์และการตรวจสอบ ซึ่งทั้งหมดเป็นเทคนิคและขั้นตอนการสร้างระบบสารสนเทศและแต่ละขั้นตอนจะมีกิจกรรมย่อยและเอกสารประกอบตลอดการพัฒนา

สุดท้าย คือ กิจกรรมของการพัฒนาระบบสารสนเทศ ซึ่งประกอบด้วย 20 กิจกรรม โดยที่กิจกรรมเหล่านี้ผู้พัฒนาสามารถเพิ่มหรือลดได้ตามความเหมาะสม ได้แต่ การตรวจสอบเอกสาร การสร้างอีเวนท์ไดอะแกรม การสร้าง O/C ไดอะแกรม และวิธีการตรวจสอบคุณภาพของระบบ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

SSADM ภายใต้แบบจำลองน้ำพุ

จากที่กล่าวถึงในบทที่ 2 บทที่ 3 และบทที่ 4 พบว่าแต่ละวิธีมีข้อดีที่แตกต่างกันออกไป กล่าวคือ วิธีเชิงโครงสร้าง (SSADM) มีลักษณะเด่นที่มีขั้นตอนและกิจกรรมที่ชัดเจนสามารถติดตามได้ตลอดการพัฒนา วิธีเชิงวัตถุมีคุณสมบัติที่สามารถนำกลับมาใช้ใหม่ได้ (Reusable) มีความถูกต้องและตรวจสอบได้ (Correctness and Verification) สามารถขยายและปรับปรุงได้ง่าย (Extensible) มีความเสถียรและคงทน (Robustness) ส่วนแบบจำลองน้ำพุสามารถพัฒนาโครงงานในลักษณะของการทำซ้ำ (Iterative) และการเวียนบังเกิด (Recursive) ได้ดี ดังนั้นการประยุกต์จึงนำลักษณะเด่นของแต่ละวิธีมาผสมผสานเข้าด้วยกัน โดยนำโครงสร้างของ SSADM กับคุณสมบัติทางวิศวกรรมซอฟต์แวร์ของวิธีเชิงวัตถุมาสอดแทรกในแบบจำลองน้ำพุ ทั้งระดับของระบบ (System-Level) และระบบย่อย (Sub-System Level) ซึ่งแต่ละวงกลมของแต่ละระดับจะมีขั้นตอนและกิจกรรมต่าง ๆ ประกอบอยู่ภายในอย่างชัดเจน การพัฒนาตามวิธีนี้คล้ายกับวิธีทั่วไป คือเริ่มจาก ศึกษาความเหมาะสม วิเคราะห์ ออกแบบ นำเสนอ และบำรุงรักษาระบบตามลำดับ แต่วิธีปฏิบัติจะแตกต่างจากการพัฒนาระบบสารสนเทศเชิงวัตถุทั่ว ๆ ไปโดยเริ่มจากการเก็บข้อมูลต่าง ๆ ของผู้ใช้ เช่น นโยบายขององค์กร ข้อจำกัดทางฮาร์ดแวร์และซอฟต์แวร์ ความต้องการของผู้ใช้ เป็นต้น พร้อมกับวาดแผนผังข้อมูลหรือดาต้าโฟลว์ก่อน (กล่าวคือคล้ายกับการวาดแผนผังฟังก์ชันของวิธีเชิงวัตถุ) จากนั้นค่อยแปลงไปสู่วิธีเชิงวัตถุเริ่มจาก วัตถุ แอตทริบิวต์และเมธอดตามลำดับ การที่โมเดลมีลักษณะเช่นนี้เพราะผู้พัฒนาระบบส่วนใหญ่มักจะเคยชินกับการวิเคราะห์และออกแบบเชิงโครงสร้างมากกว่าวิธีเชิงวัตถุ จึงทำให้โมเดลประยุกต์เริ่มต้นวิเคราะห์จากฟังก์ชันการทำงานของระบบก่อนส่วนอื่น ๆ เหมือนกับวิธีเชิงโครงสร้างทั่วไปและทำให้มีข้อดีกับการพัฒนาดังนี้ ประการแรก ในกรณีที่ต้องการนำระบบเดิมที่วิเคราะห์และออกแบบจากวิธีเชิงโครงสร้างมาเปลี่ยนแปลงเป็นวิธีเชิงวัตถุย่อมทำได้ แต่จะต้องมีส่วนประกอบและกฎเกณฑ์อื่น ๆ เพิ่มเติมเพื่อทำให้การเปลี่ยนแปลงนั้นตรงกับวิธีเชิงวัตถุมากขึ้น ประการที่สอง ฟังก์ชันของแต่ละวัตถุเป็นฟังก์ชันที่เกิดขึ้นมีความชัดเจนและเป็นสิ่งที่เกิดขึ้นจริงกับระบบงานนั้น ๆ ประการที่สาม ระดับของแบบจำลองถูกแบ่งเป็น 2 ระดับทำให้สามารถพัฒนาระบบย่อยได้หลายระบบในเวลาเดียวกัน ประการที่สี่ สามารถติดตามโครงงานได้อย่างใกล้ชิดและทราบทุกขั้นตอนตลอดทั้งวงจรหรือที่เรียกว่า “Mile-stone” เหมือนวิธีเชิงโครงสร้างประการสุดท้าย มีเอกสารประกอบในทุก ๆ ขั้นตอนอย่างละเอียด ซึ่งรายละเอียดทั้งหมดจะกล่าวในหัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 หลักการของ SSADM ภายใต้แบบจำลองน้ำพุ

โมเดลประยุกต์แบ่งการพัฒนาออกเป็น 2 ระดับ คือ ระดับของระบบ และ ระดับของระบบย่อย โดยทั้ง 2 ระดับ มีลักษณะเหมือนกับแบบจำลองน้ำพุ คือมีวงกลมแทนขั้นตอนการพัฒนาตั้งแต่ต้นจนจบและแต่ละวงกลมมีขั้นตอนการปฏิบัติที่ชัดเจน

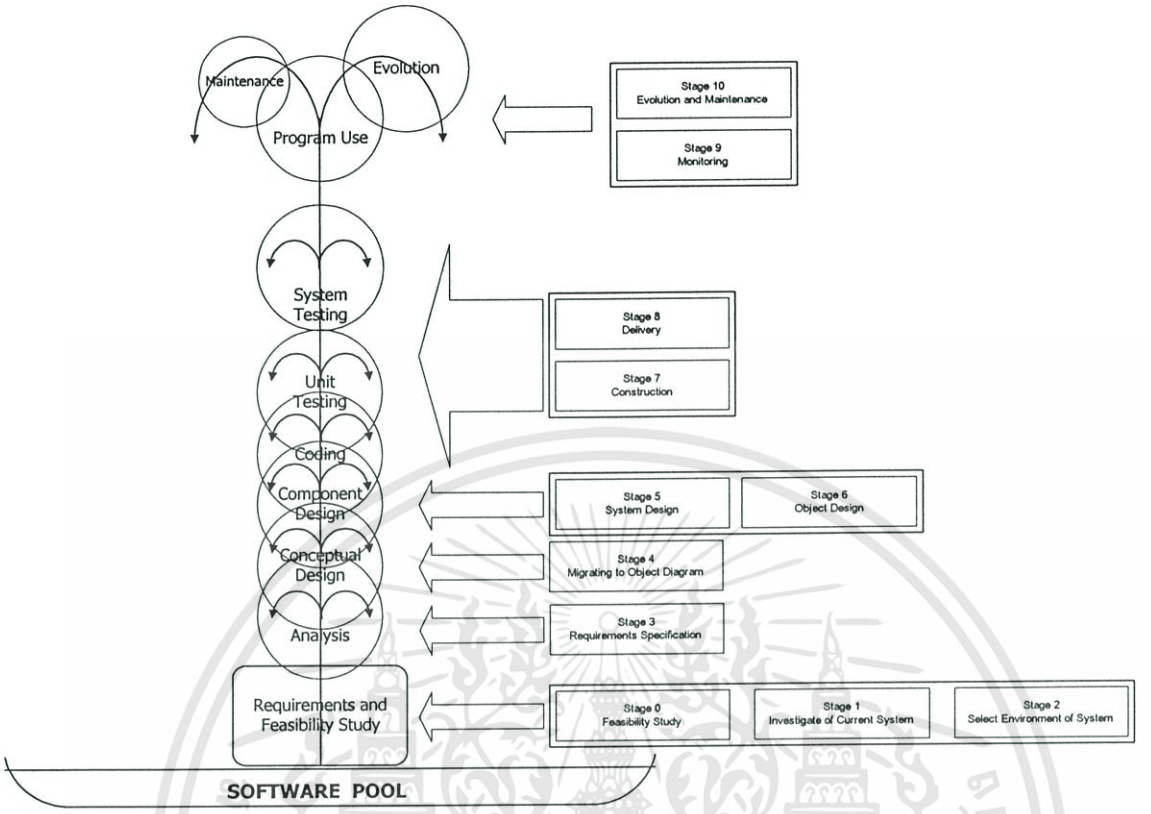
5.1.1 ระดับของระบบ (System-Level)

ระดับของระบบเป็นระดับแรกของแบบจำลองประยุกต์มีหลักการคล้ายกับแบบจำลองน้ำพุ คือ รวมการวิเคราะห์ ออกแบบ อิมพลีเมนต์และการทดสอบของระบบสารสนเทศไว้ที่เดียวกันและระดับของระบบประกอบด้วยระบบย่อยหลายระบบดังนั้นการทำงานระหว่างระบบจึงมีความเกี่ยวข้องกันต่ำกว่าระดับของระบบย่อย แต่มีความเกี่ยวเนื่องของกลุ่มการทำงานสูงกว่าและระบบย่อยจะถูกกำหนดจากระดับของระบบ ซึ่งแตกต่างกันที่ขั้นตอนการพัฒนระบบสารสนเทศมีความชัดเจนกว่า โดยแต่ละวงกลมถูกแทนที่ด้วยระยะและการพัฒนาถูกแทนด้วยขั้นตอน กิจกรรมและงาน เริ่มตั้งแต่ระยะที่ 0-10 โดยแต่ละระยะมีขั้นตอนที่เกี่ยวข้องกันทั้งวงจรชีวิตของการพัฒนา

การพัฒนาระบบสารสนเทศทั่วไปมักเริ่มที่ศึกษาความเหมาะสมและความต้องการของผู้ใช้ก่อนเสมอ โดยผู้พัฒนาไม่สามารถเริ่มเขียนโปรแกรม หรือ ออกแบบโปรแกรมก่อนที่จะรวบรวมความต้องการจากผู้ใช้ได้ ดังนั้นโมเดลประยุกต์มีวิธีปฏิบัติเช่นเดียวกับวิธีอื่น กล่าวคือ เริ่มจากศึกษาความเหมาะสม วิเคราะห์ ออกแบบ อิมพลีเมนต์และทดสอบระบบ ตามลำดับ จนกระทั่งระบบสารสนเทศที่ต้องการบรรลุตามวัตถุประสงค์ที่ต้องการได้

จากรูปที่ 5.1 แสดงการดำเนินงานของระดับของระบบ ระยะที่ 0-2 ประกอบกันอยู่ภายในส่วนของการศึกษาความเหมาะสมกับความต้องการของผู้ใช้ (Requirements and Feasibility Study) ของโมเดลประยุกต์ ระยะที่ 3 อยู่ในส่วนของการวิเคราะห์ (Analysis) ระยะที่ 4 อยู่ในส่วนของการออกแบบระดับแนวคิด (Conceptual Design) ระยะที่ 5-6 อยู่ในส่วนของการออกแบบองค์ประกอบ (Component Design) ระยะที่ 7-8 กระจายอยู่ภายในส่วนของการเขียนโปรแกรม (Coding) การทดสอบโปรแกรม (Unit Test) และทดสอบระบบ (System Test) สุดท้าย ระยะที่ 9-10 อยู่ภายในส่วนของการนำระบบสารสนเทศไปใช้งาน (Program Use, Evolution, Maintenance)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 แสดงการประยุกต์ของโมเดลในระดับของระบบ

5.1.2 ระดับของระบบย่อย (Sub-System Level)

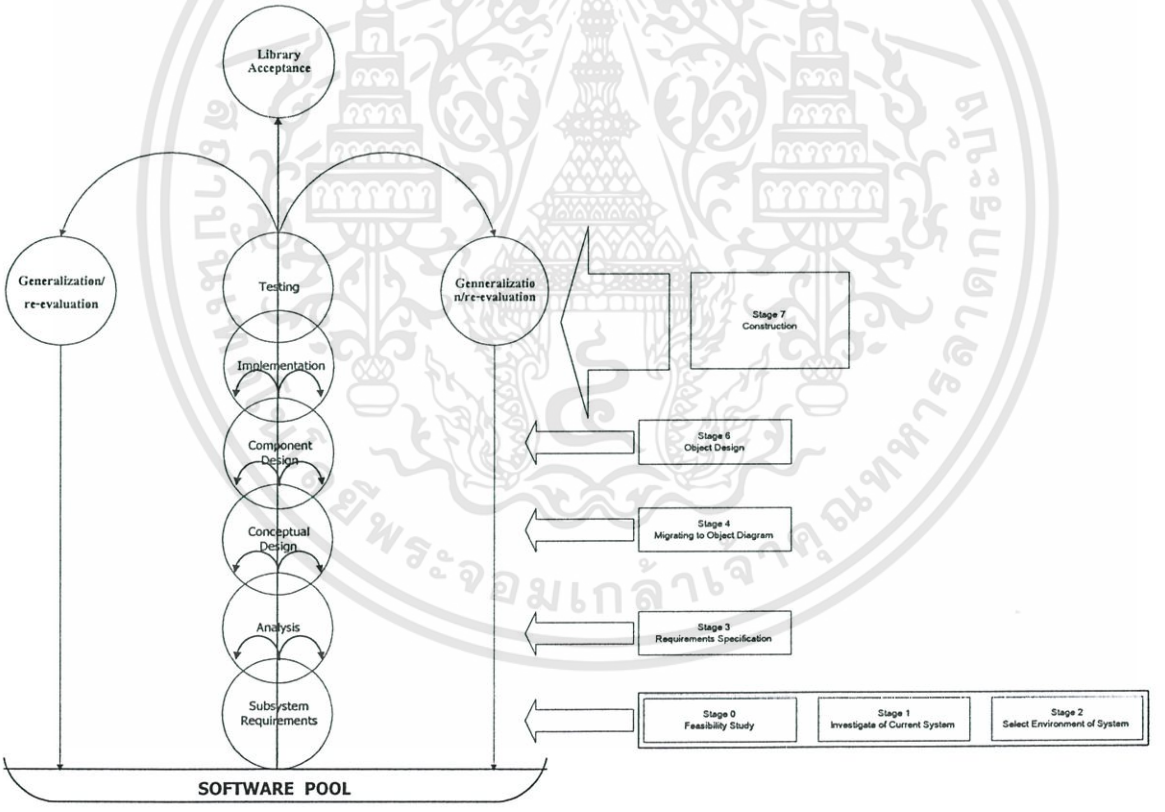
ระดับของระบบย่อยเป็นระดับที่กล่าวถึงการพัฒนาระบบของแต่ละระบบย่อย ซึ่งมุ่งเน้นไปที่การพัฒนาแต่ละระบบย่อยมากกว่า จึงทำให้ความเกี่ยวเนื่องของกลุ่มย่อยนั้นมากกว่าระดับของระบบ แต่การพัฒนายังคงให้ความสำคัญกับการพัฒนาลักษณะทำซ้ำ (IDP - Iterative Development Process) การพัฒนาของระบบย่อยส่วนใหญ่สามารถนำกลับมาใช้ใหม่ได้ เพราะแต่ละระบบย่อยเป็นการพัฒนาที่เฉพาะเจาะจง (Specification) หรือเป็นการพัฒนาที่ระบบย่อยอื่นสามารถนำไปใช้งานต่อไปได้ (Generalization) และระบบย่อยที่ถูกแบ่งสามารถพัฒนาแบบขนานกันได้ แม้ว่าวงจรชีวิตในแต่ละกลุ่มมีความแตกต่างกันก็ตาม การแบ่งการพัฒนาออกเป็นระบบย่อยจะขึ้นอยู่กับขอบเขตของระบบสารสนเทศที่ต้องการพัฒนา เพราะบางครั้งการพัฒนาระบบไม่จำเป็นต้องมีการแบ่งระบบย่อยเสมอไป

จากรูปที่ 5.2 กล่าวถึงขั้นตอนการพัฒนาของระบบย่อยที่สอดแทรกขั้นตอนการพัฒนา เริ่มตั้งแต่ระยะที่ 0-2 จะกระจายอยู่ในส่วนของความต้องการของระบบย่อย (Sub-System Requirements) ระยะที่ 3 อยู่ภายในส่วนของการวิเคราะห์ (Analysis) ระยะที่ 4 อยู่ภายในส่วนของการออกแบบระดับแนวคิด (Conceptual Design) ระยะที่ 6 อยู่ภายในส่วนของการออกแบบระดับองค์ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Component Design) ระยะที่ 7 อยู่ภายในส่วนของการอิมพลีเมนต์และทดสอบ (Implementation and Testing)

สาเหตุที่ระยะที่ 5 8 9และ10 ไม่ถูกรวมในการพัฒนาระดับของระบบย่อย เพราะแต่ละระบบย่อยเป็นเพียงการพัฒนาเฉพาะกลุ่มไม่ใช่การพัฒนาทั้งระบบและเมื่อแต่ละระบบย่อยถูกพัฒนาเรียบร้อยแล้วจะถูกรวบรวมอีกครั้งในขั้นตอนของระดับของระบบ ระยะที่ 5 หรือ ขั้นตอนการออกแบบระบบไม่ถูกรวมอยู่ในส่วนของการออกแบบระดับองค์ประกอบ เพราะ ระยะนี้กล่าวถึงวิธีการแบ่งระบบออกเป็นส่วนย่อย ระยะที่ 8 เป็นขั้นตอนส่งมอบระบบที่พัฒนาเสร็จเรียบร้อย ระยะที่ 9 เป็นขั้นตอนการนำไปใช้งานของระบบสารสนเทศ ส่วนระยะที่ 10 เป็นการปรับปรุงระบบสารสนเทศหลังจากนำไปใช้งานและเกิดการเปลี่ยนแปลงกับระบบสารสนเทศที่รับผิดชอบ ดังนั้นขั้นตอนทั้ง 4 จึงไม่จำเป็นต้องนำไปเป็นส่วนหนึ่งของการพัฒนาของระดับของระบบย่อย (รายละเอียดของขั้นตอนทั้ง 4 จะกล่าวในหัวข้อ 5.5)



รูปที่ 5.2 แสดงการประยุกต์ของโมเดลในระดับระบบย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ลักษณะของแบบจำลองประยุกต์

แบบจำลองประยุกต์ เป็นแบบจำลองลักษณะผสม (Hybrid Model) กล่าวคือ ถ้าพิจารณาระดับของระบบกับระดับของระบบย่อยจะมีลักษณะการพัฒนาแบบลำดับขั้น (Sequential Model) ซึ่งคล้ายกับแบบจำลองน้ำตก ส่วนภายในแต่ละวงกลมของแบบจำลองประยุกต์นั้นมีการวนซ้ำของการพัฒนาในแต่ละขั้นตอนจนกระทั่งงานที่เกิดขึ้นมีความสมบูรณ์และถูกต้องตามความต้องการทำให้มีลักษณะคล้ายกับแบบจำลองสไพเริ่ล (Spiral Model)

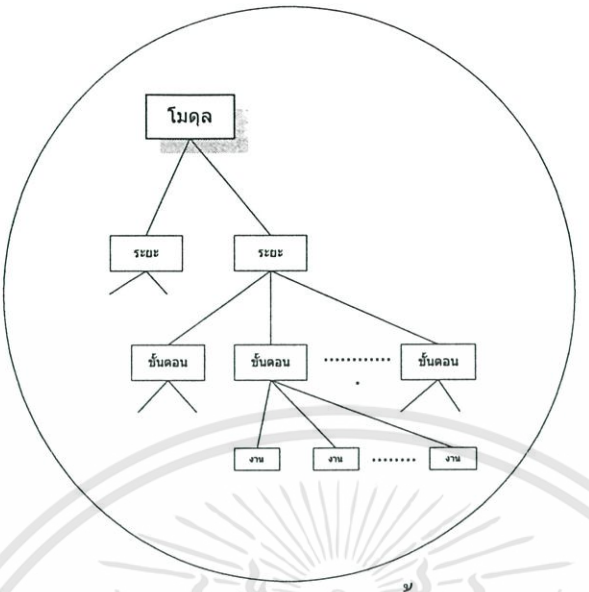
5.3 วิธีการประยุกต์ SSADM ภายใต้แบบจำลองน้ำพุ

วิธีการสอดแทรกขั้นตอน กิจกรรมและงานของวิธีเชิงโครงสร้าง (SSADM) ให้กับแบบจำลองน้ำพุ เกิดขึ้นในส่วนของ Process Lifecycle เพราะแบบจำลองน้ำพุอยู่ภายใต้วิธีการของโมเสสซึ่งแบ่งออกเป็น 2 ส่วน คือ Product Lifecycle กับ Process Lifecycle (ดังกล่าวในบทที่ 4) เพราะส่วนที่ทำให้เกิดการพัฒนาระบบสารสนเทศนั้น มีจุดเริ่มต้นที่ Process Lifecycle เป็นส่วนใหญ่ ถึงแม้ว่า Product Lifecycle มีส่วนที่ทำให้เกิดการพัฒนาก็ตาม แต่เป็นส่วนน้อย โดยขั้นตอนที่นำเข้าไปสอดแทรกจะเริ่มตั้งแต่ระยะที่ 0-10 กิจกรรมและงานแต่ละขั้นตอนของแต่ละระยะจะกระจายอยู่ทั้งวงจรชีวิตของการพัฒนาระบบที่ใช้แบบจำลองประยุกต์ทั้งระดับของระบบและระดับของระบบย่อย (บางขั้นตอน) ซึ่งทำให้วิธีการพัฒนาระบบสารสนเทศด้วยโมเดลที่ประยุกต์มาจากแบบจำลองน้ำพุมีขั้นตอนการพัฒนาที่ชัดเจนและสามารถติดตามได้ เหมือนกับวิธีเชิงโครงสร้าง (แต่ระบบสารสนเทศที่ได้ยังคงอยู่ในหลักการและแนวคิดเชิงวัตถุ)

5.4 โครงสร้างของ SSADM ภายใต้แบบจำลองน้ำพุ

โครงสร้างภายในของแบบจำลองประยุกต์แบ่งกิจกรรมการทำงานเป็นแบบบนลงล่าง ในระดับแรกแบ่งเป็น โมดูล จากโมดูล แบ่งเป็น ระยะ (Stage) จาก ระยะ เป็นขั้นตอน (Step) และจากขั้นตอนเป็น งาน ซึ่งเป็นส่วนย่อยที่สุดของแบบจำลอง ที่สามารถปฏิบัติ ตรวจสอบและประเมินผลได้ ดังแสดงในรูปที่ 5.3 โดยโครงสร้างของแบบจำลองมีลักษณะคล้ายกับแผนภาพต้นไม้

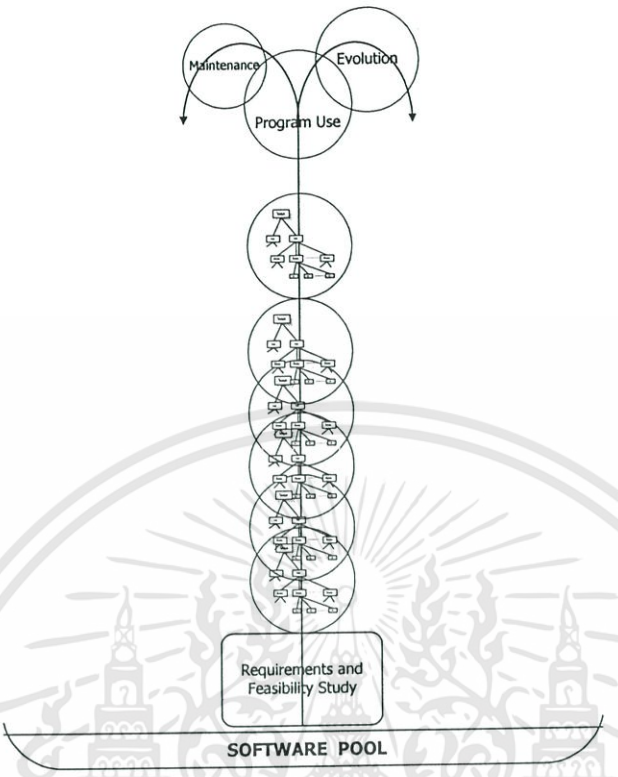
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดงโครงสร้างของ SSADM ภายใต้แบบจำลองน้ำพุ

ในภาพรวมของการประยุกต์แบบจำลองน้ำพุที่ประยุกต์กับวิธี SSADM ในแต่ละวงกลมจะประกอบด้วยขั้นตอนที่ลักษณะแบบแผนภาพต้นไม้ในทุก ๆ วงกลมตลอดทั้งวงจรของการพัฒนาของแบบนจำลองนี้ (ดังแสดงในรูปที่ 5.4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงภาพรวมของการประยุกต์แบบจำลองน้ำพุกับวิธี SSADM

5.5 ขั้นตอนและกิจกรรมของการพัฒนาระบบภายใต้โมเดลประยุกต์

ขั้นตอนการพัฒนาภายใต้โมเดลประยุกต์แบ่งออกเป็น 5 โมดูล ดังนี้

- Requirements and Feasibility Study Module
- Analysis Module
- Design Module
- Implementation and Testing Module
- System Support Module

โดยแต่ละโมดูลมีขั้นตอนและกิจกรรมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1 โมดูลศึกษาความเหมาะสมและความต้องการของผู้ใช้ (Requirements and Feasibility Study Module)

โมดูลศึกษาความเหมาะสมและความต้องการของผู้ใช้เป็นโมดูลแรกของการพัฒนาระบบของแบบจำลองประยุกต์ ซึ่งวัตถุประสงค์หลักมุ่งไปทางการเก็บรวบรวมข้อมูลทั่วไปของระบบ เช่น นโยบายขององค์กร ปัญหาต่าง ๆ ที่เกิดขึ้นกับระบบเดิม ความต้องการของผู้ใช้และการจัดเตรียมทางเลือกของสภาพแวดล้อมของระบบพร้อมกับจัดทำตารางการพัฒนา เป็นต้น ดังนั้นการปฏิบัติตามขั้นตอนต่าง ๆ ของโมดูลนี้ นักวิเคราะห์ระบบจะต้องทำการเก็บข้อมูลทุกอย่างที่ต้องการด้วยตนเอง ซึ่งบางครั้งการรับข้อเสนอหรือรับทราบปัญหาที่เกิดขึ้นทางแบบสอบถามนั้นไม่เพียงพอสำหรับการนำไปแก้ไขปัญหาให้กับระบบ กล่าวคือ อาจต้องลงไปปฏิบัติงานจริงกับพนักงานระดับปฏิบัติการ หรือลงไปคลุกคลีกับปัญหานั้นเพื่อให้ทราบถึงที่มาของปัญหาอย่างแท้จริง ซึ่งการทำในลักษณะนี้ทำให้ทราบแนวทางการแก้ไขปัญหาที่ถูกต้องและถูกวิธี จากนั้นนักวิเคราะห์จะนำเสนอแนวทางและสภาพแวดล้อมโดยทั่วไปของระบบทั้งเก่าและใหม่ตามลำดับ ซึ่งที่กล่าวมาเป็นเพียงบางส่วนเท่านั้น ในโมดูลนี้แบ่งขั้นตอนการพัฒนาออกเป็น 3 ขั้นตอน โดยเริ่มจาก ขั้นตอนที่ 0-2 คือ ศึกษาความเหมาะสมของระบบ ศึกษาความเป็นมาของระบบเดิมและนำเสนอสภาพแวดล้อมทั้งทางด้านธุรกิจและเทคโนโลยีที่มีโอกาสเป็นไปได้ในระบบใหม่ ซึ่งทั้ง 3 ขั้นตอนมีรายละเอียดและขั้นตอนดังต่อไปนี้

5.5.1.1) ระยะที่ 0 การศึกษาความเหมาะสม (Stage 0 Feasibility Study)

การศึกษาความเหมาะสมเป็นขั้นตอนที่กล่าวถึง รายละเอียดทั้งหมดของระบบเดิมโดยละเอียด เช่น ปัญหาของระบบ ความต้องการของระบบและแผนงานทั้งหมด เป็นต้น โดยแบ่งเป็นกิจกรรมหลักและกิจกรรมย่อย ซึ่งกิจกรรมหลักแบ่งออกเป็น 7 กิจกรรม คือ ศึกษาความเหมาะสมของระบบ กำหนดปัญหาหรือค้นหาปัญหาที่เกิดขึ้นกับระบบเดิม จัดทำโครงสร้างการพัฒนา จัดทำแนวทางเลือกทางธุรกิจและเทคโนโลยีที่ต้องการในระบบใหม่ จัดทำตารางการพัฒนา จัดทำโครงสร้างอินพุตและเอาท์พุตของระบบแบบคร่าว ๆ และรวบรวมเอกสาร กิจกรรมต่าง ๆ ที่กล่าวมานี้จะประกอบด้วยกิจกรรมย่อยซึ่งจะกล่าวต่อไป แต่ละกิจกรรมจะมีเอกสารและไดอะแกรมพร้อมกับคำอธิบายแสดงอย่างชัดเจน เช่น ตารางของเอนติตี ดาต้าโฟล์วไดอะแกรมและรายละเอียดของเอนติตี เป็นต้น

<u>1)ขั้นตอน 010 การเตรียมศึกษาความเหมาะสม (Step 010 Prepare for the Feasibility Study)</u>

ขั้นตอนนี้เป็นเพียงขั้นเริ่มต้นสำหรับศึกษาความต้องการและปัญหาของระบบซึ่งข้อมูลที่เก็บจะเกี่ยวกับปัญหาที่เกิดขึ้นในทางธุรกิจกับการวางแผนการพัฒนาระบบ โดยทำการรวบรวมสิ่งต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น ข้อมูลทั้งหมดขององค์กร ความต้องการ ขอบเขตของระบบที่ต้องการ เป็นต้น จากนั้นนำรายละเอียดต่าง ๆ เหล่านี้มาแปลงเป็นแบบฟอร์มของการพัฒนาระบบ คือ แผนภาพการไหลของข้อมูล (DFM – Data Flow Modeling) ในลักษณะภาพรวม (Context Diagram) สร้างแผนภาพข้อมูล (LDM – Logical Data Modeling) แบบคร่าว ๆ จัดลำดับความต้องการของผู้ใช้ในแต่ละฟังก์ชั่น กำหนดขอบเขตของระบบ จัดทำแผนการพัฒนาระบบ เป็นต้น ซึ่งแบบฟอร์มและเอกสารต่าง ๆ เหล่านี้จะนำไปใช้ในกิจกรรมต่อไป การรวบรวมข้อมูลควรจะเก็บรายละเอียดต่าง ๆ ให้ครอบคลุมมากที่สุด ซึ่งถ้ารวบรวมไม่ครบหรือขาดไปยอมทำให้มีผลกระทบกับระบบที่พัฒนาอย่างเห็นได้ชัด ตัวอย่างเช่น ระบบเงินเดือนขาดข้อมูลการบันทึกเวลาการเข้า-ออกการทำงาน ผลคือ ทำให้ไม่สามารถคิดคำนวณเงินเดือนให้กับพนักงานประเภทรายวันได้ เป็นต้น ดังนั้นการรวบรวมข้อมูลควรจะครอบคลุมส่วนสำคัญของระบบให้ได้มากที่สุดเท่าที่จะทำได้ ซึ่งกิจกรรมที่ 010 มีขั้นตอนการทำงานดังนี้

งานของขั้นตอน 010

งานที่ 10 กำหนดขอบเขตของโครงสร้างโดยการสร้างกระบวนการและดาต้าโมเดลของระบบเดิมอย่างกว้าง ๆ

งานที่ 20 ตกลงขอบเขตของโครงการพร้อมกับเอกสารที่ได้ทำการศึกษาความเหมาะสม

งานที่ 30 วางแผนสำหรับการศึกษาความเหมาะสม

งานที่ 40 สร้างแผนภูมิการไหลของข้อมูล (ระบบงานเดิม)

งานที่ 50 สร้างแผนภาพข้อมูล (LDS)

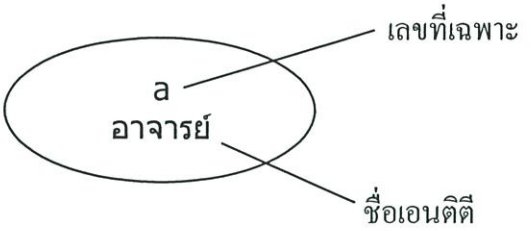
สัญลักษณ์และเอกสาร

สัญลักษณ์สำหรับการพัฒนาแผนผังการไหลของข้อมูล (Data Flow Diagram)

เอนติตีภายนอก (External Entity)

เอนติตีภายนอกที่เกี่ยวข้องกับระบบแสดงอยู่ในรูปวงรี ภายในประกอบด้วย ชื่อเอนติตี และเลขที่เฉพาะของแต่ละเอนติตีซึ่งจะใช้เป็นตัวอักษรหรือตัวเลข ดังแสดงในรูปที่ 5.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงเอนติตีภายนอกที่เกี่ยวข้องกับระบบ

กระบวนการ (Process)

กระบวนการ คือ การทำงานในระบบ แสดงในรูปสี่เหลี่ยมผืนผ้าและประกอบด้วย หมายเลข สถานที่และชื่อของกระบวนการ ดังแสดงในรูปที่ 5.6

รูปที่ 5.6 แสดงกระบวนการการทำงานของระบบ

ดาต้าสโตร์ (Data Store)

ดาต้าสโตร์ คือ สัญลักษณ์แทนแหล่งเก็บมูลซึ่งแสดงในรูปสี่เหลี่ยมผืนผ้าปลายเปิด ภายในประกอบด้วย เลขที่ ประเภทและชื่อของดาต้าสโตร์ ดังแสดงในรูปที่ 5.7

รูปที่ 5.7 แสดงสัญลักษณ์ของดาต้าสโตร์

ประเภทของดาต้าสโตร์สามารถแบ่งออกเป็น 3 แบบ คือ D, M และ T โดยแต่ละประเภทมีรายละเอียดดังนี้

ดาต้าสโตร์ประเภท D คือ ดาต้าสโตร์ที่ถูกจัดเก็บและดำเนินการโดยระบบงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดาต้าสโตร์ประเภท M คือ ดาต้าสโตร์ที่มีการกระบวนการจัดเก็บและทำงานแบบปกติหรือข้อมูลที่อยู่ในรูปของแฟ้มเอกสาร

ดาต้าสโตร์ประเภท T คือ ดาต้าสโตร์ที่ถูกสร้างขึ้นแบบชั่วคราว

ดาต้าโฟลว์ (Data Flows)

เส้นแสดงทิศทางการไหลของข้อมูลมีลักษณะเป็นลูกศรพร้อมทิศทางการไหลและลักษณะของข้อมูลที่เดินทางไปสู่กระบวนการหรือดาต้าสโตร์ที่เกี่ยวข้องกัน ดังแสดงในรูปที่ 5.8

รูปที่ 5.8 แสดงเส้นทิศทางการไหลของข้อมูล

สัญลักษณ์ของ LDM(Logical Data Model)

เอนติตี(Entity)

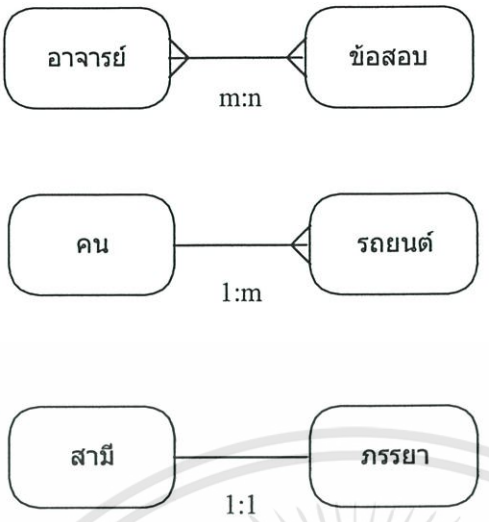
เอนติตี คือ สัญลักษณ์ของดาต้าสโตร์ใน LDS ไดอะแกรม ซึ่งแสดงในรูปสี่เหลี่ยมผืนผ้าปลายมนทั้งหมด ภายในแสดงชื่อของเอนติตี ดังแสดงในรูปที่ 5.9

รูปที่ 5.9 แสดงเอนติตีของ LDS

ความสัมพันธ์ของเอนติตี

การแสดงความสัมพันธ์ระหว่างเอนติตี เช่น ความสัมพันธ์แบบ 1:1, 1:M, M:N โดยเส้นแสดงความสัมพันธ์เป็นเส้นตรง เส้นความสัมพันธ์แบบ 1:1 แสดงเป็นเส้นตรง ความสัมพันธ์แบบ 1:M ปลายของเส้นความสัมพันธ์จะมีลักษณะเป็นสามง่าม ส่วนความสัมพันธ์แบบ M:N เส้นความสัมพันธ์ทั้ง 2 ข้างเป็นสามง่ามทั้งหมด ดังแสดงในรูปที่ 5.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10 แสดงความสัมพันธ์ของเอนติตี

ตารางและเอกสาร

ตารางอธิบายแผนภาพรวมระดับกายภาพของระบบเดิม

ตารางอธิบายแผนภาพรวมระดับกายภาพของระบบเดิมเป็นตารางที่เกิดขึ้นพร้อมกับแผนภาพรวมของระบบเดิม ซึ่งอธิบายถึงความสัมพันธ์ของเอนติตีภายนอกกับระบบที่เกิดขึ้น ภายในตารางกล่าวถึง เอนติตีภายนอก งานที่รับเข้าและออกระหว่างเอนติตีกับระบบ (Source or Recipient) และดาต้าโฟลว์ ดังแสดงในตารางที่ 5.1

ตารางที่ 5.1 แสดงความพันธ์ของเอนติตีภายนอกกับระบบเดิม

External Entity	Source/Recipient	Data Flow
อาจารย์	S	ข้อสอบ
	R	เกณฑ์ข้อสอบ

ตารางระบุความต้องการของผู้ใช้ (Requirement Catalogue)

ตารางระบุความต้องการของผู้ใช้ คือ ตารางสำหรับระบุถึงการทำงานของฟังก์ชันที่ผู้ใช้ต้องการ ภายในตารางประกอบด้วยรายละเอียดดังนี้

1. ที่มาของความต้องการ โดยต้องระบุให้ชัดเจนถึงแหล่งที่มาของความต้องการนี้ เช่น มาจากใคร มาจากแผนกใด เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ระดับของความต้องการ เช่น เป็นฟังก์ชันที่จำเป็นที่ต้องมี (Essential) เป็นฟังก์ชันที่ร้องขอธรรมดา (Desirable) หรือมีก็ได้ไม่มีก็ได้ (nice to have/optional) เป็นต้น

3. ผู้รับผิดชอบกับฟังก์ชันที่ร้องขอ

4. หมายเลขของความต้องการ

5. ความต้องการของฟังก์ชันที่ต้องการให้เกิดขึ้นกับระบบใหม่ (Function Requirement)

6. ลักษณะการทำงานของฟังก์ชันในระบบ เป็นแบบ ออนไลน์ หรือ ออฟไลน์

7. รายละเอียดการทำงานของฟังก์ชันที่เกิดขึ้น

8. ระบุถึงประโยชน์ของการมีฟังก์ชันนั้นในระบบ

9. ข้อเสนอแนะและทางแก้ไขของฟังก์ชันในกรณีที่ ไม่สามารถทำได้หรือเกิดปัญหา

10. ระบุหมายเลขของความสัมพันธ์ที่ฟังก์ชันนี้ไปสัมพันธ์กับเอกสารอื่น ซึ่งบางฟังก์ชันอาจต้องไปเชื่อมโยงกับความต้องการของฟังก์ชันอื่น ๆ เช่น ฟังก์ชันของการให้เกณฑ์ข้อสอบ ไปสัมพันธ์กับ ฟังก์ชันการบันทึกข้อสอบลงฐานข้อมูล เป็นต้น

11. ระบุหมายเลขของความสัมพันธ์ที่ฟังก์ชันนี้ไปเชื่อมโยง

12. ระบุแนวทางการทำงานของฟังก์ชันที่ต้องการ เพื่อยืนยันถึงความเป็นไปได้ของฟังก์ชันนั้นที่จะเกิดขึ้นบนระบบใหม่

รายละเอียดที่กล่าวมาข้างต้นควรแสดงอยู่ในรูปของตารางดังแสดงในตารางที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2 แสดงความต้องการของผู้ใช้

Source: Priority: E User Responsible : Req.Id 02			
Functional Requirements : ให้เกณฑ์ข้อสอบกับข้อสอบที่มีอยู่			
Non-Functional Requirement(S) :			
Description	Target Value	Acceptable Range	Comments
เกณฑ์ข้อสอบ ถ้าไม่ใส่เกณฑ์ข้อสอบ	มี 4 อย่าง ให้ค่าเป็น 0		1. ความยาก 2. ความตรง 3. ความน่าเชื่อถือ 4. อำนาจจำแนก
Benefits : ได้ข้อสอบที่มีเกณฑ์วัดมาตรฐาน			
Comments/Suggested solution:			
Relate Documents: No. 1			
Related Requirements:			
Resolution:			

2) ขั้นตอน 020 กำหนดปัญหา (Step 020 Define the Problem)

หลังจากรวบรวมข้อมูลต่าง ๆ จากขั้นตอน 010 เรียบร้อยแล้ว จากนั้นนำรายละเอียดเหล่าต่าง ๆ มาเปรียบเทียบถึงความเป็นไปได้ของระบบเดิมกับความต้องการของระบบใหม่ถึงความเป็นไปได้หรือสอดคล้องกันหรือไม่ โดยขั้นแรกกิจกรรมนี้สามารถแบ่งวัตถุประสงค์ได้ดังนี้

- ทำความเข้าใจกับฟังก์ชันที่ต้องเกิดขึ้นในระบบใหม่ให้ชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำความเข้าใจกับปัญหาที่เกิดขึ้นกับระบบให้ละเอียดโดยหาข้อบกพร่องและฟังก์ชันของระบบที่ขาดหายไป
- กำหนดผู้ใช้ให้กับระบบใหม่และต้องสอดคล้องกับความต้องการและความรับผิดชอบของผู้ใช้แต่ละคน

ผลลัพธ์ของกิจกรรมนี้ทำให้ได้รายละเอียดของระบบเดิมกับความต้องการของระบบใหม่อย่างละเอียด ซึ่งรายละเอียดต่าง ๆ สามารถที่จะนำมากำหนดขอบเขตของปัญหาของระบบได้และยังสามารถนำมาใช้เป็นบทสรุปสำหรับความต้องการใหม่ ๆ ที่จะย้ายจากระบบเดิมไปสู่ระบบใหม่ได้อย่างชัดเจน และในกิจกรรมที่ 020 มีขั้นตอนการทำงานดังนี้

งานของขั้นตอน 020

งานที่ 10 วิเคราะห์ความต้องการของผู้ใช้ที่มีต่อระบบเดิมโดยศึกษาจากภาพรวมของแผนภูมิการไหลของข้อมูลกับแผนภูมิดาต้า เพื่อนำมากำหนดรูปแบบความต้องการของผู้ใช้

งานที่ 20 นำรายละเอียดของโครงสร้างความต้องการมาขยายแผนภูมิการไหลของข้อมูลกับแผนภูมิการไหลของข้อมูล ที่ได้จากขั้นตอนที่ 010 พร้อมกับบันทึกปัญหาที่เกิดขึ้นกับระบบเดิมลงใน Requirements Catalogue

งานที่ 30 บันทึกความต้องการของผู้ใช้ลงใน User Catalogue

งานที่ 40 จัดหาและเพิ่มเติมฟังก์ชันการทำงานที่ต้องการให้มีในระบบใหม่ โดยแยกประเภทของฟังก์ชัน คือ ฟังกชันที่ต้องการใช้ กับฟังก์ชันที่เป็นสาธารณะ (Non-funtional) พร้อมกับบันทึกรายละเอียดของฟังก์ชันต่าง ๆ ลงใน Requirement Catalogue

งานที่ 50 สรุปปัญหาและความต้องการของผู้ใช้ที่อยู่ใน Requuirement Catalogue

งานที่ 60 นำเสนอปัญหาของระบบให้กับผู้บริหารโครงการรับทราบและอนุมัติ

สัญลักษณ์และเอกสาร

ตารางการทำงานของผู้ใช้ (User Catalogue)

ตารางการทำงานของผู้ใช้เป็นตารางสำหรับอธิบายการทำงานของผู้ใช้แต่ละคนที่ทำกับระบบทั้งระบบเดิมและระบบใหม่ที่จะเกิดขึ้น ภายในประกอบด้วย (ดังแสดงในตารางที่ 5.3)

1. ลักษณะงาน
2. รายละเอียดของงาน
3. ผู้รับผิดชอบในงานนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.3 แสดงตารางการทำงานของผู้ใช้ (User Catalogue)

User Catalogue	
Job Title	Job Activities Description
อาจารย์	ออกข้อสอบตามวิชาที่สอน โดยแยกตามบทและประเภทของข้อสอบ
Despatch Supervisor	อาจารย์ระดับ 4 ขึ้นไป

3) ขั้นตอน 030 Define Scenario Development

ขั้นตอนที่ 030 มีจุดประสงค์ คือ รวบรวมความต้องการของผู้ใช้และศึกษาพฤติกรรมของระบบทั้งหมด โดยใช้ Use Case Model ซึ่งประกอบด้วย ผู้กระทำ (Actor) Use Case และเส้นความสัมพันธ์ระหว่างผู้กระทำกับ Use Case ซึ่งการทำเช่นนี้เปรียบเสมือนกับการทำซ้ำหรือ การทำ 'Double Check' แต่มีข้อแตกต่างกันที่จุดประสงค์ของการนำไปใช้งาน ในกิจกรรมต้น ๆ จะเน้นทางด้านรายละเอียดของฟังก์ชันแบบเชิงโครงสร้างที่สามารถนำไปเขียนโปรแกรม ส่วนในกิจกรรมนี้เน้นลักษณะการทำงานของผู้กระทำที่มีต่อ Use Case นั้นและจะนำส่วนนี้ไปใช้ในระยะที่ 4 ต่อไป ซึ่งรายละเอียดอื่น ๆ หากพิจารณาแล้ว พบว่าทั้งสองส่วนมีความคล้ายกันในบางความต้องการดังนั้นผู้พัฒนาอาจนำเอารายละเอียดบางส่วนไปสอดแทรกไว้ใน Use Case Model ก็อาจเป็นได้ กิจกรรมที่ 030 มีขั้นตอนการทำงานดังนี้

งานของขั้นตอน 030

งานที่ 10 สร้าง Use Case Model โดยกำหนดความต้องการของผู้ใช้หรือลักษณะการทำงานของฟังก์ชันที่เกิดขึ้นภายในระบบ

งานที่ 20 กำหนดผู้กระทำให้กับระบบ ซึ่งผู้กระทำ คือ บุคคลหรือสิ่งที่เกี่ยวข้องกับระบบคล้ายกับเอนติตีของขั้นตอนที่ 010

งานที่ 30 กำหนด Use Case ให้กับระบบ Use Case คือ กำหนดรายละเอียดของระบบที่เกิดขึ้นภายใน เช่น พฤติกรรม รูปแบบของฟังก์ชัน และ วิธีการปฏิบัติงาน เป็นต้น ดังนั้นการกำหนด Use Case ที่เกิดขึ้นกับระบบควรกำหนดตามความเป็นจริงเพื่อจะนำไปกำหนดฟังก์ชันให้กับระบบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์และเอกสาร

ตารางที่ 5.4 แสดงรายละเอียดของ Use Cases

Scenario Name: การให้เกณฑ์ข้อสอบ	ID Number: 4
Short Description: คือการกำหนดเกณฑ์ของข้อสอบในแต่ละวิชาและแต่ละบทที่มีการออกข้อสอบ	
Trigger: ทำงานต่อเมื่อมีการสร้างข้อสอบจากอาจารย์และมีการจัดกลุ่มข้อสอบ Type: **External** / Temporal	

Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
เกณฑ์ข้อสอบ	อาจารย์	ข้อมูลข้อสอบผ่านการจัดเกณฑ์	ข้อสอบมีเกณฑ์
ข้อสอบผ่านการจัดกลุ่ม	ผู้จัดกลุ่ม		

Major Steps Performed	Information for Steps
1. กำหนดเกณฑ์การข้อสอบโดยการให้เกณฑ์แบ่งเป็น 4 ชนิด คือ ความยาก ความตรง ความน่าเชื่อถือและอำนาจจำแนก	เกณฑ์ข้อสอบ
2. กำหนดเกณฑ์จากหลักการจัดเกณฑ์ให้กับข้อสอบแต่ละข้อ แต่ละบท ของแต่ละวิชา	ข้อสอบผ่านการจัดกลุ่ม
3. บันทึกข้อสอบที่ให้เกณฑ์ลงสู่ฐานข้อมูล	ข้อมูลข้อสอบผ่านการจัดเกณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 5.4 แสดงเอกสารของ Use Case ที่อธิบายถึงราบะเอียดของฟังก์ชันที่เกิดขึ้นกับระบบงาน ประกอบด้วย

1. ชื่อของฟังก์ชันที่ปรากฏขึ้นกับระบบ (Scenario Name) คือ ชื่อกระบวนการที่แสดงในแผนผังการไหลของข้อมูล
2. หมายเลขของฟังก์ชัน (ID Number) คือ หมายเลขประจำกระบวนการ
3. รายละเอียดของฟังก์ชัน (Short Description)
4. สิ่งที่กระตุ้นให้ฟังก์ชันทำงาน (Trigger) คือ สิ่งรอบข้างที่เข้ามากระทบกับกระบวนการเพื่อให้กระบวนการนั้นทำงาน
4. ชนิดของฟังก์ชั่น (Type) คือ ประเภทของฟังกชั่นในแผนผังการไหลของข้อมูล ประเภทของฟังก์ชันแบ่งออกเป็น 2 ประเภทคือ External และ Temporal ฟังก์ชันประเภทแรก คือ ฟังก์ชันที่เกิดขึ้นจริงในระบบงาน ส่วนฟังก์ชันประเภท Temporal คือฟังก์ชันที่เกิดขึ้นไม่บ่อยสำหรับระบบงานเหมือนกับยูทิลิตี้ของระบบงาน
5. อินพุตที่เข้ามากระทบกับฟังก์ชันพร้อมสถานที่ต้นทาง (Major Inputs)
6. เอาต์พุตที่ออกจากฟังก์ชันพร้อมสถานที่ปลายทาง (Major Outputs)
7. ขั้นตอนการทำงานของฟังก์ชัน(Major Steps Performed)
8. ขั้นตอนของการไหลของข้อมูลที่เกี่ยวข้องกับฟังก์ชัน (Information for Steps) รายละเอียดในส่วนนี้จะนำรายละเอียดของอินพุตและเอาต์พุตมาใส่แทนที่ จากนั้นโยงความสัมพันธ์ที่เกิดขึ้นกับขั้นตอนการทำงานของฟังก์ชัน

การสร้างตารางรายละเอียดของ Use Cases ผู้วิเคราะห์ต้องสร้างเท่ากับจำนวนฟังก์ชันที่เกิดขึ้นกับระบบ หมายความว่าถ้าระบบที่รับผิดชอบมีฟังก์ชันมากเท่าไรจำนวนของตารางรายละเอียดจะมากตามเท่านั้น

4) ขั้นตอน 040 ออกแบบอินพุต-เอาต์พุต (Define Input / Output of System)

การออกแบบอินพุต-เอาต์พุตและอินเทอร์เฟซ (Interface Design) คือ การออกแบบหน้าจอสำหรับบันทึกและแสดงผลให้กับระบบ เพื่อแสดงให้เห็นถึงการโต้ตอบกันระหว่างผู้ใช้กับโปรแกรม ซึ่งการออกแบบนี้ จะต้องครอบคลุมข้อมูลที่ต้องการสื่อสารกับผู้ใช้ให้มากที่สุด ดังนั้นขั้นตอนที่ 040 นี้จะกล่าวถึงการออกแบบอินพุต-เอาต์พุตและอินเทอร์เฟซแบบคร่าว ๆ หรือที่เรียกว่าการออกแบบลักษณะ “screen layout” ให้กับระบบ โดยหลักการออกแบบนั้นผู้ออกแบบจะต้องคำนึงถึงองค์ประกอบต่าง ๆ อย่างละเอียด ซึ่งทั้งสองแบบมีขั้นตอนการทำงานดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานของขั้นตอน 040

งานที่ 10 กำหนดรูปแบบของอินเทอร์เฟซโดยคำนึงถึง รูปแบบ (Layout) ความสำคัญของเนื้อหา (Content Awareness) มีความสุนทรียภาพ (Aesthetics) ความคงเส้นคงวา (Consistency) ความสามารถของผู้ใช้ (User Experience) มีผลกระทบกับผู้ใช้น้อยที่สุด (Minimal User Effort) และต้องใช้ง่าย (Easy to Use)

งานที่ 20 กำหนดรูปแบบของอินพุตให้กับระบบโดยต้องคำนึงถึงรูปแบบและหลักการต่าง ๆ ดังนี้ ประการแรก คำนึงถึงความสำคัญของข้อมูลที่ป้อนเข้าระบบซึ่งข้อมูลบางตัวอาจจำเป็นต้องปรับปรุงทันทีหรือเก็บไว้ก่อน (on-line or batch processing ประการที่สอง ต้องคำนึงถึงเทคโนโลยีเกี่ยวกับการนำข้อมูลเข้าระบบ เนื่องจากว่าปัจจุบันมีวิธีการป้อนข้อมูลได้หลายวิธี เช่น การป้อนข้อมูลเข้าโดยใช้ แถบรหัสข้อมูล (Barcode) รูปภาพ ลายเซ็นดิจิตอล หรือที่เรียกว่า "Digital Signature" การ์ดไมโครชิฟ (micro chip card) และอื่น ๆ เป็นต้น ทั้งนี้เพื่อให้การป้อนข้อมูลสะดวก ถูกต้อง และรวดเร็วมากยิ่งขึ้น

งานที่ 30 กำหนดรูปแบบของเอาต์พุตให้กับระบบ หลักการ คือ การนำข้อมูลข่าวสารต่าง ๆ ที่มีอยู่มานำเสนอในรายงานให้ครบถ้วนมากที่สุด โดยต้องคำนึงถึงประเภท รูปแบบและรายละเอียดต่าง ๆ เป็นองค์ประกอบสำคัญ ซึ่งประเภทของรายงานแบ่งเป็น 2 ประเภท คือ รายงานตามเวลาจริง (Real-time Reports) กับ รายงานแบบแบตช์ (batch reports) ส่วนรูปแบบของรายงานปัจจุบันมักมีหลายรูปแบบ เช่น ตัวอักษร ตัวเลข ตาราง กราฟ และอื่น ๆ เป็นต้น ซึ่งรูปแบบที่แสดงนั้นต้องคำนึงถึงความเหมาะสมกับรายงานที่ต้องใช้เป็นหลัก เช่น ถ้าเป็นรายงานเปรียบเทียบยอดขายของปีก่อนกับปีปัจจุบัน อาจใช้รายงานประเภทกราฟรูปภาพกับรายงานตัวเลขสรุปยอดขายควบคู่กัน เป็นต้น หรืออาจนำรูปแบบต่าง ๆ มาผสมผสานกันเพื่อออกเป็นรายงานเดียวก็ได้

งานที่ 40 ประมาณเวลาสำหรับการทำงานของระบบที่มีต่อผู้ใช้ในแต่ละกรณีอย่างคร่าว ๆ

ดังนั้นการออกแบบหน้าจอของกิจกรรมนี้จะแสดงเพียงคร่าว ๆ เพื่อให้เจ้าของระบบได้ทราบถึงหน้าจอที่จะปรากฏในระบบ โดยผู้ออกแบบจะแสดงในลักษณะของ Storyboard หรือโครงร่างของหน้าจอ ซึ่งหน้าจอที่แสดงนี้อาจจะไม่ต้องแสดงหมดทั้งระบบเพราะส่วนที่เหลือของการพัฒนาอินเทอร์เฟซจะกล่าวในขั้นตอนของการออกแบบอินพุต-เอาต์พุตต่อไป จุดประสงค์หลักของกิจกรรมนี้คือ นำรายละเอียดต่าง ๆ ที่ได้ก่อนหน้านี้ทั้งหมดมาจัดทำรูปแบบคร่าว ๆ ของอินพุต-เอาต์พุตและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์เฟซให้กับระบบเพื่อที่จะนำไปคิดรวมกับค่าใช้จ่ายทั้งหมดของระบบก่อนลงมือทำและนำเสนอต่อเจ้าของระบบให้ทราบรายละเอียดคร่าว ๆ ก่อนตัดสินใจอนุมัติโครงการ เพราะถ้าผู้พัฒนาระบบไม่ทราบรายละเอียดและความต้องการทั้งหมดทำให้ไม่สามารถที่จะพัฒนาระบบตามความต้องการได้

5) ขั้นตอน 050 กำหนดทางเลือกทางธุรกิจและเทคนิค (Step 050 Select Feasibility Option)

ขั้นตอนที่ 050 เป็นการเสนอทางเลือกทางธุรกิจและทางเทคนิค ซึ่งทางเลือกทั้ง 2 ประเภทนั้นมีคุณสมบัติที่แตกต่างกัน แต่มีจุดประสงค์ที่เหมือนกันคือ ทางเลือกให้กับระบบ โดยทางเลือกทางธุรกิจจะกำหนดให้ครอบคลุมการทำงานทั้งหมดของระบบตั้งแต่ฟังก์ชันพื้นฐานจนถึงความต้องการใหม่ ๆ ให้มากที่สุดเท่าที่จะทำได้ ส่วนการเสนอทางเลือกทางเทคนิค คือ การเสนอเทคโนโลยีที่จะนำมาสนับสนุนระบบซึ่งเทคโนโลยีที่นำเสนอนี้จะต้องครอบคลุมทั้ง สภาวะแวดล้อม ข้อจำกัดต่าง ๆ ของฮาร์ดแวร์กับซอฟต์แวร์ เครื่องมือที่ใช้ในการพัฒนาและการบำรุงรักษาระบบ เป็นต้น ซึ่งการเสนอทางเลือกผู้เสนอควรมีทางเลือกให้กับเจ้าของระบบงานมากพอที่จะตัดสินใจได้ ดังนั้นการจัดเตรียมทางเลือกจะเสนอแนวทางแบบทั่ว ๆ ไป ซึ่งตรงข้ามกับการค้นหาวิธีการแก้ปัญหา โดยแต่ละทางเลือกของ BSOs และ TSOs ต้องอธิบายรายละเอียดทุกข้อที่มีอยู่ในทางเลือกนั้น ๆ การนำเสนอทางเลือกให้กับเจ้าของระบบมีอยู่ด้วยกัน 2 แบบ คือ แบบเต็มรูปแบบ กับ แบบย่อ แบบเต็มจะต้องมีรายละเอียดทุกทางเลือกพร้อมกับรายละเอียดของค่าใช้จ่ายทุกส่วน ส่วนแบบย่อนำเสนอเพียงแต่ทางเลือกไหนครอบคลุมส่วนใดบ้างแต่ยังคงต้องอธิบายทางเลือกแต่ละข้อที่สามารถสนับสนุนการทำงานให้กับระบบอยู่เป็นเกณฑ์ และค่าใช้จ่ายที่เกิดขึ้นก็ไม่จำเป็นต้องแสดงละเอียดเพียงแต่แสดงยอดสรุปเท่านั้นซึ่งขั้นตอนการทำงานมีดังนี้

งานของขั้นตอนที่ 050

งานที่ 10 กำหนดทางเลือกให้กับระบบใหม่อย่างคร่าว ๆ โดยคำนึงถึงความต้องการของผู้ใช้ระบบเป็นหลัก

งานที่ 20 กำหนดทางเลือกทางธุรกิจให้กับระบบทั้ง BSO และ TSO โดยเสนอมาอย่างน้อย 6 ทางเลือก

งานที่ 30 กำหนดทางเลือกทางเทคนิคให้กับระบบโดยทางเลือกทางเทคนิคต้องสนับสนุนทางเลือกทางธุรกิจทั้งหมดที่ได้เสนอมาก่อนหน้านี้

งานที่ 40 นำโครงสร้างของ BSO กับ TSO มารวมกัน จากนั้นนำไปตรวจสอบกับความต้องการของระบบว่าทางเลือกใดสามารถสนับสนุนนความต้องการของระบบได้บ้าง (ซึ่งการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบจะทำกันระหว่าง BSO TSO และ Feasibility Options เท่านั้น) พร้อมกับจัดทำรายงานของทางเลือกต่าง ๆ

งานที่ 50 เพิ่มเติมรายละเอียดของทางเลือกให้กับรายงานของทางเลือกที่ได้จัดไว้

งานที่ 60 จัดทำแผนงานการพัฒนาของระบบแบบคร่าว ๆ เพื่อให้สอดคล้องกับรายละเอียดของทางเลือกต่าง ๆ

งานที่ 70 นำเสนอทางเลือกที่สามารถสนับสนุนโครงการให้กับผู้บริหารเลือก

สัญลักษณ์และเอกสาร

ตารางที่ 5.5 แสดงทางลือกทางธุรกิจ (BSO Matrix)

	BSO 1	BSO 2	BSO 3
1. บันทึกข้อมูลลงฐานข้อมูล	X	X	X
2. ให้เกณฑ์ข้อสอบกับข้อสอบที่มีอยู่	X	X	X
3. คัดเลือกข้อสอบและพิมพ์	X	X	X
4. ลบข้อสอบที่ไม่ต้องการ	X	X	X
5. สอบถามข้อมูล	X	X	X

จากตารางที่ 5.5 แสดงทางเลือกทางธุรกิจให้กับโครงการซึ่งทางเลือกที่นำเสนอนั้นผู้วิเคราะห์ต้องนำเสนออย่างน้อย 3 ทางเลือกเพื่อให้เจ้าของโครงการสามารถตัดสินใจในการดำเนินงานได้

ท้ายสุดของกิจกรรมนี้จะได้ข้อเสนอทางเลือกทั้ง BSOs และ TSOs ที่นำมาประกอบกับการวางแผนพัฒนาระบบต่อไป แต่ที่สำคัญที่สุดคือ ทุกทางเลือกควรแสดงรายละเอียดให้มากที่สุด ทั้งรูปแบบย่อและรูปแบบเต็ม ทั้งนี้เพื่อประโยชน์ทั้งเจ้าของระบบและผู้พัฒนาระบบให้มีความเข้าใจความต้องการของระบบตรงกัน

6) ขั้นตอน 060 การจัดทำตารางดำเนินการ (Define Project Management)

การวางแผนการพัฒนาระบบนับว่ามีความจำเป็นอย่างยิ่ง ในขั้นตอนที่ 060 จึงกำหนดให้มีการจัดทำตารางการวางแผนการดำเนินงานของโครงการ โดยแบ่งออกเป็น Gantt Chart หรือ PERT/CPM และตารางทรัพยากร ซึ่งภายในกิจกรรมนี้จะแสดงภาพรวมทั้งหมดของระบบ Gantt Chart หรือ PERT/CPM แสดงถึง เทคนิคการวางแผน การกำหนดงาน การควบคุมและการพัฒนาของระบบงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการวางแผนจะแยกโครงการออกเป็นกิจกรรมต่าง ๆ พร้อมกับกำหนดระยะเวลาโดยประมาณสำหรับแต่ละกิจกรรมที่เกิดขึ้นแต่ละระดับของระบบ ส่วนการกำหนดแผนงาน แสดงถึงผังเวลาเริ่มต้นและสิ้นสุดของแต่ละกิจกรรม และแสดงถึงกิจกรรมที่วิกฤตและไม่วิกฤตที่ต้องเอาใจใส่เป็นพิเศษ สุดท้ายการควบคุมจะแสดงรายงานความคืบหน้าของโครงการเป็นระยะ ๆ อย่างใกล้ชิด ซึ่งการแสดงสิ่งต่าง ๆ เหล่านี้ จะอยู่ในรูปของตารางพร้อมคำอธิบาย ส่วนตารางทรัพยากรทำการคำนวณหา แรงงานและจำนวนของทรัพยากรที่มีอยู่ เช่น พนักงาน เครื่องมือ เครื่องจักร เป็นต้น โดยนำรายละเอียดต่าง ๆ จาก Gantt Chart หรือ PERT/CPM มาประกอบ เพื่อให้ทราบถึงจำนวนแรงงานและเครื่องมือที่มีอยู่ ว่ามีจำนวนเพียงพอต่องานหรือไม่ ถ้าจำนวนแรงงานและเครื่องมือมีไม่เพียงพอกับงานในระบบควรกำหนดหรือปรับเปลี่ยนส่วนงานระบบใดให้สอดคล้องกับระบบ เพื่อให้การทำงานมีความราบรื่นมากขึ้นและมีขั้นตอนการทำงานดังนี้

งานของขั้นตอน 060

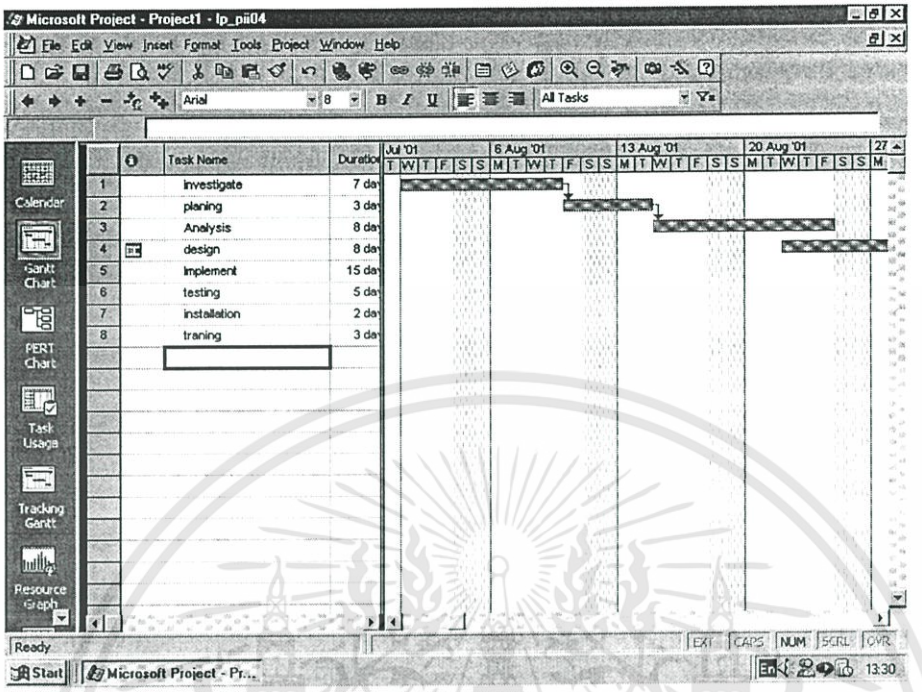
งานที่ 10 จัดทำแนวทางการพัฒนาระบบจากทางเลือกที่ได้รับพิจารณาจากผู้บริหารโดยใช้ PERT/CPM หรือ Gantt Chart พร้อมกับจัดทำแผนสำรองในกรณีที่ไม่สามารถดำเนินงานต่อไปได้

งานที่ 20 สร้างแผนผังลำดับการทำงานให้กับระบบโดยลำดับการทำงานก่อนหลังพร้อมกับแบ่งกลุ่มการทำงานให้กับระบบ (ถ้าระบบนั้นเป็นระบบที่มีขนาดใหญ่)

งานที่ 30 นำ TSO ที่ถูกเลือกมาพิจารณาถึงทรัพยากรที่จำเป็นต้องใช้กับการพัฒนาระบบ มาจัดทำแผนงานการบริหารทรัพยากรให้กับโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์และเอกสาร



รูปที่ 5.11 แสดง Gantt Chart ของระบบ ITEM BANK

7) ขั้นตอน 070 รวบรวมเอกสารของการศึกษาความเหมาะสม (Assemble Feasibility Report)

ขั้นตอนที่ 070 เป็นกิจกรรมสุดท้ายของระยะที่ 0 ที่มีไว้สำหรับคอยตรวจสอบตรวจทานและรวบรวมเอกสารต่าง ๆ ของกิจกรรมทั้งหมดที่เริ่มทำมาตั้งแต่ขั้นตอนที่ 010 ถึงขั้นตอนที่ 060 ซึ่งแต่ละกิจกรรมที่ผ่านมาจะมีเอกสารแสดงความสัมพันธ์ต่าง ๆ มากมายก่อนข้ามไปขั้นตอนถัดไปซึ่งมีเอกสารทั้งหมดมีดังนี้

1. รายละเอียดสภาพแวดล้อมของระบบปัจจุบันแบบคร่าว ๆ
2. รายละเอียดสภาพแวดล้อมที่ต้องการแบบคร่าว ๆ
3. ขอบเขตของปัญหาพร้อมรายละเอียด
3. แผนงานของระบบที่ใช้สำหรับระบบเชิงวัตถุ (Scenario Plan)
4. Requirements Catalogue
5. ตารางของการพัฒนาระบบโดยรวม (Project management)
6. โครงสร้างหน้าจอของระบบ ทั้ง อินพุต-เอาต์พุตและอินเทอร์เฟซ
7. ทางเลือกของโครงการทั้งทางธุรกิจและเทคนิคแบบคร่าว ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารที่มาจากกิจกรรม 010-060 ทั้งหมด เป็นเอกสารที่ขาดไม่ได้ ซึ่งถ้าขาดเอกสารใดไป แสดงว่าการวิเคราะห์นั้นไม่สมบูรณ์ ควรที่จะย้อนกลับไปตรวจสอบขั้นตอนต่าง ๆ เหล่านี้ใหม่อีกครั้ง ซึ่งถ้าเอกสารต่าง ๆ ไม่ครบถ้วนแล้วย่อมทำให้การข้ามไปสู่ขั้นตอนถัดไปย่อมเกิดปัญหาตามมากับระบบได้

5.5.1.2 ระยะที่ 1 ศึกษาระบบเดิม (Stage 1 Investigate of Current System)

ระยะที่ 1 เป็นขั้นตอนที่ดำเนินงานต่อจากระยะที่ 0 โดยนักวิเคราะห์จะเพิ่มรายละเอียดของระบบที่ทำการศึกษาความเหมาะสมไว้ก่อนหน้านี้

1) ขั้นตอน 110 ประเมินการศึกษาความเหมาะและตรวจสอบเอกสาร (Feasibility Evolution and Review Report)

ขั้นตอน 110 เป็นขั้นตอนเริ่มต้นของระยะที่ 1 ซึ่งก่อนที่เราจะเริ่มวิเคราะห์ระบบนั้น ควรที่จะตรวจสอบเอกสารจากขั้นตอนก่อนหน้าก่อนที่จะลงมือทำ สิ่งที่ต้องตรวจสอบในขั้นตอนนี้คือ

1. ตรวจสอบ Requirement Catalogue
2. ตรวจสอบแผนผังการไหลของข้อมูลและ Logical Data Model
3. ตรวจสอบตารางการพัฒนาระบบ
4. ตรวจสอบเอกสารจากเจ้าของระบบ

เหตุที่ต้องปฏิบัติเช่นนี้เพราะการดำเนินงานต่อไปนั้นเอกสารที่เกิดขึ้นก่อนหน้านี้เป็นสิ่งที่บอกถึงความสมบูรณ์ของขั้นตอนแรกก่อนที่จะเข้าสู่การดำเนินงานในขั้นตอนถัดไป ซึ่งถ้าตรวจพบสิ่งที่ผิดปกติหรือขาดหายไปจะได้กลับเข้าไปแก้ไขอีกครั้ง

2) ขั้นตอน 120 สำรวจแผนภูมิการไหลของข้อมูล (Investigation of DFDs)

แผนภูมิการไหลของข้อมูลที่เกิดขึ้นจากระยะที่ 0 นั้น เป็นเพียงการแสดงขอบเขตของระบบงานที่ต้องการพัฒนาเท่านั้น ในขั้นตอนที่ 120 นักวิเคราะห์ต้องนำขอบเขตเหล่านั้น มากระจายไปสู่ระดับที่ย่อยที่สุด บางครั้งก่อนที่จะทำกิจกรรมย่อยใด ๆ ก็ตาม นักวิเคราะห์อาจจำเป็นต้องสร้างขั้นตอนการไหลของข้อมูล หรือ “Document Flow Diagram” ของระบบเดิม เพื่อให้ทราบถึงการทำงานของระบบงานในรูปของเอกสารซึ่งเหมาะกับนักวิเคราะห์ที่มีประสบการณ์น้อย เพราะทำให้ทราบการทำงานของกระบวนการชัดเจนมากขึ้น เนื่องจากการกระจายกระบวนการของแผนภูมิการไหลของข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูล ในกิจกรรมนี้ยังเป็นการทำงานของระบบงานเดิมในระดับกายภาพ ดังนั้นการวิเคราะห์ต้องคำนึงอยู่เสมอว่ากระบวนการที่เกิดขึ้นเป็นสิ่งที่ระบบกำลังสนใจและมีอยู่จริง เพราะถ้ากระบวนการที่เกิดขึ้นไม่ตรงกับความเป็นจริงและความต้องการของผู้ใช้ย่อมทำให้กระบวนการต่าง ๆ ที่ เกิดขึ้นนั้นผิดพลาดไป การกระจายกระบวนการเริ่มจากแผนภูมิการไหลของข้อมูลระดับที่ 1 คือ นักวิเคราะห์นำกระบวนการที่มีอยู่ในระดับ 1 มาพิจารณาทุกตัวและนำมากระจายไปสู่ระดับ 2 3 และ4 ตามลำดับเพื่อให้ได้กระบวนการสุดท้ายของแต่ละตัว หรือเรียกกว่า "Elementary Process" เพราะกระบวนการสุดท้ายเปรียบได้กับการทำงานที่แท้จริงของระบบเพื่อให้ได้ข้อมูลตรงตามความต้องการ เมื่อกระจายกระบวนการที่มีอยู่ในระบบเรียบร้อยแล้ว ถัดมาคือ เอกสารประกอบแผนภูมิการไหลของข้อมูล ได้แก่ รายละเอียดของเอนติตีภายนอกที่มีความสัมพันธ์กับกระบวนการและรายละเอียดของอินพุต-เอาต์พุตที่สัมพันธ์กับเอนติตีภายนอกโดยทำจากกระบวนการสุดท้ายขึ้นมาหรือเรียกว่า "ล่างขึ้นบน" เพราะทำให้ทราบถึงขอบเขตของระบบได้ชัดเจนขึ้น จากนั้นนำเอกสารและกระบวนการทุกตัวที่เกิดขึ้นกับขั้นตอนนี้ทั้งหมดไปปรับปรุงกับความต้องการหรือปัญหาที่เกิดขึ้นให้กับระบบที่มีอยู่เพื่อให้ได้ข้อมูลที่เป็นปัจจุบันและมีขั้นตอนการทำงานดังต่อไปนี้

งานของขั้นตอน 120

งานที่ 10 ถ้าหากสามารถสร้าง Document Flow ให้กับระบบได้ควรทำ เพราะจะทำให้เข้าใจการไหลเวียนของข้อมูลในระดับกายภาพได้ดียิ่งขึ้น (Document Flow ไม่มีในขั้นตอนของการศึกษาความเหมาะสม)

งานที่ 20 ตรวจสอบแผนภูมิการไหลของข้อมูลและกระจายแผนภูมิไปสู่ระดับที่ 1

งานที่ 30 กระจายกระบวนการที่อยู่ในแผนภูมิระดับที่ 1 ไปสู่ระดับที่ 2 และ 3 ตามลำดับ

งานที่ 40 สร้างกระบวนการสุดท้าย (Elementary Process) ที่เกิดจากการกระจายพร้อมกับใส่รายละเอียดของกระบวนการนั้น จากนั้นปรับปรุงรายละเอียดของเอนติตีภายนอกที่เกี่ยวข้องและบันทึกราละเอียดของการติดต่อระหว่างกระบวนการสุดท้ายแต่ละตัวที่มีต่อเอนติตี หรือที่เรียกว่า ตารางอินพุต-เอาต์พุต (I/O Description)

สัญลักษณ์และเอกสาร

สัญลักษณ์ของกระบวนการสุดท้าย (Elementary Process)

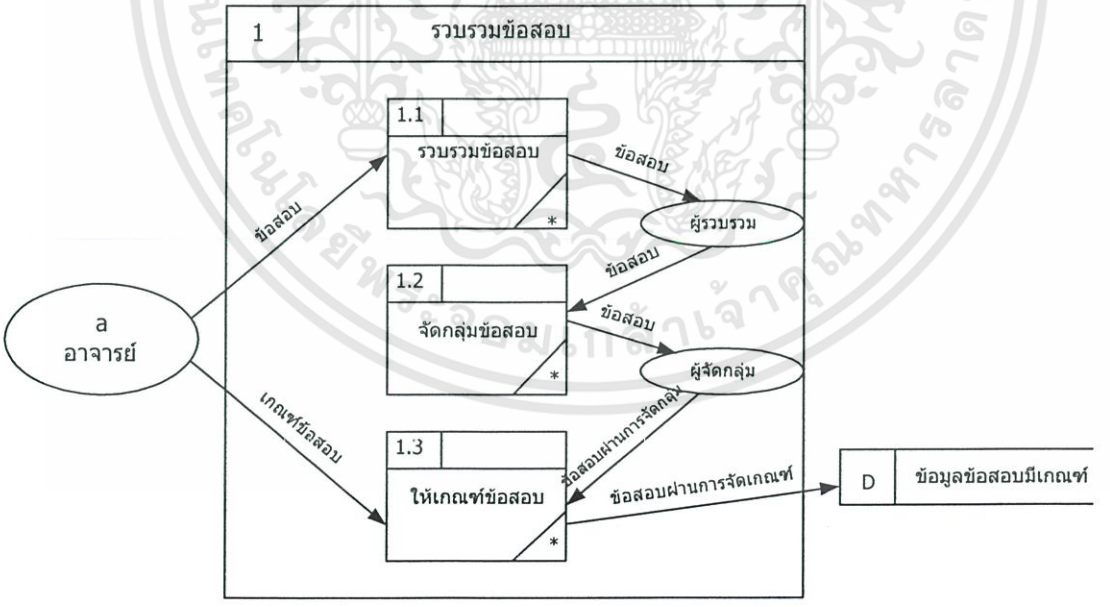
กระบวนการสุดท้าย คือ กระบวนการทำงานของระบบที่ไม่สามารถกระจายได้อีก ซึ่งเกิดจากการกระจายของกระบวนการใหญ่ในระบบ สัญลักษณ์เหมือนกับกระบวนการทั่วไปของโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประยุกต์แต่แตกต่างกันที่ ชื่อของกระบวนการ คือ ไม่ระบุหมายเลขของกระบวนการ มีลำดับชั้นตามกลุ่มของกระบวนการก่อนหน้าที่เกี่ยวข้องและมุมข้างล่างมีเครื่องหมาย * เพื่อแสดงถึงกระบวนการสุดท้ายของกระบวนการนั้น ดังแสดงในรูปที่ 5.12

รูปที่ 5.12 แสดงกระบวนการสุดท้าย (Elementary Process)

กระบวนการสุดท้ายจะอยู่ภายใต้กระบวนการที่สูงกว่าและมีความเกี่ยวข้องกัน ดังแสดงในรูปที่ 5.13 ที่แสดงถึงการกระจายกระบวนการของแผนภูมิการไหลของข้อมูลระดับที่ 1 ออกเป็นกระบวนการย่อย ซึ่งภายในประกอบด้วยกระบวนการย่อย 3 กระบวนการ คือ กระบวนการรวบรวมข้อสอบ กระบวนการจัดกลุ่มข้อสอบและกระบวนการให้เกณฑ์ข้อสอบ สังเกตว่ากระบวนการทั้งหมดจะอยู่ในกระบวนการที่ 1 ซึ่งอยู่ในระดับที่สูงกว่า



รูปที่ 5.13 แสดงการกระจายกระบวนการของแผนภูมิการไหลของข้อมูลระดับที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางรายละเอียดของกระบวนการสุดท้าย (EPD- Elementary Process Desription)

กระบวนการสุดท้าย คือ กระบวนการที่ย่อยที่สุดของการทำงานในกระบวนการหลักของระดับที่สูงกว่า ดังนั้น ผู้วิเคราะห์ควรที่จะมีเอกสารประกอบการอธิบายการทำงานของกระบวนการนั้น โดยรายละเอียดเหล่านั้น ควรจะอธิบายในรูปของตาราง ซึ่งมีรายละเอียดดังนี้

1. เลขที่ของกระบวนการย่อย
2. ชื่อของกระบวนการย่อย
3. รายละเอียดการทำงานของกระบวนการย่อย ซึ่งในรายละเอียดต้องกล่าวถึงวิธีการทำงานของกระบวนการและสิ่งที่กระบวนการย่อยนั้นรับผิดชอบให้ชัดเจน

จากรายละเอียดทั้งหมด สามารถนำมาสร้างตารางได้ดังตารางที่ 5.6

ตารางที่ 5.6 แสดงตารางรายละเอียดของกระบวนการย่อย

Elementary Process Description
Process ID: 1.1
Process Name: เก็บข้อสอบรายข้อ
Description: ผู้ออกข้อสอบให้ข้อมูลข้อสอบรายข้อซึ่งสามารถออกข้อสอบได้โดยอาจารย์หลายคน โดยข้อสอบดังกล่าวประกอบด้วยข้อสอบหลายวิชา ในแต่ละวิชามีได้หลายบท แต่ละบทมีข้อสอบหลายข้อ และแต่ละข้อจะมีคำตอบตัวเลือกได้ 4-5 ตัวเลือก

3) ขั้นตอน 130 สำรวจดาต้าโมเดล (Investigation of Data)

หลังจากสำรวจดาต้าโฟลว์ในส่วนของกระบวนการเสร็จสิ้นแล้ว ย้อนกลับมาพิจารณาถึงตัวดาต้าโมเดลหรืออี อาร์ ไดอะแกรม งานหลักของขั้นตอน 130 คือ การตรวจสอบเอนติตีทุกตัวที่เกิดขึ้นกับระบบ เริ่มตั้งแต่ตรวจสอบความสัมพันธ์ที่เกิดขึ้นระหว่างเอนติตี ซึ่งสิ่งที่กำหนดนั้นมีความถูกต้องหรือผิดกฎเกณฑ์หรือไม่ เช่น ความสัมพันธ์แบบ 1:1, 1:M, M:N เป็นต้น ซึ่งถามกฎของการระบุความสัมพันธ์ที่เกิดขึ้นกับอี อาร์ ไดอะแกรมนั้น ควรจัดให้อยู่ในรูปความสัมพันธ์แบบ 1:1 และ 1:M เพราะถ้ามีความสัมพันธ์แบบ M:N จะทำให้เกิดความซ้ำซ้อนกันของแอตทริบิวต์ระหว่างเอนติตีได้ ถ้าเกิดกรณีนี้ควรแยกเอนติตีออกซึ่งเรียกว่า “Associate Entity” เพื่อลดความสัมพันธ์ที่ซับซ้อนของเอนติตีในระบบ ต่อมานำเอนติตีเหล่านั้นมาพิจารณาเพื่อกำหนดแอตทริบิวต์ แอตทริบิวต์ คือ รายละเอียดของเอนติตีแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละตัว ซึ่งมีอยู่ด้วยกันหลายประเภท เช่น คีย์หลัก (Primary key) คีย์คู่แข่ง (Candidate Key) และคีย์รอง (Foreign Key) เป็นต้น หลังจากที่ปรับปรุงความสัมพันธ์และกำหนดแอตทริบิวต์ให้กับเอนติตีเรียบร้อยแล้ว ควรจัดทำตารางความสัมพันธ์ระหว่างเอนติตีกับกระบวนการขึ้น เพื่ออ้างอิงถึงความสัมพันธ์ที่เกิดขึ้นในระบบ ซึ่งความสัมพันธ์ที่เกิดขึ้นมีอยู่หลายรูปแบบ (อ่านอย่างเดียว บันทึก ลบ หรือทั้งอ่านและเขียน) ตัวอย่างเช่น กระบวนการ A สัมพันธ์กับเอนติตี B ในลักษณะอ่านข้อมูลเพียงอย่างเดียว หรือบันทึกอย่างเดียว เป็นต้น การทำในลักษณะเช่นนี้เรียกว่า "Process Cross-Reference" หรือ "Process-to-Data Matirx" หลังจากทำตารางความสัมพันธ์ที่เกิดขึ้นระหว่างเอนติตีกับกระบวนการเรียบร้อยแล้วควรทำเส้นทางของการเข้าถึงข้อมูลของแต่ละกระบวนการหรือเรียกว่า "Access Paths" เพื่อยืนยันความถูกต้องของตารางความสัมพันธ์ที่ได้สร้างไว้ สุดท้ายควรปรับปรุงอี อาร์ ไดอะแกรมให้เรียบร้อยแล้วย้อนกลับไปบันทึกรายละเอียดความต้องการของผู้ใช้อีกครั้งเพื่อที่จะทำให้เอกสารมีความเป็นปัจจุบันมากที่สุด

งานของขั้นตอน 130

งานที่ 10 ตรวจสอบแผนภูมิของดาต้า (LDS)

งานที่ 20 กำหนดแอตทริบิวต์ให้กับเอนติตี

งานที่ 30 ตรวจสอบ LDM กับกระบวนการสุดท้ายทุกตัวที่อยู่ใน DFM ว่าสามารถสนับสนุนซึ่งกันและกันหรือไม่ (อาจจะอธิบายด้วยตารางแมทริกซ์ที่แสดงระหว่างกระบวนการกับ LDM)

งานที่ 40 บันทึกความต้องการใหม่ ๆ ที่เกิดขึ้นกับดาต้าหรือปัญหาที่เกิดขึ้นกับดาต้าปัจจุบันลงใน Requirement Catalogue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์และเอกสาร

ตารางที่ 5.7 แสดงความสัมพันธ์ระหว่างเอนติตีและกระบวนการ

Process/Entity Matrix
(Required System)

	ผู้ออกข้อสอบ	ผู้ให้เกณฑ์	ข้อสอบ	ประวัติการออก
1.1 เก็บข้อสอบรายข้อ	R			
1.2 สร้างเกณฑ์ข้อสอบ		R		
1.3 รวบรวมข้อสอบและเกณฑ์ข้อสอบ			C	
2.1 รับคำขอ			R	A
2.2 คัดเลือกข้อสอบ			R	
3.1 ให้ข้อมูลข้อสอบ			R	
3.2 คัดเลือกข้อสอบที่ต้องการลบ			R	
3.3 ลบข้อสอบ			D	D

จากตารางที่ 5.7 แสดงความสัมพันธ์ระหว่างกระบวนการกับเอนติตี โดยระบุความสัมพันธ์ที่เกิดขึ้น ของกระบวนการกับเอนติตี โดยความสัมพันธ์ระหว่างกันมีอยู่ 4 ประเภท คือ สร้าง (Create) เรียกดูข้อมูล (Request) ลบข้อมูล (Delete) จัดเก็บข้อมูล (Archive)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.8 แสดงรายละเอียดของเอนติตี

Entity Name: ผู้ออกข้อสอบ				
Description: ทำการออกข้อสอบให้กับระบบ				
Attribute			**Primary Key**	**Foreign Key**
1. รหัสผู้ออกข้อสอบ 2. ชื่อผู้ออกข้อสอบ 3. นามสกุลผู้ออกข้อสอบ 4. คำอธิบาย			Yes	
Must/may be	**Either/or**	**Link Phrase**	**One and only one/ one or more**	**Object Entity Name**
May be		**ออก**	**One or more**	**ข้อสอบ**
Entity Volume: Max.__100___. Min ___1_____. Average. _____50____.				
User Role			**Access**	
ผู้รวบรวมข้อสอบ			**Create, Read, Modify, Delete**	
Growth Rate:				
Archiving: เก็บอยู่ในระบบจนกว่าจะลบออกหรือเลิกใช้ระบบ				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.9 แสดงรายละเอียดแอตทริบิวต์ของแต่ละเอนติตี

Name: เฉลย		
Description: เฉลยข้อสอบของโจทย์แต่ละข้อ		
Cross Reference	**Type**	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Numeric	**Length:** 1	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation:		
User Role	**Access**	

4) ขั้นตอน 140 เปลี่ยนไดอะแกรมระดับกายภาพไปสู่ระดับตรรกะของระบบปัจจุบัน (Convert Physical to Logical view of Current System)

จากขั้นตอน 120 และ 130 นักวิเคราะห์ได้แผนภูมิระดับกายภายของระบบปัจจุบันพอมาถึงขั้นตอน 140 นี้นำแผนภูมิเหล่านั้นมาเปลี่ยนแปลงระดับกายภาพไปสู่ระดับตรรกะ การเปลี่ยนควรเริ่มจากอี อาร์ ไดอะแกรมก่อน ดาต้าสโตร์ที่แสดงในขั้นตอนที่ 130 เรียกว่า "LDM" นั้นควรมีความสัมพันธ์กับแผนภูมิการไหลของข้อมูล แต่ความสัมพันธ์ที่เกิดขึ้นระหว่างระดับกายภาพกับระดับตรรกะนั้นมีความแตกต่างกัน กล่าวคือ อี อาร์ ไดอะแกรมแสดงในรูปของความสัมพันธ์ของเอนติตี แต่แผนภูมิการไหลของข้อมูลแสดงความสัมพันธ์ในรูปแบบของดาต้าสโตร์กับกระบวนการ ดังนั้นผู้วิเคราะห์ควรจัดกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของเอนติตีที่เกี่ยวข้องกันไว้ในดาต้าสโตร์เดียวกัน เพราะเอนติตีที่แสดงความสัมพันธ์ในแผนภูมิไม่สามารถนำไปอ้างอิงในส่วนของแผนภูมิการไหลของข้อมูลได้ การจัดกลุ่มเอนติตีควรจัดให้อยู่ในรูปของดาต้าสโตร์ เพื่อสามารถนำไปอ้างอิงระหว่างไดอะแกรมได้ ขั้นตอน 120 มีขั้นตอนการทำงานดังต่อไปนี้

งานของขั้นตอน 140

งานที่ 10 ย้ายแผนภูมิการไหลของข้อมูลระดับกายภาพไปสู่ระดับตรรกะ โดยเริ่มจากระดับล่างของแผนภูมิ ซึ่งจะพิจารณาในแต่ละกระบวนการที่เกิดขึ้นในขั้นตอน 120 ก่อนการย้ายนั้น ควรตรวจสอบแผนภูมิให้รอบครอบก่อนซึ่งอาจมีบางช่วงหรือบางอย่างขาดหายไปในขั้นตอนการวิเคราะห์ ถ้าบางกระบวนการหายไปควรนำกลับไปวิเคราะห์อีกครั้ง ถ้าไม่มีข้อผิดพลาดควรทำการเริ่มย้ายแผนภูมิ โดยเริ่มจากกระบวนการสุดท้ายที่อยู่ระดับล่างของแผนภูมิก่อนเป็นอันดับแรก การย้ายนั้นควรขึ้นกับประสบการณ์และวิธีการจัดการของนักวิเคราะห์ระบบเป็นหลัก

งานที่ 20 ตรวจสอบอี อาร์ ไดอะแกรม โดยทำการจับกลุ่มของเอนติตีที่เกี่ยวข้องไว้ที่เดียวกันเพื่อนำไปสร้าง LDS ในระดับตรรกะให้กับระบบ ซึ่งการจับกลุ่มนี้ในแต่ละแฟ้มข้อมูลในทางตรรกะอาจมีมากกว่าหนึ่งเอนติตีได้

งานที่ 30 ตรวจสอบกระบวนการสุดท้ายที่อยู่ในระดับล่างของแผนภูมิการไหลของข้อมูล โดยพิจารณาแต่ละกระบวนการ กล่าวคือ กระบวนการใดที่มีลักษณะการทำงานที่คล้ายกันควรยุบรวมกันหรือกระบวนการใดที่เดิมมีการติดต่อกันผ่านฐานข้อมูลซึ่งไม่มีความจำเป็นที่ต้องใช้ฐานข้อมูลนั้นควรยุบการเชื่อมต่อเปลี่ยนมาทำการเชื่อมต่อโดยตรงแทน

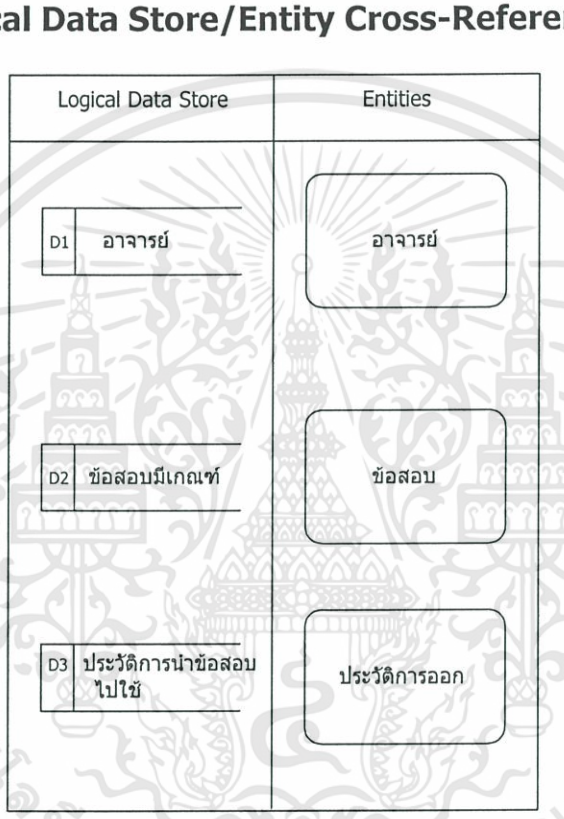
งานที่ 40 วาดแผนภูมิระดับตรรกะที่ย้ายจากระดับกายภาพ พร้อมกับตรวจสอบแผนภูมิการไหลของข้อมูลและแผนภูมิของดาต้าที่ย้ายให้ถูกต้อง

งานที่ 50 บันทึกสิ่งที่เปลี่ยนแปลงที่เกิดขึ้นจากการย้ายแผนภูมิลงใน Requirement Catalogue เพื่อเตรียมย้ายไปสู่การวิเคราะห์ให้กับระบบใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์และเอกสาร

จากรูปที่ 5.14 แสดงตารางความสัมพันธ์ของดาต้าสโตร์ทางตรรกะกับเอนติตี ซึ่งเรียกว่า "Logical Data Store/Entity Cross-Reference" ซึ่งแบ่งออกเป็น 2 คอลัมน์ ทางซ้ายแสดงดาต้าสโตร์แบบตรรกะ ส่วนทางด้านขวาแสดงความสัมพันธ์ที่เกิดขึ้นในแต่ละเอนติตี ซึ่งทั้ง 2 คอลัมน์ต้องสัมพันธ์กัน บางครั้งเอนติตีหนึ่งอาจมีดาต้าสโตร์ได้มากกว่าหนึ่งดาต้าสโตร์ได้



รูปที่ 5.14 แสดงความสัมพันธ์ระหว่างดาต้าสโตร์ทางตรรกะกับเอนติตี

5) ขั้นตอน 150 รวบรวมเอกสารของการศึกษาระบบเดิม (Assemble Document)

ขั้นตอน 150 เป็นขั้นตอนสุดท้ายของระยะที่ 1 ซึ่งเป็นการรวบรวมเอกสารที่เกิดขึ้นของขั้นตอน โดยเอกสารที่ปรากฏในขั้นตอนนี้ มีดังนี้

- Context Diagram
- Logical Data Flow Model
- Current Environment LDM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Logical Data Store/Entity Cross-Reference (Process-to-Data Matrix)
- Requirements Catalogue
- User Catalogue

โดยเอกสารทั้งหมดที่กล่าวมาข้างต้นหากขาดหรือไม่ครบนักวิเคราะห์ควรกลับไปขั้นตอนต่าง ๆ ของระยะที่ 1 โดยเริ่มตั้งแต่ขั้นตอนที่ 110 ใหม่อีกครั้งเพื่อสร้างสิ่งที่ขาดหายไปก่อนที่จะข้ามขั้นตอนนี้ไป

5.5.1.3 ระยะที่ 2 การกำหนดสภาพแวดล้อมของระบบ (Stage 2 Select Environment of System)

ระยะที่ 2 คือ ระยะของการนำเสนอทางเลือกทางธุรกิจและเทคนิคให้กับผู้ใช้ระบบ ซึ่งผู้วิเคราะห์ต้องนำเสนอทั้งงบประมาณค่าใช้จ่ายและเทคโนโลยีที่สามารถสนับสนุนตามความต้องการ โดยยึดตามหลักความต้องการของผู้ใช้ระบบ ซึ่งเสนอในรูปแบบของรายงาน ภายในขั้นตอนนี้แบ่งกิจกรรมออกเป็น 3 กิจกรรม ดังต่อไปนี้

1) ขั้นตอน 210 ตรวจสอบทางเลือกทางธุรกิจและทางเทคนิคจากขั้นตอนที่ 050 (Review BSO and TSO from Step 050)

จากขั้นตอน 050 ทางเลือกที่เสนอไว้เป็นทางเลือกทางด้านธุรกิจและเทคนิคแบบคร่าว ๆ จากทางเลือกที่มีอยู่ในขั้นตอน 050 นำไปเสนอกับเจ้าของโครงงานโดยกรองทางเลือกที่อยู่ให้เหลือประมาณ 2-3 ทางเลือกเพื่อคัดสรรทางเลือกที่มีความเป็นไปได้ให้ตรงกับความต้องการของผู้มากที่สุด หลังจากนั้นนำทางเลือกที่ได้คัดเลือกแล้วมาเพิ่มเติมรายละเอียด ซึ่งรายละเอียดของทางเลือกทางธุรกิจกล่าวถึงความชัดเจนของแต่ละทางเลือก เช่น ค่าใช้จ่ายของการพัฒนาระบบ งบประมาณ และอื่น ๆ เป็นต้น ส่วนทางเลือก ทางเทคนิค กล่าวถึง เทคโนโลยีที่จะมาสนับสนุนหรือนำมาแก้ไขปัญหาที่เกิดขึ้นกับระบบให้ตรงกับความต้องการ โดยอาจต้องอาศัยช่างเทคนิคเข้ามาให้รายละเอียดในจุดนี้ เช่น รายละเอียดของฮาร์ดแวร์ ซอฟต์แวร์ สภาพแวดล้อมของระบบ ระบบปฏิบัติการและเครื่องมือต่าง ๆ เป็นต้น หลังจากที่เพิ่มรายละเอียดต่าง ๆ ให้กับทางเลือกทั้งสองเรียบร้อยแล้ว ควรนำทางเลือกทั้งสองมาจัดทำตารางความสัมพันธ์ระหว่างเทคโนโลยีที่นำมาใช้กับความเป็นไปได้ที่เกิดขึ้นกับระบบงาน กล่าวคือ เปรียบเทียบถึงความสอดคล้องของการเลือกแนวทางทั้งสองว่ามีความสอดคล้องกันหรือไม่ เช่น ถ้าเลือกทางเลือกทางธุรกิจที่ 1 กับ ทางเลือกทางเทคนิคที่ 2 มีความสอดคล้องกัน 50% กับถ้าเลือกทางเลือกทางธุรกิจที่ 2 กับทางเลือกทางเทคนิคที่ 3 มีความสอดคล้องกัน 80% เป็นต้น ซึ่งการทำเช่นนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำให้ผู้วิเคราะห์และเจ้าของระบบสามารถทราบถึงผลกระทบที่เกิดขึ้นกับระบบหรือที่เรียกว่า "Impact Analysis" ได้ชัดเจนขึ้น และทำให้สามารถกำหนดทิศทางในการพัฒนาระบบต่อไปได้อีกด้วย

งานของขั้นตอน 210

งานที่ 10 กำหนด BSO และ TSO จากความต้องการของผู้ใช้ให้กับระบบใหม่

งานที่ 2 สร้างทางเลือกให้ BSO และ TSO อย่างน้อย 6 ทางเลือก

งานที่ 30 นำ BSO และ TSO ไปนำเสนอให้กับผู้ใช้ พร้อมกับแสดงความสัมพันธ์ของทางเลือกโดยเชื่อมโยงกันระหว่าง BSO TSO และ Feasibility Option ให้ผู้ใช้ทราบ

งานที่ 40 เพิ่มรายละเอียดของทางเลือกที่ได้เลือกไว้ พร้อมกับแสดงความสัมพันธ์ของ BSO ที่เกิดขึ้นกับ DFM และ LDM และแสดงสภาพแวดล้อมททางเทคนิคที่จะเกิดขึ้นกับระบบสำหรับการเลือกทางเลือกนั้น ๆ

งานที่ 50 นำทางเลือกที่มาพิจารณาถึงขีดความสามารถของระบบกับทางเลือกว่าสามารถรองรับกับความต้องการของระบบได้หรือไม่

งานที่ 60 นำเสนอทางเลือกที่ได้เลือกแล้ว มาจัดทำรายละเอียดที่สมบูรณ์ โดยอาศัยสภาวะแวดล้อมทางธุรกิจและทางเทคนิค

2) ขั้นตอน 220 เลือกทางเลือกทางธุรกิจและทางเทคนิค (Select BSO and TSO)

หลังจากที่เพิ่มรายละเอียดของทางเลือกทั้งสองในขั้นตอน 210 เรียบร้อยแล้ว จากนั้นนักวิเคราะห์และเจ้าของระบบควรนำทางเลือกที่ถูกเลือกมาคิดคำนวณอัตราผลตอบแทนที่เกิดขึ้นกับทุก ๆ ทางเลือก (ซึ่งในทางเลือกทางธุรกิจที่เกิดขึ้นจะเป็นเพียงแค่การกำหนดค่าใช้จ่ายที่เกิดขึ้น ไม่ได้คำนวณผลตอบแทนของระบบใด ๆ ทั้งสิ้น) การคิดอัตราผลตอบแทนทางธุรกิจนั้นควรนำทั้ง 2 ทางเลือกมาคำนวณ กล่าวคือ นำทางเลือกทางธุรกิจกับทางเลือกทางเทคนิคที่ได้ทำการจับคู่หรือเห็นว่าเหมาะสมมาคำนวน ซึ่งอาจจะมีมากกว่า 1 คู่ ในการคำนวณแต่ละกรณี ค่าใช้จ่ายที่ควรกล่าวถึง เช่น ค่าแรง ค่าใช้จ่ายในการพัฒนา เช่น ค่าแผ่นดิสก์ กระดาษ หมึกพิมพ์ และอื่น ๆ เป็นต้น ค่าใช้จ่ายในการอบรมพนักงาน (ทั้งในกรณีที่ต้องการพัฒนาเองหรือจ้างจากแหล่งอื่น) ค่าลิขสิทธิ์ซอฟต์แวร์ ค่าเสียเวลาในการทดสอบระบบ และอื่น ๆ ซึ่งทั้งหมดควรแสดงในรูปของตัวเลข ถ้าสังเกตพบว่าค่าใช้จ่ายที่กล่าวไปนั้นมีทั้งค่าใช้จ่ายที่สามารถจับต้องได้และจับต้องไม่ได้ ควรแยกออกจากกันให้ชัดเจนในรูปแบบของรายงาน หลังจากได้ตัวเลขของค่าใช้จ่ายทั้งหมดแล้วควรให้นักวิเคราะห์ทางการเงินนำตัวเลขของแต่ละทางเลือกมาคำนวณอัตราผลตอบแทนทางการเงินเพื่อให้ทราบถึงผลกระทบของการลงทุนกับการพัฒนาระบบ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวณหางวดคืนทุน (Payback Analysis) มูลค่าปัจจุบันสุทธิ (Net Present Value) และอัตราผลตอบแทนการลงทุน (Return of Investment) เป็นต้น เพื่อนำเสนอเป็นผลสรุปของทางเลือกทั้งหมดที่เกิดขึ้นกับระบบให้กับผู้บริหารตัดสินใจเลือกทางเลือกพร้อมกับจัดสรรทรัพยากรต่าง ๆ ที่จะเกิดขึ้นตลอดการพัฒนาระบบต่อไป

งานของขั้นตอน 220

งานที่ 10 นำ BSO และ TSO ที่เลือกแล้วมาคำนวณหาต้นทุน ได้แก่ ต้นทุนการพัฒนาและต้นทุนสำหรับการดำเนินงาน โดยต้นทุนการพัฒนา คำนวณจาก กำลังคน วัสดุอุปกรณ์ที่เกี่ยวข้องกับการพัฒนา (แผ่นดิสก์ กระดาษพิมพ์ หมึก และอื่น ๆ) ค่าอบรมและอื่น ๆ ส่วนต้นทุนสำหรับการดำเนินงาน เช่น ค่าลิขสิทธิ์ ซอฟต์แวร์ที่ใช้ในการพัฒนา เงินเดือนพนักงาน เวลาการดำเนินงาน และค่าโสหุ้ยต่าง ๆ เป็นต้น โดยแสดงออกมาในรูปแบบของรายงาน

งานที่ 20 นำ BSO และ TSO ที่เลือกแล้วมาคำนวณหาผลตอบแทน ซึ่งการคำนวณนั้นจะต้องคำนวณทั้งผลตอบแทนที่เห็นได้และที่ไม่สามาถเห็นได้อย่างชัดเจน พร้อมกับแสดงรายละเอียดต่าง ๆ ในรูปแบบรายงาน

งานที่ 30 คำนวณหาผลกระทบของโครงการ โดยอาศัยเทคนิคการคำนวณทางการเงิน เช่น คำนวณหางวดเวลาคืนทุน (Payback Analysis) มูลค่าปัจจุบันสุทธิ (Net Present Value) อัตราผลตอบแทนจากการลงทุน (Return On Investment)

งานที่ 40 สรุปผลการคำนวณหาต้นทุนและผลตอบแทนของ BSO และ TSO

3) ขั้นตอน 230 สรุปทางเลือกทางธุรกิจและทางเทคนิค (Summary BSO and TSO)

หลังจากคำนวณและเพิ่มเติมรายละเอียดต่าง ๆ ให้กับทางเลือกทั้ง 2 พร้อมกับเลือกทางเลือกเรียบร้อยแล้วในขั้นตอน 220 ดังนั้นกิจกรรมนี้จะทำการตรวจสอบเอกสารที่เกิดขึ้นในกิจกรรมที่ผ่านมา โดยเอกสารที่ได้ก่อนออกจากขั้นตอนที่ 2 นี้ คือ รายงานทางเลือกทางธุรกิจ รายงานทางเลือกทางเทคนิค รายงานการวิเคราะห์การลงทุน (Cost-Benefit Analysis Report) รายงานผลกระทบในการลงทุน (Cost-Effective Report) รายงานที่เกิดขึ้นของขั้นตอนนี้ต้องแสดงทั้งหมดไม่แสดงเฉพาะทางเลือกที่ถูกเลือกแล้วเท่านั้น กล่าวคือ มีทางเลือกทางธุรกิจและทางเทคนิคกี่ทางเลือกควรแสดงทั้งหมด เพื่อนำไปเป็นเอกสารอ้างอิงในการพิจารณาในระยะที่ 3 ต่อไป การสรุปทางเลือกทางธุรกิจและทางเทคนิคนั้นควรทำในระดับของผู้บริหารโครงการกับเจ้าของโครงการ ซึ่งการเลือกนี้จะถูกทำซ้ำต่อเมื่อผู้บริหารไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารตัดสินใจได้ กล่าวคือ ถ้าตัดสนใจเลือกทางเลือกทั้งสองไม่ได้อาจมาจากเอกสารไม่ครบหรือไม่มีความเหมาะสม ดังนั้นการเลือกทางเลือกก็ไม่สมบูรณ์ ทางแก้ไขอาจแก้ไขโดยย้อนกลับไปสู่ขั้นตอน 210 ของระยะที่ 2 อีกครั้งเพื่อให้การเลือกทางเลือกนี้สมบูรณ์ขึ้น

5.5.2 โมดูลการวิเคราะห์ (Analysis Module)

โมดูลการวิเคราะห์ เป็น โมดูลที่กล่าวถึงนำความต้องการของผู้ใช้ เข้ามาประยุกต์ในระบบงานที่รับผิดชอบ โดยสิ่งที่นักวิเคราะห์นำมาเพิ่มเติมนั้นเป็นสิ่งที่เกิดขึ้นมาจากขั้นตอนก่อนหน้า ในโมดูลการวิเคราะห์สามารถแบ่งออกเป็น 2 ขั้นตอนหลัก คือ ขั้นตอน Requirements Specification กับขั้นตอนการย้ายการวิเคราะห์ไปสู่วิธีการเชิงวัตถุ ก่อนที่จะย้ายไปสู่ขั้นตอนของการย้ายไปสู่รูปแบบเชิงวัตถุได้นั้น นักวิเคราะห์จำเป็นต้องทำการปรับปรุงไดอะแกรมให้ตรงกับความต้องการาของผู้ใช้เสียก่อน ส่วนขั้นตอนการย้ายนั้น ไดอะแกรมที่เกิดขึ้นในขั้นตอนที่ 3 บางส่วนจะถูกนำมาปรับปรุงให้อยู่ในรูปของวิธีเชิงวัตถุ เช่น LDS ไดอะแกรม จะถูกเปลี่ยนไปสู่ O/C ไดอะแกรม แอตทริบิวต์ที่อยู่เอกสารต่าง ๆ ที่ต้องนำมาใส่ในวัตถุหรือคลาส และฟังก์ชันที่ปรากฏในคาต้าโฟลว์ไดอะแกรมจะไปอยู่ในส่วนของเมธอดของวัตถุหรือคลาส เป็นต้น ซึ่งทำให้วิธีการพัฒนาระบบไปสู่วิธีเชิงวัตถุมากขึ้น นอกจากนี้ไดอะแกรมต่าง ๆ ที่เกิดขึ้นในระยะที่ 3 ที่ไม่ได้กล่าวถึง เช่น ELH (Entity Life History) ECD (Effect Corespondent Diagram) เป็นต้น ไดอะแกรมเหล่านี้เป็นส่วนช่วยเหลือในการพิจารณาถึงสิ่งที่เกิดขึ้นในแต่ละวัตถุหรือคลาสนั้น รายะเอียดของแต่ละขั้นตอนมีงานดังนี้

5.5.2.1 ระยะที่ 3 ระบุความต้องการ (Requirements Specification)

1) ขั้นตอน 310 ตรวจสอบเอกสารความต้องการของผู้ใช้ (Review Requirement Catalogue)

งานของขั้นตอน 310

งานที่ 10 ปรับปรุงและตรวจสอบ Requirements Catalogue กับทาเลือกทางธุรกิจและเทคนิคของระบบอย่างละเอียด ก่อนนำไปปรับปรุงแผนภูมิทั้งหมดให้เป็นไปตามความต้องการของผู้ใช้

งานที่ 20 พิจารณาจุดประสงค์ของการพัฒนาระบบ โดยอาศัย Requirements Catalogue และ BSO ที่เกิดจากระยะ 1 และ 2 เพื่อนำมาเป็นแนวทางของการพัฒนาระบบ เพราะการพัฒนาระบบอาจไม่จำเป็นต้องสร้างใหม่ทั้งหมด แนวทางการพัฒนาระบบ สามารถแบ่งการพัฒนาออกเป็น 3 ประเภทคือ พัฒนาเพียงบางส่วนของระบบ (Bussiness

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

process improvement – BPI) โดยปรับปรุงเฉพาะส่วนที่ยังไม่มีหรือไม่สามารถรองรับความต้องการเท่านั้น ประเภทที่สองคือ พัฒนาระบบใหม่หมดโดยเปลี่ยนแปลงเป็นระบบคอมพิวเตอร์ทั้งหมด (Business Process Automation - BPA) สุดท้ายพัฒนาระบบใหม่โดยทำการปรับปรุงระบบเดิมให้มีขีดความสามารถดีขึ้นกว่าเดิม (Business Process Reengineering - BPR) ซึ่งการปรับปรุงประเภทนี้จะเน้นไปทางด้าน ปรับปรุงประสิทธิภาพของระบบเดิม เช่น ปรับปรุงเรื่องเวลา ความเร็ว ต้นทุน ความสามารถ เป็นต้น [3]

งานที่ 30 เลือกรูปแบบการพัฒนาระบบ พร้อมกับบันทึกรายละเอียดให้กับ Project Report

2) ขั้นตอน 320 วิเคราะห์ระบบเครือข่าย <ถ้าต้องการ> (Network Analysis, if need)

งานของขั้นตอน 320

งานที่ 10 ตรวจสอบระบบเครือข่ายที่องค์กรมีอยู่ เพื่อให้ทราบถึงประสิทธิภาพของระบบที่มีอยู่เพียงพอหรือตรงต่อความต้องการของโครงการหรือไม่

งานที่ 20 สร้างแผนผังของระบบเครือข่ายโดยใช้ LCD (Location Connectivity Diagram) เพื่อจัดกลุ่มของกระบวนการ ดาต้า ที่มีลักษณะคล้ายกันไว้ที่เดียวกันเพราะจะทำให้ง่ายต่อการจัดการและวางแผนในการออกแบบเครือข่าย

งานที่ 30 จัดทำรายละเอียดของความสัมพันธ์ ที่เกิดขึ้นระหว่างดาต้ากับกระบวนการ ด้าต้ากับตำแหน่งที่ใช้ในระบบเครือข่าย และ กระบวนการกับตำแหน่งที่กระบวนการนั้น ๆ ไปกระทำกับระบบเครือข่าย ซึ่งการจัดทำรายละเอียดนี้ อาจจะจัดทำในรูปของตารางความสัมพันธ์ (Matrix)

งานที่ 40 คำนวณหาปริมาณงานของผู้ใช้แต่ละคน ณ เวลาหนึ่ง ที่ต้องใช้ระบบผ่านเครือข่าย วิธีการคำนวณเริ่มจาก ระบุจำนวนไคลเอ็นต์ที่มีอยู่ในระบบ ระบุจำนวนทรานส์แอ็กชันของผู้ใช้แต่ละคน ซึ่งการคำนวณทั้งหมดนี้จะนำไปกำหนดแบนด์วิดธ์ให้กับระบบเน็ตเวริก์และทราบถึงเวลาตอบสนองของแต่ละทรานส์แอ็กชัน

งานที่ 50 เลือกเทคโนโลยีของระบบเครือข่ายที่สามารถตอบสนองกับระบบได้ โดยคำนึงถึงปัจจัยต่าง ๆ เช่น รูปแบบของเครือข่าย ความเร็ว ความเข้ากันได้กับเครือข่ายระบบอื่น เป็นต้น

งานที่ 60 สรุปการวิเคราะห์ของระบบเครือข่ายที่มีต่อระบบงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ขั้นตอน 330 ปรับปรุง DFM และ LDM ตามความต้องการของผู้ใช้ (Develop Require DFM and LDM)

งานของขั้นตอน 330

งานที่ 10 ปรับปรุงแผนภูมิการไหลของข้อมูลตั้งแต่ระดับแรกจนถึงระดับสุดท้าย เพื่อให้สอดคล้องกับทางเลือกทางธุรกิจและเทคนิค โดยเลือกทำเฉพาะกระบวนการที่เกี่ยวข้องเท่านั้น พร้อมกับบันทึกความเปลี่ยนแปลงลง Requirements Catalogue

งานที่ 20 บันทึกอินพุต-เอาต์พุตของแต่ละกระบวนการที่เกิดกับเอนติตี (I/O Descriptions)

งานที่ 30 ตรวจสอบกระบวนการทุก ๆ ตัวที่อยู่ใน DFM ให้สอดคล้องกับ LDM ที่มีอยู่อย่างละเอียด โดยนำ LDS กับ เอนติตีมาประกอบการพิจารณา เพื่อป้องกันไม่ให้กระบวนการใด ๆ ที่จำเป็นจะต้องติดต่อกับ LDS เกิดความบกพร่อง

งานที่ 40 สร้าง User Role ให้กับ DFM กับเอนติตีภายนอก เพื่อไว้สำหรับตรวจสอบความสัมพันธ์ระหว่างกัน

งานที่ 50 ปรับปรุง LDM ให้สอดคล้องกับความต้องการของผู้ใช้ที่ต้องการให้เกิดขึ้นกับระบบใหม่ ซึ่งการปรับปรุงควรทำเฉพาะLDM ที่เกี่ยวข้องเท่านั้น บันทึกการเปลี่ยนแปลงที่เกิดขึ้นกับแอตทริบิวต์ของแต่ละเอนติตีและปรับปรุง Requirements Catalogue

งานที่ 60 ตรวจสอบ LDM จากเอกสาร โดยจะต้องระบุได้ว่า LDM ทุกตัวสนับสนุนในทุก ๆ กระบวนการสุดท้ายของ DFM ของระบบใหม่

งานที่ 70 ปรับปรุง LDM กับฟังก์ชันสาธารณะให้เหมาะสมกับ Requirements Catalogue

งานที่ 80 ตรวจสอบ LDM และ DFM ที่ได้ทำการปรับปรุงตามความต้องการของระบบใหม่อย่างละเอียดอีกครั้ง พร้อมกับตรวจสอบเอกสารที่เกี่ยวข้อง

4) ขั้นตอน 340 Derive System Function

งานของขั้นตอน 340

งานที่ 10 กำหนดฟังก์ชันให้กับ DFM ที่ถูกปรับปรุงเรียบร้อยแล้ว พร้อมกับกำหนดเหตุการณ์และสิ่งที่จะเกิดขึ้นตามมาภายหลังของแต่และฟังก์ชัน การจับกลุ่มควรนำฟังก์ชันที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัมพันธ์กันมารวมเป็นกลุ่มเดียวกัน ฟังก์ชันที่ไม่สัมพันธ์กับใครควรกำหนดเป็นฟังก์ชันเดียวแทน

งานที่ 20 กำหนดฟังก์ชันเพิ่มเติมโดยพิจารณาจาก Requirements Catalogue กับ DFM ในกรณีที่ความต้องการหลักของระบบไม่มี

งานที่ 30 สร้างโครงสร้างอินพุต-เอาต์พุตของแต่ละฟังก์ชัน

งานที่ 40 สร้างเมทริกซ์ระหว่างฟังก์ชันกับ User Role ไว้สำหรับตรวจสอบการทำงานระหว่างผู้ใช้ระบบกับฟังก์ชันที่เกิดขึ้น เพื่อให้ทราบถึงความสัมพันธ์ระหว่างกัน

งานที่ 50 ระบุระดับความสำคัญของแต่ละฟังก์ชัน

สัญลักษณ์และเอกสาร

ตารางที่ 5.10 แสดงรายละเอียดของฟังก์ชันเก็บตัวข้อสอบ

Function Definition	
Function Name: เก็บตัวข้อสอบ	**Function Id:** 01
Function Type: Update/On-Line/User	
Function Description: เป็นฟังก์ชันที่ทำหน้าที่เก็บข้อมูลตัวข้อสอบโดยลำดับแรกของการเก็บตัวข้อสอบคือระบบจะกำหนดรหัสข้อสอบให้เพื่อให้ผู้ใช้ป้อนข้อมูลวิชาและบทที่ของข้อสอบเพื่อให้ระบบกำหนดหมายเลขข้อของข้อสอบ จากนั้นผู้ใช้จึงจะใส่ข้อมูลข้อสอบส่วนที่เหลือให้ครบได้	
Error Handling:	
DFD Processes: 1.1 (เก็บข้อสอบรายข้อ)	
Events: บันทึกข้อสอบ	**Event Frequency:** 1
I/O Structures:	
I/O Description: a-1.1	
Requirements Catalogue Ref.: 01	
Related Function: None	
Enquires: None	**Enquiry Frequency:**
Common Processing: None	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.11 แสดงความสัมพันธ์ระหว่างฟังก์ชันกับการทำงานของผู้ใช้

USER ROLE/FUNCTION MATRIX

Function / User role	เก็บตัวข้อสอบ	เก็บเกณฑ์และรวบรวม	จัดทำชุดข้อสอบ	ลบข้อสอบที่ไม่ต้องการ
ผู้รวบรวมข้อสอบ	X	X		
ผู้คัดเลือกข้อสอบ			X	
ผู้ลบข้อสอบ				X

5) ขั้นตอน 350 ทำบรรทัดฐานให้กับดาต้าโมเดล (Enhance Required Data Model)

การทำบรรทัดฐานให้กับดาต้าโมเดล คือ การลดความซ้ำซ้อนของแอตทริบิวต์ของเอนติตีที่เกิดขึ้นกับระบบซึ่งการบรรทัดฐานฐานข้อมูลให้กับระบบของแบบจำลองประยุกต์นั้น (ซึ่งแบบจำลองประยุกต์เป็นแบบจำลองที่สนับสนุนทั้งแบบเชิงโครงสร้างและเชิงวัตถุ) ควรทำอย่างน้อยบรรทัดฐานระดับที่ 3 กล่าวคือ ถ้าเป็นการพัฒนาระบบแบบเชิงวัตถุการจัดการแอตทริบิวต์ของแต่ละเอนติตีนั้นจะต้องดูแล Function Dependency (FD) ของแต่เละเอนติตี แต่ถ้าระบบต้องการพัฒนาแบบเชิงโครงสร้างจะย้อนกลับมาจัดการทำบรรทัดฐานฐานข้อมูลที่เหลือให้สมบูรณ์ตามหลักการทำบรรทัดฐานฐานของมูลและถ้าเป็นการพัฒนาแบบเชิงวัตถุก็ไม่จำเป็นต้องทำบรรทัดฐานฐานข้อมูลในส่วนที่เหลือ ดังนั้นงานของขั้นตอนที่ 350 มีลำดับการทำงานดังนี้

งานของขั้นตอน 350

งานที่ 10 เปลี่ยน LDM ไปสู่รูปแบบของตารางเพื่อเตรียมตัวทำบรรทัดฐานฐานข้อมูล (Normalization)

งานที่ 20 ทำบรรทัดฐานฐานข้อมูลของตารางไปสู่ระดับ 3

งานที่ 30 นำตารางที่ทำบรรทัดฐานฐานข้อมูลแล้ว กลับสู่แผนภูมิของข้อมูล

งานที่ 40 เปรียบเทียบ LDM เดิมกับ LDM ใหม่ที่ได้หลังจากการทำบรรทัดฐานฐานข้อมูล ถึงสิ่งที่เปลี่ยนแปลงไป เพื่อตรวจสอบความถูกต้องให้ตรงความต้องการของผู้ใช้อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) ขั้นตอน 360 สร้างแบบจำลอง (Create Prototype)

งานของขั้นตอน 360

งานที่ 10 นำแบบจำลองหน้าจอที่ได้จากขั้นตอนที่ 040 มาสร้างโครงสร้างของรูปแบบหน้าจอเพิ่มเติม เช่น หน้าจอหลักและย่อย ฟอร์มต่าง ๆ รูปแบบรายงาน เป็นต้น โดยสร้างตามความต้องการที่เกิดขึ้นจาก Requirements Catalogue

งานที่ 20 สร้างโครงสร้างเมนูของแบบจำลอง

งานที่ 30 สร้างแบบจำลองพร้อมกับสร้างเส้นทางการเข้าถึงของแต่ละหน้าจอและรายงานในแต่ละแบบจำลอง

ตั้งแต่งานที่ 4-7 มีการทำซ้ำกันจนกระทั่งการสร้างแบบจำลองสิ้นสุด

งานที่ 40 สร้างแบบจำลอง

งานที่ 50 จัดเตรียมแบบจำลองเพื่อเสนอต่อเจ้าของโครงการ

งานที่ 60 นำเสนอแบบจำลองในแต่ละส่วนให้กับเจ้าของโครงการ

งานที่ 70 รับข้อเสนอแนะจากผู้ใช้

งานที่ 80 ปรับปรุงแบบจำลองตามความต้องการของผู้ใช้

7) ขั้นตอน 370 Develop Processing Specification

งานของขั้นตอน 370

งานที่ 10 สร้างตารางเหตุการณ์/เอนติตีเมตริกซ์ (Entity Acess Matrix) ของแต่ละเหตุการณ์ที่กระทบกับเอนติตีพร้อมกับเอกสารประกอบ

งานที่ 20 นำเอนติตีของ LDS แต่ละตัวมาสร้างเอนติตีไลฟ์ฮิสทรี (ELH-Entity life Histories) ไดอะแกรม โดยเริ่มจากล่างขึ้นบน กล่าวคือ เริ่มสร้างจาก LDS ระดับล่างขึ้นมาก่อน 1 LDS เท่ากับ 1 ELH ซึ่งในระบบจะมี ELH เท่ากับจำนวนของ LDS ที่มีอยู่ จากนั้นนำ ELH มาระบุขั้นตอนการทำงาน หรือ “Operation List” ทำจากบนลงล่าง

งานที่ 30 สร้าง ECD (Effect Correspondence Diagram) ไดอะแกรม ของ LDS ซึ่งเป็นการแสดงความสัมพันธ์ระหว่าง ELH กับเหตุการณ์ที่เกิดขึ้นกับ LDS พร้อมกับระบุขั้นตอนการทำงาน (operation list) หรือลำดับของผลกระทบที่มีต่อแอตทริบิวต์ของแต่ละเอนติตีตามเหตุการณ์ที่เข้ามากระทบ จากนั้น ระบุความต้องการที่เกิดขึ้นของแต่ละเหตุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การณ์โดยสร้าง EAP (Enquiry Access Path) พร้อมคำอธิบาย ซึ่ง EAP คือ การตรวจสอบข้อมูลเข้าและออกที่กระทบกับเอนติตีตามเหตุการณ์ที่เกิดขึ้น

งานที่ 40 ปรับปรุง Requirement Catalogue และ LDS ไดอะแกรมที่มีเหตุการณ์หรือความต้องการใหม่ ๆ และผลกระทบกับความต้องการในระหว่างการสร้าง ELH เพื่อให้ฟังก์ชันที่เกิดขึ้นตรงตามเหตุการณ์ที่เกิดขึ้นระหว่างการวิเคราะห์

งานที่ 50 ระบุจำนวนของข้อมูลที่สามารถบรรจุภายใน LDM ของแต่ละเอนติตี เพื่อประมาณเนื้อที่ของขนาดฐานข้อมูลที่จะต้องใช้ภายในระบบ

สัญลักษณ์และเอกสาร

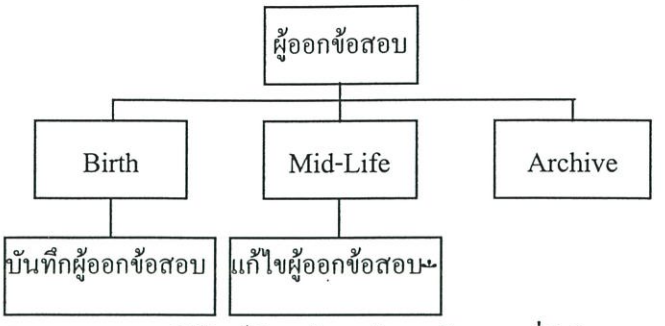
เอนติตีไลฟ์ฮิสทรี (ELH-Entity Life History)

เอนติตีไลฟ์ฮิสทรี (Entity Life History) คือ การแสดงลำดับของเอนติตีที่ถูกเหตุการณ์ใด ๆ เข้ามากระทบ สัญลักษณ์เป็นรูปสี่เหลี่ยมผืนผ้าภายในอธิบายด้วยชื่อของเอนติตี โดยเอนติตีเหล่านี้จะเปลี่ยนสถานะไปตามเหตุการณ์ที่เข้ามากระทบ ซึ่งเปรียบได้กับสเตทไดอะแกรมของระบบเชิงวัตถุ (ดังแสดงในรูปที่ 5.15)

รูปที่ 5.15 แสดงเอนติตีไลฟ์ฮิสทรีของผู้ออกข้อสอบ

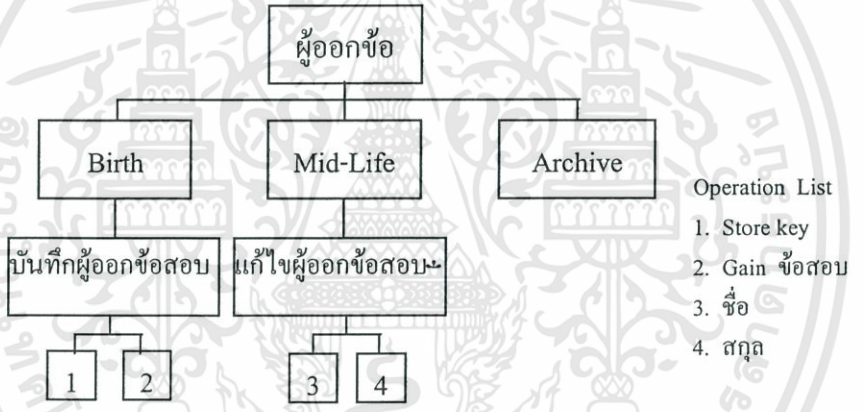
บางครั้งการทำงานซ้ำกันของวงจรชีวิตสำหรับการออกแบบสามารถแสดงได้ในไดอะแกรมเอนติตีไลฟ์ฮิสทรี ถ้ามีลักษณะการทำงานซ้ำ ณ ที่ได้ให้ใส่เครื่องหมายดอกจันบนมุมขวาของการทำงานนั้น ดังแสดงในรูปที่ 5.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.16 แสดงการทำงานของเอนติตีไลฟ์ฮิสทรีของผู้ออกข้อสอบที่มีลักษณะการทำงานซ้ำ

ปกติในเอนติตีไลฟ์ฮิสทรีที่แสดงนั้นไม่สามารถบอกรายละเอียดได้ นักเคราะห์จึงต้องกำหนดลำดับการทำงานก่อนหลังของแต่เอนติตี โดยใส่หมายเลขต่อท้ายพร้อมกับระบุโอเปเรชั่นที่ต้องทำในแต่ละเอนติตี (ดังแสดงในรูปที่ 5.17)



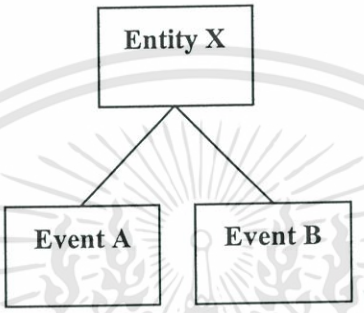
รูปที่ 5. 17 แสดงรายละเอียดเอนติตีไลฟ์ฮิสทรีของผู้ออกข้อสอบ

เหตุการณ์ (Event)

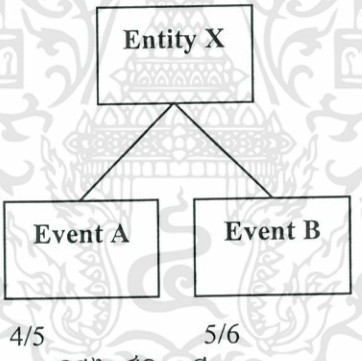
เอนติตีแต่ละตัวจะมีเหตุการณ์ที่เข้ามากระทบให้กับแต่ละเอนติตีเพื่อทำงานและเปลี่ยนสถานะไปจากเดิม ดังนั้นการสร้างไดอะแกรมของเอนติตีไลฟ์ฮิสทรีไม่สามารถแสดงถึงการเปลี่ยนแปลงที่เกิดขึ้นได้ การสร้างเหตุการณ์มีลักษณะสี่เหลี่ยมผืนผ้าที่อยู่ใต้เอนติตีที่เกี่ยวข้อง (ดังแสดงในรูปที่ 5.18) จากนั้นเหตุการณ์ต่าง ๆ เหล่านี้ที่เคยกล่าวไว้ในส่วนของเอนติตีไลฟ์ฮิสทรีที่มีหมายเลขโอเปเรชันกำกับการทำงานจะแสดงเพียงลำดับการทำงานแต่ไม่ได้แสดงถึงสถานะที่เปลี่ยนแปลงไป ถ้าผู้วิเคราะห์ใส่ลำดับการเปลี่ยนสถานะเพิ่มเติมในไดอะแกรมลงไปด้วยย่อมทำให้สามารถบังคับการเปลี่ยนแปลงและลำดับของสเตจที่เปลี่ยนไป หรือสเตทที่มีการทำซ้ำ การระบุสถานะของเหตุการณ์จะเพิ่มตัวเลขเป็นคู่ลำดับ กล่าวคือ หมายเลขตัวแรก คือ เหตุการณ์ก่อนหน้าที่เคยทำมาก่อนและหมายเลขหลัง คือ สเตทที่เปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงไปหลังจากกระบวนการก่อนหน้าเสร็จ ซึ่งหมายเลขทั้งสองถูกขั้นด้วย เครื่องหมาย '/' แต่ถ้าสเตทใดเป็นสเตทที่เริ่มต้นหมายเลขตัวแรกผู้วิเคราะห์ไม่จำเป็นต้องใส่ปล่อยเว้นว่างไว้ ส่วนหมายเลขตัวหลังยังคงต้องใส่เพื่อแสดงการเปลี่ยนสเตท ถ้าเสตทใดเป็นสเตทสุดท้ายหมายเลขตัวหลังไม่จำเป็นต้องใส่เพราะเป็นสเตทสุดท้ายที่ไม่มีการเปลี่ยนแปลง ซึ่งสเตทลักษณะนี้มีการทำงานในลักษณะของการเก็บค่าไว้กับตัวแปรหรือส่งค่าให้กับแอตทริบิวท์ตัวอื่นเรียกใช้ต่อไป แสดงดังรูปที่ 5.19



รูปที่ 5.18 แสดงเหตุการณ์ของเอนติตี X



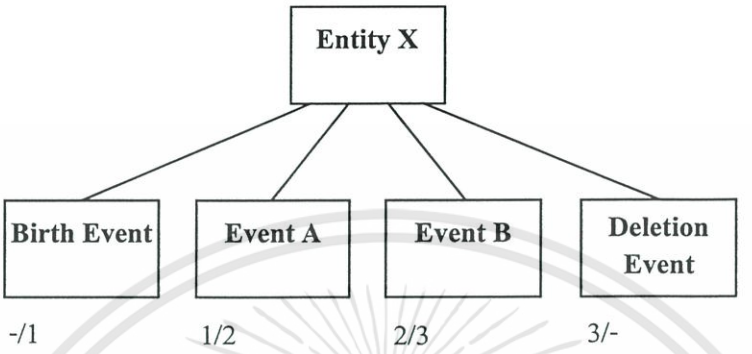
รูปที่ 5. 19 แสดงการใส่ลำดับสเตทของเอนติตีไลฟ์ฮิสทรี

บางครั้งเงื่อนไขของการทำงานในกระบวนการใด ๆ ของเอนติตีอาจมีสเตทบังคับก่อนที่จะกระทำกับกระบวนการนั้น โดยอาจจะมีเพียงหนึ่งหรือมากกว่าได้ ซึ่งจะเขียนโดย "1,2/3" หมายความว่าสเตทก่อนหน้าจะต้องผ่านการทำงานมาจากสเตทที่ 1 หรือ 2 ก่อนเข้ามาทำกับกระบวนการดังกล่าว (ดังแสดงในรูปที่ 5.20)

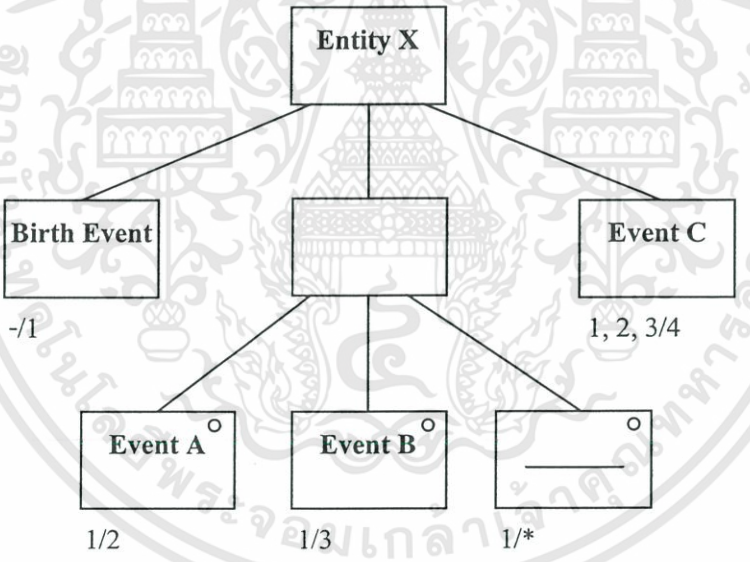
บางไดอะแกรมของเอนติตีไลฟ์ฮิสทรีมีทางเลือกและการทำซ้ำภายในผู้วิเคราะห์ต้องใส่ลำดับการทำงานของสเตทเข้าไปด้วย เพราะถือว่าเป็นกระบวนการหนึ่งที่เกิดขึ้นเมื่อมีเหตุการณ์เข้ามากระทบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการระบุสเตจให้กับเอนติตีไลฟ์ฮิสทรีนั้นเหมือนกับการระบุโดยทั่วไปดังแสดงในรูปที่ 5.21 และ 5.22 ตามลำดับ

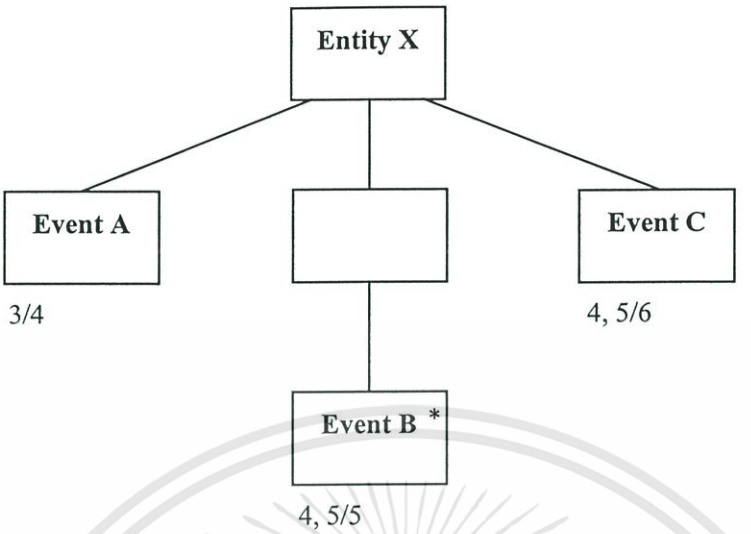


รูปที่ 5.20 แสดงการใส่ลำดับสเตทให้กับเอนติตีไลฟ์ฮิสทรี



รูปที่ 5. 21 แสดงการกำหนดทางเลือกให้กับเอนติตีไลฟ์ฮิสทรีพร้อมกับระบุลำดับของสเตท

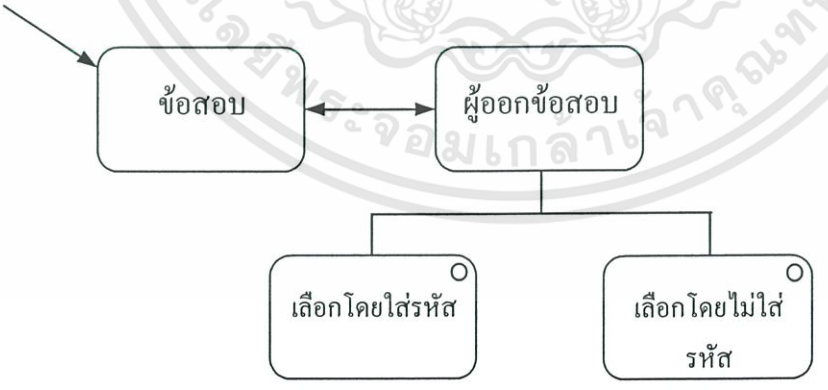
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.22 แสดงการระบุสเตทที่มี 2 สเตทในครั้งเดียวของเอนติตีไลฟ์ฮิสทรี

ECD (Effect Correspondent Diagram)

ECD คือ ไดอะแกรมแสดงความสัมพันธ์และความเคลื่อนไหวของฐานข้อมูลในระบบสารสนเทศที่กำลังพัฒนา สัญลักษณ์ของ ECD มีลักษณะเหมือนกับสัญลักษณ์ของ LDS ที่มีลักษณะเป็นรูปสี่เหลี่ยมผืนผ้าปลายมน ภายในประกอบด้วยชื่อของ ECD (ชื่อของ ECD คือ ชื่อของฐานข้อมูลที่อยู่ใน LDS) ซึ่งไดอะแกรมที่แสดงนั้นไม่ต่างไปจาก LDM ที่แสดงความสัมพันธ์ของฐานข้อมูลในระบบแต่ต่างกันที่เส้นแสดงความสัมพันธ์ โดยเส้นแสดงความสัมพันธ์นั้นแสดงเป็นเส้นตรงระหว่างกันพร้อมกับแสดงลูกศรที่มีเหตุการณ์นั้นเข้ามากระทบกับ LDS แต่ละตัว (ดังแสดงในรูปที่ 5.23)



รูปที่ 5.23 แสดง Effect Correspondent Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงถึงเหตุการณ์ที่เข้ามากระทบเอนติตีข้อสอบและผู้ออกข้อสอบโดยเหตุการณ์กระทบกับเอนติตีข้อสอบก่อนจากนั้นจึงเข้าไปกระทบกับเอนติตีผู้ออกข้อสอบ การระบุเหตุการณ์ที่เข้ามากระทบเอนติตีนั้นมีหลักการพิจารณา 8 ขั้นตอนคือ

1. พิจารณาผลกระทบของแต่ละเหตุการณ์ที่มีต่อ LDS โดยเหตุการณ์ที่นำมาพิจารณานั้นพิจารณาจาก ELHs
2. จำลองเหตุการณ์ที่เกิดขึ้นในแต่ละ LDS พร้อมกับแยกกลุ่มเอนติตีที่เกี่ยวข้อง
3. ถ้าเหตุการณ์ที่เข้ามากระทบกับเอนติตีหลักและทำให้เอนติตีนั้นเกิดทางเลือกให้กำหนดเอนติตีลูกภายใต้เอนติตีหลักตามเหตุการณ์ที่เกิดขึ้น (ซึ่งเอนติตีที่เกิดขึ้นนั้นจะเป็นลักษณะของการเลือกทำอย่างใดอย่างหนึ่ง)
4. ถ้ากลุ่มของเอนติตีใดที่เหตุการณ์เข้ามากระทบมีโอกาสเกิดการทำงานซ้ำ ๆ ให้แยกไดอะแกรมที่มีลักษณะการทำซ้ำต่างหาก
5. กำหนดลูกศรตามเหตุการณ์ที่เข้ามากระทบกับเอนติตีที่มีลักษณะความสัมพันธ์แบบหนึ่งต่อหนึ่งโดยใช้เครื่องหมายลูกศรชี้ระหว่างกัน
6. ถ้าเอนติตีใดมีผลกระทบจากเหตุการณ์และมีการทำงานลักษณะทำซ้ำมากกว่าหนึ่งเหตุการณ์แต่อยู่ภายในเอนติตีเดียวกันให้รวมลักษณะงานดังกล่าวไว้ที่เดียวกันพร้อมกับเปลี่ยนสัญลักษณะการทำซ้ำเป็นลักษณะของการเลือกแทนภายใต้การทำงานซ้ำหรือกำหนดลักษณะของการเลือกภายใต้การทำงานซ้ำ (ตามความเหมาะสม)
7. ถ้าเอนติตีมีการปรับปรุงระหว่างกันไม่จำเป็นต้องแสดง เพราะ ECD คือ การแสดงเหตุการณ์ที่มากระทบกับเอนติตี
8. แสดงรายการของข้อมูลที่เข้าถึงแต่ละเอนติตีตามเหตุการณ์ที่เข้ามากระทบ เช่น คีย์หลัก คีย์นอก หรือ ค่าของข้อมูลอื่น ๆ ที่เกี่ยวข้อง เป็นต้น

จากขั้นตอนทั้งหมดของการสร้าง ECD ไดอะแกรมที่กล่าวมาข้างต้นนั้น ECD ไดอะแกรมไม่จำเป็นต้องทำทุกขั้นตอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) ขั้นตอน 380 รับรองวัตถุประสงค์ของระบบ (Confirm System Objectives)

งานของขั้นตอน 380

งานที่ 10 ตรวจสอบฟังก์ชันและ Non-function ที่กำหนดไว้ครบถ้วนและถูกต้องหรือไม่ โดยพิจารณาแต่ละ Requirement Catalogue ที่มีอยู่เพื่อยืนยันว่าฟังก์ชันและ Non-function ที่ได้ตรงตาม Requirements Catalogue ที่ผู้ใช้ระบบระบุไว้

งานที่ 20 ทำความเข้าใจกับฟังก์ชันและ Non-function หรือ LDS ที่ถูกกำหนดขึ้นมาใหม่ตามความต้องการของผู้ใช้

งานที่ 30 ยืนยันฟังก์ชันที่กำหนดไว้ เช่น ความถูกต้อง ตรงตามความต้องการ สามารถแก้ไขปัญหาได้ถูกต้อง เป็นต้น เพื่อให้ฟังก์ชันนั้นมีความสมบูรณ์และครบถ้วนตรงกับที่ผู้ใช้ระบบต้องการ

งานที่ 40 ยืนยัน LDM ที่กำหนดไว้นั้น มีความถูกต้องและครบถ้วน

9) ขั้นตอน 390 รวมรวมเอกสารและไดอะแกรม

ขั้นตอนที่ 390 คือ การรวบเอกสารและโดอะแกรมเปรียบเสมือนการตรวจสอบสิ่งที่เกิดขึ้นของระยะที่ 3 ตั้งแต่ขั้นตอนที่ 310 จนถึงขั้นตอนที่ 380 ซึ่งถ้าเอกสารต่าง ๆ ที่เกิดขึ้นของแต่ละขั้นตอนนั้นไม่สมบูรณ์หรือไม่ครบถ้วน ผู้วิเคราะห์ควรกลับไปตรวจสอบการทำงานของระยะที่ 3 ตั้งแต่ขั้นตอนที่ 310 จนถึงขั้นตอนที่ 380 อีกครั้ง

5.5.2.2 ระยะที่ 4 การย้ายไปสู่วิธีการเชิงวัตถุ (Migrating to Object-Oriented)

1) ขั้นตอน 410 ตรวจสอบเอกสารและปรับปรุง DFM กับ LDM (Review & Evolution DFM and LDM)

ขั้นตอน 410 เป็นการตรวจสอบไดอะแกรมที่เกิดขึ้นจากระยะที่ 3 เพราะถ้าเอกสารใดขาดหายไป หรือไม่ครบสามารถที่จะกลับไปสร้างหรือแก้ไขได้ทัน ซึ่งถ้าเอกสารหรือไดอะแกรมขาดหายไปหรือมีไม่ครบในระยะที่ 4 นี้ย่อมทำให้ระบบที่เกิดขึ้นมีข้อบกพร่องทันที เพราะขั้นตอนนี้จะทำการย้ายจากวิธีเชิงโครงสร้างไปสู่วิธีเชิงวัตถุเต็มรูปแบบซึ่งไม่ทิ้งไดอะแกรมเหล่านั้นทั้งหมดแต่นำมาเพื่อเทียบเคียงและอ้างอิงในการย้ายไดอะแกรมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานของขั้นตอน 410

งานที่ 10 ตรวจสอบแผนภูมิการไหลของข้อมูลในแต่ละระดับ และ LDS ไดอะแกรม

งานที่ 20 ตรวจสอบเอนติตีไลฟ์ฮีสทรีไดอะแกรมและ ECD

งานที่ 30 ตรวจสอบรายละเอียดของแต่ละฟังก์ชัน

2) ขั้นตอน 420 เปลี่ยน LDS ไปสู่สัญลักษณ์ของวัตถุ/คลาส (Convert LDS To Object/Class Notation)

หลังจากปรับปรุงแผนภูมิดาต้าหรือ อี อาร์ ไดอะแกรมในขั้นตอน 360 เรียบร้อยแล้ว ให้ผู้วิเคราะห์ย้าย LDS เหล่านั้นไปสู่สัญลักษณ์ของวัตถุหรือคลาส มีขั้นตอนดังต่อไปนี้

งานของขั้นตอน 420

งานที่ 10 เปลี่ยนสัญลักษณ์ของ LDS ไปสู่วัตถุหรือคลาส โดยให้ 1 LDS เท่ากับ 1 วัตถุหรือคลาส

งานที่ 20 ให้นำแอตทริบิวต์ที่อยู่ในตารางไปใส่ในวัตถุที่แปลงเรียบร้อยแล้ว (ในช่องที่ 2 ของวัตถุหรือคลาส)

งานที่ 30 ขจัดคีย์หลักของแต่ละวัตถุ

งานที่ 40 เปลี่ยนคีย์รองของแต่ละวัตถุเป็นชื่อของวัตถุที่อ้างอิงถึง เพราะหลักการของวัตถุในการอ้างอิงใช้ Object Identity (OID) แทนการใช้คีย์รอง การอ้างอิงระหว่างวัตถุนั้นระบบฐานข้อมูลจะสร้างหมายเลขขึ้นมาให้เองอัตโนมัติ (ขึ้นอยู่กับฐานข้อมูลที่เลือกใช้) ซึ่งหมายเลขจะไม่ซ้ำกัน (Unique Number) ในแต่ละวัตถุที่อ้างถึง[41]

สัญลักษณ์และเอกสาร

คลาส

สัญลักษณ์ของคลาสมีลักษณะเป็นรูปสี่เหลี่ยมผืนผ้าปลายมน ภายในประกอบด้วยชื่อของคลาสแสดงเป็นตัวหนา ชื่อของคลาสเป็นคำนาม สรรพนาม เช่น คน สัตว์ สิ่งของ เป็นต้น ดังแสดงในรูปที่ 5.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.24 แสดงสัญลักษณ์ของคลาส

วัตถุ

สัญลักษณ์ของวัตถุมีลักษณะเป็นรูปสี่เหลี่ยมผืนผ้าปลายมน แบ่งออกเป็น 3 ส่วน ส่วนแรกแสดงชื่อของวัตถุ ส่วนถัดมาแสดงแอตทริบิวต์ของวัตถุ ส่วนสุดท้ายแสดงฟังก์ชันของวัตถุ ดังแสดงในรูปที่ 5.25

รูปที่ 5.25 แสดงสัญลักษณ์ของวัตถุ

ความสัมพันธ์ของคลาสและวัตถุ

การแสดงความสัมพันธ์ระหว่างคลาส/วัตถุ สามารถแบ่งเป็น 3 ลักษณะ คือ ความสัมพันธ์แบบ Aggregation ความสัมพันธ์แบบ Association และความสัมพันธ์แบบสืบทอดคุณสมบัติ (Inheritance) ซึ่งความสัมพันธ์ทั้ง 3 มีสัญลักษณ์ที่แตกต่างกัน

ความสัมพันธ์แบบ Aggregation มีลักษณะเป็นเส้นตรงปลายด้านหนึ่งเป็นรูสี่เหลี่ยมข้าวหลามตัดสีขาวปลายอีกด้านหนึ่งเป็นเครื่องหมายวงกลม ถ้าระหว่างวัตถุ/คลาสมีความสัมพันธ์แบบ 1:M เครื่องหมายวงกลมเป็นสีดำและถ้าระหว่างคลาส/วัตถุมีความสัมพันธ์แบบ 1:1 เครื่องหมายวงกลมเป็นสีขาว (ดังแสดงในรูปที่ 5.26)

ความสัมพันธ์แบบ Association สัญลักษณ์มี 2 ลักษณะ ลักษณะแรกเป็นเส้นตรงและปลายทั้งสองด้านประกอบด้วยเครื่องหมายวงกลม (กรณีที่ความสัมพันธ์ระหว่างคลาส/วัตถุเป็นแบบทางเลือก) ส่วนวงกลมของปลายทั้งสองจะเป็นสีขาวหรือดำขึ้นกับรูปแบบความสัมพันธ์ กล่าวคือ สีขาวเป็นความสัมพันธ์แบบ 1:1 สีดำเป็นความสัมพันธ์แบบ 1:M ลักษณะที่สองเป็นเส้นตรงปลายด้านหนึ่งประกอบเป็นเครื่องหมายวงกลม สีของวงกลมขึ้นกับรูปแบบความสัมพันธ์ (ดังแสดงในรูปที่ 5.27)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์แบบสืบทอดคุณสมบัติ มีสัญลักษณ์รูปส้อมที่ปลายด้านหนึ่งต่อกับคลาสหลัก ส่วนปลายส้อมต่อกับคลาสที่สืบทอดคุณสมบัติ และรอยต่อระหว่างคลาสหลักกับคลาสที่สืบทอดจะมีรูปสามเหลี่ยมด้านเท่าประกอบอยู่ (ดังแสดงในรูปที่ 5.28)

เอกสารแสดงรายละเอียดของคลาส/วัตถุ

รายละเอียดของคลาส/วัตถุ ประกอบด้วย ชื่อคลาส/วัตถุ ชนิด รูปแบบ ลักษณะงานและความเกี่ยวข้องกับคลาสอื่นในระบบ (ดังแสดงในตารางที่ 5.12)

ตารางที่ 5.12 แสดงรายละเอียดของคลาส/วัตถุ

Class name:	
class type: (e.g., device, property, role, event, . . .)	
class characteristics: (e.g., tangible, atomic, concurrent, . . .)	
Responsibilities:	Collaborators:

3) ขั้นตอน 430 กำหนดความสัมพันธ์ระหว่างวัตถุ/คลาส (Define Relation)

การกำหนดความสัมพันธ์ระหว่างวัตถุให้พิจารณาจาก O/C ไดอะแกรมที่แปลงมาจาก LDS ไปสู่ความสัมพันธ์แบบวัตถุ ซึ่งสัญลักษณ์ที่ใช้เป็นวิธีของ OMT การเปลี่ยนเส้นความสัมพันธ์ระหว่างเอนติตี

งานของขั้นตอน 430

งานที่ 10 พิจารณาความสัมพันธ์ระหว่างเอนติตีใน LDS

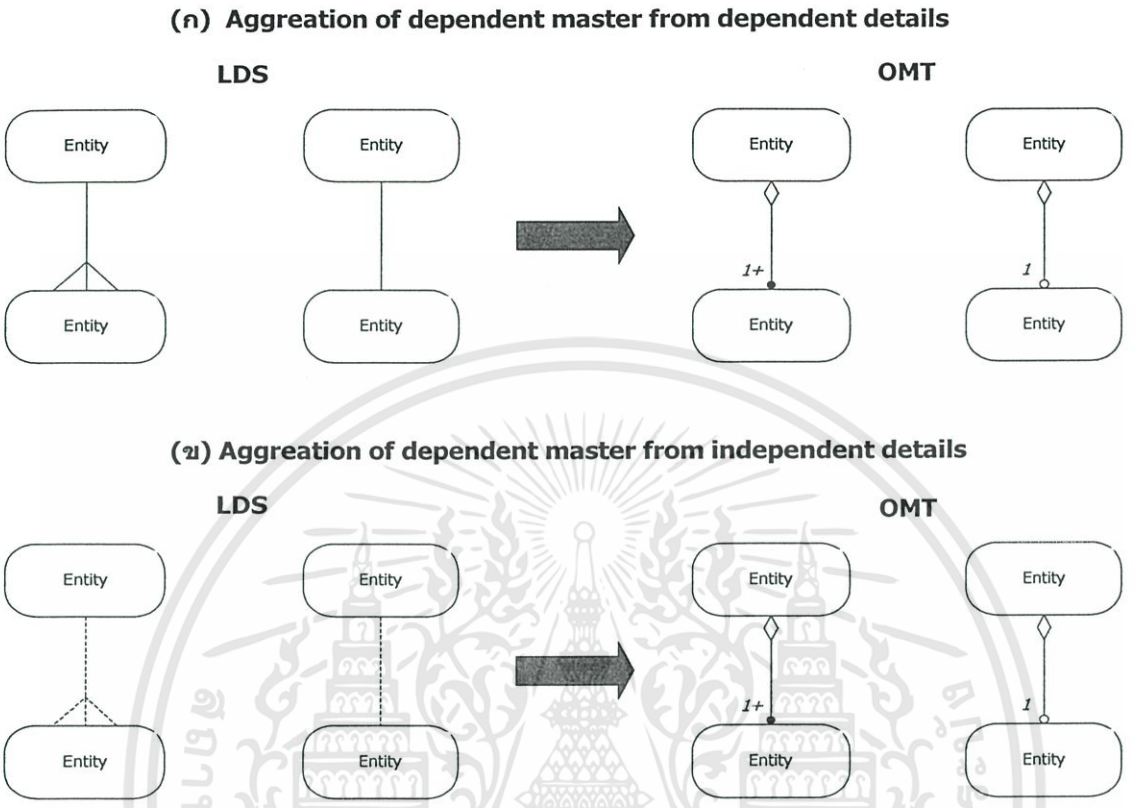
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 20 แปลงสัญลักษณ์ความสัมพันธ์ของไดอะแกรม โดยมีหลักเกณฑ์ดังนี้

- ความสัมพันธ์ระหว่างเอนติตีที่มีลักษณะ Aggregation ของ LDS รูปที่ 5.26 (ก) แสดงความสัมพันธ์ของเอนติตีหลัก (Master) มีความสัมพันธ์กับเอนติตีรายละเอียด (Detail) และต้องขึ้นตรงต่อกันทั้งในกรณี 1:M หรือ 1:1 โดยเปลี่ยนไปสู่ความสัมพันธ์แบบวิธีเชิงวัตถุ กรณีที่มีความสัมพันธ์แบบ 1:M คือ มีความสัมพันธ์กันตั้งแต่ 1 ความสัมพันธ์ขึ้นไป เพื่อเปลี่ยนไปสู่ความสัมพันธ์ในรูปของ Aggregation และระบุจำนวนของความสัมพันธ์ที่เกิดขึ้น (อาจระบุจำนวนมากกว่าได้ เช่น 1...n เป็นต้น) ส่วนความสัมพันธ์แบบ 1:1 เป็นความสัมพันธ์ที่เอนติตีหลักจำเป็นต้องมีความเกี่ยวข้องกับเอนติตีรายละเอียดอย่างน้อย 1 ความสัมพันธ์ รูปที่ 5.26 (ข) แสดงความสัมพันธ์ของเอนติตีหลักที่อาจมีความสัมพันธ์กับเอนติตีรายละเอียดหรือไม่มีก็ได้ หรือเรียกว่า "Optional" โดยความสัมพันธ์แบบ 1:M จะมีเส้นความสัมพันธ์คล้ายกับ รูปที่ 5.26(ก) ส่วนกรณีที่มีความสัมพันธ์แบบ 1:1สามารถแสดงดังรูปทางขวามือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Convert LDS to OMT notation



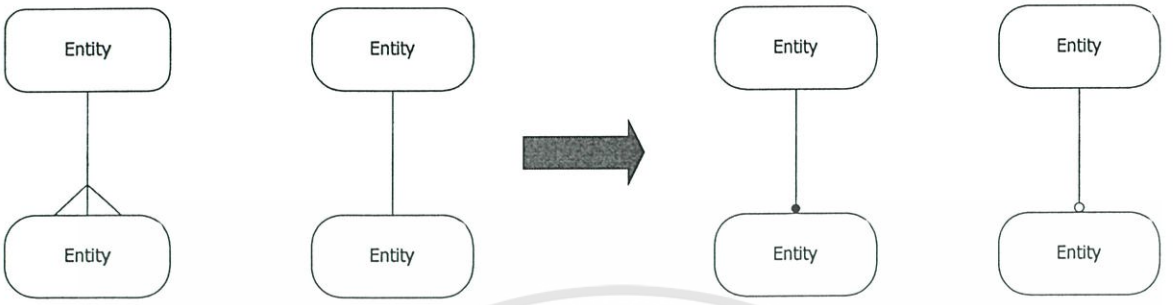
รูปที่ 5.26 ก) แสดงการเปลี่ยนความสัมพันธ์แบบเอนติตีหลักและรองขึ้นตรงต่อกัน

ข) แสดงการเปลี่ยนความสัมพันธ์แบบเอนติตีหลักไม่ขึ้นตรงกับเอนติตีรอง

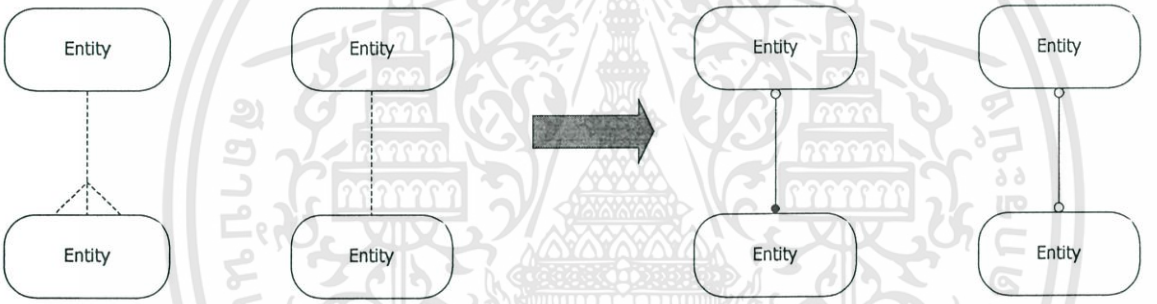
- ความสัมพันธ์ระหว่างเอนติตีที่มีลักษณะ Association ของ LDS รูปที่ 5.26 (ก) แสดงความสัมพันธ์ของเอนติตีหลักที่สัมพันธ์กับเอนติตีรายละเอียดในลักษณะที่เอนติตีหลักไม่จำเป็นต้องมีความสัมพันธ์แต่เอนติตีรายละเอียดจำเป็นต้องมีอย่างน้อย 1 ความสัมพันธ์ ส่วนรูปที่ 5.26 (ข) แสดงความสัมพันธ์ของเอนติตีหลักที่สัมพันธ์กับเอนติตีรายละเอียดในลักษณะของเอนติตีหลักยังต้องสัมพันธ์กับเอนติตีรายละเอียดถึงแม้ว่าจะเป็นความสัมพันธ์แบบ Optional ทั้งแบบ 1:M หรือ 1:1 ก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) **Association of independent master with dependent details**



(ข) **Association of dependent master with dependent details**

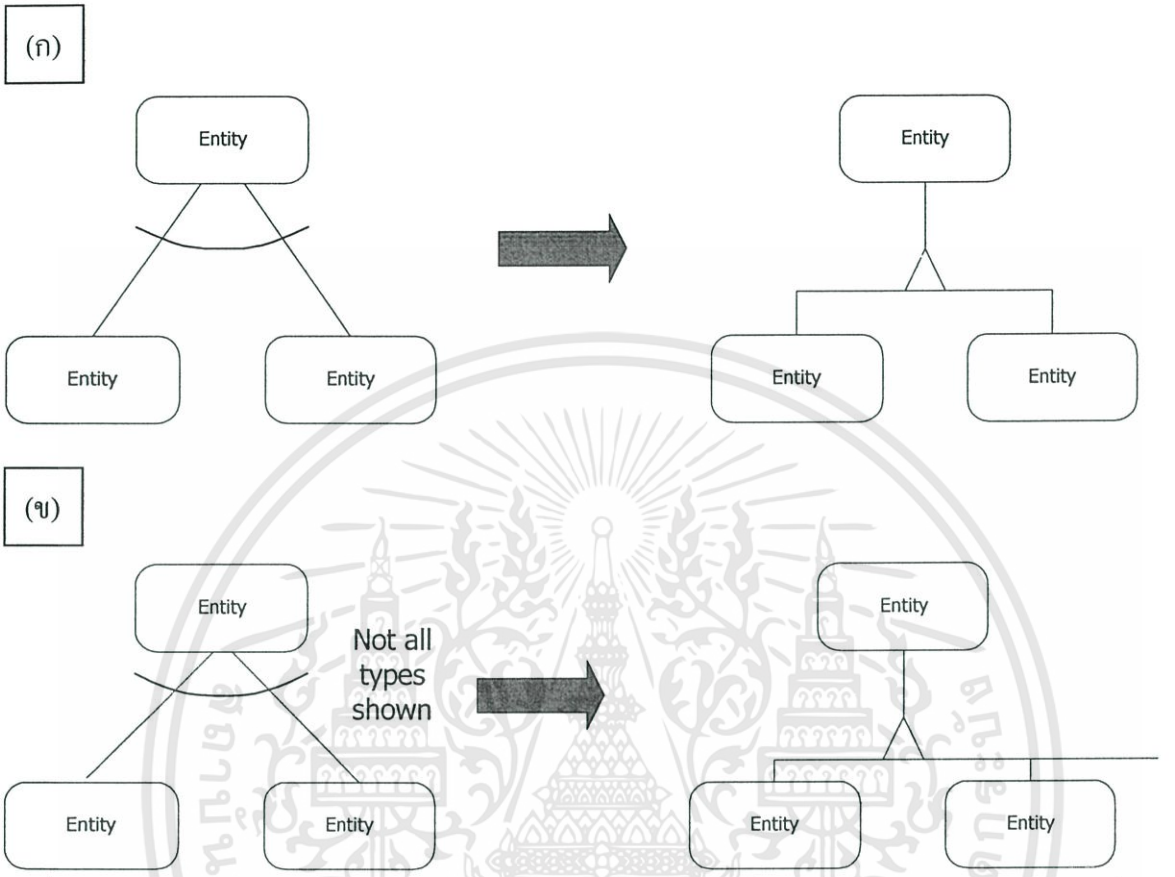


รูปที่ 5.27 ก) แสดงการเปลี่ยนความสัมพันธ์แบบเอนติตีหลักและรองขึ้นตรงต่อกัน
ข) แสดงการเปลี่ยนความสัมพันธ์แบบเอนติตีหลักไม่ขึ้นตรงกับเอนติตีรอง

- ความสัมพันธ์ระหว่างเอนติตีแบบลำดับชั้นที่มีลักษณะ Mutual คือ เอนติตีหลัก (Super-Type) กระจายความสัมพันธ์ไปสู่เอนติตีลูก (Sub-Types) จากรูปที่ 5.28 แสดงถึงการแปลงสัญลักษณะของ LDS ไปสู่ OMT กล่าวคือการกระจายเอนติตีในลักษณะนี้เหมือนกับการสืบทอดคุณสมบัติของวิธีเชิงวัตถุ ซึ่งสามารถแปลงได้ดังรูป ทั้งในกรณีที่ความสัมพันธ์แบบ Mandatory หรือ Optional

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class hierarchy of Super-Type and Sub-Types



รูปที่ 5.28 ก) แสดงการแปลงความสัมพันธ์ของเอนติตีหลักและรองกรณีที่ถ่ายทอดถึงกันแบบไม่มีเงื่อนไข

ข) แสดงการแปลงความสัมพันธ์ของเอนติตีหลักและรองกรณีที่ถ่ายทอดถึงกันแบบมีเงื่อนไข

งานที่ 30 ตรวจสอบไดอะแกรมที่แปลงสัญลักษณ์จาก LDS ไป OMT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ขั้นตอน 440 ย้ายฟังก์ชันจากแผนภูมิการไหลของข้อมูลไปสู่ฟังก์ชันของวัตถุ

การย้ายฟังก์ชันที่มีอยู่ในแผนภูมิการไหลของข้อมูลไปสู่ฟังก์ชันของวัตถุโดยขั้นตอนการพิจารณาฟังก์ชันให้นำเอนติตีไลฟ์ฮีสทรีไดอะแกรม และ ECD ไดอะแกรมมาประกอบ การพิจารณาฟังก์ชันผู้วิเคราะห์ควรเริ่มจากเอนติตีไลฟ์ฮีสทรีไดอะแกรมของแต่ละเอนติตี ซึ่งถ้าพิจารณาจาก ELH พบว่าในไดอะแกรมของเอนติตีไลฟ์ฮีสทรีไดอะแกรมประกอบด้วย "Birth", "Mid-life" และ "Archive" พร้อมทั้งเงื่อนไขต่าง ๆ และแต่ละกล่องจะแยกย่อยออกเป็นฟังก์ชันต่าง ๆ ซึ่งฟังก์ชันเหล่านี้ มาจากดาต้าโฟลว์ไดอะแกรมที่เคยได้สร้างไว้และมีขั้นตอนการทำงานดังนี้

งานของขั้นตอน 440

งานที่ 10 ตรวจสอบเอนติตีไลฟ์ฮีสทรีไดอะแกรมและ ECD ของแต่ละเอนติตีพร้อมกับ Operation List ของแต่ละไดอะแกรม

งานที่ 20 จัดกลุ่มของเอนติตีไลฟ์ฮีสทรีไดอะแกรมและ ECD ที่มีการทำงานคล้ายกันเพื่อนำมาจัดเรียงใหม่ ให้อยู่ในรูปของ Jackson Diagram เพื่อเป็นการยืนยันว่าฟังก์ชั่นและแอตทริบิวต์ภายในที่เกี่ยวข้องกับฟังก์ชันที่เกิดขึ้นนั้น ถูกต้องไม่ผิดพลาดหรือไม่ได้เกิดขึ้นมาลอย ๆ

งานที่ 30 นำฟังก์ชันในของแต่ละเอนติตีไลฟ์ฮีสทรีไดอะแกรมที่จัดกลุ่มเรียบร้อยแล้ว ไปสู่ฟังก์ชันของวัตถุ ขั้นตอนการใส่ฟังก์ชันในวัตถุผู้วิเคราะห์ นำเอนติตีไลฟ์ฮีสทรีไดอะแกรมและECD ที่เปลี่ยนไปสู่ Jackson Diagram มาพิจารณา กล่าวคือ นำชื่อฟังก์ชันที่เป็นฟังก์ชันระดับสุดท้ายของไดอะแกรมมาพิจารณาเท่านั้น เพราะฟังก์ชันที่มีการเรียกซ้ำหรือมีทางเลือกนั้น (ระดับที่อยู่ก่อนหน้าและไม่มี"operation list" กำกับ) ไม่ใช่ฟังก์ชันสุดท้ายของการทำงานไม่จำเป็นต้องคำนึงถึง แต่ฟังก์ชันที่มีหลายเลขการทำงานกำกับจึงจะเป็นฟังก์ชันที่มีการทำงานจริงและนำมาใส่ในวัตถุ กล่าวคือ ฟังก์ชันที่มี "operation list" กำกับเท่านั้นจึงจะนำมาพิจารณา

งานที่ 40 ปรับปรุงวัตถุพร้อมกับตรวจสอบวัตถุไดอะแกรม หลังจากนำฟังก์ชันมาใส่ในวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ขั้นตอน 450 การตรวจสอบการย้ายจากวิธีเชิงโครงสร้างไปสู่วิธีเชิงวัตถุ (Object-Oriented Testing)

ขั้นตอนที่ 450 เป็นขั้นตอนการตรวจสอบความถูกต้องของการย้ายจากวิธีเชิงโครงสร้างไปสู่วิธีเชิงวัตถุก่อนนำไดอะแกรมทั้งหมดไปสู่ระยะที่ 5 ที่กล่าวถึงการออกแบบระบบสารสนเทศเชิงวัตถุ และนำไปสู่ขั้นตอนการเขียนโปรแกรมต่อไป การตรวจสอบมีลำดับการทำงานดังต่อไปนี้

งานของขั้นตอนที่ 450

งานที่ 10 ตรวจสอบการเปลี่ยนแปลงจากวิธีเชิงโครงสร้างไปสู่วิธีเชิงวัตถุที่เปลี่ยนจาก LDS ไดอะแกรม

งานที่ 20 ตรวจสอบความสัมพันธ์ของแต่ละวัตถุ และ/หรือ คลาส

งานที่ 30 ตรวจสอบแอตทริบิวต์และฟังก์ชัน (ที่แปลงมาจากเอนติตีไลฟ์ฮีสทรีไดอะแกรม)ของแต่ละวัตถุ

งานที่ 40 ปรับปรุงรายละเอียดของแต่ละวัตถุให้ถูกต้อง

6) ขั้นตอน 460 รวบรวมเอกสารและตรวจสอบไดอะแกรม

ขั้นตอนที่ 460 คือ การรวบเอกสารและโดอะแกรมเปรียบเสมือนการตรวจสอบสิ่งที่เกิดขึ้นของระยะที่ 4 ตั้งแต่ขั้นตอนที่ 410 จนถึงขั้นตอนที่ 450 ซึ่งถ้าเอกสารต่าง ๆ ที่เกิดขึ้นของแต่ละขั้นตอนนั้นไม่สมบูรณ์หรือไม่ครบถ้วน ผู้วิเคราะห์ควรกลับไปตรวจสอบการทำงานของระยะที่ 4 ตั้งแต่ขั้นตอนที่ 410 จนถึงขั้นตอนที่ 450 อีกครั้ง

5.5.3 โมดูลการออกแบบ (Design Module)

5.5.3.1 ระยะที่ 5 การออกแบบระบบ (System Design)

1) ขั้นตอน 510 การจัดซื้อจัดจ้างฮาร์ดแวร์และซอฟต์แวร์ (Hardward and Software procurements)

การจัดซื้อจัดจ้างฮาร์ดแวร์และซอฟต์แวร์ เป็นขั้นตอนที่มีความจำเป็นอีกขั้นตอนหนึ่ง เนื่องจากระบบที่วิเคราะห์และออกแบบซึ่งจะสำเร็จตามวัตถุประสงค์หรือไม่นั้นขึ้นอยู่กิจกรรมนี้ด้วย โดยหน้าที่ของการจัดหาส่วนใหญ่อยู่ในความรับผิดชอบของผู้จัดการโครงการ (Project Manager) ซึ่งเป็นผู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ติดต่อและประสานงาน รายละเอียดของการจัดหาทั้งฮาร์ดแวร์และซอฟต์แวร์ ซึ่งในขั้นตอน 510 มีการทำงานดังนี้

งานของขั้นตอน 510

งานที่ 10 นำความต้องการทางธุรกิจและเทคนิคที่เกิดขึ้นจากขั้นตอนที่ 210 เพื่อนำรายละเอียดทั้งหมด มาปรับเปลี่ยนให้อยู่ในรูปของ "เอกสารแสดงข้อมูลที่ต้องการ" (RFI- Request for Information)[3] ประกอบด้วย ความต้องการทั่วไป คุณสมบัติของฮาร์ดแวร์และซอฟต์แวร์ ลักษณะของสินค้า เป็นต้น

งานที่ 20 กระจาย RFI ให้กับผู้ขายที่มาจากการสรรหา เช่น การประกาศทางวิทยุ โทรทัศน์ หนังสือพิมพ์ และสื่ออื่น ๆ หรือติดต่อทางผู้จัดการโครงการ เป็นต้น

งานที่ 30 รับรายงานในรูปของใบเสนอรายละเอียดของสินค้า (RFQ-Request for Quotations) [12] และ รายงานคุณสมบัติของฮาร์ดแวร์และซอฟต์แวร์ (RFP - Request for proposals)[3] จากผู้ขายหลายราย ซึ่งรายละเอียดของรายงานทั้ง 2 ฉบับจะแสดงรายละเอียดต่างกัน ใบเสนอรายละเอียดของสินค้า จะไม่กล่าวถึงลักษณะและคุณสมบัติของสินค้า แต่กล่าวถึงราคาของฮาร์ดแวร์ ซอฟต์แวร์ เงื่อนไขการส่งมอบและบริการหลังการขาย เป็นต้น ส่วนรายงานคุณสมบัติของฮาร์ดแวร์และซอฟต์แวร์ ประกอบด้วย ความสามารถของฮาร์ดแวร์ ซอฟต์แวร์ ติดตั้งง่าย ใช้งานสะดวก การบำรุงรักษาและความเข้ากันได้กับระบบงานอื่นที่มีอยู่ (Compatible) เป็นต้น

งานที่ 40 นำรายงานของทุกบริษัทเข้าที่ประชุมเพื่อเลือกเฉพาะบริษัทที่เหมาะสม ซึ่งการประชุมคัดเลือกบริษัทที่ทำสัญญาซื้อขายด้วยนั้น ต้องมีคุณสมบัติดังนี้ 1) มีระบบสนับสนุนหลังการขายที่ดีพอ เช่น ทีมสนับสนุน เครื่องมือ และเจ้าหน้าที่ที่มีความชำนาญ เป็นต้น 2) มีความมั่นคงในการดำเนินงาน คือ ธุรกิจดำเนินงานมาเป็นเวลาพอสมควรที่พอจะเชื่อมั่นได้ และ 3) จำนวนสินค้าที่เป็นตัวแทนจำหน่าย ซึ่งถ้ามีสินค้าที่เป็นตัวแทนจำหน่ายอยู่หลายประเภทจะทำให้การบริการหลังการขายมีคุณภาพต่ำลงไปด้วย

งานที่ 50 ทำการเลือกบริษัทจากมติของที่ประชุม พร้อมกับทำสัญญาซื้อขายอุปกรณ์ทั้งฮาร์ดแวร์และซอฟต์แวร์ พร้อมกับติดต่อบริษัทผู้ขายเข้ามาทำสัญญาและส่งมอบสินค้าให้กับบริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานสุดท้ายของขั้นตอนนี้ คือ การส่งมอบฮาร์ดแวร์และซอฟต์แวร์ซึ่งสินค้าทั้งหมดที่ส่งมอบจะต้องทำการติดตั้งและตรวจสอบก่อน การติดตั้งและตรวจสอบสินค้า คือ การตรวจสอบสภาพของฮาร์ดแวร์และซอฟต์แวร์ให้สมบูรณ์และสามารถใช้งานได้ (ซึ่งไม่ใช่การทดสอบระบบ)

สัญลักษณ์และเอกสาร

1. รายละเอียดของฮาร์ดแวร์และซอฟต์แวร์ที่ต้องการสำหรับโครงงาน
2. เอกสารเสนอราคาฮาร์ดแวร์และซอฟต์แวร์
3. มติที่ประชุมสำหรับการคัดเลือกบริษัทสำหรับจัดหาฮาร์ดแวร์และซอฟต์แวร์
4. สัญญาสำหรับการซื้อขาย

2) ขั้นตอน 520 ทบทวนและตรวจสอบความก้าวหน้าของไดอะแกรมและเอกสาร (Review and Evolution Diagram and Document)

หลังจากระยะที่ 4 ที่ได้เปลี่ยนไดอะแกรมจากวิธีเชิงโครงสร้างไปสู่ไดอะแกรมตามวิธีเชิงวัตถุนั้น ไดอะแกรมทั้งหมดจะเป็นไดอะแกรมเชิงวัตถุทั้งสิ้น วัตถุประสงค์ของขั้นตอน 520 คือ การทบทวนและตรวจสอบไดอะแกรมกับเอกสาร

เอกสารที่ต้องตรวจสอบ คือ Use Case และ Requirements Catalogue โดยปกติการปรับปรุงในขั้นตอนที่ผ่านมา ผู้พัฒนาจะปรับปรุงเฉพาะส่วนของ Requirements Catalogue แต่พอมาถึงกิจกรรมนี้วิธีการพัฒนาได้ถูกเปลี่ยนไปทำให้การใช้งานของเอกสารเปลี่ยนรูปแบบไป แต่ไม่ได้หมายความว่า Requirements Catalogue จะถูกทิ้งไป แต่วิธีการนำไปใช้งานนั้นต่างกัน กล่าวคือ Requirements Catalogue ใช้ในระดับฟังก์ชันหรือแผนภูมิการไหลของข้อมูล ส่วน Use Case นำมาใช้กับไดอะแกรมแบบเชิงวัตถุเพื่อตรวจสอบลักษณะงานของผู้ใช้ การตรวจสอบเอกสารทั้ง 2 ประเภทจะเป็นประโยชน์ต่อทีมงานพัฒนาระบบ เพราะถ้าการพัฒนาถูกแบ่งออกเป็นระบบย่อย เอกสารต่าง ๆ จึงจำเป็นต้องมีความเข้าใจกับความต้องการของผู้ใช้ให้ตรงกัน ถ้าเข้าใจไม่ตรงกันย่อมทำให้ระบบที่พัฒนาขึ้นเกิดปัญหาในส่วนของการรวมกันของระบบภายหลังได้ ดังนั้นทีมพัฒนาควรที่จะปรับเอกสารที่เกี่ยวข้องทั้งหมดให้ตรงกัน ก่อนที่จะแบ่งการพัฒนาออกเป็นระบบงานย่อยในขั้นตอนที่ 530 ต่อไป

การตรวจสอบไดอะแกรมในที่นี้ คือ ตรวจสอบไดอะแกรมเชิงวัตถุทั้งหมดโดยตรวจสอบกับ User Catalogue เพราะเป็นการยืนยันความแน่ใจกับไดอะแกรมที่เกิดขึ้นหลังจากการย้ายจากวิธีเชิงโครงสร้างไปสู่วิธีเชิงวัตถุ เนื่องจากการย้ายนั้น บางครั้งอาจมีฟังก์ชันหรือความสัมพันธ์ในบางวัตถุผิดพลาดหรือตกหล่นระหว่างการย้าย เช่น ฟังก์ชันบางตัวในแผนภูมิการไหลของข้อมูลมีลักษณะแบบทั่วไปซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ได้ระบุเป็นของวัตถุใดวัตถุหนึ่ง แต่ผู้ออกแบบได้นำไปใส่ไว้ในทุกวัตถุที่เกี่ยวข้อง แทนที่จะแยกออกมาเป็นอีกหนึ่งวัตถุต่างหาก หรือแอตทริบิวต์ในแต่ละวัตถุยังคงใช้หลักการของฐานข้อมูลเชิงสัมพันธ์อยู่ ควรเปลี่ยนไปสู่หลักการของวิธีเชิงวัตถุ เป็นต้น ดังนั้นการตรวจสอบไดอะแกรมของขั้นตอนนี้เป็นการตรวจสอบขั้นสุดท้ายก่อนที่จะมีการแบ่งระบบออกเป็นส่วนย่อยเพื่อให้กับทีมงานพัฒนาต่อไป ซึ่งบางครั้งอาจไม่จำเป็นต้องทำกิจกรรมนี้ก็ได้ ถ้าไดอะแกรมที่มีอยู่นั้นสมบูรณ์แล้ว

3) ขั้นตอน 530 กำหนดระบบงานย่อยจากวัตถุแบบจำลองวัตถุ (Define Sub-System from Object Model)

การแบ่งระบบงานออกเป็นระบบย่อยนั้นมักจะเกิดกับระบบใหญ่ ๆ ซึ่งจะถูกแบ่งระบบงานออกเป็นส่วนย่อยจาก O/C ไดอะแกรม เพื่อให้ระบบสามารถพัฒนาได้สะดวกมากขึ้น เพราะในบางโครงงาน

งานของขั้นตอน 530

งานที่ 10 จัดกลุ่มของวัตถุที่มีความสอดคล้องกันหรือกลุ่มของวัตถุที่สามารถพัฒนาไปในแนวทางเดียวกันเข้าด้วยกัน ที่เรียกว่า "แพ็กเกจ (Package)" หรือ "ระบบย่อย (Sub-System)" ซึ่งประกอบด้วย คลาส แอสโซซิเอชั่น โอเปเรชันและเหตุการณ์ ที่มีการเชื่อมต่อความสัมพันธ์ระหว่างระบบย่อยอื่น ๆ ซึ่งรวมอยู่ในกล่องสี่เหลี่ยมผืนผ้าที่ประกอบด้วย ชื่อของระบบย่อย การติดต่อระหว่างระบบย่อยกับคลาสหรือวัตถุที่เกี่ยวข้องกันโดยใช้ลูกศรคู่บอกทิศทางความสัมพันธ์ พร้อมกับมีเอกสารคอยแจกแจงรายละเอียดซึ่งเรียกว่า "Sub-system Responsibility Specification (SRS)"

งานที่ 20 รวบรวมงานและลำดับการทำงานในแต่ละวัตถุของแต่ละระบบย่อย โดยปกติของโมเดลวิเคราะห์วัตถุส่วนใหญ่มักจะมีการทำงานพร้อมกัน ซึ่งในขั้นตอนอิมพลีเมนต์กระบวนการที่อยู่ในแต่ละวัตถุมักจะสนับสนุนมากกว่าหนึ่งวัตถุ ดังนั้นสิ่งที่ต้องเกิดขึ้นกับขั้นตอนอิมพลีเมนต์ คือ ผู้ออกแบบต้องกำหนดงานของแต่ละวัตถุที่มีลักษณะการทำงานพร้อมกันและมีการทำงานแบบเดียวกันเข้าไว้ในที่เดียวกัน

งานที่ 30 จัดสรรอุปกรณ์ทั้งฮาร์ดแวร์และซอฟต์แวร์ให้กับระบบย่อย (จัดสรรให้กับกระบวนการและงานในระบบย่อย) ซึ่งการเลือกอุปกรณ์ต่าง ๆ ควรที่จะคำนึงถึงความเหมาะสมของแต่ระบบย่อยเป็นหลัก การเลือกอุปกรณ์ของกิจกรรมย่อยนี้จะเป็นกิจกรรมต่อเนื่องจาก ตั้งแต่ระยะที่ 2 (การกำหนดแนวทางการแก้ไขปัญหาทางด้านเทคนิคและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ธุรกิจ) และขั้นตอนที่ 510 (การจัดซื้ออุปกรณ์ทั้งฮาร์ดแวร์และซอฟต์แวร์) ที่เคยปฏิบัติมาแล้ว ดังนั้นในขั้นตอนย่อยนี้ คือ การจัดอุปกรณ์ที่มีอยู่ (จากการจัดซื้อและการแก้ไขปัญหา) ให้เหมาะสมกับระบบย่อยที่มีอยู่ในระบบทั้งหมดโดยต้องคำนึงถึงความต้องการของฟังก์ชันที่อยู่ในระบบย่อยด้วย

งานที่ 40 จัดสรรฐานข้อมูลให้กับระบบย่อย โดยการเลือกใช้ฐานข้อมูลควรคำนึงถึงรูปแบบของฐานข้อมูลที่มีอยู่ในปัจจุบัน เช่น ฐานข้อมูลแบบลำดับขั้น (Hierarchical) ฐานข้อมูลเชิงสัมพันธ์ (Relational) ฐานข้อมูลแบบเครือข่าย (Network) และฐานข้อมูลเชิงวัตถุ (Object-Oriented) เป็นต้น

งานที่ 50 บันทึกรายละเอียดลงไปในเอกสารของการพัฒนาระบบย่อย โดยสิ่งที่ต้องระบุถึงฮาร์ดแวร์ ซอฟต์แวร์และการเชื่อมต่อระหว่างฮาร์ดแวร์กับซอฟต์แวร์ ฮาร์ดแวร์ที่บันทึกลงในเอกสารควรระบุให้ชัดเจน เช่น จำนวน ซีพียู ความเร็วและความสามารถของการรองรับทรานส์เอ็กชันในแต่ละครั้ง เป็นต้น ส่วนซอฟต์แวร์ที่เลือกในแต่ละระบบย่อยต้องคำนึงถึง คือ ภาษาที่ใช้ แอพพิลเคชันที่เลือก สุดท้าย ควรระบุรูปแบบของฐานข้อมูลที่เลือกใช้ในแต่ละระบบย่อย

4) ขั้นตอน 540 กำหนดแผนงานการทำซ้ำ (Define Iteration Plan Development)

การกำหนดแผนงานของการทำงานซ้ำ คือ การกำหนดระยะเวลาการทำงานของระบบย่อย โดยการวางแผนการทำงานมีลักษณะและขั้นตอนการทำงานคล้ายกับขั้นตอน 060 ซึ่งนักออกแบบจะนำรายละเอียดเหล่านั้นกลับมาพิจารณาอีกครั้งเพื่อปรับเปลี่ยนให้เหมาะสมกับระบบย่อยที่ต้องรับผิดชอบ มีการทำงานดังนี้

งานของขั้นตอน 540

งานที่ 10 จัดทำแผนงานให้กับแต่ละระบบย่อย โดยนำเสนอในรูปแบบของ GANTT Chart หรือ PERT/CPM ซึ่งจะแสดงเฉพาะส่วนที่ระบบย่อยนั้นรับผิดชอบเท่านั้น

งานที่ 20 กำหนดขั้นตอนพร้อมระบุเวลาการทำงานของระบบย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ขั้นตอน 550 ตรวจสอบระบบงานย่อย <ถ้าต้องการ> (Revise Sub-System, if need)

ขั้นตอนที่ 550 เป็นขั้นตอนที่กล่าวถึงการตรวจสอบระบบย่อย ที่แบ่งเรียบร้อยแล้ว มีวัตถุประสงค์เพื่อ ตรวจสอบความเรียบร้อยและความถูกต้องของเพ็กเกจของระบบย่อย ที่ได้ทำไปแล้ว ซึ่งขั้นตอนนี้ผู้ออกแบบจะเข้ามาปฏิบัติหรือไม่ปฏิบัติก็ได้ ถ้าไม่ปฏิบัติมักจะมี 2 กรณี กล่าวคือ

1. ระบบมีขนาดไม่ใหญ่และไม่จำเป็นต้องแบ่งงานออกเป็นส่วนย่อย
2. ระบบย่อยที่แบ่งมีความชัดเจนและถูกต้องอยู่แล้ว

การตรวจสอบระบบงานย่อยของระบบจะจำเป็นหรือไม่ขึ้นอยู่กับประสบการณ์และความสามารถของผู้ออกแบบเป็นหลัก

6) ขั้นตอน 560 สรุปการทำงาน (Summary of process and deliverables)

ขั้นตอนที่ 560 คือ การสรุปการทำงานเปรียบเสมือนการตรวจสอบเอกสารต่าง ๆ ที่เกิดขึ้นของระยะที่ 5 ตั้งแต่ขั้นตอนที่ 510 จนถึงขั้นตอนที่ 550 ซึ่งถ้าเอกสารของแต่ละขั้นนั้นไม่สมบูรณ์หรือไม่ครบถ้วน ผู้วิเคราะห์ควรกลับไปตรวจสอบการทำงานของระยะที่ 5 ตั้งแต่ขั้นตอนที่ 510 จนถึงขั้นตอนที่ 550 อีกครั้ง เอกสารของขั้นตอนนี้มีดังนี้

1. รายละเอียดของฮาร์ดแวร์และซอฟต์แวร์
2. ใบเสนอราคาของบริษัทต่าง ๆ
3. มติที่ประชุม
4. ใบสัญญาการซื้อขาย
5. แพ็จเกจไดอะแกรม
6. SRS ไดอะแกรม
7. แผนการพัฒนาของระบบย่อย

5.5.3.2 ระยะที่ 6 การออกแบบเชิงวัตถุ (Object Design)

1) ขั้นตอน 610 การออกแบบอัลกอริทึม (Design Algorithms)

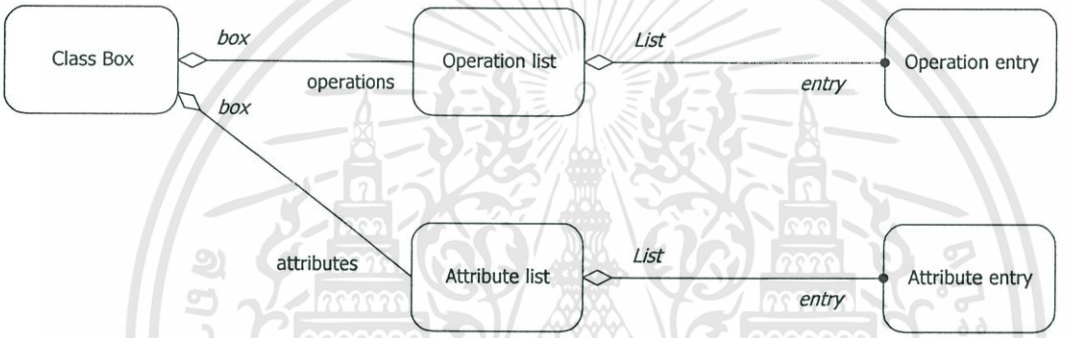
การออกแบบอัลกอริทึมของ OMT นั้น ในแผนผังการไหลของข้อมูลและรายบละเอียดของฟังก์ชันแสดงรายละเอียดคร่าว ๆ ไว้แล้ว เปรียบเสมือนกับการนำเสนออัลกอริทึมเป็นที่เรียบร้อยแล้ว แต่การออกแบบอัลกอริทึมนั้นควรคำนึงถึง ค่าใช้จ่ายของการอิมพลีเมนต์อัลกอริทึม ความเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของโครงสร้างข้อมูลที่นำมาใช้ การทำงานของโอเปเรชันภายในคลาสและความเหมาะสมของโอเปเรชันของคลาส เป็นต้น ดังนั้นงานสำหรับขั้นตอนที่ 610 มีดังนี้

งานของขั้นตอน 610

งานที่ 10 กระจายโอเปเรชันและแอตทริบิวต์ที่มีอยู่ในแต่ละคลาสออกเพื่อสามารถเห็นการทำงานได้ชัดเจนขึ้น ซึ่งเป็นสิ่งสำคัญ เพื่อแสดงว่าอัลกอริทึมที่อิมพลีเมนต์กับฟังก์ชันนั้นตรงกันและควรกำหนดค่าเริ่มต้นให้กับฟังก์ชันให้ถูกต้องก่อนใช้งาน การกระจายโอเปเรชันและแอตทริบิวต์ของคลาสสามารถแสดงได้ดังรูปที่ 5.29



รูปที่ 5.29 แสดงองค์ประกอบของคลาสที่ประกอบด้วยโอเปเรชันและแอตทริบิวต์

งานที่ 20 พิจารณาอัลกอริทึมในการอิมพลีเมนต์ควรคำนึงถึง ความซับซ้อนในการคำนวณของฟังก์ชัน ความง่ายต่อการทำความเข้าใจและอิมพลีเมนต์ มีความยืดหยุ่นและสามารถปรับปรุงได้ง่ายกับวัตถุโมเดล เป็นต้น

งานที่ 30 เลือกโครงสร้างข้อมูลที่ต้องนำไปใส่ในอัลกอริทึมของโปรแกรม ซึ่งในการอิมพลีเมนต์โครงสร้างข้อมูลผู้ออกแบบอาจต้องเลือกวิธีการนำเสนอในโปรแกรม เช่น อาร์เรย์ ลิสต์ คิว หรือ สแต็ก เป็นต้น หลังจากที่เลือกวิธีการให้กับโปรแกรมแล้ว จำเป็นต้องกำหนดหน้าที่ของโอเปเรชันที่เกิดขึ้นกับแต่ละวัตถุให้ชัดเจน บางครั้งจำเป็นต้องทำลงไปถึงระดับล่าง (Low-level) ของคลาสนั้น การทำลักษณะเช่นนี้อาจทำให้เกิดคลาสใหม่ให้กับระบบซึ่งไม่เคยเกิดขึ้นกับขั้นตอนการวิเคราะห์มาก่อน การกำหนดโอเปเรชันของคลาสนั้นผู้ออกแบบควรกำหนดกฎเกณฑ์สำหรับโอเปเรชันที่เกิดขึ้นในแต่ละวัตถุซึ่งอาจทำในลักษณะของแบบสอบถามให้กับวัตถุแต่ละตัว เพราะระหว่างการอิมพลีเมนต์นั้นแบบสอบถามเหล่านี้จะมีประโยชน์มากสำหรับการเขียนโปรแกรมให้กับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ขั้นตอน 620 การควบคุมปฏิสัมพันธ์ภายนอก (Implement control for external interactions)

จากระยะที่ 3 ที่ได้นำเสนอเอนติตีไลฟ์ฮีสทรีไดอะแกรม ผู้ออกแบบสามารถนำมาวางแผนสำหรับการอิมพลีเมนต์โดยใช้ สเตท-อีเวนท์โมเดลเข้ามาประกอบการพิจารณา ในส่วนของการออกแบบระบบจำเป็นต้องเลือกจากแผนงานที่เกิดขึ้นจริงซึ่งระหว่างการออกแบบวัตถุเราอาจต้องขจัดบางแผนงานออกไป เอนติตีไลฟ์ฮีสทรีไดอะแกรมสามารถนำมาเปลี่ยนแปลงไปสู่ขั้นตอนการทำงานของโปรแกรมในระบบหรือที่เรียกว่า "Procedure-driven System" โดยปกติการควบคุมโปรแกรมจะถูกนำเสนอในรูปแบบทั่วไป ซึ่งจะถูกกำหนดโดยสเตทของโปรแกรมในทุกไฟไนท์สเตทแมชีนเพื่อแปลงไปสู่โค้ดของโปรแกรม และทุกสเตททรานส์ซิชั่นสามารถนำไปสู่สเตทของการอินพุตค่าของข้อมูลได้ หลังจากอ่านค่าอินพุตโปรแกรมจะดำเนินการต่อโดยขึ้นอยู่กับอินพุตที่เข้ามา ซึ่งค่าที่รับเข้ามานั้นโปรแกรมจะทราบเพียงแต่ว่ามีค่าผ่านเข้าออกและรับรู้ในการรับและส่งค่าเท่านั้น โดยบางกรณีอาจเป็นเพียงการตรวจเช็คเงื่อนไขของโปรแกรมอีกครั้งหนึ่ง ซึ่งมีวิธีการเปลี่ยนแปลงเอนติตีไลฟ์ฮีสทรีไดอะแกรมไปสู่โค้ดของโปรแกรมโดยมีงานดังต่อไปนี้

งานของขั้นตอน 620

งานที่ 10 กำหนดเส้นทางเริ่มต้นของการควบคุมโปรแกรม เริ่มพิจารณาจากสเตทเริ่มต้นเพื่อกำหนดเส้นทางการทำงานของสเตทไดอะแกรมที่อีเวนท์นั้นรับผิดชอบอยู่จากนั้นให้เขียนชื่อของสเตทตามลำดับการทำงานที่เกิดขึ้นจริง เพื่อนำไปสู่ขั้นตอนและลำดับการทำงานจริงของโปรแกรม

งานที่ 20 พิจารณาสเตทที่เส้นทางในสเตทไดอะแกรมถูกแยกออกจากสเตทหลัก เพื่อกำหนดเงื่อนไขให้กับโปรแกรม

งานที่ 30 พิจารณาสเตทที่มีลักษณะการทำงานแบบลูป (loop) ในสเตทไดอะแกรม เพื่อนำไปเลือกวิธีการใช้ลูปในโปรแกรม โดยการนำสเตทไดอะแกรมมาพิจารณาเพื่อให้สามารถเลือกใช้ลูปได้ง่ายขึ้น เพราะถ้าการวนกลับของการทำงานในแต่ละสเตทที่มีลักษณะการทำงานแบบซ้ำซ้อนหลาย ๆ ชั้นสามารถที่จะพิจารณาได้ง่ายกว่าเนื่องจากว่าผู้วิเคราะห์สามารถแยกลูปต่าง ๆ เหล่านั้นได้จากการสังเกตสเตทต้นทางและปลายทางตามเส้นทางของสเตททรานส์ซิชั่นที่เกี่ยวข้องเพื่อแยกลูปแต่ละอันออกเป็นชั้น ๆ แทนที่จะเหมือนการเขียนแผนผังงานแบบเดิม เพราะในสเตทจะรวมคำสั่งหรือขั้นตอนต่าง ๆ ไว้ในที่เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 40 ตรวจสอบสเตทและทรานส์ซิชั่นที่ไม่ใช่ส่วนของเงื่อนไข สามารถที่จะนำเทคนิคของภาษาต่าง ๆ ที่เลือกใช้อิมพลีเมนต์มาช่วยจัดการทางด้านการตรวจสอบความถูกต้องของโครงสร้างภาษานั้น ๆ ซึ่งส่วนใหญ่จะนำมาใช้กับการตรวจสอบเงื่อนไขที่มีการเปรียบเทียบและการวนลูปของโปรแกรม โดยการนำมาใช้นั้นควรจะคำนึงถึงความสามารถของภาษานั้น ๆ ด้วย

3) ขั้นตอน 630 ปรับโครงสร้างของคลาสเพื่อเพิ่มการสืบทอดคุณสมบัติ (Adjust class structure to increase inheritance)

การสืบทอดคุณสมบัติของคลาสที่เกิดขึ้นจากขั้นตอนการวิเคราะห์พอมาถึงขั้นตอนการออกแบบผู้ออกแบบควรที่นำไดอะแกรมเหล่านั้นกลับมาพิจารณาถึงความถูกต้องและเหมาะสมอีกครั้งซึ่งการสืบทอดคุณสมบัติของคลาสนั้นเป็นคุณสมบัติและความจำเป็นอย่างหนึ่งของวิธีการนี้ก็ตาม แต่ควรที่จะนำกลับมาทบทวนอีกครั้งเพื่อให้ไดอะแกรมมีความถูกต้องและเหมาะสมก่อนที่จะนำไปอิมพลีเมนต์ สิ่งแรกที่ควรนำมาพิจารณาคือคลาสและโอเปเรชั่นที่เกิดจากการสืบทอด ซึ่งการเปลี่ยนแปลงคลาสและโอเปเรชั่นมีวิธีพิจารณาเพื่อให้คลาสที่มาจากการสืบทอดนั้นมีความเหมาะสม โดยพิจารณาถึงความเป็นไปได้ดังนี้

- บางโอเปเรชั่นในคลาสมีอาร์กิวเมนต์ (Arguments) ที่ไม่เท่ากันเพราะเนื่องจากมีลักษณะการทำงานที่แตกต่างกันออกไปถูกต้องหรือไม่ (ซึ่งสามารถที่จะเพิ่มเข้าไปภายหลังได้แต่ในที่นี้เราเลือกไม่เพิ่มลงไป) เช่นการโอเปเรชั่นของการแสดงภาพบนจอภาพขาวดำกับภาพสี เป็นต้น จะพารามิเตอร์มีที่ต่างกันอยู่หนึ่ง คือ “Color” ซึ่งพารามิเตอร์นี้ไม่จำเป็นต้องมีก็ได้เพราะหน้าที่การทำงานเหมือนกันเพียงแต่ว่าแสดงผลกับจอภาพที่แตกต่างกัน
- บางโอเปเรชั่นอาจจมีอาร์กิวเมนต์น้อยเพราะเป็นกรณีเฉพาะที่แตกต่างไปจากอาร์กิวเมนต์ทั่วไปใช่หรือไม่
- สามารถกำหนดแอตทริบิวต์ที่คล้ายกันให้อยู่ในคลาสที่ต่างกันและอาจให้ชื่อที่ต่างกันได้เพื่อทำให้การทำงานของโอเปเรชั่นง่ายต่อการเข้าถึงแอตทริบิวต์นั้นและสนับสนุนเทคนิคการปกปิด (Information hiding) ใช่หรือไม่
- โอเปเรชั่นหนึ่งอาจกำหนดได้ในหลายคลาสแต่เลือกที่จะกำหนดไว้ที่คลาสแม่เพียงคลาสเดียวตรงตามวัตถุประสงค์หรือไม่

จากหัวข้อที่กล่าวมานี้เป็นเพียงคำถามที่ผู้ออกแบบจะต้องคำนึงถึงว่าคลาสและโอเปเรชั่นที่เกิดจากการสืบทอดนั้นตรงตามวัตถุประสงค์หรือไม่ หลังจากนั้นสิ่งที่ต้องพิจารณาต่อไปคือพฤติกรรมของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสซึ่งควรเป็นพฤติกรรมแบบทั่วไป (Common Behavior) เพื่อสามารถที่จะขนถ่ายพฤติกรรมนี้ให้กับคลาสลูกและนำไปใช้งานได้ ซึ่งการขนถ่ายในทางวัตถุโมเดลเรียกว่า "abstracting out" บางครั้งการกำหนดพฤติกรรมทั่วไปให้กับคลาสหลักผู้ออกแบบอาจจะมีความสับสนถึงการนำไปใช้งานของคลาสลูกซึ่งในความเป็นจริงแล้วพฤติกรรมเหล่านี้จะสามารถถูกเรียกใช้งานได้ทันทีในทุก ๆ วัตถุที่สืบทอดคุณสมบัติมา สุดท้ายพิจารณาถึงความสัมพันธ์ระหว่างคลาส 2 คลาสที่มีลักษณะของการสืบทอดเพราะบางครั้งไดอะแกรมที่เกิดจากขั้นตอนวิเคราะห์อาจเหมาะสมที่จะทำเช่นนั้น แต่หลังจากนำมาพิจารณาใหม่พบว่าบางครั้งการสืบทอดไม่จำเป็นสำหรับคลาสนั้นอาจเปลี่ยนเป็นความสัมพันธ์แบบอื่น เช่น กรณีที่คลาสสแต็กสืบทอดคุณสมบัติมาจากคลาสลิสต์ ซึ่ง ณ ขณะนั้นอาจเข้าใจว่าคลาสสแต็กเป็นคลาสที่เหมาะจะทำเช่นนี้ แต่นำกลับมาพิจารณาใหม่พบว่าการทำเช่นนั้นไม่เป็นผลดีเลย เพราะโอเปเรชั่นที่เกิดขึ้นในคลาสลิสต์ไม่ได้มีส่วนกระทบต่อคลาสสแต็กเลย เมื่อมีการเพิ่มค่าหรือลบค่าในคลาสสแต็กและคลาสลิสต์ไม่ได้เรียกโอเปเรชั่นใด ๆ เลย ในทางกลับกันสิ่งที่อยู่ในคลาสลิสต์ก็ไม่ได้ใช้โอเปเรชัน pop และ push เช่นกัน ดังนั้นควรเปลี่ยนความสัมพันธ์จากการสืบทอดคุณสมบัติไปเป็นความสัมพันธ์แบบ is-a-part-of แทนและทำให้ทั้ง 2 คลาสมีการเรียกใช้โอเปเรชันระหว่างกันได้ เป็นต้น

จากที่กล่าวมาข้างต้นการพิจารณาการสืบทอดคุณสมบัติของคลาสเป็นสิ่งที่ผู้ออกแบบควรจำเป็นอย่างยิ่งเพราะทำให้ไดอะแกรมนั้นมีความถูกต้องและเหมาะสมมากยิ่งขึ้น นอกจากนี้ยังทำให้การนำไปสู่ขั้นตอนการอิมพลีเมนต์มีความชัดเจนและไม่ซับซ้อนอีกด้วย

4) ขั้นตอน 640 ออกแบบแอสโซซิเอชั่น (Design asscociations)

ความสัมพันธ์ของวัตถุเปรียบเสมือน "กาว" ที่คอยเชื่อมระหว่างสองวัตถุ ซึ่งแนวคิดของความสัมพันธ์เป็นแบบเดียวกับการเชื่อมความสัมพันธ์ของเอนติตีที่มักพบได้จากโมเดลและการวิเคราะห์ ซึ่งระหว่างการออกแบบควรที่กำหนดแนวทางให้กับวัตถุในระบบโดยต้องเลือกวิธีการอิมพลีเมนต์ของความสัมพันธ์ ให้เหมาะสมที่สุด ส่วนใหญ่ความสัมพันธ์ของวัตถุมี 2 ชนิดคือ สัมพันธ์แบบทางเดียวกับสองทาง ดังกล่าวต่อไปนี้

ความสัมพันธ์แบบทางเดียว คือ การอ้างอิงความสัมพันธ์ของวัตถุเพียงข้างเดียว ซึ่งจะอิมพลีเมนต์ในรูปของพอยน์เตอร์ กล่าวคือ ถ้ามีความสัมพันธ์แบบ 1:1 จะใช้ในรูปของพอยน์เตอร์ แต่ถ้ามีความสัมพันธ์แบบ 1:M แอตทริบิวต์ฝั่ง "one" จะไปเกาะที่ฝั่ง "many" โดยจะอิมพลีเมนต์ในรูปแบบเซ็ตของพอยน์เตอร์ ซึ่งถ้ามีความสัมพันแบบ 1:M มาเท่าไหร่ จำนวนเซ็ตของพอยน์เตอร์ย่อมมีมากตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์แบบสองทาง คือ การอ้างอิงระหว่างวัตถุซึ่งกันและกัน ซึ่งมีทางเลือกสำหรับการอิมพลีเมนต์ 3 วิธีกล่าวคือ

- การอิมพลีเมนต์แบบอ้างอิงทางเดียวเหมือนกับความสัมพันธ์แบบทางเดียว โดยให้เหตุผลว่า การอ้างอิงทั้งสองทางนั้น ไม่มีโอกาสเกิดขึ้นบ่อยครั้งนัก ถ้ามีความต้องการจะย้อนกลับก็ให้ทำเป็นกรณีไป ซึ่งวิธีนี้มีข้อดีคือ ใช้เนื้อที่ของฮาร์ดดิสก์และค่าใช้จ่ายในการบำรุงรักษาน้อยแต่ถ้ามีการอ้างอิงทั้งสองทางบ่อยครั้งเท่าไหร่ย่อมทำให้ค่าใช้จ่ายเพิ่มขึ้นมากกว่าทุกวิธี
- การอิมพลีเมนต์แบบสองทาง ใช้เทคนิคแบบ Outline คือวัตถุที่อ้างอิงระหว่างกันจะมีเซ็ตของพอยน์เตอร์ชี้ถึงกันและกัน ซึ่งทำให้การเข้าถึงข้อมูลนั้นมีความรวดเร็วขึ้น เพราะถ้ามีการอัพเดทข้อมูลในแต่ละแอตทริบิวต์ในวัตถุจะเก็บลิงค์ของพอยน์เตอร์ที่มาทำการอัพเดทไว้ด้วย เพื่อให้ทราบภายหลังว่าถูกอัพเดทโดยใคร วิธีนี้มีข้อดีคือการอัพเดทสามารถทำได้สะดวกและเหมาะกับข้อมูลจำนวนมากที่ต้องการอัพเดท
- การอิมพลีเมนต์ในแบบแฮชชิ่งฟังก์ชันเป็นวิธีที่แตกต่างไปจากพอยน์เตอร์แบบที่กล่าวมาข้างต้นคือ ในลิสต์จะเก็บค่าของความสัมพันธ์แบบเซ็ตของคู่ลำดับซึ่งจะทำให้มีความสะดวกในการหาข้อมูลได้เร็วกว่าวิธีเดิม แต่วิธีนี้มีข้อจำกัดคือไม่สามารถที่จะเพิ่มเติม อัพเดทหรือแก้ไข ข้อมูลได้สะดวกนัก

การเชื่อมแอตทริบิวต์เป็นอีกแห่งหนึ่งที่จะต้องคำนึงถึงเพราะการอ้างอิงถึงกันของวัตถุยังต้องอาศัยหลักความสัมพันธ์ของอี อาร์ ไดอะแกรม คือ ถ้าวัตถุใดเป็นแบบ M:N ให้กระจายออกมาอีกหนึ่งวัตถุใหม่เสมอ ถ้าเป็นแบบ 1:1 หรือ 1:M สามารถที่จะผ่านขั้นตอนนี้ไปได้

5) ขั้นตอน 650 ค้นหาเส้นทางการเข้าถึงข้อมูลที่เหมาะสม (Optimize access path for data)

พื้นฐานของการออกแบบโมเดลมาจากโมเดลวิเคราะห์เพื่อไปสู่การอิมพลีเมนต์ซึ่งโมเดลวิเคราะห์เป็นเพียงการนำเสนอแนวคิดทางตรรกะของระบบ แต่ขณะที่โมเดลออกแบบนำเสนอรายละเอียดของโมเดลเพื่อที่จะสนับสนุนทางการเข้าถึงข้อมูลในระดับกายภาพได้มากกว่า ดังนั้นระหว่างการออกแบบโดยเฉพาะขั้นตอนนี้นักออกแบบควรที่จะนำโมเดลที่ได้จากการวิเคราะห์มาปรับปรุงซึ่งขั้นตอนการปรับปรุงนั้นควรทำดังนี้

- ลดความซับซ้อนที่เกิดขึ้นกับการเข้าถึงข้อมูลของคลาสให้น้อยที่สุดเท่าที่จะทำได้
- ปรับปรุงประสิทธิภาพลำดับการทำงานของฟังก์ชันนัลโมเดล (อัลกอริทึม)
- จัดเก็บข้อมูลหรือแอตทริบิวต์ในแต่ละวัตถุที่มีการเปลี่ยนแปลงบ่อยให้มีความชัดเจนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลดความซับซ้อนของการเข้าถึงข้อมูลในแต่ละคลาสนั้นในบางครั้งความสัมพันธ์ของคลาสที่เกิดขึ้นอาจจะมีความสัมพันธ์ที่ซับซ้อน เช่น ต้องการหาความสามารถพิเศษทางภาษาของพนักงานในบริษัท ซึ่งประกอบด้วย คลาสบริษัท คลาสพนักงาน คลาสความสามารถพิเศษ โดยความสัมพันธ์ที่มีสามารถแสดงได้ดังรูปที่ 5.30 ซึ่งถ้าต้องการหาความสามารถพิเศษทางภาษาญี่ปุ่นของพนักงานในบริษัทต้องเกี่ยวข้องกันทั้งสามคลาส ซึ่งถ้ามีการค้นหาจากพนักงานจำนวนน้อยก็จะไม่เกิดปัญหาในเรื่องเวลา แต่ถ้ามีพนักงานจำนวนมากประมาณ 1000 คนและทุกคนก็มีความสามารถพิเศษทางภาษาด้วยกันหมด ซึ่งจำเป็นต้องทำการค้นหาจากพนักงานทั้งหมดและทำให้การค้นหานั้นมีความล่าช้าขึ้น

รูปที่ 5.30 แสดงความสัมพันธ์แบบลูกโซ่ของคลาสบริษัท พนักงานและความสามารถพิเศษ

ดังนั้นทางแก้ไข คือ ควรยุบคลาสความสามารถพิเศษ จากนั้นเพิ่มอินเด็กซ์ให้กับคลาสบริษัทซึ่งทำให้การเข้าถึงนั้นง่าย รวดเร็วและกระชับมากขึ้นดังแสดงในรูปที่ 5.31 การยุบคลาสและความสัมพันธ์ที่เกิดขึ้นนั้นผู้ออกแบบควรที่จะคำนึงถึงความจำเป็นและความเหมาะสมของระบบงานเป็นหลัก โดยอาจจะต้องตรวจสอบจากความต้องการของผู้ใช้และลักษณะการทำงานของโอเปเรชั่นของคลาสเป็นหลักด้วย

รูปที่ 5.31 แสดงอินเด็กซ์ความสามารถพิเศษให้กับคลาสพนักงาน

การปรับปรุงประสิทธิภาพลำดับการทำงานของฟังก์ชันนัลโมเดลหรืออัลกอริทึมนั้น คือ การปรับปรุงการทำงานของอัลกอริทึมให้มีลำดับการทำงานที่กระชับและรวดเร็วและเหมาะสมกับวัตถุต่าง ๆ ที่จะนำไปใช้ โดยอาจทำการทดสอบทั้งกรณีที่ดีที่สุดและกรณีที่แย่ที่สุด เพื่อหาความเหมาะสมกับงานที่นำไปใช้ เช่น การหาความสามารถทางภาษาของพนักงานซึ่งถ้าต้องการหาพนักงานที่สามารถพูดภาษาญี่ปุ่นและฝรั่งเศสได้ จากพนักงานทั้งหมด 1000 คน แต่มีเพียง 5 คนสามารถพูดภาษาญี่ปุ่น และ 100 คนสามารถพูดฝรั่งเศสได้ ซึ่งการค้นหาอาจหาจากผู้ที่พูดภาษาญี่ปุ่นได้ ก่อนจากนั้นจึงย้อนหาคนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถพูดฝรั่งเศสและนำผลลัพธ์ทั้งสองมารวมกันเพื่อหาคนที่พูดได้ทั้ง 2 ภาษา แต่ถ้าในทางที่แย่ที่สุดคือไม่มีพนักงานที่พูดได้ทั้ง 2 ภาษาเลยเป็นต้น

การจัดการแอตทริบิวต์ที่อยู่ในแต่ละวัตถุ ซึ่งแอตทริบิวต์นั้นถือเป็นข้อมูลอย่างหนึ่งในแต่ละวัตถุ ซึ่งอาจมีความซับซ้อนเกิดขึ้นในแต่ละวัตถุ ดังนั้นผู้ออกแบบควรจัดเก็บข้อมูลเหล่านี้ให้อยู่ในรูปของฐานข้อมูล โดยการจัดเก็บนั้นควรคำนึงถึงการนำไปใช้งาน เพราะการใช้งานแต่ละครั้งค่าที่อยู่ภายในสามารถเกิดการเปลี่ยนแปลงได้ตลอดเวลา วิธีการปรับปรุงค่าของข้อมูลควรคำนึงถึงหลักการดังนี้ ความถูกต้องในการปรับปรุง (by explicit code) ช่วงเวลาในการปรับปรุง (by periodic recomputation) และค่าที่ถูกใช้งาน (by using active values) วิธีแรก พิจารณาถึงความถูกต้องในการปรับปรุงค่าของแอตทริบิวต์ในแต่ละวัตถุผู้ออกแบบควรเพิ่มโค้ดของโปรแกรมสำหรับโอเปเรชั่นในการอัพเดทในแต่ละวัตถุที่แอตทริบิวท์นั้นเกิดการเปลี่ยนแปลงให้ชัดเจนและตรงกับวัตถุที่รับผิดชอบ ต่อมาพิจารณาถึงความบ่อยในการปรับปรุงค่าของแอตทริบิวต์ในแต่ละวัตถุ (ความถี่ในการปรับปรุง) กลุ่มของค่าพื้นฐานที่มีการปรับปรุงอยู่บ่อย ๆ เป็นอีกตัวอย่างหนึ่งที่ควรที่จะคำนึงถึงเพราะบางครั้งกลุ่มของแอตทริบิวท์ที่ได้รับมาจากวัตถุต่าง ๆ ปราศจากปรับปรุงค่าก่อนส่งออกมาทั้ง ๆ ที่ค่าเหล่านี้ควรทำให้เป็นค่าปัจจุบันก่อน ที่เป็นเช่นนี้เพราะกลุ่มของแอตทริบิวต์เหล่านี้กระจายอยู่ในหลายวัตถุจึงทำให้การปรับปรุงค่านั้นมีผลกระทบมากขึ้นโดยจะต้องไล่ตามไปปรับปรุงทุก ๆ ที่ที่มีค่านั้นอยู่ ดังนั้นวิธีการปรับปรุงแบบช่วงเวลาจึงเป็นวิธีที่ชัดเจนและมีข้อบกพร่องน้อยที่สุด แต่วิธีนี้ยังมีข้อบกพร่องอยู่ในกรณีที่ ณ เวลาใด ๆ กลุ่มของดาต้าที่มีเกิดการเปลี่ยนแปลงน้อยครั้งวิธีนี้จะละเลยหน้าที่ไปเพราะจะสนับสนุนกับกลุ่มแอตทริบิวต์ที่มีขนาดใหญ่เท่านั้น วิธีการสุดท้ายพิจารณาถึงค่าที่ถูกใช้งาน ณ ปัจจุบัน กล่าวคือ ค่าที่ถูกนำไปใช้งานนั้นควรเป็นค่าที่มีมูลค่าของค่านั้น ณ การใช้งานจริงในแต่ละครั้ง ซึ่งการกำหนดค่าเริ่มต้นให้กับสเตทไดอะแกรมผู้ออกแบบควรที่จะพิจารณาถึงความเหมาะสมกับโปรแกรมที่นำไปใช้งานด้วย

6) ขั้นตอน 660 ระบุการแสดงตัวแทนของวัตถุ (Determine Object Representation)

การอิมพลีเมนต์ของวัตถุจะถูกดำเนินต่อเนื่องไปเรื่อย ๆ ฉะนั้นผู้ออกแบบควรต้องพิจารณาถึงชนิดของแอตทริบิวต์ของออกเจ็กบ้างเพราะถ้ากำหนดชนิดของแอตทริบิวต์ผิดไปจากจุดประสงค์อาจจะมีผลกระทบในระยะยาว ซึ่งชนิดของแอตทริบิวต์มีอยู่หลายประเภทคือ ตัวอักษร(String) เลขจำนวนเต็ม (Integer) และ เลขจำนวนจริง (Real) เช่น วัตถุของพนักงานที่มีแอตทริบิวต์หมายเลขประกันสังคม ซึ่งเราสามารถกำหนดได้ทั้งแบบตัวอักษรหรือเลขจำนวนเต็ม ถ้ากำหนดเป็นแบบตัวอักษรก็ไม่สามารถนำไปคำนวณได้ แต่ถ้ากำหนดเป็นแบบเลขจำนวนเต็มสามารถที่จะนำไปคำนวณได้ เช่น การกำหนดหมายเลขประกันสังคมส่วนใหญ่ถูกจัดเก็บเป็นตัวอักษรเท่านั้น เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) ขั้นตอน 670 การตรวจสอบการออกแบบเชิงวัตถุ (Object Metrics)

การตรวจสอบเชิงวัตถุมีความแตกต่างจากการตรวจสอบของการวิเคราะห์เชิงวัตถุ ซึ่งการตรวจสอบของกิจกรรมนี้จะใช้เครื่องมือสำหรับการตรวจสอบหรือที่เรียกว่า "Object-Oriented Metrics" โดยการตรวจสอบนั้นยึดหลักของวิศวกรรมซอฟต์แวร์เป็นหลัก กล่าวคือ ผู้ออกแบบตรวจสอบทั้งในระดับความสัมพันธ์ระหว่างวัตถุ/คลาส (Coupling) และระดับความสัมพันธ์ของฟังก์ชันในวัตถุ/คลาส (Cohesion) ซึ่งความสัมพันธ์ของทั้งสองระดับนี้สามารถทำให้ผู้ออกแบบสามารถทราบถึงความซับซ้อนของการออกแบบ (Complexcity) ได้ และวิธีการวัดความสัมพันธ์ที่เกิดขึ้นของการออกแบบเชิงวัตถุนั้นมีอยู่หลายวิธีการ แต่ในโมเดลนี้วิธีการวัดการออกแบบเชิงวัตถุใช้หลายวิธีมารวมกันซึ่งได้แก่ วิธีของ Chidamber และ Kemerer [9] วิธีของ Larez และ Kidd [33] วิธีของ Henery และ Kafura[21] และวิธีของ Card และ Glass [8] เป็นต้น โดยวิธีการตรวจสอบของโมเดลประยุกต์มีขั้นตอนดังนี้

งานของขั้นตอน 670

งานที่ 10 หาขนาดของวัตถุภายในไดอะแกรมตามวิธีของ CK Metrics Suties[9] ซึ่งการหาขนาดของวัตถุพร้อมกับหาความซับซ้อนในแต่ละระดับของวัตถุที่สนใจ โดยมีสูตรการคำนวณหาขนาดมีดังนี้

$$WMC = \sum c_i$$

ให้ C คือคลาสที่สนใจ และมีจำนวนของเมธอดเท่ากับ 1-n เมธอดของคลาสที่สนใจ และค่ารวมของความซับซ้อนของทั้งคลาสมีค่าเท่ากับ 1 เท่านั้น หลังจากหาจำนวนรวมของเมธอดในแต่ละคลาสเรียบร้อยแล้ว

งานที่ 20 วัดความลึกของการสืบทอดคุณสมบัติ (DIT- Depth of the Inheritance Tree) คือการวัดความลึกจาก Root ถึง Child โดยปกติทั่วไปความลึกของการสืบทอดควรไม่เกิน 7 ± 2 ชั้น เพราะถ้ามีความลึกของความสัมพันธ์มากย่อมทำให้เกิดความซับซ้อนมากขึ้นเท่านั้น

งานที่ 30 วัดนับจำนวนลูกของคลาส (NOC-Number of Children) คือนับจำนวนของคลาสลูกที่อยู่ใน Root เดียวกัน ซึ่งถ้าใน root มีคลาสลูกมากเท่าใด ย่อมจะทำให้เกิดความซับซ้อนมาขึ้นตามลำดับ

งานที่ 40 วัดความสัมพันธ์ระหว่างคลาส (CBO – Coupling Between Object Classes) การวัดความสัมพันธ์ระหว่างคลาสที่เกิดขึ้นในการออกแบบเป็นสิ่งที่วัดได้ยากอย่างหนึ่ง ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีของ CK ได้แนะนำว่า ให้วัดจาก CRC ซึ่งเป็นเอกสารกำกับสำหรับการออกแบบคลาสที่เกี่ยวข้อง ซึ่งภายใน CRC กล่าวถึงรายละเอียดต่าง ๆ ของคลาสเหล่านั้นไว้

งานที่ 50 วัดความสัมพันธ์ของคลาส (RFC-Response for a Class) คือ กลุ่มของความสัมพันธ์ของเมธอดภายในวัตถุของคลาสเดียวกัน กล่าวคือ ความสัมพันธ์ของเมธอดที่กิดขึ้นจากการเรียกเมธอดจากวัตถุโดยใช้เมจเซส ดังนั้น RFC จึงนับเฉพาะจำนวนของเมธอดที่มีการเรียกใช้ในวัตถุภายในคลาสเดียวกันการหาความสัมพันธ์ของคลาสของงานที่ 5 นี้ สามารถนำวิธีการวัดของ Henery และ Kafura[21] กับ วิธีของ Card และ Glass [8] สูตรของ Henery และ Kafura มีดังนี้

1) $fan-in \times fan-out$

2) $(fan-in \times fan-out)^2$

3) $S_s \times (fan-in \times fan-out)^2$

ในวิธีการคำนวณแบบที่ 3 จะนำขนาดของ LOC ซึ่งแทนด้วย S_s ขณะที่วิธีของ Card กับ Glass[8] ให้ความคิดเห็นว่าขนาดของ LOC นั้นไม่มีความจำเป็นต่อการนำมาวัดค่าความซับซ้อนจึงนำค่า fan-out มาวัดเพียงอย่างเดียวซึ่งสามารถคำนวณได้ดังนี้

$$\overline{C}_I \equiv \frac{C_I}{n} = \frac{\sum^n (fan-out)^2}{n}$$

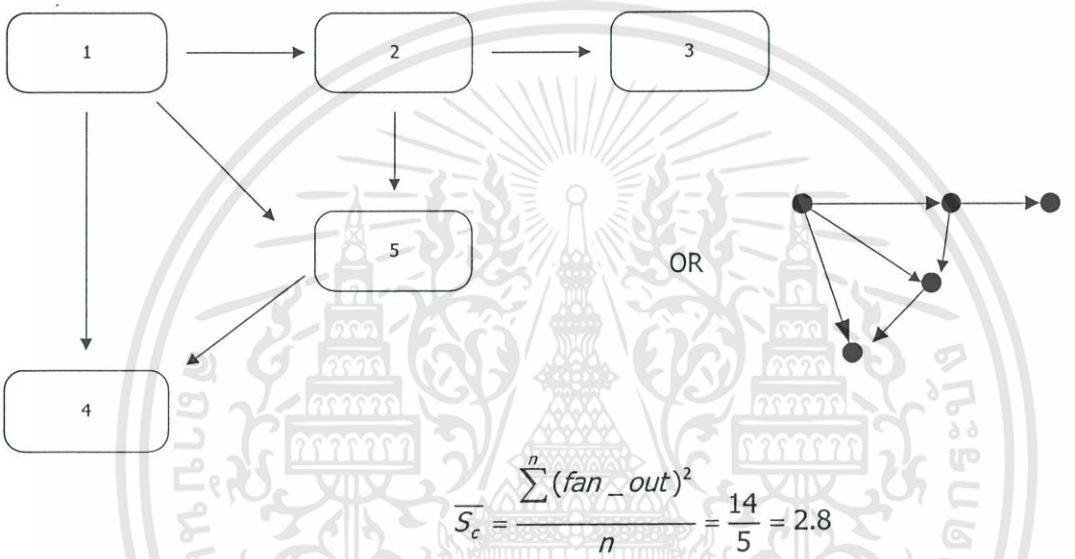
หรือ

$$\overline{S}_C = \frac{\sum^n (fan-out)^2}{n}$$

การวัดค่าความซับซ้อนเชิงโครงสร้างของโปรแกรมเชิงวัตถุด้วยการใช้ค่าจาก fan-in และ fan-out นั้น มีค่ามาตรฐานของการวัด กล่าวคือ ถ้าหากผลลัพธ์จากการคำนวณจากสูตรปรากฏว่า ค่า fan-out มีค่าในระดับต่ำ และค่า fan-in มีค่าในระดับสูง แสดงว่าการออกแบบคลาสและความสัมพันธ์กับคลาสอื่น ๆ อยู่ในเกณฑ์ที่ดี ในทางตรงกันข้าม หากค่า fan-out และ fan-in มีค่าในระดับสูงเหมือนกันแสดงว่าการออกแบบคลาสนั้นควรนำกลับไปสู่ขั้นตอนการออกแบบเสียใหม่[8] ซึ่งวิธีการคำนวณสามารถแสดงดังในรูปที่ 5.32 สิ่งที่สำคัญอีกประการหนึ่งที่สามารถแสดงถึงความซับซ้อนของความสัมพันธ์ที่เกิดขึ้นกันระหว่างคลาสได้ คือ ลำดับของการสืบทอดคุณสมบัติจากคลาสหลักไม่ควรเกิน 7 ± 2 ชั้น เพราะว่าความลึกและค่าเฉลี่ยนั้นเป็นประโยชน์ต่อการพัฒนาระบบได้ ซึ่งสามารถแสดงถึงอัตราส่วนของการนำกลับมาใช้ใหม่กับลักษณะเฉพาะของคลาสย่อยและคลาสหลักโดยอัตราส่วนของการนำกลับมาใช้ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอกถึงจำนวนของคลาสหลักทั้งหมดและจำนวนของการนำมาใช้ของระบบ ส่วนอัตราส่วนของลักษณะเฉพาะแสดงถึงจำนวนของคลาสย่อยที่มีต่อคลาสหลักพร้อมกับแสดงถึงลักษณะของ ADT ที่มีอยู่ในคลาสหลัก อย่างไรก็ตามการวัดความซับซ้อนเชิงโครงสร้างของการสืบทอดคุณสมบัติยังมีความจำเป็นอยู่ระหว่างคุณลักษณะเฉพาะทางกับการอิมพลีเมนต์ แม้ว่าภายในสามารถแสดงถึงความเกี่ยวข้องกันระหว่างเซอร์วิสหรือการเข้าถึงโครงสร้างข้อมูลภายในคลาสก็ตาม



รูปที่ 5.32 แสดงการคำนวณ fan-in และ fan-out

งานที่ 60 วัดความสัมพันธ์ระหว่างเมธอด (LCOM-Lack of Cohesion in Methods) คือ วัดความสัมพันธ์ของเมธอดในวัตถุเดียวกัน โดยวัดที่แอตทริบิวต์ของแต่ละวัตถุ วิธีการวัด คือ พิจารณาแอตทริบิวต์ของแต่ละวัตถุโดยนับจากจำนวนของการเรียกใช้โดยเมธอดอื่น กล่าวคือ ถ้าแอตทริบิวต์ใดถูกเมธอดอื่นเข้ามาเรียกใช้ให้นับค่าของ LCOM เพิ่มขึ้นทีละ 1 สำหรับการถูกเรียกใช้แต่ละครั้ง แต่ถ้าแอตทริบิวต์ใดไม่ถูกเรียกใช้ให้ LCOM=0 เช่น คลาส C มี 6 เมธอด และมี 4 เมธอดเรียกใช้แอตทริบิวต์จากเมธอดอื่น ดังนั้น LCOM = 4 เป็นต้น ซึ่งการวัดด้วยขั้นตอนนี้ทำให้ผู้ออกแบบทราบถึงความสัมพันธ์ระหว่าเมธอดในวัตถุได้เป็นอย่างดี และสามารถนำค่าเหล่านี้ไปปรับปรุงการกำหนดแอตทริบิวต์ให้แต่ละวัตถุให้เหมาะสมมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 70 หาจำนวนของโอเปเรชันของคลาสย่อยที่คาบเกี่ยวกัน (NOO - Number of Operations Overriden by a Subclass) คือ การหาจำนวนของโอเปเรชันที่สืบทอดมาจากคลาสหลัก โดยปกติโอเปเรชันในวัตถุจะถูกเรียกใช้งานจากภายในวัตถุเดียวกันหรือวัตถุที่สืบทอดกันมา ซึ่งเรียกโอเปเรชันนี้ว่า "Overriden" ซึ่งถ้าจำนวนของโอเปเรชันแบบ overridden มีเป็นจำนวนมาก ย่อมมีผลกับลำดับชั้นของคลาส เพราะทำให้การปรับปรุงและการทดสอบสำหรับการออกแบบมีความยุ่งยากตามมา ดังนั้นผู้ออกแบบที่ดีควรมีจำนวนของโอเปเรชันประเภทนี้แต่พอเหมาะกับโครงการสารสนเทศที่รับผิดชอบ

งานที่ 80 หาจำนวนของโอเปเรชันของแต่ละคลาสย่อย (NOA – Number of Operations Added by a Subclass) โดยปกติโอเปเรชันและแอตทริบิวต์ที่เกิดขึ้นกับคลาสย่อยมักจะเป็นลักษณะพิเศษที่มีความจำเป็นกับคลาสนั้น ๆ ดังนั้นการหาค่าของ NOA คือ การนับจำนวนของโอเปเรชันกับแอตทริบิวต์ภายในคลาสย่อย จึงทำให้การนับค่าของ NOA ผู้ออกแบบต้องนำค่าอของ DIT ของคลาสมาประกอบการพิจารณา เนื่องจากค่าของ NOA จะอยู่ที่ระดับล่างของคลาสที่เกี่ยวข้อง

งานที่ 90 หาลักษณะของอินเด็กซ์ (SI – Specializtion Index) เป็นวิธีการวัดเพื่อหาความเหมาะสมของจำนวนโอเปเรชัน แอตทริบิวต์และลำดับของคลาสย่อยให้มีความเหมาะสมกับระบบ ซึ่งการวัดมีวิธีการคำนวณดังนี้

$$SI = [NOO \times level]/M_{total}$$

จากสูตร กำหนดให้ Level คือลำดับของคลาส ที่อยู่ในแต่ละคลาส และให้ M_{total} เป็นจำนวนของโอเปเรชันทั้งหมดของคลาส ถ้าค่า SI จากการคำนวณมีค่ามากย่อมแสดงว่าการออกแบบคลาสในระบบมีความไม่สอดคล้องกัน ทางแก้ คือ ผู้ออกแบบควรกลับไปปรับลดโอเปเรชันในคลาสที่สนใจให้มีความกระชับมากขึ้น โอเปเรชันที่สามารถปรับลดควรเป็นโอเปเรชันในลำดับของคลาสเดียวกันและโอเปเรชันลักษณะ Overridden เท่านั้น

งานที่ 100 บันทึกรายละเอียดการวัดของวัตถุแต่วัตถุที่ปรากฏใน O/C ไดอะแกรมพร้อมกับนำค่าที่วัดได้ไปปรับปรุงแก้ไขไดอะแกรมที่มีข้อบกพร่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์และเอกสาร

ตารางที่ 5.13 แสดงรายละเอียดการตรวจสอบการออกแบบเชิงวัตถุ

MetricsTest MT01 Page ___ of ___ **Class/Object ID:** _____ **Class/Object Name:** _____ **Test ID :** _____ **Tester:** _____ **Date Designed :** _____ **Date Conducted :** _____ **Results:** **Object Test :-** ☐ **Passed** ☐ **Redesign** ☐ **Fault** **Method Test :-** ☐ **Passed** ☐ **Redesign** ☐ **Fault**
Objective: _____ _____ _____
Object Test Cases Size of Object : _____ **(Less Than 1)** Depth of the Inheritance Tree (DIT) : _____ **(7±2)** Number Of Children (NOC) : _____ Coupling Between Object Classes (CBO) : _____
Method Test Cases Lack of Cohesion in Methods (LCOM) : _____ Number of Operations Overridden by Subclass (NOO) : _____ Number of Operations Added by a Class (NOA) : _____ Specialization Index (SI) : _____
Expected Results/Notes
Actual Results/Notes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.14 แสดงรายละเอียดการวัดความสัมพันธ์ระหว่างวัตถุ

MetricsTest MT02
Class/Object ID: _____ **Class/Object Name:** _____ **Test ID :** _____ **Tester:** _____ **Date Designed :** _____ **Date Conducted :** _____ **Results:** ☐ **Passed** ☐ **Redesign** ☐ **Fault**
Diagram
Diagram Test Response for a Class (RFC) :- **(Standard Values** _____**)** Fan-in : _____ Fan-out : _____ Average : _____
Expected Results/Notes
Actual Results/Notes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 5.13 แสดงรายละเอียดของการทดสอบการออกแบบเชิงวัตถุ คือ ตารางสำหรับแสดงรายละเอียดของวัตถุโดยแบ่งออกเป็น 2 ส่วน ส่วนแรกวัดขนาดของวัตถุและจำนวนวัตถุที่เกี่ยวข้อง ส่วนสุดท้ายวัดความซับซ้อนความสัมพันธ์ของฟังก์ชันทั้งภายในวัตถุและระหว่างวัตถุ ซึ่งค่าต่าง ๆ ของการวัดทั้งสองส่วนผู้รับผิดชอบโครงการเป็นผู้กำหนดตามความเหมาะสม หรือใช้ค่ามาตรฐาน

ส่วนตารางที่ 5.14 แสดงรายละเอียดการวัดความสัมพันธ์ระหว่างวัตถุ ซึ่งการวัดความสัมพันธ์ระหว่างวัตถุซึ่งวัดทั้งระบบที่รับผิดชอบรวมถึงระบบย่อยและระดับของระบบ (ถ้าวัตถุนั้นมีความสัมพันธ์กัน)

8) ขั้นตอน 680 รวบรวมเอกสารเพื่อการตัดสินใจในการออกแบบ

ขั้นตอนที่ 680 คือ การสรุปการทำงานเปรียบเสมือนการตรวจสอบเอกสารต่าง ๆ ที่เกิดขึ้นของระยะที่ 6 ตั้งแต่ขั้นตอนที่ 610 จนถึงขั้นตอนที่ 670 ซึ่งถ้าเอกสารของแต่ละขั้นนั้นไม่สมบูรณ์หรือไม่ครบถ้วน ผู้วิเคราะห์ควรกลับไปตรวจสอบการทำงานของระยะที่ 6 ตั้งแต่ขั้นตอนที่ 610 จนถึงขั้นตอนที่ 670 อีกครั้ง เอกสารของขั้นตอนนี้มีดังนี้

5.5.4 โมดูลอิมพลีเมนต์และการทดสอบ (Implementation and Testing Module)

โมดูลที่ 4 เป็นโมดูลที่กล่าวถึงการอิมพลีเมนต์ การทดสอบคุณภาพของระบบ และการติดตั้งระบบที่ได้ออกแบบไว้ซึ่งประกอบด้วย 2 ระยะ คือ ระยะที่ 7 และ 8 โดยระยะที่ 7 กล่าวถึงการอิมพลีเมนต์ระบบพร้อมกับการตรวจสอบคุณภาพของระบบที่ได้สร้างเสร็จเรียบร้อยแล้ว การตรวจสอบระบบนั้น ทำการตรวจสอบตั้งแต่ระบบเครือข่าย ฐานข้อมูลและซอฟต์แวร์ที่เกี่ยวข้อง เช่น ระบบปฏิบัติการ แอพพลิเคชันของสำนักงาน (ไมโครซอฟต์ออฟฟิศ) เป็นต้น หลังจากนั้นผู้วิเคราะห์และออกแบบระบบจะส่งรายละเอียดของระบบงานไปสู่การอิมพลีเมนต์ต่อไป ส่วนระยะที่ 8 กล่าวถึงการติดตั้งและการนำไปใช้งานของระบบ ซึ่งการนำไปใช้งานนั้นเป็นการนำไปใช้งานจริงในองค์หรือธุรกิจนั้น ๆ โดยการนำไปใช้งานนั้นผู้พัฒนาระบบต้องเตรียมการและวางแผนการนำไปใช้อย่างละเอียดถี่ถ้วน รายละเอียดของทั้งสองขั้นตอนมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.4.1 ระยะที่ 7 การสร้างระบบ (Construction)

การอิมพลีเมนต์และการทดสอบระบบ ซึ่งประกอบด้วยขั้นตอนย่อย 5 ขั้นตอนคือ

1) ขั้นตอน 710 การสร้างและทดสอบระบบเครือข่าย (ถ้ามี) (Build and Test Networks, if exists)

การสร้างและทดสอบระบบเครือข่ายนั้น ผู้วิเคราะห์และพัฒนาระบบ ควรขอคำปรึกษาจากผู้เชี่ยวชาญทางด้านระบบเครือข่ายและให้เข้ามามีส่วนร่วมในการปฏิบัติงานในขั้นตอนนี้

งานของขั้นตอน 710

งานที่ 10 การสร้างและทดสอบ ซึ่งการทดสอบเริ่มต้นทดสอบจากการเชื่อมต่อของระบบเครือข่ายว่าครอบคลุมพื้นที่ที่ต้องการบริการหรือไม่ เช่น ทดลองเชื่อมต่อระหว่างเครื่องที่อยู่ต่างพื้นที่กัน เพื่อให้ทราบว่า ระบบเครือข่ายนั้นใช้งานได้จริง

งานที่ 20 นำแอพพลิเคชั่นหรือระบบงานจริงมาทดลองใช้งานผ่านระบบเครือข่ายเพื่อเป็นยืนยันว่าระบบนั้นสามารถทำงานได้จริงและถูกต้องตามความต้องการของผู้ใช้

งานที่ 30 วัดประสิทธิภาพของการรับส่งข้อมูลของระบบเครือข่าย ซึ่งอาจจะใช้ซอฟต์แวร์ทดสอบประสิทธิภาพที่เหมาะสม

2) ขั้นตอน 720 การสร้างและทดสอบฐานข้อมูล (Build and Test Database)

การสร้างและทดสอบฐานข้อมูลเป็นอีกขั้นตอนหนึ่งของการอิมพลีเมนต์ระบบ การสร้างฐานข้อมูลผู้ออกแบบจะนำไดอะแกรมของวัตถุที่ออกแบบไว้ มาสร้างฐานข้อมูลลงบนระบบฐานข้อมูลที่เลือกและหลังจากสร้างฐานข้อมูลเสร็จควรทำการทดสอบโครงสร้างฐานข้อมูลเพื่อแสดงถึงความถูกต้องของฐานข้อมูล เช่น การบันทึก การเรียกค่า การปรับปรุงค่า และอื่น ๆ เป็นต้น ตัวอย่างข้อมูลที่นำมาทดสอบนั้น (จำนวนตัวอย่างข้อมูลที่นำมาทดสอบขึ้นอยู่กับดุลพินิจของผู้ออกแบบฐานข้อมูล) อาจเป็นข้อมูลเดิมของระบบ เพราะข้อมูลเดิมสามารถแสดงถึงข้อบกพร่องต่าง ๆ ได้ดี ก่อนที่จะนำฐานข้อมูลที่สร้างใหม่นำไปใช้กับระบบสารสนเทศที่รับผิดชอบ

3) ขั้นตอน 730 การติดตั้งและทดสอบซอฟต์แวร์ (ถ้ามี) (Install and Test New Software Package, if necessary)

ขั้นตอน 730 คือ การติดตั้งและทดสอบซอฟต์แวร์ที่เกี่ยวข้องกับระบบงานหรือซอฟต์แวร์แพ็กเกจที่ต้องทำการจัดซื้อเพื่อมาทำงานร่วมกับระบบที่พัฒนาขึ้น โดยก่อนที่จะนำระบบงานที่พัฒนาขึ้นมาทำงานร่วมกันนั้น ควรทำการติดตั้งและทดสอบเสียก่อนซึ่งมีลำดับของการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานของขั้นตอน 730

งานที่ 10 ตรวจสอบซอฟต์แวร์และเอกสารประกอบ

งานที่ 20 ทำการติดตั้งซอฟต์แวร์

งานที่ 30 ทดลองใช้ซอฟต์แวร์แพ็กเกจเพื่อยืนยันว่าซอฟต์แวร์นั้นสามารถทำงานได้ ซึ่งถ้าระหว่างการทดลองใช้นั้นเกิดปัญหาขึ้นจะได้ทำการปรับปรุงแก้ไขได้ก่อนที่จะเกิดขึ้นกับระบบงานที่ได้พัฒนาขึ้น ซึ่งระหว่างการทดสอบนั้นอาจพบข้อผิดพลาด (Error) ทั้งทางกายและตรรกะ ทางกายภาพ เช่น ความสามารถของฮาร์ดแวร์ต่อซอฟต์แวร์แพ็กเกจ เป็นต้น ส่วนทางด้านตรรกะ คือ ซอฟต์แวร์มีความขัดแย้งกับระบบงานที่พัฒนาขึ้น อาจเป็นความไม่เข้ากันของระบบงาน เช่น การเกิดข้อผิดพลาดระหว่างการใช้งาน (Bug) และการติดตั้ง เป็นต้น (ซึ่งการทดสอบซอฟต์แวร์แพ็กเกจนั้นเจ้าของระบบควรจะต้องแจ้งข้อจำกัดของระบบงานให้ผู้ผลิตซอฟต์แวร์ทราบก่อน)

งานที่ 40 ปรับปรุงผลกระทบของซอฟต์แวร์ที่มีต่อโครงการ ซึ่งผลกระทบนี้จะถูกบันทึกในรูปของเอกสาร ซึ่งจะกล่าวถึงวิธีใช้งานและข้อผิดพลาดที่เกิดขึ้นระหว่างการทดสอบ เพื่อเก็บประวัติของซอฟต์แวร์แพ็กเกจแต่ละตัว

งานที่ 50 บันทึกประวัติผู้ขายเพื่อเก็บไว้เป็นประวัติสำหรับการบำรุงรักษาซอฟต์แวร์แพ็กเกจ

ขั้นตอนที่กล่าวมาข้างต้นเป็นวิธีการติดตั้งและทดสอบซอฟต์แวร์แพ็กเกจซึ่งบางครั้งอาจจะมีวิธีการติดตั้งและทดสอบที่ซับซ้อนหรือง่ายกว่านี้ ซึ่งสิ่งที่ได้นำเสนอไปนั้นเป็นเพียงการตรวจสอบที่มักจะพบเห็นกันทั่วไป

ข้อแนะนำ : การทดสอบซอฟต์แวร์ทางที่ดีควรให้บริษัทที่ขายซอฟต์แวร์นั้นมาติดตั้งและทดสอบให้เพราะหากเกิดข้อผิดพลาดหรือพบปัญหาใด ๆ ก็ตามจะได้ทำการแก้ไขได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์และเอกสาร

ตารางที่ 5.15 แสดงรายละเอียดของติดตั้งโปรแกรม

Installation Report

System Name: _____ Software Version: _____

Officer: _____ Date Conducted :_____

User: _____

Step of Install

_____

_____

_____

_____

_____

Problem Description

_____

_____

_____

_____

_____

Result / Notes

_____

_____

_____

_____

_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 5.15 แสดงรายละเอียดของการติดตั้งซอฟต์แวร์และ/หรือโปรแกรมที่ระบบต้องใช้งาน การติดตั้งซอฟต์แวร์และ/หรือโปรแกรมนั้น ผู้ติดตั้งต้องบันทึกขั้นตอนการติดตั้ง รุ่นของซอฟต์แวร์และ/หรือโปรแกรม ปัญหาของการติดตั้ง (ถ้ามี) ผลการติดตั้ง ผู้ติดตั้ง ผู้รับผิดชอบซอฟต์แวร์และ/หรือโปรแกรมซึ่งผู้รับผิดชอบอาจรวมถึงผู้ใช้งานพร้อมกับคำแนะนำ (ถ้ามี)

4) ขั้นตอน 740 การเขียนและทดสอบโปรแกรม (Coding and Test New Program)

ขั้นตอนที่ 740 เป็นกิจกรรมที่นับว่าเป็นส่วนสำคัญของการพัฒนาระบบงานอีกกิจกรรมหนึ่ง คือ การนำรายละเอียดทั้งหมดที่ได้จากการวิเคราะห์และออกแบบของระยะที่ 3-6 ซึ่งก่อนที่นำไปสู่การเขียนโปรแกรมให้กับระบบ โดยมีการทำงานดังต่อไปนี้

งานของขั้นตอน 740

งานที่ 10 ตรวจสอบรายละเอียดทั้งหมดเริ่มตั้งแต่ อัลกอริทึม ฟังก์ชันการทำงาน ความสัมพันธ์ระหว่างโมดูล แนวทางการแก้ไขปัญหาและความต้องการของผู้ใช้ เพื่อยืนยันความถูกต้องและความแน่นอนของระบบ

งานที่ 20 การเขียนโปรแกรม

งานที่ 30 ตรวจสอบหรือการวัดคุณภาพของระบบที่พัฒนา ซึ่งต่างจากการตรวจสอบของขั้นตอนที่ 670 ที่เป็นการตรวจสอบของวัตถุและคลาส แต่ขั้นตอนนี้จะตรวจสอบทางด้านของการวัดขนาดของโปรแกรมโดยวัดความซับซ้อนภายในคลาสเป็นหลัก ซึ่งมีสูตรการคำนวณกันอยู่หลายวิธี แต่โมเดลประยุกต์มีวิธีการคำนวณให้เลือกดังต่อไปนี้

- การวัดความซับซ้อนภายในคลาส (IntraClass Metrics) มีอยู่หลายวิธีแต่วิธีที่นิยม คือ วิธีของ McCabe และ Jacobson วิธีของ McCabe [28] จะวัดความซับซ้อนของโปรแกรมโดยวัดแบบ LOC (Line of Code) ที่นำเสนอการวัดความซับซ้อนโดยใช้ค่าไซโคลเมติค (cyclomatic number) ที่วัดจากโฟลว์กราฟของโปรแกรม ซึ่งเริ่มวัดจากโหนดเริ่มต้นจนถึงโหนดสิ้นสุดและมีสูตรการคำนวณดังนี้

$$v(F) = e - n + 2$$

โดยที่ F มี arcs จำนวน e เส้น และมี n โหนด ค่าไซโคลเมติคเป็นการวัดจำนวนเส้นทางที่เป็นอิสระเชิงเส้น (linearly independent paths) ของ F การวัดนี้จะเป็นไปตามวัตถุประสงค์และเป็นประโยชน์เมื่อนับเส้นทางที่เป็นอิสระเชิงเส้นได้ถูกต้อง ค่าไซโคลเมติคสามารถคำนวณได้อีกวิธีหนึ่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$v(F)=1+d$$

โดยที่ d คือจำนวนโหนดที่มีทิศทางออกจากตัวเอง (out degree) มากกว่า 1 ซึ่งอยู่ใน F นั้น จากทฤษฎีการวัดที่ McCabe แนะนำไว้ค่า V ของโมดูลควรมีค่าระหว่าง 3-7 แต่ไม่ควรเกิน 10 จึงจะเป็นค่าที่เหมาะสม

- การวัดความซับซ้อนของโปรแกรมยังมีอีกลักษณะหนึ่งคือ พิจารณาจากระดับโครงสร้างของโปรแกรมทั้งหมดจากโฟลว์กราฟ และกำหนดค่าปัจจัยความซับซ้อน (ev) ไว้ดังนี้

$$ev(F)=v(F)-m$$

โดยที่ m คือจำนวนของโฟลว์กราฟย่อยของ F ซึ่งอยู่ในรูปของ D-Structured prime หรือเรียกว่าโครงสร้าง D จากโฟลว์กราฟ F มีโฟลว์กราฟย่อยที่เป็นโครงสร้าง D เพียงโครงสร้างเดี่ยวคือ โครงสร้างของ D_1 ซึ่งอยู่ทางขวาของโฟลว์กราฟ ดังนั้น m = 1 และ v(F) = 5 สามารถแทนค่าตามสูตรได้ดังนี้ ev(F) = 5-1 ฉะนั้นค่าปัจจัยความซับซ้อนของ ev(F) จึงมีค่าเท่ากับ 4 เป็นต้น ถ้าโฟลว์กราฟที่นำมาวัดความซับซ้อนมีโครงสร้างของโฟลว์กราฟลักษณะของโครงสร้าง D ประกอบอยู่สามารถลดค่าความซับซ้อนลงได้อีก [1] ส่วนการวัดขนาดของโปรแกรมของ Thomas กับ Jacobson มีสูตรการคำนวณดังนี้

$$s=(AW_A+MW_M)$$

ให้ s แทนคลาส ส่วน A กับ M แทนแอตทริบิวต์และโอเปอร์เรชั่นตามลำดับ ค่าถ่วงน้ำหนักของ W_A และ W_M เป็นค่าคงที่ โดยที่ W_A มีค่าประมาณ 1 และ W_M มีค่าประมาณ 5-20 ซึ่งขึ้นอยู่กับค่าเฉลี่ยของโอเปอร์เรชั่นหรือเซอร์วิสของคลาสที่นำมาวัด แต่กรณีของ Lorenz[33] จะขึ้นอยู่กับขนาดของเซอร์วิสโดยเฉลี่ยและจำนวนของเซอร์วิสที่เฉลี่ยต่อคลาส ซึ่งทั้ง 2 วิธีที่นำเสนอไปนี้สามารถทำให้ผู้พัฒนาทราบถึงความสัมพันธ์และความซับซ้อนของโปรแกรมที่นำเสนอได้เป็นอย่างดี และสามารถนำไปเป็นแนวทางการปรับปรุงระบบ ประมาณค่าใช้จ่ายที่จะเกิดขึ้นต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์และตาราง

ตารางที่ 5.16 แสดงการวัดความซับซ้อนของฟังก์ชันภายในวัตถุ

Test Plan Page ___ of ___
Object Name: _____ **Program ID:** _____ **Version Number:** _____ **Tester:** _____ **Date Designed :** _____ **Date Conducted :** _____ **Results:** ☐ **Passed** ☐ **Open Items**
Test ID: _____ **Requirement Addressed:** _____ **Objective:**
Flow Graph
Script (interclass Metrics)
Expected Results/Notes
Actual Results/Notes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 5.16 แสดงความซับซ้อนของฟังก์ชันภายในวัตถุ ซึ่งการวัดนี้ผู้ออกแบบและโปรแกรมเมอร์เป็นผู้วัดการทำงานของฟังก์ชัน การกำหนดค่าควรกำหนดเป็นค่ามาตรฐานหรือตามความเหมาะสมของฟังก์ชัน

5) ขั้นตอน 750 รวบรวมเอกสาร (Assembly Report)

ขั้นตอนที่ 710 คือ การสรุปการทำงานเปรียบเสมือนการตรวจสอบเอกสารต่าง ๆ ที่เกิดขึ้นของระยะที่ 7 ตั้งแต่ขั้นตอนที่ 710 จนถึงขั้นตอนที่ 740 ซึ่งถ้าเอกสารของแต่ละขั้นนั้นไม่สมบูรณ์หรือไม่ครบถ้วน ผู้พัฒนาโปรแกรมควรกลับไปตรวจสอบการทำงานของระยะที่ 7 ตั้งแต่ขั้นตอนที่ 710 จนถึงขั้นตอนที่ 740 อีกครั้ง เอกสารของขั้นตอนนี้มีดังนี้

5.5.4.2 ระยะที่ 8 การส่งมอบ (Delivery)

ระยะที่ 8 คือ ขั้นตอนของการส่งมอบและการติดตั้งระบบงานที่ต่อเนื่องจากระยะที่ 7 แบ่งออกเป็น 5 ขั้นตอน เริ่มตั้งแต่ การทดสอบระบบจริงกับซอฟต์แวร์แพ็กเกจ(ขั้นตอนสุดท้ายของการทดสอบ) วางแผนการนำระบบงานใช้กับองค์กร ติดตั้งฐานข้อมูล อบรมผู้ใช้และเปลี่ยนไปสู่การทำงานของระบบใหม่ โดยแต่ละขั้นตอนประกอบด้วยรายละเอียดต่าง ๆ พร้อมเอกสารแสดงอย่างชัดเจนและมีลำดับการทำงานดังนี้

1) ขั้นตอน 810 ตรวจสอบการทำงานของระบบ (Conduct System)

จุดประสงค์ของขั้นตอนนี้ คือ การตรวจสอบระบบเป็นครั้งสุดท้ายเพื่อให้มั่นใจว่า ระบบที่พัฒนาขึ้นสามารถทำงานรวมกับซอฟต์แวร์แพ็กเกจได้ ซึ่งกิจกรรมนี้อยู่ในขั้นตอนการส่งมอบฉะนั้นการตรวจสอบควรให้ เจ้าของโครงการ นักวิเคราะห์ระบบและผู้พัฒนาโปรแกรม เข้ามาเกี่ยวข้องกับกิจกรรมนี้ด้วย เพื่อมาช่วยกันตรวจสอบระบบ เพราะถ้าพบข้อผิดผลาดสามารถที่จะแก้ไขได้ทันที การตรวจสอบระบบมีขั้นตอนดังนี้

งานของขั้นตอน 810

งานที่ 10 ตรวจสอบข้อมูลที่อยู่ในระบบเพื่อความถูกต้องหลังจากนำข้อมูลขององค์กรใส่ลงไปในระบบงาน

งานที่ 20 ตรวจสอบซอฟต์แวร์แพ็กเกจและโปรแกรมที่พัฒนาระบบไปพร้อมกัน โดยทดสอบการใช้งานในทุก ๆ โมดูลของระบบงานซึ่งระหว่างการทดสอบควรทดสอบกรณีของการทำงานที่มักจะเกิดขึ้นซ้ำ ๆ ของการปฏิบัติงานแต่ละครั้ง เช่น การป้อนข้อมูลผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภท ข้อมูลเกิน หรือ ทำการเปิดซอฟต์แวร์แพ็จเกจพร้อมกับโปรแกรมระบบงานพร้อมกัน เป็นต้น

งานที่ 30 บันทึกรายละเอียดต่าง ๆ ที่เกิดขึ้น เพราะรายละเอียดต่าง ๆ เหล่านี้สามารถนำไปปรับปรุงแก้ไขหรือเพิ่มเติมให้กับระบบ กรณีที่พบข้อผิดพลาดที่เกิดขึ้น

2) ขั้นตอน 820 วางแผนการติดตั้งโปรแกรม (Prepare Conversion Plan)

เมื่อทดสอบและตรวจสอบซอฟต์แวร์แพ็กเกจของระบบงานที่พัฒนาเป็นที่เรียบร้อยแล้ว และผลที่ได้นั้นเป็นที่พอใจและถูกต้องตามความต้องการก็จะเข้าสู่ขั้นตอนการติดตั้ง ทดสอบระบบและนำไปใช้งานในแต่ละแผนก ส่วนหรือฝ่ายภายในองค์กร ซึ่งการติดตั้งและการทดสอบเป็นขั้นตอนหลักของขั้นตอน 820 มีการทำงานดังต่อไปนี้

งานของขั้นตอน 820

งานที่ 10 กำหนดรูปแบบการติดตั้งระบบใหม่ โดยทั่วไปรูปแบบของการติดตั้งระบบมี 4 วิธี ซึ่งแตกต่างกัน การเลือกรูปแบบการติดตั้งนักวิเคราะห์ควรจะต้องคำนึงถึงความเหมาะสมและลักษณะของระบบที่พัฒนาเป็นสำคัญ เนื่องจากการเลือกรูปแบบผิดย่อมทำให้การนำไปใช้งานเกิดความล่าช้าและเกิดผลเสียกับองค์กร การติดตั้งระบบใหม่ทั้ง 4 วิธี มีรายละเอียดดังนี้

- การติดตั้งแบบทันที (Abrupt Cut-Over) คือ การนำระบบที่พัฒนาใหม่ไปแทนทีระบบเดิมทันทีโดยไม่คำนึงถึงฐานข้อมูลที่มีอยู่เดิม ซึ่งการติดตั้งแบบนี้เป็นการติดตั้งที่มีความเสี่ยงกับระบบเดิมและระบบใหม่ เพราะหากระบบที่ติดตั้งมีการทำงานที่ผิดพลาดขึ้นจะไม่สามารถกู้คืนข้อมูลหรือระบบงานเดิมขึ้นมาได้
- การติดตั้งแบบขนาน (Parallel Conversion) คือ การติดตั้งระบบใหม่ให้ทำงานไปพร้อมกับระบบเดิมจนกว่าระบบใหม่มีข้อมูลต่าง ๆ เท่ากับระบบเดิมจึงค่อยนำไปสู่การใช้ระบบใหม่อย่างเต็มรูปแบบ ซึ่งมีข้อดี คือ ถ้าหากระบบใดระบบหนึ่งเกิดผิดพลาดหรือใช้งานไม่ได้ย่อมไม่ส่งผลกระทบต่องานขององค์กร แต่วิธีนี้จะทำให้องค์กรมีค่าใช้จ่ายเพิ่มขึ้น ทางด้านทรัพยากรของเครื่อง เช่น หน่วยความจำ สตอเรจ(Storage) เป็นต้น เพราะระบบถูกใช้งานพร้อมกัน 2 ระบบ
- การติดตั้งแบบเฉพาะที่ (Location Conversion) คือ การเริ่มติดตั้งระบบโดยนำส่วนใดส่วนหนึ่งของระบบไปทดลองติดตั้งให้กับแผนก ส่วนหรือฝ่ายที่ต้องการเปลี่ยนไปสู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบใหม่ก่อน เมื่อระบบที่นำไปติดตั้งสามารถทำงานได้เป็นที่น่าพอใจแล้วจากนั้นจึงขยายการติดตั้งไปจนครบทั้งระบบ ซึ่งการติดตั้งลักษณะนี้ เรียกว่า "a Beta test site" (การติดตั้งวิธีนี้จะมีวิธีการคล้ายกับ การติดตั้งแบบทันทีกับการติดตั้งแบบขนาน ผสมกันอยู่) ซึ่งวิธีนี้มีข้อดี คือ ทำให้สามารถตรวจสอบและหาข้อผิดพลาดที่เกิดขึ้นกับระบบได้ง่าย เนื่องจากเป็นระบบที่ติดตั้งในแบบเฉพาะที่

- การติดตั้งแบบช่วง (Staged Conversion) มีลักษณะการทำงานที่เป็นแบบผสมผสานกันใน 3 วิธีที่กล่าวมาข้างต้น แต่หลักการของวิธีนี้ คือ การติดตั้งในลักษณะเวอร์ชันของระบบ กล่าวคือ ระบบใดที่พัฒนาหรือปรับปรุงแก้ไขเสร็จก่อนก็จะถูกนำไปติดตั้งหรือทดลองใช้จริง ซึ่งจะทำแบบนี้ไปจนครบทั้งระบบ จากที่กล่าวมาข้างต้นเป็นเพียงวิธีการติดตั้งที่นิยมใช้กันในปัจจุบันซึ่งนักวิเคราะห์ระบบจะต้องวางแผนการติดตั้งให้ชัดเจน เมื่อกำหนดแผนการติดตั้งระบบงานเรียบร้อยแล้ว จากนั้นทำการเลือกแผนงานที่จะทำการติดตั้งให้กับระบบงานใหม่พร้อมกับการติดตั้งระบบงาน

งานที่ 20 ตรวจสอบระบบงานที่ติดตั้งตามแผนงาน ซึ่งเรียกการตรวจสอบนี้ว่า "การยอมรับการตรวจสอบระบบ (System Acceptance Test)" การตรวจสอบนี้เป็นขั้นตอนตรวจสอบสุดท้ายหลังจากระบบเริ่มทำงานแล้ว เพื่อเป็นการยืนยันให้กับผู้ใช้ว่า ระบบสามารถที่จะทำงานได้ การยอมรับการตรวจสอบระบบมีวิธีการตรวจสอบดังนี้

- Verification Testing คือ การยืนยันระบบงานโดยจำลองการทำงานและข้อมูลในสภาพแวดล้อมจริง ซึ่งเรียกว่า "Alpha Testing" การทดสอบงานลักษณะนี้ทำเพื่อตรวจสอบข้อผิดพลาดที่อาจเกิดขึ้นในขั้นตอนของการติดตั้งระบบ
- Validation Testing คือ การทดสอบความถูกต้องของระบบงานกับข้อมูลจริงในสภาพแวดล้อมการทำงานจริง บางครั้งเรียกว่า "Beta Testing" การทดสอบลักษณะนี้ ได้แก่ 1) ทดสอบประสิทธิภาพของระบบ (Systems performance) คือ การทดสอบการตอบสนองของระบบในแต่ละทรานส์แอ็กชันที่ผู้ใช้ติดต่อกับระบบ 2) การทดสอบการทำงานของระบบงานเต็มประสิทธิภาพ (Peak workload processing performance) คือ การทดสอบขีดความสามารถของระบบว่าสามารถรองรับทรานส์แอ็กชันอย่างเต็มที่ได้มากน้อยเพียงใด เพื่อนำไปปรับปรุงในกรณีที่ประสิทธิภาพของระบบไม่ตรงตามความต้องการ เช่น ปรับปรุงฮาร์ดแวร์ และ/หรือ ซอฟต์แวร์ เป็นต้น 3) ทดสอบการใช้งานกับผู้ใช้ (Human engineering test) คือ การทดสอบส่วนการเชื่อมต่อกับผู้ใช้ (User Interface) ของระบบว่ามีความยากง่ายเพียงใด สามารถปรับปรุงได้หรือไม่ 4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบโพรซีเจอร์และเมธอด(Methods and procedures test) เนื่องจากระหว่างการเปลี่ยนแปลงไปสู่ระบบงานใหม่ เมธอดและโพรซีเจอร์เหล่านั้นจะต้องถูกปรับเปลี่ยนให้เหมาะสำหรับการใช้งานกับผู้ใช้เพื่อให้การปฏิบัติงานเป็นไปตามมาตรฐานของผู้ใช้ เช่น การประมวลผล ความสามารถในการรองรับงาน ณ เวลาใดเวลาหนึ่ง เป็นต้น

5) ทดสอบการกู้คืนข้อมูล (Backup and Recovery Testing) คือการทดสอบการกู้คืนข้อมูลของระบบงาน อาจทำโดยจำลองเหตุการณ์สำหรับการกู้คืนข้อมูล โดยเหตุการณ์จำลองนี้ควรทำกับข้อมูลจริงในระบบงาน

- Audit Testing คือ การรับประกันระบบที่ปราศจากข้อผิดพลาดจากผู้พัฒนาระบบ เพื่อเป็นการยืนยันว่าระบบมีข้อผิดผลาดน้อยที่สุดหรือไม่เกิดข้อผิดพลาดเลยขณะนำไปปฏิบัติงานจริง

สรุปขั้นตอน 820 คือการเตรียมแผนงานการย้ายระบบงานเดิมไปสู่ระบบงานใหม่พร้อมกับทดสอบระบบงานหลังจากที่ย้ายไปสู่ระบบงานใหม่เป็นที่เรียบร้อยแล้ว ซึ่งการทดสอบหลังการย้ายไปสู่ระบบงานใหม่นี้ เป็นสิ่งที่จำเป็นอีกอย่างหนึ่ง เพราะระบบงานที่จะปฏิบัติงานได้ดีนั้นควรที่จะทำการทดสอบทั้งก่อนและหลังการติดตั้งเพื่อป้องกันการผิดผลาดในทุกกรณี ดังนั้นกิจกรรมนี้จึงไม่ควรมองข้ามไปเพราะเป็นกิจกรรมที่กล่าวถึงการวานแผนและทดสอบระบบหลังจากที่ติดตั้งเสร็จเรียบร้อยแล้ว

3) ขั้นตอน 830 การติดตั้งฐานข้อมูล (Install Database)

จากขั้นตอน 720 เพียงการสร้างและทดสอบฐานข้อมูลโดยการสุ่มตัวอย่างฐานข้อมูลเพียงบางส่วนมาทำการทดสอบเท่านั้น ซึ่งขั้นตอนที่ 830 นี้ แตกต่างจากกิจกรรมก่อนหน้าคือ นำข้อมูลจริงที่ธุรกิจนั้นปฏิบัติงานจริง โดยมีรายละเอียดดังนี้

งานของขั้นตอน 830

งานที่ 10 นำฐานข้อมูลที่มีอยู่จริงในระบบงานเดิมย้ายไปสู่ฐานข้อมูลของระบบงานใหม่ ก่อนที่จะย้ายควรที่จะตรวจสอบโครงสร้างของฐานข้อมูลและระบบของฐานข้อมูลก่อนเพื่อความเรียบร้อยและป้องกันความผิดพลาดขณะย้ายข้อมูลลงสู่ฐานข้อมูลของระบบใหม่ โดยศึกษาคู่มือของระบบของฐานข้อมูลนั้นให้ถี่ถ้วนก่อนปฏิบัติงานซึ่งอาจจะปรึกษาผู้เชี่ยวชาญด้านระบบฐานข้อมูลหรือเจ้าหน้าที่ฝ่ายเทคนิคของบริษัทเจ้าของผลิตภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 20 ย้ายข้อมูล การย้ายข้อมูลนั้นบางครั้งผู้พัฒนาอาจจะต้องเขียนโปรแกรมขึ้นมาสำหรับการย้ายข้อมูลเข้าสู่ฐานข้อมูล (ปกติระบบฐานข้อมูลจะมีเครื่องมือหรือยูทิลิตี้ไว้สำหรับการย้ายฐานข้อมูลติดมากับตัวระบบ) เมื่อเตรียมทุกอย่างพร้อมแล้วก็ทำการย้ายข้อมูลเข้าสู่ระบบใหม่

งานที่ 30 บันทึกขั้นตอนและรายละเอียดขณะทำการย้ายข้อมูลลงฐานข้อมูลในระบบงานใหม่เพื่อป้องกันการผิดพลาดและสร้างความมั่นใจว่าข้อมูลที่ย้ายไปนั้นไม่มีผลกระทบกับระบบงานใหม่ที่กำลังจะนำไปใช้งาน

4) ขั้นตอน 840 ฝึกอบรมผู้ใช้ระบบ (Train System User)

ระบบที่ดีไม่ใช่สิ้นสุดเพียงแค่ทำการติดตั้งสำเร็จเท่านั้น แต่ต้องขึ้นอยู่กับการนำไปใช้งานซึ่งการใช้งานนั้นจะมีประสิทธิภาพได้นั้นจำเป็นต้องมีการฝึกอบรมผู้ใช้ จัดทำคู่มือการใช้งานและคัดเลือกวิทยากร ซึ่งแต่การปฏิบัติแต่ส่วนสามารถสรุปได้ดังนี้

งานของขั้นตอน 840

งานที่ 10 กำหนดรูปแบบการอบรมผู้ใช้ สามารถแบ่งเป็น การอบรมแบบหนึ่งต่อหนึ่งและแบบกลุ่ม การอบรมการใช้งานแบบหนึ่งต่อหนึ่ง เป็นวิธีที่ได้ผลดีแต่จะเสียเวลามาก เพราะหากมีผู้เข้ารับการอบรมมากระยะเวลาที่ใช้ย่อมมากขึ้นเป็นลำดับ ส่วนการฝึกอบรมแบบกลุ่มนั้น มีข้อดีคือประหยัดเวลา ตามปกติผู้ที่จะเข้ารับการอบรมการใช้งานระบบนั้นจะเป็นระดับหัวหน้าฝ่ายหรือแผนก เพื่อความสะดวกและรวดเร็ว เพราะหลังจากการเข้ารับอบรม หัวหน้างานเหล่านี้สามารถไปจัดการอบรมย่อยให้กับผู้ใต้บังคับบัญชาได้อีกต่อหนึ่ง

งานที่ 20 จัดเตรียมเอกสารหรือคู่มือการใช้งาน เป็นงานที่สำคัญอีกส่วนหนึ่งที่ไม่ควรละเลย เพราะถ้าคู่มือการใช้งานของระบบงานจัดทำไม่ดีหรือไม่ชัดเจนพอ ย่อมส่งผลต่อการอบรม ถึงแม้จะมีการจัดฝึกอบรมให้ผู้ใช้ถูกต้องตรงตามความสามารถของระบบก็ตาม

งานที่ 30 คัดเลือกผู้เข้าอบรม

งานที่ 40 คัดเลือกผู้ถ่ายทอด เพราะถ้าผู้ถ่ายทอดไม่มีความชำนาญกับระบบงานที่พัฒนาขึ้นย่อมส่งผลเช่นเดียวกัน ดังนั้นผู้ที่จะมาถ่ายทอดความรู้ควรเป็นผู้ที่รับผิดชอบกับระบบงานนั้น เช่น ผู้วิเคราะห์และออกแบบ เจ้าของระบบงาน เป็นต้น จัดได้ว่าบุคคลเหล่านี้สามารถให้ความรู้กับผู้ใช้งานได้ เพราะมีความชำนาญและเข้าใจลักษณะการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของระบบเป็นอย่างดี เพราะเป็นบุคคลที่คลุกคลีกับระบบงานที่พัฒนามาตั้งแต่ต้น จึงสมควรอย่างยิ่งที่จะเป็นผู้ถ่ายทอดความรู้ให้กับผู้ใช้ในองค์กรที่ใช้ระบบงานนั้น

5) ขั้นตอน 850 เปลี่ยนแปลงไปสู่ระบบงานใหม่ (Convert to New System)

ตั้งแต่ขั้นตอนที่ 010-840 เริ่มตั้งแต่ วิเคราะห์ ออกแบบ พัฒนา ทดสอบ การย้ายระบบงานและจัดฝึกอบรมให้กับผู้ใช้ ตามลำดับ ขั้นตอนที่ 850 ซึ่งเป็นขั้นตอนสุดท้ายของการพัฒนาระบบงาน หลังจากที่ย้ายไปสู่ระบบงานใหม่และอบรมผู้ใช้งาน ดังนั้นลักษณะงานของขั้นตอนนี้คือ ตรวจสอบ ปรับปรุงเอกสารและส่งมอบงานให้กับเจ้าของโครงการ งานแรก คือ ตรวจสอบแผนงานที่เลือกเพื่อความยืนยันความต้องการที่ชัดเจนของแผนงานนั้น พร้อมกับบันทึกและปรับปรุงรายละเอียดต่าง ๆ ที่เกิดขึ้นทั้งก่อนและหลังการติดตั้ง

6) ขั้นตอนที่ 860 ตรวจสอบการส่งมอบระบบ

ขั้นตอนที่ 860 เป็นการตรวจสอบการส่งมอบระบบ คือ ตรวจสอบงานของระยะที่ 8 ตั้งแต่ขั้นตอนที่ 810 จนถึงขั้นตอนที่ 850 ซึ่งถ้าการส่งมอบระบบงานนั้นไม่สมบูรณ์หรือไม่ครบถ้วน ผู้รับผิดชอบโครงการควรกลับไปตรวจสอบการทำงานของระยะที่ 8 ตั้งแต่ขั้นตอนที่ 810 จนถึงขั้นตอนที่ 850 อีกครั้ง เอกสารของขั้นตอนนี้มีดังนี้

5.5.5 โมดูลการสนับสนุนระบบ (System Support Module)

ส่วนใหญ่ระบบที่ถูกพัฒนาขึ้นตามขั้นตอนของวิธีการพัฒนาระบบสารสนเทศไม่ว่าวิธีการใดก็ตาม เมื่อมีการใช้งานไปได้ชักระยะหนึ่งย่อมเกิดการเปลี่ยนแปลงขึ้น โดยการเปลี่ยนแปลงที่เกิดขึ้นมักจะเกิดจากปัจจัยหลายอย่าง เช่น ความต้องการของผู้ใช้เพิ่มขึ้น เกิดความผิดพลาดของระบบหรือที่เรียกว่า “Bugs” สำหรับการใช้งานปัจจุบัน การทำงานของระบบมีความล่าช้าและระบบล้าสมัย เป็นต้น ดังนั้นการพัฒนาระบบสารสนเทศจึงจำเป็นต้องมีโมดูลนี้เพื่อมาตรวจสอบและปรับปรุงสิ่งต่าง ๆ ที่เกิดขึ้นกับระบบให้อยู่ในสภาพที่ดีอยู่เสมอ

ดังนั้นในโมเดลประยุกต์จึงจำเป็นต้องมีโมดูลที่ 5 ซึ่งเป็นโมดูลสุดท้ายที่คอยติดตามผล ปรับปรุงและบำรุงรักษาระบบ ซึ่งประกอบด้วย ระยะที่ 9 และ ระยะที่ 10 กล่าวคือ ระยะการติดตามผลและระยะการปรับปรุงและบำรุงรักษาระบบ ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.5.1 ระยะที่ 9 การติดตามผล (Monitoring)

การติดตามผลการทำงานของระบบสารสนเทศนั้นส่วนใหญ่มักกระทำหลังจากที่ผู้ใช้ได้ใช้ระบบไประยะเวลาหนึ่ง การปฏิบัติงานของผู้ใช้ระบบบางครั้งอาจทำให้ระบบนั้นเกิดปัญหาหรือข้อผิดพลาดขึ้นได้ ถึงแม้ว่าผู้พัฒนาระบบได้ทำการตรวจสอบตั้งแต่ระยะที่ 6 และ ระยะที่ 7 ก็ตาม แต่ปัญหาอาจเกิดขึ้นได้ ข้อผิดพลาดที่เกิดขึ้นนั้นมักมาจากหลายหลายสาเหตุ เช่น การทำงานผิดพลาดของโปรแกรม ข้อมูลเข้าผิดพลาดหรือรายงานต่าง ๆ ไม่ตรงกับข้อมูลที่มีอยู่ เป็นต้น ดังนั้นระยะการติดตามผลของระยะที่ 9 คือ การตรวจสอบการทำงาน การเปลี่ยนแปลงที่เกิดขึ้นกับนโยบายของธุรกิจ/องค์กร เพื่อให้ระบบปฏิบัติงานได้ถูกต้อง ซึ่งขั้นตอนของระยะที่ 9 มีดังต่อไปนี้

ขั้นตอน 910 การปรับปรุงระบบเพื่อสะท้อนความต้องการที่เปลี่ยนแปลงไป (Modify Repository to reflect change)

ขั้นตอนที่ 910 คือ การปรับปรุงเปลี่ยนแปลงรักษาโปรแกรมและระบบให้ตรงกับการทำงานของผู้ใช้ การบำรุงรักษาระบบหรือโปรแกรมนั้น ผู้วิเคราะห์ต้องเข้าใจระบบหรือโปรแกรมที่ต้องการปรับปรับและ/หรือบำรุงรักษาเป็นอย่างดี เพราะถ้าไม่เข้าเข้าระบบดีพออาจทำให้การปรับปรุงนั้นเกิดข้อผิดพลาดขึ้นได้ การทำความเข้าใจระบบและ/หรือโปรแกรมนั้นมีหลักอยู่ 5 ประการดังนี้

- ประการแรก ทำความเข้าใจกับฐานข้อมูลที่ใช้งานอยู่ในระบบ เพราะฐานข้อมูลเป็นสิ่งหนึ่งที่สามารถบอกถึงนโยบายของธุรกิจและปริมาณข้อมูลที่เข้าและออกของระบบได้
- ประการที่สอง ทำความเข้าใจการทำงานของโปรแกรมหรือฟังก์ชันต่าง ๆ ของระบบ การทำงานของฟังก์ชันสามารถแสดงถึงความสอดคล้องกันในแต่ละฟังก์ชันที่อยู่ในระบบ ถ้าในระบบมีฟังก์ชันที่เกี่ยวข้องกับระบบอื่นควรทำความเข้าใจฟังก์ชันที่เชื่อมต่อนั้นด้วย
- ประการที่สาม ทำความเข้าใจส่วนที่ติดต่อกับผู้ใช้ หรือ “Graphic User Interface (GUI)” ซึ่งสามารถทำให้ผู้วิเคราะห์เข้าใจถึงความยากง่ายของการใช้ของผู้ใช้
- ประการที่สี่ ทำความเข้าใจระบบเครือข่าย (ถ้ามี) ซึ่งจะเป็นภาพรวม ถ้าระบบนั้นมีการทำงานผ่านระบบเครือข่าย เพราะการทำงานผ่านระบบเครือข่ายนั้นการกระทำใด ๆ กับระบบใด ๆ ย่อมมีผลกระทบกับระบบที่เกี่ยวข้อง ดังนั้นผู้วิเคราะห์ไม่ควรละเลยการทำความเข้าใจระบบเครือข่ายที่ระบบทำงานอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ประการที่ห้า ทำความเข้าใจเทคโนโลยีของระบบที่ต้องการปรับปรุงและ/หรือแก้ไข เพราะระบบสารสนเทศที่ถูกใช้งานมาเป็นเวลานานอาจใช้เทคโนโลยีสมัยนั้น เช่น ใช้ฐานข้อมูลแบบ ISAM/VSAM โปรแกรมพัฒนาแบบเชิงโครงสร้าง หรือระบบเครือข่ายใช้เทคโนโลยีแบบบัส (BUS Technology) เป็นต้น

การทำความเข้าใจระบบนอกจากหลักการทั้ง 5 ที่กล่าวมาข้างต้นสิ่งหนึ่งที่สามารบอกถึงความถูกต้องและผิดพลาดของระบบนั้น คือ อินพุตและเอาต์พุต เพราะอินพุต คือ สิ่งที่ระบบรับไปประมวลผลเพื่อให้ได้เอาต์พุตตามความต้องการซึ่งถ้าอินพุตผิดผลาดย่อมทำให้เอาต์พุตผิดพลาดตามไปด้วย ประสิทธิภาพการทำงานของระบบมีส่วนสำคัญเช่นเดียวกัน ประสิทธิภาพ (Benchmark) คือ การตอบสนองของโปรแกรมหรือแอพพลิเคชั่นภายในระบบ ซึ่งระบบเมื่อนำไปใช้งานหลังจากพัฒนาเสร็จสิ้นนั้น เมื่อใช้งานมาระยะเวลาหนึ่งอาจมีการทำงานล่าช้าหรือผิดพลาดไปบ้าง ผู้วิเคราะห์จึงต้องควรพิจารณาส่วนนี้ด้วย

จากที่กล่าวในส่วนของโปรแกรมข้างต้นนั้น โปรแกรมที่ต้องพิจารณานั้น นอกจากการทำงานของโปรแกรมนั้น ยังต้องพิจารณาถึงโครงสร้าง แนวคิดของโปรแกรม การออกแบบโปรแกรมและเอกสารประกอบ สิ่งต่าง ๆ เหล่านี้เป็นอีกสิ่งหนึ่งที่ผู้วิเคราะห์ควรต้องเอาใจใส่ เพราะจะทำให้เข้าใจระบบได้ง่ายขึ้นและมีความรวดเร็วในการปรับปรุงระบบ

ดังนั้นขั้นตอนที่ 910 จึงมีวัตถุประสงค์ดังได้กล่าวมาข้างต้น คือ ทำความเข้าใจกับระบบสารสนเทศที่ใช้งานอยู่เพื่อให้ทำการปรับปรุงแก้ไขได้ง่ายและรวดเร็ว เพราะถ้ามีความเข้าใจระบบย่อมเป็นผลต่อการบำรุงรักษาและสามารถผ่านไปทำขั้นตอนถัดไปได้รวดเร็วขึ้น

ขั้นตอน 920 การควบคุมเอกสาร (Version Control)

จากความต้องการที่เปลี่ยนแปลงไปของระบบสารสนเทศที่ใช้อยู่นั้น จึงต้องแก้ไขหรือปรับปรุงระบบให้ตรงตามวัตถุประสงค์และความต้องการของผู้ใช้ กระบวนการที่เกิดขึ้นในระบบจึงต้องนำมาแก้ไขและทดสอบหลังจากดำเนินการต่าง ๆ ซึ่งการทดสอบกระบวนของระบบสารสนเทศนั้นมีหลายวิธี เช่น ทดสอบแบบ Unit Test ทดสอบแบบ System Test หรือทดสอบแบบ Reengineering Test เป็นต้น การทดสอบที่กล่าวมาข้างต้นบางครั้งเรียกว่า “Version Control” ซึ่งเป็นการตรวจสอบซอฟต์แวร์หรือโปรแกรมที่ใช้ปฏิบัติงานอยู่ขณะนั้น เพื่อสังเกตพฤติกรรมการทำงานของโปรแกรมที่ได้ทำการแก้ไขและหรือปรับปรุงให้มีการทำงานข้ากันได้กับระบบที่เชื่อมต่อหรือมีอยู่ ดังนั้นผู้ออกแบบโปรแกรมจึงต้องทดสอบโปรแกรมทุกครั้งที่แก้ไขเสร็จสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอน 930 ปรับปรุงข้อมูลทางธุรกิจให้ทันสมัย (Update Business Data)

หลังจากทำความเข้าใจการทำงานของระบบและพัฒนาโปรแกรมและ/หรือปรับปรุงแก้ไขพร้อมกับทดสอบโปรแกรมเป็นที่เรียบร้อยแล้ว สิ่งหนึ่งที่ควรทำ คือ การปรับปรุงเอกสารของโปรแกรมที่แก้ไข เพราะเอกสารเป็นส่วนหนึ่ง รายละเอียดที่แสดงภายในเอกสารนั้น ประกอบด้วย โครงสร้างของโปรแกรม แนวคิด พารามิเตอร์และการเชื่อมต่อระหว่างโปรแกรมในระบบ (ถ้ามี) ถ้าผู้วิเคราะห์และโปรแกรมเมอร์ละเลยขั้นตอนนี้ย่อมทำให้การแก้ไขโปรแกรมครั้งต่อไปมีความลำบากและล่าช้าได้

5.5.5.2 ระยะที่ 10 การปรับปรุงและบำรุงรักษา (Evolution and Maintenance)

การปรับปรุงและบำรุงรักษาเป็นขั้นตอนที่สำคัญอีกขั้นตอนหนึ่งเพราะเป็นขั้นตอนที่ทำให้ระบบที่ใช้งานอยู่ปัจจุบันสามารถตอบสนองต่อความต้องการของผู้ใช้ได้อย่างถูกต้องและมีประสิทธิภาพ ซึ่งการปรับปรุงและบำรุงรักษานั้น ผู้ที่มีหน้าที่เกี่ยวข้องอาจจะเป็นผู้วิเคราะห์และ/หรือโปรแกรมเมอร์ ที่คอยนำความต้องการและปัญหาต่าง ๆ ที่เกิดขึ้นมาปรับปรุงและแก้ไขให้กับระบบ ดังนั้นการทำงานของขั้นตอนนี้จึงมีการทำงานเวียนกลับไปยังขั้นตอนเริ่มต้นอีกครั้ง กล่าวคือ วิเคราะห์ออกแบบ อิมพลีเมนต์และทดสอบ เพื่อให้ปัญหาหรือความต้องการเหล่านั้นมีการแก้ไขได้ตรงกับความต้องการของผู้ใช้และมีขั้นตอนการดำเนินงานตามวิธีการพัฒนาระบบสารสนเทศเชิงวัตถุ ในการแก้ไขปัญหาของระบบสารสนเทศนั้นการพิจารณาข้อบกพร่องต่าง พิจารณาได้จาก การทำงานของผู้ใช้ นโยบายทางธุรกิจและเทคโนโลยี ซึ่งระยะที่ 10 มีขั้นตอนดังนี้

ขั้นตอน 1010 การให้ความช่วยเหลือผู้ใช้ (End-User Assistance)

การปรับปรุงและบำรุงรักษาระบบให้ทันสมัยอยู่ตลอดเวลานั้น การได้รับคำแนะนำหรือรายงานจากผู้ใช้เป็นสิ่งสำคัญอีกประการหนึ่งหลังจากระบบสารสนเทศได้ถูกใช้งาน การปฏิบัติงานของระบบจากผู้ใช้แต่ละครั้งอาจมีการเปลี่ยนแปลงไปตามระยะเวลา หรือตามนโยบายทางธุรกิจ ซึ่งสิ่งเหล่านี้เป็นสิ่งสำคัญสำหรับการพัฒนาระบบสารสนเทศให้มีประสิทธิภาพมากขึ้น ถึงแม้ว่าระบบที่ถูกพัฒนาขึ้นมานั้นจะมีการวางแผนที่ดี มีการพัฒนาที่ละเอียด มีเอกสารที่ครอบคลุมทุกขั้นตอนหรือผู้ใช้งานถูกอบรมมาอย่างดี ก็ตาม ระบบย่อมต้องเกิดข้อผิดพลาดและมีความต้องการใหม่ ๆ ให้กับระบบผู้วิเคราะห์ระบบจึงจำเป็นต้องรับรายงานต่าง ๆ จากผู้ใช้เพื่อพัฒนาระบบให้ดีขึ้นอยู่ตลอดเวลา ดังนั้นวัตถุประสงค์ของขั้นตอนนี้คือการรวบรวม สถิติการทำงาน ลักษณะงานที่ผู้ใช้กระทำและส่วนที่เกี่ยวข้องทั้งหมดเพื่อมาวิเคราะห์และนำไปสู่การพัฒนาระบบให้ทันสมัยอยู่ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอน 1020 การพัฒนาและทำให้ระบบดีขึ้น (System Enhancement and Improvement)

การปรับปรุงระบบให้มีประสิทธิภาพและมีความทันสมัยสามารถแบ่งการปรับปรุงออกเป็น 2 ส่วน คือ การปรับปรุงระบบและการทำให้ระบบมีความทันสมัย การปรับปรุงระบบ คือ การทำให้ระบบมีการทำงานได้มากขึ้น รองรับอินพุต-เอาต์พุตและติดต่อกับระบบที่เกี่ยวข้องได้รวดเร็วขึ้น ซึ่งการปรับปรุงระบบนั้นผู้วิเคราะห์จะปรับปรุงจาก ปัญหาทางธุรกิจและความต้องการของผู้ใช้ ส่วนการทำให้ระบบมีความทันสมัยหรือการทำรีเอ็นจิเนียริ่ง (Reengineering) ที่ทำการเปลี่ยนแปลงระบบเดิมที่มีอยู่ไปสู่รูปแบบใหม่ ถ้าพิจารณาจะคล้ายกับการทำการวิเคราะห์ของขั้นตอนที่ 310 (ส่วนของการทำ BPR) ซึ่งการปรับปรุงระบบจะเปลี่ยนไปสู่สิ่งใหม่ เช่น การเปลี่ยนรูปแบบของภาษาที่พัฒนา ส่วนของการติดต่อกับผู้ใช้ ง่ายต่อการใช้งาน หรือรูปแบบการจัดเก็บข้อมูลแบบใหม่ เป็นต้น ซึ่งขั้นตอนการทำงานของระยะที่ 10 นี้มีดังต่อไปนี้

งานของขั้นตอน 1020

งานที่ 10 วิเคราะห์ความต้องการของผู้ใช้ คือ การนำความต้องการของผู้ใช้มาพิจารณาและสร้างกระบวนการใหม่ ๆ ให้กับระบบ ซึ่งการวิเคราะห์และสร้างกระบวนการนั้นเหมือนกับการกลับไปสู่จุดเริ่มต้นของการพัฒนาระบบอีกครั้ง

งานที่ 20 สร้างกระบวนการและเขียนโปรแกรมตามความต้องการของผู้ใช้

งานที่ 30 พิจารณาโครงสร้างข้อมูลหรือการจัดเก็บฐานข้อมูลแบบใหม่ การเก็บข้อมูลให้กับระบบสารสนเทศนั้นบางระบบจัดเก็บข้อมูลในรูปแบบเดิม เช่น VSAM หรือ ISAM เป็นต้น เมื่อเทคโนโลยีของการจัดการฐานข้อมูลเปลี่ยนแปลไป เช่น ฐานข้อมูลเชิงสัมพันธ์ ฐานข้อมูลเชิงวัตถุ เป็นต้น ทำให้กระบวนการจัดเก็บข้อมูลมีความสะดวกและถูกต้องมากขึ้น ดังนั้น ผู้วิเคราะห์และออกแบบควรให้ความช่วยเหลือกับผู้ใช้เพื่อพัฒนาระบบสารสนเทศที่มีอยู่ให้มีประสิทธิภาพมากขึ้น ถ้าฐานข้อมูลเดิมของระบบสารสนเทศอยู่บนสภาพแวดล้อมเครือข่ายการปรับปรุงฐานข้อมูลนั้นจำเป็นต้องอาศัยผู้เชี่ยวชาญทางด้านเครือข่ายให้คำแนะนำและปรึกษาให้กับระบบเพื่อให้โครงสร้างข้อมูลที่จะเกิดขึ้นใหม่มีผลกระทบต่อระบบเดิมน้อยที่สุดและมีความถูกต้องมากที่สุด เพราะการปรับปรุงโครงสร้างข้อมูลบนสภาพแวดล้อมเครือข่ายความถูกต้องของการเข้าถึงข้อมูลจำเป็นอย่างยิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 40 วิเคราะห์โปรแกรมและต้นทุนของการพัฒนาโปรแกรม การวิเคราะห์ต้นทุนและคุณภาพของโปรแกรมเป็นงานที่เหมือนกับขั้นตอนที่ 670 และ 740 ซึ่งงานของขั้นตอนทั้งสองนั้น วัดคุณภาพทั้งแบบโครางสร้างและแบบเชิงวัตถุ ซึ่งผู้วิเคราะห์และพัฒนาจำเป็นต้องปฏิบัติงานดังกล่าวให้ครบถ้วน กล่าวคือการพัฒนาโปรแกรมใหม่หรือปรับปรุงของเดิมให้กับระบบเมื่อพัฒนาเสร็จสิ้นจำเป็นต้องวัดคุณภาพของโปรแกรมนั้น ซึ่งการวัดคุณภาพของโปรแกรมตามหลักการทางวิศวกรรมซอฟต์แวร์ คือ การทำซอฟต์แวร์เมตริก์ (Software Metrics) การวัดจะวัดตั้งแต่ คอนโทรลโฟลว์ ค่าความซับซ้อนของโปรแกรและทำการวัดเฉพาะโปรแกรมที่ถูกพัฒนาขึ้นมาใหม่เพื่อตอบสนองความต้องการของผู้ใช้เท่านั้น

งานที่ 50 การทดสอบโปรแกรม คือ การทดสอบสิ่งที่ได้กระทำมาตั้งแต่งานที่ 1ถึง งานที่ 4 เพื่อตรวจสอบโปรแกรมและโครงสร้างข้อมูลที่ได้ปรับปรุงและพัฒนาขึ้นมาใหม่นั้นมีความสอดคล้องและสามารถทำงานได้กับระบบเดิมที่หรือเข้ากันได้กับระบบอื่นหรือไม่ ประเภทการทดสอบโปรแกรมแบ่งออกเป็น 3 ชนิด คือ ทดสอบความเชื่อมต่อและ/หรือการรวมกันของโปรแกรมที่พัฒนา ซึ่งการทดสอบประเภทนี้จะทดสอบทางด้านตรรกะของโปรแกรม การทดสอบชนิดที่ 2 คือ การทดสอบความเปลี่ยนแปลง การทดสอบประเภทนี้จะทดสอบในกรณีที่โปรแกรมเดิมมีเปลี่ยนแปลงภาษาในการพัฒนา การทดสอบประเภทสุดท้าย คือ การทดสอบองค์ประกอบของโปรแกรม ส่วนใหญ่การทดสอบแบบนี้จะทดสอบในส่วนของโปรแกรมย่อย เช่น โพรซีเจอร์หรือฟังก์ชันย่อยที่อยู่ภายในโปรแกรมหลัก เป็นต้น การทดสอบแบบสุดท้ายนี้เป็นการเลือกปฏิบัติ ซึ่งการทดสอบนี้บางครั้งอาจง่ายต่อการทดสอบแต่ถ้าระบบเดิมถูกพัฒนาด้วยภาษาโคบอลการทดสอบแบบนี้มักเป็นปัญหาเช่นกัน เพราะภาษาโคบอลนั้นเป็นภาษาที่มีละเอียดมากและยากต่อการตรวจให้ครบถ้วนได้ การทดสอบโปรแกรมทั้งหมดที่กล่าวมานั้นเป็นส่วนที่ผู้วิเคราะห์และออกแบบควรกระทำทุกครั้งมีการปรับปรุงและพัฒนาระบบสารสนเทศ เพราะการทดสอบนั้นจะทำให้ผู้ใช้และผู้พัฒนานั้นมีความมั่นใจกับระบบที่ตนใช้อยู่มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 สรุป

จากที่กล่าวมาข้างต้นโมเดลประยุกต์ คือ การนำขั้นตอนการพัฒนาระบบสารสนเทศของวิธีเอส เอส เอ ดี เอ็ม เข้ามาประยุกต์กับแบบจำลองน้ำพุนั้น บางระยะของการพัฒนาสามารถที่ละเลยได้ (เฉพาะระบบสารสนเทศที่เคยพัฒนาตามแบบจำลองนี้มาก่อน) ถึงแม้ว่าลำดับการพัฒนาของแต่ละระยะนั้นมีความเกี่ยวข้องกันทั้งวงจรชีวิตก็ตาม ถ้ามีการปรับปรุงและ/หรือแก้ไขระบบเพียงบางส่วนสามารถละเลยบางระยะได้และลำดับการพัฒนายังมีความคาบเกี่ยวและเกี่ยวเนื่องกันเช่นเดิม ขั้นตอนการพัฒนาระบบสารสนเทศตามโมเดลที่นำเสนอนี้ บางครั้งอาจมีขั้นตอนและลำดับการทำงานเป็นจำนวนมากก็ตามแต่ขั้นตอนเหล่านี้ย่อมส่งผลให้ผู้พัฒนาระบบ เจ้าโครงการและผู้ใช้ สามารถติดตามความคืบหน้าของพัฒนาได้ตลอดโครงการอย่างชัดเจนและสามารถนำกลับไปแก้ไขได้ง่ายและรวดเร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

กรณีศึกษาระบบคลังข้อสอบ

จากบทที่ 5 กล่าวถึงวิธีการและขั้นตอนการพัฒนาระบบสารสนเทศโดยใช้โมเดลประยุกต์ ส่วนบทที่ 6 กล่าวถึงการนำขั้นตอนต่าง ๆ มาทดลองพัฒนาระบบสารสนเทศ โดยระบบสารสนเทศที่เลือกมาทำการทดลองเป็นระบบที่คอยจัดเก็บ บันทึก และแก้ไขข้อสอบของสถาบันการศึกษา เรียกว่า "ระบบคลังข้อสอบ (Item Bank System)" ระบบคลังข้อสอบที่นำมาทดลองจะถูกพัฒนาตามขั้นตอนของโมเดลประยุกต์ตั้งแต่ระยะที่ 0 ถึงระยะที่ 10 ทำให้ระบบที่นำมาทดลองมีความชัดเจนหลายส่วน เช่น ความต้องการของผู้ใช้ ฟังก์ชันการทำงาน พฤติกรรมของผู้ใช้และระบบ เป็นต้น ซึ่งระบบที่ถูกพัฒนาหรืออยู่ระหว่างการพัฒนาระบบสามารถแสดงขั้นตอนพร้อมกับสามารถติดตามขั้นตอนการพัฒนาของโมเดลที่ออกแบบได้อย่างชัดเจน

การทดลองการพัฒนาระบบจะถูกนำเสนอตามระยะ ขั้นตอนและงานตามรูปแบบของโมเดลประยุกต์ จุดเริ่มต้นของการพัฒนาจะมีส่วนคล้ายกับรูปแบบการพัฒนาโดยทั่วไป (การเริ่มต้นพัฒนาระบบของทุกวิธีมักจะเริ่มต้นที่คล้ายกันเสมอ) แต่หลังจากผ่านโมดูลที่ 1 รูปแบบและขั้นตอนจะเริ่มมีความแตกต่างจากรูปแบบเดิม เพราะขั้นตอนการพัฒนาจะเริ่มเปลี่ยนจากรูปแบบการพัฒนาตามวิธีเชิงโครงสร้างไปสู่รูปแบบการพัฒนาตามวิธีเชิงวัตถุ โดยไม่ส่งผลกระทบกับสิ่งที่ปฏิบัติมาก่อนหน้านี้ (ลำดับการพัฒนากล่าวในหัวข้อที่ 6.1)

ระบบคลังข้อสอบที่นำมาทดลองกับแบบจำลองประยุกต์เป็นเพียงต้นแบบของระบบสารสนเทศหนึ่งเท่านั้น ดังนั้นระบบคลังข้อสอบอาจไม่ครอบคลุมในทุกขั้นตอนของการพัฒนาและสามารถแสดงการพัฒนาระบบคลังข้อสอบดังต่อไปนี้

6.1 ระยะที่ 0

สถาบันการศึกษาเป็นหน่วยงานที่มีกิจกรรมหลักคือ การเรียนการสอนและกิจกรรมสำคัญที่ต้องมีการควบคู่กับการเรียนการสอนคือการวัดผลการศึกษาหรือการสอบ ดังนั้นการออกข้อสอบจึงเป็นปัจจัยสำคัญที่ต้องมีการกระทำและพิจารณาถึงขั้นตอนและวิธีการให้เหมาะสม ซึ่งการออกข้อสอบแบบปรนัยก็เป็นวิธีการออกข้อสอบแบบหนึ่งที่นิยมใช้ในการศึกษาทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติและความเป็นมาของระบบ (Overview of History and Current System)

ระบบการออกข้อสอบในลักษณะปรนัยของสถาบันศึกษาตั้งแต่อดีตจนถึงปัจจุบันโดยส่วนใหญ่จะกระทำโดยอาจารย์ผู้สอน ซึ่งอาจออกโดยอาจารย์คนเดียวหรือหลายคนขึ้นอยู่กับจำนวนของอาจารย์ผู้สอนในวิชานั้น ๆ ลักษณะการออกข้อสอบจะใช้บทเรียนที่สอนเป็นตัวกำหนดเนื้อหาของข้อสอบ และบทเรียนในส่วนใหญ่จะมีรูปแบบที่เป็นบท ๆ ดังนั้นอาจารย์ผู้สอนจะทำการออกข้อสอบโดยการกำหนดจำนวนข้อที่จะออกทั้งหมดรวมถึงการแบ่งจำนวนข้อในแต่ละบท โดยจะมีความยากง่ายของข้อสอบให้เหมาะสมกับบทเรียนที่ทำการสอนในแต่ละบท

ส่วนการนำคอมพิวเตอร์เข้ามาช่วยในการจัดทำข้อสอบในปัจจุบันส่วนใหญ่จะเป็นลักษณะการใช้เพื่อการพิมพ์และเก็บข้อมูลเท่านั้น ซึ่งทำให้เกิดประโยชน์ในด้านความรวดเร็ว ตลอดจนได้รูปแบบข้อสอบที่มีความสวยงามและระเบียบเรียบร้อย

ปัญหาของระบบ (Problem)

1. ข้อสอบขาดความเป็นมาตรฐาน เนื่องจากไม่มีการกำหนดมาตรฐานความยากง่ายของตัวข้อสอบ อันเป็นผลให้การสอบแต่ละครั้งในรายวิชาเดียวกันมีความยากง่ายไม่เท่ากัน
2. เกิดความล่าช้าในการออกข้อสอบ อันเนื่องจากต้องทำการออกข้อสอบใหม่ทุกครั้งเมื่อมีการจัดสอบ ตลอดจนคัดเลือก ความยากง่ายให้เหมาะสมกับเนื้อหาที่สอนในวิชานั้น ๆ
3. ข้อสอบที่ใช้สอบนั้น หลังจากสอบผ่านไปแล้วไม่มีการเก็บรวบรวมไว้อย่างเป็นรูปแบบ อันเป็นเหตุให้การออกข้อสอบใหม่ต้องเสียเวลาในการค้นหาถ้าหากต้องการนำกลับมาใช้ใหม่หรือเป็นแบบอย่าง
4. เกิดการออกข้อสอบซ้ำกับข้อสอบที่เคยออกแล้วครั้งก่อนบางส่วนหรือทั้งหมดอันเป็นสาเหตุให้การวัดผลนักศึกษามีประสิทธิภาพไม่เพียงพอ
   - ข้อสอบไม่มีคำตอบที่ถูกต้องกับโจทย์
   - ข้อสอบมีคำตอบที่ถูกต้องมากกว่าหนึ่งข้อ

ความต้องการของระบบ (Requirements)

1.สามารถเก็บข้อสอบของแต่ละรายวิชาที่เปิดทำการเรียนการสอนของแต่ละปีการศึกษา หลักการเก็บบันทึกข้อสอบของสถาบันการศึกษาถูกจัดเก็บแยกตามหมวด เช่น วิทยาศาสตร์ วิศวกรรมศาสตร์ เกษตรศาสตร์ และอื่น เป็นต้น จากนั้นแยกรายวิชาตามหมวดที่กำหนด เช่น หมวดวิทยาศาสตร์ (วิชาคณิตศาสตร์ วิทยาศาสตร์ประยุกต์ ฟิสิกส์ เคมี) หมวดวิศวกรรมศาสตร์ หมวดเกษตรศาสตร์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ข้อสอบของแต่ละวิชาที่ถูกจัดเก็บตามหมวดหมู่สามารถแยกประเภทข้อสอบออกเป็น 3 ประเภท ประเภทแรก ข้อสอบปรนัย คือ ข้อสอบแบบเลือกคำตอบที่ถูกต้อง ซึ่งตัวข้อสอบจะประกอบด้วยคำถามและตัวเลือกที่เป็นคำตอบที่ถูกที่สุดเพียงข้อเดียว ถัดมาเป็นข้อสอบอัตนัย คือ ข้อสอบแบบบรรยาย ตัวข้อสอบมีเพียงคำถามเพียงอย่างเดียว ประเภทสุดท้ายข้อสอบแบบถูกผิด คือ ข้อสอบที่ให้ใส่เครื่องหมายถูกและผิดตามที่โจทย์แสดง และคำตอบของข้อสอบแต่ละประเภทจะถูกจัดเก็บพร้อมกัน

3. การสร้างข้อสอบแต่ละวิชาของแต่ละประเภทข้อสอบไม่สามารถนำมาใช้ติดต่อกัน 2 เทอมไม่ได้ เพื่อป้องกันไม่ให้ข้อสอบรั่วไหล

4. การกำหนดเกณฑ์ข้อสอบต้องจะถูกกำหนดโดยผู้สอนหรือผู้ทรงคุณวุฒิของแต่ละรายวิชา ซึ่งถ้าเป็นสถานศึกษากำกับของรัฐบาลผู้กำหนดเกณฑ์ควรมีตำแหน่งนักวิชาการระดับ 4 ขึ้นไป ส่วนสถานศึกษาเอกชนผู้กำหนดเกณฑ์อาจเป็นบุคคลคนเดียวกับผู้สอนรายวิชานั้น ๆ

วัตถุประสงค์ของระบบ (Objective)

1. ป้องกันข้อสอบรั่วไหล
2. เพื่อให้การออกข้อสอบมีมาตรฐานเดียวกันทั้งสถานศึกษา
3. เพื่อป้องกันการใช้ข้อสอบซ้ำกัน ณ เวลาเดียวกัน
4. เพื่อป้องกันการผิดพลาดของตัวข้อสอบและคำตอบ
5. เพื่อความรวดเร็วสำหรับการออกข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.1 แสดงรายละเอียด Context Diagram ของระบบเดิมทางกายภาพ

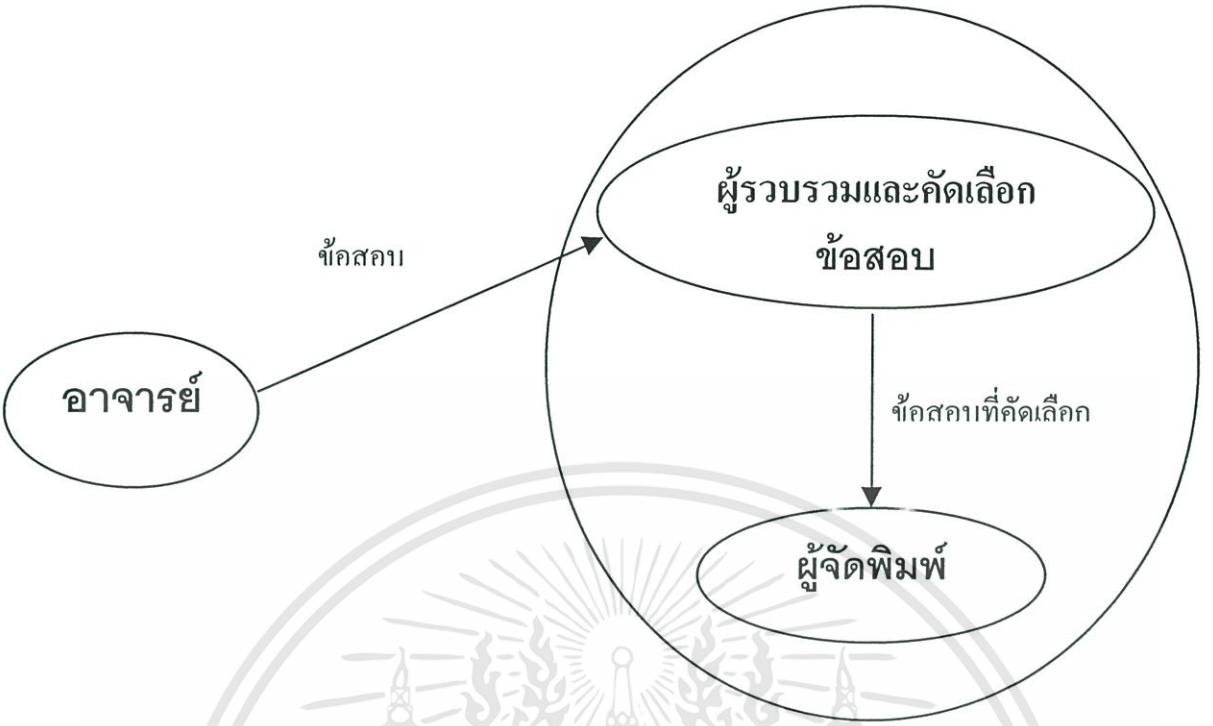
External Entity	Source/Recipient	Data Flow
อาจารย์	S	ข้อสอบ
	R	เกณฑ์ข้อสอบ

รูปที่ 6.1 แสดง Context Diagram ของระบบคลังข้อสอบ

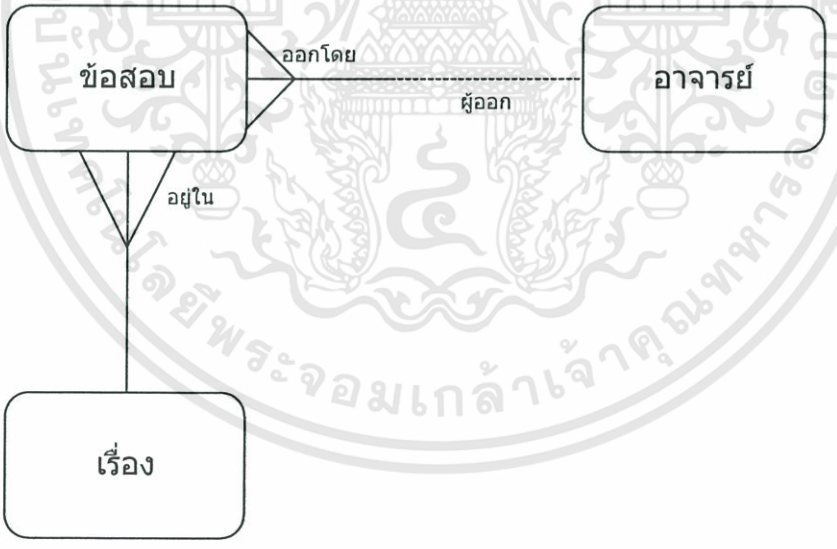
ตารางที่ 6.2 แสดงรายละเอียดการไหลเวียนของเอกสารของระบบเดิม

Source	Document	Recipient
อาจารย์	ข้อสอบ	ผู้รวบรวมและคัดเลือก
ผู้รวบรวมและคัดเลือก	ข้อสอบที่คัดเลือก	ผู้จัดพิมพ์
ผู้จัดพิมพ์	ชุดข้อสอบ	อาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

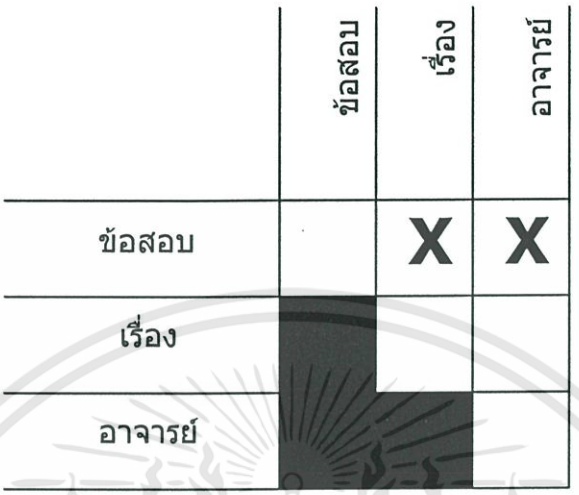


รูปที่ 6.2 แสดงรายละเอียดไดอะแกรมของการทำงานภายในระบบ



รูปที่ 6.3 แสดง LDS ไดอะแกรมของระบบเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 แสดงความสัมพันธ์ระหว่างเอนติตีของ LDS ไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.3 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 1

Source :	Priority : E	User Responsible : รศ.ดร. เอื้อน ปิ่นเงิน	Req.Id 01
Function Requirement : บันทึกข้อสอบฐานข้อมูล			
Non-Functional Requirement(S) :			
Description	Target Value	Acceptable Range	Comments
จำนวนเรื่อง	ไม่จำกัด		
จำนวนบท	ไม่จำกัด		
จำนวนข้อ	ไม่จำกัด		
คำตอบของข้อสอบ	4 หรือ 5 ข้อ		
Benefits : เพื่อรวบรวมข้อสอบลงฐานข้อมูล			
Comments/Suggested solution:			
Relate Documents:			
Related Requirements:			
Resolution:			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.4 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 2

Source :	Priority : E	User Responsible : รศ.ดร. เอื้อน ปิ่นเงิน	Req.Id 02
Function Requirement : ให้เกณฑ์ข้อสอบกับข้อสอบที่มีอยู่			
Non-Functional Requirement(S) :			
Description	**Target Value**	**Acceptable Range**	**Comments**
เกณฑ์ข้อสอบ	มี 4 อย่าง		1. ความยาก 2. ความตรง 3. ความน่าเชื่อถือ 4. อำนาจจำแนก
ถ้าไม่ใส่เกณฑ์ข้อสอบ	ให้ค่าเป็น 0		
Benefits : ได้ข้อสอบที่มีเกณฑ์วัดมาตรฐาน			
Comments/Suggested solution:			
Relate Documents: No. 1			
Related Requirements:			
Resolution:			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.5 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 3

Source :	Priority : D	User Responsible : รศ.ดร. เอื้อน ปิ่นเงิน	Req.Id 03
Function Requirement : ลบข้อสอบที่ไม่ต้องการ			
Non-Functional Requirement(S) :			
Description	Target Value	Acceptable Range	Comments
การลบ	ลบไม่ได้จำกัดข้อ		ไม่มีการเปลี่ยนแปลง ID ของข้อสอบข้ออื่นๆ เพื่อเรียงลำดับใหม่
Benefits : ข้อสอบที่ไม่ต้องการจะถูกลบออกจากระบบ			
Comments/Suggested solution:			
Relate Documents:			
Related Requirements:			
Resolution:			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.6 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 4

Source :	Priority : E	User Responsible : รศ.ดร. เอื้อน ปิ่นเงิน	Req.Id 04
Function Requirement : คัดเลือกข้อสอบและพิมพ์			
Non-Functional Requirement(S) :			
Description	**Target Value**	**Acceptable Range**	**Comments**
วิธีการเลือก จำนวนที่ต้องการเลือก เกณฑ์ระดับข้อสอบ การพิมพ์	ระบุ เรื่อง วิชา บท จำนวนข้อ เกณฑ์ ระบุได้ตามต้องการ ระบุค่า 4 อย่าง พิมพ์รายงานจากที่เลือก		1. ความยาก 2. ความตรง 3. ความน่าเชื่อถือ 4. อำนาจจำแนก
Benefits : ข้อสอบที่ไม่ต้องการจะถูกลบออกจากระบบ			
Comments/Suggested solution:			
Relate Documents:			
Related Requirements:			
Resolution:			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.7 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 5

Source :	Priority : N	User Responsible : รศ.ดร. เอื้อน ปิ่นเงิน	Req.Id 05
Function Requirement : สอบถามข้อมูล			
Non-Functional Requirement(S) :			
Description	Target Value	Acceptable Range	Comments
การสอบถามข้อสอบ การสอบถามผู้ออกข้อสอบ การสอบถามผู้ให้เกณฑ์ การสอบถามประวัติข้อสอบ	- สรุปจำนวนข้อ - ดูเนื้อหาข้อสอบ - รายชื่อผู้ออกข้อสอบ - รายชื่อผู้ให้เกณฑ์ - ประวัติการใช้ข้อสอบ		
Benefits : ได้ทราบข้อมูลที่มีอยู่ในระบบ			
Comments/Suggested solution:			
Relate Documents:			
Related Requirements:			
Resolution:			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.8 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 6

Source :	Priority : N	User Responsible : รศ.ดร. เอื้อน ปิ่นเงิน	Req.Id 06
Function Requirement : เพิ่มข้อมูลอาจารย์หรือผู้ให้เกณฑ์			
Non-Functional Requirement(S) :			
Description	**Target Value**	**Acceptable Range**	**Comments**
เพิ่มชื่ออาจารย์	ชื่ออาจารย์		
Benefits : เพิ่มชื่อให้ครบในกรณีที่ไม่มีในฐานข้อมูล			
Comments/Suggested solution:			
Relate Documents:			
Related Requirements: None			
Resolution:			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

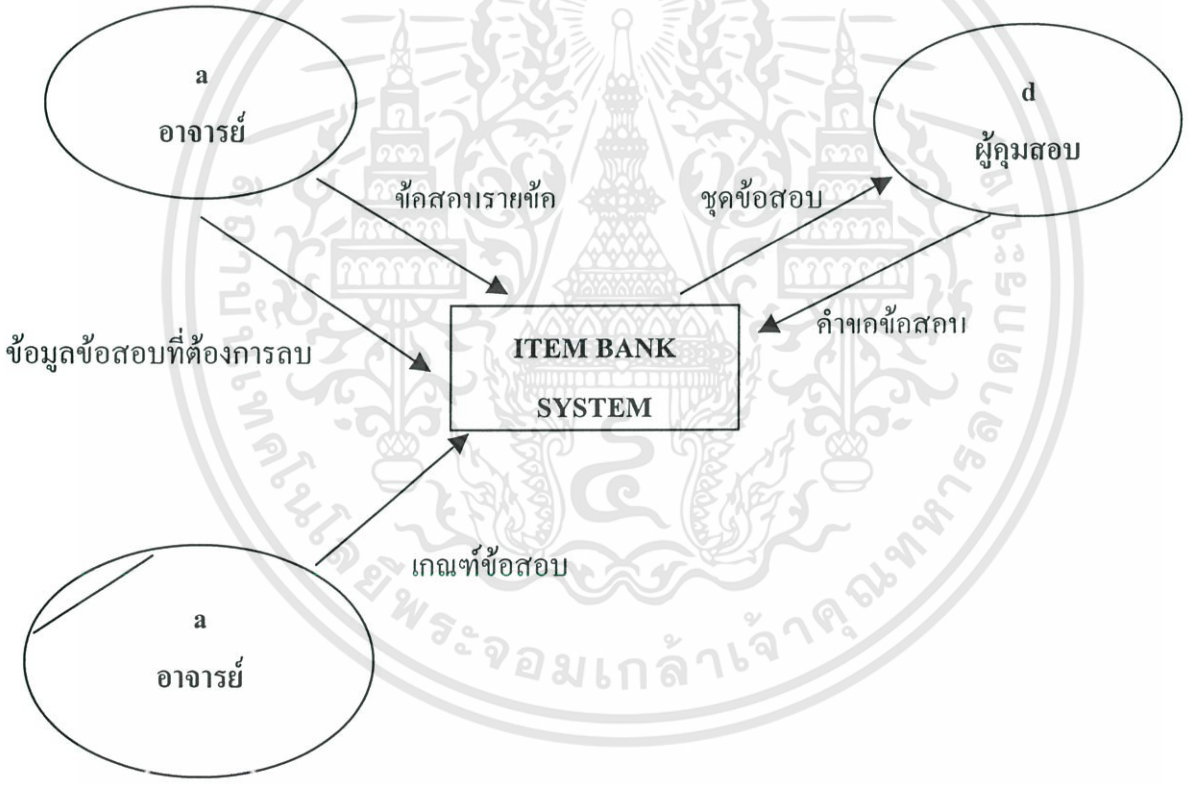
ตารางที่ 6.9 แสดงรายละเอียดความต้องการของผู้ใช้ลำดับที่ 7

Source :	Priority : N	User Responsible : รศ.ดร. เอื้อน ปิ่นเงิน	Req.Id 07
Function Requirement : แก้ไขข้อมูลอาจารย์หรือผู้ให้เกณฑ์			
Non-Functional Requirement(S) :			
Description	Target Value	Acceptable Range	Comments
แก้ไขชื่ออาจารย์	ชื่ออาจารย์		
Benefits : แก้ไขชื่อให้ถูกต้องในกรณีที่มีการเปลี่ยนชื่อหรือบันทึกผิด			
Comments/Suggested solution:			
Relate Documents:			
Related Requirements: None			
Resolution:			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.10 แสดงรายละเอียดของ Context Diagram ส่วนความต้องการของระบบเดิม

External Entity	Source/Recipient	Data Flow
อาจารย์	S	ข้อสอบรายข้อ
	S	ข้อมูลข้อสอบที่ต้องการลบ
	S	เกณฑ์ข้อสอบ
ผู้คุมสอบ	S	คำขอข้อสอบ
	R	ชุดข้อสอบ

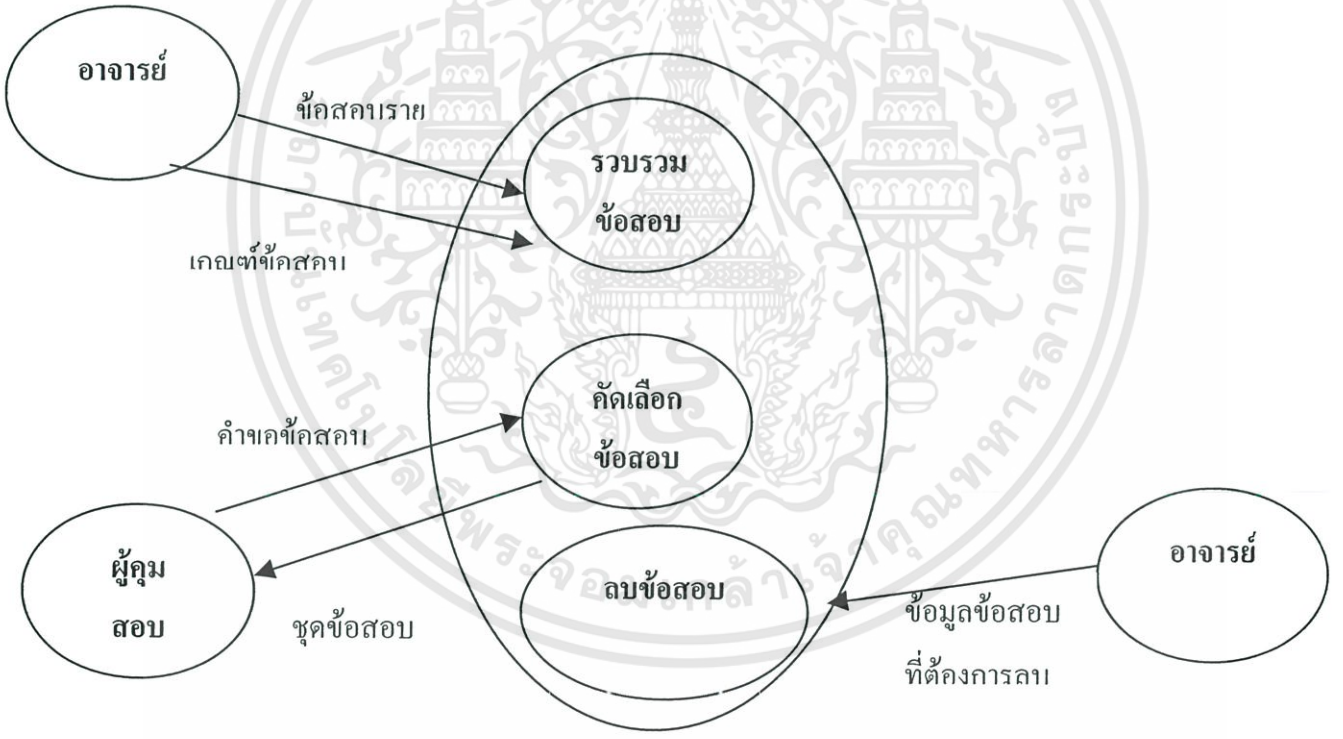


รูปที่ 6.5 แสดง context diagram ของระบบเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.11 แสดงการไหลเวียนของเอกสารของระบบที่ต้องการ

Source	Document	Recipient
อาจารย์	ข้อสอบรายข้อ	ผู้รวบรวมข้อสอบ
ผู้คุมสอบ	คำขอข้อสอบ	ผู้คัดเลือก
ผู้คัดเลือก	ชุดข้อสอบ	ผู้คุมสอบ
อาจารย์	เกณฑ์ข้อสอบ	ผู้รวบรวมข้อสอบ
อาจารย์	ข้อมูลข้อสอบที่ต้องการจะลบ	ผู้มีหน้าที่ลบข้อสอบ

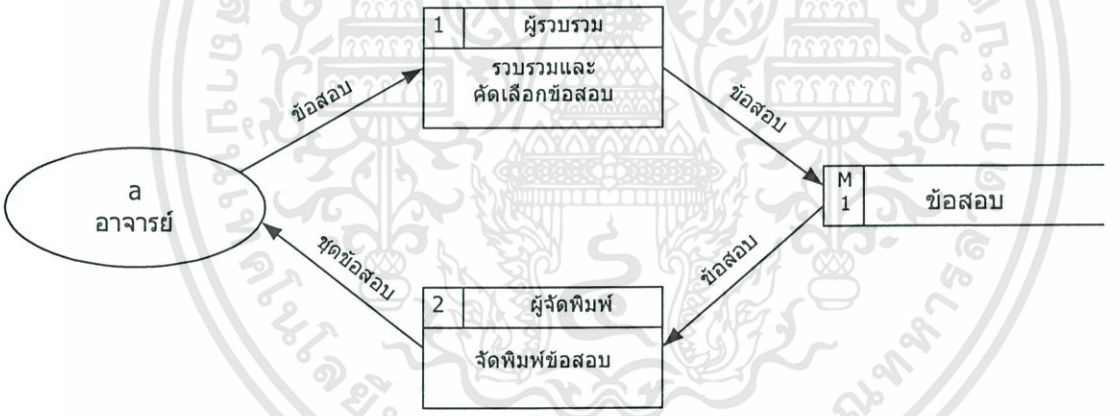


รูปที่ 6.6 แสดงการทำงานของเอนติตีภายนอกกับงานภายในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

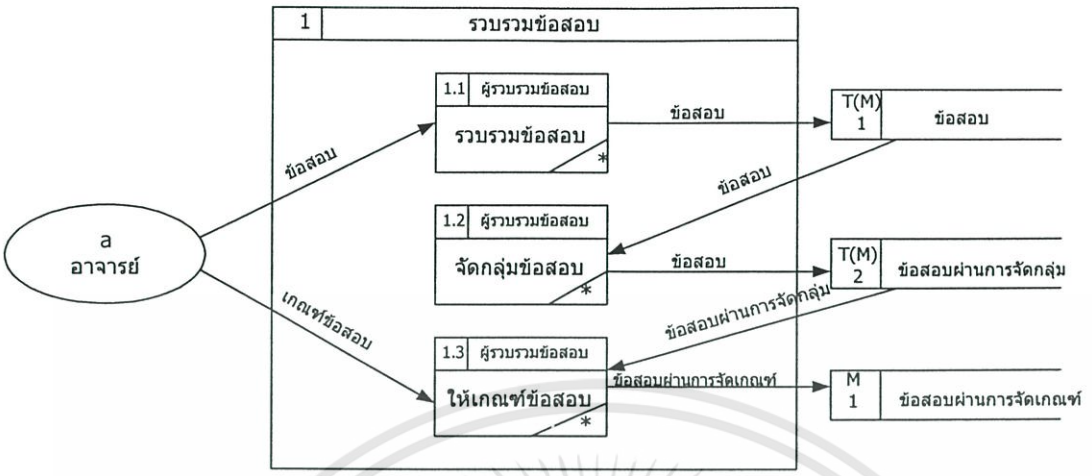
ตารางที่ 6.12 แสดงรายละเอียดของ User Catalogue

User Catalogue	
Job Title	Job Activities Description
ผู้รวบรวมข้อสอบ	รวบรวมข้อสอบและเกณฑ์ข้อสอบ บันทึกข้อสอบและเกณฑ์ข้อสอบลงฐานข้อมูล
ผู้คัดเลือกข้อสอบ	คัดเลือกข้อสอบที่ต้องการ พิมพ์ข้อสอบ
ผู้มีหน้าที่ลบข้อสอบ	ตรวจสอบข้อสอบที่ผิดพลาด ลบข้อสอบ

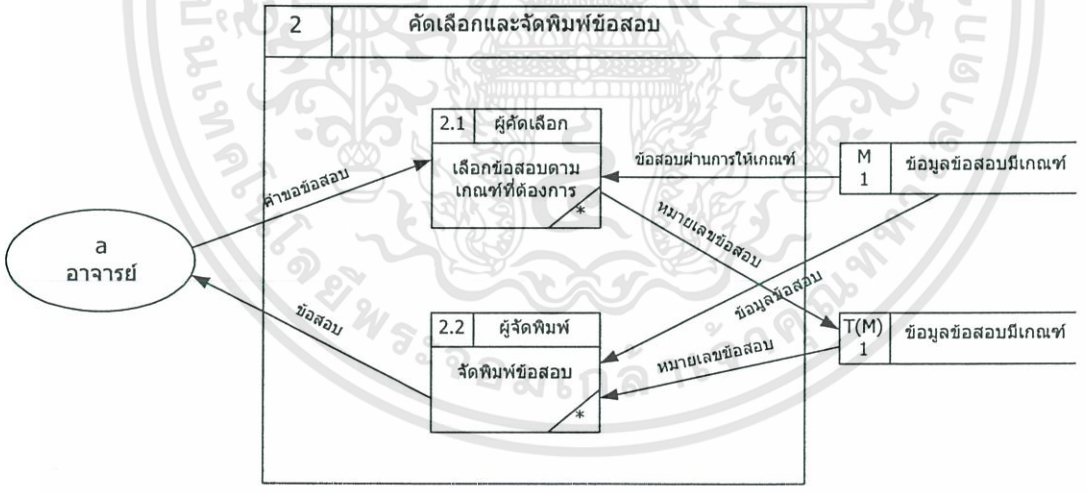


รูปที่ 6.7 แสดงแผนภูมิการไหลของข้อมูลทางกายภาพระดับที่ 1 ของระบบเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.8 แสดงแผนภูมิการไหลของข้อมูลทางกายภาพระดับที่ 2 ของระบบเดิม



รูปที่ 6.9 แสดงแผนภูมิการไหลของข้อมูลทางกายภาพระดับที่ 2 ของระบบเดิม (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.13 แสดงรายละเอียดของกระบวนการที่ 1.1

Process ID: 1.1
Process Name: รวบรวมข้อสอบ
Description: อาจารย์ให้ข้อมูลข้อสอบรายข้อซึ่งสามารถออกข้อสอบได้โดยอาจารย์หลายคน โดยข้อสอบดังกล่าวประกอบด้วยข้อสอบหลายวิชา ในแต่ละวิชามีได้หลายบท แต่ละบทมีข้อสอบหลายข้อ และแต่ละข้อจะมีคำตอบตัวเลือกได้ 4-5 ตัวเลือก

ตารางที่ 6.14 แสดงรายละเอียดของกระบวนการที่ 1.2

Process ID: 1.2
Process Name: จัดกลุ่มข้อสอบ
Description: จัดกลุ่มของข้อสอบให้ตรงตามหมวดหมู่ของแต่ละวิชา

ตารางที่ 6.15 แสดงรายละเอียดของกระบวนการที่ 1.3

Process ID: 1.3
Process Name: กำหนดเกณฑ์ข้อสอบ
Description: กำหนดเกณฑ์ให้กับข้อสอบที่ถูกสร้างไว้ เพื่อสามารถนำข้อสอบที่ถูกสร้างไว้มาใช้ประโยชน์ต่อไป

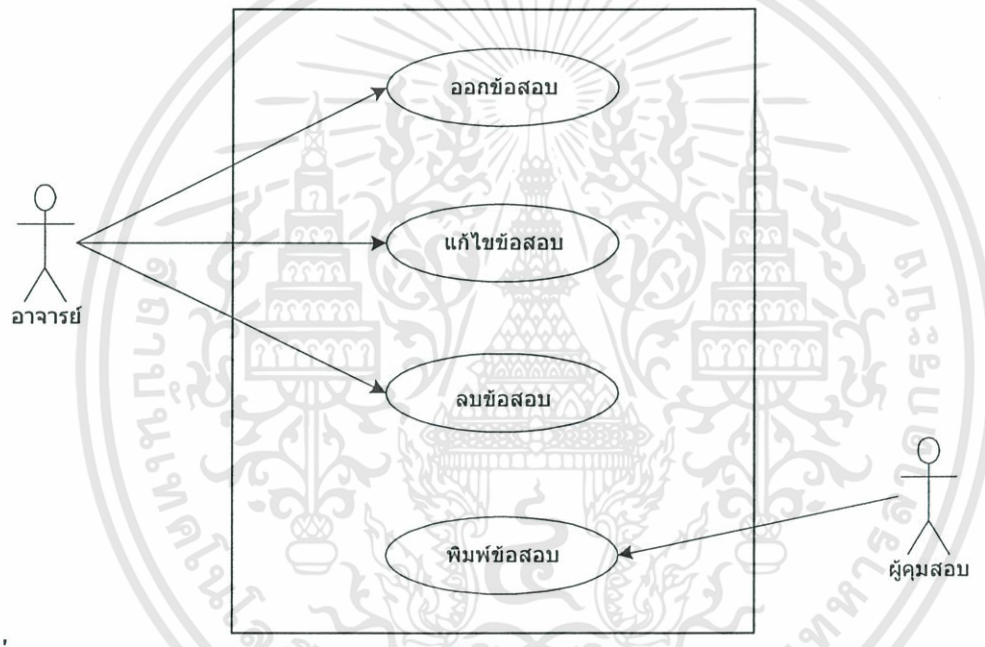
ตารางที่ 6.16 แสดงรายละเอียดของกระบวนการที่ 2.1

Process ID: 2.1
Process Name: เลือกข้อสอบ
Description: คัดเลือกข้อสอบตามความต้องหลังจากกำหนดเกณฑ์ข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.17 แสดงรายละเอียดของกระบวนการที่ 2.2

Process ID: 2.2
Process Name: จัดพิมพ์ข้อสอบ
Description: จัดพิมพ์ข้อสอบที่ถูกคัดเลือกไว้เรียบร้อยแล้ว



รูปที่ 6.10 แสดง USE CASE ของระบบคลังข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.18 แสดงรายละเอียดของแผนงานลำดับที่ 1

Scenario Name: การออกข้อสอบ	ID Number: _1_
Short Description: การออกข้อสอบและการบันทึกข้อสอบแต่ละวิชาของแต่ละบทที่ต้องการพร้อมกับให้เกณฑ์ข้อสอบ	
Trigger: ทำงานต่อเมื่อมีการสร้างข้อสอบจากอาจารย์ที่รับผิดชอบและกำหนดเกณฑ์ข้อสอบจากผู้ให้เกณฑ์ Type: **External** / Temporal	

Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
___สร้างข้อสอบ_____	อาจารย์	ข้อสอบที่ระบุประเภท	ข้อสอบรายข้อ
___กำหนดเกณฑ์ข้อสอบ___	ผู้กำหนดเกณฑ์	ข้อสอบผ่านการจัดเกณฑ์	ข้อสอบมีเกณฑ์
_____	_____	สร้างเกณฑ์ข้อสอบ	อาจารย์
_____	_____	รายละเอียดผู้ออกข้อสอบ	อาจารย์
_____	_____	รายละเอียดผู้ให้เกณฑ์ข้อสอบ	อาจารย์
_____	_____	_____	_____

Major Steps Performed	Information for Steps
1. สร้างข้อสอบตามบทและรายวิชาที่ต้องการ	ข้อสอบไม่ระบุประเภท
2. กำหนดประเภทข้อสอบ ข้อสอบแบ่งเป็น 3 ประเภท คือ ปรนัย อัตนัยและถูกผิด	ข้อสอบระบุประเภท
3. กำหนดเกณฑ์การข้อสอบโดยการให้เกณฑ์แบ่งเป็น 4 ชนิด คือ ความยาก ความตรง ความน่าเชื่อถือและอำนาจจำแนก	เกณฑ์ข้อสอบ
4. กำหนดเกณฑ์จากหลักการจัดเกณฑ์ให้กับข้อสอบแต่ละข้อ แต่ละบท ของแต่ละวิชา	ข้อสอบผ่านการจัดกลุ่ม
5. บันทึกข้อสอบที่ให้เกณฑ์ลงสู่ฐานข้อมูล	ข้อมูลข้อสอบผ่านการจัดเกณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.19 แสดงรายละเอียดของแผนงานลำดับที่ 2

Scenario Name: การคัดเลือกข้อสอบ	ID Number: _2_
Short Description: การเลือกข้อสอบราชวิชาเพื่อออกสอบประจำภาคเรียน	
Trigger: __ทำงานต่อเมื่อถึงเวลาสอบของแต่ละภาคเรียน_____ Type: **External** / Temporal	

Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
___คัดเลือกข้อสอบ_____	อาจารย์	_ชุดข้อสอบที่ผ่านการเลือก_	ผู้คุมสอบ
___คำร้องขอรับข้อสอบ____	ผู้คุมสอ บ	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____		

Major Steps Performed	Information for Steps
1. รับคำร้องเพื่อจัดเตรียมข้อสอบ	ผู้คุมสอบ
2. คัดเลือกข้อสอบจากฐานข้อมูลข้อสอบที่มีเกณฑ์	ผู้อาจารย์
3. บันทึกข้อสอบที่ถูกเลือก	ประวัติการใช้ข้อสอบ
4. นำส่งข้อสอบ	ผู้คุมสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.20 แสดงรายละเอียดของแผนงานลำดับที่ 3

Scenario Name: การลบข้อสอบ	ID Number: _3__
Short Description: การลบข้อสอบจากฐานข้อมูลข้อสอบ การลบทำกรณีที่ข้อสอบนั้นล้าสมัยหรือถูกใช้งานซ้ำ ๆ กันหลายครั้งเกินไป	
Trigger: __ทำงานเมื่อคำขอจากอาจารย์ผู้รับผิดชอบรายวิชาหรือจากผู้ให้เกณฑ์ข้อสอบ_____ Type: **External** / Temporal	

Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
ข้อสอบรายวิชาที่ต้องการลบ	อาจารย์	____แสดงผลการดำเนินงาน____	อาจารย์
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____		
_____	_____		

Major Steps Performed	Information for Steps
1. รับรายละเอียดข้อสอบเพื่อทำการลบ เช่น วิชา ประเภทข้อสอบ บทที่ เป็นต้น	ข้อสอบมีเกณฑ์
2. คัดเลือกข้อสอบที่ต้องการลบ	ข้อสอบมีเกณฑ์
3. ลบประวัติการใช้ข้อสอบที่ถูกคัดเลือก	ประวัติการใช้ข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.21 แสดงรายละเอียดของแผนงานลำดับที่ 4

Scenario Name: การให้เกณฑ์ข้อสอบ	ID Number: 4
Short Description: คือการกำหนดเกณฑ์ของข้อสอบในแต่ละวิชาและแต่ละบทที่มีการออกข้อสอบ	
Trigger: ทำงานต่อเมื่อมีการสร้างข้อสอบจากอาจารย์และมีการจัดกลุ่มข้อสอบ Type: **External** / Temporal	

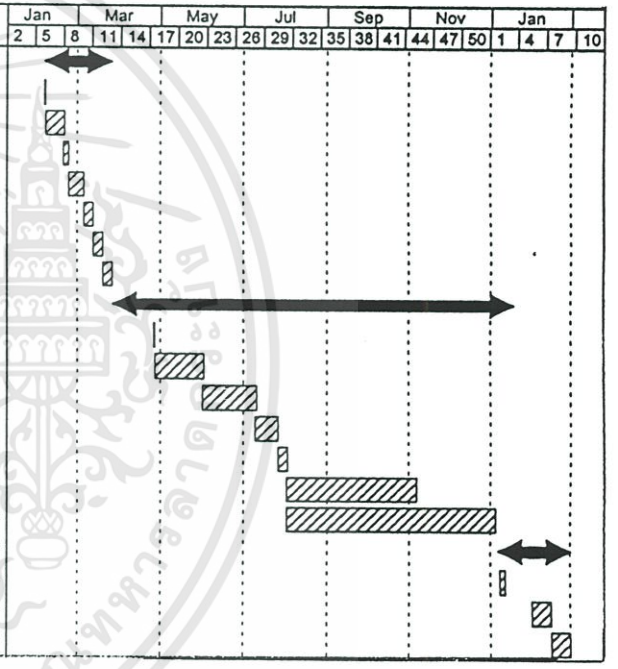
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
เกณฑ์ข้อสอบ	อาจารย์	ข้อมูลข้อสอบผ่านการจัดเกณฑ์	ข้อสอบมีเกณฑ์
ข้อสอบผ่านการจัดกลุ่ม	ผู้จัดกลุ่ม		

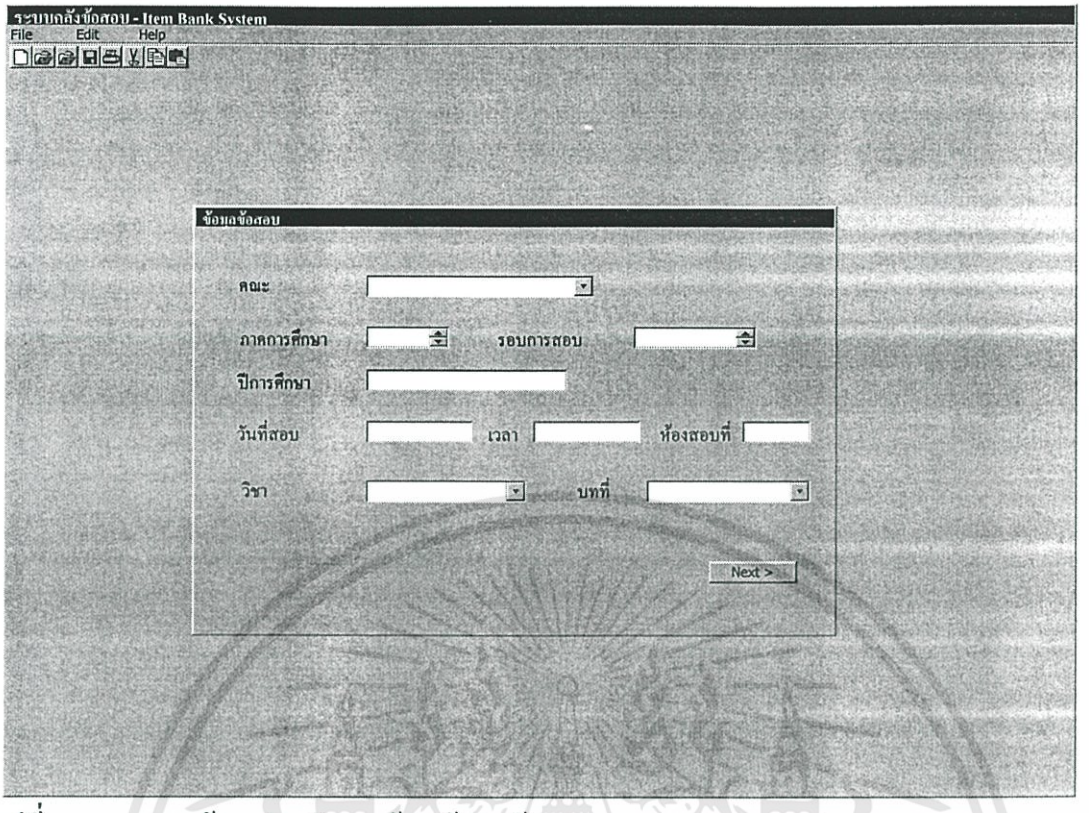
Major Steps Performed	Information for Steps
1. กำหนดเกณฑ์การข้อสอบโดยการให้เกณฑ์แบ่งเป็น 4 ชนิด คือ ความยาก ความตรง ความน่าเชื่อถือและอำนาจจำแนก	เกณฑ์ข้อสอบ
2. กำหนดเกณฑ์จากหลักการจัดเกณฑ์ให้กับข้อสอบแต่ละข้อ แต่ละบท ของแต่ละวิชา	ข้อสอบผ่านการจัดกลุ่ม
3. บันทึกข้อสอบที่ให้เกณฑ์ลงสู่ฐานข้อมูล	ข้อมูลข้อสอบผ่านการจัดเกณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

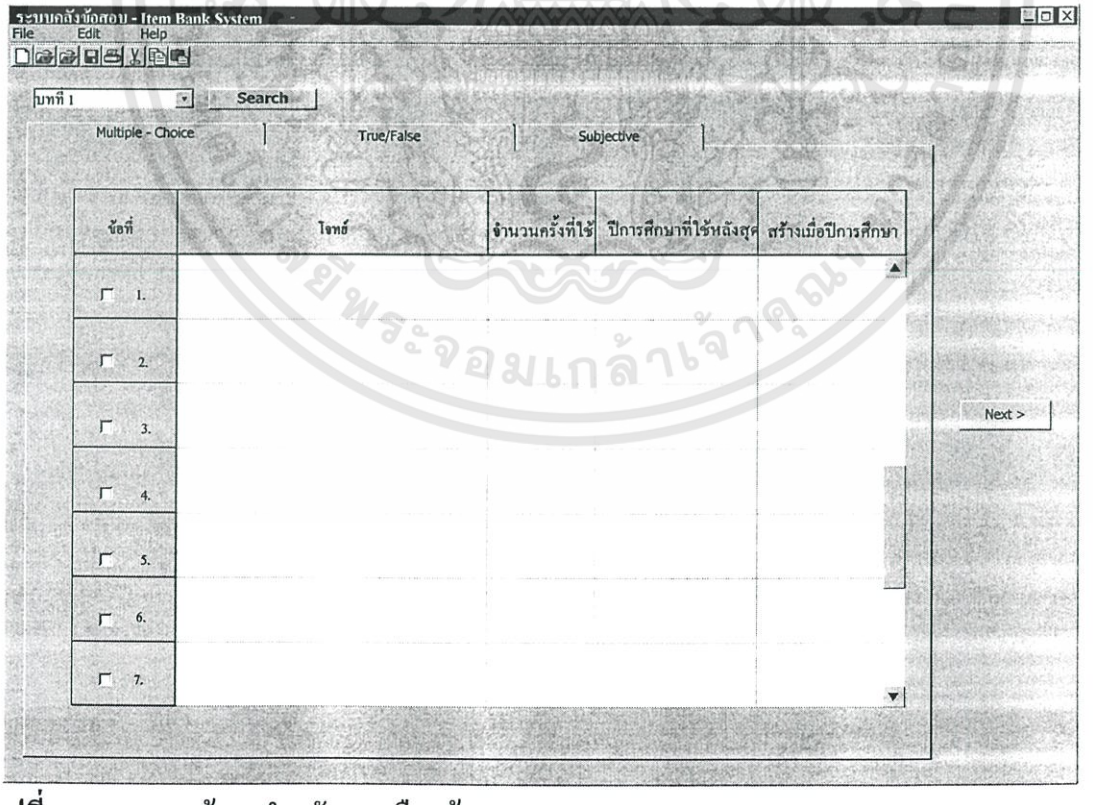
ตารางที่ 6.22 แสดงตารางระยะเวลาดำเนินงานการพัฒนาระบบคลังข้อสอบ

ID	Task Name	Start	Finish
1	**I. Analysis of the Existing System**	Mon 5/2/01	Mon 26/3/01
2	Define the Objective and Scope	Mon 5/2/01	Mon 5/2/01
3	Study the Existing System	Tue 6/2/01	Mon 19/2/01
4	Identify The Existing Problem definition	Mon 19/2/01	Thu 22/2/01
5	Collect user's requirement	Fri 23/2/01	Mon 5/3/01
6	Develop Context Diagram	Tue 6/3/01	Mon 12/3/01
7	Develop Data Flow Diagram	Tue 13/3/01	Mon 19/3/01
8	Cost and Benefit Analysis	Tue 20/3/01	Mon 26/3/01
9	**II. Analysis and Design of the Proposed System**	Tue 27/3/01	Fri 18/1/02
10	Submitted Project Proposal	Thu 26/4/01	Thu 26/4/01
11	System analysis	Fri 27/4/01	Fri 1/6/01
12	System Design	Fri 1/6/01	Tue 10/7/01
13	New system design and Conver Method	Tue 10/7/01	Thu 26/7/01
14	Testing Analysis and Design	Fri 27/7/01	Thu 2/8/01
15	Hardware and Software Precrement	Thu 2/8/01	Tue 6/11/01
16	Coding and Testing	Thu 2/8/01	Fri 4/1/02
17	**III. Implementation of the Proposed System**	Mon 7/1/02	Fri 1/3/02
18	System Implement	Tue 8/1/02	Fri 11/1/02
19	conversion	Fri 1/2/02	Thu 14/2/02
20	Install Application and System	Fri 15/2/02	Thu 28/2/02



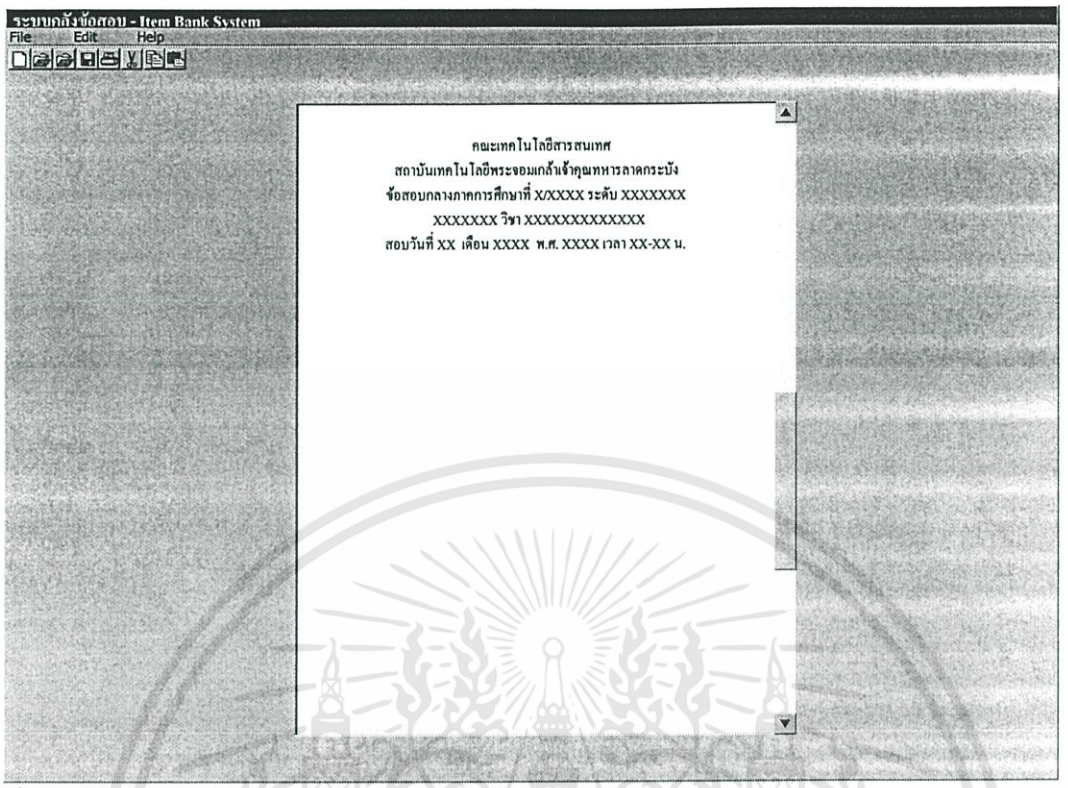


รูปที่ 6.11 แสดงหน้าจอสำหรับการป้อนข้อมูลข้อสอบ

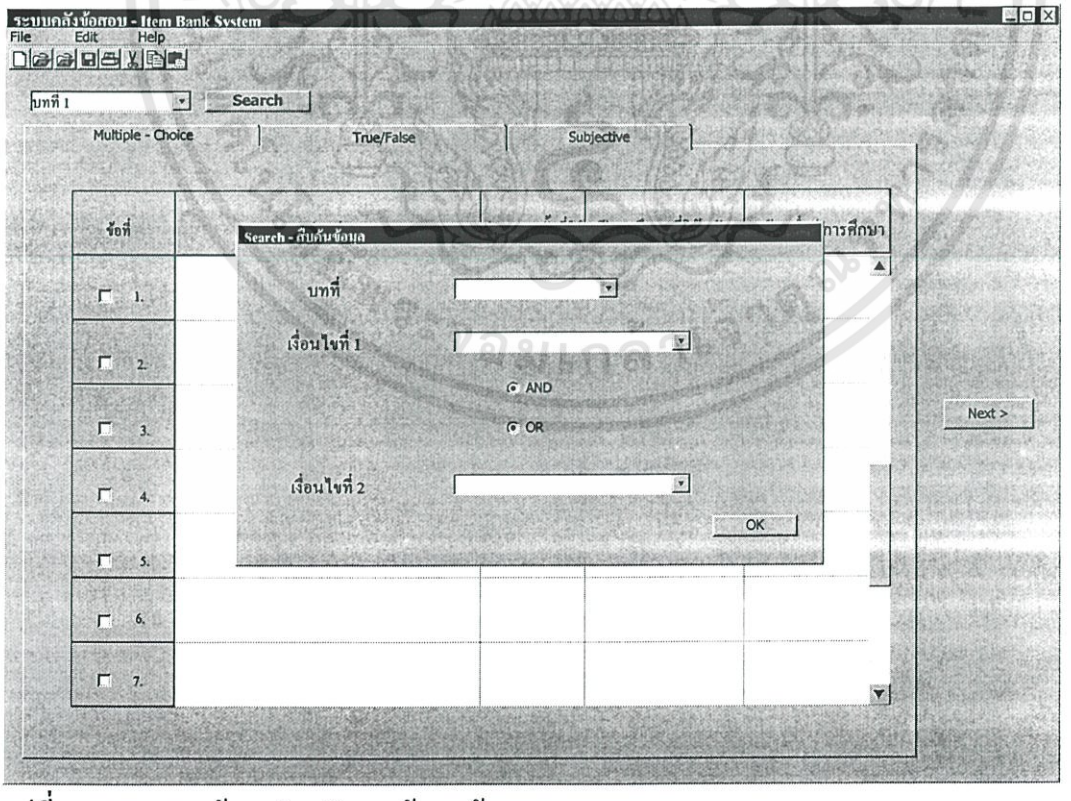


รูปที่ 6.12 แสดงหน้าจอสำหรับการเลือกข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.13 แสดงหน้าจอสำหรับพิมพ์ข้อสอบ



รูปที่ 6.14 แสดงหน้าจอสำหรับการค้นหาข้อสอบตามบท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางเลือกทางธุรกิจ (Business System Option -BSO)

ความต้องการเบื้องต้นของระบบ ITEM BANK มีดังนี้

1. บันทึกข้อมูลลงฐานข้อมูล
2. ให้เกณฑ์ข้อสอบกับข้อสอบที่มีอยู่
3. คัดเลือกข้อสอบและพิมพ์

สำหรับความต้องการเพิ่มเติมมีดังนี้

4. ลบข้อสอบที่ไม่ต้องการ
5. สอบถามข้อมูล

จากความต้องการข้างต้นสามารถให้ BSO ได้ดังนี้

BSO 1: ทำการเขียนโปรแกรมขึ้นเองเพื่อจัดการฐานข้อมูลเองทั้งหมด โดยโปรแกรมดังกล่าวสามารถเก็บบันทึกข้อมูล และสืบค้นข้อมูลเพื่อพิมพ์รายงานได้

BSO 2: ใช้โปรแกรมฐานข้อมูลเชิงสัมพันธ์ (RDBMS)ในการจัดการฐานข้อมูลและมีความสามารถในการติดต่อกับ DBMS และเป็นสร้าง Interface ให้กับผู้ใช้งานได้ โดยการทำดังกล่าวสามารถเก็บบันทึกข้อมูล และสืบค้นข้อมูลเพื่อพิมพ์รายงานได้

BSO 3: ใช้โปรแกรมฐานข้อมูลที่สามารถรองรับฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลเชิงวัตถุ (ODBMS) ได้และสามารถการจัดการฐานข้อมูลพร้อมกับมีความสามารถในการติดต่อกับฐานข้อมูลและมี Interface ติดต่อกับผู้ใช้งานได้ง่าย โดยการทำดังกล่าวสามารถเก็บบันทึกข้อมูลและสืบค้นข้อมูลเพื่อพิมพ์รายงานได้

ตารางที่ 6.23 แสดงความสัมพันธ์ของ BSO

	BSO 1	BSO 2	BSO 3
1. บันทึกข้อมูลลงฐานข้อมูล	X	X	X
2. ให้เกณฑ์ข้อสอบกับข้อสอบที่มีอยู่	X	X	X
3. คัดเลือกข้อสอบและพิมพ์	X	X	X
4. ลบข้อสอบที่ไม่ต้องการ	X	X	X
5. สอบถามข้อมูล	X	X	X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากความต้องการเบื้องต้นของแต่ละ BSO สามารถแสดง TSO เบื้องต้นได้ดังต่อไปนี้

TSO ที่ 1: ใช้คอมพิวเตอร์ส่วนบุคคลเพียงเครื่องเดียว สำหรับระบบและใช้โปรแกรมการจัดการฐานข้อมูลที่สนับสนุนการใช้ฐานข้อมูลเพียงผู้ใช้คนเดียวในเวลาหนึ่ง โปรแกรมที่นำมาใช้จะถูกพัฒนาขึ้นเองทั้งหมดทั้งส่วนการติดต่อระหว่างผู้ใช้กับระบบและส่วนของการจัดการฐานข้อมูล ซึ่งการพัฒนาขึ้นเองทั้งหมดทำให้ค่าใช้จ่ายที่เกิดขึ้นมีรายจ่ายที่สูงในส่วนของค่าแรงงานของโปรแกรมเมอร์ นอกจากนี้ยังมีผลกระทบกับระยะเวลาในการพัฒนาให้ระบบเสร็จทันเวลาที่ต้องการอีกด้วย

TSO ที่ 2: มีลักษณะเหมือนกับ TSO ที่ 1 แต่โปรแกรมที่ใช้สำหรับการพัฒนาฐานข้อมูลนั้นไม่จำเป็นต้องพัฒนาขึ้นเอง เพียงแต่เลือกผลิตภัณฑ์ที่มีอยู่ทั่วไปมาประยุกต์ใช้งาน ซึ่งระบบฐานข้อมูลที่เลือกนำมาใช้สำหรับทางเลือกนี้ใช้ระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งระบบนี้สามารถพบเห็นทั่วไปและเป็นผลิตภัณฑ์ที่ต้องเสียค่าใช้จ่ายจำนวนมากจนถึงไม่มีค่าใช้จ่าย ส่วนโปรแกรมที่ใช้ติดต่อกับผู้ใช้อาจใช้ภาษา 4GL ซึ่งเป็นภาษาที่สนับสนุนระบบฐานข้อมูลที่ถูกเลือก และการพัฒนาระบบของทางเลือกนี้จะสนับสนุนระบบเครือข่ายในอนาคตอีกด้วย

TSO ที่ 3: เป็นทางเลือกที่สนับสนุนระบบในกรณีที่ระบบนั้นต้องการขยายความสามารถในอนาคต ซึ่งทางเลือกนี้อาจมีค่าใช้จ่าย ทั้งฮาร์ดแวร์และซอฟต์แวร์มีจำนวนที่สูงกว่าทางเลือกอื่น เพราะระบบฐานข้อมูลและโปรแกรมที่นำมาใช้สำหรับการพัฒนานั้นสนับสนุนทั้งระบบสารสนเทศเชิงวัตถุและเชิงโครงสร้าง ซึ่งทำให้การพัฒนาระบบของผู้พัฒนานั้นมีความง่าย โดยสามารถเลือกตามความถนัดของผู้พัฒนาได้ และอาจทำให้ลดค่าใช้จ่ายสำหรับค่าแรงของโปรแกรมเมอร์ได้ นอกจากนี้หากเลือกพัฒนาแบบเชิงวัตถุอาจทำให้ระยะเวลาสำหรับการพัฒนาระบบนั้นมีระยะเวลาสั้นกว่าการพัฒนาแบบอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.24 แสดงตารางงบประมาณ แบบที่ 1

	2000(฿)	2001(฿)	2002(฿)	2003(฿)	2004(฿)	Total
Benefits						
registration	฿ 500,000.00	฿ 530,000.00	฿ 561,800.00	฿ 595,508.00	฿ 631,238.00	฿ 2,818,546.00
Improved customer service	฿ 70,000.00	฿ 70,000.00	฿ 70,000.00	฿ 70,000.00	฿ 70,000.00	฿ 350,000.00
Reduced inventory costs	฿ 68,000.00	฿ 68,000.00	฿ 68,000.00	฿ 68,000.00	฿ 68,000.00	฿ 340,000.00
Total benefits	฿ 638,000.00	฿ 668,000.00	฿ 699,800.00	฿ 733,508.00	฿ 769,238.00	฿ 3,508,546.00
Development costs						
Servers	฿ 200,000.00	฿ -	฿ -	฿ -	฿ -	฿ 200,000.00
Printer	฿ 90,000.00	฿ -	฿ -	฿ -	฿ -	฿ 90,000.00
Software licenses	฿ 40,000.00	฿ -	฿ -	฿ -	฿ -	฿ 40,000.00
Server software	฿ 1,060,000.00	฿ -	฿ -	฿ -	฿ -	฿ 1,060,000.00
Development labor	฿ 250,000.00	฿ -	฿ -	฿ -	฿ -	฿ 250,000.00
Total development costs	฿ 1,640,000.00	฿ -	฿ -	฿ -	฿ -	฿ 1,640,000.00
Operational costs						
Hardware	฿ 50,000.00	฿ 50,000.00	฿ 50,000.00	฿ 50,000.00	฿ 50,000.00	฿ 250,000.00
Software	฿ 20,000.00	฿ 20,000.00	฿ 20,000.00	฿ 20,000.00	฿ 20,000.00	฿ 100,000.00
Operational labor	฿ 111,788.00	฿ 116,260.00	฿ 120,910.00	฿ 125,746.00	฿ 130,776.00	฿ 605,480.00
Total operational costs	฿ 181,788.00	฿ 186,260.00	฿ 190,910.00	฿ 195,746.00	฿ 200,776.00	฿ 955,480.00
Total costs	฿ 1,821,788.00	฿ 186,260.00	฿ 190,910.00	฿ 195,746.00	฿ 200,776.00	฿ 2,595,480.00
Total	-฿ 1,183,788.00	฿ 481,740.00	฿ 508,890.00	฿ 537,762.00	฿ 568,462.00	฿ 913,066.00
Return of Investment (ROI) 35.18%		Net Present Value (NPV)			฿793,970.43	

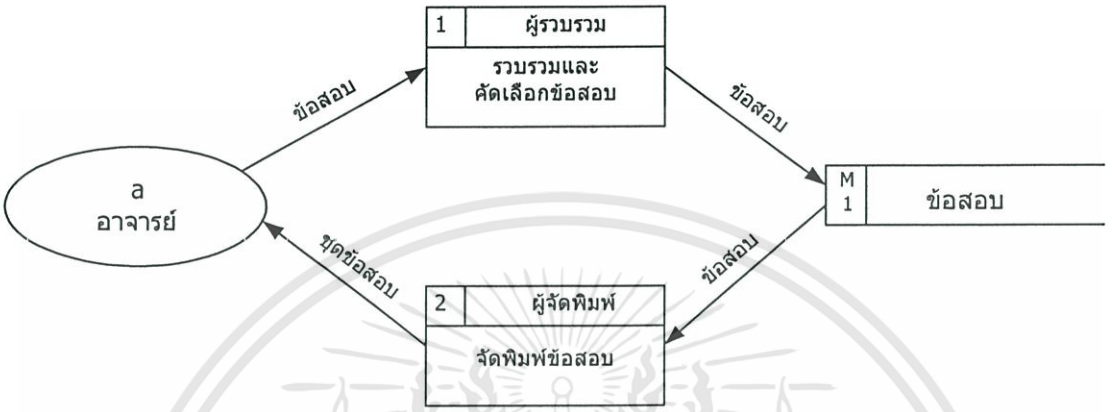
ตารางที่ 6.25 แสดงตารางงบประมาณ แบบที่ 2

	2000(฿)	2001(฿)	2002(฿)	2003(฿)	2004(฿)	Total
Benefits						
registration	฿ 500,000.00	฿ 530,000.00	฿ 561,800.00	฿ 595,508.00	฿ 631,238.00	฿ 2,818,546.00
Improved customer service	฿ 70,000.00	฿ 70,000.00	฿ 70,000.00	฿ 70,000.00	฿ 70,000.00	฿ 350,000.00
Reduced inventory costs	฿ 68,000.00	฿ 68,000.00	฿ 68,000.00	฿ 68,000.00	฿ 68,000.00	฿ 340,000.00
Total benefits	฿ 638,000.00	฿ 668,000.00	฿ 699,800.00	฿ 733,508.00	฿ 769,238.00	฿ 3,508,546.00
Development costs						
Servers	฿ 200,000.00	฿ -	฿ -	฿ -	฿ -	฿ 200,000.00
Printer	฿ 170,000.00	฿ -	฿ -	฿ -	฿ -	฿ 170,000.00
Software licenses	฿ 40,000.00	฿ -	฿ -	฿ -	฿ -	฿ 40,000.00
Server software	฿ 1,060,000.00	฿ -	฿ -	฿ -	฿ -	฿ 1,060,000.00
Development labor	฿ 400,000.00	฿ -	฿ -	฿ -	฿ -	฿ 400,000.00
Total development costs	฿ 1,870,000.00	฿ -	฿ -	฿ -	฿ -	฿ 1,870,000.00
Operational costs						
Hardware	฿ 50,000.00	฿ 50,000.00	฿ 50,000.00	฿ 50,000.00	฿ 50,000.00	฿ 250,000.00
Software	฿ 20,000.00	฿ 20,000.00	฿ 20,000.00	฿ 20,000.00	฿ 20,000.00	฿ 100,000.00
Operational labor	฿ 111,788.00	฿ 116,260.00	฿ 120,910.00	฿ 125,746.00	฿ 130,776.00	฿ 605,480.00
Total operational costs	฿ 181,788.00	฿ 186,260.00	฿ 190,910.00	฿ 195,746.00	฿ 200,776.00	฿ 955,480.00
Total costs	฿ 2,051,788.00	฿ 186,260.00	฿ 190,910.00	฿ 195,746.00	฿ 200,776.00	฿ 2,825,480.00
Total	-฿ 1,413,788.00	฿ 481,740.00	฿ 508,890.00	฿ 537,762.00	฿ 568,462.00	฿ 683,066.00
Return of Investment (ROI)	24.18%	Net Present Value (NPV)			฿593,970.43	

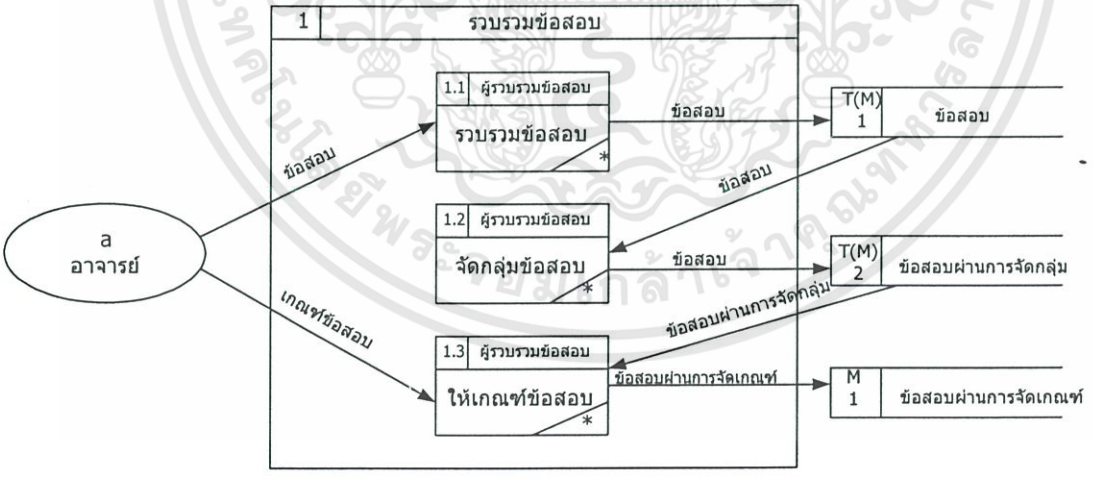
ตารางที่ 6.26 แสดงตารางงบประมาณ แบบที่ 3

	2000(฿)	2001(฿)	2002(฿)	2003(฿)	2004(฿)	Total
Benefits						
registration	฿ 500,000.00	฿ 530,000.00	฿ 561,800.00	฿ 595,508.00	฿ 631,238.00	฿ 2,818,546.00
Improved customer service	฿ 70,000.00	฿ 70,000.00	฿ 70,000.00	฿ 70,000.00	฿ 70,000.00	฿ 350,000.00
Reduced inventory costs	฿ 68,000.00	฿ 68,000.00	฿ 68,000.00	฿ 68,000.00	฿ 68,000.00	฿ 340,000.00
Total benefits	฿ 638,000.00	฿ 668,000.00	฿ 699,800.00	฿ 733,508.00	฿ 769,238.00	฿ 3,508,546.00
Development costs						
Servers	฿ 100,000.00	฿ -	฿ -	฿ -	฿ -	฿ 100,000.00
Printer	฿ 90,000.00	฿ -	฿ -	฿ -	฿ -	฿ 90,000.00
Software licenses	฿ 40,000.00	฿ -	฿ -	฿ -	฿ -	฿ 40,000.00
Server software	฿ 1,960,000.00	฿ -	฿ -	฿ -	฿ -	฿ 1,960,000.00
Development labor	฿ 100,000.00	฿ -	฿ -	฿ -	฿ -	฿ 100,000.00
Total development costs	฿ 2,290,000.00	฿ -	฿ -	฿ -	฿ -	฿ 2,290,000.00
Operational costs						
Hardware	฿ 50,000.00	฿ 50,000.00	฿ 50,000.00	฿ 50,000.00	฿ 50,000.00	฿ 250,000.00
Software	฿ 20,000.00	฿ 20,000.00	฿ 20,000.00	฿ 20,000.00	฿ 20,000.00	฿ 100,000.00
Operational labor	฿ 60,000.00	฿ 70,000.00	฿ 80,000.00	฿ 80,000.00	฿ 100,000.00	฿ 390,000.00
Total operational costs	฿ 130,000.00	฿ 140,000.00	฿ 150,000.00	฿ 150,000.00	฿ 170,000.00	฿ 740,000.00
Total costs	฿ 2,420,000.00	฿ 140,000.00	฿ 150,000.00	฿ 150,000.00	฿ 170,000.00	฿ 3,030,000.00
Total	-฿ 1,782,000.00	฿ 528,000.00	฿ 549,800.00	฿ 583,508.00	฿ 599,238.00	฿ 478,546.00
Return of Investment (ROI)	15.79%	Net Present Value (NPV)			฿416,126.96	

6.2 ระยะที่ 1

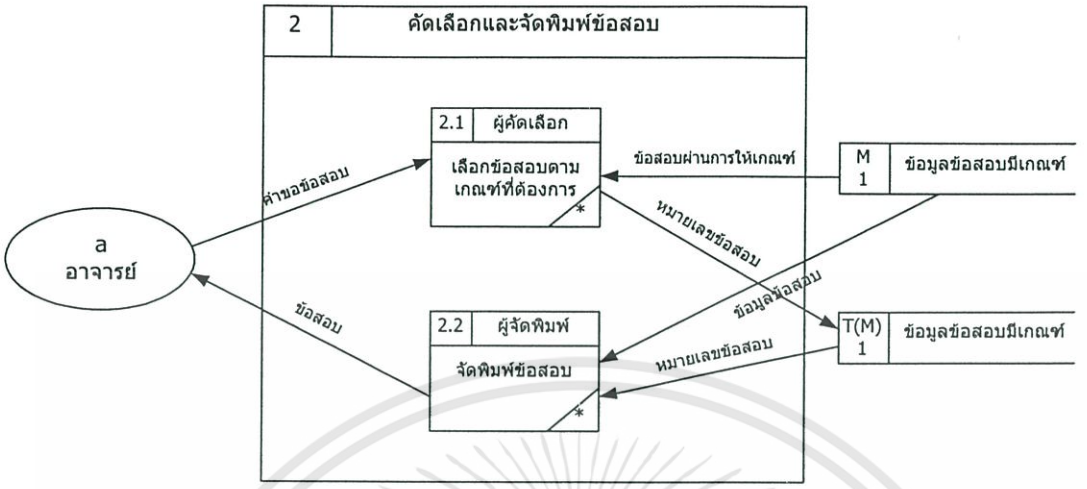


รูปที่ 6.15 แสดงแผนภูมิการไหลของข้อมูลกายภาพระดับที่ 1 ของระบบเดิม

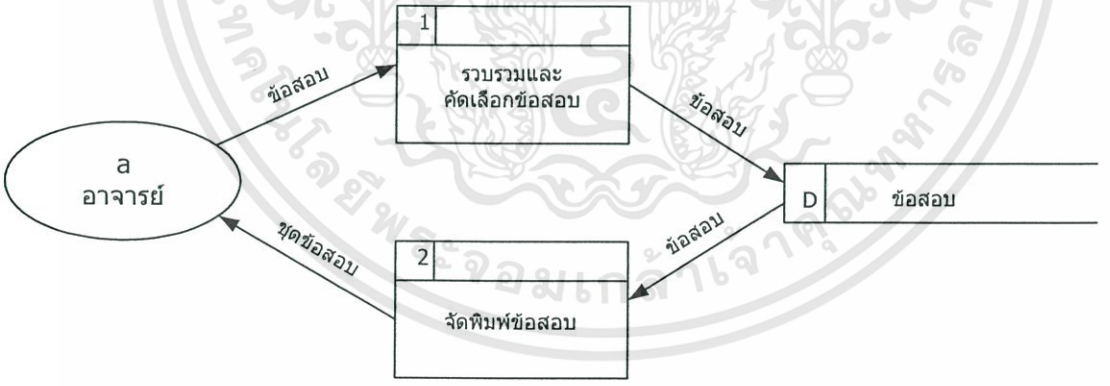


รูปที่ 6.16 แสดงแผนภูมิการไหลของข้อมูลกายภาพระดับที่ 2 ของระบบเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

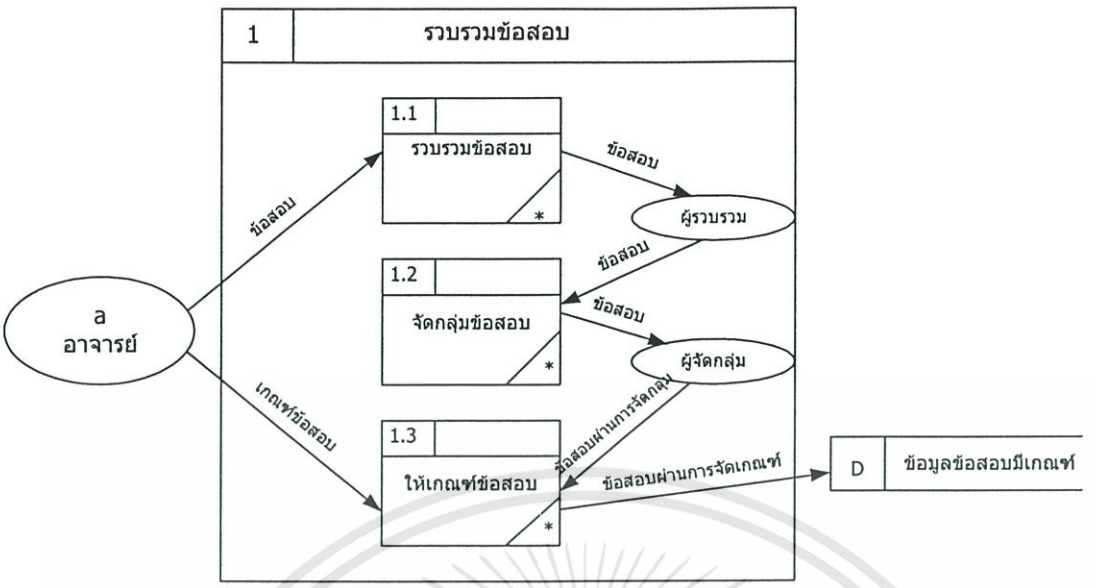


รูปที่ 6.17 แสดงแผนภูมิการไหลของข้อมูลกายภาพระดับที่ 2 ของระบบเดิม (ต่อ)

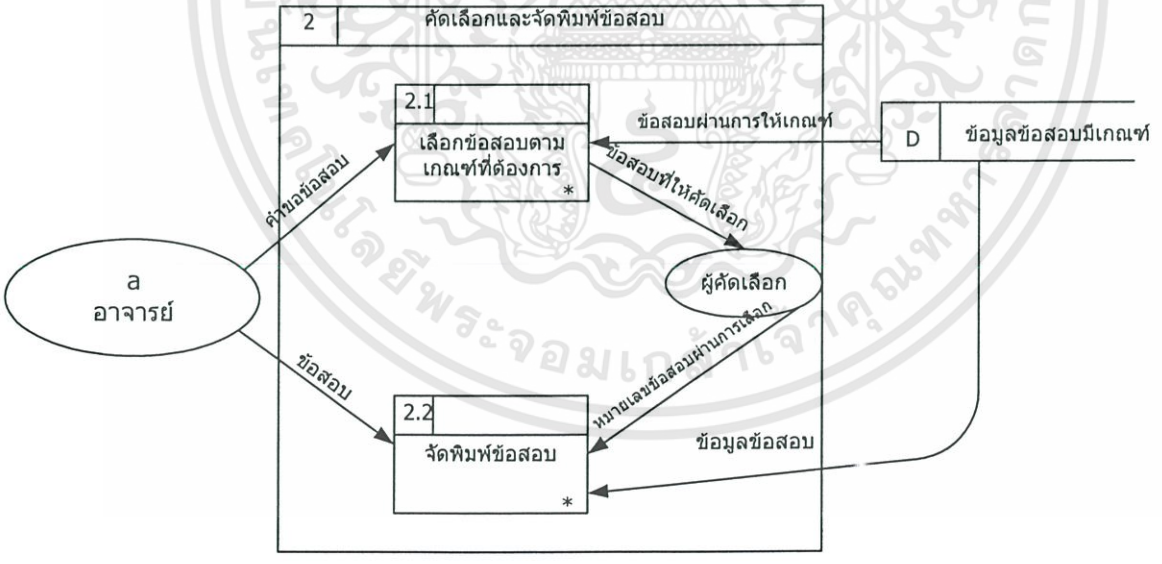


รูปที่ 6.18 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะระดับที่ 1 ของระบบเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

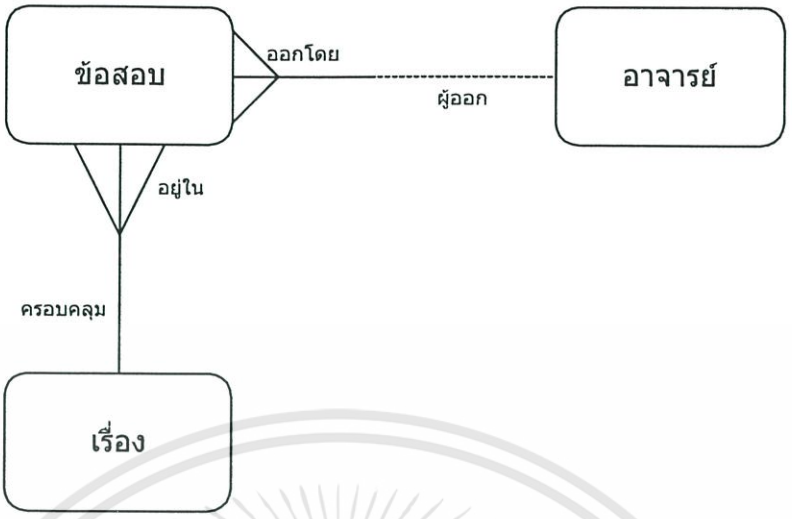


รูปที่ 6.19 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะระดับที่ 2 ของระบบเดิม



รูปที่ 6.20 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะระดับที่ 2 ของระบบเดิม (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.21 แสดงแผนภูมิของโครงสร้างข้อมูลทางตรรกะของระบบเดิม

ตารางที่ 6.27 แสดงความสัมพันธ์ของกระบวนการกับเอนติตีในระบบเดิม

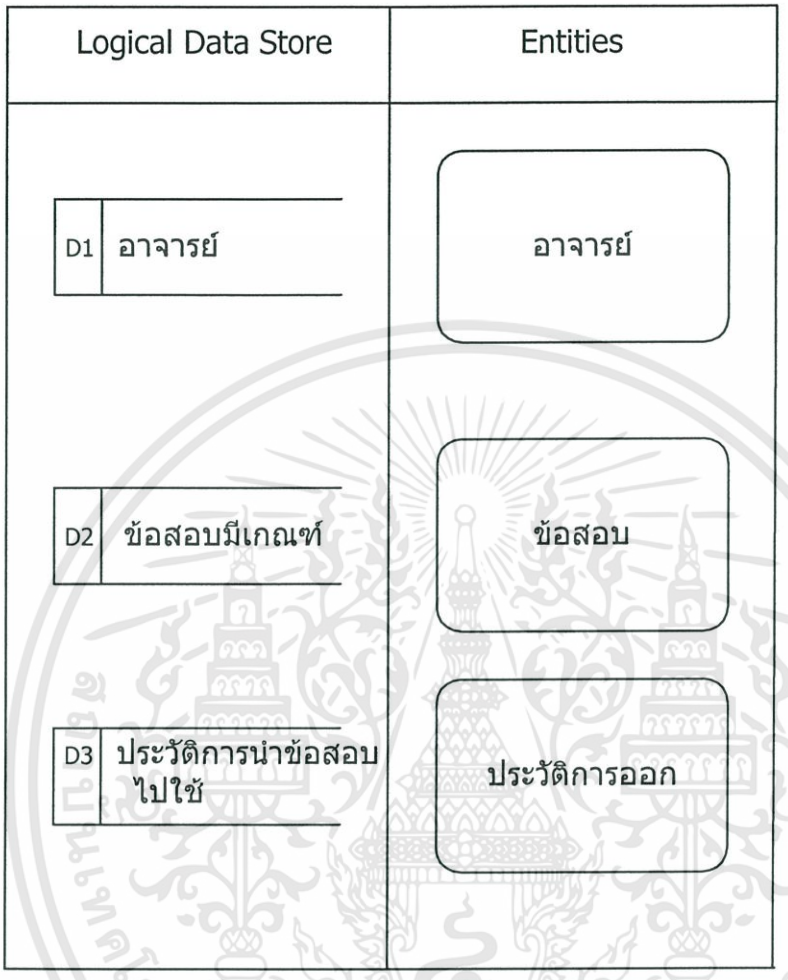
	ข้อสอบ	อาจารย์	เรื่อง
1.1	C	R	
1.2	R		C
1.3	A		R
2.1	R		R
2.2	R		

หมายเหตุ C = Create

R = Replace

A = Archive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.22 แสดงความสัมพันธ์ระหว่าง LDS กับเอนติตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ระยะที่ 2

ทางเลือกทางธุรกิจ (Business System Option -BSO)

เลือกทางเลือกทางธุรกิจ (Select Business System Options)

จากการวิเคราะห์ BSO ทั้ง 3 ทางเลือกซึ่งสามารถรองรับความต้องการที่จะเป็นต้องตัวระบบได้ทุกทางเลือก จึงเห็นว่า BSO ที่ 2 เป็นทางเลือกที่เหมาะสมที่สุดเพราะเนื่องจากว่าสามารถที่พัฒนาตัวระบบได้เองเพื่อให้ตรงตามความต้องการมากที่สุด เพราะจะทำให้สามารถเปลี่ยนแปลงหรือพัฒนาได้ตามความต้องการในภายหลัง

ทางเลือกทางเทคนิค (TECHNICAL SYSTEM OPTIONS)

TSO 1: ใช้ PC Computer เพียงเครื่องเดียวสำหรับระบบ และใช้โปรแกรมการจัดการฐานข้อมูลที่สนับสนุนการใช้ฐานข้อมูลเพียงผู้ใช้คนเดียวในหนึ่งเวลา โปรแกรมที่จะนำมาใช้นี้จะถูกพัฒนาขึ้นเองทั้งหมดทั้งส่วนของการติดต่อระหว่างผู้ใช้กับระบบ และส่วนของการจัดการฐานข้อมูล

Outline Technical Environment Description

- Hardware ใช้เครื่อง Pentium 266 MHz ขึ้นไป
- ระบบปฏิบัติการภายใต้ Windows

System Description

- การพัฒนาระบบผู้พัฒนาต้องดำเนินการเองทุกอย่างทั้งในส่วนของตัวฐานข้อมูล และตัวโปรแกรมที่ติดต่อระหว่างฐานข้อมูลกับผู้ใช้ (Interface) และตัวระบบสามารถรองรับการใช้งานได้เพียง 1 คนในหนึ่งเวลา การบันทึกเปลี่ยนแปลงข้อมูลจะกระทำโดยคน ๆ เดียว
- ตัวโปรแกรมที่ใช้ในการจัดการฐานข้อมูลและติดต่อระหว่างผู้ใช้จะเป็นโปรแกรมเดียวกัน

Impact Analysis

- แต่เดิมอาจารย์จะเป็นผู้ที่ออกข้อสอบและดูแลจัดการทั้งหมด ซึ่งเมื่อนำระบบนี้เข้ามาการจัดการในเรื่องการออกข้อสอบจะกระทำได้อย่างมีประสิทธิภาพและรวดเร็วทันต่อเหตุการณ์และสามารถจัดชุดข้อสอบได้ตรงตามความประสงค์ของการสอบแต่ละครั้ง
- ในด้านความปลอดภัยจะเห็นได้ว่า เป็นระบบที่ผู้ใช้เพียงคนเดียวจึงเป็นการดูแลความปลอดภัยโดยส่วนตัวเฉพาะผู้ใช้ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Outline Development Plan

การพัฒนาตัวระบบสามารถพัฒนาได้โดยใช้คู่มือของตัวระบบที่ถูกพัฒนาขึ้นมา ซึ่งขอบเขตในการพัฒนายังถูกจำกัดอยู่เพียงการทำงานบน PC เพียงเครื่องเดียว

Cost/Benefit Analysis

- ค่าใช้จ่ายที่ใช้ในการพัฒนา จัดการระบบ และรักษาระบบขึ้นอยู่กับการเลือกอุปกรณ์ที่จะใช้ครั้งแรกเพียงครั้งเดียว
- ระบบที่ใช้จะทำให้การทำงานของระบบเป็นไปโดยรวดเร็วมีประสิทธิภาพ ทำให้การออกข้อสอบทำได้ตามความต้องการซึ่งจะเป็นการประหยัดเวลามากขึ้น

TSO 2: จะมีลักษณะเหมือนกับ TSO อันที่ 1 แต่โปรแกรมที่จะนำมาใช้นี้จะถูกพัฒนาขึ้นเองเฉพาะในส่วนของการติดต่อระหว่างผู้ใช้กับระบบ แต่ส่วนของการจัดการฐานข้อมูลจะใช้โปรแกรมที่มีการพัฒนามาแล้ว

Outline Technical Environment Description

- Hardware ใช้เครื่อง Pentium 266 MHz ขึ้นไป
- ระบบปฏิบัติการภายใต้ Windows

System Description

- การพัฒนาระบบผู้พัฒนาต้องดำเนินการเองในส่วนของตัวโปรแกรมที่ติดต่อระหว่างฐานข้อมูลกับผู้ใช้ (Interface) และตัวระบบสามารถรองรับการใช้งานได้เพียง 1 คนในหนึ่งเวลา การบันทึกเปลี่ยนแปลงข้อมูลในฐานข้อมูลจะกระทำโดยคน ๆ เดียว
- การพัฒนาระบบโดยเลือก TSO 2 นี้โปรแกรมที่จะถูกแยกออกเป็น 2 โปรแกรมคือในส่วนแรกเป็นโปรแกรมจัดการฐานข้อมูลโดยจะต้องเลือกโปรแกรมที่มีประสิทธิภาพโดยควรจะมีลักษณะเป็นแบบ Client – Server ส่วนโปรแกรมที่ติดต่อกับผู้ใช้จะพัฒนาเองโดยให้มีลักษณะเป็นแบบ Client ที่สามารถติดต่อกับโปรแกรมจัดการฐานข้อมูลได้

Impact Analysis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ซึ่งเมื่อนำระบบนี้เข้ามาการจัดการในเรื่องการออกข้อสอบจะกระทำได้อย่างมีประสิทธิภาพและรวดเร็วทันต่อเหตุการณ์และสามารถจัดชุดข้อสอบได้ตรงตามความประสงค์ของการสอบแต่ละครั้ง

Outline Development Plan

สำหรับ TSO 2 จะมีเอกสารอ้างอิงต่าง ๆ และการพัฒนาสามารถที่พัฒนาได้ในส่วนของโปรแกรมที่ใช้ติดต่อกับผู้ใช้

Cost/Benefit Analysis

- ค่าใช้จ่ายที่ใช้ในการพัฒนาจัดการระบบ บำรุงรักษาระบบ ขึ้นอยู่กับ Hardware ที่ใช้และโปรแกรมแบ่งออกเป็น 2 ส่วนคือส่วนโปรแกรมจัดการฐานข้อมูล และโปรแกรมที่ติดต่อกับผู้ใช้ โดยเมื่อเทียบค่าใช้จ่ายกับ TSO 1 แล้วจะมีค่าใช้จ่ายที่สูงกว่าอยู่ค่อนข้างมาก

TSO 3: ใช้ PC Computer มากกว่า 1 เครื่องสำหรับระบบ โดยนำคอมพิวเตอร์ 1 เครื่องมาทำหน้าที่เป็น Server เพื่อสนับสนุนการใช้งานสำหรับผู้ใช้หลายคนในหนึ่งเวลา ส่วนคอมพิวเตอร์ที่เหลือจะใช้เป็นอุปกรณ์ในการทำงานของผู้ใช้ เครื่องคอมพิวเตอร์ทั้งหมดต้องต่อเชื่อมโยมกันเป็นระบบ Local Are Network โปรแกรมที่ใช้ในการจัดการฐานข้อมูลจะต้องรองรับการทำงานแบบผู้ใช้หลายคนในหนึ่งเวลา โดยโปรแกรมที่ใช้นี้จะถูกพัฒนาขึ้นเองทั้งหมดทั้งส่วนของการติดต่อระหว่างผู้ใช้กับระบบ และส่วนของการจัดการฐานข้อมูล

Outline Technical Environment Description

- Hardware ใช้เครื่อง Pentium 266 MHz ขึ้นไป จำนวน 2 เครื่อง
- การเชื่อมต่อเครือข่ายแบบ (Local Area Network)
- ระบบปฏิบัติการภายใต้ Windows

System Description

- การพัฒนาระบบผู้พัฒนาต้องดำเนินการเองในส่วนของตัวโแรแกรมที่ติดต่อระหว่างฐานข้อมูลกับผู้ใช้ (Interface) และพัฒนาโปรแกรมในส่วนของการจัดการฐานข้อมูลด้วย
- การพัฒนาระบบโดยเลือก TSO 3 นี้ โปรแกรมที่จะถูกแยกออกเป็น 2 โปรแกรมคือในส่วนแรกเป็นโปรแกรมจัดการฐานข้อมูลโดยจะต้องเลือกโปรแกรมที่มีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยควรจะมีลักษณะเป็นแบบ Client – Server ส่วนโปรแกรมที่ติดต่อกับผู้ใช้จะพัฒนาเองโดยให้มีลักษณะเป็นแบบ Client ที่สามารถติดต่อกับโปรแกรมจัดการฐานข้อมูลได้

Impact Analysis

- ซึ่งเมื่อนำระบบนี้เข้ามาการจัดการในเรื่องการออกข้อสอบจะกระทำได้อย่างมีประสิทธิภาพและรวดเร็วทันต่อเหตุการณ์และสามารถจัดชุดข้อสอบได้ตรงตามความประสงค์ของการสอบแต่ละครั้ง
- การใช้ฐานข้อมูลข้อสอบสามารถที่ใช้ได้มากกว่า 1 คน ในเวลาเดียวกันเพราะเนื่องจากตัวฐานข้อมูลเป็นแบบ Client – Server ดังนั้นจึงทำให้สามารถรองรับการใช้ข้อมูลได้หลายคนในเวลาเดียวกัน

Outline Development Plan

สำหรับ TSO 3 จะมีเอกสารอ้างอิงต่าง ๆ และการพัฒนาสามารถที่พัฒนาได้ในส่วนของโปรแกรมที่ใช้ติดต่อกับผู้ใช้ และในส่วนของ Hardware สามารถเพิ่มจำนวนเครื่องที่จะใช้เป็น Client ได้เรื่อย ๆ ขึ้นอยู่กับความต้องการใช้

Cost/Benefit Analysis

- ค่าใช้จ่ายที่ใช้ในการพัฒนาจัดการระบบ บำรุงรักษาระบบ ขึ้นอยู่กับ Hardware ที่ใช้และโปรแกรมแบ่งออกเป็น 2 ส่วนคือส่วนโปรแกรมจัดการฐานข้อมูล และโปรแกรมที่ติดต่อกับผู้ใช้ โดยเมื่อเทียบค่าใช้จ่ายกับ TSO 2 และจะมีค่าใช้จ่ายที่สูงกว่าอยู่ค่อนข้างมาก
- สำหรับ TSO 3 เป็นระบบที่ทำให้การทำงานในระบบ สามารถทำงานได้อย่างรวดเร็วมีประสิทธิภาพ และสามารถที่จะใช้งานตัวระบบได้หลายคนพร้อมกันอย่างมีประสิทธิภาพ อันจะส่งผลให้การออกข้อสอบมีได้ตามความต้องการที่รวดเร็วและไม่ซ้ำซ้อน

TSO 4: จะมีลักษณะเหมือนกับ TSO อันที่ 3 แต่โปรแกรมที่ใช้นี้จะถูกพัฒนาขึ้นเองเฉพาะในส่วนของการติดต่อระหว่างผู้ใช้กับระบบ แต่ส่วนของการจัดการฐานข้อมูลจะใช้โปรแกรมที่มีการพัฒนาแล้ว

Outline Technical Environment Description

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Hardware ใช้เครื่อง Pentium 266 MHz ขึ้นไป จำนวน 2 เครื่อง
- การเชื่อมต่อเครือข่ายแบบ (Local Area Network)
- ระบบปฏิบัติการภายใต้ Windows

System Description

- การพัฒนาระบบผู้พัฒนาต้องดำเนินการเองในส่วนของตัวโปรแกรมที่ติดต่อ ระหว่างฐานข้อมูลกับผู้ใช้ (Interface) ในส่วนโปรแกรมการจัดการฐานข้อมูลจะเลือกใช้โปรแกรมจัดการฐานข้อมูลที่มีขายอยู่ทั่วไปในท้องตลาดโดยเลือกโปรแกรมที่มีประสิทธิภาพ
- การพัฒนาระบบโดยเลือก TSO 4 นี้ โปรแกรมที่จะถูกแยกออกเป็น 2 โปรแกรมคือในส่วนแรกเป็นโปรแกรมจัดการฐานข้อมูลโดยจะต้องเลือกโปรแกรมที่มีประสิทธิภาพโดยควรจะมีลักษณะเป็นแบบ Client – Server ส่วนโปรแกรมที่ติดต่อกับผู้ใช้จะพัฒนาเองโดยให้มีลักษณะเป็นแบบ Client ที่สามารถติดต่อกับโปรแกรมจัดการฐานข้อมูลได้

Impact Analysis

- ซึ่งเมื่อนำระบบนี้เข้ามาการจัดการในเรื่องการออกข้อสอบจะกระทำได้อย่างมีประสิทธิภาพและรวดเร็วทันต่อเหตุการณ์และสามารถจัดชุดข้อสอบได้ตรงตามความประสงค์ของการสอบแต่ละครั้ง
- การใช้ฐานข้อมูลข้อสอบสามารถที่ใช้ได้มากว่า 1 คน ในเวลาเดียวกันเพราะเนื่องจากตัวฐานข้อมูลเป็นแบบ Client – Server ดังนั้นจึงทำให้สามารถรองรับการใช้ข้อมูลได้หลายคนในเวลาเดียวกัน

Outline Development Plan

สำหรับ TSO 4 จะมีเอกสารอ้างอิงต่าง ๆ และการพัฒนาสามารถที่พัฒนาได้ในส่วนของโปรแกรมที่ใช้ติดต่อกับผู้ใช้ และในส่วนของ Hardware สามารถเพิ่มจำนวนเครื่องที่จะใช้เป็น Client ได้เรื่อย ๆ ขึ้นอยู่กับความต้องการใช้

Cost/Benefit Analysis

- ค่าใช้จ่ายที่ใช้ในการพัฒนาจัดการระบบ บำรุงรักษาระบบ ขึ้นอยู่กับ Hardware ที่ใช้และโปรแกรมแบ่งออกเป็น 2 ส่วนคือส่วนโปรแกรมจัดการฐานข้อมูล และโปรแกรมที่ติดต่อกับผู้ใช้ โดยเมื่อเทียบค่าใช้จ่ายกับ TSO 3 แล้วจะมีค่าใช้จ่ายที่สูงกว่าอยู่ในส่วนของการเลือกโปรแกรมจัดการฐานข้อมูลซึ่งราคาจะสูงกว่าการพัฒนาเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

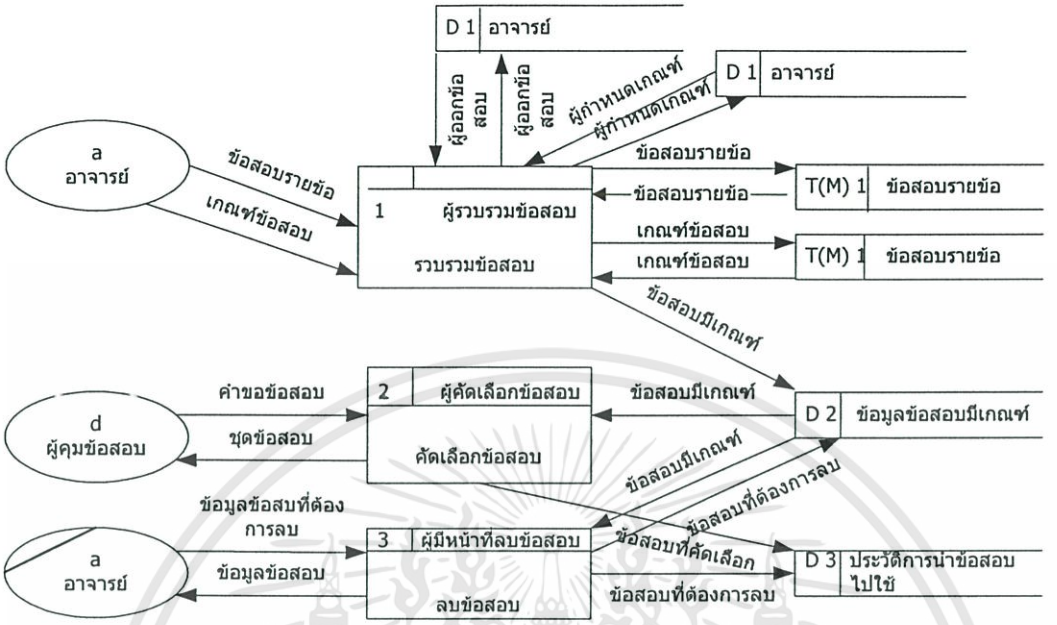
- สำหรับ TSO 4 เป็นระบบที่ทำให้การทำงานในระบบ สามารถทำงานได้อย่างรวดเร็ว มีประสิทธิภาพ และสามารถที่จะใช้งานตัวระบบได้หลายคนพร้อมกันอย่างมีประสิทธิภาพ อันจะส่งผลให้การออกข้อสอบมีได้ตามความต้องการที่รวดเร็วและไม่ซ้ำซ้อน

<u>เลือกทางเลือกทางเทคนิค (Select Technical System Options)</u>

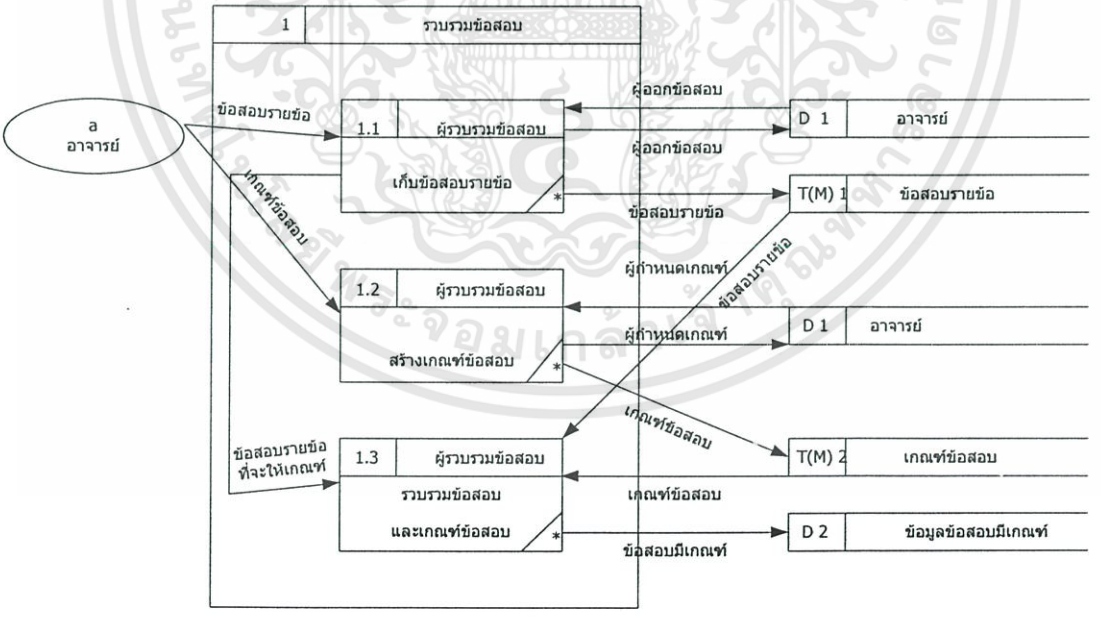
จากการวิเคราะห์ System Description, Impact Analysis, Outline Development Plan, Cost/Benefit Analysis และพิจารณาตาม BSO 2 ที่เลือกไว้แล้ว เห็นว่า TSO 1 เป็นวิธีที่ดีที่สุดเพราะสามารถรองรับ BSO 2 ได้และระบบต้องการมากที่สุด ซึ่งถือว่าเป็นทางเลือกที่ทำให้ระบบมีการทำงานเป็นไปอย่างมีประสิทธิภาพและตรงตามความต้องการของระบบ และเป้าหมายของระบบที่จะสร้างมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ระยะที่ 3

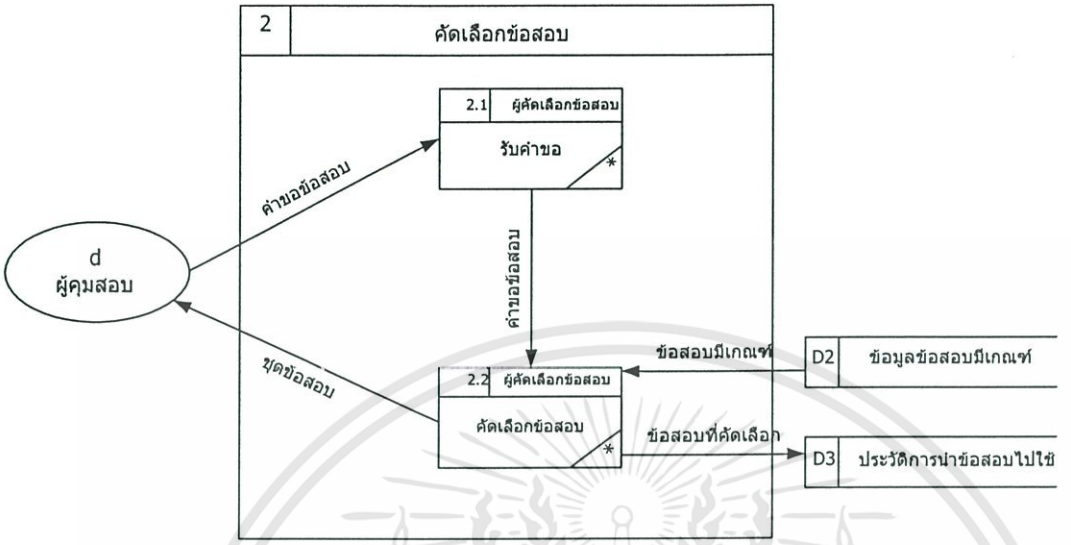


รูปที่ 6.23 แสดงแผนภูมิการไหลของข้อมูลตามความต้องการของผู้ใช้ระดับที่ 1

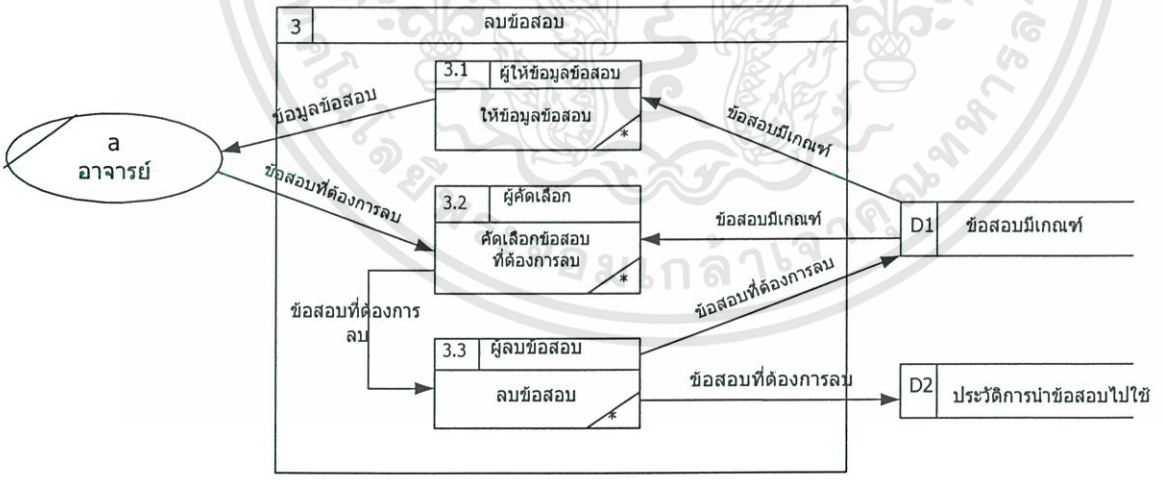


รูปที่ 6.24 แสดงแผนภูมิการไหลของข้อมูลตามความต้องการของผู้ใช้ระดับที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

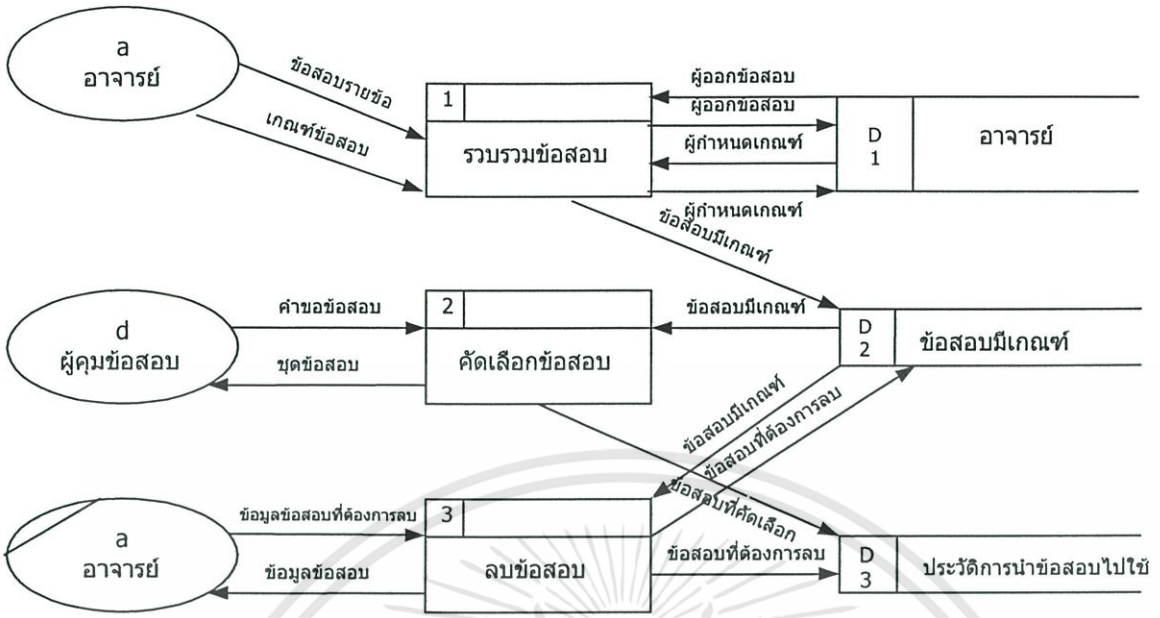


รูปที่ 6.25 แสดงแผนภูมิการไหลของข้อมูลตามความต้องการของผู้ใช้ระดับที่ 2 (ต่อ)

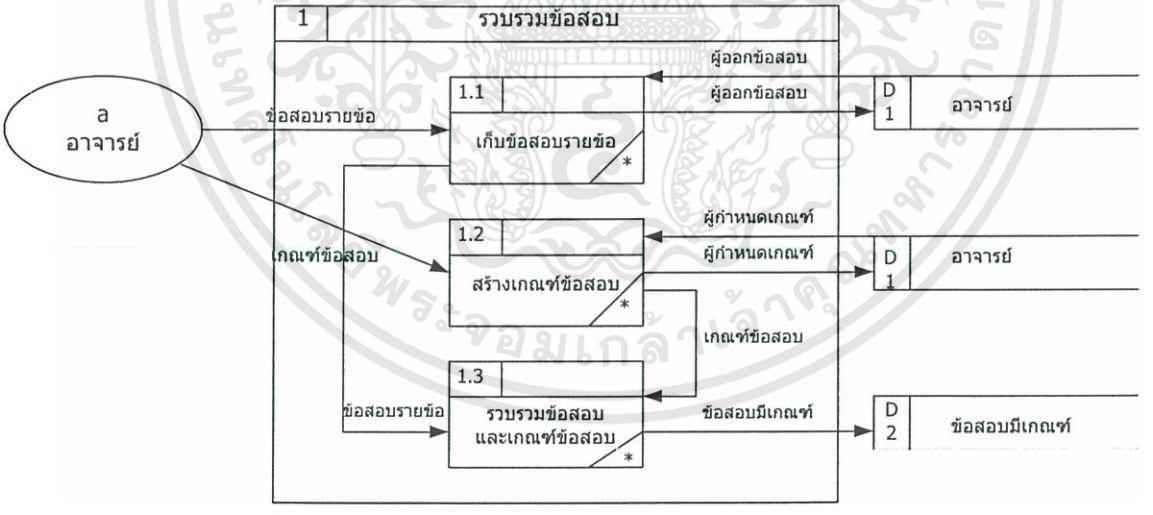


รูปที่ 6.26 แสดงแผนภูมิการไหลของข้อมูลตามความต้องการของผู้ใช้ระดับที่ 2 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

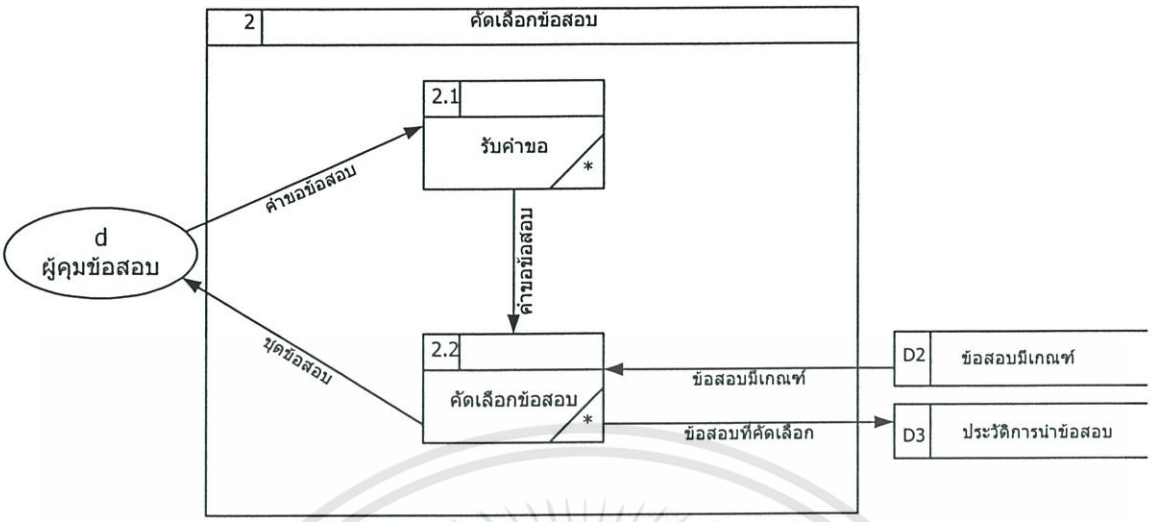


รูปที่ 6.27 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะตามความต้องการของผู้ใช้ระดับที่ 1

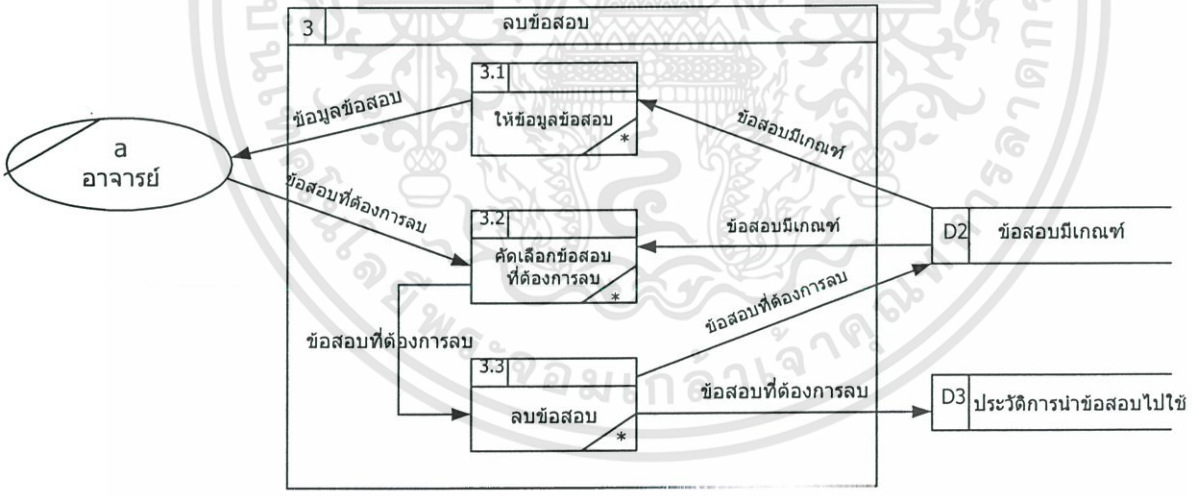


รูปที่ 6.28 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะตามความต้องการของผู้ใช้ระดับที่ 2 ของกระบวนการที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.29 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะตามความต้องการของผู้ใช้ระดับที่ 2 ของกระบวนการที่ 2



รูปที่ 6.30 แสดงแผนภูมิการไหลของข้อมูลทางตรรกะตามความต้องการของผู้ใช้ระดับที่ 2 ของกระบวนการที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.28 แสดงรายละเอียดของกระบวนการที่ 1.1

Process ID: 1.1
Process Name: เก็บข้อสอบรายข้อ
Description: อาจารย์ให้ข้อมูลข้อสอบรายข้อซึ่งสามารถออกข้อสอบได้โดยอาจารย์หลายคน โดยข้อสอบดังกล่าวประกอบด้วยข้อสอบหลายวิชา ในแต่ละวิชามีได้หลายบท แต่ละบทมีข้อสอบหลายข้อ และแต่ละข้อจะมีคำตอบตัวเลือกได้ 4-5 ตัวเลือก

ตารางที่ 6.29 แสดงรายละเอียดของกระบวนการที่ 1.2

Process ID: 1.2
Process Name: สร้างเกณฑ์ข้อสอบ
Description: ผู้กำหนดเกณฑ์ข้อสอบนำข้อสอบแต่ละข้อมาสร้างเกณฑ์ข้อสอบ โดยจะบันทึกเก็บไว้ชั่วคราวเพื่อการรวบรวมในการเก็บลงฐานข้อมูล

ตารางที่ 6.30 แสดงรายละเอียดของกระบวนการที่ 1.3

Process ID: 1.3
Process Name: รวบรวมข้อสอบและเกณฑ์ข้อสอบ
Description: ผู้รวบรวมนำข้อสอบแต่ละข้อมารวมกับเกณฑ์ที่ถูกสร้างไว้ หลังจากนั้นทำการบันทึกลงฐานข้อมูลเพื่อสามารถนำมาใช้ประโยชน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.31 แสดงรายละเอียดของกระบวนการที่ 2.1

Process ID: 2.1
Process Name: รับคำขอ
Description: ผู้คุมข้อสอบส่งคำขอข้อสอบให้กับผู้คัดเลือกข้อสอบเพื่อคัดเลือกข้อสอบ

ตารางที่ 6.32 แสดงรายละเอียดของกระบวนการที่ 2.2

Process ID: 2.2
Process Name: คัดเลือกข้อสอบ
Description: ผู้คัดเลือกข้อสอบนำคำขอข้อสอบ ไปคัดเลือกข้อสอบที่ต้องการจากแฟ้มข้อมูลข้อสอบ แล้วทำการพิมพ์ให้กับผู้รับข้อสอบ โดยข้อสอบที่ถูกเลือกจะถูกบันทึกลงในประวัติการนำข้อสอบไปใช้

ตารางที่ 6.33 แสดงรายละเอียดของกระบวนการที่ 3.1

Process ID: 3.1
Process Name: ให้ข้อมูลข้อสอบ
Description: ผู้ให้ข้อมูลข้อสอบนำข้อสอบจากฐานข้อมูลส่งให้กับอาจารย์เพื่อทำการตรวจสอบข้อสอบ

ตารางที่ 6.34 แสดงรายละเอียดของกระบวนการที่ 3.2

Process ID: 3.2
Process Name: คัดเลือกข้อสอบที่ต้องการลบ
Description: ผู้คัดเลือกข้อสอบทำการคัดเลือกข้อสอบตามข้อมูลที่ได้จากผู้ลบข้อสอบโดยเมื่อคัดเลือกได้ตามที่ต้องการจะส่งไปให้ผู้ลบข้อสอบทำการลบต่อไป

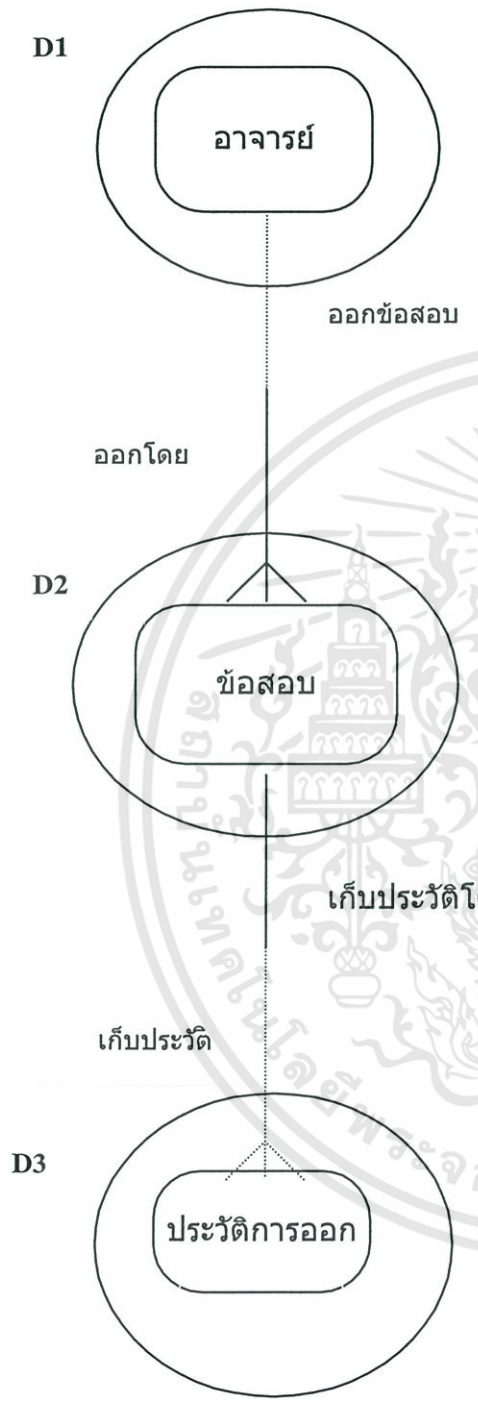
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.35 แสดงรายละเอียดของกระกบวนการที่ 3.3

Process ID: 3.3
Process Name: ลบข้อสอบ
Description: อาจารย์ทำการลบข้อสอบจากที่ได้ถูกคัดเลือกไว้แล้วโดยเมื่อลบแล้วจะไปทำการปรับปรุงแฟ้มข้อมูลข้อสอบและประวัติการนำข้อสอบไปใช้ให้เป็นปัจจุบัน

รูปที่ 6.31 แสดงแผนภูมิ LDS ทางตรรกะตามความต้องการของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.32 แสดงการกำหนดกลุ่มของแผนภูมิ LDS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.36 แสดงรายละเอียดของเอนติตีอาจารย์

Entity Name: อาจารย์				
Description: ทำการออกข้อสอบให้กับระบบ				
Attribute			**Primary Key**	**Foreign Key**
1. รหัสอาจารย์ 2. ชื่ออาจารย์ 3. นามสกุลอาจารย์ 4. คำอธิบาย			Yes	
Must/may be	**Either/or**	**Link Phrase**	**One and only one/ one or more**	**Object Entity Name**
May be		**ออก**	**One or more**	**ข้อสอบ**
Entity Volume: Max.__100___. Min ___1_____. Average. ____50____.				
User Role			**Access**	
ผู้รวบรวมข้อสอบ			**Create, Read, Modify, Delete**	
Growth Rate:				
Archiving: เก็บอยู่ในระบบจนกว่าจะลบออกหรือเลิกใช้ระบบ				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 6.37 แสดงรายละเอียดของเอนติตีประวัติการออกข้อสอบ

Entity Name: ประวัติการออก				
Description: เก็บประวัติข้อสอบแต่ละที่ถูกนำไปใช้ออกเป็นข้อสอบ				
Attribute			**Primary Key**	**Foreign Key**
1. รหัสข้อสอบที่ออก 2. จำนวนครั้งที่ออก 3. วันที่ออก วันสุดท้าย 4. รหัสตรวจสอบ			Yes	
Must/may be	**Either/or**	**Link Phrase**	**One and only one/ one or more**	**Object Entity Name**
May be		**เก็บประวัติ**	**One or more**	**ข้อสอบ**
Entity Volume: Max.__100,000___. Min ___100_____. Average. _____50,050____.				
User Role			**Access**	
ผู้คัดเลือกข้อสอบ **ผู้ออกข้อสอบ**			**Read** **Add, update**	
Growth Rate:				
Archiving: เก็บอยู่ในระบบจนกว่าจะลบออกหรือเลิกใช้ระบบ				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.38 แสดงรายละเอียดของเอนติตีข้อสอบ

Entity Name: ข้อสอบ		
Description: ข้อสอบที่เก็บอยู่ในระบบประกอบผู้เนื้อข้อสอบและเกณฑ์ของแต่ละข้อ		
Attribute	**Primary Key**	**Foreign Key**
1. รหัสข้อสอบ	Yes	
2. วิชา		
3. บทที่		
4. โจทย์		
5. ประเภทข้อสอบ		
6. ตัวเลือก		
7. เนื้อหา		
8. เฉลย		
9. รหัสผู้ออกข้อสอบ		YES
10. เกณฑ์ความยาก		
11. เกณฑ์ความตรง		
12. เกณฑ์ความน่าเชื่อถือ		
13. เกณฑ์อำนาจจำแนก		
14. วันที่ออกข้อสอบ		
15. วันที่ให้เกณฑ์ข้อสอบ		
16. วันที่สอบ		
17. เวลาสอบ		
18. จำนวนครั้งที่ใช้ข้อสอบ		

Must/may be	**Either/or**	**Link Phrase**	**One and only one/ one or more**	**Object Entity Name**
Must be		**ออกโดย**	**One and only one**	**ผู้ออกข้อสอบ**
Must be		**ให้เกณฑ์โดย**	**One and only one**	**ผู้ให้เกณฑ์**
Must be		**เก็บประวัติโดย**	**One or more**	**ประวัติการออก**

Entity Volume: Max. 100,000 . Min 10,000 . Average. 55,000 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.39 แสดงรายละเอียดของแอตทริบิวต์รหัสข้อสอบ

Name: รหัสข้อสอบ		
Description: ข้อสอบทุกข้อจะมีรหัสที่ Unique		
Cross Reference	**Type**	
ข้อสอบ ประวัติการออก ข้อมูลข้อสอบมีเกณฑ์	Entity Entity Data Store	
Domain: Numeric	Length: 8	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: สร้างโดยระบบ		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.40 แสดงรายละเอียดของแอตทริบิวต์รหัสวิชา

Name: รหัสวิชา		
Description: วิชาในข้อสอบ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Varchar2	Length: 8	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.41 แสดงรายละเอียดของแอตทริบิวต์บทที่

Name: บทที่		
Description: บทที่ของแต่ละวิชา		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Numeric	Length: 3	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.42 แสดงรายละเอียดของแอตทริบิวต์เนื้อหา

Name: เนื้อหา		
Description: รายละเอียดของตัวเลือกแต่ละข้อคำถาม		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: varchar2	Length: 2000	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.43 แสดงรายละเอียดของแอตทริบิวต์โจทย์

Name: โจทย์		
Description: โจทย์ของแต่ละข้อ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Varchar2	Length: 2000	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.44 แสดงรายละเอียดของแอตทริบิวต์ตัวเลือก

Name: ตัวเลือก		
Description: ลำดับตัวเลือกของโจทย์แต่ละข้อ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: number	Length: 1	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.45 แสดงรายละเอียดของแอตทริบิวต์ความยาก

Name: ความยาก		
Description: เกณฑ์ความยากของโจทย์แต่ละข้อ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Number	Length: 1	**Units:** Double
Mandatory: No	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation: 1 : easy , 2 : medium , 3 : hard		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.46 แสดงรายละเอียดของแอตทริบิวต์ความตรง

Name: ความตรง		
Description: เกณฑ์ความตรงของโจทย์แต่ละข้อ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Number	Length: 1	**Units:** Double
Mandatory: No	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation: 1 : มีความตรงน้อย , 2 : มีความตรงปานกลาง , 3 : มีความตรงมาก		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.47 แสดงรายละเอียดของแอตทริบิวต์ความน่าเชื่อถือ

Name: ความน่าเชื่อถือ		
Description: เกณฑ์ความน่าเชื่อถือของโจทย์แต่ละข้อ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Number	Length: 1	**Units:** Double
Mandatory: No	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation: 1 : น่าเชื่อถือน้อย, 2 : ปานกลาง, 3 : มาก		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.48 แสดงรายละเอียดของแอตทริบิวต์เวลาสอบ

Name: เวลาสอบ	
Description: เวลาสอบของข้อสอบแต่ละชุด	
Cross Reference	Type
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store
Domain: date	Length: **Units:** Double
Mandatory: No	**Optional:**
Derivation: ผู้ใช้เป็นผู้กำหนด	
Validation:	
User Role	**Access**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.49 แสดงรายละเอียดของแอตทริบิวต์เฉลย

Name: เฉลย		
Description: เฉลยข้อสอบของโจทย์แต่ละข้อ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Varchar2	Length: 2000	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.50 แสดงรายละเอียดของแอตทริบิวต์วันที่ออกข้อสอบ

Name: วันที่ออกข้อสอบ	
Description: วันที่ออกข้อสอบของข้อสอบแต่ละข้อ	
Cross Reference	Type
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store
Domain: Date	Length: **Units:** Byte
Mandatory: No	**Optional:**
Derivation: สร้างโดยระบบ	
Validation:	
User Role	**Access**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.51 แสดงรายละเอียดของแอตทริบิวต์วันที่ให้เกณฑ์ข้อสอบ

Name: วันที่ให้เกณฑ์ข้อสอบ		
Description: วันที่ให้เกณฑ์ข้อสอบของข้อสอบแต่ละข้อ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Date	Length:	**Units:** Byte
Mandatory: No	**Optional:**	
Derivation:		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.52 แสดงรายละเอียดของแอตทริบิวต์วันที่ใช้ข้อสอบครั้งสุดท้าย

Name: วันที่ใช้ข้อสอบครั้งสุดท้าย		
Description: วันที่ใช้ข้อสอบครั้งสุดท้ายของข้อสอบแต่ละข้อ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: Date	Length:	**Units:** Byte
Mandatory: No	**Optional:**	
Derivation:		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.53 แสดงรายละเอียดของแอตทริบิวต์ภาคการศึกษา

Name: ภาคการศึกษา		
Description: ภาคการศึกษาที่ออกข้อสอบ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: number	Length: 1	**Units:** Byte
Mandatory: No	**Optional:**	
Derivation:		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.54 แสดงรายละเอียดของแอตทริบิวต์ปีการศึกษา

Name: ปีการศึกษา		
Description: ปีการศึกษาที่ออกข้อสอบ		
Cross Reference	Type	
ข้อสอบ ข้อมูลข้อสอบมีเกณฑ์	Entity Data Store	
Domain: number	Length: 4	**Units:** Byte
Mandatory: No	**Optional:**	
Derivation:		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.55 แสดงรายละเอียดของแอตทริบิวต์จำนวนครั้งที่ออกข้อสอบ

Name: จำนวนครั้งที่ออกข้อสอบ		
Description: จำนวนครั้งที่ออกข้อสอบแต่ละข้อ		
Cross Reference	Type	
ประวัติการออก ประวัติการนำข้อสอบไปใช้	Entity Data Store	
Domain: number	Length: 3	**Units:** Integer
Mandatory: Yes	**Optional:**	
Derivation: สร้างโดยระบบ		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.56 แสดงรายละเอียดของแอตทริบิวต์รหัสอาจารย์

Name: รหัสอาจารย์		
Description: รหัสอาจารย์แต่ละคน		
Cross Reference	Type	
อาจารย์ ข้อสอบ อาจารย์	Entity Entity Data Store	
Domain: Numeric	Length: 3	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: สร้างโดยระบบ		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.57 แสดงรายละเอียดของแอตทริบิวต์ชื่ออาจารย์

Name: ชื่ออาจารย์		
Description: ชื่ออาจารย์แต่ละคน		
Cross Reference	Type	
อาจารย์ อาจารย์	Entity Data Store	
Domain: Varchar2	Length: 100	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.58 แสดงรายละเอียดของแอตทริบิวต์ที่อยู่อาจารย์

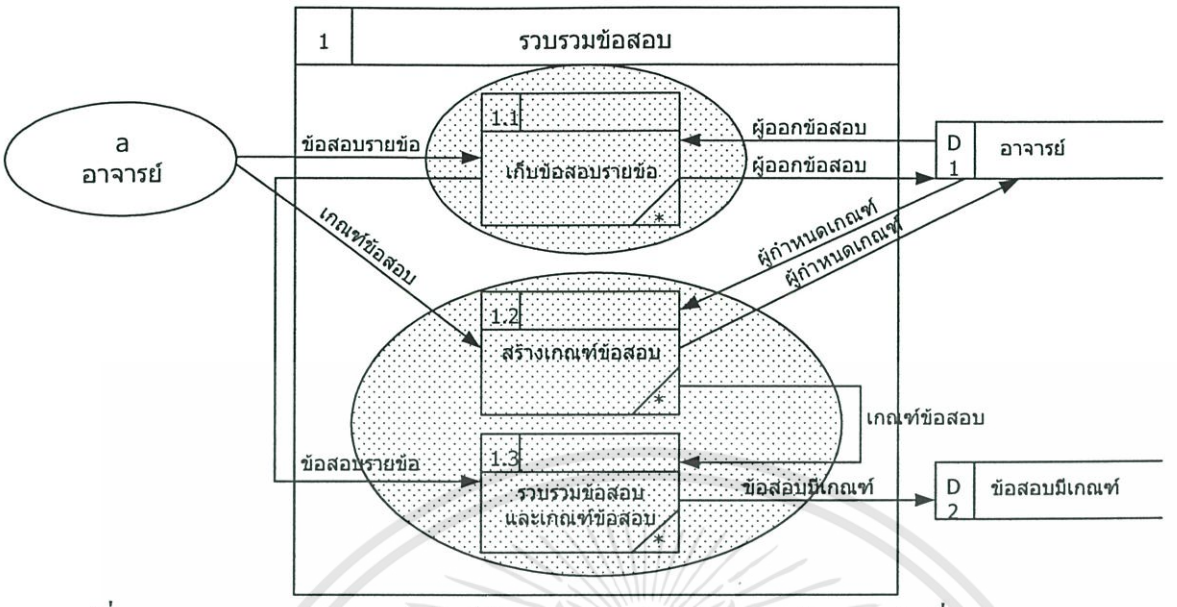
Name: ที่อยู่อาจารย์		
Description: ที่อยู่อาจารย์		
Cross Reference	Type	
อาจารย์ อาจารย์	Entity Data Store	
Domain: Varchar2	Length: 100	**Units:** Byte
Mandatory: Yes	**Optional:**	
Derivation: ผู้ใช้เป็นผู้กำหนด		
Validation:		
User Role	**Access**	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

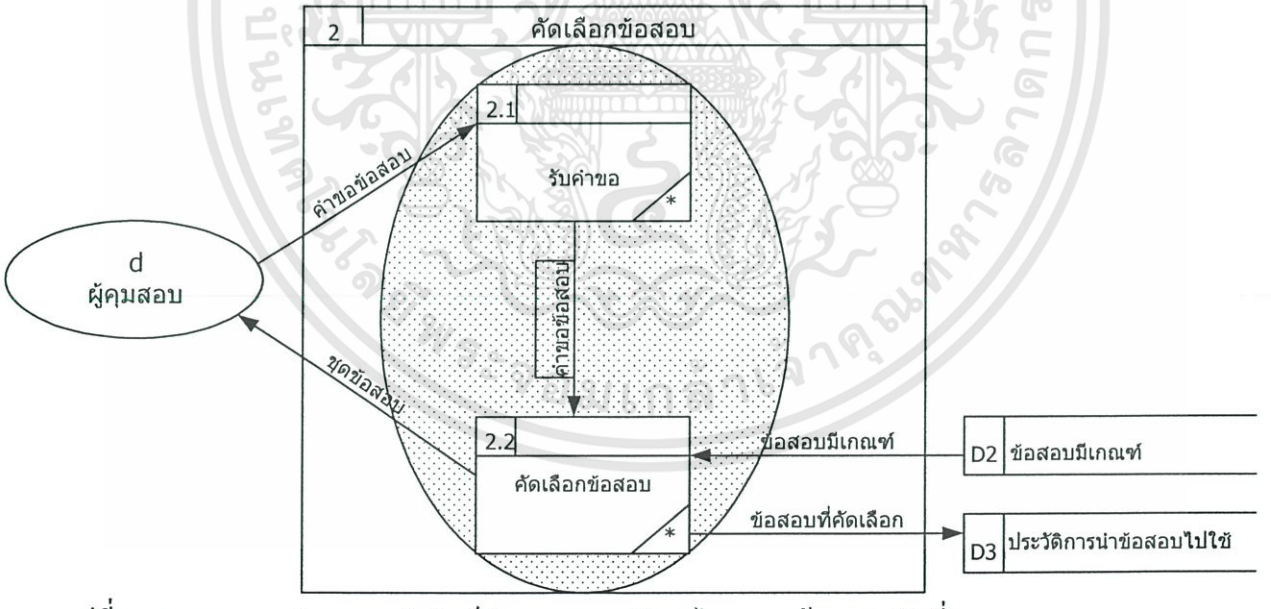
ตารางที่ 6.59 แสดงรายละเอียดของการทำบรรทัดฐานฐานข้อมูล

UNF	level	1NF	2NF	3NF	Table
<u>รหัสข้อสอบ</u>	1	<u>รหัสข้อสอบ</u>	<u>รหัสข้อสอบ</u>	<u>รหัสข้อสอบ</u>	
รหัสคณะ	1	รหัสอาจารย์	<u>รหัสอาจารย์</u>	<u>รหัสอาจารย์</u>	
รหัสวิชา	1	รหัสคณะ	<u>รหัสคณะ</u>	<u>รหัสวิชา</u>	
บทที่	1	รหัสวิชา	<u>รหัสวิชา</u>	<u>รหัสคณะ</u>	ข้อสอบ
ประเภทข้อสอบ	1	บทที่	บทที่	ประเภทข้อสอบ	
โจทย์	1	ประเภทข้อสอบ	ประเภทข้อสอบ	โจทย์	
ความตรง	1	โจทย์	โจทย์	ความตรง	
ความยาก	1	ความตรง	ความตรง	ความยาก	
ความน่าเชื่อถือ	1	ความยาก	ความยาก	ความน่าเชื่อถือ	
ตัวเลือกที่	1	ความน่าเชื่อถือ	ความน่าเชื่อถือ	วันที่บันทึกข้อสอบ	
เนื้อหา	1	ตัวเลือกที่	ตัวเลือกที่	วันที่ให้เกณฑ์ข้อสอบ	
เฉลย	1	เนื้อหา	เนื้อหา		
วันที่บันทึกข้อสอบ	1	เฉลย	เฉลย	<u>รหัสข้อสอบ</u>	
วันที่ให้เกณฑ์ข้อสอบ	1	วันที่บันทึกข้อสอบ	วันที่บันทึกข้อสอบ	<u>รหัสคณะ</u>	
วันที่ใช้ข้อสอบครั้งสุด		วันที่ให้เกณฑ์ข้อสอบ	วันที่ให้เกณฑ์ข้อสอบ	<u>รหัสวิชา</u>	
ท้าย	1	วันที่ใช้ข้อสอบครั้งสุด		บทที่	รายละเอียดข้อสอบ
ปีการศึกษา	1	ท้าย	<u>รหัสข้อสอบ</u>	ตัวเลือกที่	
เทอม	1	ปีการศึกษา	<u>รหัสวิชา</u>	เนื้อหา	
วันที่สอบ	1	เทอม	<u>รหัสคณะ</u>	เฉลย	
เวลาสอบ	1	วันที่สอบ	วันที่ใช้ข้อสอบครั้งสุด		
รหัสอาจารย์	2	เวลาสอบ	ท้าย	<u>รหัสข้อสอบ</u>	
ชื่ออาจารย์	2		ปีการศึกษา	<u>รหัสคณะ</u>	
ที่อยู่อาจารย์	2	<u>รหัสอาจารย์</u>	เทอม	<u>รหัสวิชา</u>	ประวัติการใช้ข้อสอบ
		ชื่ออาจารย์	วันที่สอบ	วันที่ใช้ข้อสอบครั้งสุด	
			เวลาสอบ	ท้าย	
				ปีการศึกษา	
			<u>รหัสอาจารย์</u>	เทอม	
			ชื่ออาจารย์	วันที่สอบ	
				เวลาสอบ	
				<u>รหัสอาจารย์</u>	อาจารย์
				ชื่ออาจารย์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

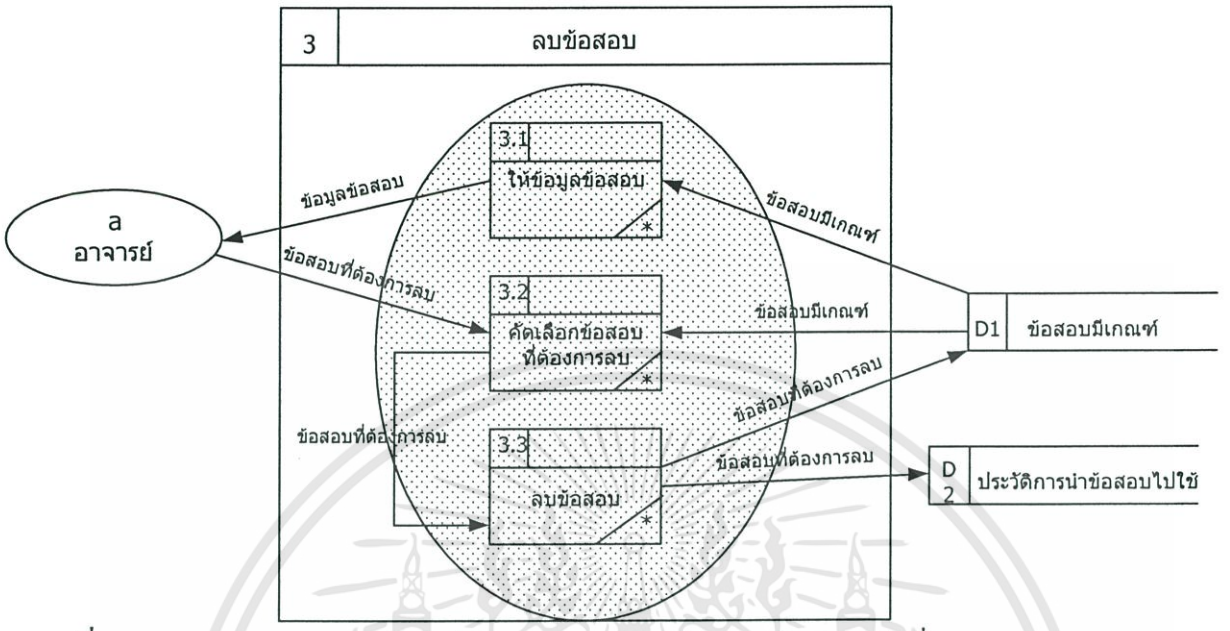


รูปที่ 6.33 แสดงการกำหนดกลุ่มฟังก์ชันของแผนภูมิการไหลของข้อมูลระดับที่ 1



รูปที่ 6.34 แสดงการกำหนดกลุ่มฟังก์ชันของแผนภูมิการไหลของข้อมูลระดับที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.35 แสดงการกำหนดกลุ่มฟังก์ชันของแผนภูมิการไหลของข้อมูลระดับที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.60 แสดงรายละเอียดของฟังก์ชันเก็บตัวข้อสอบ

Function Name: เก็บตัวข้อสอบ	**Function Id:** 01
Function Type: Update/On-Line/User	
Function Description: เป็นฟังก์ชันที่ทำหน้าที่เก็บข้อมูลตัวข้อสอบ โดยลำดับแรกของการเก็บตัวข้อสอบคือระบบจะกำหนดรหัสข้อสอบให้เพื่อให้ผู้ใช้ป้อนข้อมูลวิชาและบทที่ของข้อสอบเพื่อให้ระบบกำหนดหมายเลขข้อของข้อสอบ จากนั้นผู้ใช้จึงจะใส่ข้อมูลข้อสอบส่วนที่เหลือให้ครบได้	
Error Handling:	
DFD Processes: 1.1 (เก็บข้อสอบรายข้อ)	
Events: บันทึกข้อสอบ	**Event Frequency:** 1
I/O Structures:	
I/O Description: a-1.1	
Requirements Catalogue Ref.: 01	
Related Function: None	
Enquires: None	**Enquiry Frequency:**
Common Processing: None	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.61 แสดงรายละเอียดของฟังก์ชันเก็บเกณฑ์และรวบรวม

Function Name: เก็บเกณฑ์และรวบรวม	**Function Id:** 02
Function Type: Update/On-Line/User	
Function Description: บันทึกเกณฑ์ข้อสอบให้กับข้อสอบที่มีอยู่ในฐานข้อมูลแต่ยังไม่มีเกณฑ์ โดยจะต้องนำข้อมูลข้อสอบที่จะบันทึกเกณฑ์ขึ้นมาแสดงบนจอภาพก่อนเพื่อที่จะได้ทำการบันทึกเกณฑ์ข้อสอบ	
Error Handling: จะไม่สามารถบันทึกเกณฑ์ได้ถ้าไม่สามารถดึงข้อสอบ	
DFD Processes: 1.2 (สร้างเกณฑ์ข้อสอบ), 1.3 (รวบรวมข้อสอบและเกณฑ์ข้อสอบ)	
Events: บันทึกข้อสอบ	**Event Frequency:** 1
I/O Structures:	
I/O Description: b-1.2	
Requirements Catalogue Ref.: 02	
Related Function: 01	
Enquires: สอบถามข้อมูล	**Enquiry Frequency: 1**
Common Processing: None	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.62 แสดงรายละเอียดของฟังก์ชันจัดทำชุดข้อสอบ

Function Name: จัดทำชุดข้อสอบ	**Function Id:** 03
Function Type: Enquiry/On-Line/User	
Function Description: ผู้ใช้ใส่ข้อมูลความต้องการข้อสอบเพื่อให้ระบบจัดการเลือกข้อสอบตามงานที่กำหนด แล้วพิมพ์	
Error Handling:	
DFD Processes: 2.1(รับคำขอ), 2.2(คัดเลือกข้อสอบ)	
Events: บันทึกการใช้ข้อสอบ	**Event Frequency:** 1
I/O Structures:	
I/O Description: d-2.1, 2.2-d	
Requirements Catalogue Ref.: 04	
Related Function: None	
Enquires: None	**Enquiry Frequency:** None
Common Processing: None	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.63 แสดงรายละเอียดของฟังก์ชันลบข้อสอบที่ไม่ต้องการ

Function Name: ลบข้อสอบที่ไม่ต้องการ	**Function Id:** 04
Function Type: Update/On-Line/User	
Function Description: เป็นฟังค์ชั่นที่ใช้สำหรับลบข้อสอบที่ไม่ต้องการหรือข้อสอบที่ไม่สมบูรณ์เพื่อการบันทึกใหม่	
Error Handling:	
DFD Processes: 3.1 (ให้ข้อมูลข้อสอบ), 3.2(คัดเลือกข้อสอบที่ต้องการลบ), 3.3(ลบข้อสอบ)	
Events: ยอมรับการลบข้อสอบ	**Event Frequency:** 0.8
I/O Structures:	
I/O DescriptionDFD: 3.1-c, c-3.2	
Requirements Catalogue Ref.: 03	
Related Function: None	
Enquires: สอบถามข้อมูล	**Enquiry Frequency:** 1
Common Processing:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.64 แสดงรายละเอียดของฟังก์ชันสอบถามข้อมูล

Function Name: สอบถามข้อมูล	**Function Id:** 05
Function Type: Enquiry/On-Line/User	
Function Description: สอบถามข้อมูลข้อสอบ ประวัติการนำไปใช้ ผู้ออกข้อสอบ ผู้ให้เกณฑ์ โดยแสดงผลออกทางจอภาพ	
Error Handling: None	
DFD Processes: None	
Events: None	**Event Frequency:** None
I/O Structures:	
I/O DescriptionDFD: None	
Requirements Catalogue Ref.: 05	
Related Function: None	
Enquires: สอบถามข้อมูล	**Enquiry Frequency:** 1
Common Processing:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.65 แสดงรายละเอียดของฟังก์ชันเพิ่มอาจารย์และผู้ให้เกณฑ์

Function Name: เพิ่มอาจารย์,ผู้ให้เกณฑ์	**Function Id:** 06
Function Type: Update/On-Line/User	
Function Description: เพิ่มข้อมูลอาจารย์ ผู้ให้เกณฑ์สำหรับข้อมูลที่ยังไม่มีในฐานข้อมูล	
Error Handling: None	
DFD Processes: None	
Events: บันทึกข้อมูลอาจารย์ บันทึกผู้ให้เกณฑ์	**Event Frequency:** None
I/O Structures:	
I/O DescriptionDFD: None	
Requirements Catalogue Ref.: 06	
Related Function: None	
Enquires: None	**Enquiry Frequency:** 1
Common Processing:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.66 แสดงรายละเอียดของฟังก์ชันแก้ไขข้อมูลอาจารย์และผู้ให้เกณฑ์

Function Name: แก้ไขข้อมูลอาจารย์,ผู้ให้เกณฑ์	**Function Id:** 07
Function Type: Update/On-Line/User	
Function Description: แก้ไขข้อมูลอาจารย์ ผู้ให้เกณฑ์เพิ่มเปลี่ยนแปลงข้อมูลให้เป็นปัจจุบัน	
Error Handling: None	
DFD Processes: None	
Events: แก้ไขข้อมูลอาจารย์ แก้ไขข้อมูลผู้ให้เกณฑ์	**Event Frequency:** None
I/O Structures:	
I/O DescriptionDFD: None	
Requirements Catalogue Ref.: 07	
Related Function: None	
Enquires: None	**Enquiry Frequency:** 1
Common Processing:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.67 แสดงรายละเอียดของอินพุต-เอาต์พุต

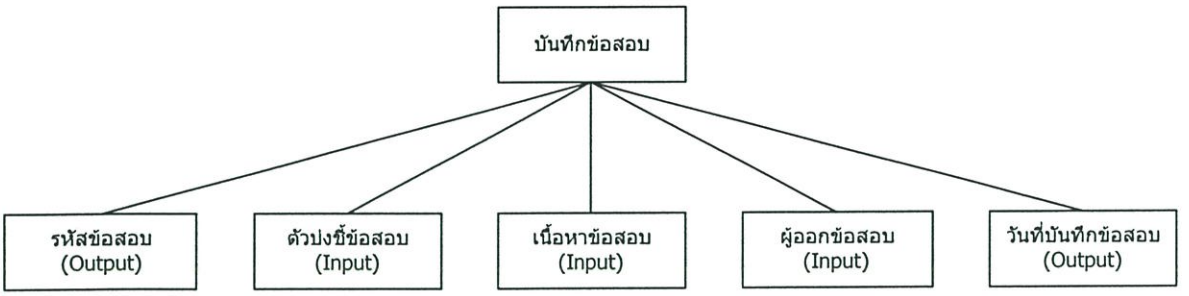
I/O Description				
From	To	**Data Flow Name**	**Data Content**	**Comments**
A	**1.1**	ข้อสอบรายข้อ	วิชา บทที่ โจทย์ ตัวเลือก เฉลย ชื่อผู้ออกข้อสอบ วันที่ออกข้อสอบ	หมายเลขข้อจะถูกกำหนดให้โดยระบบ
B	**1.2**	เกณฑ์ข้อสอบ	เกณฑ์ข้อสอบ ชื่อผู้ให้เกณฑ์ วันที่ให้เกณฑ์	ให้เกณฑ์กับข้อสอบที่สร้างแล้ว
D	**2.1**	คำขอข้อสอบ	วิชา บทที่ จำนวนข้อ เกณฑ์ข้อสอบ	เกณฑ์มี 4 ข้อ 1. ความยาก 2. ความถูกต้อง 3. ความน่าเชื่อถือ 4. อำนาจจำแนก
2.2	**D**	ชุดข้อสอบ	วิชา บทที่ เนื้อหาข้อสอบ เฉลย	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

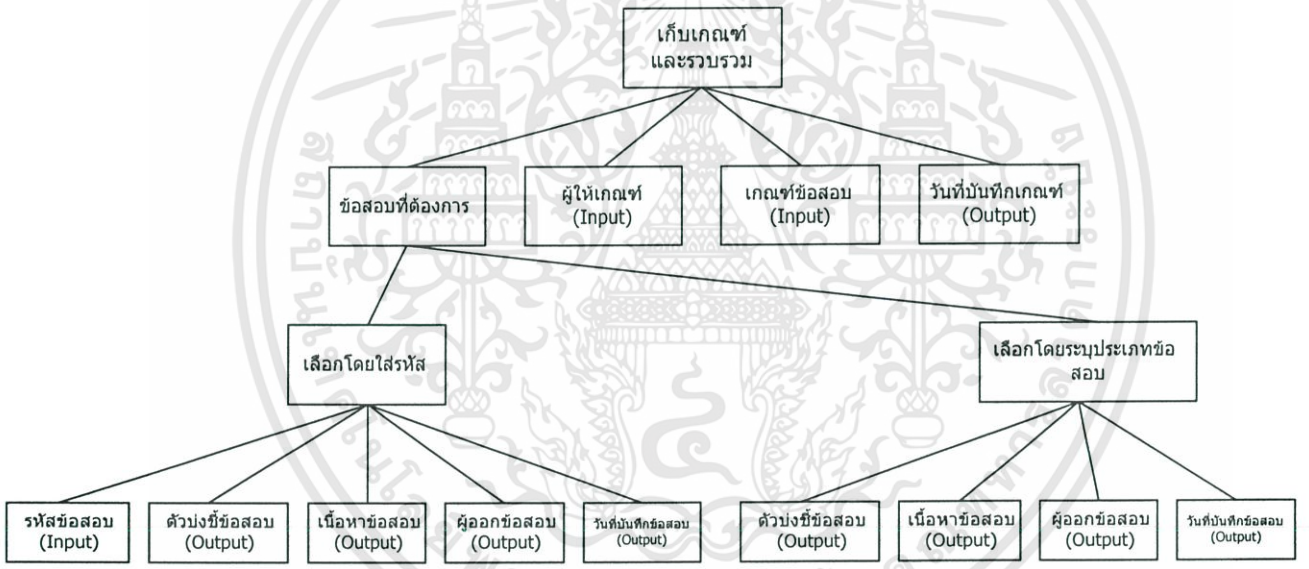
ตารางที่ 6.68 แสดงรายละเอียดของอินพุต-เอาต์พุต (ต่อ)

I/O Description				
From	To	**Data Flow Name**	**Data Content**	**Comments**
3.1	**C**	ข้อมูลข้อสอบ	หมายเลขข้อสอบ วิชา บทที่ ข้อที่ โจทย์ ตัวเลือก เฉลย ชื่อผู้ออกข้อสอบ วันที่ออกข้อสอบ เกณฑ์ข้อสอบ ชื่อผู้ออกเกณฑ์	หมายเลขข้อจะถูกกำหนดให้โดยระบบ
C	**3.1**	ข้อสอบที่ต้องการลบ	หมายเลขข้อสอบ วิชา บทที่ ข้อที่	ลบข้อสอบจากวิชา บทที่ และข้อที่เลือกไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

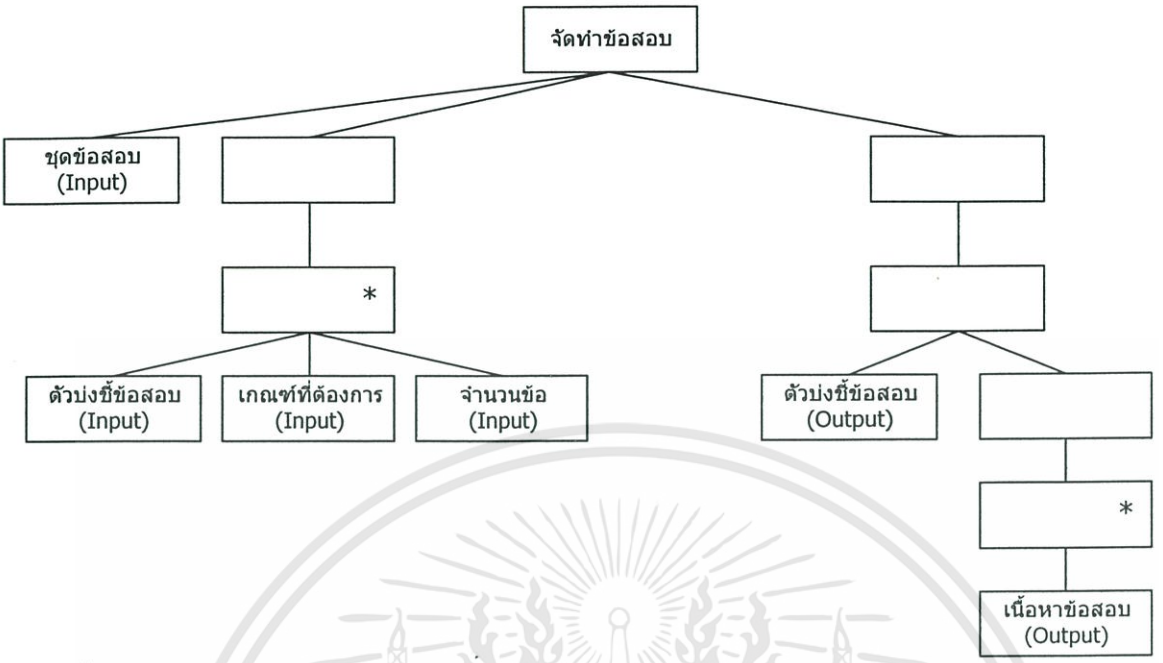


รูปที่ 6.36 แสดงแผนภูมิโครงสร้าง I/O ของการบันทึกข้อสอบ

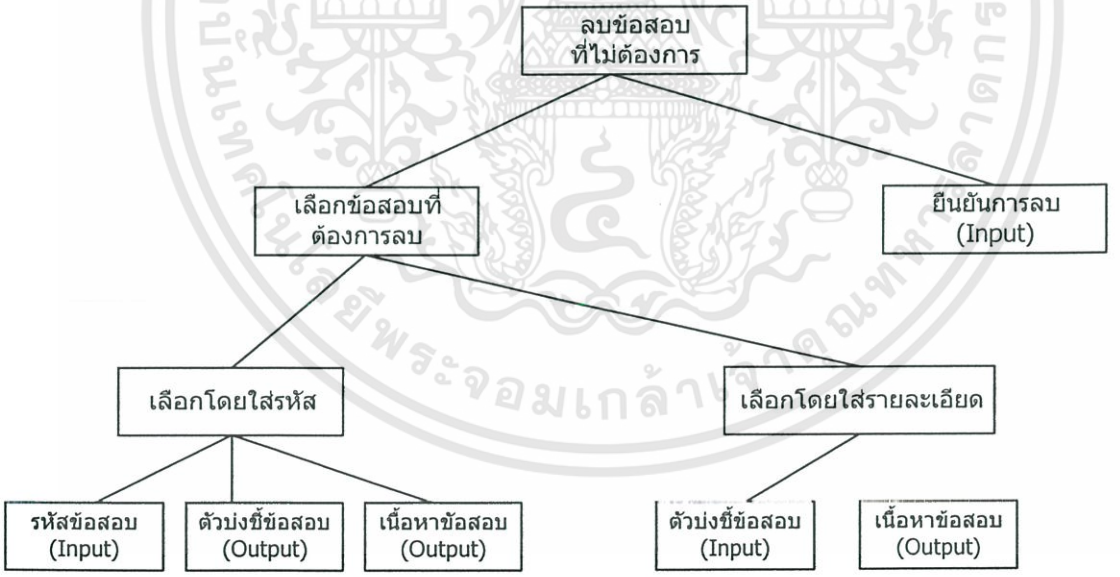


รูปที่ 6.37 แสดงแผนภูมิโครงสร้าง I/O ของการบันทึกเกณฑ์และรวบรวมข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.38 แสดงแผนภูมิโครงสร้าง I/O ของการจัดทำข้อสอบ



รูปที่ 6.39 แสดงแผนภูมิโครงสร้าง I/O ของการลบข้อสอบที่ไม่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.69 แสดงรายละเอียดของแผนภูมิ I/O ส่วนบันทึกข้อสอบ

I/O Structure Name: บันทึกข้อสอบ		
I/O Structure Element	Data Item	Comments
รหัสข้อสอบ	รหัสข้อสอบ	
ข้อสอบ	วิชา บทที่ ข้อที่	
เนื้อหาข้อสอบ	โจทย์ ตัวเลือกที่ 1 ตัวเลือกที่ 2 ตัวเลือกที่ 3 ตัวเลือกที่ 4 ตัวเลือกที่ 5 เฉลย	
ผู้ออกข้อสอบ	รหัสผู้ออกข้อสอบ ชื่อผู้ออกข้อสอบ	
วันที่บันทึกข้อสอบ	วันที่บันทึกข้อสอบ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.70 แสดงรายละเอียดของแผนภูมิ I/O ส่วนบันทึกเกณฑ์และรวบรวมข้อสอบ

I/O Structure Name: บันทึกเกณฑ์และรวบรวมข้อสอบ		
I/O Structure Element	Data Item	Comments
รหัสข้อสอบ	รหัสข้อสอบ	
ข้อสอบ	วิชา บทที่ ข้อที่	
เนื้อหาข้อสอบ	โจทย์ ตัวเลือกที่ 1 ตัวเลือกที่ 2 ตัวเลือกที่ 3 ตัวเลือกที่ 4 ตัวเลือกที่ 5 เฉลย	
ผู้ออกข้อสอบ	รหัสผู้ออกข้อสอบ ชื่อผู้ออกข้อสอบ	
วันที่บันทึกข้อสอบ	วันที่บันทึกข้อสอบ	
เกณฑ์ข้อสอบ	ความยาก ความตรง ความน่าเชื่อถือ	
วันที่บันทึกเกณฑ์	วันที่บันทึกเกณฑ์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.71 แสดงรายละเอียดของแผนภูมิ I/O ส่วนจัดทำข้อสอบ

I/O Structure Name: จัดทำข้อสอบ		
I/O Structure Element	Data Item	Comments
ชุดข้อสอบ	รหัสชุดข้อสอบ	
ตัวบ่งชี้ข้อสอบ	กลุ่มข้อสอบ วิชา บทที่	
เกณฑ์ข้อสอบ	ความยาก ความตรง ความน่าเชื่อถือ อำนาจจำแนก	
จำนวนข้อ	จำนวนข้อ	
เนื้อหาข้อสอบ	รหัสชุดข้อสอบ รหัสข้อสอบ วิชา บทที่ ข้อที่ โจทย์ ตัวเลือกที่ 1 ตัวเลือกที่ 2 ตัวเลือกที่ 3 ตัวเลือกที่ 4 ตัวเลือกที่ 5 เฉลย	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

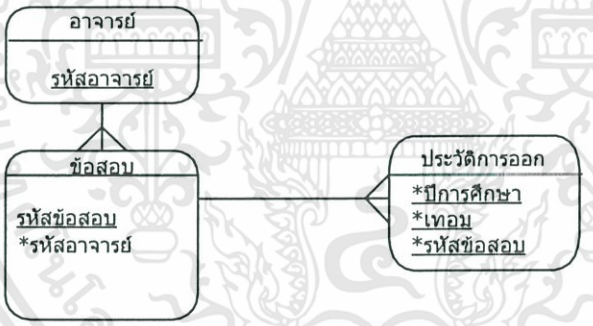
ตารางที่ 6.72 แสดงรายละเอียดของแผนภูมิ I/O ส่วนการลบข้อสอบ

I/O Structure Name: ลบข้อสอบที่ไม่ต้องการ		
I/O Structure Element	Data Item	Comments
รหัสข้อสอบ	รหัสข้อสอบ	
ตัวบ่งชี้ข้อสอบ	วิชา บทที่ ข้อที่	
เนื้อหาข้อสอบ	วิชา ข้อที่ โจทย์ ตัวเลือกที่ 1 ตัวเลือกที่ 2 ตัวเลือกที่ 3 ตัวเลือกที่ 4 ตัวเลือกที่ 5	
ยืนยันการลบ	ตัวยืนยัน	เป็น Yes หรือ No

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

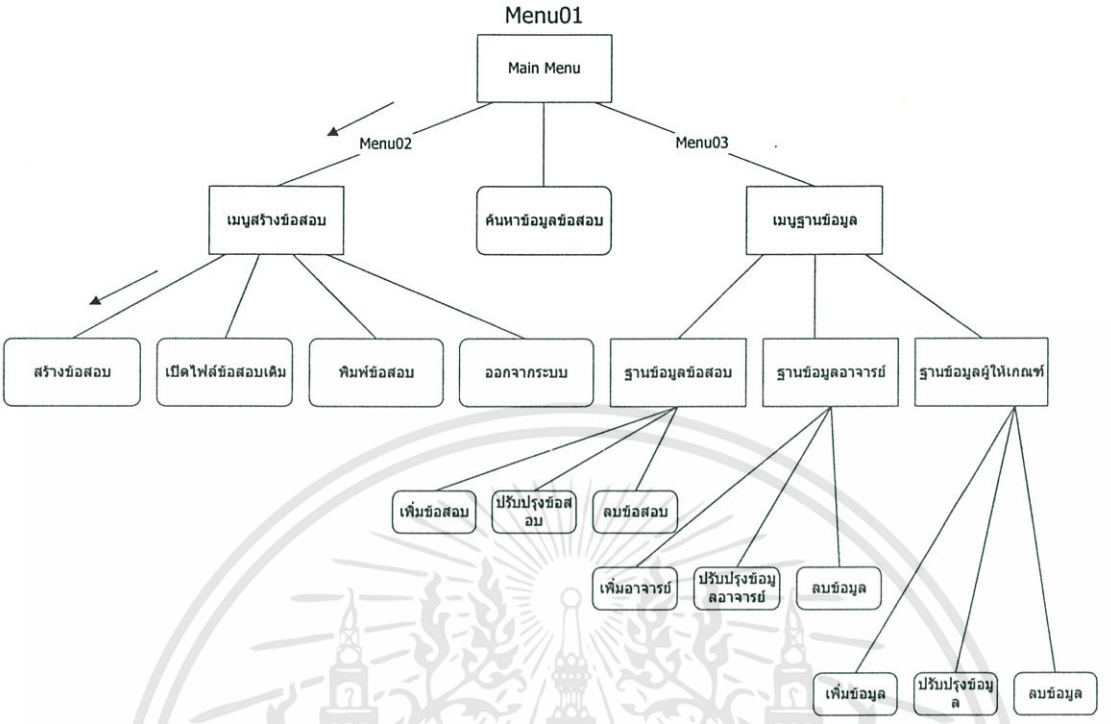
ตารางที่ 6.73 แสดงรายละเอียดของตารางความสัมพันธ์ระหว่างฟังก์ชันกับผู้ใช้

Function / User role	เก็บตัวข้อสอบ	เก็บเกณฑ์และรวบรวม	จัดทำชุดข้อสอบ	ลบข้อสอบที่ไม่ต้องการ
ผู้รวบรวมข้อสอบ	X	X		
ผู้คัดเลือกข้อสอบ			X	
ผู้ลบข้อสอบ				X



รูปที่ 6.40 แสดงความสัมพันธ์ของ LDS พร้อมกับคีย์หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.41 แสดงเส้นทางเมนูของแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.42 แสดงเส้นทางเมนูที่ 1 ของแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.74 แสดงรายละเอียดเส้นทางที่ 1 ของแบบจำลอง

Pathway No: 001	**Function Name:** สร้างข้อสอบ	**User Role:** อาจารย์
Agenda: 1. อภิปรายรายละเอียดตกลงใจจุดประสงค์การจัดทำ Prototype และการสร้าง Procedure prototype 2. อธิบายรายละเอียดถึงขั้นตอนการทำงานของ prototype tool 3. แสดงการทำงานของ prototype ต่อ user 4. ตกลงในข้อมูลรายละเอียด ที่ได้รับจากการ present prototype เพื่อพิจารณาแนวทางที่เป็นไปได้ในการปรับปรุงระบบและทำการ re-demonstration		

Component No	**Discussion Point**
4 4 4 4 4	ทำการออกข้อสอบให้ตรงตามข้อมูลที่มีอยู่ในฐานข้อมูลข้อสอบ พร้อมกับแจ้งให้ทราบถ้าข้อมูลไม่อยู่ในฐานข้อมูล มีการตรวจสอบข้อสอบที่เคยใช้งานติดต่อกันเกิน 2 ครั้ง และมีการยืนยันการทำงานก่อนการตัดสินใจหรือไม่ สุดท้ายทำการตรวจสอบรายละเอียดของงานให้ตรงกับความต้องการพร้อมกับตรวจสอบประสิทธิภาพของงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

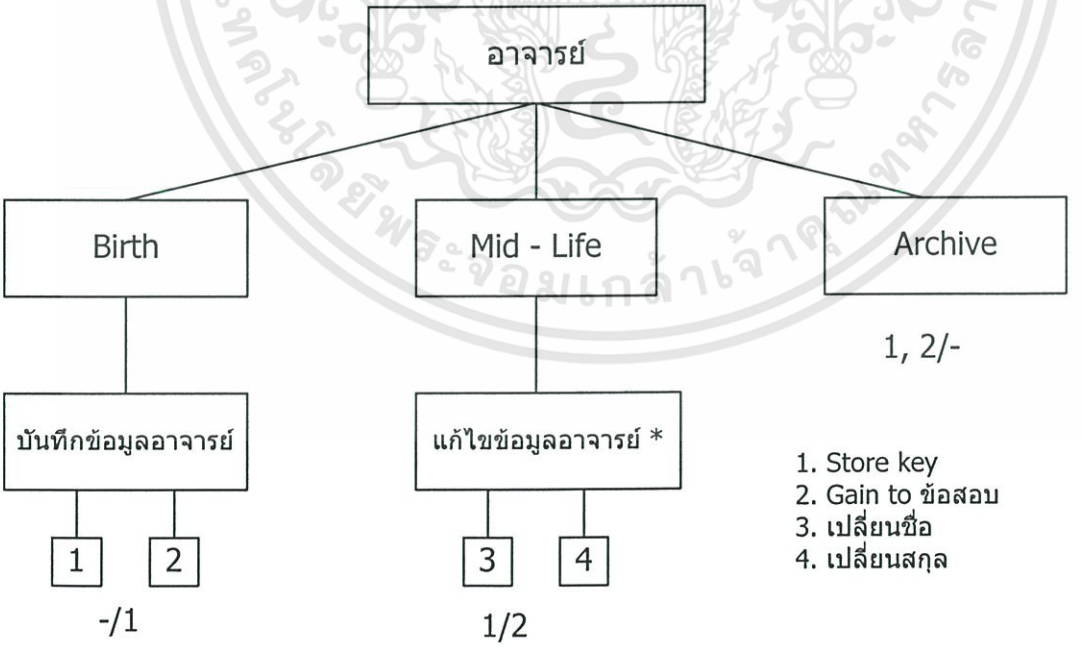
ตารางที่ 6.75 แสดงรายละเอียดของความสัมพันธ์ระหว่างเหตุการณ์กับเอนติตี

Entity \ Event	บันทึกข้อสอบ	บันทึกเกณฑ์	บันทึกการใช้ข้อสอบ	ยอมรับการลบข้อสอบ	บันทึกผู้ออกข้อสอบ	บันทึกผู้ให้เกณฑ์	แก้ไขผู้ออกข้อสอบ	แก้ไขผู้ให้เกณฑ์
ผู้ออกข้อสอบ					C	C	M	M
ข้อสอบ	C	M		D				
ประวัติการออก			C/M	D				

หมายเหตุ **C = Create**

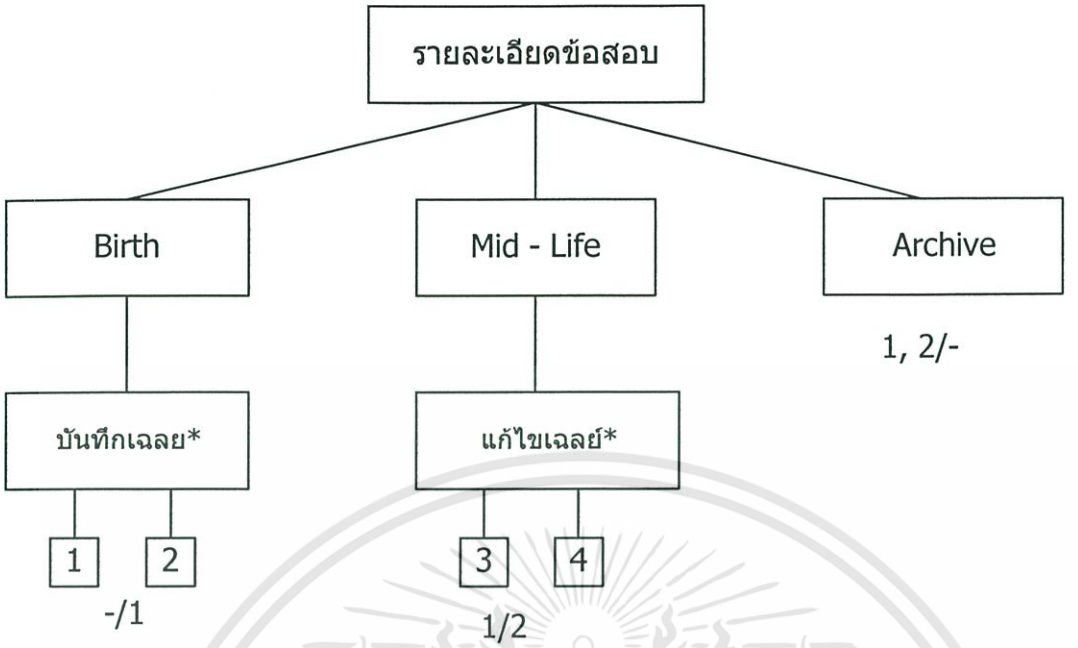
M = Modify

D = Delete

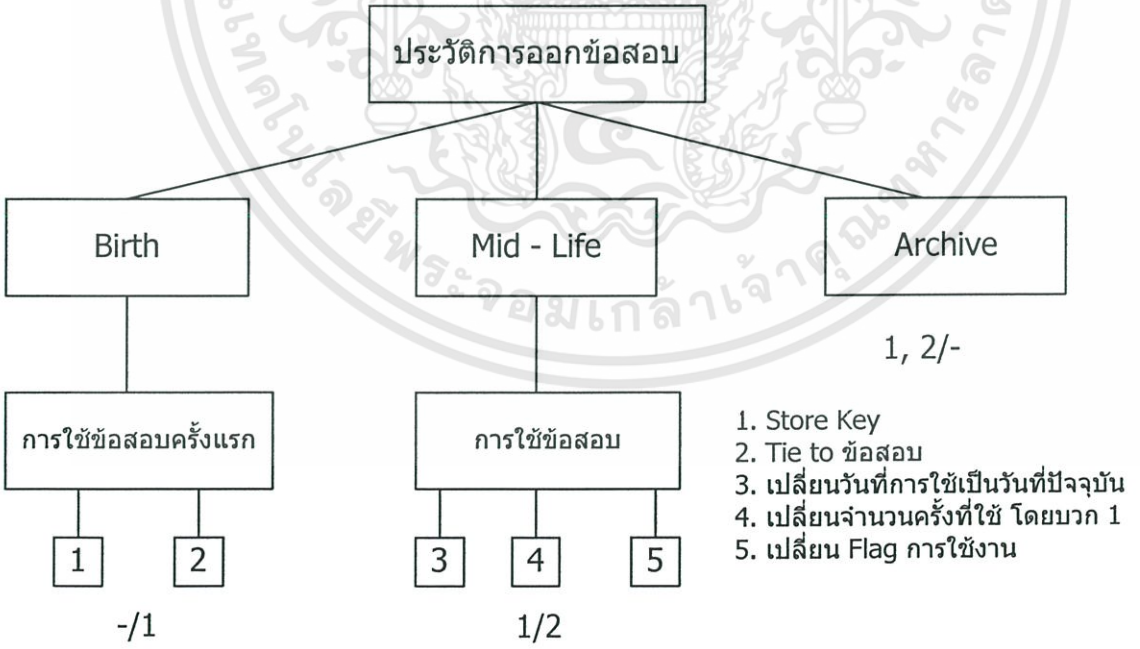


รูปที่ 6.43 แสดงเอนติตีไลฟ์ฮีสทรีไดอะแกรมของอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

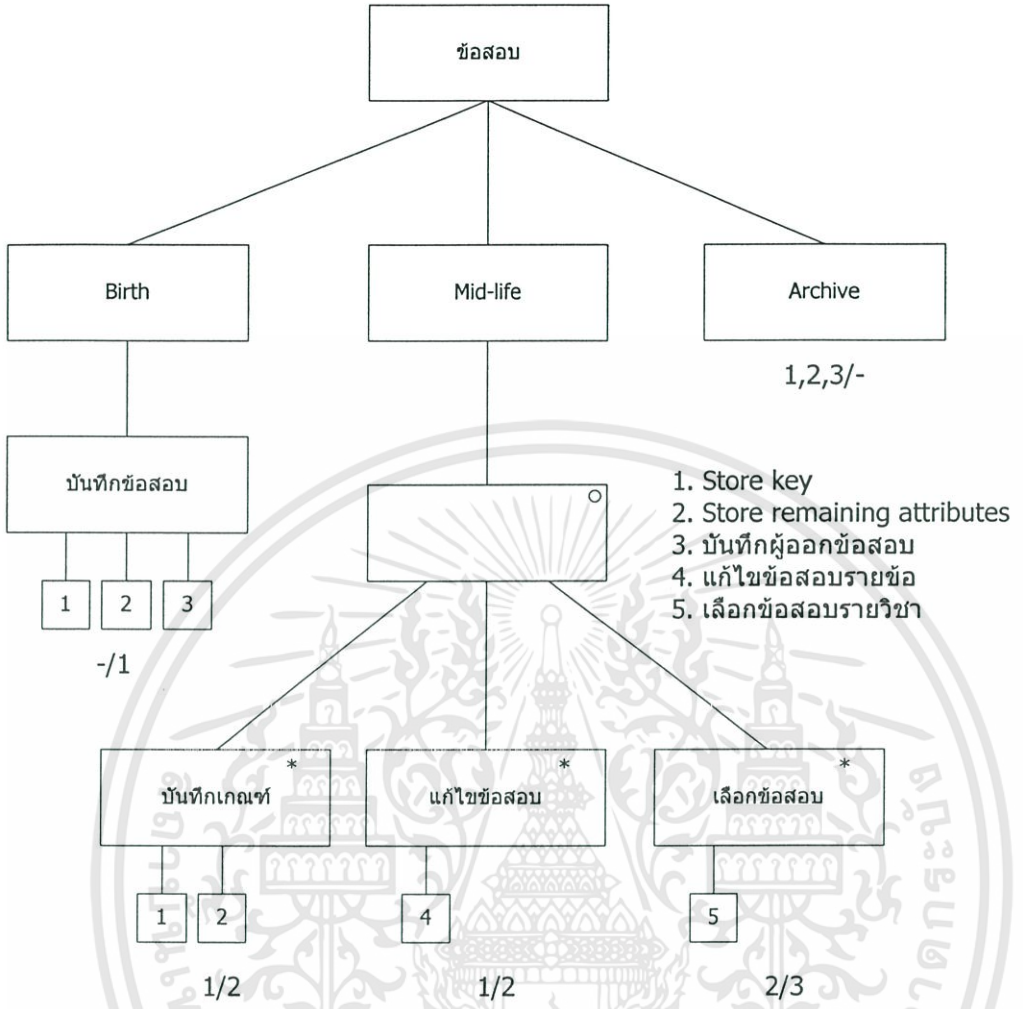


รูปที่ 6.44 แสดงเอนติตีไลฟ์ฮัสทรีไดอะแกรมของรายละเอียดข้อสอบ



รูปที่ 6.45 แสดงเอนติตีไลฟ์ฮีสทรีไดอะแกรมของประวัติการออกข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



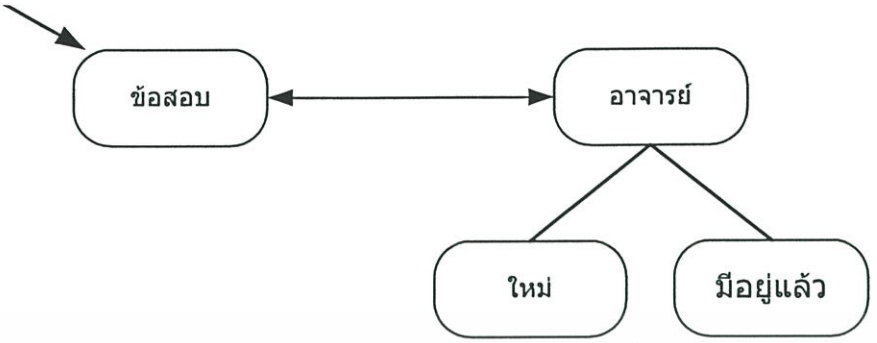
รูปที่ 6.46 แสดงเอนติตีไลฟ์ฮีสทรีไดอะแกรมของข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

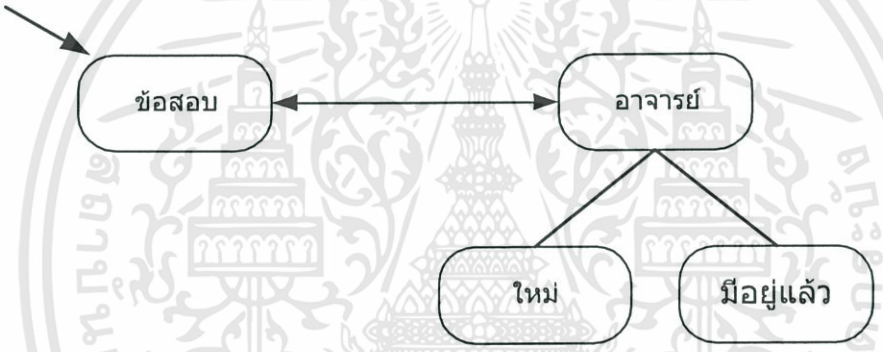
ตารางที่ 6.76 แสดงโอเปอร์เรชั่นของแต่ละเอนติตี

ENTITY	OPERATIONS
อาจารย์	1. Store key 2. Gain to ข้อสอบ 3. Replace ชื่อ 4. Replace สกุล
ข้อสอบ	1. Store key 2. Store remaining attributes 3. Tie to ผู้ออกข้อสอบ 4. แก้ไขข้อสอบรายข้อ 5. เลือกข้อสอบรายวิชา
ประวัติการออกข้อสอบ	1. Store key 2. Tie to ข้อสอบ 3. Replace วันที่ with วันที่ปัจจุบัน 4. Replace จำนวนครั้ง with จำนวนครั้ง plus 1 5. Replace flag with new flag

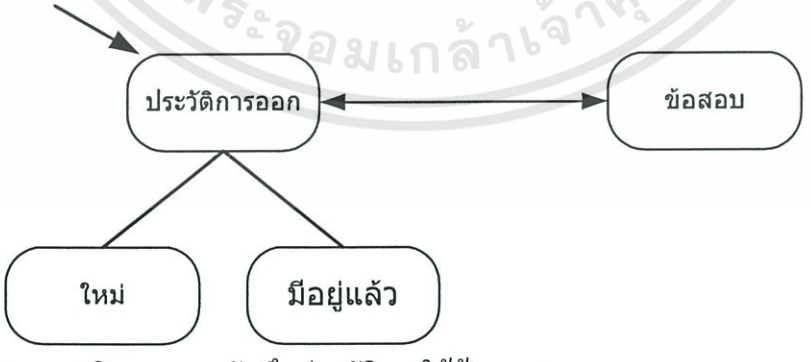
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.47 แสดงแผนภูมิ ECD การบันทึกข้อสอบกับเอนติตีอาจารย์

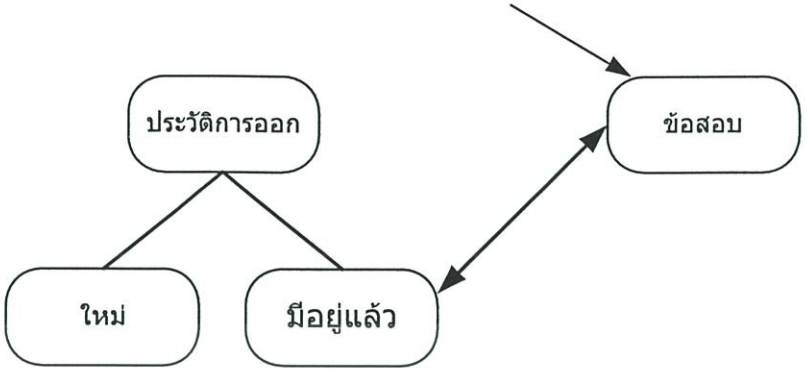


รูปที่ 6.48 แสดงแผนภูมิ ECD การบันทึกเกณฑ์ให้กับข้อสอบ

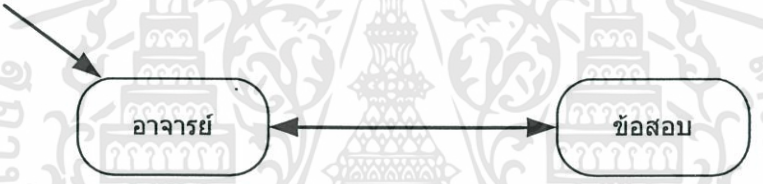


รูปที่ 6.49 แสดงแผนภูมิ ECD การบันทึกประวัติการใช้ข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



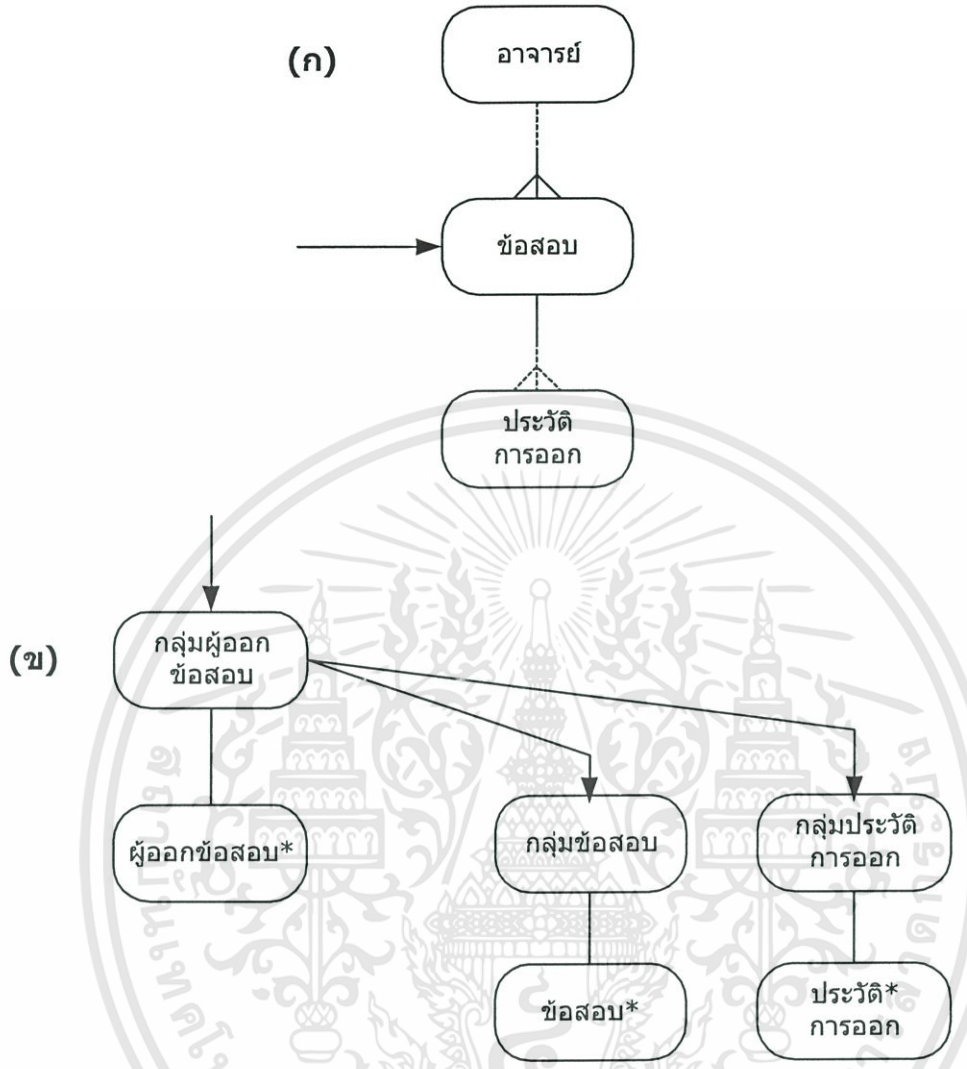
รูปที่ 6.50 แสดงแผนภูมิ ECD การลบข้อสอบ



รูปที่ 6.51 แสดงแผนภูมิ ECD การบันทึกข้อมูลอาจารย์

รูปที่ 6.52 แสดงแผนภูมิ ECD การแก้ไขข้อมูลอาจารย์

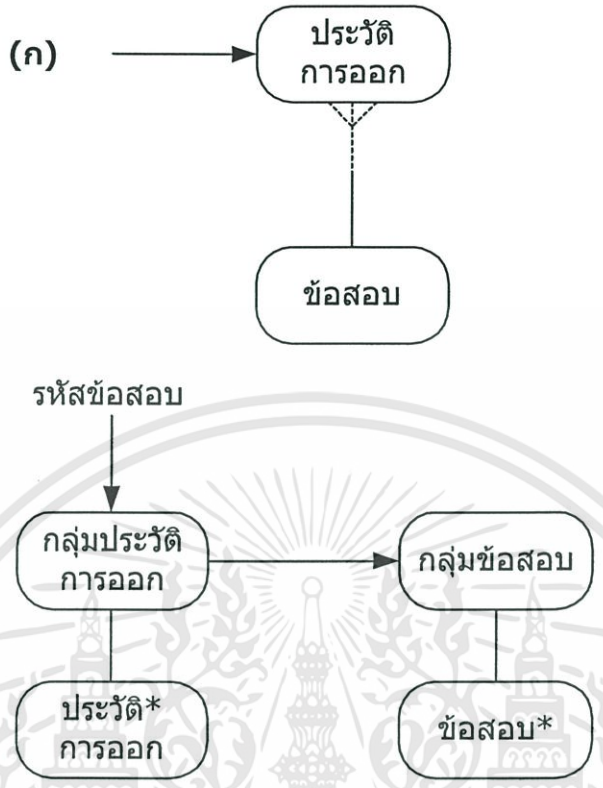
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6. 53 (ก) แสดงผลกระทบของเอนติตีข้อสอบ

(ข) แสดงแผนภูมิ EAP สำหรับการสร้างข้อสอบ

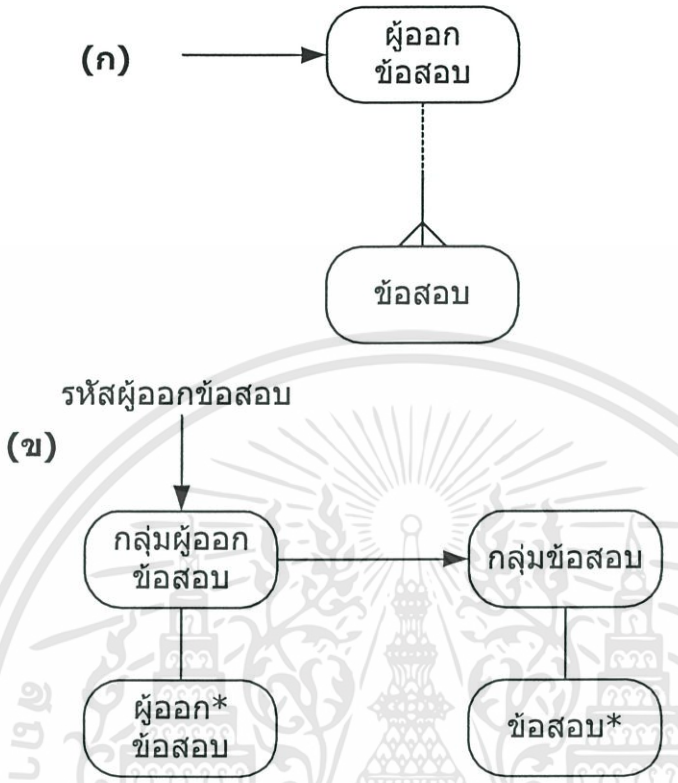
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6. 54 (ก) แสดงผลกระทบของเอนติตีประวัติข้อสอบ

(ข) แสดงแผนภูมิ EAP สำหรับการสอบถามประวัติข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

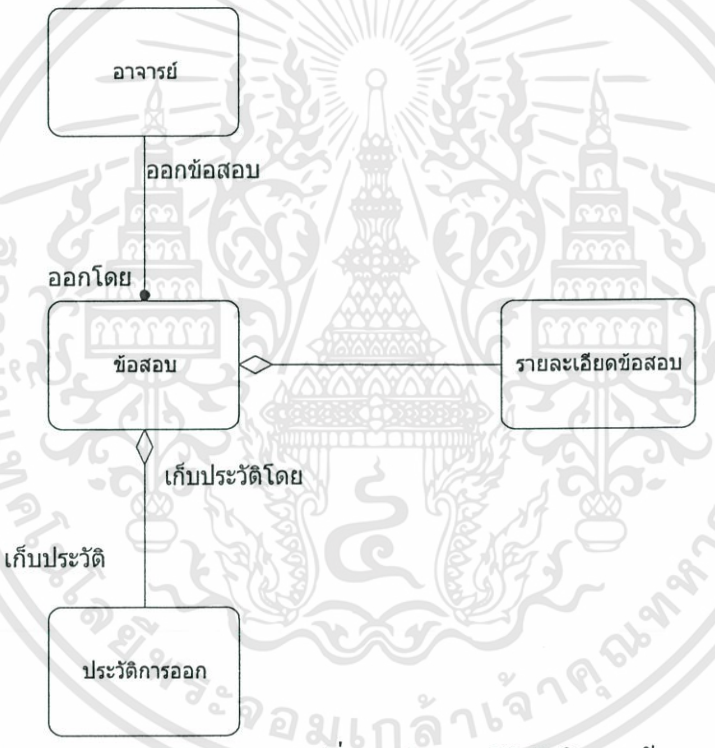


รูปที่ 6. 55 (ก) แสดงผลกระทบของเอนติตีอาจารย์

(ข) แสดงแผนภูมิ EAP สำหรับการสอบถามอาจารย์ (ผู้ออกข้อสอบ)

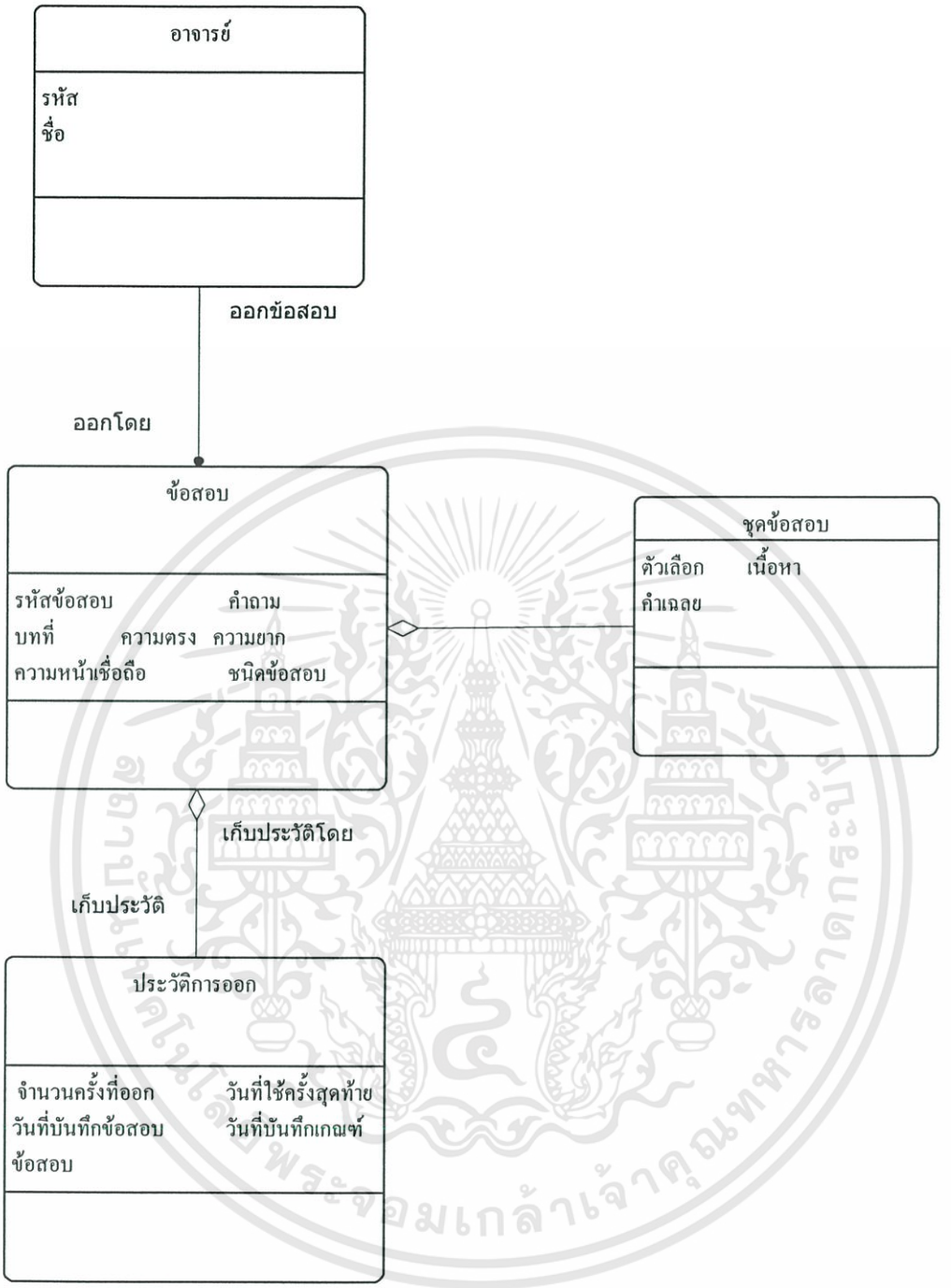
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ระยะที่ 4



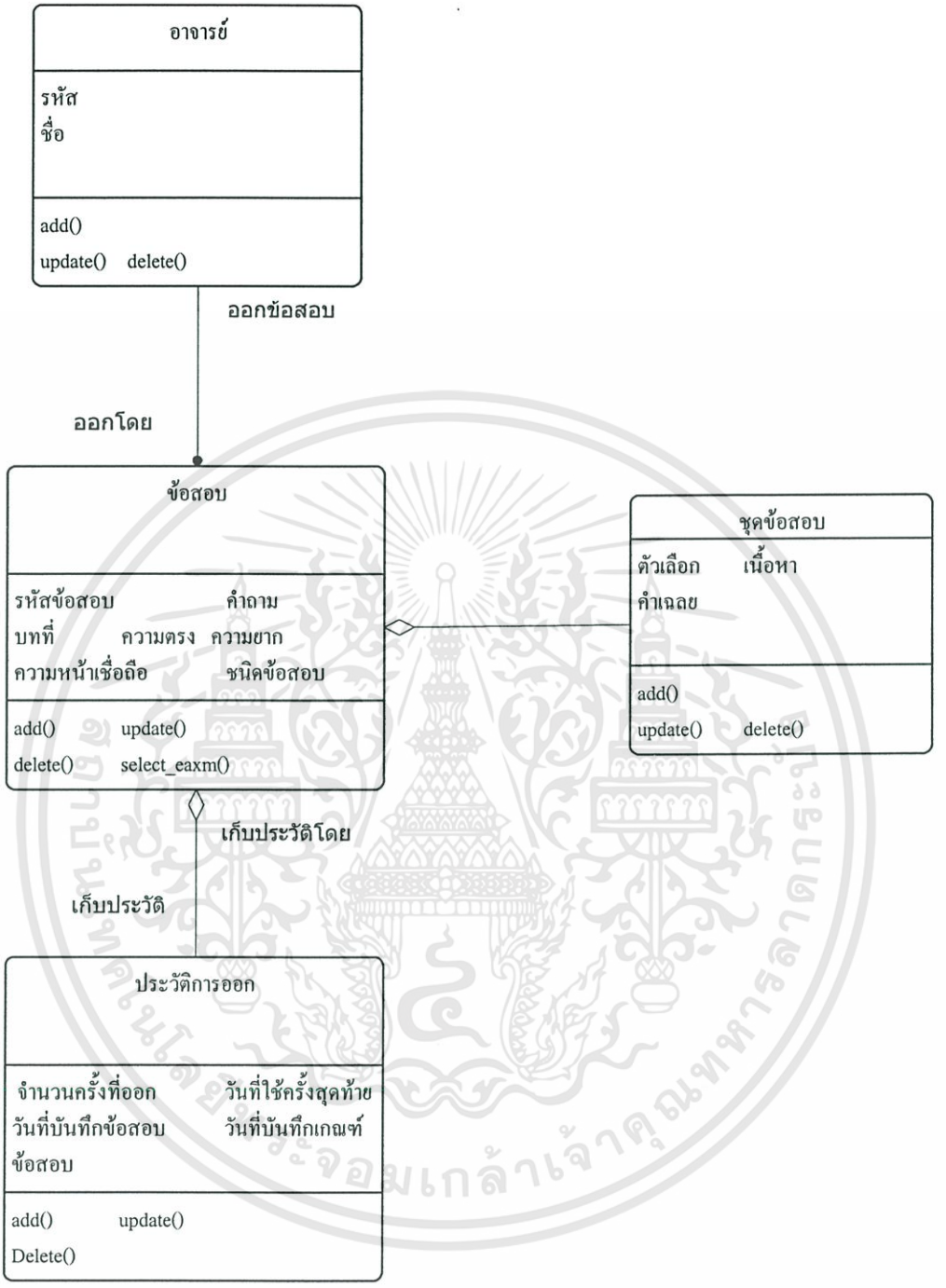
รูปที่ 6.56 แสดงไดอะแกรมของวัตถุหลังการเปลี่ยนแปลงจากวิธีเชิงโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.57 แสดงไดอะแกรมของวัตถุพร้อมกับแสดงแอตทริบิวต์ที่ปราศจากคีย์หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.58 แสดงไดอะแกรมของวัตถุพร้อมกับแอตทริบวิต์และฟังก์ชันภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.77 แสดงรายละเอียดของวัตถุอาจารย์

Class name: อาจารย์	
Class type: (e.g., device, property, role, event, . . .)	
Class characteristics: (e.g., tangible, atomic, concurrent, . . .)	
Responsibilities:	Collaborators:
รายละเอียดของอาจารย์ ภายในประกอบด้วยแอตทริบิวต์ที่จำเป็น เช่น ชื่อ ที่อยู่ สังกัด ตำแหน่งและอื่น ๆ อาจารย์สามารถแบ่งออกได้อีก 2 ประเภท เช่น อาจารย์พิเศษและอาจารย์ประจำ	• วิชา • คณะ

ตารางที่ 6.78 แสดงรายละเอียดของวัตถุข้อสอบ

Class name: ข้อสอบ	
Class type: (e.g., device, property, role, event, . . .)	
Class characteristics: (e.g., tangible, atomic, concurrent, . . .)	
Responsibilities:	Collaborators:
เก็บรายละเอียดข้อสอบทั้ง 3 ประเภทโดยแยกเก็บตามวิชาที่เปิดทำการเรียนการสอนในแต่ละคณะภายในสถาบันการศึกษา	• อาจารย์ • ประวัติการออกข้อสอบ • รายละเอียดข้อสอบ
(ประเภทข้อสอบ : ปรนัย,อัตนัย,ถูกผิด)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.79 แสดงรายละเอียดของวัตถุประวัติการออกข้อสอบ

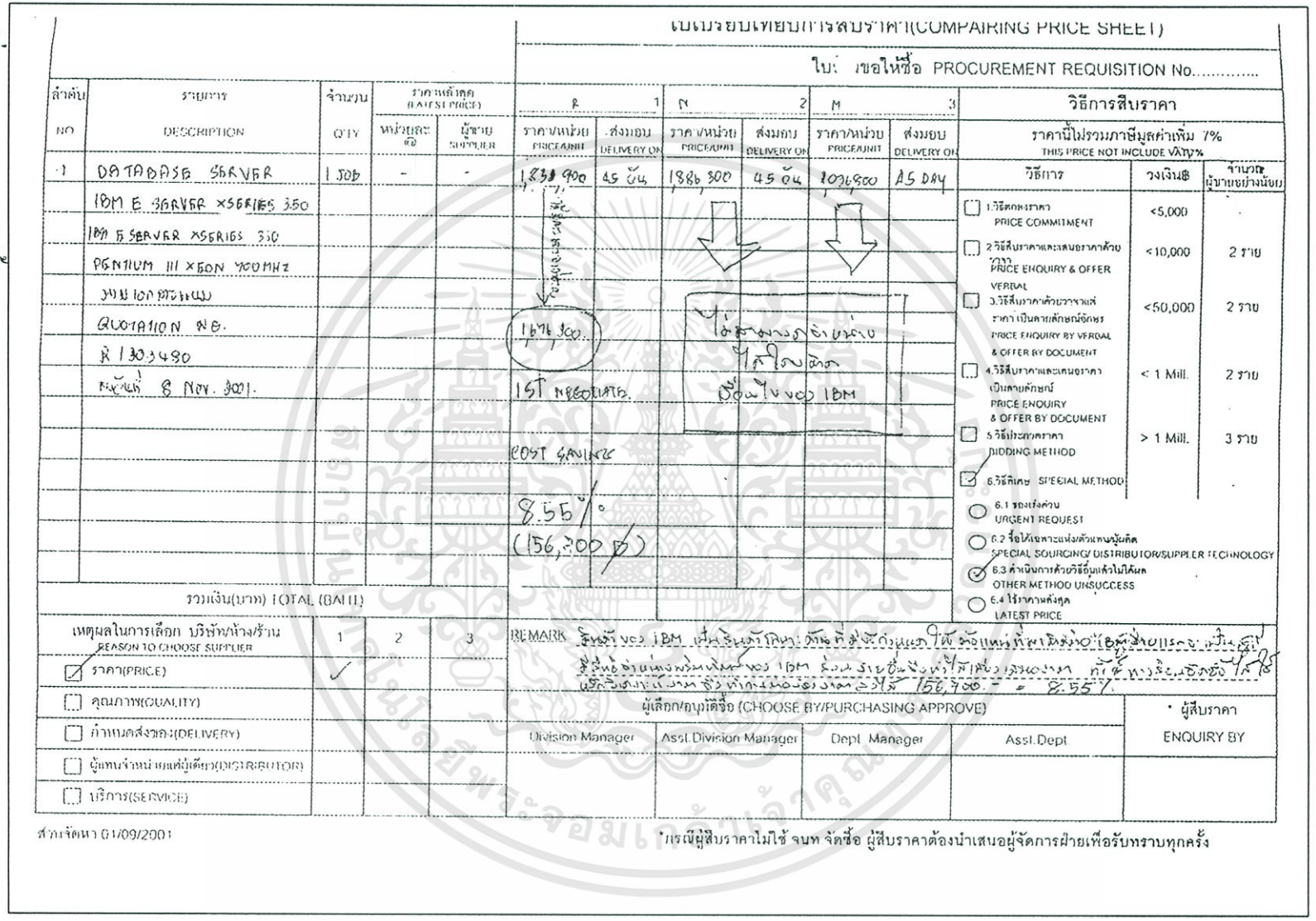
Class name: ประวัติการออกข้อสอบ	
Class type: (e.g., device, property, role, event, . . .)	
Class characteristics: (e.g., tangible, atomic, concurrent, . . .)	
Responsibilities:	Collaborators:
เก็บประวัติการใช้ข้อสอบโดยการเก็บจะเก็บบันทึกทุกครั้งที่มีการเรียกใช้ข้อสอบแต่ละข้อของข้อสอบแต่ละประเภท	• ข้อสอบ

ตารางที่ 6.80 แสดงรายละเอียดของวัตถุชุดข้อสอบ

Class name: รายละเอียดข้อสอบ	
Class type: (e.g., device, property, role, event, . . .)	
Class characteristics: (e.g., tangible, atomic, concurrent, . . .)	
Responsibilities:	Collaborators:
เก็บคำเฉลยของข้อสอบประเภท ถูกผิด และ ปรนัย	• ข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6. 59 แสดงใบขอให้ซื้อ (Procurement Requisition)



ใบเปรียบเทียบการสืบราคา(COMPAIRING PRICE SHEET)

ใบขอให้ซื้อ PROCUREMENT REQUISITION No.....

ลำดับ NO	รายการ DESCRIPTION	จำนวน QTY	ราคาหลังสุด (LATEST PRICE) หน่วยละซื้อ	ผู้ขาย SUPPLIER	1 ราคา/หน่วย PRICE/UNIT	1 ส่งมอบ DELIVERY ON	2 ราคา/หน่วย PRICE/UNIT	2 ส่งมอบ DELIVERY ON	3 ราคา/หน่วย PRICE/UNIT	3 ส่งมอบ DELIVERY ON
1	DATABASE SERVER	1 JOB	-	-	1,838,900	45 วัน	1886,300	45 วัน	1,036,800	AS DAY
	IBM E SERVER xSERIES 350									
	IBM E SERVER xSERIES 330									
	PENTIUM III xEON 700MHz									
	พร้อมอุปกรณ์									
	QUOTATION NO.				1,476,300					
	ลว 13 ก.ย. 43									
	ลงวันที่ 8 Nov. 2001.				1ST NEGOTIATE.					
					COST SAVING					
					8.55%					
					(156,200 ฿)					
รวมเงิน(บาท) TOTAL (BAHT)										

ไม่ได้ราคาที่ต่ำกว่าเดิมมาก ยี่ห้อของ IBM

วิธีการสืบราคา

ราคานี้ไม่รวมภาษีมูลค่าเพิ่ม 7% THIS PRICE NOT INCLUDE VAT7%

วิธีการ	วงเงิน฿	จำนวนผู้ขายอย่างน้อย
☐ 1.วิธีตกลงราคา PRICE COMMITMENT	<5,000	-
☐ 2.วิธีสืบราคาและเสนอราคาด้วยวาจา PRICE ENQUIRY & OFFER VERBAL	<10,000	2 ราย
☐ 3.วิธีสืบราคาด้วยวาจาแต่ราคาเป็นลายลักษณ์อักษร PRICE ENQUIRY BY VERBAL & OFFER BY DOCUMENT	<50,000	2 ราย
☐ 4.วิธีสืบราคาและเสนอราคาเป็นลายลักษณ์ PRICE ENQUIRY & OFFER BY DOCUMENT	< 1 Mill.	2 ราย
☐ 5.วิธีประกวดราคา BIDDING METHOD	> 1 Mill.	3 ราย
☑ 6.วิธีพิเศษ SPECIAL METHOD		-
○ 6.1 รีบด่วน URGENT REQUEST		
○ 6.2 ซื้อโดยเจาะแหล่งตัวแทนผู้ผลิต SPECIAL SOURCING/ DISTRIBUTOR/SUPPLIER TECHNOLOGY		
◉ 6.3 ดำเนินการด้วยวิธีอื่นแล้วไม่ได้ผล OTHER METHOD UNSUCCESS		
○ 6.4 ใช้ราคาหลังสุด LATEST PRICE		

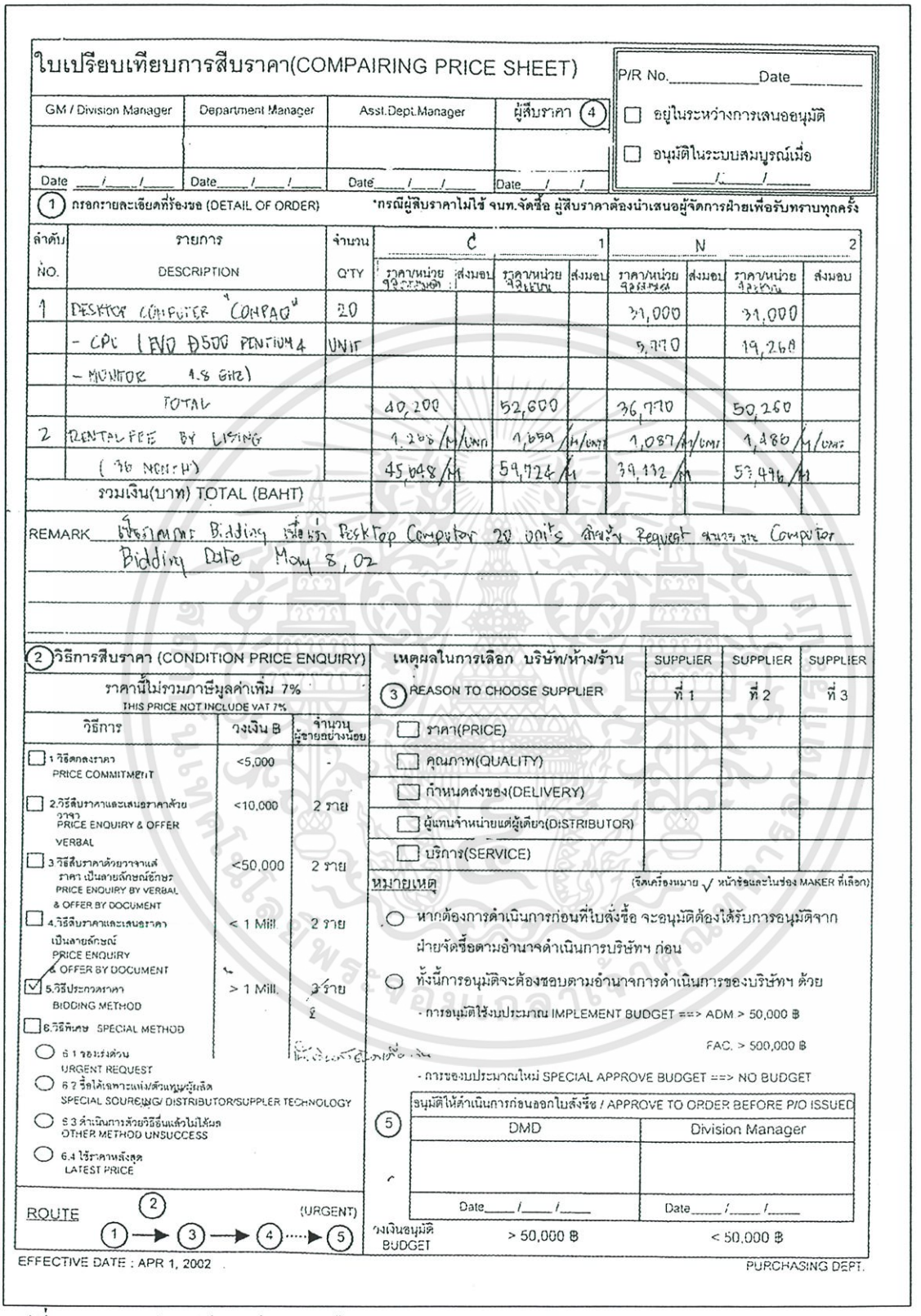
เหตุผลในการเลือก บริษัท/ห้าง/ร้าน REASON TO CHOOSE SUPPLIER	1	2	3
☑ ราคา(PRICE)	✓		
☐ คุณภาพ(QUALITY)			
☐ กำหนดส่งของ(DELIVERY)			
☐ ผู้แทนจำหน่ายแต่ผู้เดียว(DISTRIBUTOR)			
☐ บริการ(SERVICE)			

REMARK ราคาที่ได้ของ IBM เป็นราคาที่ต่ำที่สุดถึงแม้ว่าจะมีผู้ขายรายอื่นเสนอราคาต่ำกว่า แต่ไม่ได้ยี่ห้อของ IBM ซึ่งมีความจำเป็นต้องใช้ยี่ห้อ IBM เพื่อความเข้ากันได้กับของเดิม ที่มีอยู่ และราคาที่ต่ำกว่าเดิมมาก คือ 156,200 - = 8.55%

ผู้เลือก/อนุมัติซื้อ (CHOOSE BY/PURCHASING APPROVE)				* ผู้สืบราคา
Division Manager	Assi.Division Manager	Dept. Manager	Assi.Dept	ENQUIRY BY

ส่วนจัดหา 01/09/2001

*กรณีผู้สืบราคาไม่ใช่ จนท จัดซื้อ ผู้สืบราคาต้องนำเสนอผู้จัดการฝ่ายเพื่อรับทราบทุกครั้ง



ใบเปรียบเทียบการสืบราคา(COMPAIRING PRICE SHEET)

P/R No._____Date_____

☐ อยู่ในระหว่างการเสนออนุมัติ

☐ อนุมัติในระบบสมบูรณ์เมื่อ ___/___/___

GM / Division Manager	Department Manager	Asst.Dept.Manager	ผู้สืบราคา (4)
Date ___/___/___	Date ___/___/___	Date ___/___/___	Date ___/___/___

(1) กรอกรายละเอียดที่ร้องขอ (DETAIL OF ORDER) *กรณีผู้สืบราคาไม่ใช้ จนท.จัดซื้อ ผู้สืบราคาต้องนำเสนอผู้จัดการฝ่ายเพื่อรับทราบทุกครั้ง

ลำดับ NO.	รายการ DESCRIPTION	จำนวน Q'TY	C ราคา/หน่วย	C ส่งมอบ	1 ราคา/หน่วย	1 ส่งมอบ	N ราคา/หน่วย	N ส่งมอบ	2 ราคา/หน่วย	2 ส่งมอบ
1	DESKTOP COMPUTER "COMPAQ"	20					31,000		31,000	
	- CPU INTEL P500 PENTIUM 4	UNIT					5,770		19,260	
	- MONITOR 15"									
	TOTAL		40,200		52,600		36,770		50,260	
2	RENTAL FEE BY LISING		1,268 /M/UNIT		1,659 /M/UNIT		1,087 /M/UNIT		1,180 /M/UNIT	
	(36 MONTH)		45,648 /M		59,724 /M		39,132 /M		53,496 /M	
	รวมเงิน(บาท) TOTAL (BAHT)									

REMARK เปิดราคาโดย Bidding เพื่อหา Desktop Computer 20 units ตามใบ Request ขอมาจาก Computer Bidding Date May 8, 02

(2) วิธีการสืบราคา (CONDITION PRICE ENQUIRY)

ราคานี้ไม่รวมภาษีมูลค่าเพิ่ม 7%
THIS PRICE NOT INCLUDE VAT 7%

วิธีการ	วงเงิน ฿	จำนวนผู้ขายอย่างน้อย
☐ 1 วิธีตกลงราคา PRICE COMMITMENT	<5,000	-
☐ 2.วิธีสืบราคาและเสนอราคาด้วยวาจา PRICE ENQUIRY & OFFER VERBAL	<10,000	2 ราย
☐ 3 วิธีสืบราคาด้วยวาจาและราคาเป็นลายลักษณ์อักษร PRICE ENQUIRY BY VERBAL & OFFER BY DOCUMENT	<50,000	2 ราย
☐ 4.วิธีสืบราคาและเสนอราคาเป็นลายลักษณ์ PRICE ENQUIRY & OFFER BY DOCUMENT	< 1 Mill.	2 ราย
☑ 5.วิธีประกวดราคา BIDDING METHOD	> 1 Mill.	3 ราย
☐ 6.วิธีพิเศษ SPECIAL METHOD		

- ○ 6.1 รวมร่วม URGENT REQUEST
- ○ 6.2 ซื้อได้เฉพาะแห่ง/ตัวแทนผู้ผลิต SPECIAL SOURCING/ DISTRIBUTOR'SUPPLER TECHNOLOGY
- ○ 6.3 ดำเนินการด้วยวิธีอื่นแล้วไม่ได้ผล OTHER METHOD UNSUCCESS
- ○ 6.4 ใช้ราคาครั้งสุดท้าย LATEST PRICE

ROUTE (2) → (1) → (3) → (4) ⋯▶ (5) (URGENT)

เหตุผลในการเลือก บริษัท/ห้าง/ร้าน (3) REASON TO CHOOSE SUPPLIER	SUPPLIER ที่ 1	SUPPLIER ที่ 2	SUPPLIER ที่ 3
☐ ราคา(PRICE)			
☐ คุณภาพ(QUALITY)			
☐ กำหนดส่งของ(DELIVERY)			
☐ ผู้แทนจำหน่ายแต่ผู้เดียว(DISTRIBUTOR)			
☐ บริการ(SERVICE)			

หมายเหตุ (ขีดเครื่องหมาย √ หน้าข้อและในช่อง MAKER ที่เลือก)

○ หากต้องการดำเนินการก่อนที่ใบสั่งซื้อ จะอนุมัติต้องได้รับการอนุมัติจากฝ่ายจัดซื้อตามอำนาจดำเนินการบริษัทฯ ก่อน

○ ทั้งนี้การอนุมัติจะต้องชอบตามอำนาจการดำเนินการของบริษัทฯ ด้วย

- การอนุมัติใช้งบประมาณ IMPLEMENT BUDGET ==> ADM > 50,000 ฿
  FAC. > 500,000 ฿
- การของบประมาณใหม่ SPECIAL APPROVE BUDGET ==> NO BUDGET

(5) อนุมัติให้ดำเนินการก่อนออกใบสั่งซื้อ / APPROVE TO ORDER BEFORE P/O ISSUED

DMD	Division Manager
Date ___/___/___	Date ___/___/___

วงเงินอนุมัติ BUDGET > 50,000 ฿ | < 50,000 ฿

EFFECTIVE DATE : APR 1, 2002 PURCHASING DEPT.

รูปที่ 6.60 แสดงใบเปรียบเทียบการสืบราคา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริษัท นิ เทเลคอม จำกัด NI TELECOM CO., LTD.

QUOTATION - ใบเสนอราคา

TO : _____ DATE : _____

KIND ATTN _____ FROM.: _____

CC : _____ NO. : _____

WE ARE PLEASED TO SUBMIT YOU THE FOLLOW ING QUOTATION AND OFFER TO SELL THE GOODS BY OUR CONDITION INDICATED

ITEM	DESCRIPTION	Q'TY	UNIT PRICE	AMOUNT
1	COMPAQ DESKPRO EVO D500 PENTIUM4 1.8 GHZ	20	31,000.00	620,000.00
	(SMALL FORM FACTOR MODEL) - TWO TONE COLOR			
	-INTEL PENTIUM4 1.8 GHZ			
	-256 KB INTEGRATED L2 CACHE			
	-133 MHZ FRONT SIDE BUS/INTEL 845 CHIPSET			
	-40 GB (7200 RPM) SMART III ULTRA ATA/100			
	-128 MB SDRAM (PC133) EXPANDABLE 3 GB			
	-1.44 MB DISKETTE DRIVE			
	-48X CD-ROM DRIVE			
	-INTEGRATED AUDIO WITH INTERNAL SPEAKER			
	-INTEGRATED 10/100 MBPS			
	-NVDIA VANTA 16MB 4X AGP			
	-MOUSE & KEYBOARD PS/2			
	-MS WINDOWS 98 THAI SE			
	-COMPAQ 15" COLOR MONITOR 2 TONE S550	20	5,770.00	115,400.00
2	FOR RENT / UNIT / PER MONTH	20	1,087.00	21,740.00

ALL PRICE ARE EXCLUSIVE VAT 7 % ราคายังไม่รวมภาษีมูลค่าเพิ่ม 7 %

TERMS AND CONDITION

VALIDITY : 3 DAYS AFTER QUOTATION / บริษัท ฯ ขอสงวนสิทธิ์ในการเปลี่ยนแปลงราคาโดยไม่ต้องแจ้งล่วงหน้า

PAYMENT : 30 DAYS AFTER INVOICE DATE

DELIVERY : 7-10 DAYS AFTER RECEIVE YOUR P/O

WARRANTY : 3-YEARS ON SITE SERVICE

ระยะเวลาในการเช่า ขั้นต่ำ 36 เดือน

SUBMIT BY _____ APPROVED BY: _____

( ) ( )

SALE DEPT. MANAGING DIRECTOR

รูปที่ 6.61 แสดงใบเสนอราคาของบริษัท นิ เทเลคอม จำกัด แบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริษัท นิ เทเลคอม จำกัด NI TELECOM CO., LTD.

QUOTATION - ใบเสนอราคา

TO : _____ DATE : _____

KIND ATTN _____ FROM.: _____

CC : _____ NO. : _____

WE ARE PLEASED TO SUBMIT YOU THE FOLLOW ING QUOTATION AND OFFER TO SELL THE GOODS BY OUR CONDITION INDICATED

ITEM	DESCRIPTION	QTY	UNIT PRICE	AMOUNT
1	COMPAQ DESKPRO EVO D500 PENTIUM4 1.8 GHZ (SMALL FORM FACTOR MODEL) - TWO TONE COLOR -INTEL PENTIUM4 1.8 GHZ -256 KB INTEGRATED L2 CACHE -133 MHZ FRONT SIDE BUS/INTEL 845 CHIPSET -40 GB (7200 RPM) SMART III ULTRA ATA/100 -128 MB SDRAM (PC133) EXPANDABLE 3 GB -1.44 MB DISKETTE DRIVE -48X CD-ROM DRIVE -INTEGRATED AUDIO WITH INTERNAL SPEAKER -INTEGRATED 10/100 MBPS -NVDIA VANTA 16MB 4X AGP -MOUSE & KEYBOARD PS/2 -MS WINDOWS 98 THAI SE	20	31,000.00	620,000.00
	-COMPAQ 15" TFT COLOR MONITOR 2 TONE TFT5015	20	19,260.00	385,200.00
2	FOR RENT / UNIT / PER MONTH	20	1,486.00	29,720.00

ALL PRICE ARE EXCLUSIVE VAT 7 % ราคายังไม่รวมภาษีมูลค่าเพิ่ม 7 %

TERMS AND CONDITION

VALIDITY : 3 DAYS AFTER QUOTATION / บริษัท ฯ ขอสงวนสิทธิ์ในการเปลี่ยนแปลงราคาโดยไม่ต้องแจ้งล่วงหน้า

PAYMENT : 30 DAYS AFTER INVOICE DATE

DELIVERY : 7-10 DAYS AFTER RECEIVE YOUR P/O

WARRANTY : 3 YEARS ON SITE SERVICE

ระยะเวลาในการเช่า ขั้นต่ำ 36 เดือน

SUBMIT BY _____ APPROVED BY: _____

( ) ( )

SALE DEPT. MANAGING DIRECTOR

รูปที่ 6.62 แสดงใบเสนอราคาของบริษัท นิ เทเลคอม จำกัด แบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CD
CD systems Ltd.

QUOTATION

Page 1 of 1

Attention :	Ref No.
Company :	Date :
Tel :	Attachment :
Fax :	Refer :

We are pleased to submit you the following quotation and after to sell the products or service here in at price terms and terms stated

No	Part No.	Description	Qty.	Net Price	Amount
		PC's Workstation : Alternative 1			
1	470033-067	**Compaq Deskpro EVO D500 Pentium4 1.8-GHz (Small Form Factor Model) - Two Tone Color** - Intel Pentium4 1.8-GHz - 256KB Integrated ECC - 133MHz Front Side Bus/ Intel 845 Chipset - 40GB (7200RMP) QSI SMART III Ultra ATA/100 - 128MB SDRAM (PC133) expandable 3GB - 1.44MB Diskette Drive - 48X CD-ROM Drive - Integrated AC97 Audio with Internal Speaker - Integrated Intel PRO/100 VM Network Card - NVDIA Vanta 16MB 4X AGP **- 15" TFT Color Monitor - Two Tone Color** - Mouse & Keyboard PS/2 - Mouse pad - MS Windows 98 Thai SE Warranty : 3 Years On-Site Service in Bangkok aera, 50 km from Compaq Center. ***** This price is exclusive VAT7%.**	20	52,600	1,052,000.00

Price Validity : 7 Days

Delivery Terms : 4 Weeks

CD
Project Sales Team
Department

We look forward to give you our best service

Proposed By

Senior Account Executive Manufacturing & Retail

For More Information Please Contact

รูปที่ 6.63 แสดงใบเสนอราคาของบริษัท CD SYSTEMS จำกัด แบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CD systems Ltd.

QUOTATION

Attention :	Ref No.
Company :	Date :
Tel :	Attachment :
Fax :	Refer :

We are pleased to submit you the following quotation and after to sell the products or service here in at price terms and terms stated

No	Part No.	Description	Qty.	Net Price	Amount
		PC's Workstation : Alternative 2			
1	470033-067	**Compaq Deskpro EVO D500 Pentium4 1.8-GHz (Small Form Factor Model) - Two Tone Color** - intel Pentium4 1.8-GHz - 256KB Integrated ECC - 133MHz Front Side Bus/ Intel 845 Chipset - 40GB (7200RMP) QSI SMART III Ultra ATA/100 - 128MB SDRAM (PC133) expandable 3GB - 1.44MB Diskette Drive - 48X CD-ROM Drive - Integrated AC97 Audio with Internal Speaker - Integrated Intel PRO/100 VM Network Card - NVDIA Vanta 16MB 4X AGP - **Compaq 15" Color Monitor (Two Tone Monitor)** - Mouse & Keyboard PS/2 - Mouse pad - MS Windows 98 Thai SE Warranty : 3 Years On-Site Service in Bangkok aera. 50 km from Compaq Center.	20	40,200	804,000.00
		*** This price is exclusive VAT7%.			

Price Validity : 7 Days

Delivery Terms : 4 Weeks

CD
Project Sales_Team
Department

We look forward to give you our best service

Proposed By

Senior Account Executive Manufacturing & Retail

For More Information Please Contact

รูปที่ 6.64 แสดงใบเสนอราคาของบริษัท CD SYSTEMS จำกัด แบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CD
CD systems Ltd.

QUOTATION

Attention :	Ref No.
Company :	Date :
Tel :	Attachment :
Fax :	Refer :

We are pleased to submit you the following quotation and after to sell the products or service here in at price terms and terms stated

No	Part No.	Description	Qty.	Net Price	Amount
3	470033-067	**Compaq Deskpro EVO D500 Pentium4 1.8-GHz (Small Form Factor Model) - Two Tone Color** - Intel Pentium4 1.8-GHz - 256KB Integrated ECC - 133MHz Front Side Bus/ Intel 845 Chipset - 40GB (7200RMP) QSI SMART III Ultra ATA/100 - 128MB SDRAM (PC133) expandable 3GB - 1.44MB Diskette Drive - 48X CD-ROM Drive - Integrated AC97 Audio with Internal Speaker - Integrated Intel PRO/100 VM Network Card - NVDIA Vanta 16MB 4X AGP - Compaq 15" Color Monitor (Two Tone Monitor) - Mouse & Keyboard PS/2 - MS Windows 98 Thai SE Warranty : 3 Years On-Site Service in Bangkok aera, 50 km from Compaq Center.	1	42,500	42,500.00
4	470033-067	**Compaq Deskpro EVO D500 Pentium4 1.8-GHz (Small Form Factor Model) - Two Tone Color** - Intel Pentium4 1.8-GHz - 256KB Integrated ECC - 133MHz Front Side Bus/ Intel 845 Chipset - 40GB (7200RMP) QSI SMART III Ultra ATA/100 - 128MB SDRAM (PC133) expandable 3GB - 1.44MB Diskette Drive - 48X CD-ROM Drive - Integrated AC97 Audio with Internal Speaker - Integrated Intel PRO/100 VM Network Card - NVDIA Vanta 16MB 4X AGP - Compaq 15" Color Monitor (Two Tone Monitor) - 15" TFT Color Monitor - Two Tone Color - Mouse & Keyboard PS/2 - MS Windows 98 Thai SE Warranty : 3 Years On-Site Service in Bangkok aera, 50 km from Compaq Center. ***** This price is exclusive VAT7% and estimated price.**	1	57,500	57,500.00

Price Validity : 7 Days

Delivery Terms : 4 Weeks

CD
Project Sales_Team
Department

We look forward to give you our best service

For More Information Please Contact

รูปที่ 6.65 แสดงใบเสนอราคาของบริษัท CD SYSTEMS จำกัด แบบที่ 2 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

QUOTATION

Compaq Financial Services
COMPAQ

Compaq Computer Corporation (Thailand) L
18th Floor, M. Thai Tower
All Seasons Place, 87 Wireless Road,
Lumpini, Patumwan, Bangkok 10330, Thailand

Sale Manager
Tel. Office : 0-2654-0788 ext. 3231
Fax 0-2654-0720-1

Date :
Attention :
cc :
Company :
Tel :
Fax :
Payment : Quaterly in Advanced
ease Term 36 Months
Validity : 30 days

Lease Type : FMV Lease

Reference No.

We are pleased to submit you the following described herein at price, items, and terms stated.

Unit : Bah

Items	Product Description	Product Cost / Unit	Qty.	Lease Amount	Quarterly Rental Payment
1	Compaq Deskpro EVO D500 P4 1.8 GHz (Small form) + 15" TFT monitor	52,600	20	1,052,000	99,572
2	Compaq Deskpro EVO D500 P4 1.8 GHz (Small form) + 15" monitor	40,200	20	804,000	76,099

~~Lease Term & Conditions~~

a) Rental payment is Quarterly in Advance
b) Rental is excluded VAT 7%
c) Actual offers is contingent upon Cpmpaq review and apporval of all Credit information supplied by the customer :
   1) Certificate of the company registration and company's profile
   2) List of the company shareholder
   3) Annual Report for 3-year (audited report with Note to Account)
d) At the end of Lease - Customer can choose to return, purchase or extend the lease term

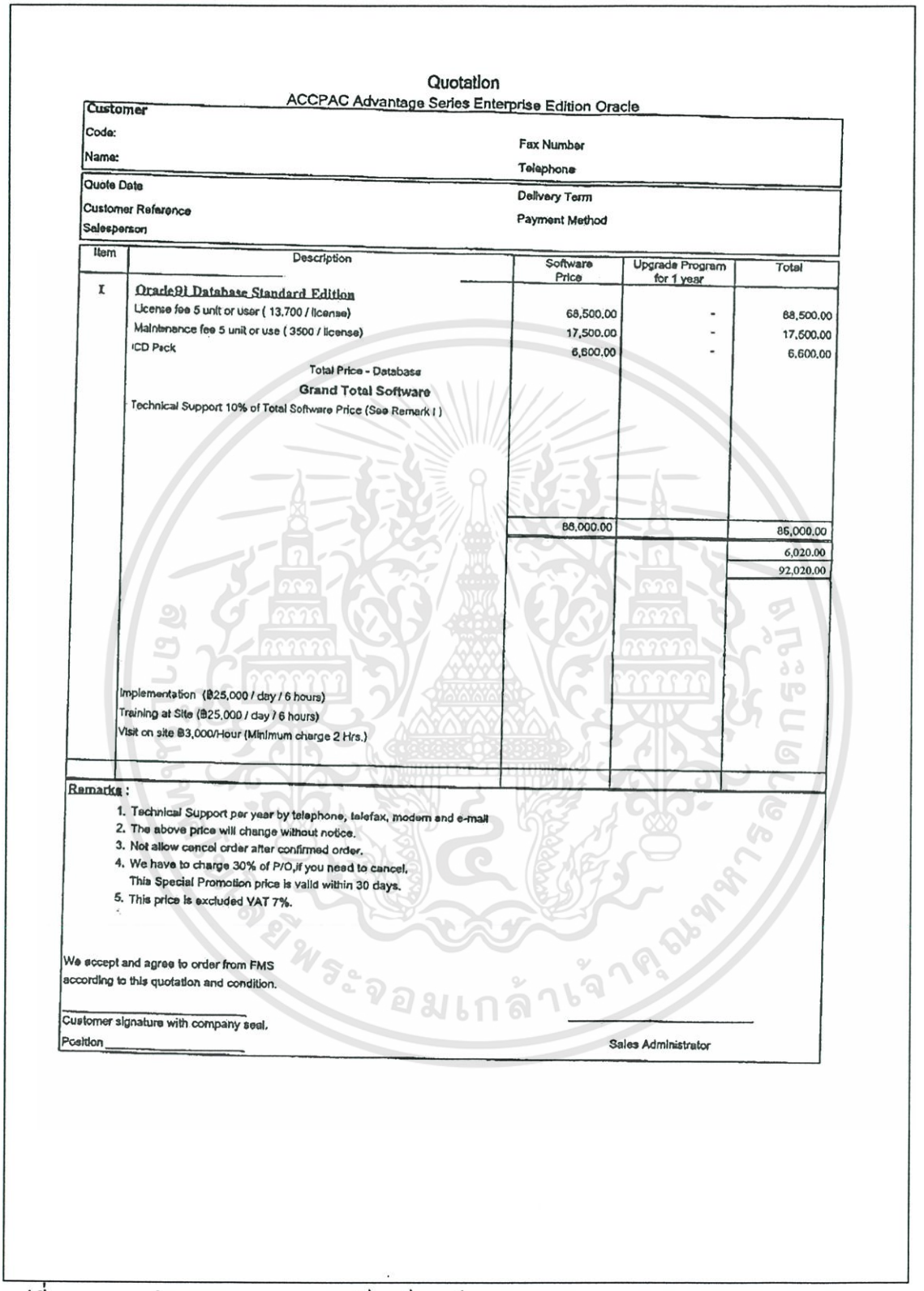
We hope this quotation is favorable to you and we look forward to your reply. Thank You for considering COMPAQ LEASIN as your IT business partner.

Warmest Regards

Sale Manager
Compaq Financial Services

รูปที่ 6.66 แสดงใบเสนอราคาของบริษัท คอมแพ็คคอมพิวเตอร์ (ประเทศไทย) จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Quotation

ACCPAC Advantage Series Enterprise Edition Oracle

Customer	
Code:	
Name:	Fax Number
	Telephone
Quote Date	Delivery Term
Customer Reference	Payment Method
Salesperson	

Item	Description	Software Price	Upgrade Program for 1 year	Total
I	**Oracle9i Database Standard Edition**			
	License fee 5 unit or user (13,700 / license)	68,500.00	-	68,500.00
	Maintenance fee 5 unit or use (3500 / license)	17,500.00	-	17,500.00
	CD Pack	6,600.00	-	6,600.00
	Total Price - Database			
	Grand Total Software			
	Technical Support 10% of Total Software Price (See Remark I)			
		86,000.00		86,000.00
				6,020.00
				92,020.00
	Implementation (฿25,000 / day / 6 hours)			
	Training at Site (฿25,000 / day / 6 hours)			
	Visit on site ฿3,000/Hour (Minimum charge 2 Hrs.)			

Remarks :

1. Technical Support per year by telephone, telefax, modem and e-mail
2. The above price will change without notice.
3. Not allow cancel order after confirmed order.
4. We have to charge 30% of P/O,if you need to cancel.
   This Special Promotion price is valid within 30 days.
5. This price is excluded VAT 7%.

We accept and agree to order from FMS
according to this quotation and condition.

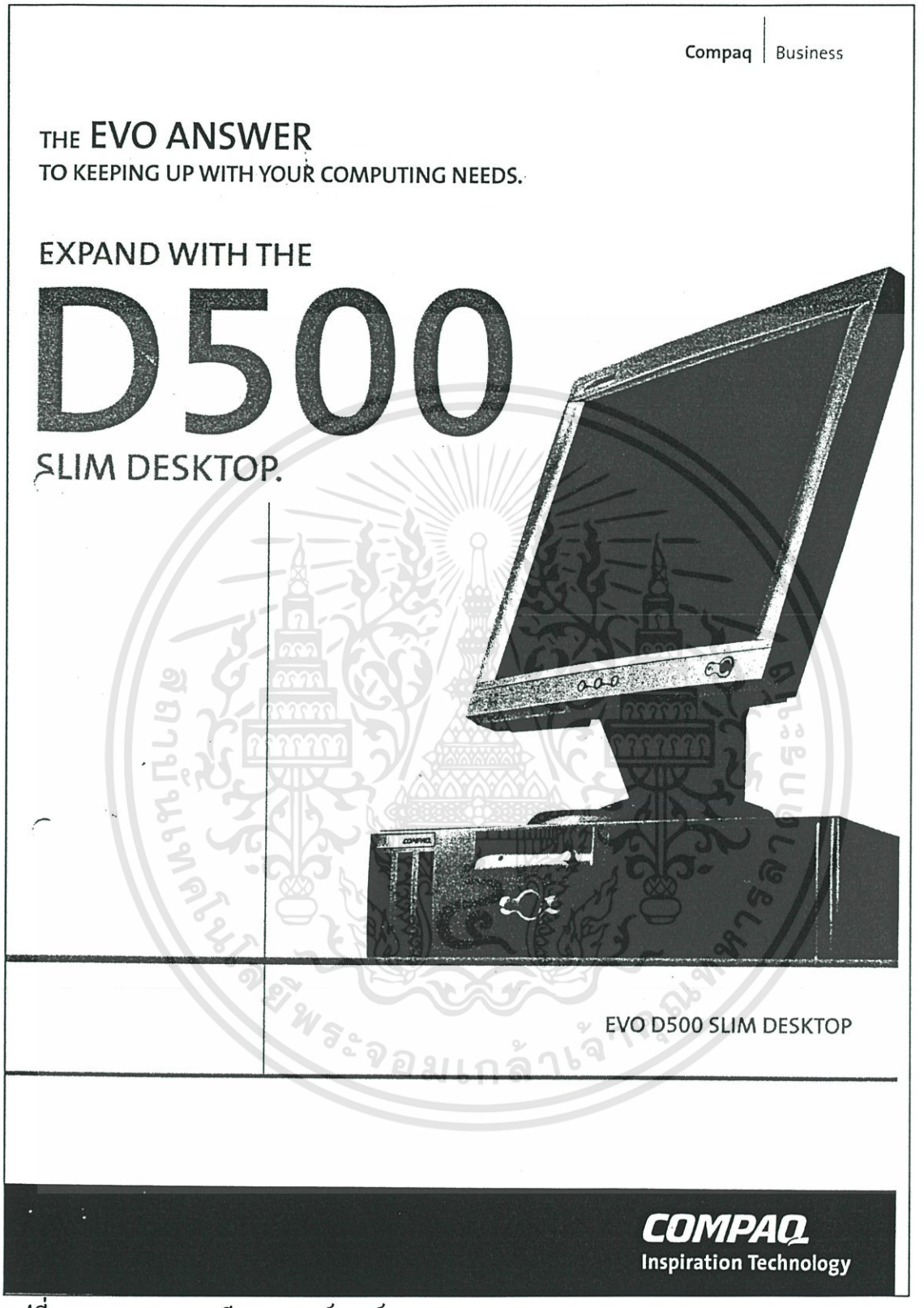
Customer signature with company seal.

Position

Sales Administrator

รูปที่ 6.67 แสดงใบเสนอราคาของซอฟต์แวร์ฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.68 แสดงรายละเอียดของฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Compaq | Business

THE EVO D500 SLIM DESKTOP offers a host of expansion option- including 1 AGP slot, 3 PCI slots and 5 drive bays

EVO D500 DESKTOP SERIES

CHIPSET	Intel® 845
PROCESSOR	Intel® Pentium® 4 – 1.5GHz to 2.0GHz
STANDARD MEMORY	128 or 256MB SDRAM (PC133)
MAXIMUM MEMORY	3GB SDRAM (PC133)
HARD DRIVE	7200 rpm: 20GB SMART III Ultra ATA/100
KEYBOARD/MOUSE	Easy Access Keyboard and 2 Button Scroll Mouse
NIC	Integrated Intel PRO/100 VM Network Connection
MODEM	V.90 56K PCI Modem (Optional)
REMOVABLE STORAGE	1.44MB Diskette Drive
GRAPHICS	NVIDIA Vanta 16MB 4X AGP
OPTICAL STORAGE	48X CD-ROM Drive (Optional)
AUDIO	Integrated Audio with internal speaker
SOFTWARE	Microsoft Windows 2000 SP2, Windows 98SE, Windows NT 4.0 SP6a, Windows XP Home* or Windows XP Professional* (*when available)
MANAGEABILITY	Insight Manager LC and Altiris eXpress
SLOTS	1 AGP, 3PCI
BAYS	5 bays
DIMENSIONS	H5.72" X W15.25" X D17.9" / H14.5cm x W38.7cm x D45.5cm
AUDIO/ USB PORTS	2 front audio and 2 front USB ports (optional)
SERVICEABILITY FEATURES	Tool-less drive removal, Tool-less hood removal, Easy access to motherboard
WARRANTY	3-year parts, labour and on-site service

Each and every Evo solution combines innovation with flexibility to lower the total cost of ownership. For users, that means having access to all the latest technology, while to IT managers it means reduced headaches and lower IT costs.

The sleek industrial designs of the Evo range are based on current trends in consumer electronics while fitting in perfectly with office interiors around the world.

In fact, be it in finance, IT or management, every corporate manager will be pleased with this new evolution in business computing.

RANGE OF SUPERIOR FEATURES

- **SUPERFAST PERFORMANCE** The new Intel 845 chipset offers four times faster front side bus speeds and Pentium 4 processors that range from 1.5GHz to 2.0GHz
- **UPGRADEABLE MEMORY** To support increasingly high-performance business application, each Evo model has 128 or 256MB of SDRAM PC133 pre-installed. Upgradeable to 3GB.
- **ADVANCED GRAPHICS** Combining high performance with low cost, the twin options of NVIDIA and Matrox 4X AGP graphics controllers are easy to install and easy to use.
- **FULL MULTIMEDIA CAPABILITY** The premium internal speaker and integrated CD-ROM drive offer a practical approach to today's multimedia-heavy business environment.
- **SMART STORAGE OPTIONS** The SMART (Self-Monitoring Analysis and Reporting Technology) III Ultra ATA/100 hard drives incorporate Drive Protection System technology which warns users of impending faults – giving them the opportunity to back up vital data.
- **EXPANSION** An optional I/O module offers 2 audio and 2 USB ports for convenience and ease of use.

THE COMPAQ ADVANTAGE

- **PRODUCT PREDICTABILITY** Compaq Evo solutions are easy to deploy and manage since they have a single software image and are available in globally defined models.
- **EXTENDED MANAGEABILITY** All Compaq Evo solutions come with industry-standard software such as Compaq Intelligent Manageability, Altiris eXpress and PC Transplant.
- **SERVICE AND SUPPORT** Compaq and its partners provide more than 100,000 sales and authorized service locations worldwide. Other unique Compaq Support Services include CarePaq, Intelligent Manageability and PC Lifecycle Soluti

OPTIONS FOR EVO DESKTOPS

GRAPHICS	GeForce2 16MB MX200, GeForce2 MX AGP32 Sngl ALL, Matrox G550 DH-VGA ALL, DVI Cable ALL, G200MMS Quad A/D DVI ALL, G200MMS A/D ANALOG ALL, G200MMS DVI CABLE ALL
MEMORY	64, 128, 256MB SDRAM 133MHz DIMM Memory
MONITOR	Professional Series: P1210 22", P920 19" (2-Tone color) Value Series: V720 17", V570 15" (2-Tone color) Standard Series: S920 19", S720 17" (2-Tone color)
OPTICAL DRIVES	CD-ROM Drive, DVD-ROM Drive, DVD-RAM Drive, DVD-RW Drive, CD-RW Drive, Combo Drive
REMOVABLE STORAGE	Floppy Drive, Tape Drive

Compaq PCs use genuine Microsoft® Windows®
www.microsoft.com/piracy/howtotell

www.compaq.co.th
Compaq Call Center 0-2654-0808

©2001 Compaq Computer Corporation. Compaq and the Compaq logo are registered in the U.S. Patent and Trademark Office. Inspiration Technology is a trademark of Compaq Information Technologies Group, L.P. in the U.S. and other countries. Intel, the Intel inside logo and the Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Microsoft and Windows are either registered trademarks of their respective companies. The information contained in this brochure is subject to change without prior notice. Compaq shall not be liable for errors or omissions contained herein.

COMPAQ
Inspiration Technology

รูปที่ 6.69 แสดงรายละเอียดของฮาร์ดแวร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.81 แสดงการเปรียบเทียบราคาของผู้เสนอขายสินค้า

บริษัท

โทร แฟกซ์

WE ARE PLEASE TO SUBMIT YOU THE FOLLOWING DECRIPTIONS AND THE PRICES AS FOLLOWS:

PRODUCT	DECRIPTION	QTY	PER UNIT BAHT	AMOUNT BAHT	WARRANTY PERIOD	PER UNIT BAHT	AMOUNT BAHT	PER UNIT BAHT	AMOUNT BAHT	เจราล่ร่องกับบริษัท		ผลการเจราล่ร่อง PER UNIT BAHT	ผลการเจราล่ร่อง AMOUNT BAHT	%ที่ได้รับจากการล่ร่อง
	DATABASE SERVER													
	IBM E SERVER Xseries 350 (RAC 4U)	2	320,000.00	640,000.00	TYPE 2	361,500.00	725,000.00	355,000.00	710,000.00	300,000.00	600,000.00	300,000.00	600,000.00	6.25%
86825RY	IBM E SERVER Xseries 350 PENTIUM IIIXeon 700 MHZ													
	PENTIUM IIIXeon 700MHz/133MHz													
	4-waSMP Support													
	cache Memory 2MB													
	Chipkill Memory													
	1 x 1.44MB floppy disk drive													
	internal Storge 218.2GB													
	Expansion Slots 6 (Active PCI)													
	Expansion Bays 8,6 Hot plug													
	Network Integrated 10/10Ethernet													
	48X-20X CD-rom													
	Integrated System Mgmt Processor													
	Six Fans (four for disks,memory,PCI;two for processors)													
	Power supply 270W													
	Hot-plug Components:PCI,Power,Fans,HDDs													
	Light path Disgnostics													
	Light path Disgnostics													
	Predictive Failure Analysis support HDDs,processor,VRMs													
	fans powersupplies and memory													
00N7944	Netfinity 700 MHz/2MB Pentium III Xeon Processor	2	157,000.00	314,000.00		153,900.00	307,800.00	164,000.00	328,000.00	150,900.00	301,800.00	147,000.00	29,400.00	6.37%
06P3601	10/100 Ethermet Server Adapter	4	3,200.00	12,800.00		3,600.00	14,400.00	5,500.00	22,000.00	3,000.00	12,000.00	3,100.00	12,400.00	3.13%
33L3115	Netfinity 256MB 100 MHz ECC SDRAM RDIMM	16	4,000.00	64,000.00		5,000.00	80,000.00	6,300.00	100,800.00	4,000.00	64,000.00	4,200.00	67,200.00	5%
35311RX	EXP30Storage Expansion Unit	1	100,000.00	100,000.00		69,500.00	69,500.00	124,000.00	124,000.00	679,000.00	67,900.00	65,000.00	65,000.00	3.50%
	(std) SCSI PCI Card to Backplane Cable	1												
	fans powersupplies and memory													
00N7944	Netfinity 700 MHz/2MB Pentium III Xeon Processor	2	157,000.00	314,000.00		153,900.00	307,800.00	164,000.00	328,000.00	150,900.00	301,800.00	147,000.00	29,400.00	6.37%
06P3601	10/100 Ethermet Server Adapter	4	3,200.00	12,800.00		3,600.00	14,400.00	5,500.00	22,000.00	3,000.00	12,000.00	3,100.00	12,400.00	3.13%
33L3115	Netfinity 256MB 100 MHz ECC SDRAM RDIMM	16	4,000.00	64,000.00		5,000.00	80,000.00	6,300.00	100,800.00	4,000.00	64,000.00	4,200.00	67,200.00	5%
35311RX	EXP30Storage Expansion Unit	1	100,000.00	100,000.00		69,500.00	69,500.00	124,000.00	124,000.00	679,000.00	67,900.00	65,000.00	65,000.00	3.50%
	(std) SCSI PCI Card to Backplane Cable	1												
37L6380	IBM Netfinity 270W Hot-Swap Redundant Power Supply	4	6,700.00	26,800.00		8,500.00	34,000.00	9,000.00	36,000.00	6,500.00	26,000.00	6,000.00	24,000.00	10.45%
06P5736	SERVERAID 4Mx ULTRA 160 SCSI CONTROLLER	2	26,300.00	52,600.00		30,000.00	6,000.00	29,500.00	59,000.00	26,000.00	52,000.00	25,500.00	51,000.00	3.04%
37L7205	(External)18.2GB 10K-4 Ultra 160 SCSI Hot-Dwap SL HOD	6	22,000.00	132,000.00		19,500.00	117,000.00	2,500.00	15,000.00	19,000.00	114,000.00	18,000.00	108,000.00	18.18%
37L7206	(External)36.4GB 10K-4 Ultra 160 SCSI Hot-Dwap SL HOD	5	30,600.00	153,000.00		27,000.00	135,000.00	33,900.00	169,500.00	26,500.00	132,500.00	26,000.00	130,000.00	15.03%
03K9310	NF 2m External SCSI Cable	1	3,500.00	3,500.00		4,500.00	4,500.00	5,900.00	5,900.00	3,500.00	3,500.00	3,100.00	3,100.00	15.03%
28L3644	Space Sver II Keyboard	1	7,500.00	7,500.00		2,900.00	2,900.00	10,000.00	10,000.00	2,900.00	2,900.00	2,900.00	2,900.00	11.43%
28L3673	Sleek 2 Button Stealth black Mouse	1	1,200.00	1,200.00		400.00	400.00	3,400.00	3,400.00	400.00	1,400.00	1,000.00	1,000.00	61.33%
	LTO 3580-L11(1 DATA, 1Cleanning)	1	240,000.00	240,000.00		250,900.00	250,900.00	269,000.00	269,000.00	230,000.00	230,000.00	235,000.00	235,000.00	16.67%
28T0320	3 yr. On-site 24x7 4hr.Reponse time	2	13,500.00	27,000.00		14,200.00	28,400.00	1,400.00	28,000.00	13,000.00	26,000.00	13,500.00	27,000.00	2.08%
	RACK													
AM-9627S	19" Open Rack 27U Germany Export	1	11,000.00	11,000.00		4,500.00	4,500.00	1,400.00	14,000.00	4,500.00	4,500.00	9,000.00	9,000.00	18.18%
AM-9953	AC-Power Distribution 12 Outlets	1	2,000.00	2,000.00		1,600.00	1,600.00	3,300.00	3,300.00	1,500.00	1,500.00	1,500.00	1,500.00	2.50%
AM-9957	Component Shelves Side 40 cm.	1	1,000.00	1,000.00		1,000.00	1,000.00	1,700.00	1,700.00	1,000.00	1,000.00	900.00	900.00	10%
AM-9962	Ventilating Fans 2x04"	1	1,500.00	1,500.00		1,400.00	1,400.00	2,200.00	2,200.00	1,400.00	1,400.00	1,200.00	1,200.00	20%
	Software													
	IBM Director,IBM ServerGuide,Norton Antivirus													
	(The above price not included VAT)													
	SUBTOTAL			1,789,900.00			1,838,300.00		2,036,800.00		1,641,400.00		1,633,200.00	8.75%
	Value added Tax		7.00%	125,293.00										
	GRAND TOTAL			1,915,193.00										

Remaks:

CU will not install and responsible for any illegal softwares

The price is subject to change without prior notics.

Delivery date : Within 30-45 days after the date of confirmation.

Payment within 30 days after delivery date

Warranty (Mon-fri/8:00-5:00 pm /Next day response)

Type 1 : 3 year parts & on-site service;1st year by IBM,2nd-3rd year by CU

Type 2 : 3 year parts & 3 year on-site by IBM

Type 3 : 1 year parts & on - site service by CU

Type 4 : 1 year parts & Carry-in

Type 5 : 1 year parts & 1 year on-site by IBM

Sales Executive

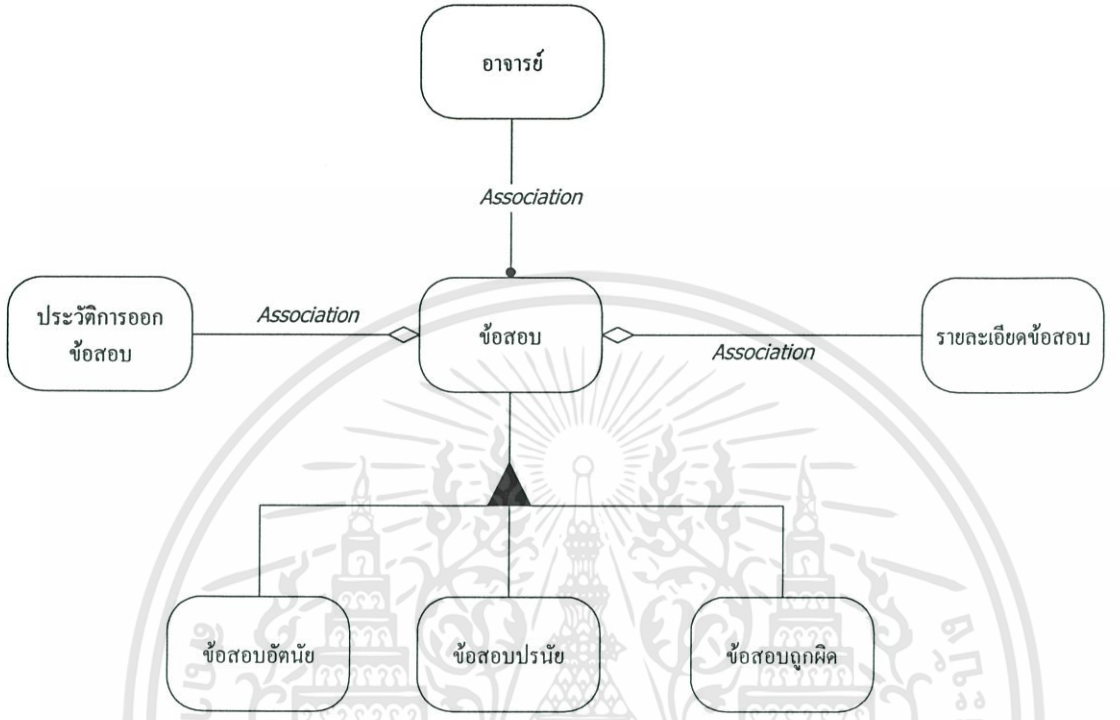
E-mail :

Confirmed By: ( )

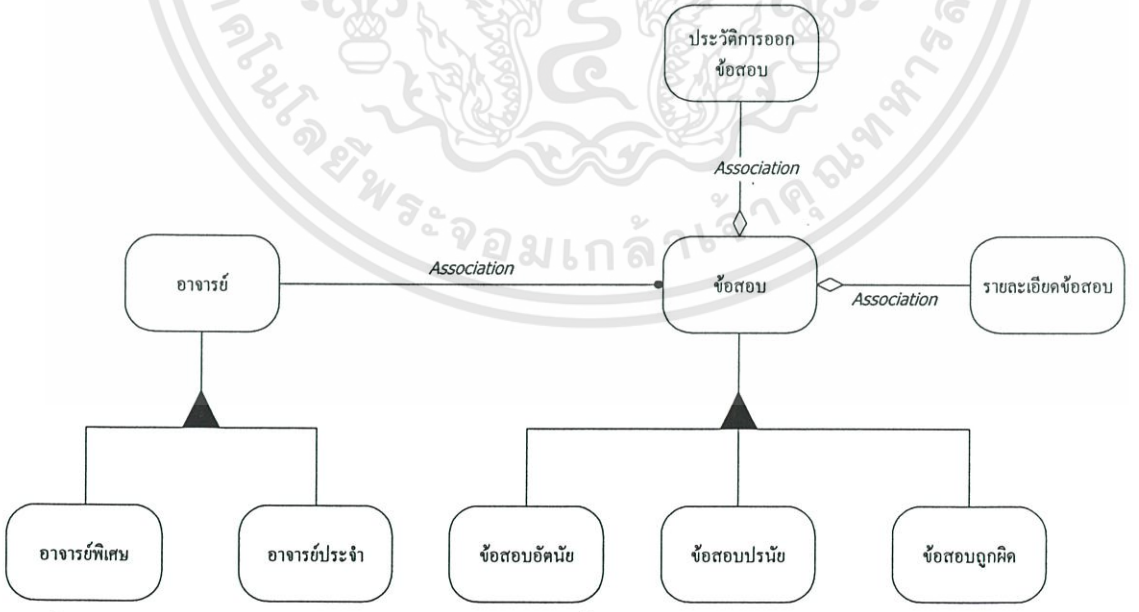
Date: / / 2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7 ระยะที่ 6



รูปที่ 6.70 แสดงการปรับปรุงไดอะแกรมของวัตถุครั้งที่ 1



รูปที่ 6.71 แสดงการปรับปรุงไดอะแกรมของวัตถุครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.82 แสดงรายละเอียดของฟังก์ชันเพิ่มข้อมูลของวัตถุข้อสอบ

Function Definition (Algorithm)	**FD02**
Function Name : CREATE ITEM	Function ID: 5
Class/Object Name : ข้อสอบ	Class/Object ID: 2
Designer :	Date Conducted: 17/2/43
Programmer : พิบูล	

Function Type

☐ Private	☑ Public	☐ Protect
☐ Overridden	☐ Friends	☐ Inheritance
☐ Polymorphism	☑ Return (value, reference)	
☐ Other :		

Algorithm Guide Line

```
IF  FOUND ( )
    SHOWMESSAGE
ELSE
    INSERT NEW REC
END
```

Collaborator	**Parameter**
สร้างข้อสอบหลายข้อของข้อสอบแต่ละประเภท	ส่งค่า T/F กลับไปยัง CLASS TEACHER เพื่อแจ้งการบันทึกข้อสอบ

Notes / Comment

ค่าที่ส่งกลับอาจเป็น DIGIT BOOLEAN หรืออื่นๆที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.83 แสดงรายละเอียดของฟังก์ชันปรับปรุงข้อมูลของวัตถุข้อสอบ

Function Definition (Algorithm)	**FD02**
Function Name : UPDATE ITEM	Function ID: 6
Class/Object Name : ข้อสอบ	Class/Object ID: 2
Designer :	Date Conducted: 19/2/43
Programmer : พิบูล	

Function Type

☐ Private	☑ Public	☐ Protect
☐ Overridden	☐ Friends	☐ Inheritance
☐ Polymorphism	☑ Return (value, reference)	
☐ Other :		

Algorithm Guide Line

```
IF FOUND ( )
    ปรับปรุงตามประเภทข้อสอบ (รหัสข้อสอบ,ประเภทข้อสอบ,ข้อคำถาม,...,ฯลฯ)
ELSE
    SHOW MESSAGE
ENDIF
```

Collaborator	**Parameter**
แก้ไขบันทึกข้อสอบรายข้อของแต่ละประเภท	ส่งค่ากลับเพื่อแจ้ง CLASS_TEACHER

Notes / Comment

ค่าที่ส่งกลับอาจเป็นชนิด BOOLEAN,Digit, Char หรืออื่น ๆ ที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.84 แสดงรายละเอียดของฟังก์ชันลบข้อมูลของวัตถุข้อสอบ

Function Definition (Algorithm)	**FD02**
Function Name : Delete Exam	Function ID: 8
Class/Object Name : ข้อสอบ	Class/Object ID: 2
Designer :	Date Conducted: 20/2/53
Programmer : พิบูล	

Function Type

☐ Private	☑ Public	☐ Protect
☐ Overridden	☐ Friends	☐ Inheritance
☐ Polymorphism	☑ Return (value, reference)	
☐ Other :		

Algorithm Guide Line

```
If found()
    Showmessage()
    if Delete
        Delete Record
    endif
endif
```

Collaborator	**Parameter**
ลบข้อสอบตามข้อแต่ละประเภทที่เลือก	รับค่า รหัสข้อสอบ, รหัสคณะ, รหัสวิชา
	ส่งค่ากลับ `return result`

Notes / Comment

แจ้งข้อความเตือนก่อนลบและแจ้งหลังจากลบเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.85 แสดงรายละเอียดของฟังก์ชันค้นหาข้อมูลของวัตถุข้อสอบ

Function Definition (Algorithm)	**FD02**
Function Name : Search Item	Function ID: 9
Class/Object Name : ข้อสอบ	Class/Object ID: 2
Designer :	Date Conducted: 21/2/43
Programmer : พิบูล	

Function Type

☐ Private	☑ Public	☐ Protect
☐ Overridden	☐ Friends	☐ Inheritance
☐ Polymorphism	☑ Return (value, reference)	
☐ Other :		

Algorithm Guide Line

```
If not eof()
    ค้นหาตามรหัสข้อสอบ, ประเภทข้อสอบ, รหัสคณะ, รหัสวิชา
else
    not found()
end if
```

Collaborator	**Parameter**
ค้นหาข้อสอบตามข้อ ประเภทและวิชา ของคณะ	รับค่า รหัสข้อสอบ, รหัสวิชา รหัสคณะ Return : รหัสข้อสอบ,ข้อคำถาม

Notes / Comment

แสดงรายละเอียดข้อสอบตามที่ต้องการทางหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.86 แสดงรายละเอียดของการวัดความซับซ้อนของวัตถุ

MetricsTest

MT01

Page ___ of ___

Class/Object ID: 1 **Class/Object Name:** ข้อสอบ **Test ID :** 1

Tester: สุณี **Date Designed :** 5/2/43 **Date Conducted :** 13/7/43

Results:

Object Test :-

☑ **Passed** ☐ **Redesign** ☐ **Fault**

Method Test :-

☐ **Passed** ☑ **Redesign** ☐ **Fault**

Objective:

เพื่อวัดระดับความซับซ้อนของข้อสอบ

Object Test Cases

Size of Object : 1 **(Less Than 1)**

Depth of the Inheritance Tree (DIT) : 1 **(7±2)**

Number Of Children (NOC) : 3

Coupling Between Object Classes (CBO) : 4

Method Test Cases

Lack of Cohesion in Methods (LCOM) : 3

Number of Operations Overridden by Subclass (NOO) : 3

Number of Operations Added by a Class (NOA) : –

Specialization Index (SI) : $\frac{3x1}{3} = 1$

Expected Results/Notes

Actual Results/Notes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.87 แสดงรายละเอียดการวัดความสัมพันธ์ระหว่างวัตถุ

MetricsTest

MT02

Class/Object ID: 1 **Class/Object Name:** ข้อสอบ **Test ID :** 1

Tester: สุณี **Date Designed :** 2/2/43 **Date Conducted :** 13/7/43

Results:

☐ **Passed** ☐ **Redesign** ☐ **Fault**

Diagram

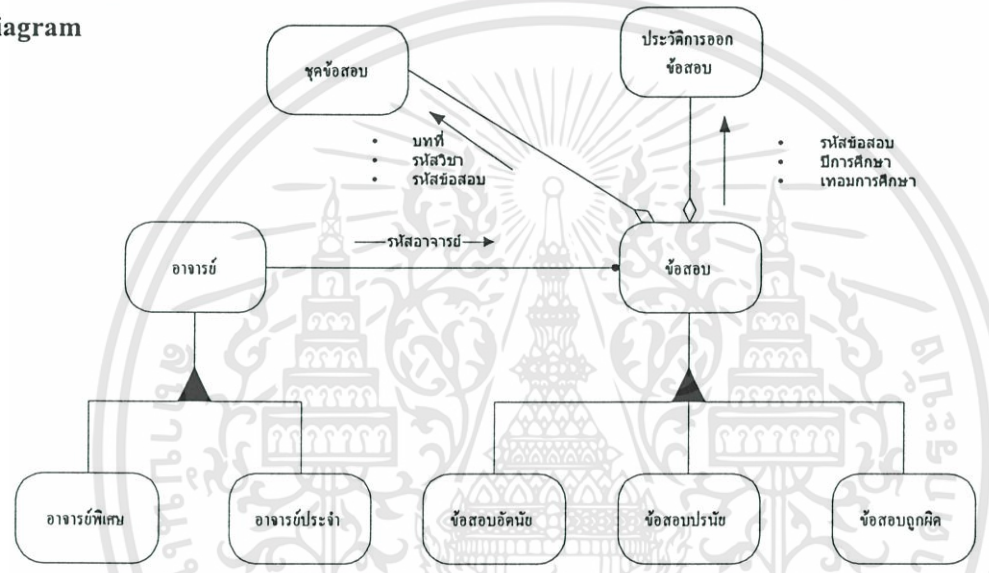


Diagram Test

Response for a Class (RFC) :- **(Standard Values** 10 **)**

Fan-in : 1 Fan-out : 6 Average : _____

Expected Results/Notes

Actual Results/Notes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.8 ระยะที่ 7

ตารางที่ 6.88 แสดงการวัดความซับซ้อนของฟังก์ชันภายในวัตถุ

Test Plan Page ___ of ___

Object Name: ข้อสอบ **Program ID:** 5 **Version Number:** 1

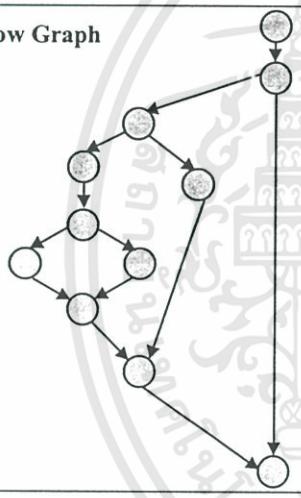
Tester: นิภาพร **Date Designed :** 2/2/43 **Date Conducted :** 13/7/43

Results: ☐ Passed ☐ Open Items

Test ID: 1 (TfrmCreateItem)

Objective:

Flow Graph



Complexity = e – n + 2

= 13 – 11 + 2

= 4

Script (interclass Metrics)

McCabe Metric

Expected Results/Notes

Actual Results/Notes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.89 แสดงรายละเอียดของการติดตั้งโปรแกรม

Installation Report

System Name: Item Bank

Software Version: 1

Officer: พีระ

Date Conducted : 7/2/44

User: ประกิจ

Step of Install

1. สร้าง Folder Item Bank
2. Install Item Bank System

Problem Description

"MFC 007" conflict (Windows 95)

Result / Notes

Copy file conftl.dll -> c://windows/System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.9 ระยะที่ 8

ตารางที่ 6.90 แสดงรายละเอียดของการตรวจสอบซอฟต์แวร์

Quality Control Report

System Name : Item Bank Page 1 of 1

Date Conducted : 12/2/44

Detail of Test

System Software	: ☑ Passed	☐ Conflict	
System Validation	: ☑ Passed	☐ Error	
Throughput	: ☑ Passed	☐ Error	
Application Software	: ☑ Passed	☐ Conflict	
Network System	: ☐ Passed	☐ Conflict	☑ No have

Result/Note

_____

_____

_____

_____

_____ _____

( ) ( )

System Analyst Project Manager

_____ _____

( ) ( )

System Programmer System User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.91 แสดงรายละเอียดของวิธีการติดตั้งระบบ

Prepare Conversion Plan
System Name : Item Bank Page ____ of ____ Date Conducted : 7/2/44 Officer : พีระ
Conversion Plan ☑ Abrupt Cut-Over ☐ Parallel ☐ Location ☐ Staged
Step of Conversion Plan เปลี่ยนจากงานมือ เป็นระบบคอมพิวเตอร์
Result/Note

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.92 แสดงรายละเอียดของการติดตั้งซอฟต์แวร์ฐานข้อมูล

Install Database Report

System Name : Item Bank

Page 1 of 1

Product Name : Personal Oracle 8

Date Conducted : 5/2/44

Officer (DBA) : วราภรณ์

Step of installation

1. **Installation**
   - ☑ Passed ☐ Conflict
2. **Data Test**
   - ☑ Passed ☐ Failed ☐ Other : _____
3. **Validation Test**
   - ☑ Passed ☐ Failed ☐ Other : _____
4. **Backup Test**
   - ☑ Passed ☐ Failed ☐ Other : _____
5. **Recovery Test**
   - ☑ Passed ☐ Failed ☐ Other : _____

Result/Note

_____

_____

_____

_____

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.10 ระยะที่ 9

ตารางที่ 6.93 แสดงรายละเอียดของปัญหาที่เกิดขึ้นของผู้ใช้

MEMO
Date: 9/3/44 From: คุณคำรงค์เกียรติ To: คุณวีระวัฒน์ ☑ Urgent ☐ Please Execute ASAP ☐ For your information (FYI)
Problem / Notes A001 : การบันทึกข้อสอบ – ไม่มีการตรวจสอบว่าข้อคำถามซ้ำกับที่เคยมีอยู่แล้วหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.11 ระยะที่ 10

ตารางที่ 6.94 แสดงรายละเอียดความต้องการการปรับปรุงระบบจากผู้ใช้

Improvement Report
Date: 15/2/45 From: ปิยะวรรณ To: วรางคณา ☐ Urgent ☑ Please Execute ASAP ☐ For your information (FYI)
Requirement เลือกข้อสอบแบบสุ่ม ของแต่ละประเภทข้อสอบในแต่ละวิชา
Notes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
บทสรุปและข้อเสนอแนะ

7.1 บทสรุป

การพัฒนาระบบสารสนเทศส่วนใหญ่มักนิยมใช้วิธีเชิงโครงสร้างภายใต้แบบจำลองต่าง ๆ ที่มีอยู่ทั่วไป ซึ่งแบบจำลองที่นำใช้การพัฒนานั้นมีเพียง 2-3 แบบจำลองเท่านั้น เช่น แบบจำลองน้ำตก แบบจำลองสไพริสและแบบจำลองน้ำพุ เป็นต้น แต่ละแบบจำลองนั้นมีข้อดีและข้อเสียแตกต่างกัน การนำแบบจำลองเหล่านี้มาใช้งานนั้น สิ่งที่ผู้วิเคราะห์ต้องคำนึงถึงเป็นสิ่งแรก คือ ความเหมาะสมของแบบจำลองกับโครงการที่ต้องการพัฒนา ดังนั้นแบบจำลองจึงมีส่วนจำเป็นและสำคัญกับการพัฒนาระบบสารสนเทศเป็นอย่างมาก

นอกจากแบบจำลองขั้นตอนสำหรับการพัฒนามีความสำคัญเช่นกัน เพราะขั้นตอนการพัฒนาของแต่ละแบบจำลองสามารถทำให้ระบบสารสนเทศนั้นสำเร็จตามวัตถุประสงค์ ขั้นตอนการดำเนินงานของแต่ละแบบจำลองมีความแตกต่างกัน บางแบบจำลองมีขั้นตอนมากบางแบบจำลองมีขั้นตอนการทำงานน้อย

จากบทที่ 2 กล่าวถึงวิธี เอส เอส เอ ดี เอ็ม ภายใต้แบบจำลองน้ำตก ซึ่งวิธีนี้ถูกคิดค้นระหว่างปี 1980 โดยรัฐบาลอังกฤษได้มอบหมายให้หน่วยงานของรัฐที่ชื่อว่า CCTA (Central Computing and Telecommunications Agency) พัฒนาขึ้นเพื่อเป็นแบบแผนสำหรับการพัฒนาระบบสารสนเทศให้กับองค์กรต่าง ๆ ของรัฐและเอกชน ภายในประเทศ เพื่อให้การพัฒนาระบบสารสนเทศเป็นไปในทิศทางเดียวกัน ซึ่งแบ่งขั้นตอนการพัฒนาออกเป็น โมดุล ระยะ ขั้นตอนและงาน ซึ่งผู้พัฒนาระบบสารสนเทศสามารถติดตามขั้นตอนการทำงานได้ตลอดการพัฒนาระบบสารสนเทศ และขั้นตอนการพัฒนาที่เกิดขึ้นจะอ้างอิงถึงกันเพื่อตรวจสอบความถูกต้องและความสมบูรณ์ของระบบจากผู้ใช้ทุกครั้ง ถ้าขั้นตอนแรกไม่สมบูรณ์หรือไม่ถูกต้องก็ไม่สามารถที่จะข้ามไปสู่ขั้นตอนถัดไปได้

เมื่อความต้องการของผู้ใช้เริ่มเปลี่ยนแปลงบ่อยครั้งประกอบกับเทคโนโลยีสำหรับการพัฒนาระบบสารสนเทศเริ่มมีวิวัฒนาการขึ้นทำให้วิธีเชิงโครงสร้างที่กล่าวมานั้นเริ่มประสบปัญหาสำหรับการพัฒนาระบบสารสนเทศเป็นอย่างมาก จึงมีนักวิจัยหลายท่านคิดวิธีการวิเคราะห์และออกแบบระบบสารสนเทศให้ เข้าใจง่าย รวดเร็วและสามารถนำกลับมาใช้ใหม่ได้ ซึ่งเรียกว่า “วิธีเชิงวัตถุ” วิธีเชิงวัตถุ คือ การวิเคราะห์และการออกแบบ ระบบเชิงวัตถุที่มีลักษณะเป็นไปตามความจริงของปัญหาเพื่อที่ทำให้การนำเสนอเป็นไปได้ง่ายและถูกต้อง วิธีเชิงวัตถุมีการพัฒนาหลายรูปแบบ เช่น วิธีของ Peter Coad / Edward Yourdon, James Rumbaugh หรือ Greedy Booch เป็นต้น แต่วิธีที่นิยม คือ วิธีของ James

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rumbaugh หรือเรียกว่า "Object Modeling Technique (OMT)" เพราะเป็นวิธีที่เข้าใจง่ายและเหมาะสมกับแอพพลิเคชั่นที่เป็นฐานข้อมูล การวิเคราะห์และออกแบบของวิธีเชิงวัตถุแบ่งออกเป็น 3 แบบจำลอง คือ แบบจำลองวัตถุ แบบจำลองไดนามิค และแบบจำลองฟังก์ชัน แบบจำลองทั้ง 3 มีหน้าที่และความรับผิดชอบที่ต่างกันออกไป แบบจำลองวัตถุ แสดงรายละเอียดของวัตถุ คลาส แอตทริบิวต์และความสัมพันธ์ในรูปของไดอะแกรม แบบจำลองไดนามิค แสดงพฤติกรรมของวัตถุและคลาสที่มีการเปลี่ยนแปลงเมื่อมีเหตุการณ์เข้ามากระทบโดยแสดงในรูปของ อีเวนท์เทรส สเตทไดอะแกรมและรายละเอียดของกิจกรรม ส่วนแบบจำลองฟังก์ชัน แสดงการทำงานของกระบวนการที่อยู่ภายในวัตถุและคลาสแต่ละตัว ซึ่งอยู่ในรูปของแผนภูมิการไหลของข้อมูล ทุกแบบจำลองแสดงรายละเอียดในรูปของเอกสารประกอบเพื่อให้สามารถนำไปแก้ไขในภายหลังได้

ส่วนการออกแบบ คือ ขั้นตอนที่เชื่อมต่อระหว่างการวิเคราะห์กับการออกแบบระบบ ซึ่งคอยตรวจสอบไดอะแกรมต่าง ๆ ที่เกิดพร้อมกับเพิ่มเติมรายละเอียดของบางขั้นตอนที่ขาดหายไประหว่างการวิเคราะห์และแปลงจากแนวคิดทางตรรกะไปสู่แนวคิดระดับกายภาพเพื่อนำไปสู่ขั้นตอนอิมพลีเมนต์ การเปลี่ยนแปลงเริ่มจาก แบบจำลองวัตถุ แบบจำลองไดนามิค และแบบจำลองฟังก์ชัน ตามลำดับ แบบจำลองวัตถุนำมาพิจารณาถึงความสัมพันธ์ของแอตทริบิวต์และลดความซ้ำซ้อนระหว่างคลาสที่เกี่ยวข้องในไดอะแกรม แบบจำลองไดนามิคสามารถนำไปกำหนดขั้นตอนการทำงานของกระบวนการในวัตถุ ส่วนแบบจำลองฟังก์ชันกับแบบจำลองไดนามิคนำไปสร้างเป็นฟังก์ชัน นอกจากนี้เหตุการณ์จากแบบจำลองไดนามิคยังสามารถนำมาเป็นตัวควบคุมฟังก์ชันของวัตถุได้อีก โดยแบบจำลองทั้งสามผู้ออกแบบจะต้องนำมารวมกันให้เป็นกลุ่มเดียวกันหรือที่เรียกว่า "Packaging" ซึ่งขั้นตอนนี้จะต้องเลือกข้อมูลที่จะนำมากำหนดเป็นข้อมูลแบบปกปิด ความสอดคล้องกันของเอนติตี และการเชื่อมต่อกันระหว่างการวิเคราะห์กับการออกแบบระบบเพื่อให้เป็นอันหนึ่งอันเดียวกันทั้งระบบและง่ายต่อการจัดการในขั้นตอนการอิมพลีเมนต์ต่อไป

แบบจำลองน้ำพุ คือ แบบจำลองที่เหมาะสมกับการพัฒนาระบบสารสนเทศเชิงวัตถุ เนื่องจากมีลักษณะการทำซ้ำและการเวียนบังเกิดของแบบจำลองน้ำพุ เป็นรูปแบบการพัฒนาระบบสารสนเทศที่มีความยืดหยุ่นและครอบคลุมทั้งการวิเคราะห์และการออกแบบ จึงทำให้มีลักษณะของการนำกลับมาใช้ใหม่ได้และแบบจำลองนี้อยู่ภายใต้วิธีการของวิศวกรรมซอฟต์แวร์ หรือโมเสส (MOSES) ซึ่งประกอบด้วย Product Lifecycle, Process Lifecycle และกิจกรรมต่าง ๆ ตลอดทั้งวงจรชีวิตของการพัฒนาระบบ

Product Lifecycle คือ วงจรชีวิตทางธุรกิจและการพัฒนาของวงจรชีวิต Product Lifecycle จะแบ่งเป็นโครงการระยะสั้นกับระยะยาว โครงการระยะสั้นมีระยะเวลาการพัฒนาประมาณ 1 ปี และถ้าเป็นโครงระยะยาวมีระยะเวลาการพัฒนามากกว่า 1 ปี โดยโครงการสามารถแบ่งออกเป็นส่วนย่อย ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะทำให้สะดวกในการบริหารและจัดการโครงการนั้น ซึ่งในแต่ละโครงการจะประกอบด้วย Business Planning, Build และ Delivery ซึ่งเป็นขั้นตอนทางธุรกิจทั้งสิ้น ใน Business Planning และ Delivery เป็นขั้นตอนที่มีความสำคัญ เพราะเป็นส่วนเริ่มต้นและตัดสินใจของโครงงาน ส่วน Build กล่าวถึงเทคนิคการพัฒนาระบบสารสนเทศ ซึ่งทั้งหมดจะทำซ้ำ ๆ กันเช่นนี้จนกระทั่งสิ้นสุดโครงการ และรวมไปถึงการบำรุงรักษาระบบ

Process Lifecycle คือ วงจรชีวิตของการพัฒนา โดยกล่าวถึงขั้นตอน Build ที่อยู่ใน Product Lifecycle ซึ่งครอบคลุมตั้งแต่ การวางแผน การศึกษาความเหมาะสม วิเคราะห์ออกแบบ อิมพลีเมนต์ และการตรวจสอบ ซึ่งทั้งหมดเป็นเทคนิคและขั้นตอนการสร้างระบบสารสนเทศและแต่ละขั้นตอนจะมีกิจกรรมย่อยและเอกสารประกอบตลอดการพัฒนา และประกอบด้วยกิจกรรม 20 กิจกรรม โดยผู้พัฒนาสามารถเพิ่มหรือลดกิจกรรมได้ตามความเหมาะสม

แบบจำลองประยุกต์ เป็น แบบจำลองที่ผสมผสานระหว่าง วิธี เอส เอส เอ ดี เอ็มกับวิธีเชิงวัตถุภายใต้แบบจำลองน้ำพุ ซึ่งแบบจำลองประยุกต์นำข้อเด่นของแต่ละวิธีมาสอดแทรกทั้งวงจรชีวิตของการพัฒนาระบบสารสนเทศ เริ่มต้นที่วิธีเชิงโครงสร้างผู้วิจัยนำขั้นตอนการดำเนินงานที่สามารถติดตามได้ของวิธี เอส เอส เอ ดี เอ็ม มาประยุกต์ ส่วนวิธีเชิงวัตถุนำคุณสมบัติของวิศวกรรมซอฟต์แวร์ เช่น สามารถนำกลับมาใช้ใหม่ได้ (Reusable) มีความถูกต้องและตรวจสอบได้ (Correctness and Verification) สามารถขยายและปรับปรุงได้ง่าย (Extensible) มีความเสถียรและคงทน (Robustness) สุดท้ายนำลักษณะการพัฒนาโครงงานของแบบจำลองน้ำพุที่มีลักษณะของการทำซ้ำ (Iterative) และการเวียนบังเกิด (Recursive) ได้ดี ดังนั้นแบบจำลองประยุกต์ที่ถูกพัฒนาขึ้นจึงสามารถนำไปพัฒนากับระบบสารสนเทศทั้งแบบเชิงวัตถุและเชิงโครงสร้างได้และผลลัพธ์ที่ได้ไม่ต่างจากวิธีการพัฒนาระบบสารสนเทศโดยทั่วไป ส่วนการพัฒนาตามแบบจำลองประยุกต์ถูกแบ่งออกเป็น 2 ระดับ คือ ระดับของระบบ (System-Level) และระบบย่อย (Sub-System Level) ซึ่งแต่ระดับประกอบด้วยวงกลมที่แสดงถึงขั้นตอนและกิจกรรมต่าง ๆ ประกอบอยู่ภายในอย่างชัดเจน

7.2 ข้อเสนอแนะ

แม้ว่าแบบจำลองประยุกต์ที่ผู้วิจัยพัฒนาขึ้นนั้น มีขั้นตอนการพัฒนาที่มากกว่าวิธีการวิเคราะห์และออกแบบโดยทั่วไป แต่สิ่งที่เกิดขึ้นระหว่างการพัฒนาของแบบจำลองประยุกต์นั้นสามารถติดตามและแสดงวิธีการทำงานได้ชัดเจนกว่าวิธีอื่น ๆ พร้อมกับยังสามารถนำไปพัฒนากับระบบสารสนเทศทั้งแบบเชิงโครงสร้างและแบบเชิงวัตถุ เพราะแบบจำลองนี้อยู่ภายใต้แบบจำลองที่สามารถนำไปใช้ได้กับทุกระบบสารสนเทศที่ต้องการ ซึ่งบางครั้งบางกิจกรรมที่เกิดขึ้นของแต่ละขั้นตอนนั้นไม่มีความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นต่อระบบนั้นก็ตาม แต่กิจกรรมที่ไม่เกี่ยวข้องนั้นสามารถที่จะละเลยไปได้เพราะขั้นตอนจะมีคำอธิบายให้ผู้พัฒนาทราบโดยตลอด เช่น ระบบสารสนเทศที่ไม่จำเป็นต้องวางแผนสร้างระบบย่อย ๆ หรือหลายทีมงานก็ไม่จำเป็นต้องปฏิบัติ หรือ ระบบสารสนเทศที่ไม่จำเป็นต้องสร้างระบบให้สามารถทำงานภายใต้ระบบเครือข่าย เป็นต้น

แบบจำลองประยุกต์ที่พัฒนาขึ้นเมื่อเปรียบเทียบกับแบบจำลองหรือวิธีการอื่น ที่พบเห็นโดยทั่วไปสำหรับการพัฒนาระบบสารสนเทศนั้น จะมีความละเอียดมากกว่าซึ่งผู้วิจัยเชื่อว่าขั้นตอนบางขั้นตอนนั้นไม่มีอยู่ในวิธีอื่น เช่น การวัดความซับซ้อนของการวิเคราะห์และออกแบบ การเปลี่ยนจากวิธีเชิงโครงสร้างไปสู่วิธีเชิงวัตถุ เป็นต้น การพัฒนาระบบสารสนเทศโดยทั่วไป (รวมถึงแบบจำลองประยุกต์) มักมีการเริ่มต้นการพัฒนาที่เหมือนกันและจะแต่ต่างกันที่ขั้นตอนการวิเคราะห์และออกแบบ สุดท้ายผลลัพธ์ คือ โครงงานที่ตรงกับความต้องการของผู้ใช้เหมือนกัน ดังนั้นแบบจำลองประยุกต์จึงเป็นอีกทางเลือกหนึ่งสำหรับการนำไปใช้สำหรับพัฒนาระบบสารสนเทศที่เห็นว่าเหมาะสม

7.3 แนวทางการวิจัยต่อ

แบบจำลองประยุกต์ เป็น การประยุกต์กับหลายวิธีจึงทำให้สัญลักษณ์และขั้นตอนต่าง ๆ ที่เกิดขึ้นนั้นมีความหลากหลายอยู่บ้าง แต่เป็นแบบจำลองที่มีความชัดเจนและติดตามได้ง่ายกว่าวิธีอื่นก็ตาม แต่สิ่งที่เป็นแนวทางสำหรับการวิจัยต่อ ผู้วิจัยมีข้อเสนอดังนี้

1. ปรับปรุงสัญลักษณ์ให้เป็นเอกเทศ
2. สร้างรูปแบบและเอกสารให้กระชับและชัดเจนในแต่ละขั้นตอน
3. สร้างเครื่องมือสำหรับการวิเคราะห์และออกแบบของแบบจำลองประยุกต์
4. ควรเลือกนำแบบจำลองไปใช้กับระบบสารสนเทศที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] นิภาพร ประภาศิริ, เอื้อน ปิ่นเงิน. "การวัดความซับซ้อนของซอฟต์แวร์." **สารสนเทศลาดกระบัง,** ปีที่ 3, ฉบับที่ 1, มกราคม 2541. หน้า 42-55.

[2] พัฒนพงษ์ ฉันทมิตร โอภาส, เอื้อน ปิ่นเงิน. "แบบจำลองน้ำพุเพื่อการพัฒนาระบบเชิงวัตถุ" **วารสารพระจอมเกล้าลาดกระบัง**, ปีที่ 7, ฉบับที่ 1, มีนาคม 2542. หน้า 48-57.

[3] Alan Dennis and Barbara Haley Wixon. **System Analysis and Design.** USA: John Wiley & Son, Inc. 2000.

[4] Albrecht A.J. and Gaffney J.E. "Software function, source line of code and development effort prediction: a software science validation" IEEE Transactions on Software Engineering, Vol 9, No. 6, 1983. pp. 639-647.

[5] Arturo I Concepcion, Sunny Lin and Scott J. Simon. "Managing the Software Development by Using the Recursive Multi-Threaded (RMT) Tool." [CD-ROM]. IEEE. 1999.

[6] Barry W. Boehm. "A Spiral Model of Software Development and Enhancement." IEEE Engineering Management Review, Vol.23, No.4, Winter 1995. pp. 69-80.

[7] Brachman, R. J. "I lied about the trees or, defaults and definitions in knowledge representation." AI Magazine, Vol. 6, No. 3, 1985. pp. 80-93.

[8] Card, D. N. and R. L. Glass. **Measuring Software Design Quality.** New Jersey : Prentice-Hall, Inc. 1990.

[9] Chidamber S.R. and Kemerer C.F. "Towards a metrics suite for object-oriented design" OOPSLA'91 ACM Conference of Object-oriented Programming Systems, Languages and Applications, Addison-Wesley. 1991.

[10] Coleman, D., Hayes, F. and Bear, S. "Introducing objectcharts or how to use statecharts in object-oriented design." IEEE Transactions on Software Engineering, Vol. 18, No. 1, 1992. pp. 9-18.

[11] de Champeaux, D. **Object-oriented analysis and top-down software development.** European Conference on Object-oriented Programming 1991, Lecture Notes in Computer Science, No. 512, Springer-Verlag, Berlin, 1991. pp. 360-376.

[12] Effy Oz. **Management Information Systems.** USA: International Thomson Publishing Company. 2000.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[13] Grady Booch. **Object-Oriented Design with Applications.** The Benjamin/Cummings Publishing Company Inc., Redwood City, CA. 1991.

[14] Haifeng Qian. et. al. "A Combined Functional and Object-Oriented Approach to Software Design." [CD-ROM]. IEEE. 1995.

[15] Henderson-Sellers, B. "Object Technology: w(h)ither the future?" Technology of Object-Oriented Languages and Systems Tools12, 1993.

[16] Henderson-Sellers, B. and Edwards, J.M. "The Object-Oriented system life cycle." Communications of The ACM, Vol. 33, No. 9, 1990, pp. 142-159.

[17] Henderson-Sellers, B. and Edwards, J.M. **Book Two of Object-Oriented Knowledge: The Working Object.** Sydney: Prentice-Hall, Inc. 1994.

[18] Henderson-Sellers, B. and Edwards, J.M. "The fountain model for object-oriented system development." Object Magazine, Vol. 3, No. 2, 1993. pp. 71-79.

[19] Henderson-Sellers, B. and Pant, Y.R. "When should we generalize classes to make them reusable?." Object Magazine, Vol. 3, No. 4, 1993, pp. 73-75.

[20] Henderson-Sellers, B. **Object-Oriented Metrics: Measures of Complexity.** New Jersey : Prentice-Hall, Inc. 1996.

[21] Henry, S. and D. Kafura. "Software Structure Metrics Based on Information Flow." IEEE Transaction on Software Engineering, Vol. 7, No. 5, 1981, pp. 510-518.

[22] Ian Graham. **Object-Oriented Methods.** 2nd ed. London : Addison-Wesley publishing Company, Inc. 1994.

[23] Ian Graham. **Migrating to Object Technology.** London : Addison-Wesley publishing Company, Inc. 1995.

[24] IBM Object-Oriented Technology Center. **Developing Object-Oriented Software: an experience-based approach.** New Jersey: Prentice-Hall, Inc. 1997.

[25] Ivar Jacobson et. al. **Object Oriented Software Engineering : A use case driven approach**. Wokingham: Addsion-Wesley, 1992.

[26] Jack R. Meredith and Samuel J. Mantel, Jr. **Project Management: A Managerial Approach.** 3rd ed. USA: Addison-Wesley publishing Company, Inc. 1995.

[27] James Martin and James J. Odell. **Object-Oriented Methods : A Foundation.** New Jersey : Prentice-Hall, Inc. 1995.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[28] James Rumbaugh et. al. **Object-Oriented Modeling and Design.** New Jersey : Prentice-Hall, Inc. 1991.

[29] Jean-March Nerson. "Applying Object-Oriented Analysis and Design" Communication of the ACM, Vol.35, No. 9, 1992.

[30] Jeffrey L. Whitten et. al. **Systems Analysis and Design Methods.** 4th ed. McGraw-Hill, Inc. 1998.

[31] Johnson, R. E. and Foote, B. "Designing reusable classes" Journal Object-Oriented Programming, Vol. 1, No. 2, 1988, pp. 22-35.

[32] Lewi, J. et. al. "Object-Oriented Approach to Software Development : A walk through a number of topics." [CD-ROM]. IEEE. 1991.

[33] Lorenz, M. **Object-Oriented Software Development: A Practical Guide.** Prentice Hall, Englewood Cliffs, NJ, pp. 227. 1993.

[34] Luiz Fernando Capretz and Miriam A. M. Capretz. **Object-Oriented Software : Design and Maintenance.** London : World Scientific. 1996.

[35] McCabe, T. J. "A complexity measure." IEEE Transaction on Software Engineering, Vol. 2, No. 4, 1976. pp. 308-320.

[36] McGregor, J. D. and Sykes, D. A. **Object-Oriented Software Development: Engineering Software for Reuse.** Van Nostrand Reinhold. New York, 1992. pp. 352.

[37] Meyer, B. **Object-Oriented Software Construction.** 2nd ed. New Jersey : Prentice-Hall, Inc. 1998.

[38] Page-Jones, M., Constantine, L. L. and Weiss, S. "Modeling object-oriented systems: the Uniform Object Notation." Computer Language, Vol. 7, No. 10, 1990. pp. 69-87.

[39] Peter Coad and Edward Yourdon. **Object-Oriented Analysis.** 2nd ed. New Jersey : Prentice-Hall, Inc. 1991.

[40] Peter Coad and Edward Yourdon. **Object-Oriented Design.** New Jersey : Prentice-Hall, Inc. 1991.

[41] Peter Rob and Carlos Coronel. **Database Systems: Design implement and management 4th.** Course Technology. 2000.

[42] Philip L. Weaver et. al. **Practical SSADM version 4+ : A Complete Tutorial Guide.** 2nd ed. Great Britain : Pitman Publishing. 1998.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[43] Philip L. Weaver. **Practical SSADM version 4: A Complete Tutorial Guide.** London : Pitman Publishing. 1993.

[44] Robert G. Fichman and Chris F. Kemerer. "Object-Oriented and Conventional Analysis and Design Methodologies." [CD-ROM]. IEEE. 1992.

[45] Roger S. Pressman. **Software Engineering: A paractitioner's approach 4th.** McGrow-Hill, Inc. 1997.

[46] Royce W. W. "Managing the Development of Large Software Systems: Concepts and Techniques," 1970 WESCON Technical Papers, Western Electronic Show and Convention, Los Angeles, August 1970, pp. A/1-1-A/1-9. Reprinted in: Proceeding of the 11th International Conference of Software Enginieering, Pittsburgh, May 1989, pp. 328-338.

[47] Shuguang Hong, Geert van den Goor and Sjaak Brinkkemper. "A Formal Approach to the Comparison of Object-Oriented Analysis and Design Methodologies" [CD-ROM]. IEEE. 1993.

[48] Stefan Sigfried. **Understanding Object-Oriented Software Engineering.** New York : IEEE Press. 1996.

[49] Stephen Montgomery. **Building Object-Oriented Software.** New York : McGraw-Hill, Inc. 1998.

[50] Stephen R. Schach. **Classical and Object-Oriented Software Engineering with UML and JAVA.** 4th ed. McGrow-Hill, Inc. 1999.

[51] T. Aoki and T. Katayama. "Formal Model Approach for Reliable Object-Oriented Information System Design." Sci/ISAS'98, 1998. pp. 228-235.

[52] T. Aoki and T. Katayama. "Unification and Consistency Verification of Object-Oriented Analysis Models" IEEE Software Engineering Conference 98, Proceedings 1998 Asia Pacific, 1998. pp. 296-303.

[53] Wirfs-Brock, A. and Wilkerson, B. "Variables limit reusability" Journal Object-Oriented Programming. Vol. 2, No. 1, 1989a, pp. 34-40.

[54] Wirfs-Brock, R. J., Wilkerson, B. and Wiener, L., **Design Object Oriented Software.** Prentice-Hall, Englewood Cliffs, N.J., 1990.

[55] Wong C. "A Successful Software Development," IEEE Transactions of Software Engineering, Vol. 10, No. 11, 1984, pp. 714-724.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

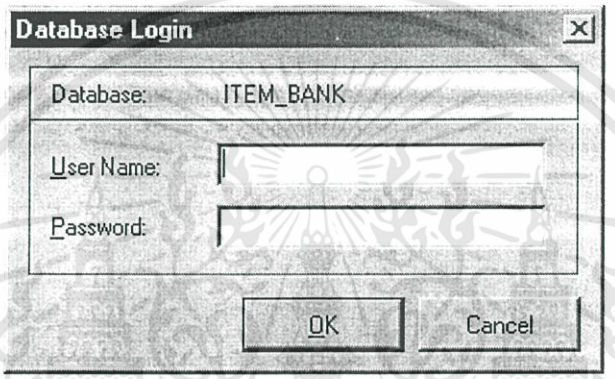
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
ระบบคลังข้อสอบ (ต้นแบบ) ตามกรณีศึกษา

การพัฒนาระบบคลังข้อสอบตามแบบจำลองประยุกต์สามารถแสดงในรูปของแอพพลิเคชัน (ต้นแบบ) ซึ่งมีลำดับของหน้าจอและการทำงานดังต่อไปนี้

1. การเข้าสู่ระบบคลังข้อสอบ

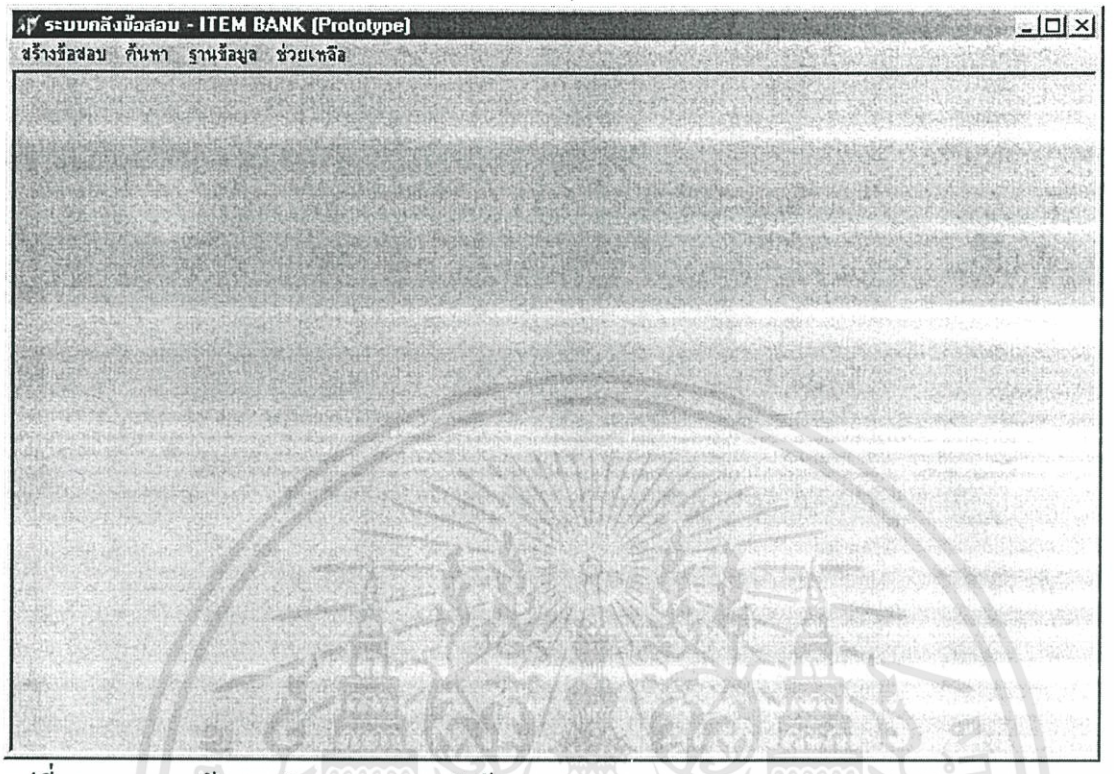


รูปที่ ก.1 แสดงหน้าจอสำหรับเข้าสู่ระบบคลังข้อสอบ

จากรูปที่ ก.1 แสดงหน้าจอสำหรับเข้าสู่ระบบคลังข้อสอบโดยการเข้าสู่ระบบนั้น ผู้ใช้แต่ละคนจะมีรหัสผ่านของตนเองเพื่อเข้าสู่ระบบ ซึ่งรหัสผ่านของแต่ละคนนั้นสามารถขอได้จาก DBA (Database Administrator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมนูหลักของระบบคลังข้อสอบ



รูปที่ ก.2 แสดงหน้าจอหลักของระบบคลังข้อสอบ

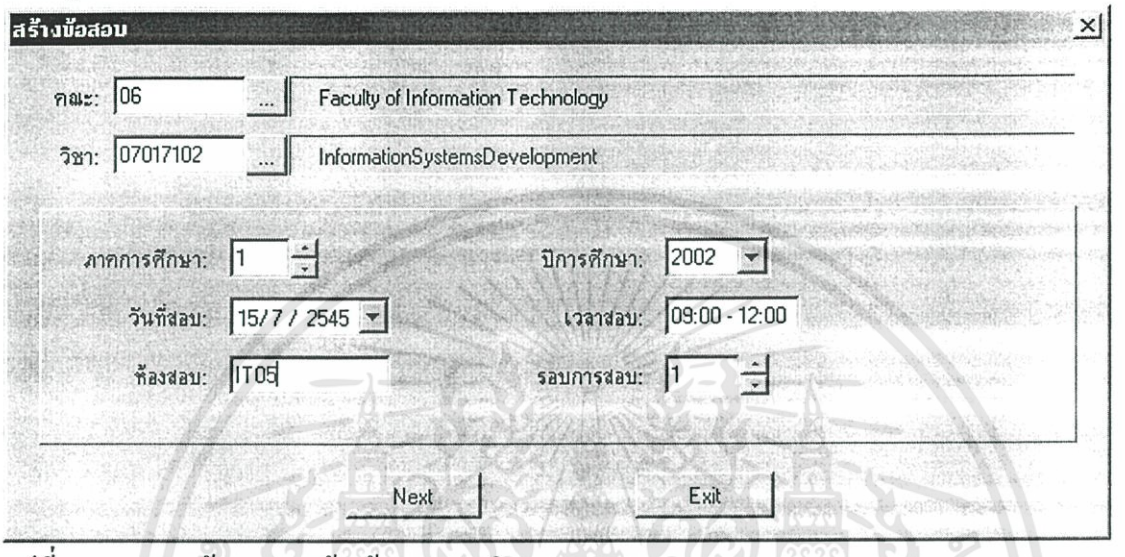
จากรูปที่ ก.2 แสดงหน้าจอหลักของระบบหลังจากผู้ใช้ป้อนรหัสผ่านของตนเองเรียบร้อยแล้ว ระบบจะแสดงหน้าจอหลักโดยแบ่งเมนูการทำงานออกเป็น 4 ส่วน คือ ส่วนของการสร้างข้อสอบ การค้นหา การสร้างฐานข้อมูลและการช่วยเหลือ ซึ่งแต่ละส่วนมีเมนูย่อยประกอบอยู่ภายในดังต่อไปนี้

- เมนูการสร้างข้อสอบ คือ เมนูสำหรับการสร้างข้อสอบเพื่อจัดทำข้อสอบให้นักศึกษาสอบตามหลักสูตรของสถาบันการศึกษา
- เมนูฐานข้อมูล คือ เมนูสำหรับบันทึก แก้ไข ลบข้อมูลให้กับระบบ เช่น ฐานข้อมูลข้อสอบ ฐานข้อมูลอาจารย์ ฐานข้อมูลคณะและฐานข้อมูลวิชา เป็นต้น
    - ฐานข้อมูลข้อสอบ คือ ตัวข้อสอบของแต่ละวิชาของแต่ละคณะ ซึ่งการบันทึกข้อสอบสามารถเพิ่มเติม แก้ไขและลบ ตามความต้องการ ข้อสอบสามารถบันทึกแยกตามประเภทข้อสอบ เช่น ปรนัย อัตนัย และ ถูก/ผิด เป็นต้น
    - ฐานข้อมูลอาจารย์ คือ การบันทึก แก้ไขและลบข้อมูลของอาจารย์ภายในสถาบันการศึกษา
    - ฐานข้อมูลคณะ คือ การบันทึก แก้ไขและลบข้อมูลคณะภายในสถาบันการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฐานข้อมูลวิชา คือ ข้อมูลวิชาของแต่ละคณะที่ทำการเรียนการสอนภายในสถาบันการศึกษาซึ่งผู้ใช้สามารถบันทึก แก้ไขและลบข้อมูลได้ตามความต้องการ

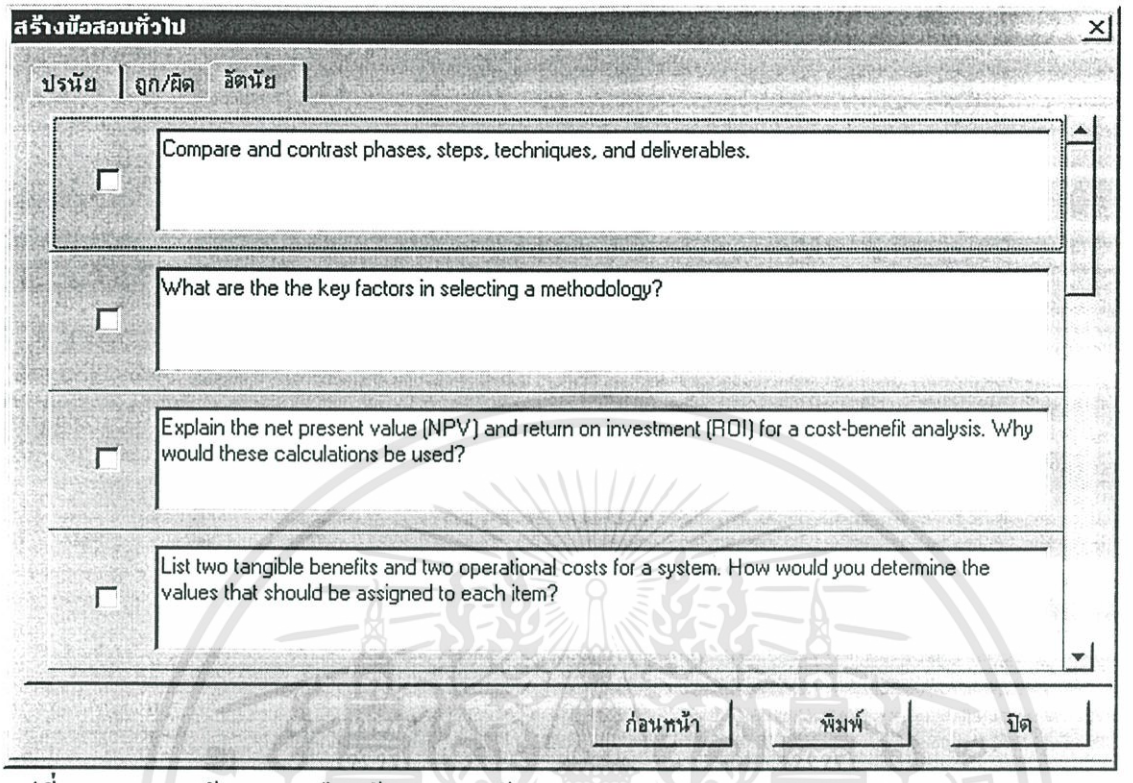
3. เมนูการสร้างข้อสอบ



รูปที่ ก.3 แสดงหน้าจอการสร้างข้อสอบรายวิชา

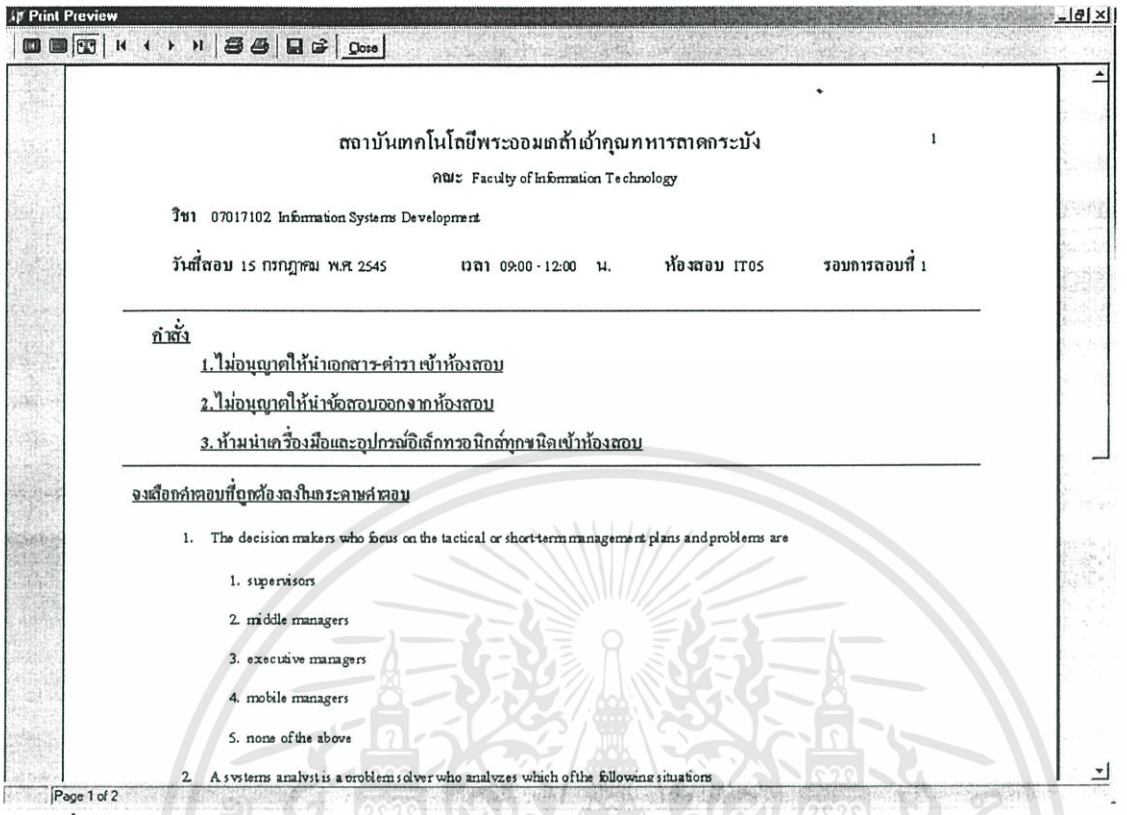
จากรูปที่ ก.3 แสดงหน้าจอการสร้างข้อสอบของแต่ละวิชา ซึ่งผู้ออกข้อสอบต้องใส่ข้อมูลเบื้องต้นของการสร้างข้อก่อนเลือกตัวข้อส อบจากฐานข้อมูลข้อสอบ ข้อมูลเบื้องต้น ได้แก่ คณะ วิชา ภาคการศึกษา ปีการศึกษา วันที่สอบ เวลาสอบ ห้องสอบและรอบการสอบ เป็นต้น จากนั้นกดปุ่ม 'NEXT' เพื่อทำการเลือกตัวข้อสอบจากฐานข้อมูลข้อสอบต่อไป (แสดงดังรูปที่ ก.4) เมื่อเลือกข้อสอบตามต้องการเรียบร้อยจะเข้าสู่หน้าจอเตรียมพิมพ์ข้อสอบ (แสดงดังรูปที่ ก.5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.4 แสดงหน้าจอการเลือกข้อสอบรายข้อ

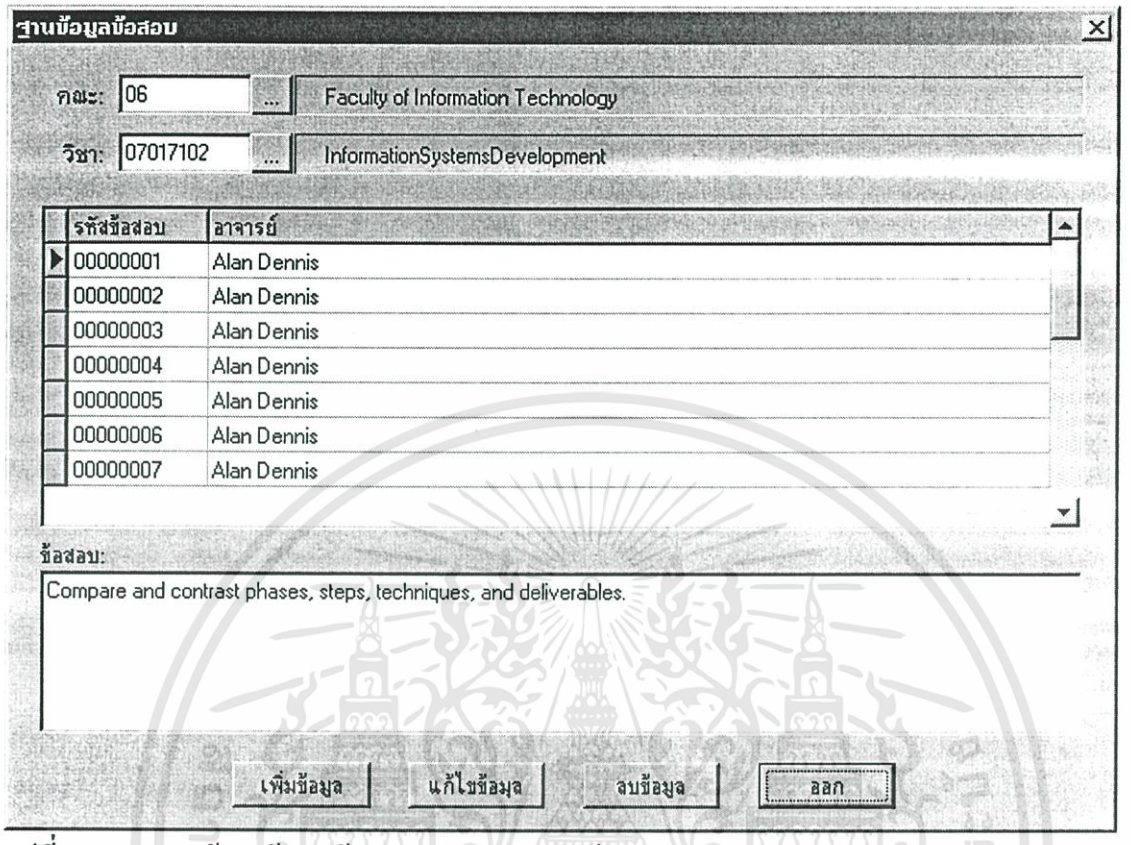
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.5 แสดงหน้าจอเตรียมพิมพ์ข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

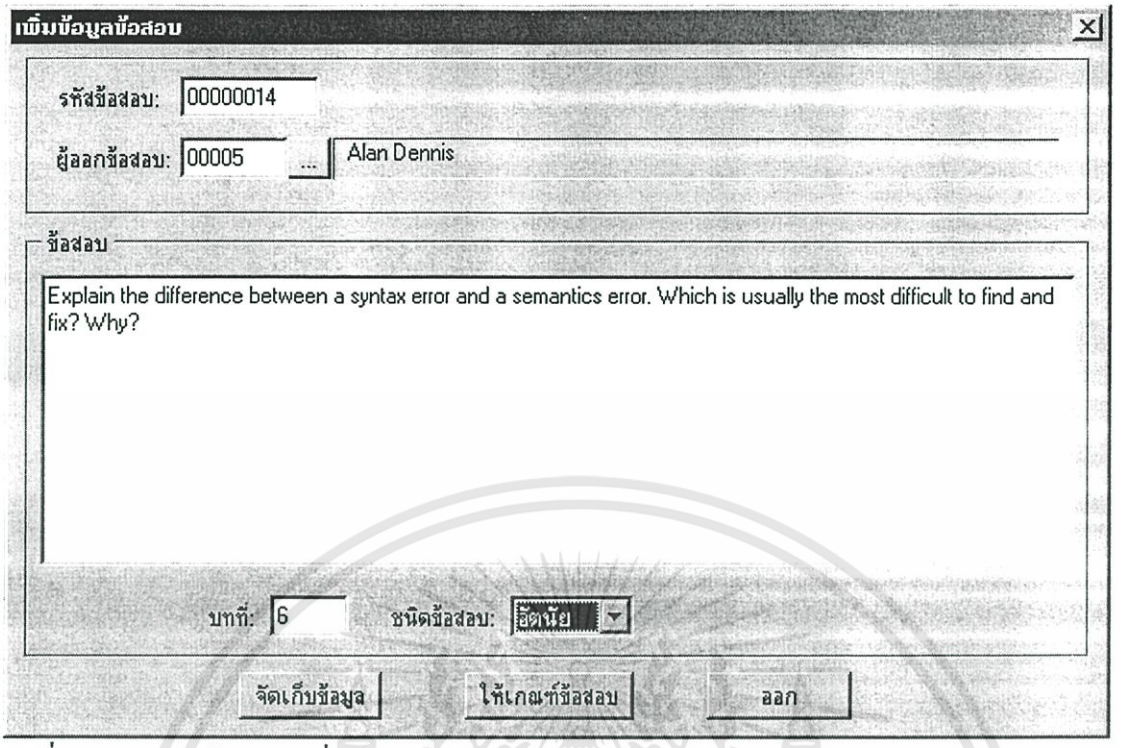
4. เมนูฐานข้อมูลข้อสอบ



รูปที่ ก.6 แสดงหน้าจอข้อมูลข้อสอบของระบบคลังข้อสอบ

จากรูปที่ ก.6 แสดงหน้าจอฐานข้อมูลข้อสอบ หน้าจอแสดงรายละเอียดข้อสอบแต่ละวิชาของคณะและอยู่ภายในเมนูของฐานข้อมูล หน้าจอฐานข้อมูลข้อสอบประกอบด้วยปุ่ม 4 ปุ่ม คือ เพิ่มข้อมูล แก้ไขข้อมูล ลบข้อมูล และ ออกจากหน้าจอฐานข้อมูลข้อสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

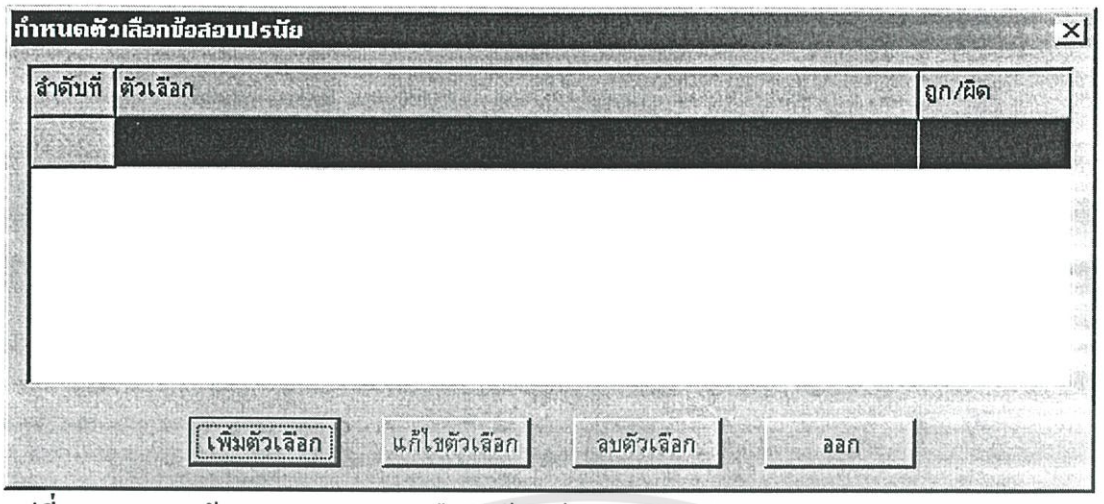


รูปที่ ก.7 แสดงหน้าจอการเพิ่มข้อมูลข้อสอบ

จากรูปที่ ก.7 แสดงหน้าจอการเพิ่มข้อมูลข้อสอบซึ่งผู้บันทึกข้อสอบต้องให้รหัสข้อสอบของแต่วิชาที่เลือกและรหัสผู้ออกข้อสอบ จากนั้นบันทึกข้อคำถามของข้อสอบแต่ละข้อและเลือกประเภทของข้อสอบ ได้แต่ ข้อสอบประเภทปรนัย ข้อสอบประเภทอัตนัยและข้อสอบถูก/ผิด เป็นต้น ข้อสอบประเภทอัตนัยผู้ออกข้อสอบต้องให้รายละเอียดตัวเลือกของข้อคำถามตามต้องการ (แสดงดังรูปที่ ก.8 และ รูปที่ ก.9 ตามลำดับ) ข้อสอบประเภทอัตนัยและถูก/ผิด ผู้ออกข้อสอบต้องบันทึกคำตอบลงฐานข้อมูล แต่ข้อสอบประเภทปรนัยผู้ออกข้อสอบไม่จำเป็นต้องบันทึกคำตอบ ส่วนการบันทึกเกณฑ์ข้อสอบนั้นผู้ออกข้อสอบจะให้เกณฑ์ทันทีที่บันทึกข้อคำถามหรือบันทึกภายหลังได้

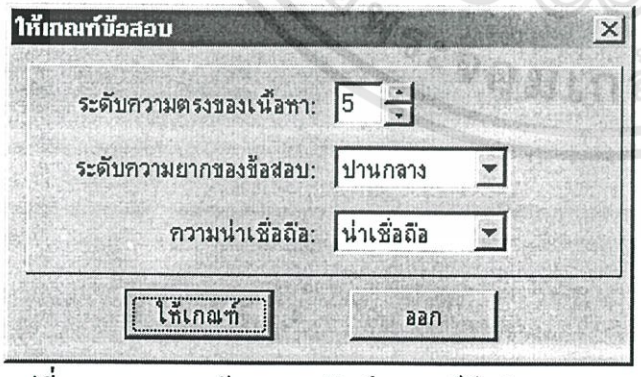
การบันทึกเกณฑ์ข้อสอบถูกแบ่งออกเป็น 3 ประเภท คือ ระดับความตรงกับเนื้อหาที่ทำการเรียนการสอน ระดับความยากของข้อคำถามและระดับความหน้าเชื่อถือของข้อคำถาม (แสดงดังรูปที่ ก.10)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.8 แสดงหน้าจอการกำหนดตัวเลือกให้กับข้อสอบประเภทปรนัย

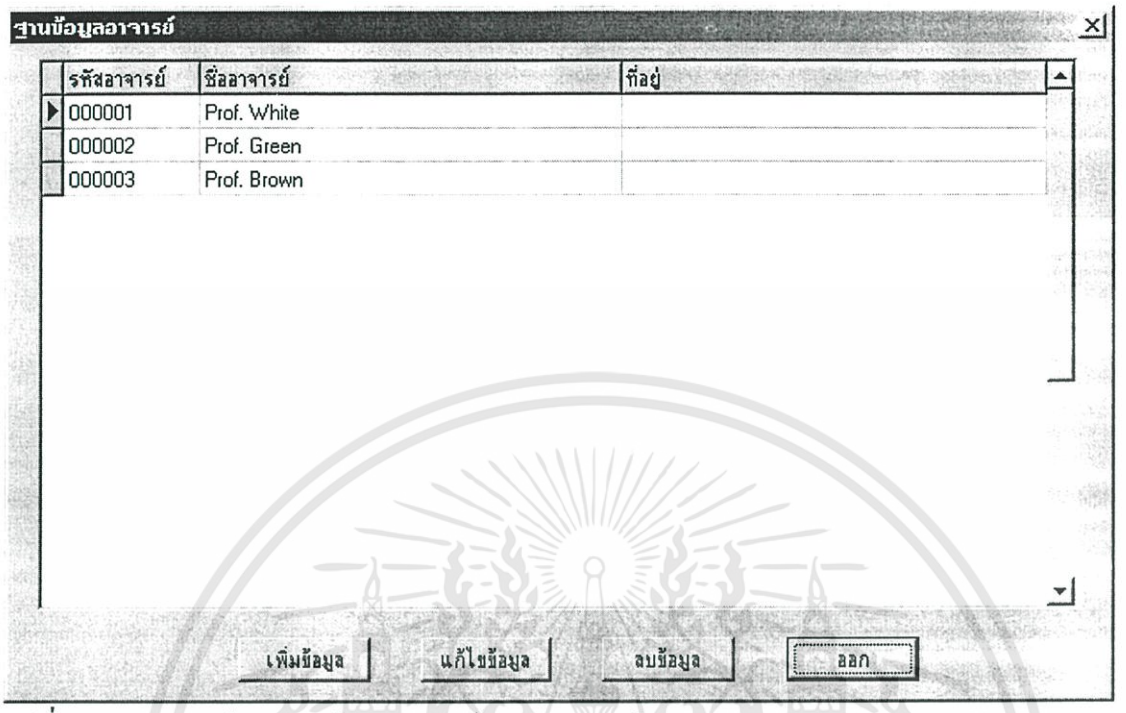
รูปที่ ก.9 แสดงหน้าเพิ่มตัวเลือกให้กับข้อสอบประเภทปรนัย



รูปที่ ก.10 แสดงหน้าจอการบันทึกเกณฑ์ข้อคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

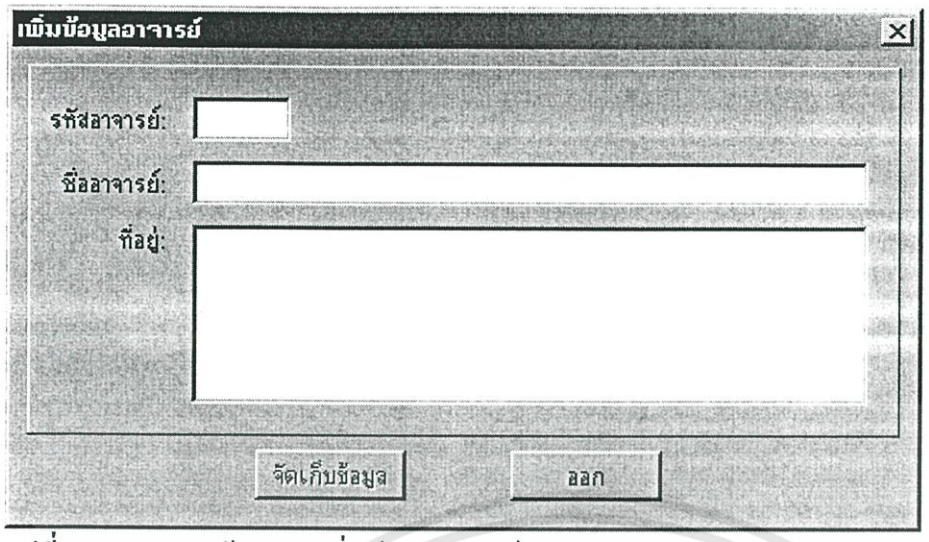
5. เมนูฐานข้อมูลอาจารย์



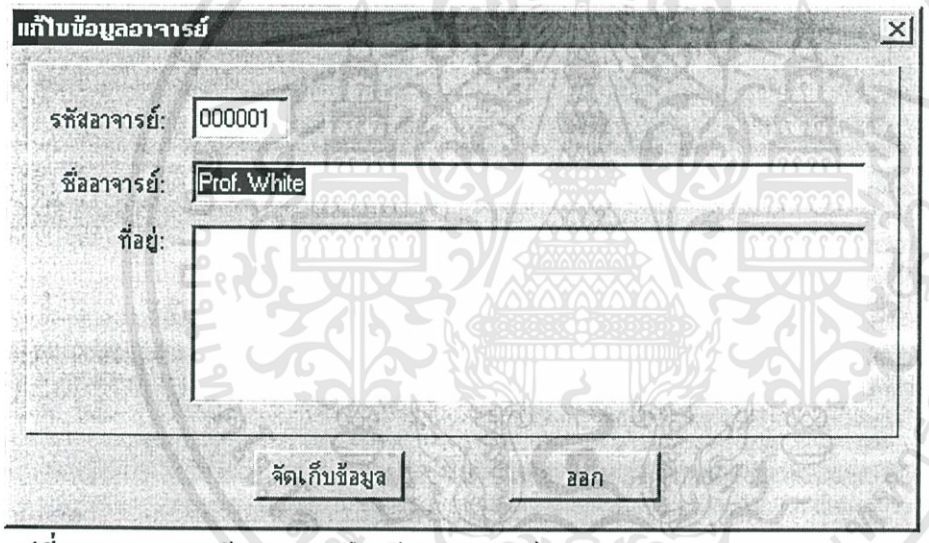
รูปที่ ก.11 แสดงหน้าจอฐานข้อมูลอาจารย์ในระบบคลังข้อสอบ

จากรูปที่ ก.11 แสดงหน้าจอฐานข้อมูลอาจารย์ของสถาบันการศึกษา หน้าจอแสดงรายชื่ออาจารย์อยู่ภายในเมนูของฐานข้อมูล หน้าจอฐานข้อมูลอาจารย์ประกอบด้วยปุ่ม 4 ปุ่ม คือ เพิ่มข้อมูล แก้ไขข้อมูล ลบข้อมูล และ ออกจากหน้าจอฐานข้อมูลอาจารย์ หน้าจอการเพิ่มและแก้ไขข้อมูลแสดงดังรูปที่ ก.12 และ รูปที่ ก.13 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



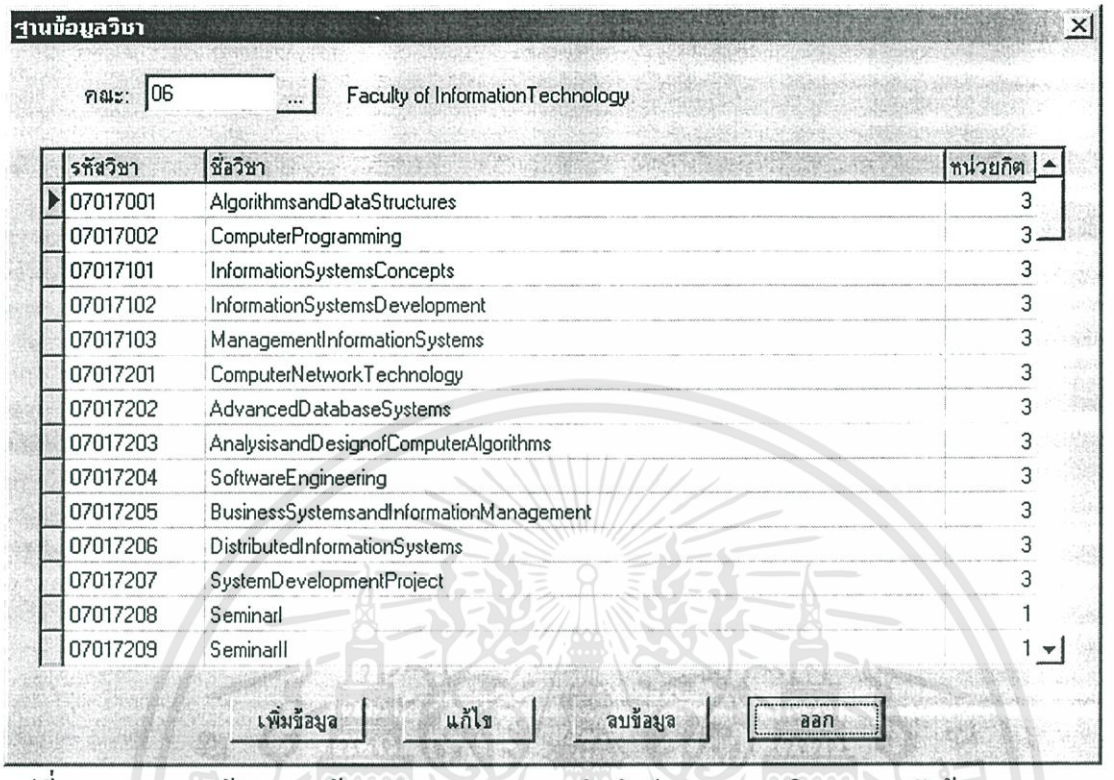
รูปที่ ก.12 แสดงหน้าจอการเพิ่มข้อมูลอาจารย์



รูปที่ ก.13 แสดงหน้าจอการแก้ไขข้อมูลอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

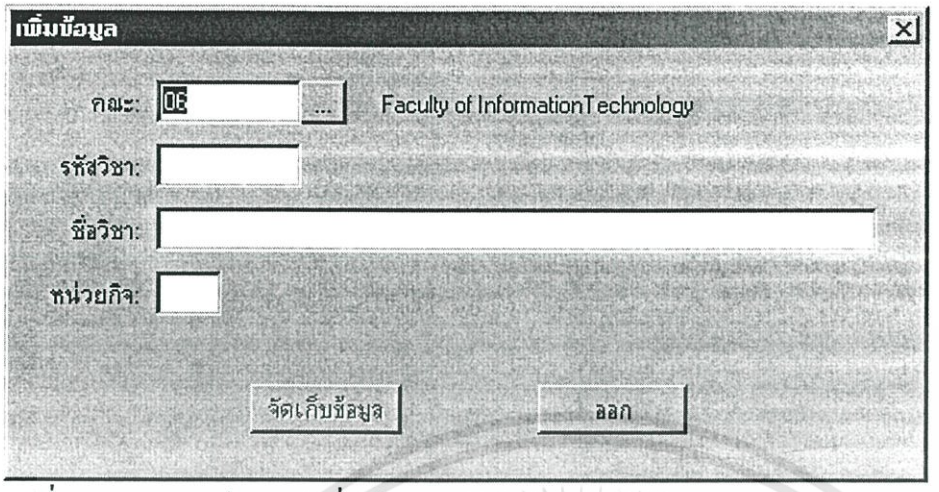
6. เมนูฐานข้อมูลวิชา



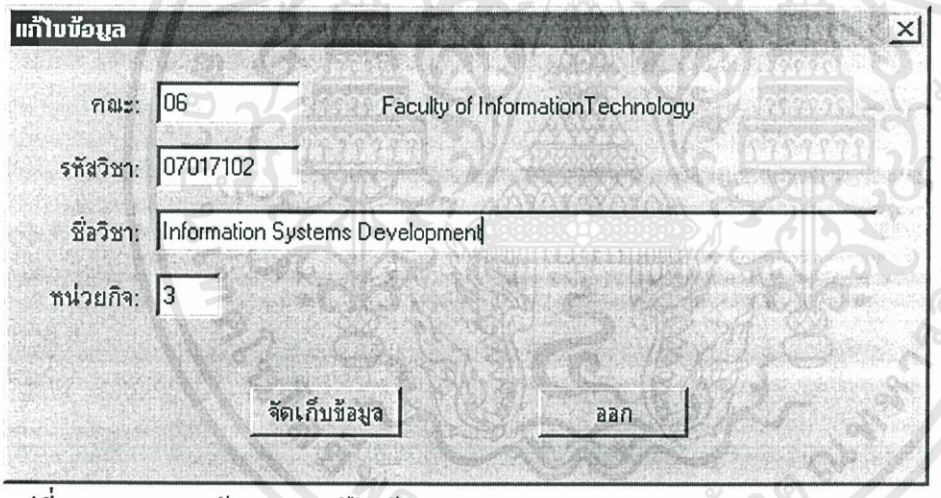
รูปที่ ก.14 แสดงหน้าจอฐานข้อมูลวิชาของคณะเทคโนโลยีสารสนเทศในระบบคลังข้อสอบ

จากรูปที่ ก.14 แสดงหน้าจอฐานข้อมูลวิชาของสถาบันการศึกษา หน้าจอแสดงวิชาอยู่ภายในเมนูของฐานข้อมูล หน้าจอฐานข้อมูลวิชาประกอบด้วยปุ่ม 4 ปุ่ม คือ เพิ่มข้อมูล แก้ไขข้อมูล ลบข้อมูล และ ออกจากหน้าจอฐานข้อมูลวิชา หน้าจอการเพิ่มและแก้ไขข้อมูลแสดงดังรูปที่ ก.15 และรูปที่ ก.16 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



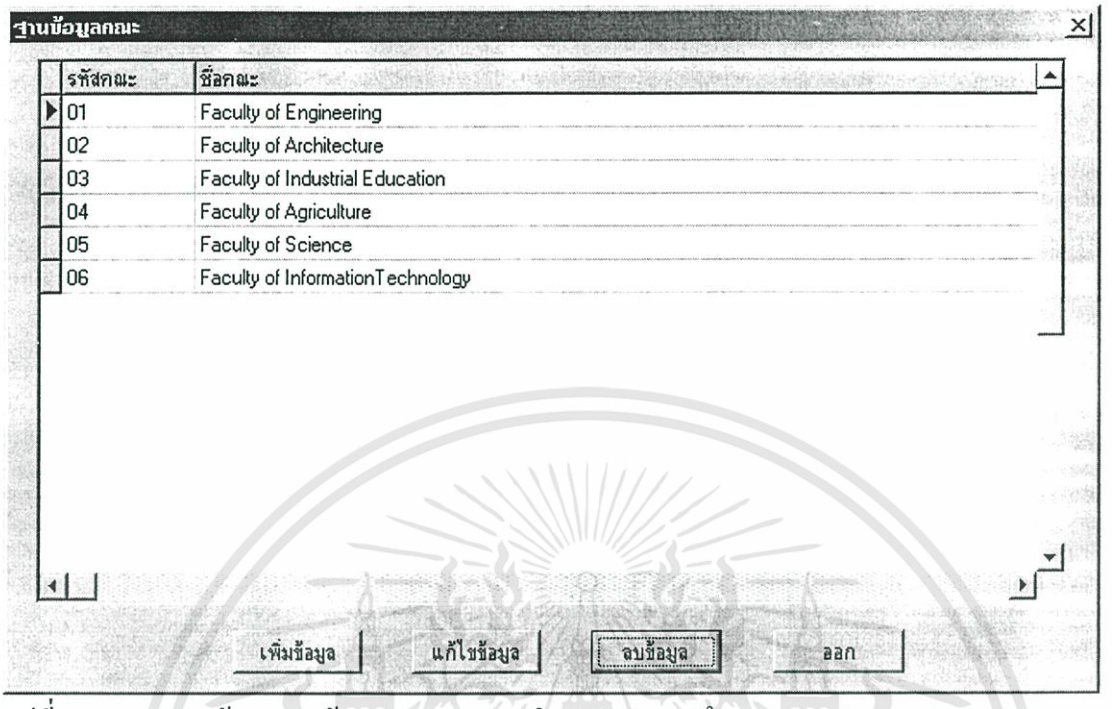
รูปที่ ก.15 แสดงหน้าจอการเพิ่มข้อมูลวิชา



รูปที่ ก.16 แสดงหน้าจอการแก้ไขข้อมูลวิชา

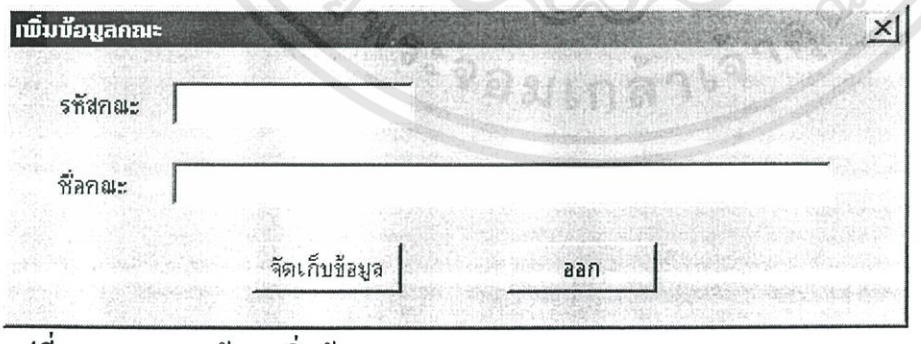
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เมนูฐานข้อมูลคณะ



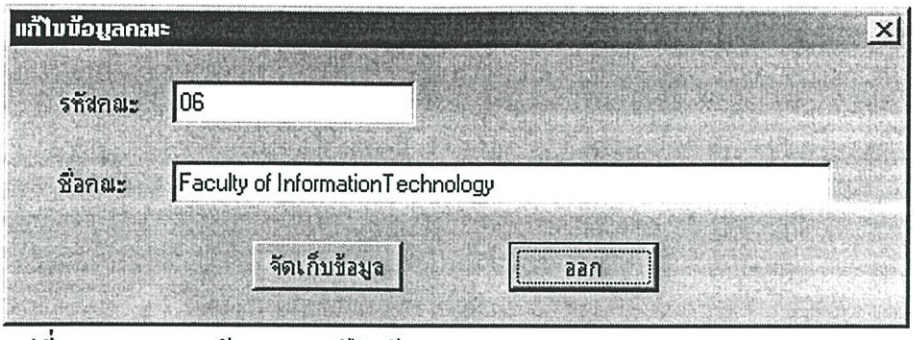
รูปที่ ก.17 แสดงหน้าจอฐานข้อมูลของคณะภายในสถาบันการศึกษา

จากรูปที่ ก.17 แสดงหน้าจอฐานข้อมูลคณะภายในสถาบันการศึกษา หน้าจอแสดงคณะอยู่ภายในเมนูของฐานข้อมูล หน้าจอฐานข้อมูลคณะประกอบด้วยปุ่ม 4 ปุ่ม คือ เพิ่มข้อมูล แก้ไขข้อมูล ลบข้อมูล และ ออกจากหน้าจอฐานข้อมูลคณะ หน้าจอการเพิ่มและแก้ไขข้อมูลแสดงดังรูปที่ ก.18 และ รูปที่ ก.19 ตามลำดับ



รูปที่ ก.18 แสดงหน้าจอเพิ่มข้อมูลคณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.19 แสดงหน้าจอการแก้ไขข้อมูลคณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
บทความที่ตีพิมพ์ในวารสาร

พัฒนพงษ์ ฉันทมิตร โอภาสและเอื้อน ปิ่นเงิน. “แบบจำลองน้ำพุเพื่อการพัฒนาระบบเชิงวัตถุ.” วารสารพระจอมเกล้าลาดกระบัง, ปีที่7, ฉบับที่ 1 มีนาคม 2542.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วารสารพระจอมเกล้าลาดกระบัง

JOURNAL OF KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ISSN 0858-8430 Vol.7 No.1 March 1999 ปีที่ 7 ฉบับที่ 1 มีนาคม 2542 **2542**

บทความวิจัย

เครื่องวัดปริมาณน้ำฝนส่งข้อมูลผ่านคลื่นวิทยุ 1
พีระวุฒิ สุวรรณจันทร์

การตอบสนองของใบผักกาดหัวต่อการพรางแสง 9
วิรัตน์ ภูวิวัฒน์

การประเมินประสิทธิภาพระบบพัฒนาทรัพยากรมนุษย์ด้วยสื่อประสม 20
โอฬาร วงศ์วิรัตน์ สุรสิทธิ์ ราตรี Seiichi Abe

การประยุกต์ใช้งานของ LIDAR เพื่อการประมาณค่าระดับความสูงและความหนาของโทรโพพอส 28
พีระวุฒิ สุวรรณจันทร์

บทความวิชาการ

เรื่องน่ารู้บางประการเกี่ยวกับถั่วแดงหลวงในประเทศไทย 35
ทิพวรรณ ลิมังกูร

การแก้ปัญหาการวิเคราะห์ผลตอบเมื่อค่าความคลาดเคลื่อนมีการแจกแจงแบบไม่ปกติ 40
ชานินทร์ ศรสุวรรณนา

แบบจำลองน้ำท่าเพื่อการพัฒนาระบบเชิงวัตถุ 48
พิสุทธางษ์ จันทร์ตาไชยวาส อร้อย ปันเรือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบจำลองน้ำพุเพื่อการพัฒนาระบบเชิงวัตถุ

FOUNTAIN MODEL FOR OBJECT-ORIENTED SYSTEM DEVELOPMENT

พัฒนพงษ์ ฉันทมิตรโอภาส* เอื้อน ปิ่นเงิน**

บทคัดย่อ

รูปแบบการพัฒนาระบบสารสนเทศที่นิยมใช้กันในอดีตจนถึงปัจจุบันคือ Waterfall Model เมื่อระบบงานมีความสลับซับซ้อนมากขึ้นจึงได้มีการนำเสนอรูปแบบการพัฒนาเชิงวัตถุเพื่อให้ได้ระบบที่มีประสิทธิภาพมากขึ้น บทความนี้เป็นการนำเสนอแบบจำลองน้ำพุที่เหมาะกับเทคโนโลยีเชิงวัตถุ ซึ่งแบ่งขั้นตอนการพัฒนาออกเป็น 10 ขั้นตอน แต่ละขั้นตอนมีลักษณะการทำซ้ำและการเวียนบังเกิดหรือนำกลับมาใช้ใหม่ ระบบงานที่ได้จากการพัฒนาแบบเชิงวัตถุจะเป็นระบบที่สามารถเปลี่ยนแปลงได้ง่ายและมีความน่าเชื่อถือสูง

Abstract

The Waterfall Model has been extensively used for information system development for many years. When the application architecture are more complicated, an Object-Oriented Approach has been introduced. This paper presents a Fountain Model which is suitable for Object-Oriented Technology. The Fountain Model divides a development process into 10 stages. Each stage employs iteration, recursion and reusability. The systems developed via Object-Oriented Approach are highly promising to achieve reusability and reliability.

* นักศึกษาปริญญาโท คณะเทคโนโลยีสารสนเทศ สถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

** ผู้ช่วยศาสตราจารย์ ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

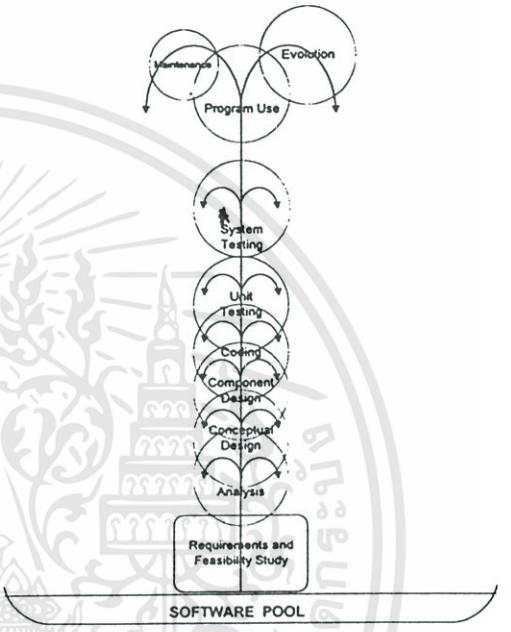
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บทนำ

การพัฒนาระบบสารสนเทศเชิงโครงสร้างที่ใช้แบบจำลองน้ำตก (Waterfall Model) เริ่มมีข้อจำกัดเมื่อระบบมีความซับซ้อนและมีการเปลี่ยนแปลงบ่อยครั้งขึ้น ต่อมาจึงได้มีการคิดค้นวิธีการพัฒนาระบบสารสนเทศเชิงวัตถุขึ้นมาเพื่อแก้ไขปัญหาดังกล่าว เนื่องจากวิธีการนี้มีอยู่ด้วยกันหลายแบบ บทความนี้จึงขอนำเสนอแบบจำลองน้ำพุ [2] ซึ่งสามารถนำไปประยุกต์ใช้กับวิธีเชิงวัตถุได้ เพราะแบบจำลองนี้มีลักษณะการวิเคราะห์แบบบนลงล่าง และมีการออกแบบจากล่างขึ้นบน [7] ซึ่งภายในใช้รูปแบบของ OMT [5] ที่ผสมผสานแนวคิดระหว่างเชิงวัตถุกับเชิงโครงสร้าง [8] ส่วนลักษณะเฉพาะของแบบจำลองน้ำพุ คือ มีการทำซ้ำ (iterative) และการเวียนบังเกิด(recursive) ของแบบจำลองน้ำพุ เป็นรูปแบบการพัฒนาระบบสารสนเทศที่มีความยืดหยุ่นและครอบคลุมทั้งการวิเคราะห์ การออกแบบ จึงทำให้มีลักษณะของการนำกลับมาใช้ใหม่ได้ แม้ว่ากิจกรรมบางกิจกรรมในการพัฒนาระบบตามแบบจำลองนี้ไม่สามารถเริ่มก่อนกิจกรรมหนึ่งได้ เช่น ผู้พัฒนาไม่สามารถเขียนโปรแกรมก่อนการออกแบบหรือไม่สามารถออกแบบก่อนศึกษาถึงความต้องการของผู้ใช้ได้ เป็นต้น (รูปที่ 1) แต่การพัฒนาระบบโดยใช้แบบจำลองน้ำพุจะยังมีกิจกรรมที่มีการคาบเกี่ยวกัน และรวมกันของกิจกรรมเกือบตลอดทั้งวงจรการพัฒนาในระบบแบบเชิงวัตถุ[6]

แบบจำลองน้ำพุมีรูปแบบการพัฒนาที่เปรียบได้กับน้ำพุที่พุ่งคือเส้นทางแสดงการพัฒนาระบบที่พุ่งขึ้นจากส่วนกลางและไหลลงสู่สระเบื้องล่างเพื่อสามารถนำกลับมาใช้ใหม่ได้ (reusability) โดยทั่วไปการพัฒนาระบบจะเริ่มจากการวิเคราะห์ การออกแบบ และการอิมพลีเมนต์ (implementation) ซึ่งสอดคล้องกับระบบเชิงวัตถุคือจะครอบคลุมสองในสามส่วน ของวงจรการพัฒนาระบบ (หรือทั้งหมด) ขณะที่สระน้ำคือคลังของคลาสหรือที่เก็บซอฟต์แวร์ส่วนกลาง (software pool หรือ repository)

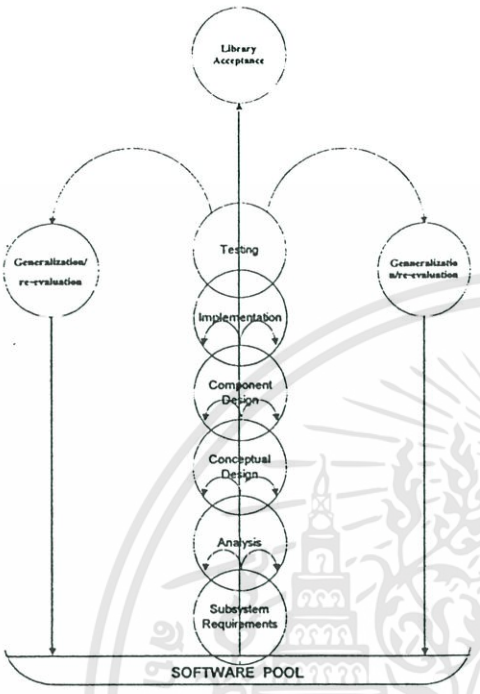
รูปที่ 1 แสดงถึงลักษณะทั่วไปในระดับของระบบ (systems level) ของการพัฒนาเชิงวัตถุซึ่งมีการรวมกันในระดับสูงของการวิเคราะห์ ออกแบบ การอิมพลีเมนต์ และขั้นตอนการทดสอบ



รูปที่ 1 รูปแบบทั่วไปของแบบจำลองน้ำพุที่นำไปพัฒนาระบบสารสนเทศแบบเชิงวัตถุ

การพัฒนาระบบสารสนเทศเชิงวัตถุสามารถแบ่งเป็นกลุ่มย่อย (subsystem หรือ cluster) [1] โดยแต่ละกลุ่มย่อยคือกลุ่มของงานที่ทำงานใกล้ชิดกันคือมีระดับความเกี่ยวเนื่องระหว่างกันสูงแต่มีความเกี่ยวเนื่องระหว่างกลุ่มย่อยนั้นต่ำกว่า แต่ละกลุ่มย่อยถูกกำหนดจากส่วนการวิเคราะห์และออกแบบในระดับของระบบ (systems level) ส่วนขั้นตอนการพัฒนาในแต่ละคลาสของกลุ่มย่อยสามารถทำลักษณะขนานกันได้แม้ว่าวงจรชีวิตการพัฒนาในแต่ละกลุ่มย่อยมีความแตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 ขั้นตอนการพัฒนาของแต่ละระบบย่อยของแบบจำลองน้ำพุ

รูปที่ 2 แสดงถึงรายละเอียดของแต่ละกลุ่มย่อยซึ่งกำหนดให้วงกลมแต่ละวง (bubble)แทนขั้นตอนต่าง ๆ ของกลุ่มย่อย ส่วนนี้จะเน้นในเรื่องของกลุ่มย่อยมากกว่าระดับของระบบ(สังเกตจากรูปที่ 1)และจากรูปพิจารณาว่าการคาบเกี่ยวกันของขั้นตอนจะน้อยลง แต่ลักษณะการทำซ้ำยังคงเป็นส่วนที่สำคัญและเห็นได้ชัดเจนในวงจรชีวิต ดังนั้นกลุ่มย่อยที่พัฒนาควรทำให้อยู่ในรูปทั่วไป (Generalization) เพื่อสามารถนำกลับมาใช้ใหม่ได้

ในขั้นตอนการวิเคราะห์จะเป็นการแสดงความเป็นจริง ในขณะที่ขั้นตอนการออกแบบจะแสดงด้วยการออกแบบระบบสารสนเทศ จุดประสงค์หลักของการออกแบบเชิงวัตถุคือ เพื่อช่วยสนับสนุนการออกแบบเชิงวิศวกรรมซอฟต์แวร์ที่ดีในแง่ ความถูกต้อง (correctness) การเป็นโมดุล (modularity) ความสามารถในการนำกลับมาใช้ใหม่(reusability)และ abstraction เป็นต้น และลดข้อจำกัดทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่มี ดังนั้นการออกแบบจึงมุ่งประเด็นไปที่คำว่า "ทำอย่างไร" มากกว่าพยายามทำให้รอยต่อของทั้งการวิเคราะห์และการออกแบบเชิงวัตถุมีความเชื่อมกันอย่างสนิทมากยิ่งขึ้น

โดยปกติขั้นตอนและรูปแบบของ SDLC ที่มีลักษณะเชิงโครงสร้างจะมองไม่เห็นสองขั้นตอนคือ ขั้นตอน generalization ที่ประกอบด้วยการรวมกลุ่ม (clustering) และการประเมินผลแต่ละกลุ่มย่อย และขั้นตอนการนำคลาสเข้าสู่คลังเพื่อสามารถนำกลับมาใช้ใหม่ (ดังแสดงในรูปที่ 2) แม้ว่าการพัฒนากลุ่มย่อยให้อยู่ในรูปแบบทั่วไปและนำกลับมาใช้ใหม่ได้ มักเกิดขึ้นหลังจากสิ้นสุดกระบวนการพัฒนา แต่ในความเป็นจริงการพัฒนาคลาสหรือกลุ่มย่อยเหล่านี้ยังขึ้นอยู่กับประสบการณ์ของผู้พัฒนาระบบ ซึ่งสามารถกำหนดคลาสเหล่านั้นได้ตั้งแต่ช่วงแรกของวงจรการพัฒนา

2. ขั้นตอนการพัฒนาระบบภายใต้แบบจำลองน้ำพุ

การพัฒนาระบบแบบเชิงวัตถุควรพัฒนาที่ระดับของระบบและระบบย่อย ณ เวลาเดียวกันหรือคล้ายกัน ดังนั้นจึงต้องกำหนดบางโปรเซสที่ระดับของระบบและบางโปรเซสที่ระบบย่อย(ดังแสดงในตารางที่ 1) โดยระดับของระบบมีความสัมพันธ์กับความต้องการของผู้ใช้และรายละเอียดของปัญหาเพื่อนำมาตั้งสมมุติฐานให้ตรงกับปัญหาที่เกิดขึ้น ขั้นตอนของการศึกษาความเหมาะสมเป็นตัวอย่างที่เห็นได้ชัดเจนมาก เช่น การประเมินแนวทางการแก้ไขปัญหา ปรับปรุงระบบที่มีอยู่เดิม สร้างระบบใหม่ทดแทนระบบเก่า หรืออาจไม่จำเป็นต้องทำสิ่งใดกับระบบเดิมเลย เป็นต้น ในกรณีนี้ขั้นตอนที่ 1 ไม่มีผลกระทบกับรายละเอียดต่าง ๆ ของวงจรชีวิตแบบเชิงวัตถุ (แต่มีความสำคัญกับขั้นตอนที่ 3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 แสดงขั้นตอนทั้ง 10 ขั้นตอนของ แบบจำลองน้ำพุ ในรูปที่ 1 และ 2

ขั้นตอนที่	S/ss	รายละเอียด
1	S	ศึกษาความเหมาะสมและความต้องการของผู้ใช้ (User Requirements and Feasibility Study)
2	S	กำหนดซอฟต์แวร์ตามความต้องการของผู้ใช้ (Software User Requirements)
3	S/ss	การวิเคราะห์ (Analysis)
4	Ss	การออกแบบระดับแนวคิด (Conceptual Design)
5	Ss	การออกแบบองค์ประกอบ (Component Design)
6	Ss	การอิมพลีเมนต์และทดสอบ (Implementation and Testing)
7	S	การรวบรวมระบบย่อยและทดสอบระบบหลัก(Subsystem Integration and System Testing)
8	Ss	การวางหลักเกณฑ์และการประเมินคุณภาพ (Generalization and Re-evaluation)
9	S	การส่งมอบและการทดสอบเพื่อตอบรับ (Program Delivery/Acceptance Testing)
10	S/ss	การบำรุงรักษาและการเปลี่ยนแปลงระบบ (Evolution and Maintenance)

หมายเหตุ S = System Lifecycle Phase; ss = Subsystem Lifecycle Phase

การทำซ้ำระหว่างกิจกรรมนับว่ามีความสำคัญมากกับการพัฒนาหรือเปลี่ยนแปลงของระบบที่ใช้งานอยู่ ดังนั้นวิธีนี้จึงสนับสนุนการทำงานที่มีลักษณะทำซ้ำกันของโปรเซสและการเกิดขึ้นหรือนำมาใช้ใหม่ เพราะขั้นตอนที่เกิดขึ้นสามารถถูกนำกลับมาพัฒนาต่อและยังสามารถพัฒนาแบบขนานได้ คือ ขั้นตอนที่แตกต่างกันของระบบสามารถที่จะกระทำได้ ณ เวลาเดียวกันหรือพัฒนาไปพร้อมๆ กันได้ แม้ว่าจะต่างโปรเซสกันในแต่ละขั้นตอนก็ตามส่วนขั้นตอนการบริหารโครงงานมีลักษณะคล้ายกับการเริ่มต้นกระทำระหว่างการทำซ้ำและการเวียนบังเกิดอีกครั้ง เพื่อนำมาลำดับเหตุการณ์ที่เกิดขึ้นหรือหาจุดสิ้นสุด(break-points)ของเอกสารที่ได้รับมา เช่น ทำความเข้าใจกับความต้องการของผู้ใช้ให้ชัดเจนขึ้น เป็นต้น และทำให้เอกสารที่ได้จากการทำซ้ำของขั้นตอนต่าง ๆ มีความเหมาะสมที่จะนำมากำหนดจุดสิ้นสุดเพื่อให้ได้เอกสารที่เป็นชุดปัจจุบันของขั้นตอนนั้น โดยแต่ละระบบที่พัฒนาให้กับองค์ประกอบนั้นย่อมไม่เหมือนกัน จึงทำให้พิจารณาได้ยากว่าเมื่อไรจึงจะหยุดการทำซ้ำหรือการเวียนบังเกิดของเอกสารต่าง ๆ ดังนั้นควรมีการควบคุมเอกสาร (Version Control) ต่าง ๆ เหล่านี้ในแต่ละขั้นตอนของวงจรชีวิตที่พัฒนา เช่น กลุ่มย่อยที่ n ของขั้นตอนที่ 5 อาจนำเอกสารที่ได้มาทำซ้ำกันถึง 3 ครั้งจึงได้เอกสารที่ชัดเจนและตรงความต้องการ เป็นต้น ดังนั้นคำว่า "Version Control" เป็นส่วนประกอบของการทำซ้ำในกระบวนการพัฒนาที่นับว่ามีความสำคัญมาก

ส่วนต่อไปจะแสดงถึงรายละเอียดทั้ง 10 ขั้นตอนของการพัฒนาตามแบบจำลองน้ำพุ

ขั้นตอนที่ 1: ศึกษาความเหมาะสมและความต้องการของผู้ใช้

การสอบถามความต้องการของผู้ใช้เป็นทางหนึ่งที่ทำให้ทราบถึงความต้องการของผู้ใช้ได้ชัดเจนมากขึ้น ส่วน Feasibility คือการดำเนินการค้นหาความเหมาะสม เช่น ความเหมาะสมทางเทคโนโลยีกับค่าใช้จ่ายที่เกิดขึ้น ค่าบำรุงรักษา ค่าใช้จ่ายทางด้านการฝึกอบรม เป็นต้น ส่วนการบริหารโครงงานแบบวิศวกรรมซอฟต์แวร์การประเมินค่าใช้จ่ายของโครงงานนับว่ามีความจำเป็นกับโปรเซสและผลิตภัณฑ์ที่พัฒนา ซึ่งค่าใช้จ่ายที่ประเมินนี้จะเป็นส่วนหนึ่งของขั้นตอนที่ 1 และอาจจะอยู่ในขั้นตอนที่ 2 ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2: กำหนดซอฟต์แวร์ตามความต้องการของผู้ใช้

ในขั้นตอนที่ 2 แตกต่างจากขั้นตอนที่ 1 อยู่ 2 ประเด็นหลักคือ การตัดสินใจเลือกซอฟต์แวร์ให้เหมาะสมกับระบบถูกเลือก ณ ขั้นตอนนี้และความต้องการของผู้ใช้จะมีรายละเอียดมากขึ้นเพื่อแสดงให้เห็นถึงความต้องการที่แท้จริงที่มีต่อระบบมากกว่า ขั้นตอนก่อน แสดงว่าความแตกต่างระหว่างขั้นตอน 1 กับ 2 คือแนวทางการแก้ปัญหามีความเชื่อถือมากขึ้นกว่าขั้นตอนที่ 1 และจากความต้องการของผู้ใช้ที่ได้จะรวมถึง screen layout, respond time เป็นต้น ส่วนประเด็นหลักของขั้นตอนนี้เจาะจงในเรื่องแนวทางการแก้ไขปัญหาให้ตรงกับปัญหาที่เกิดขึ้นให้มากที่สุด

ขั้นตอนที่ 3: การวิเคราะห์

การวิเคราะห์นับว่าเป็นแบบแผนที่ดีและขั้นตอนนี้มุ่งประเด็นทั้งหมดไปยังเอกภพของปัญหามากกว่าที่จะเป็นแนวทางการแก้ปัญหา โดยการวิเคราะห์จะเริ่มทำกับทุกระบบย่อยเพื่อให้ได้เอกสารต่าง ๆ ที่ตรงและเพียงพอกับระบบที่ต้องพัฒนา การวิเคราะห์เป็นส่วนที่แสดงถึงรายละเอียดของรูปแบบเชิงวัตถุในทาง เอกภพของปัญหา(ผลที่ได้จากการวิเคราะห์คือรายละเอียดของรูปแบบเชิงวัตถุที่มีต่อเอกภพของปัญหา)

ขั้นตอนที่ 4: การออกแบบระดับแนวคิด

เป็นการย้ายจากเอกภพของปัญหาไปสู่แนวทางแก้ไข ซึ่งโครงสร้างของการออกแบบระดับแนวคิดยังมีลักษณะของแบบจำลองระดับแนวคิดอยู่ (คล้ายกับการวิเคราะห์)แต่สิ่งที่ต้องควรคำนึงถึงในการออกแบบคลาส คือต้องอยู่ภายในขอบเขตของปัญหา (Solution-Domain) หรือสามารถทำงานร่วมกับระบบที่ออกแบบได้ (Computer-Oriented) เพราะขั้นตอนนี้นับว่าเป็นตัวแปรที่สำคัญของแบบจำลองน้ำพุซึ่งคลาสเหล่านี้สามารถนำกลับไปใช้กับระบบอื่นๆ ได้

การออกแบบระดับแนวคิดคือการนำเอกสารที่ได้จากขั้นตอนการวิเคราะห์และขั้นตอนซอฟต์แวร์มาดำเนินตามกิจกรรมที่กำหนด โดยพิจารณากับฮาร์ดแวร์และซอฟต์แวร์ลักษณะหนึ่งต่อหนึ่ง เพื่อให้ได้เอนทิตี้ (Entity) ตรงตามความเป็นจริงมากที่สุด (แสดงดังรูปที่ 3 และ 4) ซึ่งขั้นตอนการออกแบบระดับแนวคิดมีลักษณะคล้ายกับขั้นตอนการวิเคราะห์ แต่ยกเว้นในเรื่องของ ความต้องการทางด้านความสอดคล้องหรือการตอบสนองกับคอมพิวเตอร์ ซึ่งลักษณะดังกล่าวนั้นเกิดจากการสอบถามหรือพูดคุยกับลูกค้า/ผู้ใช้เพื่อมากำหนดความต้องการ ดังนั้นจึงเหมือนกับเข้าไปทำซ้ำในส่วนของการสอบถามความต้องการของผู้ใช้อีกครั้งและทำให้เป็นที่มาของความสอดคล้องกับคอมพิวเตอร์

ขั้นตอนที่ 5: การออกแบบองค์ประกอบ

ตอนนี้กล่าวได้ว่าเข้ามาสู่ขั้นตอนการออกแบบองค์ประกอบซึ่งเป็นขั้นตอนที่ 5 และขั้นตอนนี้เป็นการย้ายจาก "สถาปัตยกรรม" ของขั้นตอนที่ 4 ไปสู่การออกแบบคลาสให้มากขึ้น โดยจะพิจารณาออกแบบในแต่ละคลาส

แม้ว่าขั้นตอนที่ 4 และ 5 มีลักษณะที่คล้ายกันก็ตามแต่ในแบบจำลองน้ำพุพบว่าทั้ง 2 ขั้นตอนมีลักษณะแตกต่างกันคือสามารถที่จะเปลี่ยนแปลงเทคนิก (ลักษณะ) จากระดับของระบบไปสู่ระดับคลาสได้บ่อยขึ้นและถ้าเป็นโครงงานหรือโครงการใหญ่ ๆ จะช่วยในเรื่องของการจัดการหรือการบริหารงานให้ง่ายขึ้นในกรณีการส่งมอบและจัดการเกี่ยวกับเอกสาร อย่างไรก็ดีถ้ามีทีมงานที่เล็กหรือมีบุคคลที่ทำตั้งแต่วิเคราะห์ ออกแบบ และเขียนโปรแกรม ก็ไม่สามารถที่จะแยกแยะความแตกต่างระหว่างการออกแบบระดับแนวคิดกับการออกแบบองค์ประกอบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของคลาสที่ถูกออกแบบประกอบด้วยรายละเอียดของ Interface และกระบวนการทำงานภายใน (Internal Working) ซึ่งรายละเอียดของความต้องการมากเท่าไรก็ย่อมเป็นผลดีต่อการออกแบบคลาสในระดับล่างของ abstraction จึงทำให้คลาสนั้นมีความสมบูรณ์มากขึ้น นอกจากนี้ยังทำให้คลาสนั้นง่ายต่อการวิเคราะห์หรือการออกแบบระดับแนวคิด

ขั้นตอนที่ 6: การอิมพลีเมนต์และทดสอบ

นำรายละเอียดจากขั้นตอนต่าง ๆ มาพัฒนาโดยเขียนโปรแกรมให้กับแต่ละระบบย่อย หลังจากนั้นควรทดสอบระบบย่อยที่ได้พัฒนาขึ้น การทดสอบมีด้วยกันหลายวิธีแต่แบบจำลองน้ำพุใช้วิธีทดสอบแบบ White-Box และ Black-Box ซึ่งการทดสอบแบบ White-Box (structural testing) คือ การทดสอบในแต่ละโมดุลอย่างละเอียดตั้งแต่คำสั่ง (statements) แรกจนไปถึงคำสั่งสุดท้ายของโมดุลหรือทดสอบอัลกอริทึม ส่วน Black-Box คือ การทดสอบที่เน้นไปทางผลลัพธ์ให้ตรงกับความต้องการของผู้ใช้เป็นสำคัญ เช่น respond time, interface เป็นต้น โดยไม่คำนึงถึงรายละเอียดของโมดุล[4]

ขั้นตอนที่ 7: การรวบรวมระบบย่อยและทดสอบระบบหลัก

เมื่อแต่ละระบบย่อยผ่านการทดสอบเรียบร้อยแล้ว คลาสต่าง ๆ จะถูกเก็บเข้าคลังข้อมูลของโครงงาน (Project Library) เพื่อนำมาใช้อีกครั้ง ในกรณีที่โครงงานที่พัฒนามีขนาดกลางจนถึงขนาดใหญ่ ระบบย่อยต่าง ๆ จะถูกรวมเข้าด้วยกันเพื่อทดสอบให้แน่ใจว่าแต่ละระบบสามารถทำงานรวมกันได้

การทดสอบระบบของขั้นตอนนี้มีความใกล้เคียงกับขั้นตอนการทดสอบที่กล่าวมาแล้ว โดยขั้นตอนดังกล่าวเป็นการทดสอบแบบโครงสร้างของโครงงานแต่ขั้นตอนนี้คือทดสอบถึงความสามารถโดยรวมของระบบ ลักษณะทั่วไปของคลาสต่าง ๆ ของระบบเชิงวัตถุที่ทำการทดสอบในขั้นตอนที่ 6 เป็นการทดสอบแบบ "Alpha test" คือทดสอบแบบโครงสร้างมากกว่าทางด้านกายภาพ ดังนั้นขั้นตอนที่ 7 จะทดสอบเพื่อแสดงว่าแต่ละคลาสสามารถทำงานร่วมกันได้เนื่องจากว่าผ่านการทดสอบแล้วและถูกออกแบบมาเป็นอย่างดี ซึ่งการทดสอบนี้เรียกว่า "Beta Testing" คือนำเอางานของแต่ละทีมงานมารวมกันเพื่อทดสอบระบบถึงรายละเอียดต่าง ๆ ให้ตรงตามความต้องการก่อนที่จะนำไปใช้งานจริง

ขั้นตอนที่ 8: การวางหลักเกณฑ์และการประเมินคุณภาพ

จุดประสงค์หลักของขั้นตอนนี้คือการปรับปรุงคุณภาพของคลาสในโครงงานซึ่งสามารถทำต่างหากจากโครงงาน เช่น ทำการประเมินลำดับชั้นต่าง ๆ ของคลาส สร้างคลาสขึ้นมาใหม่ หรือปรับปรุงคลาสให้มีประสิทธิภาพมากขึ้นและหรือน้อยลง เป็นต้น

ส่วนด้านการบริหารโครงงานเน้นทางด้านการพัฒนาระบบงานให้มีความสมบูรณ์มากขึ้นเช่น ความสัมพันธ์ต่าง ๆ ของระบบที่มีต่อผู้ใช้ กฎเกณฑ์ทางธุรกิจและตรวจสอบถึงคุณภาพของคลาสภายในคลังให้เป็นที่ยอมรับได้ เป็นต้น

ขั้นตอนที่ 9: การส่งมอบและการทดสอบเพื่อตอบรับ

คือส่งมอบระบบที่พัฒนาขึ้นและนำไปใช้งาน แต่ระบบที่ส่งมอบนั้นผู้ใช้ยังไม่สามารถนำไปใช้งานทันทีเพราะต้องผ่านการทดสอบและเป็นที่ยอมรับเสียก่อนว่าควรที่จะยอมรับระบบและนำไปใช้งานหรือไม่ เนื่องจากสภาวะแวดล้อมแบบเชิงวัตถุนั้นเป็นเทคโนโลยีที่ใหม่ และทำให้การทดสอบนี้เป็นขั้นตอนการทดสอบขั้นสุดท้ายที่กระทำโดยผู้ใช้หรือลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 10: การบำรุงรักษาและการเปลี่ยนแปลงระบบ

การบำรุงรักษาและการประเมินระบบนั้นจะมุ่งเน้นไปทางระดับของระบบ(ดังแสดงตามรูปที่ 1) มากกว่าระบบย่อยหรือคลาส แม้ว่าการเกิดความเปลี่ยนแปลงที่เกิดขึ้นจากความต้องการของผู้ใช้จะเป็นเพียงส่วนหนึ่งของระบบเท่านั้นแต่ทำให้ย้อนกลับไปสู่ระดับคลาสและเริ่มทำให้กิจกรรมต่าง ๆ ในวงจรชีวิตของแบบจำลองน้ำพุสามารถเกิดขึ้นใหม่หรือนำกลับมาใช้อีกครั้ง

ทางปฏิบัติผู้พัฒนาสามารถที่จะบำรุงรักษาระบบไปพร้อม ๆ กันเพื่อให้สัมพันธ์กับผู้ใช้ซึ่งทำให้ระบบสามารถพัฒนาได้รวดเร็วและมากขึ้น จากกลุ่มของเมตริกซ์ที่นำไปสู่การพัฒนาโปรเซสนั้นเป็นตัวบ่งชี้ที่สำคัญที่แสดงให้เห็นถึงความยืดหยุ่นของการพัฒนากลุ่มซอฟต์แวร์ ดังนั้นการวางแผนพัฒนาระบบแบบเชิงวัตถุจึงมีประโยชน์กับเทคโนโลยีแบบเชิงวัตถุเพราะสามารถตอบสนองความต้องการของผู้ใช้ได้รวดเร็วและมีประสิทธิมากขึ้นด้วย

3. แบบจำลองน้ำพุสำหรับการนำกลับมาใช้ใหม่

ลักษณะการเกิดขึ้นใหม่ของระบบในวงจรชีวิตการพัฒนาซอฟต์แวร์มีอยู่ 3 ลักษณะคือ การเกิดขึ้นใหม่ของระบบย่อย นำเพียงบางส่วนมาปรับปรุงและ/หรือนำกลับมาใช้ใหม่ การนำกลับมาใช้ใหม่ทั้งหมด เป็นต้น แม้ว่าการเกิดขึ้นใหม่ของระบบย่อยเป็นความต้องการเพียงบางส่วนแต่มีความแตกต่างจากขั้นตอนของวงจรชีวิต ส่วนขั้นตอนการพัฒนาของระบบย่อยทั้ง 3 มีดังนี้

- ถ้าเป็นการพัฒนาระบบย่อยใหม่ทั้งหมด จะเริ่มต้นตั้งแต่ขั้นตอนที่ 1 – 10 ที่กล่าวไว้แล้ว (ดังตารางที่ 1 และรูปที่ 2)
- การนำกลับมาใช้ใหม่หรือปรับปรุงเพียงบางส่วนขั้นตอนต่าง ๆ ของวงจรชีวิตที่กระทำกับคลังของคลาสมีหลายขั้นตอน แต่ขั้นตอนที่ดูจะมีความสำคัญน้อยลงไปคือขั้นตอนการออกแบบทั้ง 2 ขั้นตอนและขั้นตอนที่ 8 (แสดงดังรูปที่ 3)

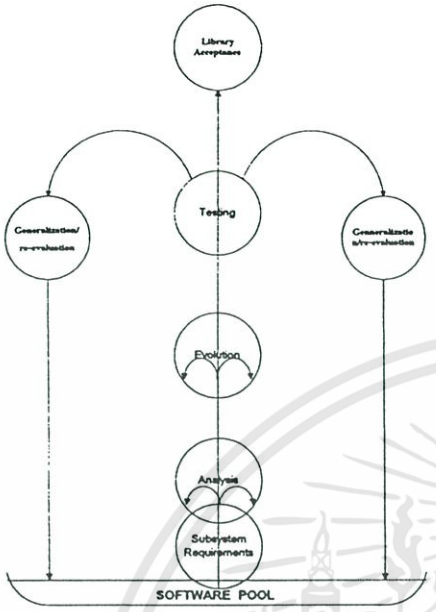
การนำคลาสทั้งหมดกลับมาใช้ใหม่นั้นควรกระทำทุกขั้นตอน ยกเว้นเพียงขั้นตอนที่ 4-6 (แสดงดังรูปที่ 4)

ลักษณะการสร้างระบบย่อยของแบบจำลองน้ำพุที่เกิดขึ้นใหม่ต้นทุนส่วนใหญ่จะอยู่ในรูปของเวลาและทรัพยากรที่ใช้ เช่น ความต้องการของผู้ใช้ การวิเคราะห์ การออกแบบเป็นต้น ผลของการสร้างทำให้จำนวนของคลาสที่นำไปใช้กับโครงงานต่อ ๆ ไปลดลงไปได้มากเปรียบเสมือนกับสร้างรอไว้แล้ว ส่วนขั้นตอนการพัฒนาของระบบย่อยยังคงต้องยึดถือกิจกรรมที่กระทำในวงจรชีวิตเป็นสำคัญ (ดังแสดงรูปที่ 2)

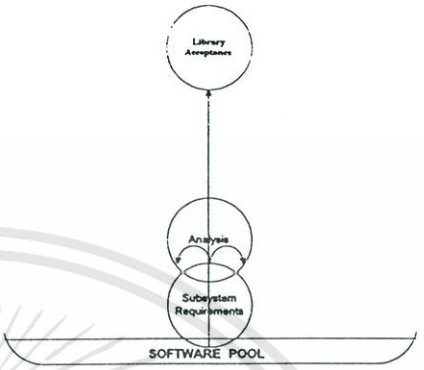
การนำคลาสจากคลังมาพัฒนาหรือปรับปรุงกิจกรรมของการพัฒนาถือเป็นขั้นตอนแรกที่ต้องนำมาพิจารณาภายในคลังของคลาสและควรพัฒนาตามขั้นตอน คลาสที่ถูกเลือกมาพัฒนาหรือปรับปรุงสามารถลดต้นทุนด้านเวลาและยังทำให้ระบบที่นำไปใช้มีความมั่นคงขึ้น ซึ่งการพัฒนาหรือปรับปรุงคลาสจะเกิดขึ้นบ่อยแค่ไหนขึ้นอยู่กับว่าระบบที่พัฒนานำไปใช้บ่อยแค่ไหนด้วย (แสดงดังรูปที่ 3)

การนำคลาสกลับมาใช้ใหม่ของ Application ถัดมามีโอกาสเป็นไปได้มาก ซึ่งอาจไม่ต้องใช้ทั้งหมด อย่างไรก็ตามควรวิเคราะห์และพิจารณาถึงคุณลักษณะทั่วไปของคลาสก่อน เพราะขอบเขตของปัญหายังคงกำหนดกลุ่มของคลาสในคลังที่จะนำกลับมาใช้ใหม่และคลาสที่นำมาใช้จึงไม่ต้องผ่านขั้นตอนการออกแบบ ซึ่งการนำกลุ่มของคลาสกลับมาใช้ใหม่ทำให้ต้องนำคลาสเหล่านี้มาจัดการอีกครั้งหนึ่ง จึงควรตรวจสอบคลาสที่จะนำมาใช้ใหม่ด้วยกิจกรรมเพื่อให้ได้คลาสที่เหมาะสมและตรงกับความต้องการมากที่สุด(แสดงดังรูปที่ 4)

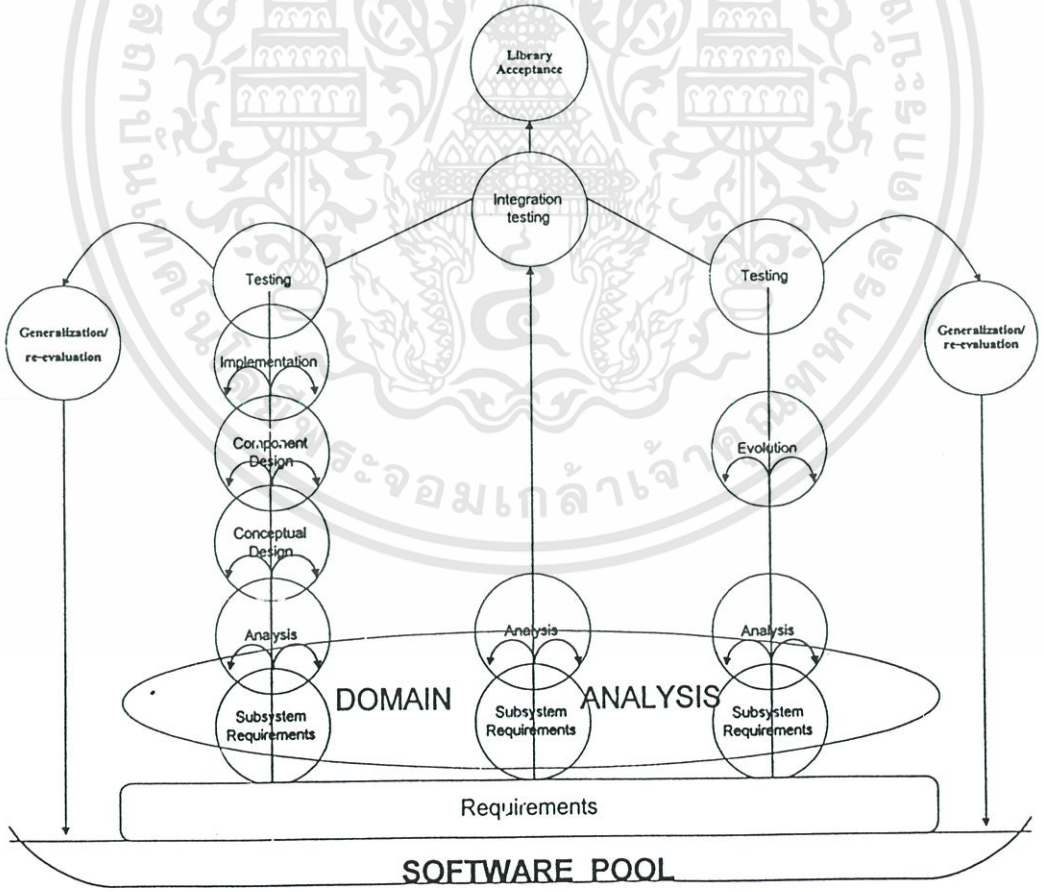
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 ขั้นตอนการนำคลาสจากคลังกลับมาปรับปรุง



รูปที่ 4 ขั้นตอนการนำคลาสหรือระบบย่อยในคลังกลับมาใช้ใหม่โดยตรง



รูปที่ 5 ความสัมพันธ์ของแต่ละระบบย่อยที่แตกต่างกันภายใต้การวิเคราะห์เดียวกัน

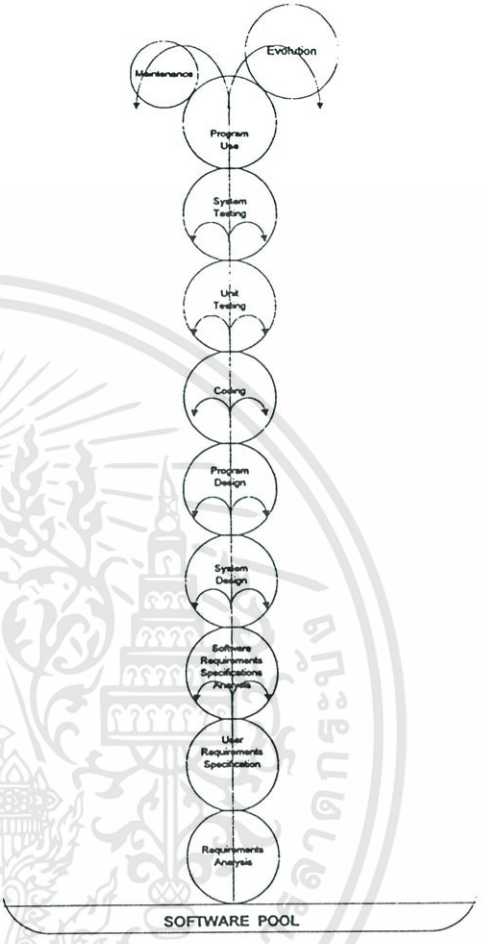
วารสารพระจอมเกล้าลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าคลาสที่ถูกนำไปประยุกต์สามารถเข้ากับวงจรของ Application ได้ ขั้นตอนการวิเคราะห์จะนำคลาสเหล่านี้ไปรวมกับ Application และทำให้สอดคล้องกันภายในวงชีวิตของระบบ ซึ่งระบบที่ใช้การวิเคราะห์แบบนี้ทำให้สิ่งแรกที่ได้จากการวิเคราะห์คือ มีความเป็น Generalized และเอกสารต่าง ๆ สามารถนำไปใช้กับการพัฒนาโครงการต่อไปได้แนวคิดของแบบจำลองน้ำพุมีลักษณะการใช้งานที่คล้ายกันกับที่อธิบายไว้ในวงจรชีวิตแบบเชิงวัตถุ [3] ซึ่งเป็นการรวบรวมของระบบย่อย ขอบเขตการวิเคราะห์ การนำกลับมาใช้ใหม่ ซอฟต์แวร์ขั้นพื้นฐาน การทำซ้ำและการเวียนบังเกิดของวงจรชีวิตในระบบย่อยของแบบจำลองน้ำพุที่แสดงในรูปที่ 5 พบว่าระบบย่อยทั้งสามมีความแตกต่างกันคือมีการเพิ่มเติมระบบที่จำเป็นและนำบางส่วนมาปรับปรุง ซึ่งทั้งหมดนำมาจากคลังของคลาสทั้งสิ้น

สำหรับการจัดการกับซอฟต์แวร์ขนาดใหญ่(ระบบ) จำนวนของน้ำพุอาจจะแสดงถึงความแตกต่างของโครงการที่พัฒนา

4. แบบจำลองน้ำพุสำหรับ NON-O-O SDLC

รูปแบบของแบบจำลองน้ำพุสามารถนำไปประยุกต์กับวงจรการพัฒนาระบบที่ไม่เป็นเชิงวัตถุได้ ซึ่งวิธีการทำจะทำตามขั้นตอนของวงจรชีวิต แต่บางขั้นตอนที่คาบเกี่ยวกันอาจทำให้คู่ไม่ชัดเจนสำหรับการพัฒนาแบบปกติทั่ว ๆ ไป ดังนั้นความสำคัญของวงจรชีวิตแบบเชิงวัตถุจึงไม่มีความหมายในที่นี้ วงกลมแต่ละวงจะมีความสำคัญกับวงอื่น ๆ สรุปได้ว่าการพัฒนาที่จะนำแบบจำลองน้ำพุมาแทนการพัฒนาแบบปกติซึ่งใช้แบบจำลองน้ำตกนั้นไม่มีความเหมาะสมเนื่องจากว่าระบบที่ถูกพัฒนาไม่ได้ถูกมองแบบเชิงวัตถุ (แสดงดังรูปที่ 6)



รูปที่ 6 ลักษณะของแบบจำลองน้ำพุ ที่อยู่ในรูปของการพัฒนาระบบสารสนเทศแบบทั่วไป

5. สรุป

จากที่กล่าวมาข้างต้นแบบจำลองน้ำพุเป็นแบบจำลองที่สามารถรองรับการพัฒนาระบบสารสนเทศเชิงวัตถุได้เหมาะสมกว่าแบบจำลองน้ำตกที่ใช้อยู่ในปัจจุบันเพราะการพัฒนาแบ่งออกเป็น 2 ระดับและแต่ละระดับประกอบด้วย 10 ขั้นตอน นอกจากนี้ในแต่ละขั้นตอนมีลักษณะการทำซ้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และการเวียนบังเกิดกับทุก ๆ ขั้นตอนซึ่งแบบจำลองน้ำตกไม่มี และปัจจุบันเครื่องมือต่าง ๆ ที่มีลักษณะวิชวล (Visual) มากขึ้นซึ่งสอดคล้องกับแบบจำลองน้ำพุทางด้านคลังของซอฟต์แวร์หรือคลังของคลาสที่สามารถถูกเรียกใช้ได้ทุกครั้งเมื่อต้องการ ส่วนด้านการบำรุงรักษาหรือปรับปรุงเปลี่ยนแปลงระบบสามารถดำเนินการได้ง่ายและรวดเร็วกว่าเดิม

6 เอกสารอ้างอิง

[1] Bertrand Meyer, **Object -Oriented Software Construction 2nd**, Prentice Hall, 1997.

[2] Brains Henderson-Sellers and JM Edwards, **Book Two of Object-Oriented Knowledge: The Working Object**, Prentice Hall, Sydney, 1993.

[3] Brains Henderson-Sellers and JM Edward, "The object-oriented systems lifecycle", **Communications of The ACM**, Vol 33, No. 9 : 142-159, 1990.

[4] Carlo Ghezzi, Dino Mandrioli and Mehdi Jazayeri, **Fundamentals of Software Engineering**, Prentice Hall, 1997.

[5] J. Ramgaugh, M. Blaha, M. Premerlani, F.Eddy and W.Lorensen, **Object-Oriented modeling and design**, Prentice-Hall International, 1991.

[6] Ian Graham, **Migrating To Object Technology**, Add-Wesley, 1995.

[7] Luiz F. Capretz and Miriam A. M. Capretz, **Object-Orieted Software: Design and Maintenance**, World Scientific Publishing, 1996.

[8] T. Aoki and T. Katayama, **Formal Model Approach for Reliable Object-Oriented Information System Design**, Sci/ISAS'98, pp228-235, 1998.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.
สรุปขั้นตอนและกิจกรรมของแบบจำลองประยุกต์

1. โมดูลศึกษาความเหมาะสมและความต้องการของผู้ใช้ (Requirements and Feasibility Study Module)

ระยะที่ 0 การศึกษาความเหมาะสม (Stage 0 Feasibility Study)

ขั้นตอน 010 การเตรียมศึกษาความเหมาะสม (Prepare for the Feasibility Study)

งานที่ 10 กำหนดขอบเขตของโครงสร้างโดยการสร้างกระบวนการและดาต้าโมเดลของระบบเดิมอย่างกว้าง ๆ

งานที่ 20 ตกลงขอบเขตของโครงการพร้อมกับเอกสารที่ได้ทำการศึกษาความเหมาะสม

งานที่ 30 วางแผนสำหรับการศึกษาความเหมาะสม

งานที่ 40 สร้างแผนภูมิการไหลของข้อมูล (ระบบงานเดิม)

งานที่ 50 สร้างแผนภาพข้อมูล (LDS)

ขั้นตอน 020 กำหนดปัญหา (Define the Problem)

งานที่ 10 วิเคราะห์ความต้องการของผู้ใช้ที่มีต่อระบบเดิมโดยศึกษาจากภาพรวมของแผนภูมิการไหลของข้อมูลกับแผนภูมิดาต้า เพื่อนำมากำหนดรูปแบบความต้องการของผู้ใช้

งานที่ 20 นำรายละเอียดของโครงสร้างความต้องการมาขยายแผนภูมิการไหลของข้อมูลกับแผนภูมิการไหลของข้อมูล ที่ได้จากขั้นตอนที่ 010 พร้อมกับบันทึกปัญหาที่เกิดขึ้นกับระบบเดิมลงใน Requirements Catalogue

งานที่ 30 บันทึกความต้องการของผู้ใช้ลงใน User Catalogue

งานที่ 40 จัดหาและเพิ่มเติมฟังก์ชันการทำงานที่ต้องการให้มีในระบบใหม่ โดยแยกประเภทของฟังก์ชัน คือ ฟังกชันที่ต้องการใช้ กับฟังก์ชันที่เป็นสาธารณะ (Non-funtional) พร้อมกับบันทึกรายละเอียดของฟังก์ชันต่าง ๆ ลงใน Requirement Catalogue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 50 สรุปปัญหาและความต้องการของผู้ใช้ที่อยู่ใน Requirement Catalogue

งานที่ 60 นำเสนอปัญหาของระบบให้กับผู้บริหารโครงการรับทราบและอนุมัติ

ขั้นตอน 030 Define Scenario Development

งานที่ 10 สร้าง Use Case Model โดยกำหนดความต้องการของผู้ใช้หรือลักษณะการทำงานของฟังก์ชันที่เกิดขึ้นภายในระบบ

งานที่ 20 กำหนดผู้กระทำให้กับระบบ ซึ่งผู้กระทำ คือ บุคคลหรือสิ่งที่เกี่ยวข้องกับระบบคล้ายกับเอนติตีของขั้นตอนที่ 010

งานที่ 30 กำหนด Use Case ให้กับระบบ Use Case คือ กำหนดรายละเอียดของระบบที่เกิดขึ้นภายใน เช่น พฤติกรรม รูปแบบของฟังก์ชัน และ วิธีการปฏิบัติงาน เป็นต้น ดังนั้นการกำหนด Use Case ที่เกิดขึ้นกับระบบควรกำหนดตามความเป็นจริงเพื่อจะนำไปกำหนดฟังก์ชันให้กับระบบต่อไป

ขั้นตอน 040 ออกแบบอินพุต-เอาต์พุต (Define Input / Output of System)

งานที่ 10 กำหนดรูปแบบของอินเทอร์เฟซ

งานที่ 20 กำหนดรูปแบบของอินพุตให้กับระบบ

งานที่ 30 กำหนดรูปแบบของเอาต์พุตให้กับระบบ

งานที่ 40 ประมาณเวลาสำหรับการทำงานของระบบที่มีต่อผู้ใช้ในแต่ละกรณีอย่างคร่าว ๆ

ขั้นตอน 050 กำหนดทางเลือกทางธุรกิจและเทคนิค (Select Feasibility Option)

งานที่ 10 กำหนดทางเลือกให้กับระบบใหม่อย่างคร่าว ๆ โดยคำนึงถึงความต้องการของผู้ใช้ระบบเป็นหลัก

งานที่ 20 กำหนดทางเลือกทางธุรกิจให้กับระบบทั้ง BSO และ TSO โดยเสนอมาอย่างน้อย 6 ทางเลือก

งานที่ 30 กำหนดทางเลือกทางเทคนิคให้กับระบบโดยทางเลือกทางเทคนิคต้องสนับสนุนทางเลือกทางธุรกิจทั้งหมดที่ได้เสนอมาก่อนหน้านี้

งานที่ 40 นำโครงสร้างของ BSO กับ TSO มารวมกัน จากนั้นนำไปตรวจสอบกับความต้องการของระบบว่าทางเลือกใดสามารถสนับสนุนนความต้องการของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบได้บ้าง (ซึ่งการตรวจสอบจะทำกันระหว่าง BSO TSO และ Feasibility Options เท่านั้น) พร้อมกับจัดทำรายงานของทางเลือกต่าง ๆ

งานที่ 50 เพิ่มเติมรายละเอียดของทางเลือกให้กับรายงานของทางเลือกที่ได้จัดไว้

งานที่ 60 จัดทำแผนงานการพัฒนาของระบบแบบคร่าว ๆ เพื่อให้สอดคล้องกับรายละเอียดของทางเลือกต่าง ๆ

งานที่ 70 นำเสนอทางเลือกที่สามารถสนับสนุนโครงการให้กับผู้บริหารเลือก

ขั้นตอน 060 การจัดทำตารางดำเนินการ (Define Project Management)

งานที่ 10 จัดทำแนวทางการพัฒนาระบบจากทางเลือกที่ได้รับพิจารณาจากผู้บริหาร โดยใช้ PERT/CPM หรือ Gantt Chart พร้อมกับจัดทำแผนสำรองในกรณีที่ไม่สามารถดำเนินงานต่อไปได้

งานที่ 20 สร้างแผนผังลำดับการทำงานให้กับระบบโดยลำดับการทำงานก่อนหลัง พร้อมกับแบ่งกลุ่มการทำงานให้กับระบบ (ถ้าระบบนั้นเป็นระบบที่มีขนาดใหญ่)

งานที่ 30 นำ TSO ที่ถูกเลือกมาพิจารณาถึงทรัพยากรที่จำเป็นต้องใช้กับการพัฒนาระบบ มาจัดทำแผนงานการบริหารทรัพยากรให้กับโครงงาน

ขั้นตอน 070 รวบรวมเอกสารของการศึกษาความเหมาะสม (Assemble Feasibility Report)

1. รายละเอียดสภาพแวดล้อมของระบบปัจจุบันแบบคร่าว ๆ
2. รายละเอียดสภาพแวดล้อมที่ต้องการแบบคร่าว ๆ
3. ขอบเขตของปัญหาพร้อมรายละเอียด
3. แผนงานของระบบที่ใช้สำหรับระบบเชิงวัตถุ (Scenario Plan)
4. Requirements Catalogue
5. ตารางของการพัฒนาระบบโดยรวม (Project management)
6. โครงสร้างหน้าจอของระบบ ทั้ง อินพุต-เอาต์พุตและอินเทอร์เฟซ
7. ทางเลือกของโครงการทั้งทางธุรกิจและเทคนิคแบบคร่าว ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะที่ 1 ศึกษาระบบเดิม (Stage 1 Investigate of Current System)

ขั้นตอน 110 ประเมินการศึกษาความเหมาะและตรวจสอบเอกสาร (Feasibility Evolution and Review Report)

1. ตรวจสอบ Requirement Catalogue
2. ตรวจสอบแผนผังการไหลของข้อมูลและ Logical Data Model
3. ตรวจสอบตารางการพัฒนาระบบ
4. ตรวจสอบเอกสารจากเจ้าของระบบ

ขั้นตอน 120 สำรวจแผนภูมิการไหลของข้อมูล (Investigation of DFDs)

งานที่ 10 ถ้าหากสามารถสร้าง Document Flow ให้กับระบบได้ควรทำ เพราะจะทำให้เข้าใจการไหลเวียนของข้อมูลในระดับกายภาพได้ดียิ่งขึ้น (Document Flow ไม่มีในขั้นตอนของการศึกษาความเหมาะสม)

งานที่ 20 ตรวจสอบแผนภูมิการไหลของข้อมูลและกระจายแผนภูมิไปสู่ระดับที่ 1

งานที่ 30 กระจายกระบวนการที่อยู่ในแผนภูมิระดับที่ 1 ไปสู่ระดับที่ 2 และ 3 ตามลำดับ

งานที่40 สร้างกระบวนการสุดท้าย (Elementary Process) ที่เกิดจากการกระจายพร้อมกับใส่รายละเอียดของกระบวนการนั้น จากนั้นปรับปรุงรายละเอียดของเอนติตีภายนอกที่เกี่ยวข้องและบันทึกราละเอียดของการติดต่อระหว่างกระบวนการสุดท้ายแต่ละตัวที่มีต่อเอนติตี หรือที่เรียกว่า ตารางอินพุต-เอาต์พุต (I/O Description)

ขั้นตอน 130 สำรวจดาต้าโมเดล (Investigation of Data)

งานที่ 10 ตรวจสอบแผนภูมิของดาต้า (LDS)

งานที่ 20 กำหนดแอตทริบิวต์ให้กับเอนติตี

งานที่ 30 ตรวจสอบ LDM กับกระบวนการสุดท้ายทุกตัวที่อยู่ใน DFM ว่าสามารถสนับสนุนซึ่งกันและกันหรือไม่ (อาจจะอธิบายด้วยตารางแมทริกซ์ที่แสดงระหว่างกระบวนการกับ LDM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 40 บันทึกความต้องการใหม่ ๆ ที่เกิดขึ้นกับดาต้าหรือปัญหาที่เกิดขึ้นกับดาต้าปัจจุบันลงใน Requirement Catalogue

ขั้นตอน 140 เปลี่ยนไดอะแกรมระดับกายภาพไปสู่ระดับตรรกะของระบบปัจจุบัน (Convert Physical to Logical view of Current System)

งานที่ 10 ย้ายแผนภูมิการไหลของข้อมูลระดับกายภาพไปสูระดับตรรกะ โดยเริ่มจากระดับล่างของแผนภูมิ ซึ่งจะพิจารณาในแต่ละกระบวนการที่เกิดขึ้นในขั้นตอน 120 ก่อนการย้ายนั้น ควรตรวจสอบแผนภูมิให้รอบครอบก่อนซึ่งอาจมีบางช่วงหรือบางอย่างขาดหายไปในขั้นตอนการวิเคราะห์ ถ้าบางกระบวนการหายไปควรนำกลับไปวิเคราะห์อีกครั้ง ถ้าไม่มีข้อผิดพลาดควรทำการเริ่มย้ายแผนภูมิ โดยเริ่มจากกระบวนการสุดท้ายที่อยู่ระดับล่างของแผนภูมิก่อนเป็นอันดับแรก การย้ายนั้นควรขึ้นกับประสบการณ์และวิธีการจัดการของนักวิเคราะห์ระบบเป็นหลัก

งานที่ 20 ตรวจสอบอี อาร์ ไดอะแกรม โดยทำการจับกลุ่มของเอนติตีที่เกี่ยวข้องไว้ที่เดียวกัน เพื่อนำไปสร้าง LDS ในระดับตรรกะให้กับระบบ ซึ่งการจับกลุ่มนี้ในแต่ละแฟ้มข้อมูลในทางตรรกะอาจมีมากกว่าหนึ่งเอนติตีได้

งานที่ 30 ตรวจสอบกระบวนการสุดท้ายที่อยู่ในระดับล่างของแผนภูมิการไหลของข้อมูล โดยพิจารณาแต่ละกระบวนการ กล่าวคือ กระบวนการใดที่มีลักษณะการทำงานที่คล้ายกันควรยุบรวมกันหรือกระบวนการใดที่เดิมมีการติดต่อกันผ่านฐานข้อมูลซึ่งไม่มีความจำเป็นที่ต้องใช้ฐานข้อมูลนั้นควรยุบการเชื่อมต่อเปลี่ยนมาทำการเชื่อมต่อโดยตรงแทน

งานที่ 40 วาดแผนภูมิระดับตรรกะที่ย้ายจากระดับกายภาพ พร้อมกับตรวจสอบแผนภูมิการไหลของข้อมูลและแผนภูมิของดาต้าที่ย้ายให้ถูกต้อง

งานที่ 50 บันทึกสิ่งที่เปลี่ยนแปลงที่เกิดขึ้นจากการย้ายแผนภูมิลงใน Requirement Catalogue เพื่อเตรียมย้ายไปสู่การวิเคราะห์ให้กับระบบใหม่

ขั้นตอน 150 รวบรวมเอกสารของการศึกษาระบบเดิม (Assemble Document)

- Context Diagram
- Logical Data Flow Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Current Environment LDM
- Logical Data Store/Entity Cross-Reference (Process-to-Data Matrix)
- Requirements Catalogue
- User Catalogue

ระยะที่ 2 การกำหนดสภาพแวดล้อมของระบบ (Stage 2 Select Environment of System)

ขั้นตอน 210 ตรวจสอบทางเลือกทางธุรกิจและทางเทคนิคจากขั้นตอนที่ 050 (Review BSO and TSO from Step 050)

งานที่ 10 กำหนด BSO และ TSO จากความต้องการของผู้ใช้ให้กับระบบใหม่

งานที่ 20 สร้างทางเลือกให้ BSO และ TSO อย่างน้อย 3-6 ทางเลือก

งานที่ 30 นำ BSO และ TSO ไปนำเสนอให้กับผู้ใช้ พร้อมกับแสดงความสัมพันธ์ของทางเลือก โดยเชื่อมโยงกันระหว่าง BSO TSO และ Feasibility Option ให้ผู้ใช้ทราบ

งานที่ 40 เพิ่มรายละเอียดของทางเลือกที่ได้เลือกไว้ พร้อมกับแสดงความสัมพันธ์ของ BSO ที่เกิดขึ้นกับ DFM และ LDM และแสดงสภาพแวดล้อมททางเทคนิคที่จะเกิดขึ้นกับระบบสำหรับการเลือกทางเลือกนั้น ๆ

งานที่ 50 นำทางเลือกที่มาพิจารณาถึงขีดความสามารถของระบบกับทางเลือกว่าสามารถรองรับกับความต้องการของระบบได้หรือไม่

งานที่ 60 นำเสนอทางเลือกที่ได้เลือกแล้ว มาจัดทำรายละเอียดที่สมบูรณ์ โดยอาศัยสภาวะแวดล้อมทางธุรกิจและทางเทคนิค

ขั้นตอน 220 เลือกทางเลือกทางธุรกิจและทางเทคนิค (Select BSO and TSO)

งานที่ 10 นำ BSO และ TSO ที่เลือกแล้วมาคำนวณหาต้นทุน ได้แก่ ต้นทุนการพัฒนาและต้นทุนสำหรับการดำเนินงาน โดยต้นทุนการพัฒนา คำนวณจาก กำลังคน วัสดุอุปกรณ์ที่เกี่ยวข้องกับการพัฒนา (แผ่นดิสก์ กระดาษพิมพ์ หมึก และอื่น ๆ) ค่าอบรมและอื่น ๆ ส่วนต้นทุนสำหรับการดำเนินงาน เช่น ค่าลิขสิทธิ์ ซอฟต์แวร์ที่ใช้ในการพัฒนา เงินเดือนพนักงาน เวลาการดำเนินงาน และค่าโสหุ้ยต่าง ๆ เป็นต้น โดยแสดงออกมาในรูปแบบของรายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 20 นำ BSO และ TSO ที่เลือกแล้วมาคำนวณหาผลตอบแทน ซึ่งการคำนวณนั้นจะต้องคำนวณทั้งผลตอบแทนที่เห็นได้และที่ไม่สามาถเห็นได้อย่างชัดเจน พร้อมกับแสดงรายละเอียดต่าง ๆ ในรูปแบบรายงาน

งานที่ 30 คำนวณหาผลกระทบของโครงการ โดยอาศัยเทคนิคการคำนวณทางการเงิน เช่น คำนวณหางวดเวลาคืนทุน (Payback Analysis) มูลค่าปัจจุบันสุทธิ (Net Present Value) อัตราผลตอบแทนจากการลงทุน (Return On Investment)

งานที่ 40 สรุปผลการคำนวณหาต้นทุนและผลตอบแทนของ BSO และ TSO

ขั้นตอน 230 สรุปทางเลือกทางธุรกิจและทางเทคนิค (Summary BSO and TSO)

2. โมดูลการวิเคราะห์ (Analysis Module)

ระยะที่ 3 ระบุความต้องการ (Requirements Specification)

ขั้นตอน 310 ตรวจสอบเอกสารความต้องการของผู้ใช้ (Review Requirement Catalogue)

งานที่ 10 ปรับปรุงและตรวจสอบ Requirements Catalogue กับทาเลือกทางธุรกิจและเทคนิคของระบบอย่างละเอียด ก่อนนำไปปรับปรุงแผนภูมิทั้งหมดให้เป็นไปตามความต้องการของผู้ใช้

งานที่ 20 พิจารณาจุดประสงค์ของการพัฒนาระบบ

งานที่ 30 เลือกรูปแบบการพัฒนาระบบ พร้อมกับบันทึกรายละเอียดให้กับ Project Report

ขั้นตอน 320 วิเคราะห์ระบบเครือข่าย <ถ้าต้องการ> (Network Analysis, if need)

งานที่ 10 ตรวจสอบระบบเครือข่ายที่องค์กรมีอยู่ เพื่อให้ทราบถึงประสิทธิภาพของระบบที่มีอยู่เพียงพอหรือตรงต่อความต้องการของโครงการหรือไม่

งานที่ 20 สร้างแผนผังของระบบเครือข่ายโดยใช้ LCD (Location Connectivity Diagram) เพื่อจัดกลุ่มของกระบวนการ ดาต้า ที่มีลักษณะคล้ายกันไว้ที่เดียวกันเพราะจะทำให้ง่ายต่อการจัดการและวางแผนในการออกแบบเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 30 จัดทำรายละเอียดของความสัมพันธ์ ที่เกิดขึ้นระหว่างดาต้ากับกระบวนการ ด้าต้ากับตำแหน่งที่ใช้ในระบบเครือข่าย และ กระบวนการกับตำแหน่งที่กระบวนการนั้น ๆ ไปกระทำกับระบบเครือข่าย ซึ่งการจัดทำรายละเอียดนี้อาจจะจัดทำในรูปของตารางความสัมพันธ์ (Matrix)

งานที่ 40 คำนวณหาปริมาณงานของผู้ใช้แต่ละคน ณ เวลาหนึ่ง ที่ต้องใช้ระบบผ่านเครือข่าย วิธีการคำนวณเริ่มจาก ระบุจำนวนไคลเอ็นต์ที่มีอยู่ในระบบ ระบุจำนวนทรานส์แอ็กชันของผู้ใช้แต่ละคน ซึ่งการคำนวณทั้งหมดนี้จะนำไปกำหนดแบนด์วิดธ์ให้กับระบบเน็ตเวริก์และทราบถึงเวลาตอบสนองของแต่ละทรานส์แอ็กชัน

งานที่ 50 เลือกเทคโนโลยีของระบบเครือข่ายที่สามารถตอบสนองกับระบบได้ โดยคำนึงถึง ปัจจัยต่าง ๆ เช่น รูปแบบของเครือข่าย ความเร็ว ความเข้ากันได้กับเครือข่ายระบบอื่น เป็นต้น

งานที่ 60 สรุปการวิเคราะห์ของระบบเครือข่ายที่มีต่อระบบงาน

ขั้นตอน 330 ปรับปรุง DFM และ LDM ตามความต้องการของผู้ใช้ (Develop Require DFM and LDM)

งานที่ 10 ปรับปรุงแผนภูมิการไหลของข้อมูลตั้งแต่ระดับแรกจนถึงระดับสุดท้าย เพื่อให้สอดคล้องกับทางเลือกทางธุรกิจและเทคนิค โดยเลือกทำเฉพาะกระบวนการที่เกี่ยวข้องเท่านั้น พร้อมกับบันทึกความเปลี่ยนแปลงลง Requirements Catalogue

งานที่ 20 บันทึกอินพุต-เอาต์พุตของแต่ละกระบวนการที่เกิดกับเอนติตี (I/O Descriptions)

งานที่ 30 ตรวจสอบกระบวนการทุก ๆ ตัวที่อยู่ใน DFM ให้สอดคล้องกับ LDM ที่มีอยู่อย่างละเอียด โดยนำ LDS กับ เอนติตีมาประกอบการพิจารณา เพื่อป้องกันไม่ให้กระบวนการใด ๆ ที่จำเป็นจะต้องติดต่อกับ LDS เกิดความบกพร่อง

งานที่ 40 สร้าง User Role ให้กับ DFM กับเอนติตีภายนอก เพื่อไว้สำหรับตรวจสอบความสัมพันธ์ระหว่างกัน

งานที่ 50 ปรับปรุง LDM ให้สอดคล้องกับความต้องการของผู้ใช้ที่ต้องการให้เกิดขึ้นกับระบบใหม่ ซึ่งการปรับปรุงควรทำเฉพาะLDM ที่เกี่ยวข้องเท่านั้น บันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนแปลงที่เกิดขึ้นกับแอตทริบิวต์ของแต่ละเอนติตีและปรับปรุง Requirements Catalogue

งานที่ 60 ตรวจสอบ LDM จากเอกสาร โดยจะต้องระบุได้ว่า LDM ทุกตัวสนับสนุนในทุก ๆ กระบวนการสุดท้ายของ DFM ของระบบใหม่

งานที่ 70 ปรับปรุง LDM กับฟังก์ชันสาธารณะให้เหมาะสมกับ Requirements Catalogue

งานที่ 80 ตรวจสอบ LDM และ DFM ที่ได้ทำการปรับปรุงตามความต้องการของระบบใหม่อย่างละเอียดอีกครั้ง พร้อมกับตรวจสอบเอกสารที่เกี่ยวข้อง

ขั้นตอน 340 Derive System Function

งานที่ 10 กำหนดฟังก์ชันให้กับ DFM ที่ถูกปรับปรุงเรียบร้อยแล้ว พร้อมกับกำหนดเหตุการณ์และสิ่งที่จะเกิดขึ้นตามมาภายหลังของแต่และฟังก์ชัน การจับกลุ่มควรนำฟังก์ชันที่สัมพันธ์กันมารวมเป็นกลุ่มเดียวกัน ฟังก์ชันที่ไม่สัมพันธ์กับใครควรกำหนดเป็นฟังก์ชันเดียวแทน

งานที่ 20 กำหนดฟังก์ชันเพิ่มเติมโดยพิจารณาจาก Requirements Catalogue กับ DFM ในกรณีที่ความต้องการหลักของระบบไม่มี

งานที่ 30 สร้างโครงสร้างอินพุต-เอาต์พุตของแต่ละฟังก์ชัน

งานที่ 40 สร้างเมทริกซ์ระหว่างฟังก์ชันกับ User Role ไว้สำหรับตรวจสอบการทำงานระหว่างผู้ใช้ระบบกับฟังก์ชันที่เกิดขึ้น เพื่อให้ทราบถึงความสัมพันธ์ระหว่างกัน

งานที่ 50 ระบุระดับความสำคัญของแต่ละฟังก์ชัน

ขั้นตอน 350 ทำบรรทัดฐานให้กับดาต้าโมเดล (Enhance Required Data Model)

งานที่ 10 เปลี่ยน LDM ไปสู่รูปแบบของตารางเพื่อเตรียมตัวทำบรรทัดฐานฐานข้อมูล (Normalization)

งานที่ 20 ทำบรรทัดฐานฐานข้อมูลของตารางไปสู่ระดับ 3

งานที่ 30 นำตารางที่ทำบรรทัดฐานฐานข้อมูลแล้ว กลับสู่แผนภูมิของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 40 เปรียบเทียบ LDM เดิมกับ LDM ใหม่ที่ได้หลังจากการทำบรรทัดฐานฐานข้อมูล ถึงสิ่งที่เปลี่ยนแปลงไป เพื่อตรวจสอบความถูกต้องให้ตรงความต้องการของผู้ใช้อีกครั้ง

ขั้นตอน 360 สร้างแบบจำลอง (Create Prototype)

งานที่ 10 นำแบบจำลองหน้าจอที่ได้จากขั้นตอนที่ 040 มาสร้างโครงสร้างของรูปแบบหน้าจอเพิ่มเติม เช่น หน้าจอหลักและย่อย ฟอร์มต่าง ๆ รูปแบบรายงาน เป็นต้น โดยสร้างตามความต้องการที่เกิดขึ้นจาก Requirements Catalogue

งานที่ 20 สร้างโครงสร้างเมนูของแบบจำลอง

งานที่ 30 สร้างแบบจำลองพร้อมกับสร้างเส้นทางการเข้าถึงของแต่ละหน้าจอและรายงานในแต่ละแบบจำลอง

ตั้งแต่งานที่ 4-7 มีการทำซ้ำกันจนกระทั่งการสร้างแบบจำลองสิ้นสุด

งานที่ 40 สร้างแบบจำลอง

งานที่ 50 จัดเตรียมแบบจำลองเพื่อเสนอต่อเจ้าของโครงการ

งานที่ 60 นำเสนอแบบจำลองในแต่ละส่วนให้กับเจ้าของโครงการ

งานที่ 70 รับข้อเสนอแนะจากผู้ใช้

งานที่ 80 ปรับปรุงแบบจำลองตามความต้องการของผู้ใช้

ขั้นตอน 370 การระบุกระบวนการพัฒนา (Develop Processing Specification)

งานที่ 10 สร้างตารางเหตุการณ์/เอนติตีเมตริกซ์ (Entity Acess Matrix) ของแต่ละเหตุการณ์ที่กระทบกับเอนติตีพร้อมกับเอกสารประกอบ

งานที่ 20 นำเอนติตีของ LDS แต่ละตัวมาสร้างเอนติตีไลฟ์ฮิสทรี (ELH-Entity life Histories) ไดอะแกรม โดยเริ่มจากล่างขึ้นบน กล่าวคือ เริ่มสร้างจาก LDS ระดับล่างขึ้นมาก่อน 1 LDS เท่ากับ 1 ELH ซึ่งในระบบจะมี ELH เท่ากับจำนวนของ LDS ที่มีอยู่ จากนั้นนำ ELH มาระบุขั้นตอนการทำงาน หรือ "Operation List" ทำจากบนลงล่าง

งานที่ 30 สร้าง ECD (Effect Correspondence Diagram) ไดอะแกรม ของ LDS ซึ่งเป็นการแสดงความสัมพันธ์ระหว่าง ELH กับเหตุการณ์ที่เกิดขึ้นกับ LDS พร้อมกับระบุขั้นตอนการทำงาน (operation list) หรือลำดับของผลกระทบที่มีต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอตทริบิวต์ของแต่ละเอนติตีตามเหตุการณ์ที่เข้ามากระทบ จากนั้น ระบุความต้องการที่เกิดขึ้นของแต่ละเหตุการณ์โดยสร้าง EAP (Enquiry Access Path) พร้อมคำอธิบาย ซึ่ง EAP คือ การตรวจสอบข้อมูลเข้าและออกที่กระทบกับเอนติตีตามเหตุการณ์ที่เกิดขึ้น

งานที่ 40 ปรับปรุง Requirement Catalogue และ LDS ไดอะแกรมที่มีเหตุการณ์หรือความต้องการใหม่ ๆ และผลกระทบกับความต้องการในระหว่างการสร้าง ELH เพื่อให้ฟังก์ชันที่เกิดขึ้นตรงตามเหตุการณ์ที่เกิดขึ้นระหว่างการวิเคราะห์

งานที่ 50 ระบุจำนวนของข้อมูลที่สามารถบรรจุภายใน LDM ของแต่ละเอนติตี เพื่อประมาณเนื้อที่ของขนาดฐานข้อมูลที่จะต้องใช้ภายในระบบ

ขั้นตอน 380 รับรองวัตถุประสงค์ของระบบ (Confirm System Objectives)

งานที่ 10 ตรวจสอบฟังก์ชันและ Non-function ที่กำหนดไว้ครบถ้วนและถูกต้องหรือไม่ โดยพิจารณาแต่ละ Requirement Catalogue ที่มีอยู่เพื่อยืนยันว่าฟังก์ชันและ Non-function ที่ได้ตรงตาม Requirements Catalogue ที่ผู้ใช้ระบบระบุไว้

งานที่ 20 ทำความเข้าใจกับฟังก์ชันและ Non-function หรือ LDS ที่ถูกกำหนดขึ้นมาใหม่ตามความต้องการของผู้ใช้

งานที่ 30 ยืนยันฟังก์ชันที่กำหนดไว้ เช่น ความถูกต้อง ตรงตามความต้องการ สามารถแก้ไขปัญหาได้ถูกต้อง เป็นต้น เพื่อให้ฟังก์ชันนั้นมีความสมบูรณ์และครบถ้วนตรงกับที่ผู้ใช้ระบบต้องการ

งานที่ 40 ยืนยัน LDM ที่กำหนดไว้นั้น มีความถูกต้องและครบถ้วน

ขั้นตอน 390 รวมรวมเอกสารและไดอะแกรม

- Data Flow Diagram (Update)
- Logical Data Store (Update)
- Location Connectivity Diagram (LCD), ถ้ามีระบบเครือข่าย
- Grouping Function Diagram
- Entity Life Histories Diagram
- Effect Correspondence Diagram
- Input-Output Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Requirements Catalogue

ระยะที่ 4 การย้ายไปสู่วิธีการเชิงวัตถุ (Migrating to Object Diagram)

ขั้นตอน 410 ตรวจสอบเอกสารและปรับปรุง DFM กับ LDM (Review & Evolution DFM and LDM)

งานที่ 10 ตรวจสอบแผนภูมิการไหลของข้อมูลในแต่ละระดับ และ LDS ไดอะแกรม

งานที่ 20 ตรวจสอบเอนติตีไลฟ์ฮีสทรีไดอะแกรมและ ECD

งานที่ 30 ตรวจสอบรายละเอียดของแต่ละฟังก์ชัน

ขั้นตอน 420 เปลี่ยน LDS ไปสู่สัญลักษณ์ของวัตถุ/คลาส (Convert LDS To Object/Class Notation)

งานที่ 10 เปลี่ยนสัญลักษณ์ของ LDS ไปสู่วัตถุหรือคลาส โดยให้ 1 LDS เท่ากับ 1 วัตถุหรือคลาส

งานที่ 20 ให้นำแอตทริบิวต์ที่อยู่ในตารางไปใส่ในวัตถุที่แปลงเรียบร้อยแล้ว (ในช่องที่ 2 ของวัตถุหรือคลาส)

งานที่ 30 ขจัดคีย์หลักของแต่ละวัตถุ

งานที่ 40 เปลี่ยนคีย์รองของแต่ละวัตถุเป็นชื่อของวัตถุที่อ้างอิงถึง เพราะหลักการของวัตถุในการอ้างอิงใช้ Object Identity (OID) แทนการใช้คีย์รอง การอ้างอิงระหว่างวัตถุนั้นระบบฐานข้อมูลจะสร้างหมายเลขขึ้นมาให้เองอัตโนมัติ (ขึ้นอยู่กับฐานข้อมูลที่เลือกใช้) ซึ่งหมายเลขจะไม่ซ้ำกัน (Unique Number) ในแต่ละวัตถุที่อ้างถึง

ขั้นตอน 430 กำหนดความสัมพันธ์ระหว่างวัตถุ/คลาส (Define Relation)

งานที่ 10 พิจารณาความสัมพันธ์ระหว่างเอนติตีใน LDS

งานที่ 20 แปลงสัญลักษณ์ความสัมพันธ์ของไดอะแกรม

งานที่ 30 ตรวจสอบไดอะแกรมที่แปลงสัญลักษณ์จาก LDS ไป OMT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอน 440 ย้ายฟังก์ชันจากแผนภูมิการไหลของข้อมูลไปสู่ฟังก์ชันของวัตถุ

งานที่ 10 ตรวจสอบเอนติตีไลฟ์ฮีสทรีไดอะแกรมและ ECD ของแต่ละเอนติตีพร้อมกับ Operation List ของแต่ละไดอะแกรม

งานที่ 20 จัดกลุ่มของเอนติตีไลฟ์ฮีสทรีไดอะแกรมและ ECD ที่มีการทำงานคล้ายกันเพื่อนำมาจัดเรียงใหม่ ให้อยู่ในรูปของ Jackson Diagram เพื่อเป็นการยืนยันว่าฟังก์ชั่นและแอตทริบิวต์ภายในที่เกี่ยวข้องกับฟังก์ชันที่เกิดขึ้นนั้น ถูกต้องไม่ผิดพลาดหรือไม่ได้เกิดขึ้นมาลอย ๆ

งานที่ 30 นำฟังก์ชันในของแต่ละเอนติตีไลฟ์ฮีสทรีไดอะแกรมที่จัดกลุ่มเรียบร้อยแล้วไปสู่ ฟังก์ชันของวัตถุ ขั้นตอนการใส่ฟังก์ชันในวัตถุผู้วิเคราะห์ นำเอนติตีไลฟ์ฮีสทรีไดอะแกรมและECD ที่เปลี่ยนไปสู่ Jackson Diagram มาพิจารณา กล่าวคือ นำชื่อฟังก์ชันที่เป็นฟังก์ชันระดับสุดท้ายของไดอะแกรมมาพิจารณาเท่านั้น เพราะฟังก์ชันที่มีการเรียกซ้ำหรือมีทางเลือกนั้น (ระดับที่อยู่ก่อนหน้าและไม่มี"operation list" กำกับ) ไม่ใช่ฟังก์ชันสุดท้ายของการทำงานไม่จำเป็นต้องคำนึงถึง แต่ฟังก์ชันที่มีหลายเลขการทำงานกำกับจึงจะเป็นฟังก์ชันที่มีการทำงานจริงและนำมาใส่ในวัตถุ กล่าวคือ ฟังก์ชันที่มี “operation list” กำกับเท่านั้นจึงจะนำมาพิจารณา

งานที่ 40 ปรับปรุงวัตถุพร้อมกับตรวจสอบวัตถุไดอะแกรม หลังจากนำฟังก์ชันมาใส่ในวัตถุ

ขั้นตอน 450 การตรวจสอบการย้ายจากวิธีเชิงโครงสร้างไปสู่วิธีเชิงวัตถุ (Object-Oriented Testing)

งานที่ 10 ตรวจสอบการเปลี่ยนแปลงจากวิธีเชิงโครงสร้างไปสู่วิธีเชิงวัตถุที่เปลี่ยนจาก LDS ไดอะแกรม

งานที่ 20 ตรวจสอบความสัมพันธ์ของแต่ละวัตถุ และ/หรือ คลาส

งานที่ 30 ตรวจสอบแอตทริบิวต์และฟังก์ชัน (ที่แปลงมาจากเอนติตีไลฟ์ฮีสทรีไดอะแกรม)ของแต่ละวัตถุ

งานที่ 40 ปรับปรุงรายละเอียดของแต่ละวัตถุให้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอน 460 รวบรวมเอกสารและตรวจสอบไดอะแกรม

- Object/Class Diagram
- Jackson Diagram
- CRC

3. โมดูลการออกแบบ (Design Module)

ระยะที่ 5 การออกแบบระบบ (System Design)

ขั้นตอน 510 การจัดซื้อจัดจ้างฮาร์ดแวร์และซอฟต์แวร์ (Hardware and Software procurements)

งานที่ 10 นำความต้องการทางธุรกิจและเทคนิคที่เกิดขึ้นจากขั้นตอนที่ 210 เพื่อนำรายละเอียดทั้งหมด มาปรับเปลี่ยนให้อยู่ในรูปของ "เอกสารแสดงข้อมูลที่ต้องการ" (RFI- Request for Information)

งานที่ 20 กระจาย RFI ให้กับผู้ขายที่มาจากการสรรหา เช่น การประกาศทางวิทยุ โทรทัศน์ หนังสือพิมพ์ และสื่ออื่น ๆ หรือติดต่อทางผู้จัดการโครงการ เป็นต้น

งานที่ 30 รับรายงานในรูปของใบเสนอรายละเอียดของสินค้า (RFQ-Request for Quotations)และ รายงานคุณสมบัติของฮาร์ดแวร์และซอฟต์แวร์ (RFP - Request for proposals) จากผู้ขาย

งานที่ 40 นำรายงานของทุกบริษัทเข้าที่ประชุมเพื่อเลือกเฉพาะบริษัทที่เหมาะสม ซึ่งการประชุมคัดเลือกบริษัทที่ทำสัญญาซื้อขายด้วย

งานที่ 50 ทำการเลือกบริษัทจากมติของที่ประชุม พร้อมกับทำสัญญาซื้อขายอุปกรณ์ทั้งฮาร์ดแวร์และซอฟต์แวร์ พร้อมกับติดต่อบริษัทผู้ขายเข้ามาทำสัญญาและส่งมอบสินค้าให้กับบริษัท

ขั้นตอน 520 ทบทวนและตรวจสอบความก้าวหน้าของไดอะแกรมและเอกสาร (Review and Evolution Diagram and Document)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอน 530 กำหนดระบบงานย่อยจากแบบจำลองวัตถุ (Define Sub-System from Object Model)

งานที่ 10 จัดกลุ่มของวัตถุที่มีความสอดคล้องกันหรือกลุ่มของวัตถุที่สามารถพัฒนาไปในแนวทางเดียวกันเข้าด้วยกัน ที่เรียกว่า "แพ็กเกจ (Package)" หรือ "ระบบย่อย (Sub-System) พร้อมกับมีเอกสารคอยแจกแจงรายละเอียดซึ่งเรียกว่า "Sub-system Responsibility Specification (SRS)"

งานที่ 20 รวบรวมงานและลำดับการทำงานในแต่ละวัตถุของแต่ละระบบย่อย

งานที่ 30 จัดสรรอุปกรณ์ทั้งฮาร์ดแวร์และซอฟต์แวร์ให้กับระบบย่อย (จัดสรรให้กับกระบวนการและงานในระบบย่อย)

งานที่ 40 จัดสรรฐานข้อมูลให้กับระบบย่อย

งานที่ 50 บันทึกรายละเอียดลงไปในเอกสารของการพัฒนาระบบย่อย

ขั้นตอน 540 กำหนดแผนงานการทำซ้ำ (Define Iteration Plan Development)

งานที่ 10 จัดทำแผนงานให้กับแต่ละระบบย่อย โดยนำเสนอในรูปแบบของ GANTT Chart หรือ PERT/CPM ซึ่งจะแสดงเฉพาะส่วนที่ระบบย่อยนั้นรับผิดชอบเท่านั้น

งานที่ 20 กำหนดขั้นตอนพร้อมระบุเวลาการทำงานของระบบย่อย

ขั้นตอน 550 ตรวจสอบระบบงานย่อย <ถ้าต้องการ> (Revise Sub-System, if need)

ขั้นตอน 560 สรุปการทำงาน (Summary of process and deliverables)

1. รายละเอียดของฮาร์ดแวร์และซอฟต์แวร์
2. ใบเสนอราคาของบริษัทต่าง ๆ
3. มติที่ประชุม
4. ใบสัญญาการซื้อขาย
5. แพ็จเกจไดอะแกรม
6. SRS ไดอะแกรม
7. แผนการพัฒนาของระบบย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะที่ 6 การออกแบบเชิงวัตถุ (Object Design)

ขั้นตอน 610 การออกแบบอัลกอริทึม (Design Algorithms)

งานที่ 10 กระจายโอเปเรชันและแอตทริบิวต์ที่มีอยู่ในแต่ละคลาสออกเพื่อสามารถเห็นการทำงานได้ชัดเจนขึ้น

งานที่ 20 พิจารณาอัลกอริทึมในการอิมพลีเมนต์ควรคำนึงถึง ความซับซ้อนในการคำนวณของฟังก์ชัน ความง่ายต่อการทำความเข้าใจและอิมพลีเมนต์ มีความยืดหยุ่นและสามารถปรับปรุงได้ง่ายกับวัตถุโมเดล เป็นต้น

งานที่ 30 เลือกโครงสร้างข้อมูลที่ต้องนำไปใส่ในอัลกอริทึมของโปรแกรม

ขั้นตอน 620 การควบคุมปฏิสัมพันธ์ภายนอก (Implement control for external interactions)

งานที่ 10 กำหนดเส้นทางเริ่มต้นของการควบคุมโปรแกรม เริ่มพิจารณาจากสเตทเริ่มต้นเพื่อกำหนดเส้นทางการทำงานของเอนติตีไลฟ์ฮีสทรีไดอะแกรมที่อีเวนท์นั้นรับผิดชอบอยู่จากนั้นให้เขียนชื่อของสเตทตามลำดับการทำงานที่เกิดขึ้นจริง เพื่อนำไปสู่ขั้นตอนและลำดับการทำงานจริงของโปรแกรม

งานที่ 20 พิจารณาสเตทที่เส้นทางในเอนติตีไลฟ์ฮีสทรีไดอะแกรมถูกแยกออกจากสเตทหลัก เพื่อกำหนดเงื่อนไขให้กับโปรแกรม

งานที่ 30 พิจารณาสเตทที่มีลักษณะการทำงานแบบลูป (loop) ในเอนติตีไลฟ์ฮีสทรีไดอะแกรม เพื่อนำไปเลือกวิธีการใช้ลูปในโปรแกรม โดยการนำเอนติตีไลฟ์ฮีสทรีไดอะแกรมมาพิจารณาเพื่อให้สามารถเลือกใช้ลูปได้ง่ายขึ้น เพราะถ้าการวนกลับของการทำงานในแต่ละสเตทที่มีลักษณะการทำงานแบบซ้ำซ้อนหลาย ๆ ชั้นสามารถที่จะพิจารณาได้ง่ายกว่าเนื่องจากว่าผู้วิเคราะห์สามารถแยกลูปต่าง ๆ เหล่านั้นได้จากการสังเกตสเตทต้นทางและปลายทางตามเส้นทางของสเตททรานส์ซิชั่นที่เกี่ยวข้องเพื่อแยกลูปแต่ละอันออกเป็นชั้น ๆ แทนที่จะเหมือนการเขียนแผนผังงานแบบเดิม เพราะในสเตทจะรวมคำสั่งหรือขั้นตอนต่าง ๆ ไว้ในที่เดียวกัน

งานที่ 40 ตรวจสอบสเตทและทรานส์ซิชั่นที่ไม่ใช่ส่วนของเงื่อนไข สามารถที่จะนำเทคนิคของภาษาต่าง ๆ ที่เลือกใช้อิมพลีเมนต์มาช่วยจัดการทางด้านการตรวจสอบความถูกต้องของโครงสร้างภาษานั้น ๆ ซึ่งส่วนใหญ่จะนำมาใช้กับการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบเงื่อนไขที่มีการเปรียบเทียบและการวนลูปของโปรแกรม โดยการนำมาใช้นั้นควรจะคำนึงถึงความสามารถของภาษานั้น ๆ ด้วย

ขั้นตอน 630 ปรับโครงสร้างของคลาสเพื่อเพิ่มการสืบทอดคุณสมบัติ (Adjust class structure to increase inheritance)

งานที่ 10 พิจารณาฟังก์ชันภายในคลาส/วัตถุให้เหมาะสม

งานที่ 20 พิจารณาความสัมพันธ์ระหว่างคลาสที่มีลักษณะของสืบทอดคุณสมบัติ

ขั้นตอน 640 ออกแบบแอสโซซิเอชั่น (Design asscociations)

งานที่ 10 พิจารณาความสัมพันธ์ของคลาส/วัตถุจากแบบจำลองวัตถุ

งานที่ 20 ปรับปรุงความสัมพันธ์ของคลาส/วัตถุตามความเหมาะสม

ขั้นตอน 650 ค้นหาเส้นทางการเข้าถึงข้อมูลที่เหมาะสม (Optimize access path for data)

ขั้นตอน 660 ระบุการแสดงตัวแทนของวัตถุ (Determine Object Representation)

ขั้นตอน 670 การตรวสอบการออกแบบเชิงวัตถุ (Object Metrics)

งานที่ 10 หาขนาดของวัตถุภายในไดอะแกรมตามวิธีของ CK Metrics Suties

งานที่ 20 วัดความลึกของการสืบทอดคุณสมบัติ (DIT- Depth of the Inheritance Tree) โดยวัดความลึกจาก Root ถึง Child

งานที่ 30 วัดนับจำนวนลูกของคลาส (NOC-Number of Children) คือนับจำนวนของคลาสลูกที่อยู่ใน Root เดียวกัน

งานที่ 40 วัดความสัมพันธ์ระหว่างคลาส (CBO – Coupling Between Object Classes)

งานที่ 50 วัดความสัมพันธ์ของคลาส (RFC-Response for a Class) คือ กลุ่มของความสัมพันธ์ของเมธอดภายในวัตถุของคลาสเดียวกัน

งานที่ 60 วัดความสัมพันธ์ระหว่างเมธอด (LCOM-Lack of Cohesion in Methods) คือ วัดความสัมพันธ์ของเมธอดในวัตถุเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 70 หาจำนวนของโอเปเรชันของคลาสย่อยที่คาบเกี่ยวกัน (NOO - Number of Operations Overriden by a Subclass) คือ การหาจำนวนของโอเปเรชันที่สืบทอดมาจากคลาสหลัก

งานที่ 80 หาจำนวนของโอเปเรชันของแต่ละคลาสย่อย (NOA – Number of Operations Added by a Subclass)

งานที่ 90 หาลักษณะของอินเด็กซ์ (SI – Specializtion Index) เป็นวิธีการวัดเพื่อหาความเหมาะสมของจำนวนโอเปเรชัน แอตทริบิวต์และลำดับของคลาสย่อยให้มีความเหมาะสมกับระบบ

งานที่ 100 บันทึกรายละเอียดการวัดของวัตถุแต่วัตถุที่ปรากฏใน O/C ไดอะแกรม พร้อมกับนำค่าที่วัดได้ไปปรับปรุงแก้ไขไดอะแกรมที่มีข้อบกพร่อง

ขั้นตอน 680 รวบรวมเอกสารเพื่อการตัดสินใจในการออกแบบ

- Object/Class Diagram (Update)
- Object Metrics Test

4. โมดูลอิมพลีเมนต์และการทดสอบ (Implementation and Testing Module)

ระยะที่ 7 การสร้างระบบ (Construction)

ขั้นตอน 710 การสร้างและทดสอบระบบเครือข่าย (ถ้ามี) (Build and Test Networks, if exists)

งานที่ 10 การสร้างและทดสอบ ซึ่งการทดสอบเริ่มต้นทดสอบจากการเชื่อมต่อของระบบเครือข่ายว่าครอบคลุมพื้นที่ที่ต้องการบริการหรือไม่ เช่น ทดลองเชื่อมต่อระหว่างเครื่องที่อยู่ต่างพื้นที่กัน เพื่อให้ทราบว่า ระบบเครือข่ายนั้นใช้งานได้จริง

งานที่ 20 นำแอพพลิเคชั่นหรือระบบงานจริงมาทดลองใช้งานผ่านระบบเครือข่ายเพื่อเป็นยืนยันว่าระบบนั้นสามารถทำงานได้จริงและถูกต้องตามความต้องการของผู้ใช้

งานที่ 30 วัดประสิทธิภาพของการรับส่งข้อมูลของระบบเครือข่าย ซึ่งอาจจะใช้ซอฟต์แวร์ทดสอบประสิทธิภาพที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอน 720 การสร้างและทดสอบฐานข้อมูล (Build and Test Database)

ขั้นตอน 730 การติดตั้งและทดสอบซอฟต์แวร์ (ถ้ามี) (Install and Test New Software Package, if necessary)

งานที่ 10 ตรวจสอบซอฟต์แวร์และเอกสารประกอบ

งานที่ 20 ทำการติดตั้งซอฟต์แวร์

งานที่ 30 ทดลองใช้ซอฟต์แวร์แพ็กเกจเพื่อยืนยันว่าซอฟต์แวร์นั้นสามารถทำงานได้

งานที่ 40 ปรับปรุงผลกระทบของซอฟต์แวร์ที่มีต่อโครงการ

งานที่ 50 บันทึกประวัติผู้ขายเพื่อเก็บไว้เป็นประวัติสำหรับการบำรุงรักษาซอฟต์แวร์แพ็กเกจ

ขั้นตอน 740 การเขียนและทดสอบโปรแกรม (Coding and Test New Program)

งานที่ 10 ตรวจสอบรายละเอียดทั้งหมดเริ่มตั้งแต่ อัลกอริทึม ฟังก์ชันการทำงาน ความสัมพันธ์ระหว่างโมดูล แนวทางการแก้ไขปัญหาและความต้องการของผู้ใช้เพื่อยืนยันความถูกต้องและความแน่นอนของระบบ

งานที่ 20 การเขียนโปรแกรม

งานที่ 30 ตรวจสอบหรือการวัดคุณภาพของระบบที่พัฒนา

ขั้นตอน 750 รวบรวมเอกสาร (Assembly Report)

- Program
- Program Testing Report
- Software Installation Report
- Database Installation Report
- Network Install and Testing Report

ระยะที่ 8 การส่งมอบ (Delivery)

ขั้นตอน 810 ตรวจสอบการทำงานของระบบ (Conduct System)

งานที่ 10 ตรวจสอบข้อมูลที่อยู่ในระบบเพื่อความถูกต้องหลังจากนำข้อมูลขององค์กรใส่ลงไปในระบบงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานที่ 20 ตรวจสอบซอฟต์แวร์แพ็กเกจและโปรแกรมที่พัฒนาระบบ

งานที่ 30 บันทึกรายละเอียดต่าง ๆ ที่เกิดขึ้น เพราะรายละเอียดต่าง ๆ เหล่านี้สามารถนำไปปรับปรุงแก้ไขหรือเพิ่มเติมให้กับระบบ กรณีที่พบข้อผิดพลาดที่เกิดขึ้น

ขั้นตอน 820 วางแผนการติดตั้งโปรแกรม (Prepare Conversion Plan)

งานที่ 10 กำหนดรูปแบบการติดตั้งระบบใหม่

งานที่ 20 ตรวจสอบระบบงานที่ติดตั้งตามแผนงาน

ขั้นตอน 830 การติดตั้งฐานข้อมูล (Install Database)

งานที่ 10 นำฐานข้อมูลที่มีอยู่จริงในระบบงานเดิมย้ายไปสู่ฐานข้อมูลของระบบงานใหม่

งานที่ 20 ย้ายข้อมูล การย้ายข้อมูลนั้นบางครั้งผู้พัฒนาอาจจะต้องเขียนโปรแกรมขึ้นมาสำหรับการย้ายข้อมูลเข้าสู่ฐานข้อมูล (ปกติระบบฐานข้อมูลจะมีเครื่องมือหรือยูทิลิตี้ไว้สำหรับการย้ายฐานข้อมูลติดมากับตัวระบบ) เมื่อเตรียมทุกอย่างพร้อมแล้วก็ทำการย้ายข้อมูลเข้าสู่ระบบใหม่

งานที่ 30 บันทึกขั้นตอนและรายละเอียดขณะทำการย้ายข้อมูลลงฐานข้อมูลในระบบงานใหม่ เพื่อป้องกันการผิดพลาดและสร้างความมั่นใจว่าข้อมูลที่ย้ายไปนั้นไม่มีผลกระทบกับระบบงานใหม่ที่กำลังจะนำไปใช้งาน

ขั้นตอน 840 ฝึกอบรมผู้ใช้ระบบ (Train System User)

งานที่ 10 กำหนดรูปแบบการอบรมผู้ใช้

งานที่ 20 จัดเตรียมเอกสารหรือคู่มือการใช้งาน

งานที่ 30 คัดเลือกผู้เข้าอบรม

งานที่ 40 คัดเลือกผู้ถ่ายทอด

ขั้นตอน 850 เปลี่ยนแปลงไปสู่ระบบงานใหม่ (Convert to New System)

ขั้นตอนที่ 860 ตรวจสอบการส่งมอบระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. โมดูลสนับสนุนระบบ (System Support Module)

ระยะที่ 9 การติดตามผล (Monitoring)

ขั้นตอนที่ 910 การปรับปรุงระบบเพื่อสะท้อนความต้องการที่เปลี่ยนแปลงไป (Modify Repository to reflect change)

ขั้นตอน 920 การควบคุมเอกสาร (Version Control)

ขั้นตอน 930 ปรับปรุงข้อมูลทางธุรกิจให้ทันสมัย (Update Business Data)

ระยะที่ 10 การปรับปรุงและบำรุงรักษา (Evolution and Maintenance)

ขั้นตอน 1010 การให้ความช่วยเหลือผู้ใช้ (End-User Assistance)

ขั้นตอน 1020 การพัฒนาและทำให้ระบบดีขึ้น (System Enhancement and Improvement)

งานที่ 10 วิเคราะห์ความต้องการของผู้ใช้

งานที่ 20 สร้างกระบวนการและเขียนโปรแกรมตามความต้องการของผู้ใช้

งานที่ 30 พิจารณาโครงสร้างข้อมูลหรือการจัดเก็บฐานข้อมูลแบบใหม่

งานที่ 40 วิเคราะห์โปรแกรมและต้นทุนของการพัฒนาโปรแกรม

งานที่ 50 การทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ – สกุล นายพัฒนพงษ์ ฉันทมิตรโอภาส

วันที่เกิด 15 ตุลาคม พ.ศ. 2514

สถานที่เกิด จังหวัดกรุงเทพมหานคร

วุฒิการศึกษา บริหารธุรกิจบัณฑิต (บธ.บ)
วิชาเอก คอมพิวเตอร์ธุรกิจ
มหาวิทยาลัยเกริก
สำเร็จปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้