

การใช้ยูเอ็มแอลในการพัฒนาระบบงานโฆษณาประชาสัมพันธ์

Public Relation system development with UML

โดย

นายนคร ไชยแสนขมภู

รหัส 42067062



H001798

อาจารย์ที่ปรึกษา

ผศ.ดร. อาริต ธรรมโน

วัน เดือน ปี.....	10	ม.ค.	2550
เลขทะเบียน.....	01798		
เลขเรียกหนังสือ.....	วท. ๑๖	114.๓	2544
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."			

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2544
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การใช้ยูเอ็มแอลในการพัฒนาระบบงาน โฆษณาประชาสัมพันธ์
นักศึกษา	นายนคร ไชยแสนชมภู
อาจารย์ที่ปรึกษา	ศศ.ดร. อาริต ธรรมโน
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

การวิเคราะห์และออกแบบระบบงาน โฆษณาประชาสัมพันธ์ ได้นำเอาหลักการออกแบบเชิงวัตถุมาช่วยในการวิเคราะห์และทำการออกแบบโดยใช้ UML (Unified Modeling Language) ซึ่ง UML เป็นเครื่องมือที่มีประสิทธิภาพและทำการแก้ไขปรับปรุงได้ง่าย นอกจากนี้ผู้ใช้ระบบยังทำความเข้าใจกับระบบได้ง่ายขึ้นเพื่อนำมาใช้ในระบบ MIS ช่วยผู้บริหารตัดสินใจในการเลือกผู้ผลิตโฆษณาทำสัญญาจ้าง ติดตามงาน โฆษณาประชาสัมพันธ์ และวางแผนการดำเนินงานในอนาคต

Title	Public Relation System Development With UML
Student	Mr. Nakorn Chaisanchompoo
Advisor	Asst.Dr. Arit Thamano
Level of Study	Master of Science in formation Technology
Major	Information Science
Academic Year	2001

ABSTRACT

The analysis and design of Public Relation system receive object oriented Method to analysis and design with UML (Unified Modeling Language). UML is Tools. It support the efficiency system and easy to improvement. Unless user is easy to understand management information system. The executive can use system to decide , track and forecast the public relation project in future.

กิตติกรรมประกาศ

โครงการพัฒนาระบบฉบับนี้ สำเร็จลุล่วงได้ต้องขอขอบพระคุณ ผศ.ดร. อาริต ธรรมโน อาจารย์ที่ปรึกษา และขอขอบคุณ คุณอนันท์ทวุฒิ เอื้อพัฒน์วงศ์ ที่ได้เอื้อเฟื้อหนังสือและคำแนะนำในการจัดทำโครงการนี้

นคร ไชยแสนชมภู



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
บทที่ 1	1
1. บทนำ	1
1.1 ความเป็นมาของการศึกษาพัฒนาระบบงาน	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	2
1.4 แผนการดำเนินการศึกษา	2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ	2
บทที่ 2	3
2. ทฤษฎีที่เกี่ยวข้อง	3
2.1 แนวคิดเกี่ยวกับการพัฒนาระบบสารสนเทศเชิงออปเจกต์	3
2.2 ความรู้พื้นฐานเกี่ยวกับเทคโนโลยีออปเจกต์	4
2.3 การวิเคราะห์และออกแบบเชิงออปเจกต์ด้วยภาษาจำลอง Unified Modeling Language (UML)	8
บทที่ 3	12
3. ไตอะแกรมของ UML ที่ใช้ในการวิเคราะห์และออกแบบ	12
3.1 Use-Case Diagram	13
3.2 Sequence Diagram	16
3.3 Collaboration Diagram	17
3.4 Class Diagram	18
3.5 Object Diagram	24
3.6 State Diagram	25
3.7 Activity Diagram	26

3.8 Component Diagram	27
3.9 Deployment Diagram	28
บทที่ 4	29
4. วิธีดำเนินการศึกษา	29
4.1 วิเคราะห์และออกแบบระบบงานปัจจุบัน	29
4.2 วิเคราะห์และออกแบบระบบเชิงวัตถุโดยใช้ UML	29
4.2.1 แนะนำ	29
4.2.2 วิเคราะห์ความต้องการ	29
4.2.3 Object-Oriented Analysis (OOA)	31
4.2.4 Object-Oriented Design (OOD)	48
4.2.5 การออกแบบฐานข้อมูล	51
4.2.6 Screen Layout User Interface	58
บทที่ 5	76
5. ผลการดำเนินการศึกษา	76
บทที่ 6	82
6. สรุปผลการศึกษาและข้อเสนอแนะ	82
บรรณานุกรม	84
ภาคผนวก ก	85



บทที่ 1

บทนำ

1.1 ความเป็นมาของการศึกษาพัฒนาระบบงาน

ระบบงานโฆษณาประชาสัมพันธ์ เป็นระบบที่ส่งเสริมกิจกรรมการดำเนินงานของหน่วยงาน ไม่ว่าจะอยู่ในรูปของการโฆษณาหรือประชาสัมพันธ์ ในขณะที่ระบบงานยังคงไม่ได้นำโปรแกรมที่เป็นการเก็บข้อมูลในรูปแบบที่จะนำมาในการวิเคราะห์ ดังนั้นระบบงานโฆษณาประชาสัมพันธ์จึงเป็นโครงการที่อยู่ในแผนงานระบบข้อมูล MIS โดยมีการแยกข้อมูลเป็น 2 ประเภท ประเภทแรกคือข้อมูลที่ใช้ในการปฏิบัติงานของแต่ละกองงาน ข้อมูลส่วนนี้อยู่ในระบบงานประจำวันของแต่ละกองงาน ส่วนข้อมูลประเภทที่ 2 เป็นข้อมูลที่ใช้เพื่อการวิเคราะห์และตัดสินใจ ซึ่งจะดึงจากข้อมูลที่ใช้ในการวิเคราะห์ทั้งปัจจุบันและการวิเคราะห์ย้อนหลัง

ดังนั้นในการศึกษาโครงการนี้ จึงได้ทำการศึกษาและพัฒนาระบบงานโฆษณาประชาสัมพันธ์ให้สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น โดยได้นำเอาการออกแบบเชิงวัตถุมาเป็นแนวคิดในการวิเคราะห์และออกแบบระบบ เพราะในการออกแบบเชิงวัตถุสามารถช่วยลดปัญหาในการบำรุงรักษาและทำการปรับปรุงให้ง่ายขึ้น

1.2 วัตถุประสงค์

- 1.2.1 เพื่อทำการศึกษาถึงทฤษฎีในการพัฒนาระบบการออกแบบเชิงวัตถุ บนพื้นฐานแนวความคิดของ UML
- 1.2.2 เพื่อสามารถนำทฤษฎีมาประยุกต์ใช้กับระบบงานจริงให้มีประสิทธิภาพและถูกต้องมากขึ้น
- 1.2.3 เพื่อเพิ่มประสิทธิภาพการทำงานให้มีความคล่องตัวมากขึ้น
- 1.2.4 เพื่อเป็นเก็บข้อมูลสำหรับการวิเคราะห์วางแผนงานในอนาคต
- 1.2.5 เพื่อจัดทำรายงานเสนอผู้บริหารได้เร็วขึ้น

1.3 ขอบเขตของโครงการ

- 1.3.1 สามารถนำเอาหลักการออกแบบระบบเชิงวัตถุมาประยุกต์ใช้กับระบบงานจริงได้
- 1.3.2 สามารถใช้ CASE TOOLS มาช่วยในการวาดไดอะแกรมต่าง ๆ ที่ได้ออกแบบไว้เพื่อตรวจสอบไวยากรณ์หาข้อผิดพลาดได้

1.4 แผนการดำเนินการศึกษา

ลำดับที่	รายละเอียด	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
1.	ศึกษาความเป็นไปได้ของระบบ				
2.	ศึกษาการใช้เครื่องมือ UML				
3.	วิเคราะห์และออกแบบระบบ				
4.	พัฒนาโปรแกรม				
5.	ทดสอบและปรับปรุงแก้ไข				
6.	สรุปผลการศึกษาและเขียนรายงาน				

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

- 1.5.1 สามารถนำแนวคิดและทฤษฎีต่าง ๆ ในการออกแบบเชิงวัตถุมาใช้ทำโครงการและนำไปประยุกต์ใช้กับงานจริงได้
- 1.5.2 สามารถนำ CASE TOOL มาช่วยในขั้นตอนการทำงานให้รวดเร็วมากขึ้น

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 แนวคิดเกี่ยวกับการพัฒนาระบบสารสนเทศเชิงออบเจกต์

การพัฒนาระบบสารสนเทศและการเขียนโปรแกรมในปัจจุบันได้หันมาใช้เทคโนโลยีแบบออบเจกต์มากขึ้น เนื่องจากความต้องการของระบบที่มีความซับซ้อนมากขึ้นเรื่อย ๆ ในขณะที่ช่วงเวลาการพัฒนาต้องใช้น้อยลงกว่าในอดีตมาก ดังนั้นจึงต้องมีการหาเทคโนโลยีใหม่ ๆ มาใช้ในการวิเคราะห์ ออกแบบ พัฒนา และบำรุงรักษาระบบ ให้มีคุณภาพและรวดเร็วขึ้น เครื่องมือที่สำคัญในการวิเคราะห์และออกแบบระบบเชิงออบเจกต์และภาษาสำหรับจำลองแบบระบบโดยใช้ภาพ (Visual Modeling) มีความสำคัญมาในการวิเคราะห์ การบันทึกและการสื่อสารความต้องการของระบบ ภาษาการจำลองของระบบที่ได้รับการยอมรับจากวงการคอมพิวเตอร์ทั่วไปคือภาษา UML (Unified Modeling Language) โค้ดแกรมต่าง ๆ ที่เขียนขึ้นโดยใช้สัญลักษณ์มาตรฐาน UML นี้จะช่วยในการสื่อสารและบันทึกภาพของระบบสารสนเทศที่มีความซับซ้อนโดยนำเสนอในมุมมองต่าง ๆ

การวิเคราะห์ ออกแบบ และพัฒนาระบบเชิงออบเจกต์ที่เรียกว่า “Object-Oriented Paradigm” คือแนวทางของเทคโนโลยีในการแก้ไขปัญหาหนึ่งโดยวิธีการแตกปัญหาใหญ่ให้กลายเป็นออบเจกต์ย่อยการจัดกลุ่มของปัญหาในลักษณะนี้แตกต่างไปจากวิธีการพัฒนาแบบดั้งเดิมที่เรียกว่า “Procedural Oriented Paradigm” วิธีการเชิงออบเจกต์นี้จะจัดกลุ่มข้อมูลพื้นฐานและฟังก์ชันพื้นฐานที่กระทำกับข้อมูลนั้นไปพร้อม ๆ กันและจัดกลุ่มกันในรูปของออบเจกต์ นอกจากนี้ยังมีหลักการของการซ่อนรายละเอียดของข้อมูลหรือวิธีการภายในออกจากอิทธิพลของโปรแกรมภายนอก (Information Hiding) เหล่านี้เป็นส่วนประกอบที่ทำให้นักพัฒนาโปรแกรมสามารถจัดการกับโปรแกรมที่มีขนาดใหญ่ และซับซ้อนได้ดีกว่าวิธีการพัฒนาโปรแกรมแบบเชิงโครงสร้าง การออกแบบจึงต้องมีการเปลี่ยนแนวคิดพื้นฐานในการพัฒนาจากวิธีการดั้งเดิม ปัจจุบันเป็นที่ยอมรับกันว่าวงจรหรือโมเดลในการพัฒนาระบบควรจะเป็นในแนวของการวนรอบและการเพิ่มพูน (Iterative and Incremental Model) และมีการประยุกต์ใช้หลักการของสถาปัตยกรรมซอฟต์แวร์ (Software Architecture) หรือคือหลักการในการออกแบบระบบในภาพรวมในรูปแบบต่าง ๆ ด้วย

โดยที่มีการประยุกต์ใช้ภาษาแบบจำลอง UML ในการนำมาช่วยในกระบวนการวิเคราะห์ ออกแบบ และพัฒนาระบบในทุกขั้นตอนด้วย .

2.2 ความรู้พื้นฐานเกี่ยวกับเทคโนโลยีออปเจกต์

หลักการพื้นฐานของออปเจกต์ที่สำคัญคือการวิเคราะห์และออกแบบระบบ กระทำโดยการพิจารณาองค์ประกอบรวมของทั้ง “ข้อมูล” หรือ “แอตทริบิวต์” (Attributes) (หรือบางที่เรียกว่า “คุณสมบัติ - Properties” หรือ “ฟิลด์ของข้อมูล-Data Field”) และพฤติกรรม (Behavior) หรือ “ฟังก์ชันพื้นฐาน” (Function หรือเมธอด Methods) ที่กระทำกับข้อมูลนั้นไปพร้อมกัน โดยการจัดเป็นกลุ่มของโปรแกรมที่สมบูรณ์ในตัวเองเรียกว่าเป็น “คลาส” และ “ออปเจกต์” พื้นฐานสำคัญของการทำงานในระบบเชิงออปเจกต์ก็คือ การที่ออปเจกต์สามารถสื่อสารต่อกันและกันได้ โดยออปเจกต์หนึ่งอาจจะส่งข่าวสาร (เมสเสจ-Message) ไปเรียกใช้งานเมธอร์ดในอีกออปเจกต์หนึ่งได้และเนื่องจากคุณสมบัติของออปเจกต์ที่เรียกว่า “Encapsulation” เป็นหลักการที่ช่วยซ่อนรายละเอียดของทั้งข้อมูลและวิธีการของเมธอร์ดภายในซึ่งผู้ใช้ไม่จำเป็นต้องทราบแต่ใช้งานออปเจกต์นี้ได้ เพียงแค่รู้จักชื่อของเมธอร์ดและอาร์กิวเมนต์ที่ใช้

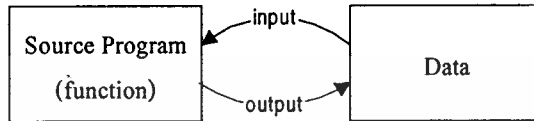
Object Oriented Software Engineering (OOSE) เป็นการพัฒนากระบวนการคอมพิวเตอร์ซึ่งรวมไปถึงการวิเคราะห์ ออกแบบ การพัฒนาโปรแกรม และการนำไปใช้ ซึ่ง OOSE มองว่าการพัฒนาโปรแกรมคือการแก้ปัญหา

OOSE ประกอบด้วย 3 ขั้นตอนใหญ่ คือ

1. Object-Oriented Analysis (OOA) เป็นขั้นตอนการวิเคราะห์เพื่อให้ทราบว่า Problem Domain คืออะไร และเพื่อทำความเข้าใจในรายละเอียดของปัญหาเหล่านั้น เป็นการหาคำตอบให้กับคำถามที่ว่า “อะไรคือปัญหาที่จะถูกแก้ไข”
2. Object-Oriented Design (OOD) เป็นขั้นตอนการออกแบบหรือจำลอง (Model) วิธีการเพื่อแก้ปัญหาใน Problem Domain ซึ่งเป็นการหาคำตอบให้กับคำถามที่ว่า “จะแก้ปัญหานี้ได้อย่างไร”
3. Object-Oriented Programming (OOP) เป็นขั้นตอนการสร้างหนทางแก้ปัญหาในรายละเอียดให้เกิดขึ้นและใช้งานได้จริง เป็นการตอบคำถามที่ว่า “วิธีการแก้ปัญหานี้ทำอย่างไร”

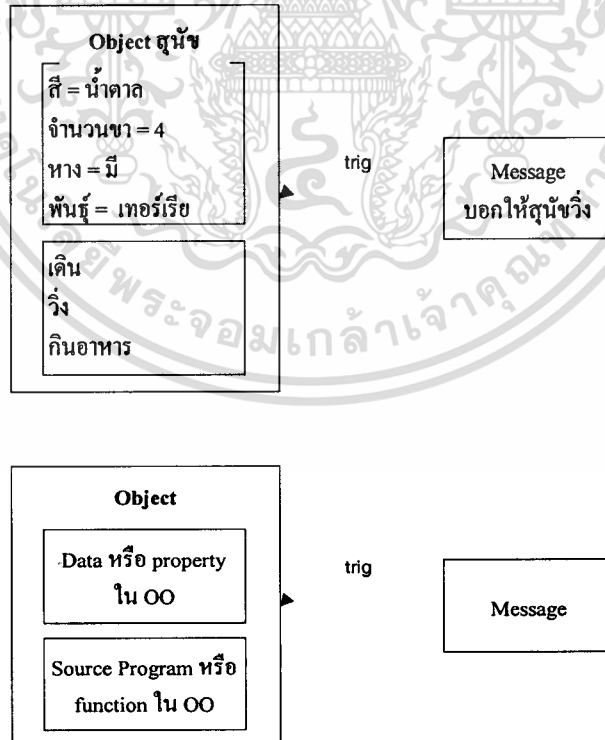
แนวคิดแบบ OO ในลักษณะการโปรแกรม

จากการโปรแกรมแบบเดิมที่เป็นแบบ Structure คือมีการโค้ดเป็นลำดับ, วนลูป, เงื่อนไข นี้คือตัวโปรแกรมโดยแยกออกมาจากข้อมูลที่จะทำงานเมื่อโปรแกรมรันก็จะทำงานตาม step ไปเรื่อย ๆ เมื่อต้องการข้อมูลก็อ่านเข้ามาต้องการแก้ไขก็เขียนออกไป



รูปที่ 2.1 แนวคิดการโปรแกรมแบบเดิม

แต่ในการโปรแกรมแบบ OO จะมองทุกอย่างให้เป็นเหมือนวัตถุที่อยู่รอบตัวเรา เช่น หนังสือ, ปากกา, รถยนต์ โดยข้อแตกต่างที่สำคัญจากการโปรแกรมแบบ structure ก็คือจะรวมตัว source program (function) และ data เข้าอยู่ด้วยกันใน object การจะให้ object ทำงานต้องมี message บางอย่างมาบอกให้ object ทำ function



รูปที่ 2.2 แสดง Object ที่ instance จากคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีคำศัพท์ที่ต้องรู้อย่างน้อย 2 คำคือ object class, object instance

object class หมายถึงเพียงบอกแต่ว่าเป็นชนิดของวัตถุนั้น ๆ เช่น สุนัข ตัว class จะต้องบอกว่าจะไรบ้างที่บอกลักษณะการเป็นสุนัขคือมีสี่อะไร, มีกึ๋นขา, มีหางหรือเปล่า, พันธุ์อะไร และตัว class จะต้องบอกหน้าที่ของสุนัขนั้นทำอะไรได้บ้าง เช่นเดินได้, วิ่งได้, กินอาหาร

object instance จะหมายถึงการนำชนิดของวัตถุมาแทนค่าแล้วเป็นวัตถุจริง ๆ เช่นจากตัวอย่างข้างบนเมื่อ instance แล้วได้สุนัข 1 ตัวพันธุ์เทอร์เรีย มี 4 ขามีหางสีน้ำตาลสามารถเดิน, วิ่ง, กินอาหารได้

ตัวอย่างการประกาศ class ของ JAVA

```
class dog
{
    // กำหนดตัวแปรเพื่อกำหนดคุณสมบัติของคลาสสุนัข
    string color;
    int leg;
    boolean trail;
    string variety;

    // กำหนดฟังก์ชันการทำงานของคลาสสุนัข
    walk()
    { // สุนัขเดิน
    }

    run()
    { // สุนัขวิ่ง
    }

    eat()
    { // สุนัขกินอาหาร
    }
}
```

นี่คือการประกาศคลาสที่มีคุณสมบัติและการทำงานของสุนัขต่อมาเราต้อง instance สุนัขให้มีชีวิตจริง ๆ ขึ้นมาสามารถ instance สุนัขได้หลายตัวจากคลาส

ตัวอย่างการ instance

```
Mydog = new dog();
```

```
Yourdog = new dog();
```

ตัวอย่างการเรียกใช้ฟังก์ชันให้สุนัขวิ่ง, กิน

```
Mydog.run;
```

```
Yourdog.eat;
```

OO มีคุณสมบัติที่สำคัญ 4 อย่าง

1. Abstraction – ความสามารถในการอธิบายโครงสร้างชีวิตจริงได้ คือการนิยามคลาสต้องเป็นภาษาแท้ ๆ ไปเลยเช่น Java (ไม่ใช่ javascript) คือไม่ว่าจะเป็นอย่างไร โปรแกรมที่ออกมาโครงสร้างต้องเป็นรูปแบบคลาสไปเลย
2. Encapsulation – โครงสร้างของโปรแกรมต้องจบในตัวเองเป็นก้อน ทำงานด้วยตนเอง แต่ก็สามารถแยกส่วนออกมาทำงานได้เป็นอิสระเช่นกัน เช่น ถ้าสุนัขเป็น object ที่เราออกแบบ และมีตัวหมัด (อีก object) หนึ่งซ่อนอยู่ในสุนัขและเราก็สามารถนำตัวหมัดไปใส่ในตัว object แมวได้ เพราะข้อมูลถูกอัดเป็นแคปซูลรวมอยู่ในตัวเดียว
3. Inheritance – การถ่ายทอดทางพันธุกรรมได้ ถ้ามี object อยู่แล้วเช่น เครื่องยนต์โตโยต้า ถ้าจะสร้างรุ่นใหม่ขึ้นมาก็สามารถนำเครื่องยนต์รุ่นเก่ามาถ่ายทอดพันธุกรรมได้เลยแล้วคิดแปลงเพียงเล็กน้อยก็กลายเป็นเครื่องรุ่นใหม่ขึ้นมา (reusable)
4. Polymorphism – การแตกหน่อในวิธีการทำและเอกลักษณ์ poly หมายถึงหลายรูปแบบ, morph คือการแปลงสภาพ เช่นกำหนด object บินได้ แมลงวันก็บินได้ (สันสะท้อนปีก) ส่วนนกก็บินได้ (กระพือปีกแบบแอร์โร ไดนามิค) ถ้าได้โปรแกรมสร้างคลาส TflyingObject แต่มีการแตกหน่อเป็น 2 สายพันธุ์ ภาษาที่อ้างเป็น OO ต้องมีคุณสมบัติข้อนี้จะทำให้เกิดการกระจายไปอย่างไม่จำกัดแต่ที่สำคัญคือการออกแบบที่ดีใน OOP อย่างแท้จริง

2.3 การวิเคราะห์และออกแบบเชิงออบเจกต์ด้วยภาษาจำลอง Unified Modeling Language (UML)

UML ย่อมาจาก Unified Modeling Language ซึ่งได้รวมสิ่งที่ดีที่สุดจาก

- Data Modeling concepts (Entity Relationship Diagrams)
- Business Modeling (work flow)
- Object Modeling
- Component Modeling

UML เป็นภาษามาตรฐานสำหรับ visualizing, specifying, constructing, และ documenting ที่สร้างขึ้นจากระบบ software-intensive สามารถถูกใช้กับโปรเซสทั้งหมด, ผ่านไปถึงวงจรชีวิตของการพัฒนาและข้ามไปถึงเทคโนโลยีที่ใช้ implement แตกต่างกันไป

แนวความคิดของ UML ใช้ในการ:

- แสดงขอบเขตของระบบและฟังก์ชันงานหลัก โดยการใช้ use cases และ actors
- แสดงว่า use case มีความน่าเชื่อถือกับ interaction diagrams
- นำเสนอโครงสร้างที่เป็น static ของระบบ โดยการใช้ class diagrams
- โมเดล behavior ของ objects กับ state transition diagrams
- แสดงให้เห็นถึง physical implementation architecture ด้วย component & deployment diagrams
- ขยายฟังก์ชันงานด้วย stereotypes

ความเป็นมาของ UML เดือนตุลาคม 1994 Jim Rumbaugh ผู้คิดค้น OMT (Object Modeling Technique) ตีพิมพ์ในปี 1994 ได้ลาออกจาก General Electric มาทำงานร่วมกับ Grady Booch แห่งบริษัท Rational Software เพื่อพยายามรวมวิธีการของ Booch (Booch 's Method) ตีพิมพ์ในปี 1994 กับ OMT มารวมเข้าด้วยกันจนในที่สุดร่าง Unified Method Version 0.8 ประกาศตัวเมื่อเดือนตุลาคมปี 1995 และบริษัท Rational ได้เข้าซื้อกิจการบริษัท Objectory โดย Iva Jacobson ผู้สร้าง OOSE (Object-Oriented Software Engineering) ตีพิมพ์ในปี 1994, 1995 เป็นการนำเอา the use case driven approach มารวมด้วยและชื่อ Objectory ได้กลายเป็นชื่อของกระบวนการใหม่ที่ใช้ UML เป็นมาตรฐานคือ Rational Objectory Process

เหตุผลของการรวมคือหนึ่งเนื่องจากวิธีการทั้งสามได้พัฒนาเข้าหากันเรื่อย ๆ จึงควรจำกัดความแตกต่างที่จะทำให้ผู้ใช้สับสน เหตุผลที่สองเพื่อทำให้เกิดเสถียรภาพในวงการอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์ โดยมีภาษาสื่อความหมายให้เป็นไปในทิศทางเดียวกัน สามเพื่อพัฒนาขีดความสามารถของวิธีการทั้งสาม นำไปสู่การแก้ปัญหาที่วิธีทั้งสามไม่เคยทำได้มาก่อน

ผลที่ได้คือการเปิดตัว UML 0.9 และ 0.91 ในเดือนมิถุนายนและตุลาคม 1996 ตามลำดับ ในระหว่างนี้ Booch, Rumbaugh และ Jacobson ได้เป็นที่รู้จักในนาม “สามเกลอ” (the three amigos)

OMG (Object Management Group) เป็นองค์กรกำหนดมาตรฐานทางเทคโนโลยีออบเจกต์ ได้จัดตั้งคณะกรรมการเพื่อสร้างมาตรฐานวิธีการพัฒนาซอฟต์แวร์อีกครั้งโดยคณะกรรมการดังกล่าวได้ส่ง Request For Proposal (RFP) ไปยังบริษัทต่าง ๆ ซึ่งผลทำให้มีการเสนอ proposal มากมาย ส่วนทาง Rational นั้นได้ก่อตั้ง UML Partners Consortium ประกอบด้วย ดิจิตอล , ฮิวเลตต์-แพคการ์ด , ไมโครซอฟต์ , ไอบีเอ็ม , ออราเคิล , เท็กซัสอินสตรูเมนต์ ได้ร่วมกันร่างข้อกำหนด UML 1.0 เสนอต่อ OMG ในเดือนมกราคม 1997 และอีกกลุ่มหนึ่งนำโดย IBM & ObjectTime เสนอ proposal แยกต่างหากต่อ OMG และต่อมาก็ได้เข้าร่วมกับ UML Partners Consortium นำแนวคิดมารวมกันเพื่อพัฒนาเป็น UML 1.1 นำเสนอต่อ OMG อีกครั้งในเดือนกันยายน 2540

ไคอะแกรมของ UML คือการที่ไคอะแกรมเหล่านี้ใช้เทคนิค สัญลักษณ์และความหมายของสัญลักษณ์ที่เหมือนกันและสอดคล้องกัน ทำให้ทีมงานที่เกี่ยวข้องในขั้นตอนการพัฒนาตั้งแต่ผู้ใช้ , นักวิเคราะห์ระบบ, ธุรกิจ นักวิเคราะห์และนักออกแบบทางคอมพิวเตอร์ รวมถึงโปรแกรมเมอร์สามารถสื่อสารด้วยภาษาสัญลักษณ์ที่ต่อเนื่อง มีลักษณะการมององค์รวม ไคอะแกรมทั้ง 9 ชนิดสามารถนำใช้งานเพื่ออธิบายระบบที่สนใจในมุมมองต่าง ๆ ทั้งส่วนที่เป็นโครงสร้างคงที่ของระบบ (Static Structure) และส่วนของระบบที่มีพฤติกรรมและความเคลื่อนไหว (Dynamic Behavior)

ชนิดของไคอะแกรม	ประโยชน์	เป้าหมาย
Use Case Diagram	ปฏิสัมพันธ์ระหว่างผู้ใช้ภายนอกและฟังก์ชันการทำงานหลักภายในระบบ	อธิบายความต้องการทางธุรกิจของระบบ
Class Diagram	แสดงโครงสร้างหยุดนิ่งของระบบในลักษณะความสัมพันธ์ระหว่างออบเจกต์	อธิบายความสัมพันธ์ระหว่างคลาสเพื่ออธิบายรายละเอียดของแต่ละ Use Case
Object Diagram	แสดงโครงสร้างหยุดนิ่งของระบบใน	อธิบายความสัมพันธ์ระหว่างออบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ลักษณะความสัมพันธ์ระหว่างออปเจกต์	เจกต์ จะใช้งานเมื่อต้องการอธิบายเจาะจงด้วยออปเจกต์ให้ภาพที่ชัดเจนกว่าการอธิบายด้วยคลาส
Sequence Diagram	แสดงการปฏิสัมพันธ์ระหว่างออปเจกต์สำหรับแต่ละ Use Case โดยมีลำดับของเวลาด้วย	อธิบายพฤติกรรมและขั้นตอนการทำงานของแต่ละ Use Case โดยแสดงเป็นการโต้ตอบระหว่างออปเจกต์
Collaboration Diagram	แสดงการปฏิสัมพันธ์ระหว่างออปเจกต์โดยไม่มีลำดับของเวลาเข้ามาเกี่ยวข้อง	อธิบายพฤติกรรมการทำงานได้ตอบกันระหว่างออปเจกต์เพื่ออธิบายรายละเอียดของแต่ละ Use Case
Statechart Diagram	แสดงสถานะต่าง ๆ ของแต่ละออปเจกต์ ซึ่งอาจจะเปลี่ยนค่าไปตามกิจกรรมที่เกิดขึ้นในระบบ	ใช้อธิบายพฤติกรรมการทำงานได้ตอบกันระหว่างออปเจกต์เพื่ออธิบายรายละเอียดของแต่ละ Use Case
Activity Diagram	แสดงกระบวนการทำงานทางธุรกิจ หรือแสดงปฏิสัมพันธ์ของกลุ่มของออปเจกต์ เพื่อแสดงลำดับการไหลและกิจกรรมของแต่ละ Use Case หรือกิจกรรมของหลายคลาส	แสดงการไหลของกิจกรรมที่เกิดขึ้นสำหรับแต่ละ Use Case
Component Diagram	แสดงองค์ประกอบหรือไฟล์จริงของระบบ (เช่น ไฟล์ exe และ dll) ที่ออกแบบและสถานที่เก็บ	ใช้เพื่อแสดงโครงสร้างทางฟิสิกส์ หรือโครงสร้างที่เก็บจริงของระบบ
Deployment Diagram	แสดงโครงสร้างขณะรันโปรแกรมในระบบ เช่น โปรแกรมไปติดตั้งที่เครื่องไหนบ้างภายในเครือข่าย	แสดงการนำโปรแกรมที่พัฒนาไปติดตั้งในฮาร์ดแวร์ในระบบ

ตารางที่ 2.1 ชนิดของไดอะแกรมที่มีใช้ใน UML

ชนิดของ ไดอะแกรม	คล้ายคลึงกับไดอะแกรม	ใช้ในขั้นตอน
Use Case Diagram	Context Diagram	Use Case เป็นไดอะแกรมหลังใช้ตั้งแต่ขั้นตอนแรกในการพัฒนาระบบ (การรวบรวมความต้องการ) ไปจนทุกขั้นตอนของการพัฒนา
Class Diagram	Data Model	การวิเคราะห์และการออกแบบระบบ
Object Diagram	Data Model	การวิเคราะห์และการออกแบบระบบ
Sequence Diagram	Process Model	การวิเคราะห์และการออกแบบระบบ
Collaboration Diagram	Process Model	การวิเคราะห์และการออกแบบ
Statechart Diagram	State Transition Diagram	การวิเคราะห์และการออกแบบ
Activity Diagram	Flowchart	การวิเคราะห์และออกแบบระบบ
Component Diagram		การวิเคราะห์สถาปัตยกรรมระบบ การออกแบบการสร้าง การออก
Deployment Diagram		การวิเคราะห์สถาปัตยกรรมระบบ การออกแบบการสร้าง การออก

ตารางที่ 2.2 ชนิดของไดอะแกรมเทียบกับการวิเคราะห์และออกแบบระบบ

บทที่ 3

ไต่อะแกรมของ UML ที่ใช้ในการวิเคราะห์และออกแบบ

UML กำหนดสถาปัตยกรรมของระบบโดยพิจารณาจาก 5 มุมมองคือ

1. Use-Case View เป็นมุมมองที่แสดงหน้าที่ของระบบ โดยจะพิจารณาจากมุมมองของผู้ใช้ (User) ที่อยู่นอกระบบ Use-Case View ประกอบด้วย Use Case Diagram ซึ่งจะอธิบายถึงหน้าที่ของระบบและองค์ประกอบที่เกี่ยวข้อง
2. Logical View เป็นมุมมองที่แสดงการออกแบบภายในระบบเพื่อให้ได้หน้าที่ของระบบ ความต้องการของผู้ใช้ Logical View จะประกอบด้วย Class Diagram, Object Diagram ไต่อะแกรมนี้จะแสดง โครงสร้างที่คงที่ และ State Diagram, Sequence Diagram, Collaboration Diagram และ Activity Diagram อธิบายเป็น โครงสร้างแบบไดนามิก
3. Component View เป็นมุมมองที่แสดงการติดตั้งและความสัมพันธ์ระหว่างหน่วยต่าง ๆ ที่ประกอบกันเป็นระบบ
4. Concurrency View เป็นมุมมองที่แสดงการแบ่งการทำงานของระบบออกเป็นขั้นตอนการทำงานต่าง ๆ ซึ่งสามารถทำงานร่วมกัน จะประกอบด้วย State Diagram, Sequence Diagram, Collaboration Diagram และ Activity Diagram จะอธิบายการควบคุมการทำงานร่วมกันของ กระบวนการ (Process) การสื่อสารระหว่างกระบวนการ
5. Deployment View เป็นมุมมองที่แสดงสถาปัตยกรรมทางกายภาพของระบบ ซึ่งเป็นองค์ประกอบภายในระบบ (คอมพิวเตอร์และอุปกรณ์อื่น ๆ)

ภาพรวมของ UML Diagram คือแบบจำลองแสดงสถาปัตยกรรมของระบบในมุมมองดังกล่าวข้างต้น UML Diagram จะประกอบด้วยไต่อะแกรมต่าง ๆ 9 Diagram มีรายละเอียดดังต่อไปนี้

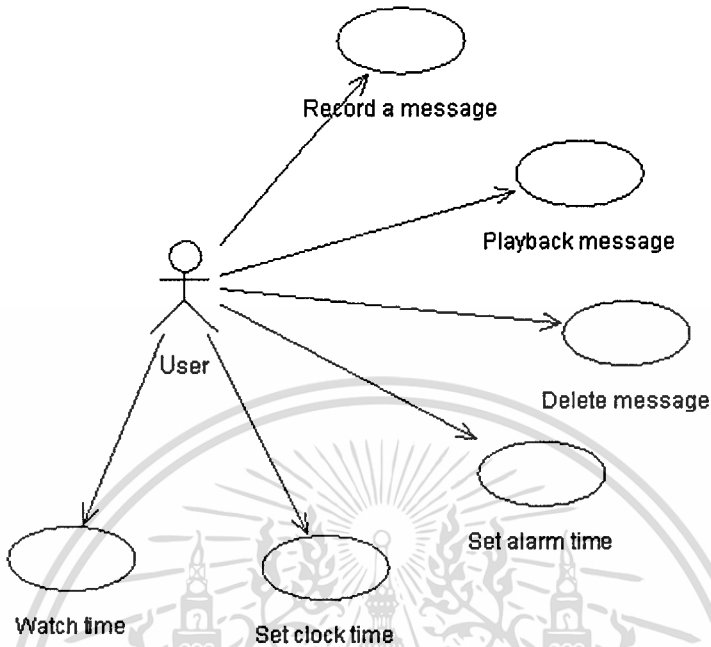
1. Use-Case Diagram

use case เป็นส่วนช่วยให้วิเคราะห์ความต้องการของระบบ วิธีการต่าง ๆ จะเริ่มจากการพิจารณาสถานการณ์ต่าง ๆ ที่จะเริ่มจากการพิจารณาสถานการณ์ต่าง ๆ ที่เกิดขึ้นในระบบแล้วหาความต้องการต่าง ๆ ที่เกิดขึ้นออกมาโดยการบรรยายสถานการณ์ดังกล่าวมักจะใช้คำพูดธรรมดา ไม่มีการบันทึกจนกระทั่ง Ivar Jacobson โดยใช้คำว่า use case โดยกล่าวว่า use case หนึ่ง ๆ ก็คือ การโต้ตอบระหว่างผู้ใช้กับระบบหรือระหว่างระบบด้วยกันในเรื่องใดเรื่องหนึ่ง อาจกล่าวได้ว่า use case นั้นแทนพีเจอร์หนึ่ง ๆ ของซอฟต์แวร์ที่จะสร้างนั่นเอง use case เป็นจุดเริ่มต้นในการบันทึกความต้องการของผู้ใช้ก่อนที่จะเริ่มวิเคราะห์ระบบต่อไป

แผนผัง use case ประกอบด้วย

1. Actors คือบางคนหรือบางสิ่งหรือเป็นอีกระบบหนึ่งที่ต้องติดต่อกับระบบที่จะพัฒนา
2. Use Cases คือรูปแบบของพฤติกรรมที่ระบบแสดงแต่ละ use case คือลำดับของความสัมพันธ์ Transactions แสดง โดย actor และระบบในการสนทนากัน

Use case diagrams ถูกสร้างเพื่อเป็นมุมมองที่สัมพันธ์กันระหว่าง actors และ use cases ซึ่ง Use case diagram ประกอบด้วย Use-Case, External Actor (Actor) และ Relationship โดยแต่ละ use case แสดงแต่ละหน้าที่ของระบบ แต่ละ Actor แสดงแต่ละองค์ประกอบที่มีความสัมพันธ์กับ Use-Case (ผู้ใช้อุปกรณ์ เพิ่มข้อมูล ระบบอื่น ๆ) ในขณะที่ Relationship แสดงความสัมพันธ์แบบต่าง ๆ ระหว่าง Use-Case กับ Use-Case, Use-Case กับ Actor และ Actor กับ Actor ตัวอย่างของ Relationship ซึ่งได้แก่ Generalization, Association และ Dependency



รูปที่ 3.1 Use Case Diagram

จากรูปแสดงตัวอย่าง Use-Case Diagram ระบบ Digital Sound Recorder ประกอบด้วย Use-Case ดังนี้

1. Record a message ผู้ใช้เลือกช่องสำหรับบันทึกเสียงจากไดเรกทอรีและกดปุ่ม “Record” ถ้าช่องสำหรับบันทึกเสียงมีข้อมูลเก็บบอยู่ ข้อมูลจะถูกลบ โดยระบบจะเริ่มบันทึกเสียงจากไมโครโฟนจนกระทั่งกดปุ่ม “Stop” หรือเมมโมรีเต็ม
2. Playback a message ผู้ใช้เลือกช่องสำหรับบันทึกเสียงและกดปุ่ม “Play” ถ้ามีการบันทึกไว้ก็จะออกเสียงที่ลำโพงจนกระทั่งจบหรือผู้ใช้กดปุ่ม Stop
3. Delete a message ผู้ใช้เลือกช่องสำหรับบันทึกเสียงแล้วกดปุ่ม “Delete” เสียงที่ได้บันทึกไว้จะถูกลบจากเมมโมรีและจะสามารถนำเมมโมรีที่ว่างมาใช้ใหม่ได้
4. Set the alarm time ผู้ใช้สามารถปิดเปิดสัญญาณเตือนได้
5. Set the clock time ผู้ใช้สามารถตั้งนาฬิกาได้
6. Watch the time ระบบจะแสดงเวลาและวันที่ปัจจุบันที่หน้าจอ

จากรูปมี Actor ดังนี้

1. User เป็นผู้ใช้งาน Digital Sound Recorder

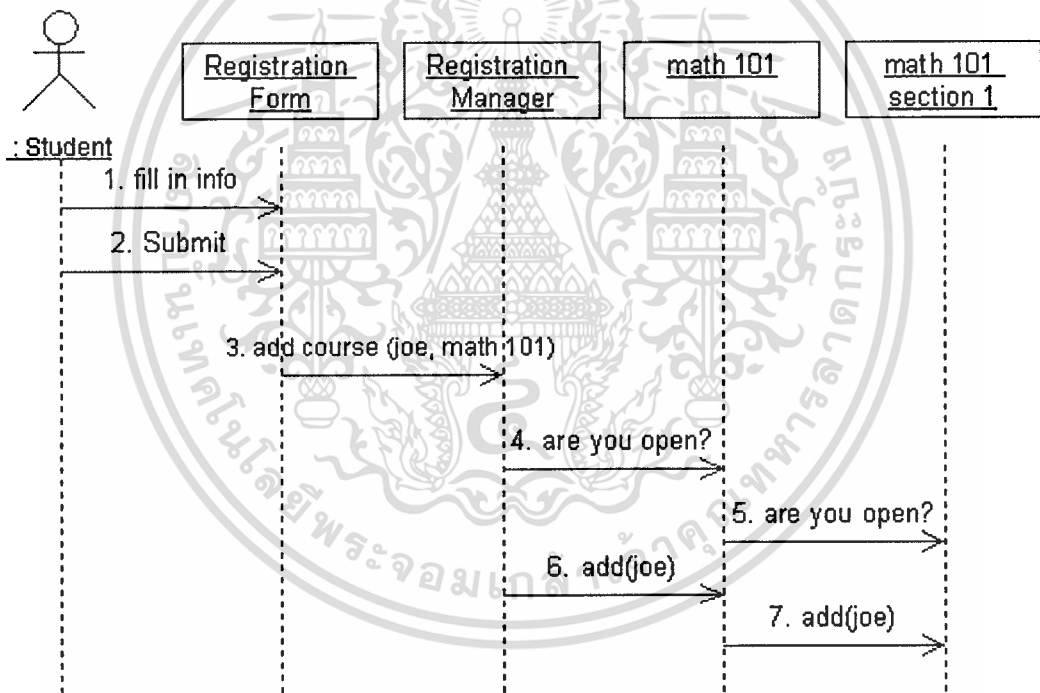
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ use case ยังสามารถเชื่อมโยงถึงกันเองได้โดยความสัมพันธ์หนึ่งใน 2 วิธีคือ การขยาย (extends) และการใช้งาน (uses) โดยมีป้ายบอกกำกับไว้ที่เส้นเชื่อม <<extends>> จะใช้เมื่อ use case หนึ่งเป็นกรณีเฉพาะหรือทำอะไรเพิ่มเติมจาก use case ที่ถูกขยายนั่น ส่วน <<uses>> จะใช้ในกรณีที่ use case หลายอันมีการทำงานบางอย่างนั้นออกมาเป็น use case ต่างหากให้ use case ที่เหลือใช้งาน บางครั้งอาจมองดูการใช้ <<extends>> และ <<uses>> จะคล้ายกันในแง่ที่เป็นการเอางานออกมาใช้เหมือนกันแต่ความแตกต่างที่ทางความหมายคือกลุ่มของ use case ที่ใช้ <<extends>> นั้น actor จะกระทำกับ use case ทุกตัวที่ได้ขยายออกไป แต่ในกลุ่ม use case ที่ใช้ <<uses>> นั้น actor จะติดต่อโดยตรงกับ use case เดียวที่โยงถึงอยู่เท่านั้น



2. Sequence Diagram

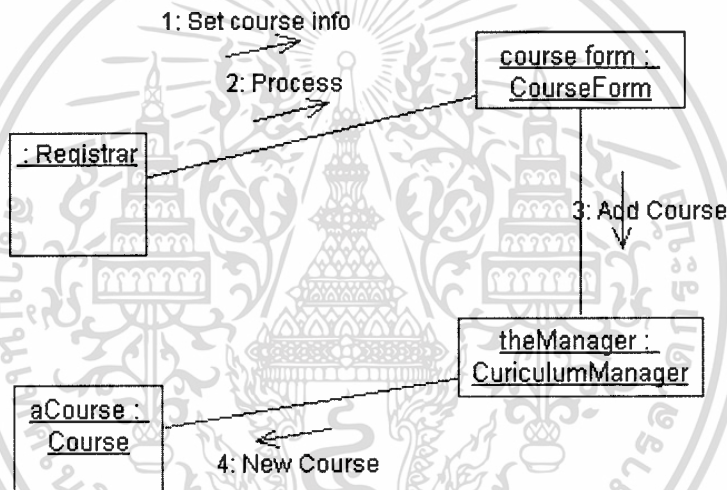
Sequence Diagram จะแสดงถึง object interactions เป็นการโต้ตอบกันระหว่างออบเจกต์ โดยการจัดให้อยู่ใน time sequence หมายถึงการแสดงการปฏิสัมพันธ์ (Interaction) ระหว่าง Object ตามลำดับของการปฏิสัมพันธ์และเหตุการณ์ต่าง ๆ ที่เกิดขึ้น ณ เวลาที่กำหนด ดังรูป แสดง Sequence Diagram ระบบลงทะเบียนของมหาวิทยาลัย การปฏิสัมพันธ์เริ่มจากการที่ Object ใน Class Student กรอกข้อมูลแล้วส่งข้อมูลโดยใช้ Message Submit ติดต่อกับออบเจกต์ในคลาส Registration form ซึ่งติดต่อกับออบเจกต์ ในคลาส Registration manager โดยการใช้ Message Add Course การปฏิสัมพันธ์จะดำเนินไปจนกระทั่งขั้นตอนสุดท้ายคือ Object ใน Class Math ทำการเพิ่มจำนวนนักศึกษาสำหรับชั้นเรียน Math101 Section 1 โดยการใช้ Message Add



รูปที่ 3.2 Sequence Diagram

3. Collaboration Diagram

Collaboration diagram แสดง object interactions ทำให้เป็นระบบรอบ ๆ objects และเชื่อมโยงไปยัง objects อื่นๆ Collaboration Diagram เป็น Diagram ซึ่งแสดงการปฏิสัมพันธ์ (Interaction) ระหว่าง Object ตามลำดับของการปฏิสัมพันธ์และเหตุการณ์ต่าง ๆ ที่เกิดขึ้น ณ เวลาที่กำหนด เช่นเดียวกับ Sequence Diagram แต่ Collaboration Diagram จะเน้นหนักในการอธิบายบริบท (Context) ของระบบ ดังรูปที่ 3.3 แสดง Collaboration Diagram ระบบลงทะเบียนของมหาวิทยาลัยซึ่งแสดงลำดับของการปฏิสัมพันธ์ในการเพิ่มข้อมูลวิชา (Create A Course)

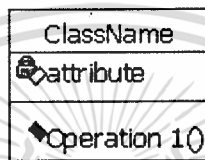


รูปที่ 3.3 Collaboration Diagram

4. Class Diagrams

Class Diagram ถือเป็นไดอะแกรมที่สำคัญที่สุดในการออกแบบออบเจกต์ วิธีการทางออบเจกต์ทุกตัวจะละเว้น Class Diagram ไม่ได้ซึ่งจะใช้อธิบายชนิดต่าง ๆ ของออบเจกต์ พร้อมทั้งความสัมพันธ์แบบสถิต (static relationship) ระหว่างออบเจกต์ด้วย

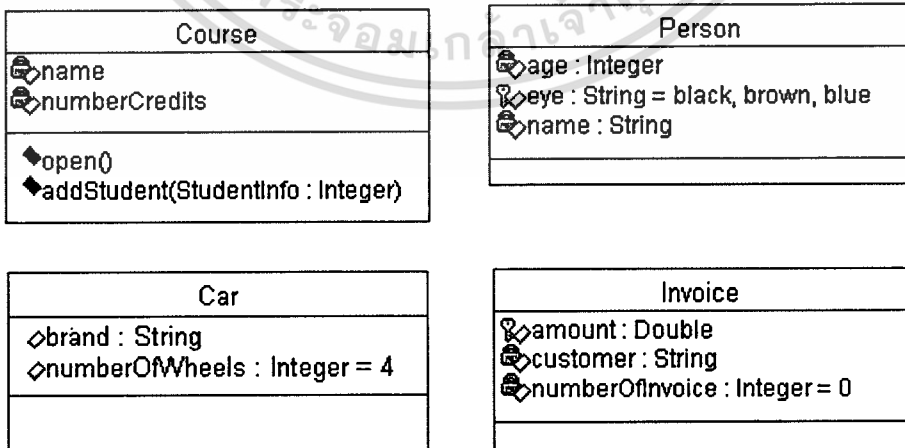
- class diagram ถูกใช้เพื่อแสดง classes และ relationship ของคลาส
- class diagram สร้างเป็นโครงสร้างภายในของระบบที่คงที่



รูปที่ 3.4 Class

คลาสโดยทั่วไปจะแบ่งเป็น 3 ส่วน

1. ชื่อของคลาส
2. แอตทริบิวต์ ในโครงสร้างของ class ถูกนำเสนอโดยแอตทริบิวต์ของคลาส แอตทริบิวต์อาจถูกพบโดยการตรวจสอบ class definitions, problem requirements, และการประยุกต์ domain knowledge
3. โอเปอเรชัน Operations พฤติกรรมของ class ถูกแสดง โดยการทำงานของคลาส การทำงานอาจถูกพบโดยการตรวจสอบ interaction diagrams



รูปที่ 3.5 แสดงรูปคลาสต่าง ๆ

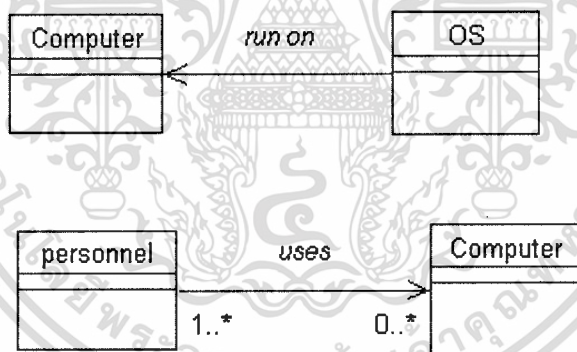
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางในการหาคลาส ให้พิจารณาจากข้อมูลที่ต้องการเก็บ แปรรูปวิเคราะห์ และอื่น ๆ พิจารณาระบบภายนอกซึ่งระบบที่จะพัฒนามีการติดต่อกับ และพิจารณาจากรูปแบบ (Pattern), Class Library และองค์ประกอบของระบบอื่น ๆ ที่เคยศึกษาหรือพัฒนา , อุปกรณ์ที่มีในระบบ, แผนผังองค์การ, บทบาทของ Actor

ความสัมพันธ์

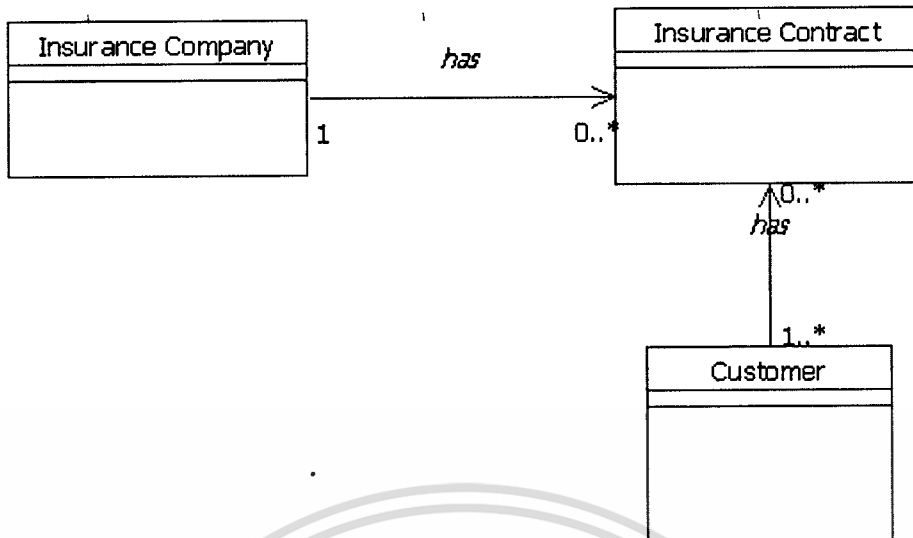
Relationships ได้จัดเตรียมเส้นทางติดต่อสื่อสารระหว่าง objects ความสัมพันธ์ระหว่างคลาสมี 4 ชนิดคือ: (Association , Aggregation, Generalization, Dependency)

1. **Association** คือการเชื่อมต่อแบบ bi-directional ระหว่างคลาสในลักษณะของการรู้จักซึ่งกันและกัน ใน association หนึ่ง ๆ จะมีบทบาท (role) ของคลาสอยู่สองบทบาทในทิศทางตรงข้ามกัน กล่าวคือใน association ระหว่างคลาส A กับคลาส B จะมี role จาก A ไป B และ role จาก B ไป A สัญลักษณ์ที่ใช้แทน association เป็นเส้นตรงเชื่อมระหว่าง 2 คลาสซึ่งมี role กำกับไว้ที่ปลายทั้งสองข้าง ซึ่งไม่จำเป็นต้องระบุเสมอไป



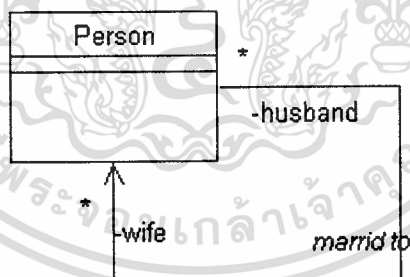
รูปที่ 3.6 ความสัมพันธ์ของคลาสแบบ association

จากรูปความสัมพันธ์ของคลาส Computer และ OS จะบอกได้ว่ามี OS มารันบน Computer ต่อมา เราสามารถใส่ multiplicity เพิ่มขึ้นได้ ความสัมพันธ์ของ 2 คลาสคือคลาสบุคคลที่ใช้คอมพิวเตอร์ โดยมีความสัมพันธ์ในลักษณะ many-to-many บุคคล 1 คนอาจจะใช้เครื่องคอมพิวเตอร์หลายเครื่องหรือไม่ใช้เลย และในทางกลับกันคอมพิวเตอร์ 1 เครื่องอาจถูกใช้หลายคนหรือคนเดียว



รูปที่ 3.7 ความสัมพันธ์ของคลาสแบบ association ที่มี Cardinality

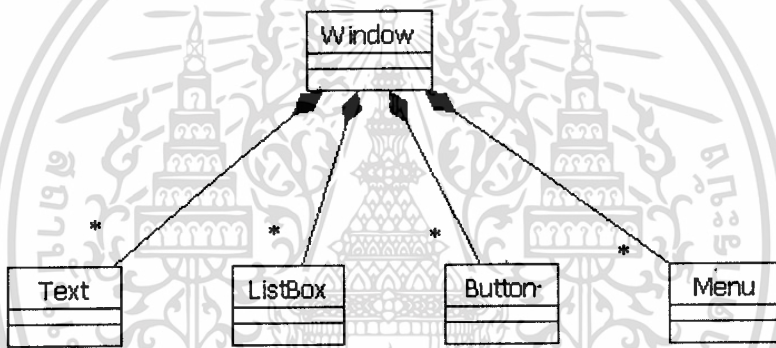
ความสัมพันธ์อย่างง่ายของธุรกิจประกันภัยคือมีบริษัทประกันภัย 1 บริษัทสามารถมีสัญญาประกันภัยได้ตั้งแต่ศูนย์จนถึงหลายสัญญา แล้วลูกค้าบริษัทประกันภัยก็สามารถที่จะมีสัญญาประกันภัยตั้งแต่ศูนย์จนถึงหลายสัญญาได้ สัญญาประกันภัย 1 สัญญาที่ทำต่อหนึ่งบริษัทประกันภัยและหนึ่งสัญญาสามารถที่จะทำกับลูกค้า 1 คนจนถึงหลาย ๆ คนได้



รูปที่ 3.8 แสดงถึงความสัมพันธ์แบบ recursive association

ความสัมพันธ์ที่เป็นไปได้ว่ามีความสัมพันธ์กับคลาสเดียวกันเรียกว่า recursive association จากรูปข้างบนสามีอาจแต่งงานกับภรรยา ซึ่งทั้งสามีและภรรยาเป็นบุคคลทั้งคู่ ถ้าบุคคลที่ไม่ได้แต่งงานซึ่งทั้งคู่จะไม่สามารถใช้กฎของสามีหรือภรรยาได้ ซึ่งความสัมพันธ์แบบนี้อาจจะไม่ค่อยเหมาะสมนัก

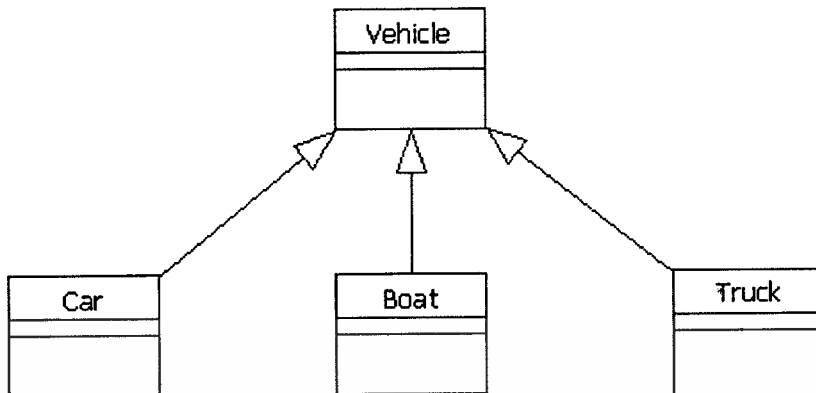
2. **Aggregation** เป็นความสัมพันธ์แบบ Whole/Part ซึ่งจะมีความสัมพันธ์ที่เคร่งครัดกว่า association กล่าวคือ Part เป็นส่วนหนึ่งของ Whole แต่ Whole ไม่เป็นส่วนหนึ่งของ Part ความสัมพันธ์แบบ aggregation ถูกแสดงด้วยเส้นตรงเชื่อมความสัมพันธ์ระหว่างคลาสด้วยรูปสี่เหลี่ยมขนมเปียกปูนต่อไปยังคลาสที่แสดงอยู่ทั้งหมด (Whole)
- Compositon Aggregation** เป็นความสัมพันธ์แบบ Whole/Part มีข้อกำหนดว่า Whole ต้องรับผิดชอบ Part ทุกอย่าง อายุขัยของ Part จึงขึ้นกับ Whole สัญลักษณ์ที่ใช้ใน UML จะคล้ายกับ Aggregation แต่ใช้สี่เหลี่ยมขนมเปียกปูนที่



รูปที่ 3.9 แสดงความสัมพันธ์ระหว่างคลาสแบบ aggregate

ตัวรูปสี่เหลี่ยมข้าวหลามตัดจะแสดงถึง composition aggregate จากรูปหน้าต่างประกอบ ด้วยหลายเมนู, ปุ่ม, ลิสต์บ็อกและข้อความ การวาดรูปอาจจะวาดเป็นรูปต้นไม้ก็ได้ถ้ามีคลาสมาก ๆ โดยรวมปลายทางเป็นปลายเดียว

3. **Generalization** หรือ Inheritance เป็นความสัมพันธ์แบบ superclass/subclass หรือ base/derived class สัญลักษณ์ที่ใช้แทน Generalization เป็นลูกศรหัวสามเหลี่ยม โปร่งชี้จาก subclass ไปยัง superclass



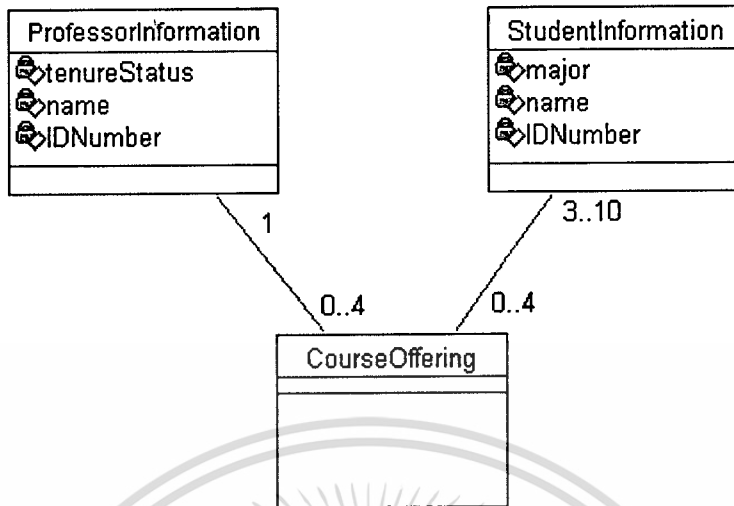
รูปที่ 3.10 แสดงความสัมพันธ์ระหว่างคลาสแบบ Generalization

จากรูป Vehicle เป็น superclass โดย subclass กำเนิดมาจาก superclass โดยการสืบทอด (generalization - specialization)

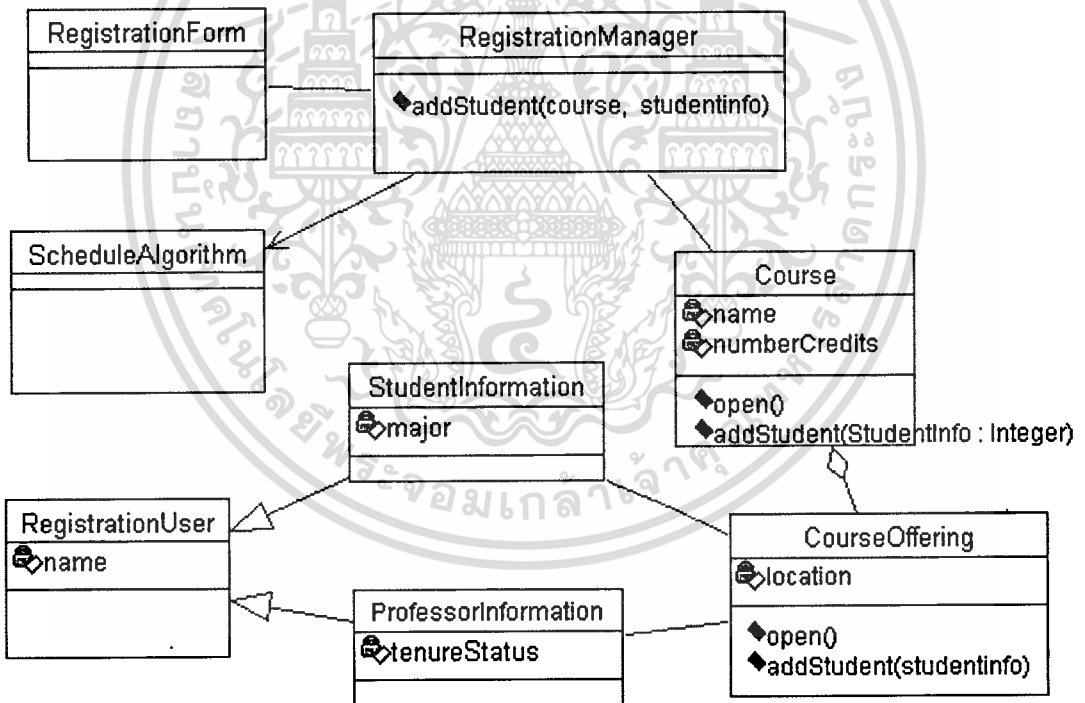
4. **Dependency** หรือ coupling เป็นความสัมพันธ์อย่างอ่อนระหว่างคลาสที่ไม่ได้มี association โดยตรงแต่มีการใช้งานที่ทำให้ขึ้นตรงต่อกัน เมื่อมีการแก้ไขคลาสหนึ่งจึงมีผลไปยังคลาสต่าง ๆ ที่ dependent ต่อคลาสที่แก้ไข ถูกแสดงเป็น dashed line pointing from Class Diagram ประกอบด้วย Class และ Relationship ระหว่าง Class โดยแต่ละ Class แสดงแต่ละองค์ประกอบที่มีในระบบ ในขณะที่ Relationship แสดงความสัมพันธ์แบบต่าง ๆ ระหว่าง Class ซึ่ง ได้แก่ Generalization, Association และ Dependency ดังรูป 3.11 แสดง Class Diagram ระบบลงทะเบียนของมหาวิทยาลัยซึ่งประกอบด้วย Class ดังนี้

1. ProfessorInformation
2. StudentInformation
3. Courseoffering

โดยมีความสัมพันธ์ระหว่าง Class เป็นแบบ Association



รูปที่ 3.11 Class Diagram มีความสัมพันธ์แบบ Association



รูปที่ 3.12 รูปแสดงความสัมพันธ์ระหว่างคลาส

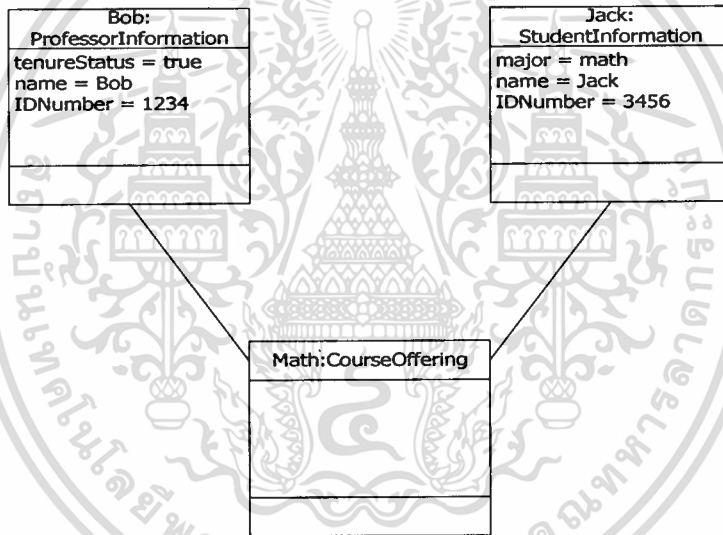
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Object Diagram

Object Diagram ประกอบด้วย Object และ Relationship ระหว่าง Object โดยแต่ละ Object แสดง Instance ของแต่ละ Class ที่มีในระบบ ในขณะที่ Relationship แสดงความสัมพันธ์แบบต่างๆ ระหว่าง Object ซึ่ง ได้แก่ Generalization, Association และ Dependency ดังรูป แสดง Object Diagram ระบบลงทะเบียนนิสิตมหาวิทยาลัยซึ่งประกอบด้วย

1. Bob:ProfessorInformation
2. Jack:StudentInformation
3. Math:Courseoffering

โดยมีความสัมพันธ์ระหว่าง Object เป็นแบบ Association



รูปที่ 3.13 Object Diagram

6. State Diagram

State Diagram ประกอบด้วยสถานะ (State) ต่าง ๆ ของ Object เหตุการณ์ (Event) ต่าง ๆ ที่ทำให้สถานะของ Object เปลี่ยนและการกระทำ (Action) ที่เกิดขึ้นเมื่อสถานะของ Object เปลี่ยน การเขียน State Diagram ใช้รูปสี่เหลี่ยมมุมมนแทน State และมีชื่อของ State ระบุอยู่ จะใช้เครื่องถูกศรแทน Transition โดยลากจาก State เริ่มต้นไปยัง State ที่ต้องการ ที่ถูกศรจะมีชื่อหรือกิจกรรมกำกับอยู่ดังนี้

Condition/Action

Condition หมายถึงเงื่อนไขในการเข้าหรือออกจาก State ส่วน Action หมายถึงกิจกรรมที่ทำระหว่างการเปลี่ยน State นอกจากนี้ใน State ยังสามารถระบุรายละเอียดได้มากขึ้นดังนี้

Entry/Action

Do/Action

Exit/Action

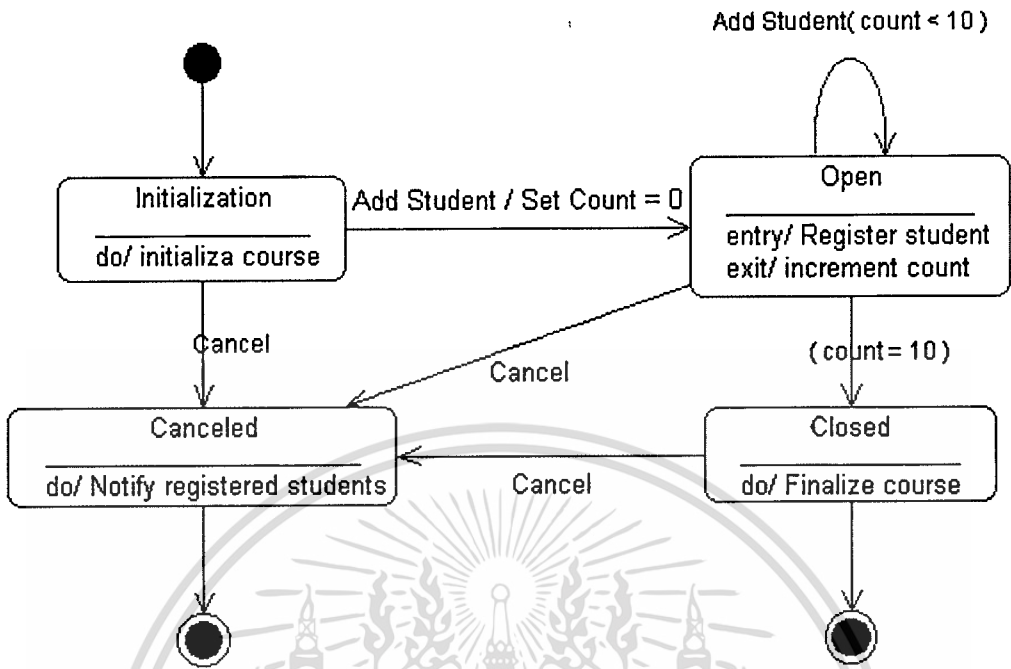
Condition/Action

Entry/Action หมายถึงเมื่อเข้ามายัง State นี้ให้ทำ Action, Do/Action หมายถึงหลังจากเข้ามายัง State นี้แล้วหากไม่มีเงื่อนไขให้ทำกิจกรรม Action, Exit/Action หมายถึง ขณะที่จะออกจาก State นี้ให้ทำกิจกรรม Action และ Condition/Action หมายถึงขณะที่ยังอยู่ใน State นี้เมื่อเกิดเงื่อนไขใด ๆ ที่กำหนดโดย Condition ให้ทำกิจกรรม Action

จากรูปที่ 3.14 แสดง State Diagram ระบบลงทะเบียนของมหาวิทยาลัยซึ่งประกอบด้วยสถานะ

1. Initialization
2. Open
3. Canceled
4. Closed

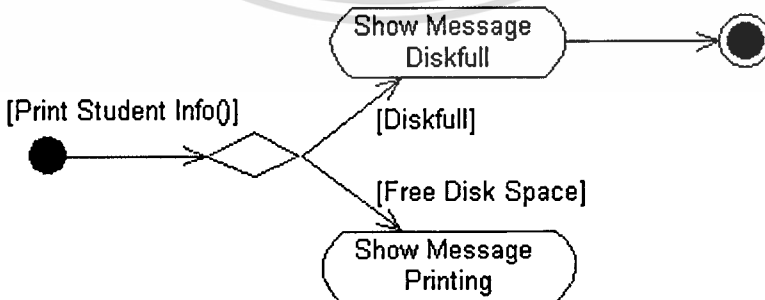
สำหรับเหตุการณ์ที่ทำให้สถานะของ Object เปลี่ยนได้แก่การ Add Student, Set Count และ Cancel สำหรับการกระทำที่เกิดขึ้นเมื่อสถานะของ Object เปลี่ยนได้แก่การ Initialize Course, Register Student, Exit เป็นต้น



รูปที่ 3.14 State Diagram

7. Activity Diagram

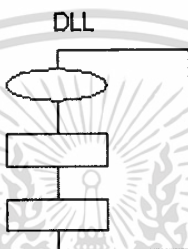
Activity Diagram เป็น Diagram ซึ่งแสดงขั้นตอนการทำงานในการปฏิบัติการ (Operation) โดยประกอบด้วยสถานะ (State) ต่าง ๆ ที่เกิดขึ้นระหว่างการทำงาน และผลจากการทำงานในขั้นตอนต่าง ๆ ดังรูป แสดง Activity Diagram ระบบลงทะเบียนของมหาวิทยาลัยซึ่งแสดงขั้นตอนการพิมพ์ข้อมูลนักศึกษา ในกรณีที่ Disk เต็มจะมีการแจ้งบนจอภาพ ในกรณีที่ Disk มีเนื้อที่พอก็มีการแจ้งบนจอภาพเช่นกัน



รูปที่ 3.15 Activity Diagram

8. Component Diagram

Component Diagram เป็น Diagram ที่แสดงโครงสร้างทางกายภาพของซอฟต์แวร์โดย Diagram จะประกอบด้วยองค์ประกอบของซอฟต์แวร์ซึ่งอยู่ในรูปแบบต่าง ๆ (Text, Binary และ Executable) โดยคอมโพเนนต์ไอคอนจะถูกแสดงแทนด้วยซอฟต์แวร์โมดูลด้วยอินเทอร์เฟซที่อธิบายอย่างชัดเจน ในคอมโพเนนต์จะระบุไว้อย่างชัดเจนว่ามีชนิดเป็นแบบไหน เช่น ActiveX, Applet, Application, DLL และ Executables



รูปที่ 3.16 DLL file ไอคอนที่ถูกใช้แสดงแทน Data Link Library (DLL) file



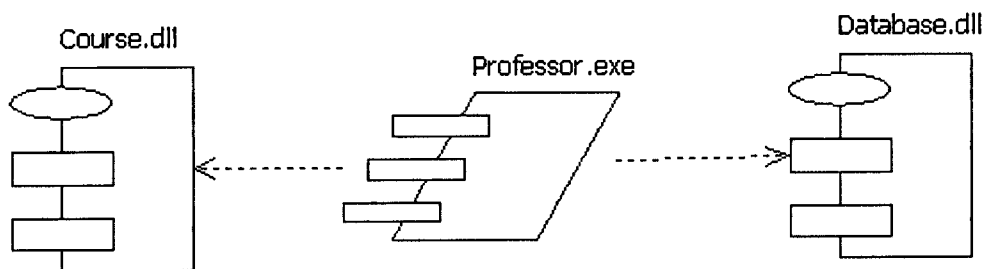
รูปที่ 3.17 ไอคอนที่ใช้แทนเอ็กซีคิวต์เทเบิลไฟล์คือไฟล์ที่มีนามสกุล .EXE

ตัวอย่างแสดง Component Diagram ระบบลงทะเบียนของมหาวิทยาลัยซึ่งประกอบด้วย ส่วนที่เป็น Text ได้แก่ Professorinformation และ Addcourseoffering ส่วนที่เป็น Binary ได้แก่

1. Course.Dll
2. Database.Dll

ส่วนที่เป็น Executable ได้แก่

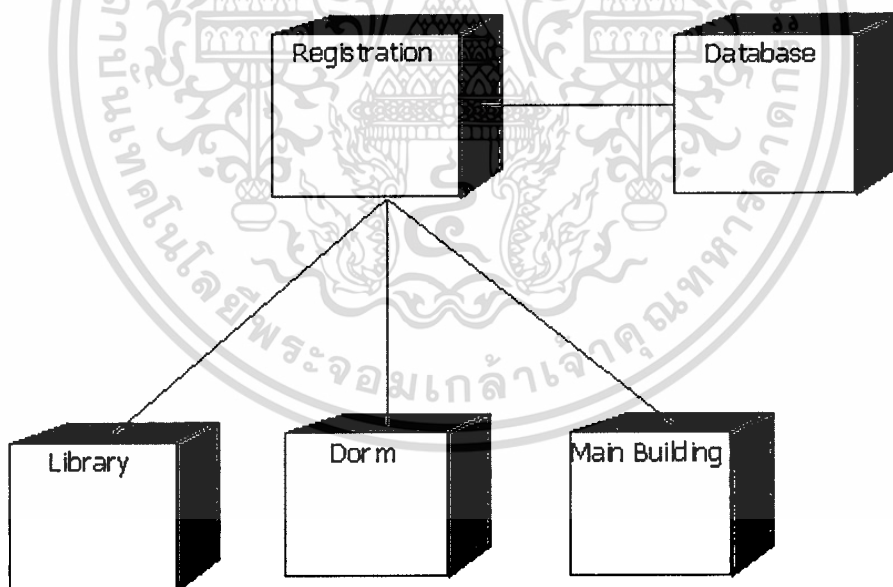
1. Professor.exe จะเห็นว่า Professor.exe จะขึ้นอยู่กับ (Dependency) ส่วนอื่น ๆ



รูปที่ 3.18 Component Diagram

9. Deployment Diagram

Deployment Diagram เป็น Diagram ซึ่งแสดงสถาปัตยกรรมของฮาร์ดแวร์และซอฟต์แวร์ในระบบ ตลอดจนความสัมพันธ์ระหว่างฮาร์ดแวร์และซอฟต์แวร์ ดังรูป แสดง Deployment Diagram ระบบลงทะเบียนของมหาวิทยาลัยซึ่งมีการติดตั้งฮาร์ดแวร์และซอฟต์แวร์กระจายไปยังอาคารต่าง ๆ



รูปที่ 3.19 Deployment Diagram

บทที่ 4

วิธีดำเนินการศึกษา

4.1 วิเคราะห์และออกแบบระบบงานปัจจุบัน

งานโฆษณาประชาสัมพันธ์ของฝ่ายปฏิบัติการด้านการใช้ไฟฟ้า การไฟฟ้าฝ่ายผลิตแห่งประเทศไทยมีหน้าที่และความรับผิดชอบในการเผยแพร่ประชาสัมพันธ์การดำเนินโครงการจัดการด้านการใช้ไฟฟ้าไปสู่สาธารณะ และกลุ่มเป้าหมายต่าง ๆ บริษัทผู้ผลิตร้านค้า ผู้ประกอบการ อาคาร ธุรกิจ อุตสาหกรรม สถาบัน วิศวกร ตลอดจนสื่อมวลชนในทุกแขนงเพื่อสร้างภาพลักษณ์ที่ดีในการดำเนินโครงการ

4.2 วิเคราะห์และออกแบบระบบเชิงวัตถุโดยใช้ UML

4.2.1 แนะนำ

UML ใช้สัญลักษณ์มาตรฐานในการวิเคราะห์และออกแบบซอฟต์แวร์เชิงวัตถุ UML Diagram สามารถทำงานกับระบบซอฟต์แวร์ที่ซับซ้อน UML จะมาตรฐานของสัญลักษณ์ที่มีไคอะแกรมหลาย ๆ แบบ เพื่อความเข้าระบบมากยิ่งขึ้น

4.2.2 วิเคราะห์ความต้องการ

ในระบบงานโฆษณาประชาสัมพันธ์มีความต้องการเก็บข้อมูลทางด้านงบประมาณที่ใช้ง่ายในการผลิตและวางสื่อโฆษณาประชาสัมพันธ์รวมถึงการติดตามงานโฆษณาประชาสัมพันธ์ ซึ่งข้อมูลมีลักษณะของในหนึ่งโครงการจะมีทั้งงานโฆษณาและงานประชาสัมพันธ์ แต่ละงานจะแบ่งได้อีกเป็นสื่อต่าง ๆ ที่นำเสนออย่างเช่น สื่อโทรทัศน์ วิทยุ หรือแผ่นพับ รวมไปถึงอินเทอร์เน็ต

ในระบบงานมีความต้องการเพิ่มขึ้นอีก

1. รายงานสำหรับผู้บริหาร

1.1 การใช้จ่ายเงินค่าโฆษณาประชาสัมพันธ์

1.2 การติดตามข้อมูลการผลิตและวางสื่อโฆษณาประชาสัมพันธ์

- 1.3 สิทธิการวางสื่อในงานโฆษณาประชาสัมพันธ์
- 1.4 ประวัติของโฆษณาประชาสัมพันธ์
- 1.5 ประวัติของบริษัทผู้ผลิตโฆษณาประชาสัมพันธ์
2. สามารถดูข้อมูลสรุปได้จากจอมอนิเตอร์
 - 2.1 รายงานสรุปการใช้จ่ายเงินในโครงการ
 - 2.2 ระยะเวลาการดำเนินโครงการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 Object-Oriented Analysis (OOA)

4.2.3.1 การกำหนด Problem Domain

- ขอบเขตของระบบนี้จะกล่าวถึงเรื่องงบประมาณแต่ละปีในงานโฆษณาประชาสัมพันธ์
- การติดตามสื่อที่เผยแพร่
- การจัดทำรายงานเสนอผู้บริหาร

4.2.3.2 Use Cases และ Actor ที่มีในระบบ

Actor ของระบบ

Actor	คำอธิบาย
Executive	ผู้บริหารเป็นพนักงานคนหนึ่งที่รับตำแหน่งบริหาร
Employee	พนักงานที่บันทึกข้อมูล

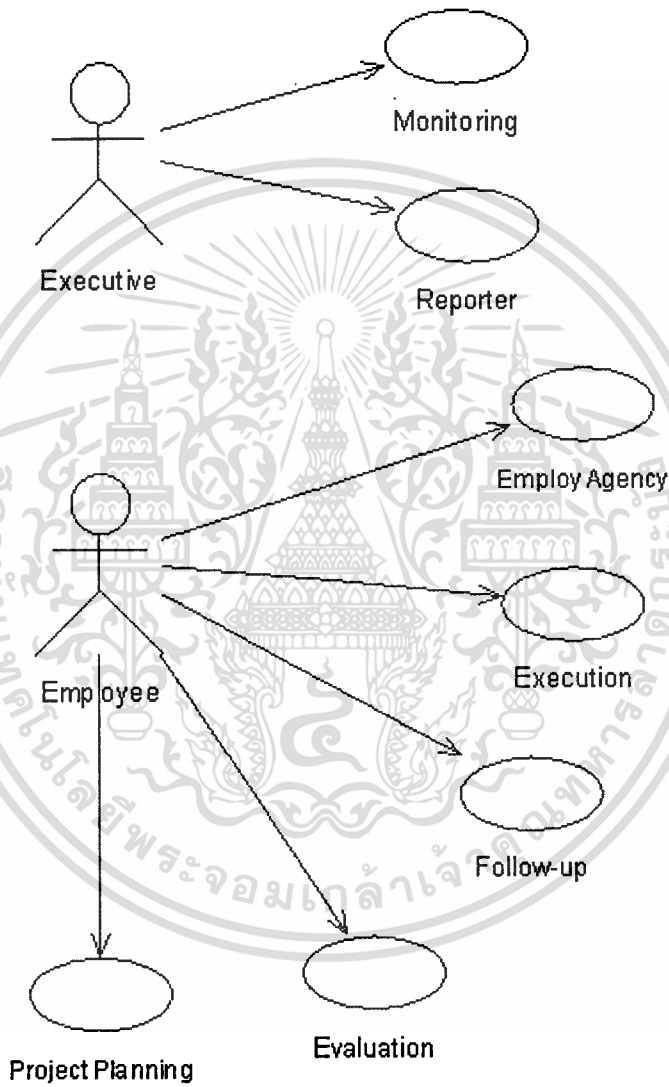
Use case ของระบบ

Use cases อธิบายถึงหน้าที่ของระบบจากมุมมองที่ผู้ใช้งาน จะเป็นบุคคลหรืออุปกรณ์คอมพิวเตอร์ แต่ละ use case จะเป็นฟังก์ชันการทำงานต่างกันในระบบและจะให้ผลลัพธ์ที่ต่างกันในระบบจะมี Use cases ดังตาราง

Use Case	คำอธิบาย
Employ Agency	บันทึกการจัดจ้าง Agency ในการผลิตและวางสื่อ
Execution	บันทึกการการผลิตและวางสื่อ โดยที่หน่วยงานผลิตเองหน่วยงานผลิต
Follow Up	ติดตามการดำเนินงาน
Evaluation	ประเมินผลงานของโฆษณาประชาสัมพันธ์
Project Planning	บันทึกงบประมาณของโครงการ โฆษณาประชาสัมพันธ์
Monitoring	ติดตามข้อมูลการดำเนินงาน
Reporter	รายงานสรุปให้กับผู้บริหาร

4.2.3.3 Use Cases Diagram

เป็นการกำหนดส่วนของ Use Case Diagram ซึ่งมาจาก Actor และ Use Case ที่มีในระบบ ในขั้นตอนนี้จะเป็นการกำหนดรายละเอียดของ Actor ว่ามีใครทำหน้าที่อะไรบ้าง



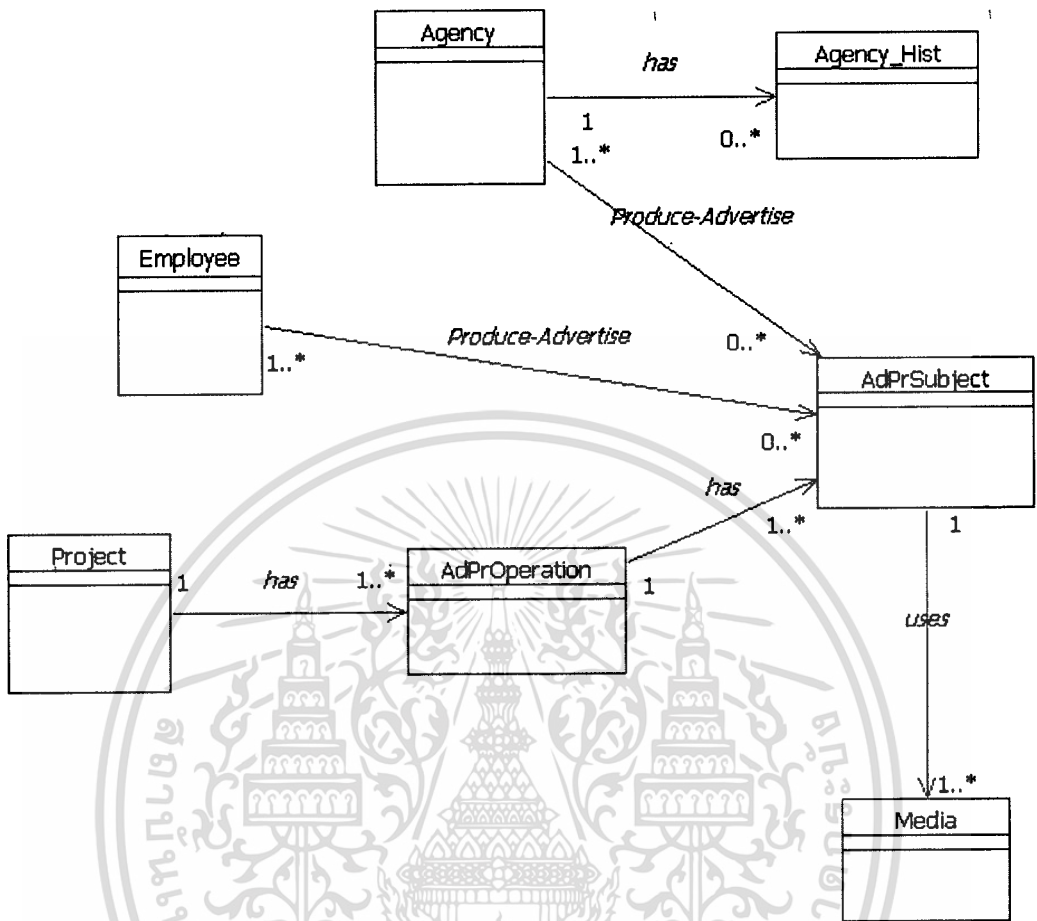
รูปที่ 4.1 Use Case Diagram ของระบบงานโฆษณาประชาสัมพันธ์

4.2.3.4 Class Diagram

หลังจากวิเคราะห์ความต้องการแล้ว ในขั้นตอนนี้จะต้องวิเคราะห์ความต้องการแล้วนำเสนอ โดย Class Diagram ว่าขอบเขตควรมีคลาสอะไรบ้างในขั้นตอนนี้จะต้องสร้าง Class Diagram โดยบอกลักษณะของ Object นั้น ขั้นตอนที่วิเคราะห์นี้ถือเป็นขั้นตอนที่สำคัญขั้นตอนหนึ่งในการศึกษาและทำการออกแบบระบบ โครงสร้างของระบบตามขอบเขตของปัญหาที่เกิดขึ้น จะเป็นการศึกษาระบบต่อจาก Use Case Diagram คือต้องทำการรวบรวมคลาสทั้งหมดที่สัมพันธ์กัน ขึ้นมาเป็นหมวดหมู่คือ Class Diagram เพื่อเป็นการลดความซ้ำซ้อนของระบบ ซึ่งจะอยู่ในส่วนของ Logical View

จากการวิเคราะห์จะได้ Class เบื้องต้นจากระบบงานโฆษณาประชาสัมพันธ์

1. Class Project โครงการงานโฆษณาประชาสัมพันธ์
2. Class AdPrOperation งานโฆษณาหรืองานประชาสัมพันธ์
3. Class AdPrSubject หัวข้อย่อยในงานโฆษณาประชาสัมพันธ์
4. Class Agency บริษัทผู้ผลิตงานโฆษณาประชาสัมพันธ์
5. Class Agency_Hist ผลงานที่ผ่านมาของบริษัทผู้ผลิตงานโฆษณาประชาสัมพันธ์
6. Class Employee รายชื่อพนักงานสังกัดหน่วยงานในองค์กรที่สามารถผลิตหรือวางสื่องานโฆษณาประชาสัมพันธ์
7. Class Media สื่อที่ใช้เผยแพร่ต่าง ๆ



รูปที่ 4.2 Class Diagram พื้นฐานของระบบงานโฆษณาประชาสัมพันธ์

จาก Abstraction ที่มีอยู่ใน Class Diagram พื้นฐานจะสามารถอธิบายได้ว่า

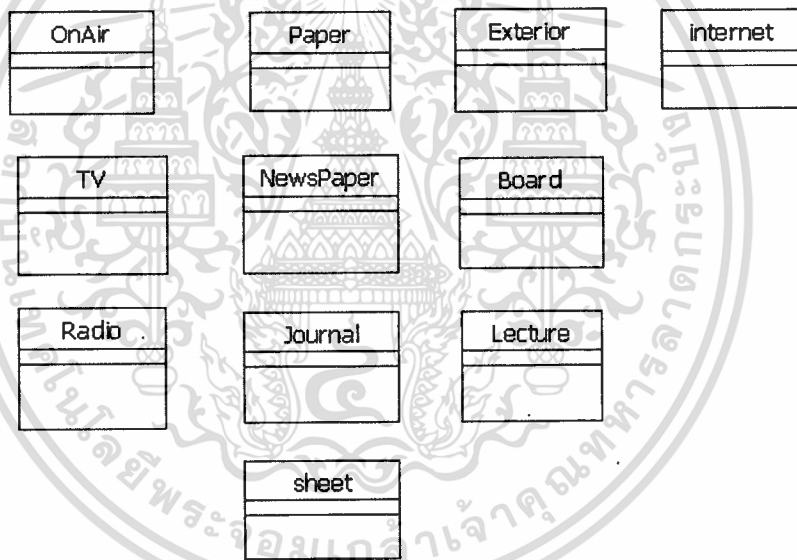
1. ในโครงการโฆษณาประชาสัมพันธ์ตั้งโครงการขึ้นมาจะมีการดำเนินการอย่างน้อย 1 งานหรือมากกว่านั้น ซึ่งรายการดำเนินการอาจเป็นงานโฆษณาหรือประชาสัมพันธ์อย่างใดอย่างหนึ่งหรือเป็นทั้งสองอย่าง หรือยังไม่มีที่ตั้งโครงการขึ้นมา
2. ถ้ามีงานการดำเนินงานนั้น 1 งานอาจจะมีหัวข้อที่ใช้สำหรับสื่อชิ้นนั้นขึ้น 1 หัวข้อหรือมากกว่านั้น
3. ในหัวข้องานโฆษณาประชาสัมพันธ์ 1 หัวข้ออาจจะมีการใช้สื่อหลาย ๆ ประเภทเช่น โทรทัศน์, วิทยุ, หนังสือพิมพ์, แผ่นพับ, การจัดบอร์ด, การบรรยายในสถานที่ต่าง ๆ, บทความลงวารสาร หรือ อินเทอร์เน็ตผ่านทาง web

ทำให้หลากหลายเพิ่มขึ้นในระบบ

- สื่อที่ออกอากาศ
- สื่อที่เป็นกระดาษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สื่อที่นำเสนอนอกสถานที่
- สื่อที่นำเสนอทางอินเทอร์เน็ต
- โทรทัศน์
- วิทยุ
- หนังสือพิมพ์
- แผ่นพับ
- การจัดบอร์ด
- การบรรยายในสถานที่ต่างๆ
- บทความลงวารสาร



รูปที่ 4.3 คลาสที่เพิ่มขึ้นมาจากคลาส media

จากนั้นให้ดูพฤติกรรมของแต่ละคลาส

- Media มีวัตถุประสงค์อะไรที่ใช้สื่อนี้, กลุ่มเป้าหมาย สามารถเพิ่มสื่อได้
- OnAir ออกอากาศวันที่เท่าไร, ออกไปหรือยัง, ใช้เวลาเท่าไร
- Paper ชื่อของกระดาษ, ฉบับที่, วันที่, จากหน้าที่ ถึงหน้าที่
- Exterior สถานที่อะไร, จำนวนคนเข้าชมหรือฟังบรรยาย
- Internet ใช้บริการอะไรของอินเทอร์เน็ต
- TV ออกอากาศครั้งที่เท่าไร, ช่องอะไร, เวลาเท่าไร

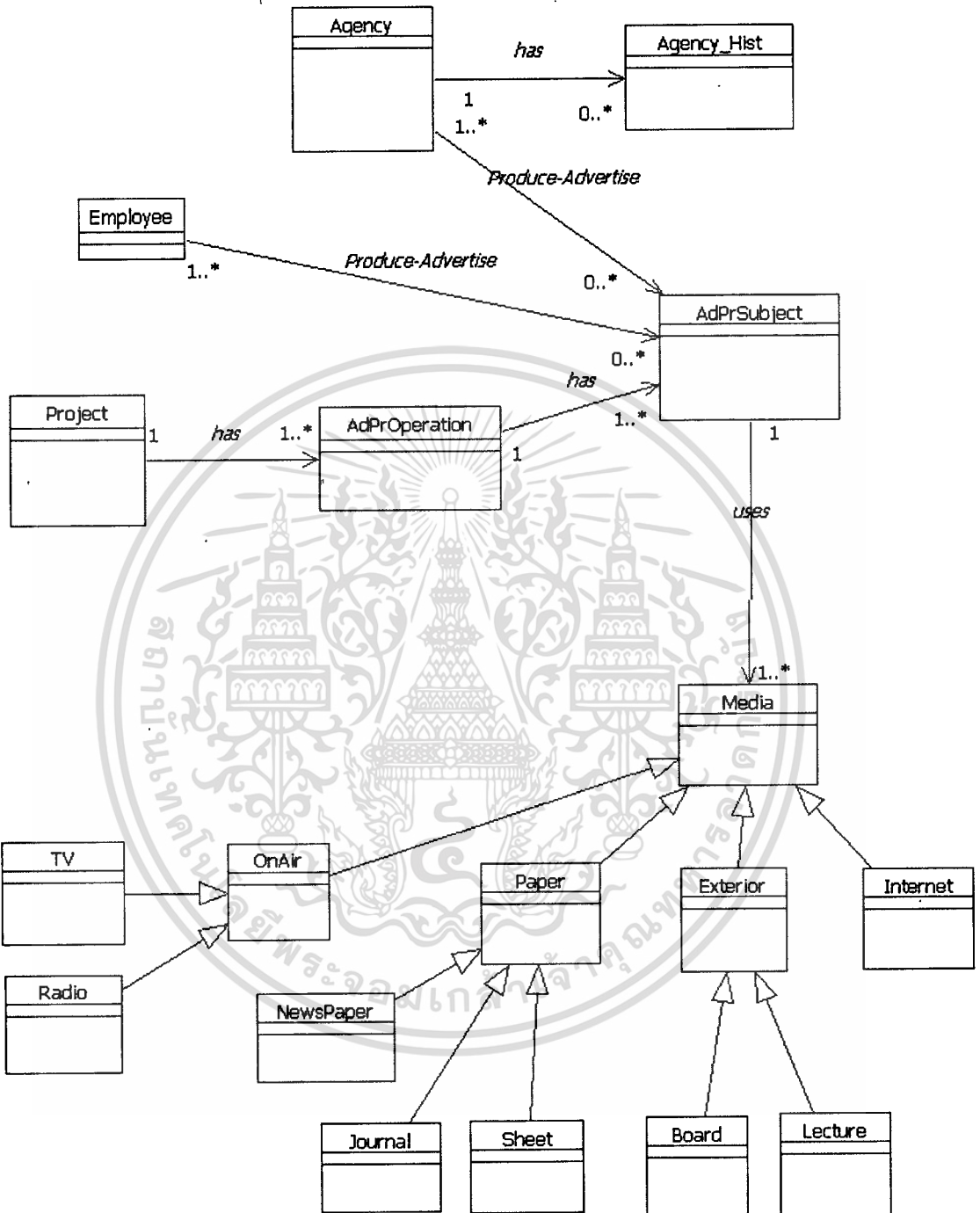
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Radio ออกอากาศครั้งที่เท่าไร, ช่องอะไร, เวลาเท่าไร
- NewsPaper หัวข้อที่ลงหนังสือพิมพ์, ภาพ, เนื้อหา
- Journal ที่บอทความ
- Sheet ขนาดเท่าไร, จำนวนที่พิมพ์
- Board ไม่มี attribute เพิ่ม
- Lecture ไม่มี attribute เพิ่ม

4. แล้วแต่หัวข้อมาจะมีการว่าจ้างบริษัทผู้ผลิตโฆษณาหรือทำการผลิตและวางสื่อเอง

สามารถทำให้ Class Diagram พื้นฐานสมบูรณ์ขึ้น ได้ด้วยการเพิ่มเติม Attributes และ Function ของแต่ละ Class ซึ่งจะทำได้ Class Diagram ดังต่อไปนี้





รูปที่ 4.4 คลาสไดอะแกรมระบบงานโฆษณาประชาสัมพันธ์ที่ได้รับการปรับปรุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาแต่ละคลาสว่าจะมี attribute และ function ดังนี้

คลาสโครงการ (PROJECT)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
PROJ_ID	รหัสโครงการ	CHAR	3
PROJ_NAME	ชื่อโครงการ	CHAR	200
PROJ_DATE_S	วันที่เริ่มต้นโครงการ	DATETIME	
PROJ_DATE_E	วันที่สิ้นสุดโครงการ	DATETIME	
PROJ_PR_BUDG_P	งบประมาณการผลิตสื่อสำหรับงาน ประชาสัมพันธ์	FLOAT	
PROJ_AD_BUDG_P	งบประมาณการผลิตสื่อสำหรับงาน โฆษณา	FLOAT	
PROJ_PR_BUDG_A	งบประมาณการวางสื่อสำหรับงานประชา สัมพันธ์	FLOAT	
PROJ_AD_BUDG_A	งบประมาณการวางสื่อสำหรับงาน โฆษณา	FLOAT	
APPLIANCE_CODE	รหัสอุปกรณ์ไฟฟ้าสำหรับการโฆษณา ประชาสัมพันธ์	CHAR	6
ฟังก์ชัน	จุดประสงค์		
ADDPROJ()	เพื่อเพิ่มโครงการใหม่		
EDITPROJ()	แก้ไขข้อมูลโครงการ		
DELETEPROJ()	ลบข้อมูลโครงการ		
REFRESHPROJ()	ทำการ refresh ข้อมูล		

คลาสการดำเนินงานโฆษณาประชาสัมพันธ์ (ADPROPERATION)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
ADPR_ID	รหัสงานโฆษณาประชาสัมพันธ์	CHAR	5
ADPR_TYPE	ประเภทงานโฆษณาหรือประชาสัมพันธ์	CHAR	1
ADPR_NAME	ชื่องานโฆษณาประชาสัมพันธ์	CHAR	200
ADPR_B_PROD	งบประมาณการผลิตสื่อ	FLOAT	

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADPR_B_ADV	งบประมาณการวางสื่อ	FLOAT	
ADPR_DATE_S	วันที่เริ่มงานโฆษณาประชาสัมพันธ์	DATETIME	
ADPR_DATE_E	วันที่สิ้นสุดงานโฆษณาประชาสัมพันธ์	DATETIME	
ฟังก์ชัน	จุดประสงค์		
ADDOPERATION()	เพิ่มงานโฆษณาประชาสัมพันธ์ใหม่		
EDITOPERATION()	แก้ไขข้อมูลงานโฆษณาประชาสัมพันธ์		
DELETEOPERATION()	ลบข้อมูลงานโฆษณาประชาสัมพันธ์		
REFRESHOPERATION()	ทำการ refresh ข้อมูล		

คลาสหัวข้องานโฆษณาประชาสัมพันธ์ (ADPRSUBJECT)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
ADPR_SUBJ_ID	รหัสหัวข้องานโฆษณาประชาสัมพันธ์	CHAR	7
ADPR_SUBJ_NAME	ชื่อหัวข้องานโฆษณาประชาสัมพันธ์	CHAR	200
ADPR_SUBJ_B_PROD	งบประมาณการผลิตสื่อหัวข้องานโฆษณาประชาสัมพันธ์	FLOAT	
ADPR_SUBJ_B_ADV	งบประมาณการวางสื่อหัวข้องานโฆษณาประชาสัมพันธ์	FLOAT	
ADPR_SUBJ_A_PROD	ค่าใช้จ่ายจริงในการผลิตสื่อหัวข้องานโฆษณาประชาสัมพันธ์	FLOAT	
ADPR_SUBJ_A_ADV	ค่าใช้จ่ายจริงในการวางสื่อหัวข้องานโฆษณาประชาสัมพันธ์	FLOAT	
ADPR_P_PROD_DATE_S	แผนการเริ่มต้นการผลิตสื่อวันที่	DATETIME	
ADPR_P_PROD_DATE_E	แผนการสิ้นสุดการผลิตสื่อวันที่	DATETIME	
ADPR_P_ADV_DATE_S	แผนการเริ่มต้นการวางสื่อวันที่	DATETIME	
ADPR_P_ADV_DATE_E	แผนการสิ้นสุดการวางสื่อวันที่	DATETIME	
ADPR_A_PROD_DATE_S	วันที่เริ่มต้นการผลิตสื่อจริง	DATETIME	
ADPR_A_PROD_DATE_E	วันที่สิ้นสุดการผลิตสื่อจริง	DATETIME	
ADPR_A_ADV_DATE_S	วันที่เริ่มต้นการวางสื่อจริง	DATETIME	
ADPR_A_ADV_DATE_E	วันที่สิ้นสุดการวางสื่อจริง	DATETIME	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน	จุดประสงค์
ADDSUBJECT()	เพิ่มหัวข้องาน โฆษณาประชาสัมพันธ์ใหม่
EDITSUBJECT()	แก้ไขข้อมูลหัวข้องาน โฆษณาประชาสัมพันธ์
DELETESUBJECT()	ลบข้อมูลหัวข้องาน โฆษณาประชาสัมพันธ์
REFRESHSUBJECT()	ทำการ refresh ข้อมูล

คลาสพนักงาน (EMPLOYEE)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
EMP_ID	รหัสพนักงาน	CHAR	6
TITLE	ตำแหน่ง	CHAR	5
NAME	ชื่อพนักงาน	CHAR	30
SEX	เพศ	FLOAT	1
DEPT_ID	รหัสสังกัด	FLOAT	7

คลาสบริษัทผู้ผลิตงาน โฆษณาประชาสัมพันธ์ (AGENCY)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
AGENCY_ID	รหัสผู้ผลิตงาน โฆษณาประชาสัมพันธ์	CHAR	6
AGENCY_NAME	ชื่อผู้ผลิตงาน โฆษณาประชาสัมพันธ์	CHAR	200
AGENCY_ADDR	ที่อยู่ผู้ผลิตงาน โฆษณาประชาสัมพันธ์	CHAR	200
ZIP_CODE	รหัสไปรษณีย์	CHAR	6
TELEPHONE	เบอร์โทรศัพท์	CHAR	30
FAX	เบอร์แฟกซ์	CHAR	30
ฟังก์ชัน	จุดประสงค์		
ADDAGENCY()	เพิ่มผู้ผลิตงาน โฆษณาประชาสัมพันธ์ใหม่		

คลาสผลงานที่ผ่านมาของบริษัทผู้ผลิตงาน โฆษณาประชาสัมพันธ์ (AGENCY_HIST)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
JOB_NAME	รหัสผู้ผลิตงาน โฆษณาประชาสัมพันธ์	CHAR	200
ORG_NAME	ชื่อผู้ผลิตงาน โฆษณาประชาสัมพันธ์	CHAR	200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRICE	ที่อยู่ผู้ผลิตงานโฆษณาประชาสัมพันธ์	FLOAT	
ฟังก์ชัน	จุดประสงค์		
ADDAGENCYHIST()	เพิ่มผลงานที่ผ่านมาของผู้ผลิตงานโฆษณาประชาสัมพันธ์ใหม่		

คลาสสื่อนำเสนอ (MEDIA)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
OBJECTIVE	วัตถุประสงค์	CHAR	200
GROUP	กลุ่มเป้าหมาย	CHAR	200
ฟังก์ชัน	จุดประสงค์		
ADDMEDIA()	เพิ่มสื่อนำเสนอใหม่		

คลาสออกอากาศ (ONAIR)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
ONAIR_DATE	วันที่ออกอากาศ	DATETIME	
ONAIR_YN	เพื่อตรวจสอบการออกอากาศ 1 = ออกอากาศ 2 = ยังไม่ได้ออกอากาศ 3 = เลื่อนการออกอากาศ	CHAR	1
AMOUNT_SEC	เวลาที่ออกอากาศเป็นวินาที	FLOAT	
AMOUNT_MN	เวลาที่ออกอากาศเป็นนาที	FLOAT	
AMOUNT_HOUR	เวลาที่ออกอากาศเป็นชั่วโมง	FLOAT	

คลาสโทรทัศน์ (TV)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
CHANNEL	สถานีโทรทัศน์ช่อง	CHAR	10
TVPROGRAM	ชื่อรายการโทรทัศน์	CHAR	200
TV_ONAIR_NO	ครั้งที่ออกอากาศ	FLOAT	
TIME	เวลาที่ออกอากาศ	CHAR	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสวิทยุ (RADIO)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
STATION	สถานีวิทยุคลื่น	CHAR	10
SPOTNAME	ชื่อรายการวิทยุ	CHAR	200
SPOT_ONAIR_NO	ครั้งที่ออกอากาศ	FLOAT	
TIME	เวลาที่ออกอากาศ	CHAR	5

คลาสสิ่งพิมพ์ (PAPER)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
VOLUME	ปีที่	FLOAT	
NO	ฉบับที่	FLOAT	
DAY	วันที่	CHAR	2
MONTH	เดือน	CHAR	2
YEAR	ปี	CHAR	4
PNAME	ชื่อสิ่งพิมพ์	CHAR	200
S_PAGE	เริ่มต้นที่หน้า	FLOAT	
E_PAGE	สิ้นสุดที่หน้า	FLOAT	

คลาสหนังสือพิมพ์ (NEWSPAPER)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
HEADLINE	หัวข้อหลัก	VARCHAR	200
SUBHEADLINE	หัวข้อย่อย	VARCHAR	200
VISUAL	ภาพ	IMAGE	
BODY	เนื้อหาข่าว	VARCHAR	5000
LOGO	สัญลักษณ์	IMAGE	

คลาสวารสาร (JOURNAL)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
ARTICLE	ชื่อบทความ	VARCHAR	200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสแผ่นพับ (SHEET)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
WIDTH	ความกว้าง (ซ.ม.)	FLOAT	
HEIGHT	ความยาว (ซ.ม.)	FLOAT	
QUANTITY	จำนวนที่พิมพ์	FLOAT	

คลาสนำสื่อสถานที่อื่น ๆ (EXTERIOR)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
PLACE	ชื่อสถานที่	VARCHAR	200
AMOUNT	ปริมาณคนเข้าชมหรือฟัง	FLOAT	

คลาสการจัดนิทรรศการ (BOARD)

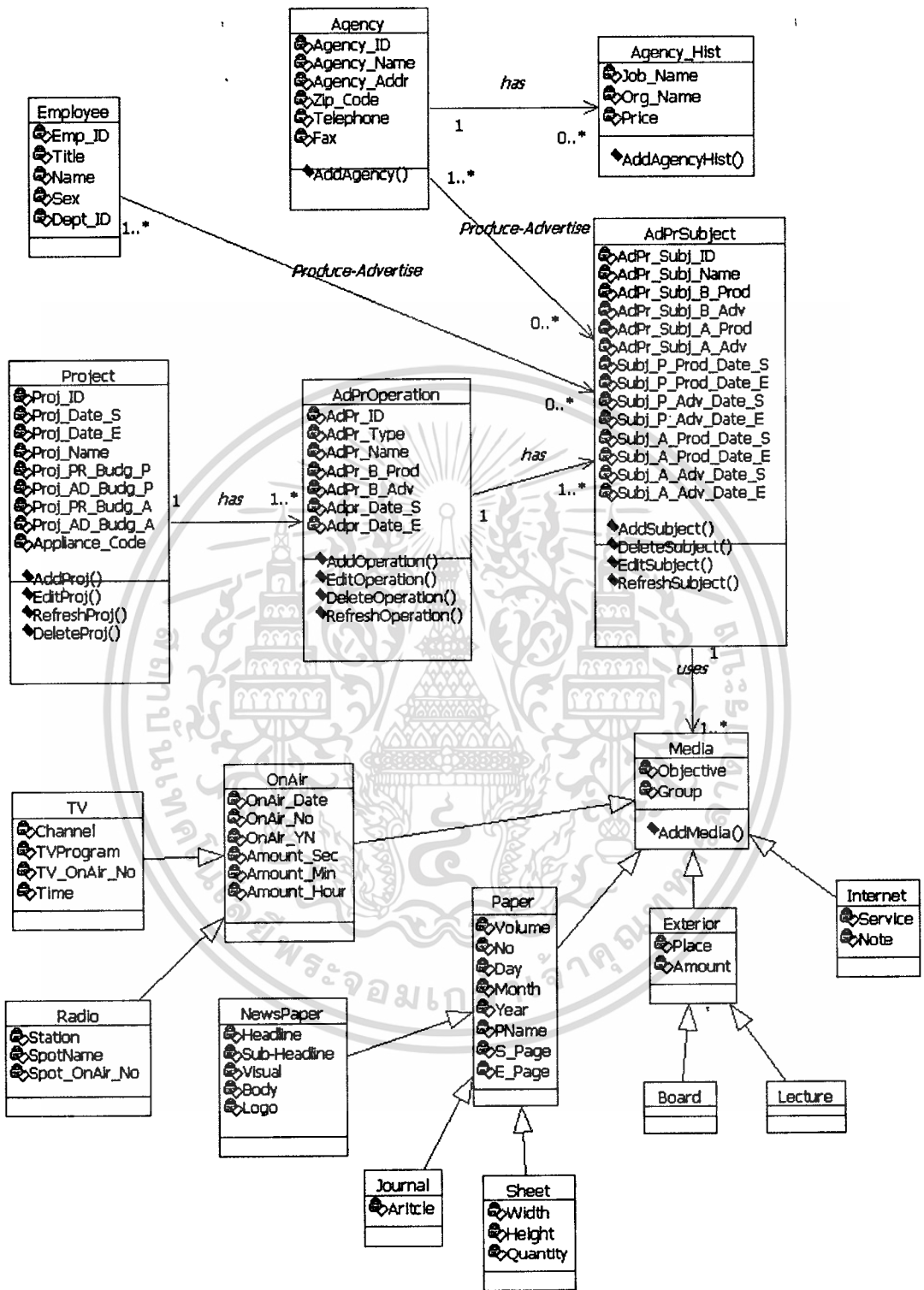
แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด

คลาสการจัดบรรยาย (LECTURE)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด

คลาสสื่ออินเทอร์เน็ต (INTERNET)

แอตทริบิวต์	ความหมาย	ชนิดข้อมูล	ขนาด
SERVICE	ใช้บริการต่าง ๆ บนอินเทอร์เน็ตเช่น Web Site หรือ อีเมล	CHAR	200
NOTE	บันทึก	CHAR	200

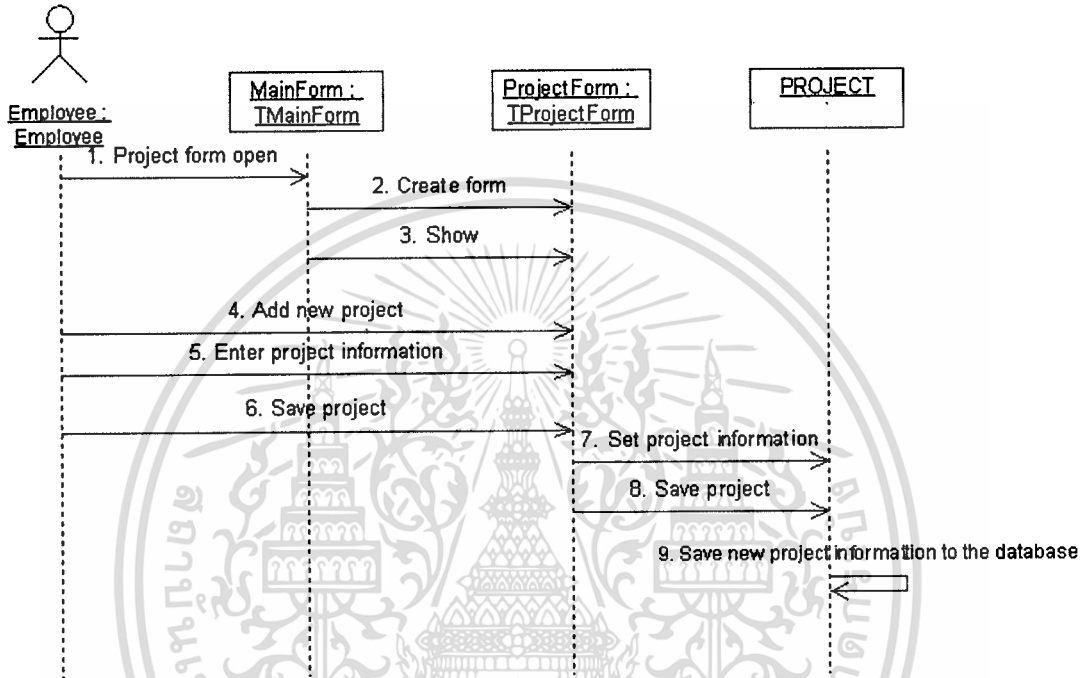


รูปที่ 4.5 คลาสไดอะแกรมระบบงานที่ได้รับการปรับปรุง มี attribute และ Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

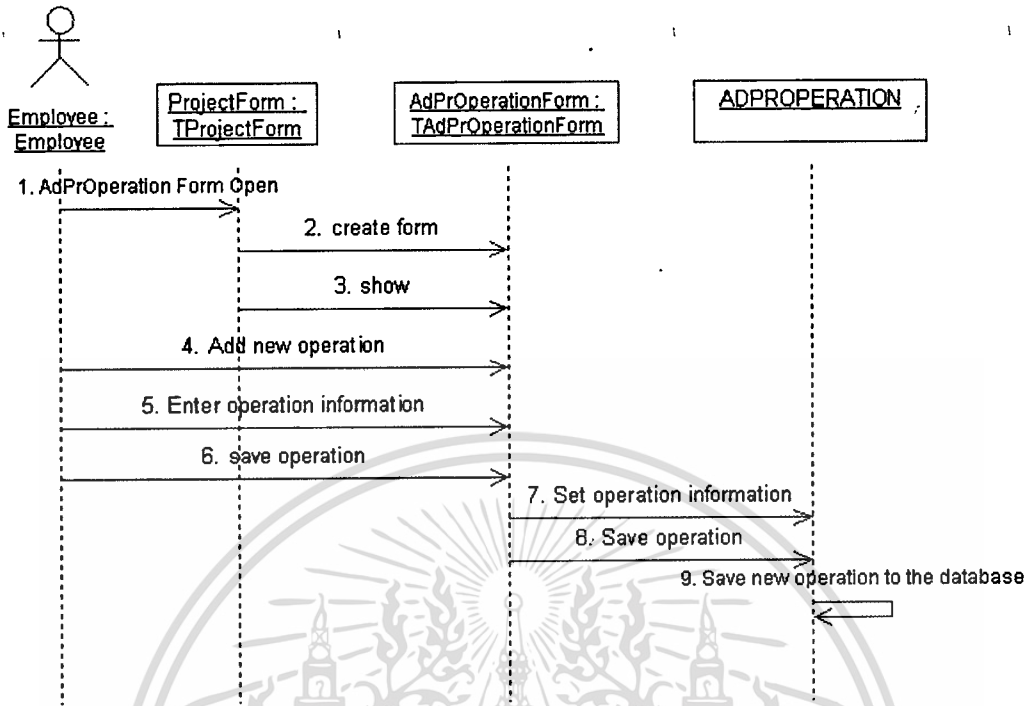
4.2.3.5 การทำงานของกิจกรรมต่าง ๆ

เพื่อบรรยายถึงกิจกรรมที่เกิดจากกิจกรรมของ Class ต่าง ๆ ในแต่ละ Use Case สามารถใช้ Sequence Diagram ดังนี้



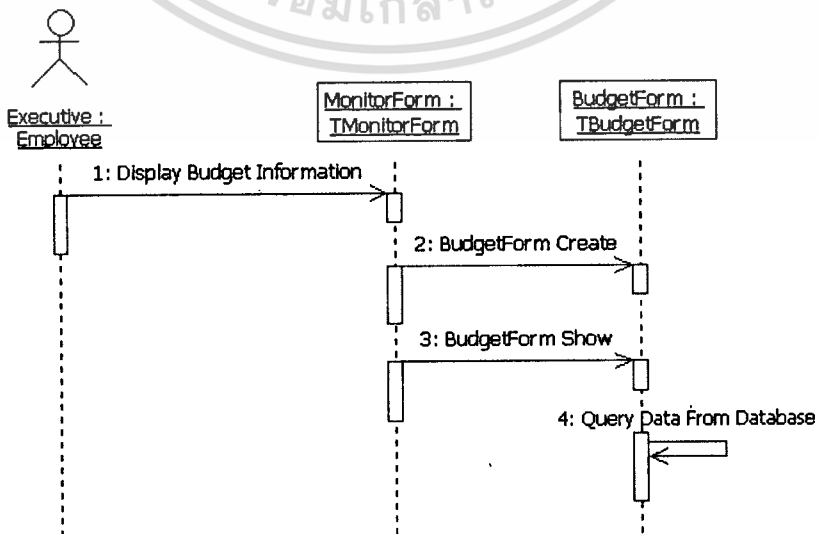
รูปที่ 4.6 เป็น Sequence Diagram ของ use case Project Planning การเพิ่ม โครงการงาน โฆษณา ประชาสัมพันธ์

ในระบบงานนี้ไม่มีการคำนวณที่ซับซ้อนลักษณะของ Sequence Diagram จะมีลักษณะใกล้เคียงกันคือ คลาสต่าง ๆ มีการบันทึก, แก้ไข, ลบ, แสดงผล ในรูปที่ 4.6 Actor : Employee ให้แอปเจกต์ MainForm เปิดหน้าต่างบันทึกข้อมูลโครงการซึ่ง คลาส TmainForm จะส่ง message ให้แอปเจกต์ ProjectForm ทำการสร้างหน้าต่างและแสดงผลขึ้นมา หลังจากนั้น Actor : Employee ก็จะสามารถเพิ่มข้อมูลโครงการใหม่ได้และส่งให้บันทึกข้อมูลลงไป แอปเจกต์ ProjectForm เมื่อถูกส่งให้บันทึกข้อมูลก็จะส่งข้อมูลไปบันทึกที่ฐานข้อมูล จากรูปที่ 4.7 Actor : Employee สามารถที่จะเข้าไปเพิ่มงานการดำเนินการได้อีก ซึ่งจะมี Form เพิ่มขึ้นมาคือ :AdPrOperationForm

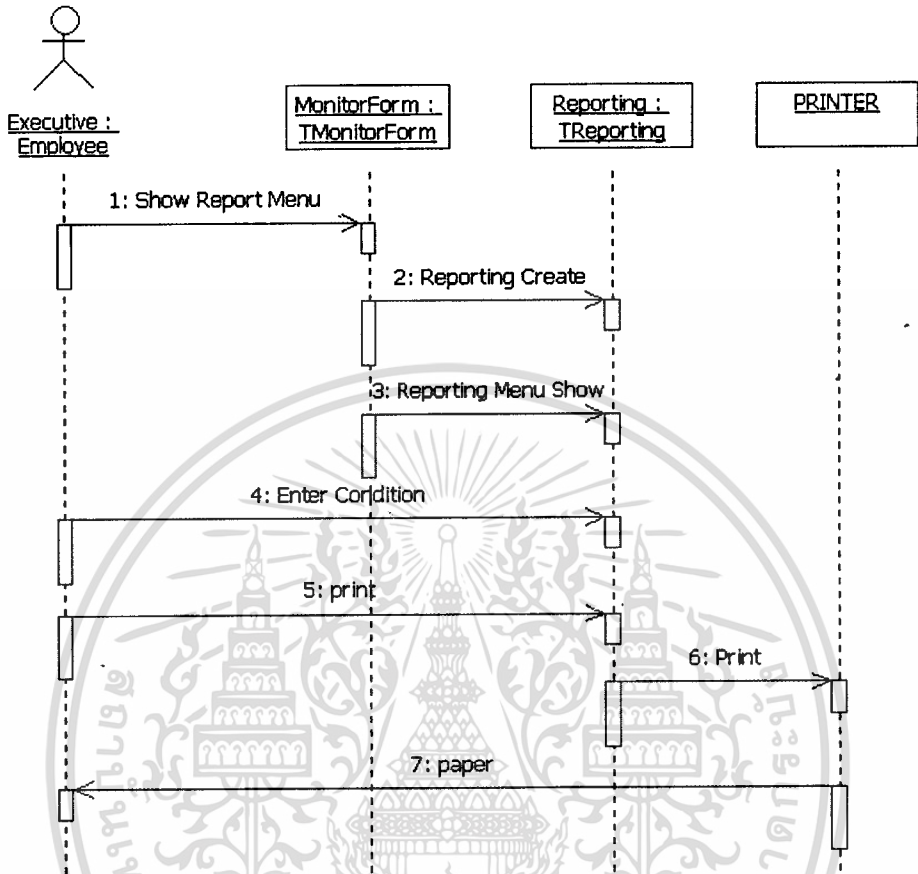


รูปที่ 4.7 เป็น Sequence Diagram ของ use case Project Planning การเพิ่มงานการดำเนินงาน

รูปที่ 4.8 เป็น Sequence Diagram จาก Use Case Monitoring ของ Actor : Executive ใช้ในการดูข้อมูลทั่วไป ซึ่ง Actor : Executive จะบอกให้ออบเจกต์ MonitorForm ทำการแสดงผลข้อมูลงบประมาณการใช้จ่ายในโครงการ จากนั้นออบเจกต์ MonitorForm ก็จะบอกให้ออบเจกต์ BudgetForm ทำการสร้างหน้าต่างและแสดงผลให้กับ Actor : Executive ในรูปที่ 4.9 ก็เหมือนกัน Actor : Executive ต้องการ ด้รายงานที่เป็นกระดาษจากออบเจกต์ Reporting



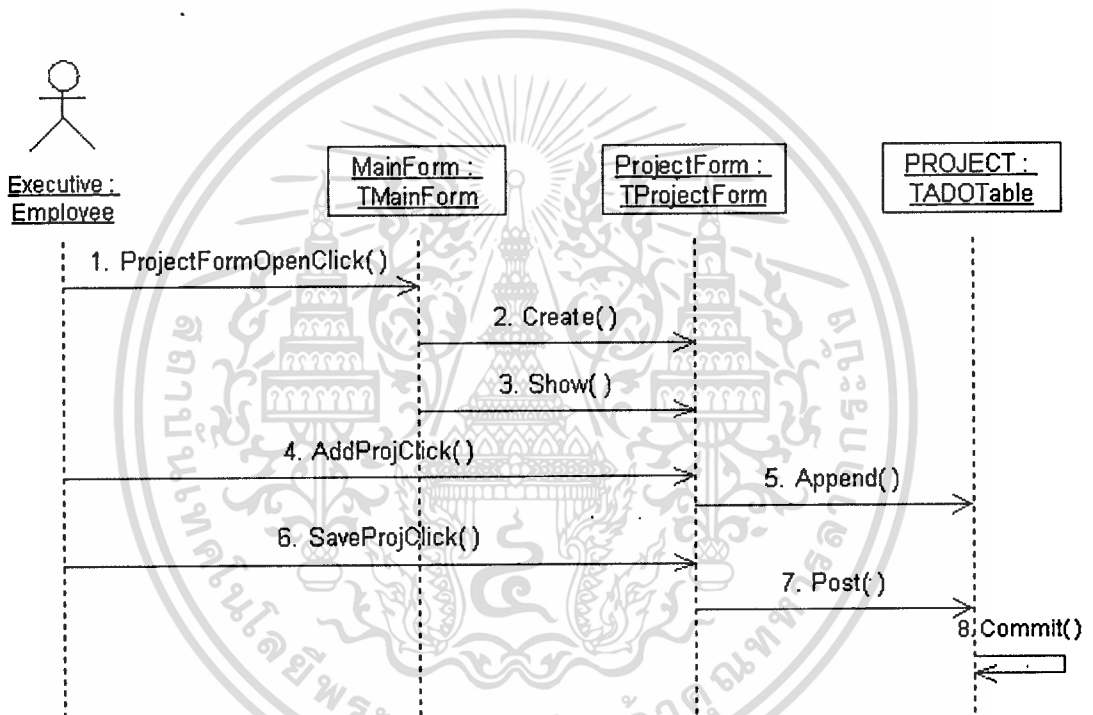
รูปที่ 4.8 เป็น Sequence Diagram ของ use case Monitoring เพื่อตรวจสอบการใช้จ่ายเงิน



รูปที่ 4.9 เป็น Sequence Diagram ของ use case Reporter ในการเรียกดูรายงานของผู้บริหาร

4.2.4 Object-Oriented Design (OOD)

โดยหลักการแล้วในขั้นตอนนี้จะนำเอา Class Diagram ที่ได้วิเคราะห์ไว้มาทำการตรวจสอบว่าจะมีข้อบกพร่องอะไรอีกบ้างโดยดูจาก Sequence Diagram ด้วยว่ายังขาดขั้นตอนอะไรอีกบ้างมาปรับปรุงให้ดีขึ้นและเขียน Class Diagram ขึ้นมาใหม่ให้สมบูรณ์เพื่อจะได้นำไปออกแบบฐานข้อมูลถ้ามีการเก็บข้อมูลที่ถาวร จากรูป 4.10 สามารถที่จะเปลี่ยนความหมายของคำพูดให้เป็นฟังก์ชันงานที่อยู่ในคลาสที่เรียกใช้ได้ซึ่งก็จะเป็นการตรวจสอบ method ในคลาสไปในตัวว่าครบหรือไม่



รูปที่ 4.10 Sequence Diagram ที่ปรับปรุงมาจากตอนวิเคราะห์ว่าควรมีฟังก์ชันอะไรบ้างในฟอร์ม

4.2.4.1 ออกแบบสถาปัตยกรรมแอปพลิเคชัน

การออกแบบโดยใช้ Component Diagram คือ UML diagram ที่แสดงว่าระบบมีคอมโพเนนต์อะไรบ้างและขึ้นอยู่กับคอมโพเนนต์อะไรบ้าง

ให้ดูจากไดอะแกรมว่าทั้ง Source Code และ runtime component ในระบบ การเพิ่มคอมโพเนนต์เข้าไปให้ดูชนิดของไอคอนที่แตกต่างใน C++, Java หรือ Visual Basic projects ส่วนใหญ่จะใช้ไอคอน Package Specification, Package Body และ Executable icon โดย Package Specification icon จะถูกใช้สำหรับ .H ไฟล์ ส่วน Package Specification หรือ Component ไอคอนใช้เป็น .JAVA ไฟล์, Visual Basic project หรือ DLL ไฟล์และ Package Body ไอคอนจะใช้เป็น .CPP ไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปเราจะสร้าง Component View package สำหรับแต่ละ Logical View package ตัวอย่างคือถ้าใน Logical view package มีการชื่อ Orders ภายในจะบรรจุคลาส Order, OrderItem และ OrderForm มาที่ Component View package จะบรรจุคอมโพเนนต์ที่เชื่อมโยงกับคลาส Order, OrderItem และ OrderForm

จาก logical View ที่ได้ออกแบบไว้นามาสร้าง Component Diagram การออกแบบโดยระบบจะแบ่งออกเป็น Presentation Logic Subsystem ซึ่งมี GUI คือ ADPRSystem.exe สำหรับพนักงานอินพุตข้อมูลและ ADPRReport.exe สำหรับผู้บริหารดูรายงานทางหน้าจอคอมพิวเตอร์ หรือพิมพ์รายงานออกมา

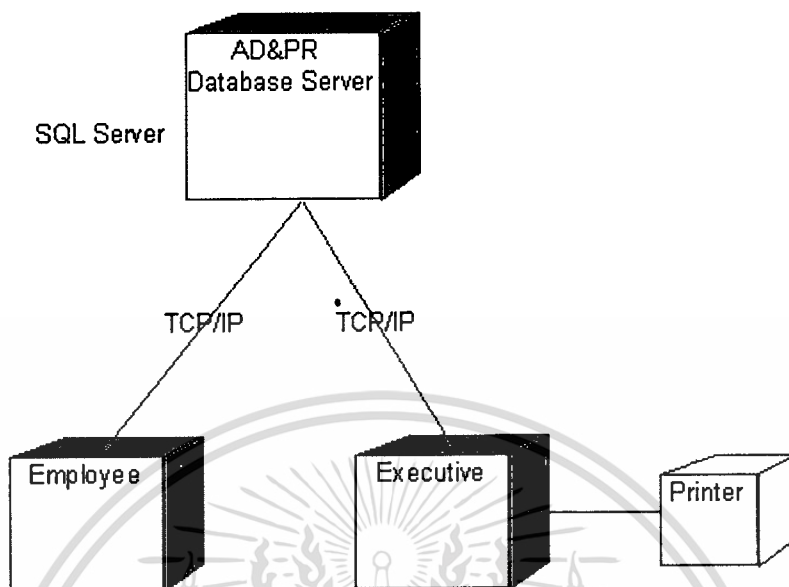


รูปที่ 4.11 Component Diagram ในระบบ

ใน Database Logic Subsystem มีฐานข้อมูลเพียงตัวเดียวซึ่งจะประกอบด้วยเทเบิลต่าง ๆ หลายเทเบิล ซึ่งจะต้องอาศัยการออกแบบฐานข้อมูลในช่วง Persistent Data Design ก่อน

4.2.4.2 ออกแบบสถาปัตยกรรมระบบ

ทางเลือกของระบบมีหลายทาง ที่ใช้ระบบ Client/Server เพราะมีผู้ใช้งานในระบบน้อยคือ มีพนักงานที่อินพุตข้อมูลและผู้บริหารจะดูข้อมูลสรุปในการดำเนินงาน



รูปที่ 4.12 Deployment Diagram แสดงสถาปัตยกรรมของระบบงานโฆษณาประชาสัมพันธ์

ความต้องการของระบบด้าน Hardware

- Server
 - a. CPU 500 MHz
 - b. RAM 128 MB
 - c. Lan Card 100-BaseT
 - d. Hard Disk 20GB
 - e. Hub
 - f. UPS (อุปกรณ์สำรองไฟ)
- PC (Client)
 - g. CPU 500 MHz
 - h. RAM 64 MB
 - i. Lan Card 100-BaseT

ความต้องการของระบบด้าน Software

- Delphi 5.0 ขึ้นไป
- Windows 98, Window NT

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 การออกแบบฐานข้อมูล

เป็นขั้นตอนที่ออกแบบการเก็บข้อมูลไว้ถาวร ในการดำเนินงานนี้ได้ใช้ Relational Database เป็นส่วนฐานข้อมูลที่ใช้เก็บ ทำให้ต้องมีการแปลงจาก Class Diagram ของระบบที่สร้างไว้ขั้นต้น สามารถออกแบบเป็น Relational Database โดยมีหลักดังนี้

- กำหนดให้ Attribute ตัวใดตัวหนึ่งหรือกลุ่มใดกลุ่มหนึ่งเป็น Primary Key
- สร้าง Table ที่มีทุก ๆ Attribute ของคลาสและมี Primary Key ตามที่กำหนด ซึ่ง Primary Key จะต้องเป็น not null เสมอ
- สำหรับ attribute อื่น ๆ ที่ไม่ใช่ primary key ก็พิจารณาว่าตัวไหนเป็น null หรือ not null

1. จากความสัมพันธ์เชิง Association ระหว่างโครงการ (Project) กับการดำเนินงาน (AdPrOperation)

PROJECT						
	Column Name	Datatype	Length	Precision	Scale	Allow Nulls
PK	PROJ_ID	char	3	0	0	<input type="checkbox"/>
	PROJ_DATE_S	datetime	8	0	0	<input checked="" type="checkbox"/>
	PROJ_DATE_E	datetime	8	0	0	<input checked="" type="checkbox"/>
	PROJ_NAME	char	200	0	0	<input checked="" type="checkbox"/>
	PROJ_PR_BUDG_P	float	8	53	0	<input checked="" type="checkbox"/>
	PROJ_AD_BUDG_P	float	8	53	0	<input checked="" type="checkbox"/>
	PROJ_PR_BUDG_A	float	8	53	0	<input checked="" type="checkbox"/>
	PROJ_AD_BUDG_A	float	8	53	0	<input checked="" type="checkbox"/>
	DEPT_ID	char	7	0	0	<input checked="" type="checkbox"/>
	APPLIANCE_CODE	char	6	0	0	<input checked="" type="checkbox"/>

ADPROperation						
	Column Name	Datatype	Length	Precision	Scale	Allow Nulls
PK	ADPR_ID	char	5	0	0	<input type="checkbox"/>
	ADPR_TYPE	char	1	0	0	<input checked="" type="checkbox"/>
	ADPR_NAME	char	200	0	0	<input checked="" type="checkbox"/>
	ADPR_B_PROD	float	8	53	0	<input checked="" type="checkbox"/>
	ADPR_B_ADV	float	8	53	0	<input checked="" type="checkbox"/>
	ADPR_DATE_S	datetime	8	0	0	<input checked="" type="checkbox"/>
	ADPR_DATE_E	datetime	8	0	0	<input checked="" type="checkbox"/>
	PROJ_ID	char	3	0	0	<input type="checkbox"/>

รูปที่ 4.13 แสดงความสัมพันธ์ของเทเบิล PROJECT กับ ADPROperation

ความสัมพันธ์ระหว่างคลาส PROJECT กับ ADPROperation แบบ Association (one-to-many)

สามารถเก็บเป็น Relational Database โดยการสร้างให้หน้า Primary Key ด้าน one ไปเป็น Foreign Key ด้าน many ได้
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Key ของ Table ในด้าน many ซึ่งด้าน many มีค่า minimal cardinality เป็น 1 เพราะฉะนั้น Foreign Key จึงมีค่าเป็น null ด้วย

2. จากความสัมพันธ์เชิง Association ระหว่างหัวข้องานโฆษณาประชาสัมพันธ์ (AdPrSubject)

กับการดำเนินงาน (AdPrOperation) มีลักษณะ one-to-many เหมือนความสัมพันธ์ระหว่างคลาสชื่อ

1

ADPROperation						
	Column Name	Datatype	Length	Precision	Scale	Allow Nulls
PK	ADPR_ID	char	5	0	0	<input type="checkbox"/>
	ADPR_TYPE	char	1	0	0	<input checked="" type="checkbox"/>
	ADPR_NAME	char	200	0	0	<input checked="" type="checkbox"/>
	ADPR_B_PROD	float	8	53	0	<input checked="" type="checkbox"/>
	ADPR_B_ADV	float	8	53	0	<input checked="" type="checkbox"/>
	ADPR_DATE_S	datetime	8	0	0	<input checked="" type="checkbox"/>
	ADPR_DATE_E	datetime	8	0	0	<input checked="" type="checkbox"/>
	PROJ_ID	char	3	0	0	<input type="checkbox"/>

ADPRSUBJECT						
	Column Name	Datatype	Length	Precision	Scale	Allow Nulls
PK	ADPR_SUBJ_ID	char	7	0	0	<input type="checkbox"/>
	ADPR_SUBJ_NAME	char	200	0	0	<input checked="" type="checkbox"/>
	ADPR_SUBJ_B_PROD	float	8	53	0	<input checked="" type="checkbox"/>
	ADPR_SUBJ_B_ADV	float	8	53	0	<input checked="" type="checkbox"/>
	ADPR_SUBJ_A_PROD	float	8	53	0	<input checked="" type="checkbox"/>
	ADPR_SUBJ_A_ADV	float	8	53	0	<input checked="" type="checkbox"/>
	SUBJ_P_PROD_DATE_S	datetime	8	0	0	<input checked="" type="checkbox"/>
	SUBJ_P_PROD_DATE_E	datetime	8	0	0	<input checked="" type="checkbox"/>
	SUBJ_P_ADV_DATE_S	datetime	8	0	0	<input checked="" type="checkbox"/>
	SUBJ_P_ADV_DATE_E	datetime	8	0	0	<input checked="" type="checkbox"/>
	SUBJ_A_PROD_DATE_S	datetime	8	0	0	<input checked="" type="checkbox"/>
	SUBJ_A_PROD_DATE_E	datetime	8	0	0	<input checked="" type="checkbox"/>
	SUBJ_A_ADV_DATE_S	datetime	8	0	0	<input checked="" type="checkbox"/>
	SUBJ_A_ADV_DATE_E	datetime	8	0	0	<input checked="" type="checkbox"/>
	OBJECTIVE	char	200	0	0	<input checked="" type="checkbox"/>
	GROUPGOAL	char	200	0	0	<input checked="" type="checkbox"/>
	ADPR_ID	char	5	0	0	<input type="checkbox"/>

รูปที่ 4.14 แสดงความสัมพันธ์ของเทเบิล ADPROperation กับ ADPRSubject

3. จากความสัมพันธ์เชิง Association ระหว่างบริษัทผู้ผลิตงานโฆษณา (Agency) กับผลงานที่ผ่านมาของบริษัทผู้ผลิตงานโฆษณา (AgencyHist)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AGENCY						
Column Name	Datatype	Length	Precision	Scale	Allow Null	
AGENCY_ID	char	6	0	0		<input type="checkbox"/>
AGENCY_NAME	char	200	0	0		<input checked="" type="checkbox"/>
AGENCY_ADDR	char	200	0	0		<input checked="" type="checkbox"/>
ZIP_CODE	char	6	0	0		<input checked="" type="checkbox"/>
TELEPHONE	char	30	0	0		<input checked="" type="checkbox"/>
FAX	char	30	0	0		<input checked="" type="checkbox"/>

AGENCY_HIST						
Column Name	Datatype	Length	Precision	Scale	Allow Null	
JOB_NAME	char	200	0	0		<input type="checkbox"/>
ORG_NAME	char	200	0	0		<input checked="" type="checkbox"/>
PRICE	float	8	53	0		<input checked="" type="checkbox"/>
AGENCY_ID	char	6	0	0		<input checked="" type="checkbox"/>

รูปที่ 4.15 แสดงความสัมพันธ์ของเทเบิล AGENCY กับ AGENCY_HIST

- จากความสัมพันธ์เชิง Association ระหว่างสื่อโฆษณาประชาสัมพันธ์ (Media) กับหัวข้องานโฆษณาประชาสัมพันธ์(AdprSubject)

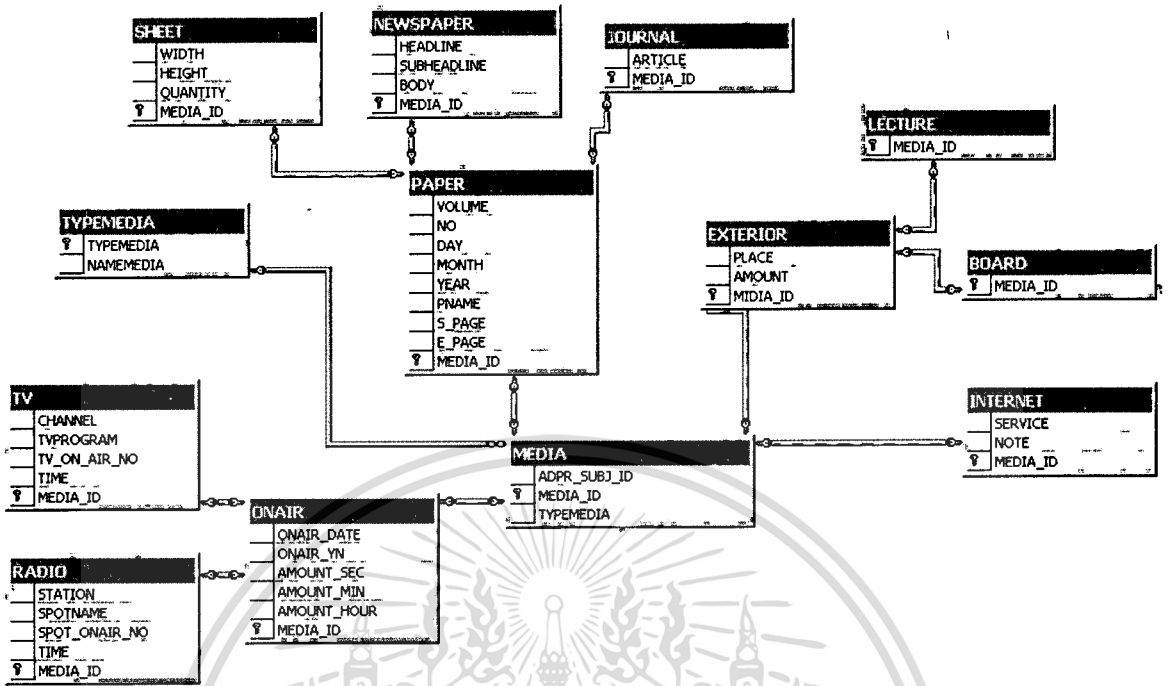
MEDIA						
Column Name	Datatype	Length	Precision	Scale	Allow Nulls	
ADPR_SUBJ_ID	char	7	0	0		
MEDIA_ID	char	10	0	0		
TYPEMEDIA	char	2	0	0		

ADPRSUBJECT						
Column Name	Datatype	Length	Precision	Scale	Allow Nulls	
ADPR_SUBJ_ID	char	7	0	0		
ADPR_SUBJ_NAME	char	200	0	0		
ADPR_SUBJ_B_PROD	float	8	53	0		
ADPR_SUBJ_B_ADV	float	8	53	0		
ADPR_SUBJ_A_PROD	float	8	53	0		
ADPR_SUBJ_A_ADV	float	8	53	0		
SUBJ_P_PROD_DATE_S	datetime	8	0	0		
SUBJ_P_PROD_DATE_E	datetime	8	0	0		
SUBJ_P_ADV_DATE_S	datetime	8	0	0		
SUBJ_P_ADV_DATE_E	datetime	8	0	0		
SUBJ_A_PROD_DATE_S	datetime	8	0	0		
SUBJ_A_PROD_DATE_E	datetime	8	0	0		
SUBJ_A_ADV_DATE_S	datetime	8	0	0		
SUBJ_A_ADV_DATE_E	datetime	8	0	0		
OBJECTIVE	char	200	0	0		
GROUPGOAL	char	200	0	0		
ADPR_ID	char	5	0	0		

รูปที่ 4.16 แสดงความสัมพันธ์ของเทเบิล ADPRSubject กับ MEDIA

- จากความสัมพันธ์เชิง Generalization ระหว่างสื่อโฆษณาประชาสัมพันธ์ (Media) กับคลาสต่าง ๆ ที่เป็นสืบทอดมาจากคลาส Media

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 แสดงความสัมพันธ์ของเทเบิล MEDIA กับเทเบิลต่างๆ ที่ได้สืบทอดคลาสมา

หลักการทั่วไปในการแปลง Generalization Class ให้เป็นเทเบิล มี 2 กรณี

1. กรณี Generalization แบบ Total-Overlapping , Partial-Overlapping ให้สร้างเทเบิลของ Superclass และของทุก Subclass ใช้หลักการ one-to-one association โดยสร้าง Foreign Key ไว้ที่เทเบิลของ subclass และ foreignkey นั้นต้องกำหนดให้เป็น not null เป็น Primary Key ในเทเบิลของ subclass ด้วย
2. กรณี Generalization ที่เป็นแบบ Total-Exclusive, Partial Exclusive ให้สร้างเทเบิลของ Superclass และของทุก ๆ Subclass (ใช้หลักการ one-to-one association) หลังจากนั้นให้สร้าง associate table ซึ่งมีเพียง field เดียวคือ Primary Key และทำดังนี้
 - a. กรณี Total-Exclusive ให้ Foreign Key ของ Table ของ Superclass เป็นตัวเดียวกันกับ Primary Key อ้างอิงมาที่ Primary Key ของ Associate Table ที่สร้างขึ้น แต่ถ้าเป็น Partial-Exclusive ให้เพิ่ม Foreign Key ซึ่งจะมีค่าเดียวกันกับ Primary Key เสมอหรือมีค่าเป็น null ก็ได้ ให้อ้างอิงไปยัง Associate Table
 - b. สร้าง Foreign Key เพิ่มเข้ายัง Table ของ Superclass อ้างอิงมายัง Associate Table โดยมีเงื่อนไขว่า Foreign Key ที่ Associate Table นั้นจะสามารถมีค่าเป็น Null ได้ ในกรณีที่ เป็น Partial-Exclusive และถูกตั้งเป็น Not Null ในกรณีเป็น Total-Exclusive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Total-Overlapping การเลือกอย่างใดอย่างหนึ่งหรือทั้งหมดเสมอ ไม่มีการยกเว้น

Total-Exclusive การเลือกอย่างใดอย่างหนึ่งเท่านั้น ไม่มีการยกเว้น

Partial-Exclusive การเลือกอย่างใดอย่างหนึ่ง ไม่สามารถเลือกทั้งหมดได้แต่อาจจะไม่เลือกทั้งหมดได้

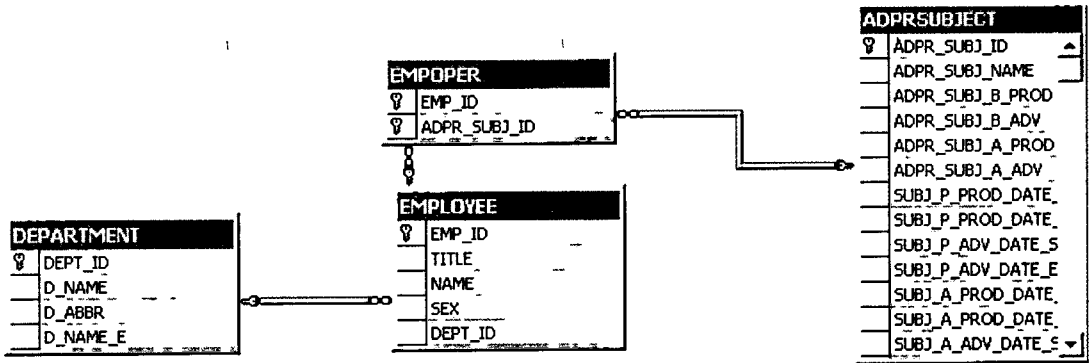
Partial-Overlapping การเลือกบางอย่างได้ ตัวอย่าง “นักเรียนบางคนเป็นชาย นักเรียนชายบางคนเป็นหัวหน้าห้อง แต่หัวหน้าห้องบางคนอาจไม่ใช่ นักเรียนชาย และก็มีนักเรียนอีกหลาย ๆ คนที่ไม่ใช่ นักเรียนชายและไม่ได้เป็นหัวหน้าห้อง” คุณสมบัติแบบนี้คือ Partial มาจากคำว่า บางคน และ Overlapping มาจากคำว่า หัวหน้าห้องบางคนอาจไม่ใช่ผู้ชาย

อย่างในความสัมพันธ์ของ Media กับคลาสต่าง ๆ ที่สืบทอดมาเป็นแบบ Total-Exclusive คือจะเลือกสื่อใดสื่อหนึ่งได้เท่านั้น ไม่สามารถเลือกทั้งสองได้

6. ความสัมพันธ์ที่เป็น many-to-many มีคลาส ADPRSUBJECT คือคลาสหัวข้องานโฆษณา ประชาสัมพันธ์ที่สัมพันธ์กับคลาส AGENCY และคลาส EMPLOYEE ทั้งผู้ผลิตงานโฆษณาสามารถที่จะทำหัวข้องานโฆษณาประชาสัมพันธ์ได้หลายหัวข้อหรือไม่ทำเลย เหมือนกันกับพนักงานอาจจะมีพนักงานหลายคนที่ทำหรือไม่ทำเลย



รูปที่ 4.18 แสดงความสัมพันธ์ของเทเบิล ADPRSubject กับ AGENCY และเทเบิลเพิ่ม AGENCYOPER



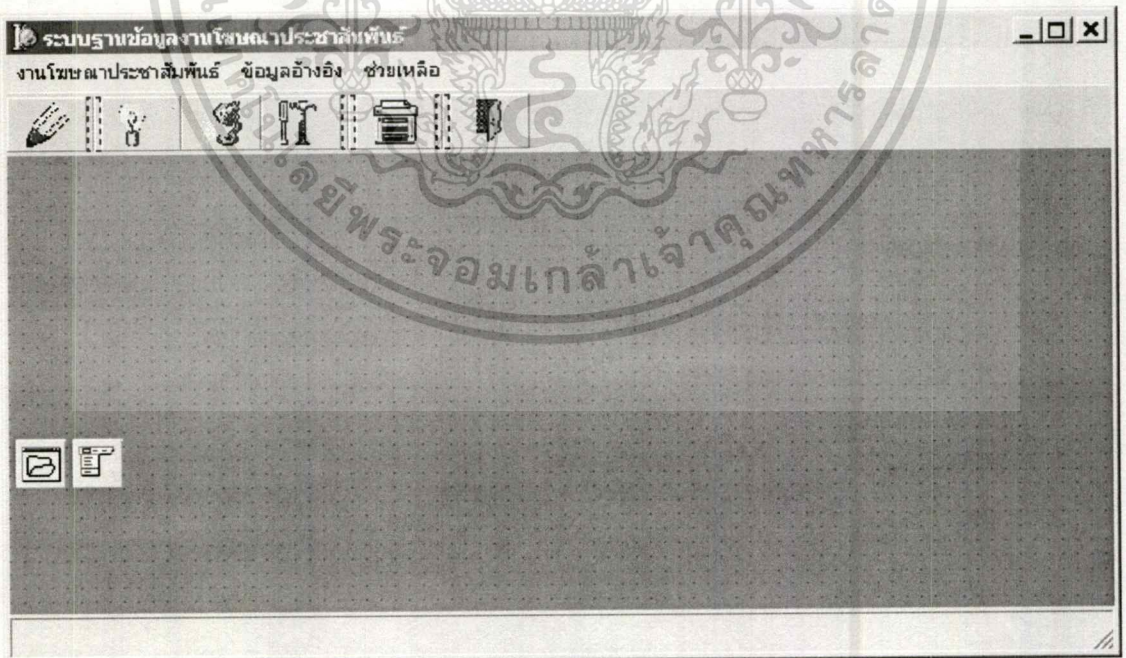
รูปที่ 4.19 แสดงความสัมพันธ์ของเทเบิล ADPRSubject กับ EMPLOYEE และเทเบิลเพิ่ม EMPOPER



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6 Screen Layout User Interface

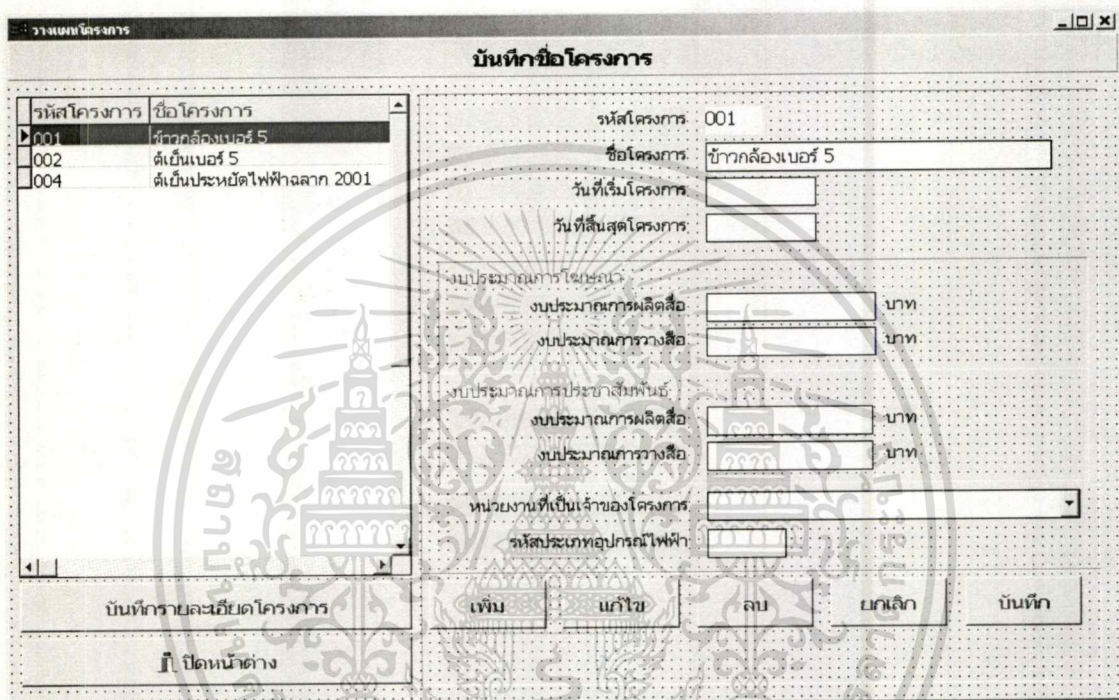
ชื่อฟอร์ม	MainForm
วัตถุประสงค์	เป็น Main Menu ในการเรียกฟังก์ชันอื่น ๆ
กิจกรรม	<p>คลิกที่ปุ่ม toolbar หรือว่าที่ menu ซึ่งจะมีฟังก์ชันการทำงานอยู่ภายในงานโฆษณาประชาสัมพันธ์</p> <ul style="list-style-type: none"> ● วางแผนโครงการ ● การดำเนินการ โดยบริษัทผู้ผลิต โฆษณา ● การดำเนินการ โดยพนักงาน ● การติดตาม ● การประเมินผลงาน <p>ข้อมูลอ้างอิง</p> <ul style="list-style-type: none"> ● บันทึกรายชื่อบริษัท โฆษณา ● บันทึกชนิดของสื่อโฆษณาประชาสัมพันธ์



รูปที่ 4.20 แสดง Main Menu ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟอร์ม	ProjectForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับเก็บข้อมูลโครงการ
กิจกรรม	มีปุ่มสำหรับเพิ่ม แก้ไข ลบ ยกเลิก บันทึกโครงการใหม่เข้าไปในระบบ และปุ่มสำหรับเข้าไปในรายละเอียดโครงการ



รูปที่ 4.21 แสดงหน้าต่างการบันทึกชื่อโครงการ

ชื่อฟอร์ม	AdPrOperationForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับเก็บข้อมูลรายละเอียดโครงการว่ามีงานโฆษณาประชาสัมพันธ์อะไรบ้าง
กิจกรรม	สามารถที่จะเพิ่ม แก้ไข ลบ ยกเลิก บันทึก ได้และปุ่มสำหรับเข้าไปดูรายละเอียดแต่ละงานโฆษณาประชาสัมพันธ์ซึ่งเป็นรายละเอียดของข้อมูลโครงการ

รูปที่ 4.22 แสดงหน้าต่างการบันทึกข้อมูลงานโฆษณาประชาสัมพันธ์

ชื่อฟอร์ม	AdPrSubjectForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับเก็บข้อมูลหัวข้อย่อของงานโฆษณาประชาสัมพันธ์
กิจกรรม	สามารถเพิ่ม แก้ไข ลบ ยกเลิก บันทึกได้ และปุ่มรายละเอียดเพื่อนำเสนอซึ่งเป็นปุ่มที่จะเลือกหัวข้อย่อนี้เป็นสื่อประเภทไหนเป็นรายละเอียดของงานโฆษณาประชาสัมพันธ์

วังเขมเนิกการ

บันทึกรายละเอียดงานโฆษณาประชาสัมพันธ์

รหัสงานโฆษณาประชาสัมพันธ์
 รหัสหัวข้องานโฆษณาประชาสัมพันธ์
 ชื่อหัวข้องานโฆษณาประชาสัมพันธ์

งบประมาณ

การผลิตสื่อ: บาท การวางสื่อ: บาท
 ใช้จริง: บาท ใช้จริง: บาท

แผนการวันที่: เริ่มจริงวันที่: แผนการวันที่: เริ่มจริงวันที่:

วันที่เริ่มต้น: วันที่เริ่มต้น:
 วันที่สิ้นสุด: วันที่สิ้นสุด:

วัตถุประสงค์:

กลุ่มเป้าหมาย:

เพิ่ม แก้ไข ลบ ยกเลิก บันทึก

รายละเอียดเพื่อนำเสนอ ไปดูหน้าต่าง

ชื่อหัวข้องานโฆษณาประชาสัมพันธ์

รูปที่ 4.23 แสดงหน้าต่างการบันทึกข้อมูลหัวข้องานโฆษณาประชาสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟอร์ม	MediaForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับเก็บข้อมูลเพื่อนำเสนอจากหัวข้องานโฆษณาประชาสัมพันธ์
กิจกรรม	ในหน้าต่างนี้จะสามารถเพิ่ม แก้ไข ลบ ยกเลิก ดูข้อมูล การเพิ่มเช่นมีการออกอากาศทางโทรทัศน์หลายครั้ง



รูปที่ 4.24 แสดงหน้าต่างการบันทึกข้อมูลเพื่อนำเสนอ

ชื่อฟอร์ม	TVForm, RadioForm, NewspaperForm, JournalForm, SheetForm, BoardForm, LectureForm, InternetForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับเก็บข้อมูลสื่อต่าง ๆ
กิจกรรม	เมื่อมีการแสดงหน้าต่างสำหรับสื่อแล้วสามารถบันทึกข้อมูลได้อย่างเดียว

สื่อภาพ

โทรทัศน์

วันที่ออกอากาศ ออกอากาศ ออกแล้ว ยังไม่ออก

จำนวนเวลา ชั่วโมง นาที วินาที

สถานีโทรทัศน์ช่อง

รายการโทรทัศน์ที่ออกอากาศ

ครั้งที่ เวลา

บันทึก

รูปที่ 4.25 แสดงหน้าต่างการบันทึกข้อมูลสื่อโทรทัศน์

สื่อภาพ

วิทยุ

วันที่ออกอากาศ ออกอากาศ ออกแล้ว ยังไม่ออก

จำนวนเวลา ชั่วโมง นาที วินาที

สถานี

ชื่อรายการวิทยุ

ครั้งที่ เวลา

บันทึก

รูปที่ 4.26 แสดงหน้าต่างการบันทึกข้อมูลสื่อวิทยุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อฯเสนอ

หนังสือพิมพ์

ปีที่

ฉบับที่ วันที่

ชื่อหนังสือพิมพ์

เริ่มที่หน้า สิ้นสุดที่หน้า

หัวข้อ

หัวข้อย่อย

เนื้อหาย่อ

บันทึก ปิดหน้าต่าง

รูปที่ 4.27 แสดงหน้าต่างการบันทึกข้อมูลสื่อหนังสือพิมพ์

สื่อฯเสนอ

วารสาร

ปีที่

ฉบับที่ วันที่

ชื่อวารสาร

เริ่มที่หน้า สิ้นสุดที่หน้า

ชื่อบทความ

บันทึก ปิดหน้าต่าง

รูปที่ 4.28 แสดงหน้าต่างการบันทึกข้อมูลบทความลงวารสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อการสอน

แผ่นพับ

ปีที่

ฉบับที่ วันที่พิมพ์

ชื่อแผ่นพับ

เริ่มที่หน้า สิ้นสุดที่หน้า

ความกว้าง ซม.

ความยาว ซม.

ปริมาณ ชุด

บันทึก ปิดหน้าต่าง

รูปที่ 4.29 แสดงหน้าต่างการบันทึกข้อมูลแผ่นพับ

สื่อการสอน

จัดนิทรรศการ

สถานที่

จำนวนคนฟังหรือชม คน

บันทึก ปิดหน้าต่าง

รูปที่ 4.30 แสดงหน้าต่างการบันทึกข้อมูลการจัดนิทรรศการ

สื่อการสอน

บรรยาย

สถานที่

จำนวนคนฟังหรือชม คน

บันทึก ปิดหน้าต่าง

รูปที่ 4.31 แสดงหน้าต่างการบันทึกข้อมูลงานบรรยาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อการสอน

อินเทอร์เน็ต

บริการ

บันทึก

บันทึก

ปิดหน้าต่าง

รูปที่ 4.32 แสดงหน้าต่างการบันทึกข้อมูลสื่ออินเทอร์เน็ต



ชื่อฟอร์ม	AgencyForm, AgencyHistForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับเก็บข้อมูลของผู้ผลิตงานโฆษณาประชาสัมพันธ์
กิจกรรม	สามารถเพิ่ม แก้ไข ลบ ยกเลิก บันทึกได้

บันทึกข้อมูลบริษัทผู้ผลิตโฆษณาประชาสัมพันธ์

รหัสบริษัทผู้ผลิตโฆษณา:

ชื่อบริษัทผู้ผลิตโฆษณา:

ที่อยู่ของบริษัทผู้ผลิตโฆษณา:

รหัสไปรษณีย์:

หมายเลขโทรศัพท์:

หมายเลขโทรสาร:

รายละเอียดผลงานที่ผ่านมาของผู้ผลิตโฆษณา

รหัสบริษัท	ชื่องาน	ชื่อหน่วยงานที่จ้าง	ราคา
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

ผลงานผู้ผลิตโฆษณา

รูปที่ 4.33 แสดงหน้าต่างการบันทึกข้อมูลบริษัทผู้ผลิตงานโฆษณาประชาสัมพันธ์

ข้อมูลผลงานที่ผ่านมาของผู้ผลิตโฆษณา

รหัสบริษัทผู้ผลิตโฆษณา:

ชื่องาน:

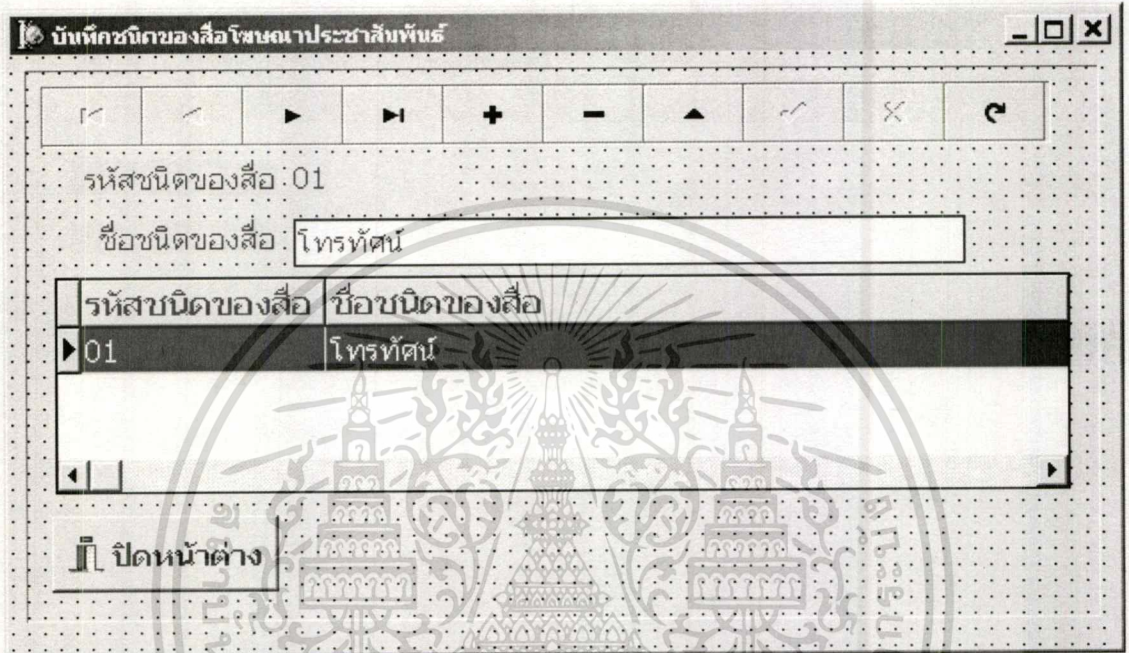
ชื่อหน่วยงานที่จ้าง:

ราคา: บาท

รูปที่ 4.34 แสดงหน้าต่างการบันทึกข้อมูลผลงานที่ผ่านมาของผู้ผลิตโฆษณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟอร์ม	TypeMediaForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับเก็บข้อมูลประเภทสื่อต่าง ๆ
กิจกรรม	สามารถเพิ่ม แก้ไข ลบ ยกเลิก บันทึกได้



รูปที่ 4.35 แสดงหน้าต่างการบันทึกข้อมูลชนิดของสื่อ

ชื่อฟอร์ม	OperAgencyForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับเก็บข้อมูลหัวข้องานโฆษณาประชาสัมพันธ์ว่ามีบริษัทผู้ผลิตโฆษณารายไหนบ้างมารับประมูลไปทำ
กิจกรรม	สามารถเพิ่มผู้ผลิตโฆษณา ลบ และบันทึกได้

การดำเนินการโดยบริษัทผู้ผลิตโฆษณา

บันทึกการดำเนินการโดยบริษัทผู้ผลิตโฆษณา

โครงการ: งานโฆษณาประชาสัมพันธ์

รหัสโครงการ	ชื่อโครงการ	ชื่องานโฆษณาประชาสัมพันธ์	ลักษณะงาน
001	ข่าวกล้องเบอร์ 5		
002	ตู้เย็นเบอร์ 5		
004	ตู้เย็นประหยัดไฟฟ้าฉลาก 2001		

หัวข้องานโฆษณาประชาสัมพันธ์:

ชื่อหัวข้องานโฆษณาประชาสัมพันธ์	วัตถุประสงค์

ชื่อบริษัทผู้ผลิตโฆษณา

บริษัทผู้ผลิตโฆษณา

ไปหน้าต่าง

รูปที่ 4.36 แสดงหน้าต่างการบันทึกการดำเนินงานโดยผู้ผลิตโฆษณา

ชื่อฟอร์ม	OperEmpForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับเก็บข้อมูลว่าพนักงานในองค์กรคนไหนรับทำงาน โฆษณาประชาสัมพันธ์ไป
กิจกรรม	เพิ่มข้อมูลพนักงานที่รับทำงาน โฆษณาประชาสัมพันธ์

การดำเนินการโดยพนักงาน

บันทึกการดำเนินการโดยพนักงาน

โครงการ : งานโฆษณาประชาสัมพันธ์

รหัสโครงการ	ชื่อโครงการ	ชื่องานโฆษณาประชาสัมพันธ์	ลักษณะงาน
001	ข่าวกล้องเบอร์ 5		
002	ตู้เย็นเบอร์ 5		
004	ตู้เย็นประหยัดไฟฟ้าลากล 2001		

หัวข้องานโฆษณาประชาสัมพันธ์ :

ชื่อหัวข้องานโฆษณาประชาสัมพันธ์	วัตถุประสงค์

ชื่อพนักงานที่ดำเนินการ :

พนักงาน :

ไปคนหน้าต่าง

รูปที่ 4.37 แสดงหน้าต่างการบันทึกข้อมูลการดำเนินการโดยพนักงาน

ชื่อฟอร์ม	FollowUpForm
วัตถุประสงค์	เป็นหน้าต่างสำหรับการดูข้อมูลติดตามการดำเนินงาน
กิจกรรม	สามารถดูงานที่ยังไม่ได้ทำและงานที่เกินกำหนดการได้

การติดตามงานโฆษณาประชาสัมพันธ์

งานที่ยังไม่ได้ดำเนินการ | งานที่ดำเนินการตามกำหนด

โครงการ DBLookupComboBox1

ชื่อหัวข้องานโฆษณาประชาสัมพันธ์ วันที่วางแผนเริ่มต้นผลิตสื่อ วันสุดท้ายการผลิตสื่อ

ปิดหน้าต่าง

รูปที่ 4.38 แสดงหน้าต่างการติดตามงานโฆษณาประชาสัมพันธ์

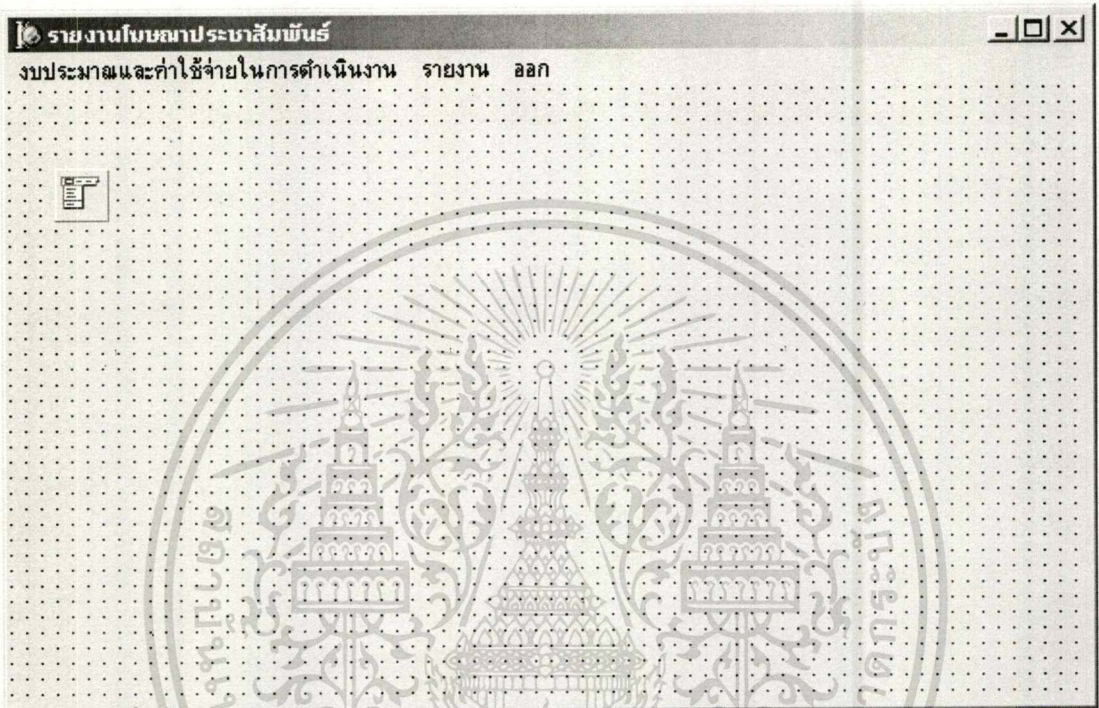
ชื่อฟอร์ม	EvaluationForm
วัตถุประสงค์	เป็นหน้าตาต่างสำหรับการดูข้อมูลการดำเนินการ
กิจกรรม	ดูงบประมาณที่ใช้จ่ายไป

[ดู การประเมินผล] - 10 x					
การประเมินงานโฆษณาประชาสัมพันธ์					
ค่าใช้จ่าย แผนการดำเนินงาน					
ชื่อโครงการ		ผลิตสื่อโฆษณา	วางสื่อโฆษณา	ผลิตสื่อประชาสัมพันธ์	วางสื่อประชาสัมพันธ์
งานโฆษณา			งานประชาสัมพันธ์		
ชื่องานโฆษณา	ผลิตสื่อโฆษณา	วางสื่อโฆษณา	ชื่องานประชาสัมพันธ์	ผลิตสื่อประชาสัมพันธ์	วางสื่อประชาสัมพันธ์
หัวข้องานโฆษณา			หัวข้องานประชาสัมพันธ์		
ชื่อหัวข้องานโฆษณา	ผลิตสื่อ (ใช้จริง)	วางสื่อ (ใช้จริง)	ชื่อหัวข้องานประชาสัมพันธ์	ผลิตสื่อ (ใช้จริง)	วางสื่อ (ใช้จริง)
[] ปิดหน้าต่าง					

รูปที่ 4.39 แสดงหน้าต่างการประเมินงานโฆษณาประชาสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟอร์ม	MonitorForm
วัตถุประสงค์	เป็นเมนูเมนูสำหรับ Monitor และพิมพ์รายงาน
กิจกรรม	คลิกเปิดหน้าต่าง Monitoring และ Reporting



รูปที่ 4.40 แสดงหน้าต่างการเมนูของการ Monitoring งานโฆษณาประชาสัมพันธ์

ชื่อฟอร์ม	BudgetForm
วัตถุประสงค์	สำหรับค่าใช้จ่ายในการดำเนินงาน
กิจกรรม	แสดงผลตามข้อมูลที่มีในฐานข้อมูล

๒๑ งบประมาณงานในเขตประชาสัมพันธ์ - ๒๑ x

งบประมาณงานโฆษณาประชาสัมพันธ์

ค่าใช้จ่าย | แผนการดำเนินงาน

ชื่อโครงการ	ผลิตสื่อโฆษณา	วางสื่อโฆษณา	ผลิตสื่อประชาสัมพันธ์	วางสื่อประชาสัมพันธ์	
งานโฆษณา			งานประชาสัมพันธ์		
ชื่องานโฆษณา	ผลิตสื่อโฆษณา	วางสื่อโฆษณา	ชื่องานประชาสัมพันธ์	ผลิตสื่อประชาสัมพันธ์	วางสื่อประชาสัมพันธ์
หัวข้องานโฆษณา			หัวข้องานประชาสัมพันธ์		
ชื่อหัวข้องานโฆษณา	ผลิตสื่อ (ใช้จริง)	วางสื่อ (ใช้จริง)	ชื่อหัวข้องานประชาสัมพันธ์	ผลิตสื่อ (ใช้จริง)	วางสื่อ (ใช้จริง)

๓. ปิดหน้าต่าง

รูปที่ 4.41 แสดงหน้าต่างงบประมาณงานโฆษณาประชาสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟอร์ม	Reporting
วัตถุประสงค์	มีเมนูสำหรับเลือกเงื่อนไขและพิมพ์รายงาน
กิจกรรม	มีเงื่อนไขให้เลือกและพิมพ์รายงาน

รายงานกองประชาสัมพันธ์

งบประมาณค่าโฆษณาประชาสัมพันธ์ปี

รายงานการติดตามข้อมูลการผลิตและวางสื่อโฆษณาประชาสัมพันธ์

รายงานสถิติการวางสื่อในงานโฆษณาประชาสัมพันธ์

รายงานประวัติของโฆษณาประชาสัมพันธ์

รายงานประวัติของบริษัทโฆษณาประชาสัมพันธ์

เงื่อนไข

ปี

โครงการ DBLookupComboBox1

งานโฆษณาประชาสัมพันธ์ DBLookupComboBox6

ยกเลิกเงื่อนไข

 Print Preview

ปิด

รูปที่ 4.42 แสดงหน้าต่างเมนูการพิมพ์รายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการดำเนินการศึกษา

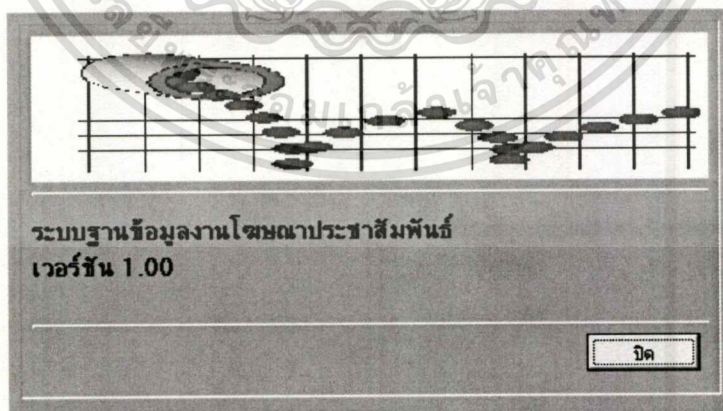
5.1 ผลการดำเนินการศึกษา

โปรแกรมที่ได้ทำการเขียนขึ้นจากการออกแบบระบบงานโฆษณาประชาสัมพันธ์ มาจาก Use Case Diagram รวมไปถึง Sequence Diagram เพื่อให้ใช้งานได้ตรงตามที่ได้วิเคราะห์และออกแบบ รูปข้างล่างเป็นการอิมพีเมนต์ระบบขึ้นมาโดยดูจากการวิเคราะห์และออกแบบ

5.1.1 การใช้งานระบบตามฟังก์ชันที่ได้ออกแบบไว้

Actor – พนักงานที่ทำหน้าที่บันทึกข้อมูลเข้าระบบ

มีฟังก์ชันการทำงานดังนี้ เรียกโปรแกรม ADPRSystem.exe ขึ้นมาจะปรากฏหน้าต่างดังรูปที่ 5.1 หลังจากนั้นระบบจะมีหน้าต่างถาม Username และ Password เพื่อเป็นการรักษาความปลอดภัยให้กับระบบ



รูปที่ 5.1 หน้าต่างแสดงชื่อโปรแกรม

Database Login

Database: ADOCADPRDB

User Name: pr01

Password:

OK Cancel

รูปที่ 5.2 หน้าต่างล็อกอินเข้าระบบ

Use Case Project Planning

แสดงหน้าต่าบบันทึกรบประมาณของ Use Case Project Planning ถ้าล็อกอิน Username , Password ถูกต้องจะเห็นหน้าต่าเมนูระบบหลักดังรูปข้างล่างเป็นฟังก์ชันงานของการบันทึกรบประมาณเพื่อกอตรวจสอบการดำเนินการ

บันทึกชื่อโครงการ

รหัสโครงการ	ชื่อโครงการ
001	โครงการหลอดประหยัดไฟฟ้า
002	โครงการแอร์เบอร์ 5
003	โครงการผู้เข็นรถลาก 2001

รหัสโครงการ 003

ชื่อโครงการ โครงการผู้เข็นรถลาก 2001

วันที่เริ่มโครงการ 03/04/2001

วันที่สิ้นสุดโครงการ 02/04/2002

งบประมาณการโฆษณา

งบประมาณการผลิตสื่อ 25,000,000.00 บาท

งบประมาณการวางสื่อ 25,000,000.00 บาท

งบประมาณการประชาสัมพันธ์

งบประมาณการผลิตสื่อ 10,000,000.00 บาท

งบประมาณการวางสื่อ 12,000,000.00 บาท

หน่วยงานที่เป็นเจ้าของโครงการ

รหัสประเภทอุปกรณ์ไฟฟ้า

บันทึกรายละเอียดโครงการ

ไป หน้าต่า

เพิ่ม แก้ไข ลบ ยกเลิก บันทึก

รูปที่ 5.3 หน้าต่าบบันทึกรบประมาณแต่ละโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบงานบัญชีงบประมาณรายจ่ายประจำปี - [วางแผนโครงการ]

งานโครงการรายจ่ายประจำปี ร้อยละ 50% ช่วยเหลือ

บันทึกรายละเอียดโครงการ

ชื่อโครงการ โครงการหลอดประหยัดไฟฟ้า โครงการแอร์เบอร์ 5 โครงการตู้เย็นฉลาด 2001	ชื่องานโฆษณาประชาสัมพันธ์ ใช้หลอดคอมช่วยชาติ ใช้หลอดคอมประหยัดไฟฟ้าในบ้าน
---	---

ลักษณะงาน โฆษณา ประชาสัมพันธ์

วันที่เริ่มดำเนินงาน 10/02/2000 วันที่สิ้นสุดงาน 10/03/2000

รหัสโฆษณาประชาสัมพันธ์ 00102

ชื่อโฆษณาประชาสัมพันธ์ ใช้หลอดคอมประหยัดไฟฟ้าในบ้าน

งบประมาณ
การผลิตสื่อ 300,000.00 บาท การวางสื่อ 1,000,000.00 บาท

เพิ่ม แก้ไข ลบ ยกเลิก บันทึก

ชื่องาน	ผลิตสื่อ (ใช้จริง)	วางสื่อ (ใช้จริง)

รายละเอียดงานโฆษณาประชาสัมพันธ์ ปิดหน้าต่าง

รูปที่ 5.4 หน้าต่างบันทึกรายละเอียดการดำเนินงานแต่ละโครงการ

ระบบงานบัญชีงบประมาณรายจ่ายประจำปี - [วางแผนโครงการ]

งานโครงการรายจ่ายประจำปี ร้อยละ 50% ช่วยเหลือ

บันทึกรายละเอียดงานโฆษณาประชาสัมพันธ์

รหัสงานโฆษณาประชาสัมพันธ์ 00101 ใช้หลอดคอมช่วยชาติ

รหัสหัวข้องานโฆษณาประชาสัมพันธ์ 0010101

ชื่อหัวข้องานโฆษณาประชาสัมพันธ์

งบประมาณ

การผลิตสื่อ 1,000,000.00 บาท	การวางสื่อ 1,000,000.00 บาท
ใช้จริง 980,000.00 บาท	ใช้จริง 1,000,000.00 บาท

แผนการวันที่ เริ่มจริงวันที่

วันที่เริ่มต้น		วันที่เริ่มต้น	
วันที่สิ้นสุด		วันที่สิ้นสุด	

วัตถุประสงค์

กลุ่มเป้าหมาย

เพิ่ม แก้ไข ลบ ยกเลิก บันทึก

รายละเอียดสื่อโฆษณา ปิดหน้าต่าง

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการดำเนินงาน โดยผู้จัดทำเอกสารนี้ไม่รับผิดชอบต่อเนื้อหาและข้อมูลใดๆ ที่ปรากฏในเอกสารนี้ และไม่รับประกันว่าข้อมูลใดๆ ที่ปรากฏในเอกสารนี้จะไม่มีการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Actor – ผู้บริหาร สามารถดูข้อมูลจากรายงานได้ว่าได้ดำเนินการอะไรไปบ้างแล้ว

มีฟังก์ชันการทำงานดังนี้ เรียกโปรแกรม ADPRReport.exe ขึ้นมาจะปรากฏหน้าต่างให้ล็อกอินเข้าระบบเพื่อพิมพ์รายงาน โดยเลือกเงื่อนไข

Use Case Monitoring และ Use Case Reporter



รูปที่ 5.6 หน้าต่างของรายงานการใช้เงินงบประมาณในการดำเนินการงานโฆษณาประชาสัมพันธ์

Print Preview

การใช้จ่ายเงินค่าโฆษณาประชาสัมพันธ์ปี

ชื่อโครงการ	โครงการ	งบประมาณ			เบิกจ่ายจริง		
		โฆษณา	ประชาสัมพันธ์	รวม	โฆษณา	ประชาสัมพันธ์	รวม
ใช้หลอดหอมช่วยชาติ	ผลิต	0.00	200,000.00	200,000.00	0.00	100,180.00	100,180
	วางสื่อ	0.00	200,000.00	200,000.00	0.00	92,500.00	92,500
	รวม	0.00	400,000.00	400,000.00	0.00	192,680.00	192,680
ทุกที่มีหลอดหอมใช้ ช่วยชาติประหนึ่งผลิตง	ผลิต	0.00	100,000.00	100,000.00	0.00	30,000.00	30,000
	วางสื่อ	0.00	200,000.00	200,000.00	0.00	80,000.00	80,000
	รวม	0.00	300,000.00	300,000.00	0.00	110,000.00	110,000
ใช้หลอดหอมประหยัดไฟฟ้าในบ้าน	ผลิต	30,000.00	0.00	30,000.00	15,000.00	0.00	15,000
	วางสื่อ	50,000.00	0.00	50,000.00	20,000.00	0.00	20,000
	รวม	80,000.00	0.00	80,000.00	35,000.00	0.00	35,000
ข้ากลองเบอร์ 5	ผลิต	5,555,000.00	0.00	5,555,000.00	6,250.00	0.00	6,250
	วางสื่อ	1,000,000.00	0.00	1,000,000.00	30,000.00	0.00	30,000
	รวม	6,555,000.00	0.00	6,555,000.00	36,250.00	0.00	36,250

Page 1 of 3

รูปที่ 5.7 ตัวอย่างรายงานการใช้จ่ายเงินค่าโฆษณาประชาสัมพันธ์

Print Preview

การติดตามข้อมูลการผลิตและวางสื่อโฆษณาประชาสัมพันธ์

ชื่อโครงการ	โครงการ	งบประมาณ(บาท)	แผนงาน		ผลการดำเนินงาน	
			เริ่มต้น	สิ้นสุด	เบิกจ่ายแล้ว	คงเหลือเงิน
ใช้หลอดหอมทุกบ้านช่วยชาติ	งานโฆษณาประชาสัมพันธ์	200.00	15/09/2000	15/10/2000	180.00	20.00
	ผลิต					
	วางสื่อ					
เสริมความรู้กับเด็กนักเรียน	งานโฆษณาประชาสัมพันธ์	100,000.00	15/09/2000	20/12/2000	100,000.00	0.00
	ผลิต					
	วางสื่อ					
รวมงานโฆษณาประชาสัมพันธ์	งานโฆษณาประชาสัมพันธ์	185,700.00	15/09/2000	20/12/2000	192,000.00	-7,000.00
	ผลิต					
	วางสื่อ					
โฆษณา 1 / หัวข้อ 1	งานโฆษณาประชาสัมพันธ์	20,000.00	15/09/2000	15/10/2000	15,000.00	5,000.00
	ผลิต					
	วางสื่อ					
รวมงานโฆษณาประชาสัมพันธ์	งานโฆษณาประชาสัมพันธ์	50,000.00	15/09/2000	20/12/2000	35,000.00	15,000.00
	ผลิต					
	วางสื่อ					

Page 1 of 4

รูปที่ 5.8 ตัวอย่างรายงานการติดตามข้อมูลการผลิตและวางสื่อโฆษณาประชาสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่แสดงเป็นตัวอย่างผ่านมาเป็นหน้าต่างที่ได้มาจากการวิเคราะห์และออกแบบเพื่อนำมาสร้างเป็นโปรแกรมให้สามารถใช้งานได้ รวมไปถึงรายงาน ซึ่งการสร้างโปรแกรมในโครงการนี้อาจจะยังไม่สมบูรณ์ จึงต้องมีการพัฒนากันต่อไปเพื่อให้ได้งานที่ต้องตามความต้องการของผู้ใช้งาน

เนื่องจากการวิเคราะห์ในขั้นตอนแรกอาจจะไม่ได้ไคอะแกรมที่ถูกต้องที่สุดดังนั้นในขั้นตอนของ OOD จะต้องมีการ Refinement คือกระบวนการเพิ่มเติมหรือทำให้เหมาะสมจากการวิเคราะห์ เพื่อให้ง่ายต่อการนำมาพัฒนาเป็นระบบในเครื่องคอมพิวเตอร์ได้ในที่สุด การทำ Refinement ทำได้ทุก ๆ ไคอะแกรม ไม่ว่าจะเป็น Use Case Diagram, Class Diagram, Sequence Diagram หรือ State Diagram

ในการทำ Refinement จะมีผลกระทบระหว่างไคอะแกรมเสมอ การทำ Refinement ต้องมีการเรียงลำดับจาก Use Case Diagram -> Class Diagram -> Sequence Diagram -> State Diagram -> Use Case -> Class Diagram ... เมื่อทำ Refinement แล้วต้องทำให้ Diagram ต่าง ๆ ที่เคยทำไว้มีคอมพิวเตอร์มาเกี่ยวข้องเช่นการเพิ่ม Class ของ User Interface เข้าไปยัง Class Diagram และเพิ่มฟังก์ชันการอินพุตข้อมูลเข้าไปและการแสดง attribute ไว้ในทุก ๆ Class

บทที่ 6

สรุปผลการศึกษาและข้อเสนอแนะ

6.1 สรุปผลการศึกษา

สำหรับการศึกษาโครงการงานโฆษณาประชาสัมพันธ์ได้นำเอาวิธีการวิเคราะห์และออกแบบรวมถึงการสร้าง โปรแกรม ซึ่งใช้ UML เป็นสัญลักษณ์ที่เป็นรูปภาพสำหรับการออกแบบ

ไคอะแกรมของ UML ช่วยให้ผู้ออกแบบสามารถแสดงความคิดเป็นรูปภาพ จะช่วยให้โปรแกรมเมอร์ให้เข้าใจตัวระบบมากขึ้น โดยจะอธิบายการทำงานของระบบหลาย ๆ ไคอะแกรม อย่างเช่น use case diagram จะอธิบายถึงขอบเขตที่มีในระบบว่ามีฟังก์ชันอะไรบ้าง ที่ผู้ใช้งานจะเข้ามาติดต่อกับระบบโดยตรงหรือโดยอ้อม ซึ่งจะเหมือนกับ DFD ใน Context diagram แต่ DFD จะกล่าวถึงเส้นทางเดินของข้อมูลมากกว่า กระบวนการออกแบบของ UML ใช้ในลักษณะการวางแผน และแก้ไขเพิ่มเติม ไคอะแกรมไปเรื่อยๆจนกว่าจะสมบูรณ์ เป็นข้อดีเพราะในบางครั้งจะไม่สามารถวิเคราะห์และออกแบบในครั้งเดียวแล้วสมบูรณ์ กระบวนการวิเคราะห์และออกแบบสามารถนำไคอะแกรมต่าง ๆ มาแก้ไขหรือเพิ่มเติมส่วนที่ขาดหายไปจนครบ

ไคอะแกรมของ UML มีไคอะแกรมที่สำคัญ ๆ ใช้ในการวิเคราะห์และออกแบบอย่างมีประสิทธิภาพ ถ้าเป็น Static Diagram คือ ไคอะแกรมที่แสดงภาพลักษณะคงที่ของขอบเขตงานคือการมีอยู่ของคลาสต่าง ๆ และความสัมพันธ์ของคลาสนั้นๆ ในระบบ โดยไม่ได้แสดงกิจกรรมที่เกิดขึ้น ไคอะแกรมที่ได้พูดถึงในโครงการนี้คือ Use Case Diagram จะประกอบด้วยขอบเขตของปัญหาและความสัมพันธ์ของส่วนต่าง ๆ ไคอะแกรมที่สำคัญอีกอันคือ Class Diagram จะแสดงถึงคลาสที่มีทั้งหมดในระบบ ซึ่งคลาสหนึ่งจะมีความสัมพันธ์ไม่ว่าจะเป็น Aggregation, Generalization, Association กับคลาสนั้น ๆ อย่างน้อย 1 ความสัมพันธ์เสมอ

Dynamic Diagram คือ ไคอะแกรมที่แสดงภาพในเชิงกิจกรรม (Dynamic) ของฟังก์ชันต่าง ๆ ในระบบซึ่งในการวิเคราะห์จะมี 2 ไคอะแกรมคือ Sequence Diagram เป็นไคอะแกรมที่แสดงถึงกิจกรรมรวมของระบบ โดยกิจกรรมจะเรียกใช้งานฟังก์ชันที่มีอยู่ในคลาสนั้น ๆ นั้นเอง ส่วน State Diagram เป็นไคอะแกรมที่แสดงภาพที่ลึกลงไปของคลาสแต่ละตัวว่ามีสถานะอะไรบ้างและจะเปลี่ยนสถานะของคลาสเมื่อใด

6.2 ข้อเสนอแนะ

1. การศึกษาและดำเนินการในโครงการนี้ ได้ใช้ Relational Database เป็นฐานข้อมูลซึ่งต้องมีขั้นตอนการแปลงจาก Class Diagram มาเป็น Relational Database จึงมีข้อเสนอแนะว่าในอนาคตน่าจะมีการศึกษาถึง Object Database ว่าการทำ Persistent Database ที่แปลงจาก Class Diagram มาเป็น Object Database มีความสะดวกกว่าหรือไม่
2. ต้องฝึกฝนการนำประโยชน์ของ UML เรื่องการ Reuse มาใช้ให้เกิดประโยชน์สูงสุด ซึ่งต้องใช้ประสบการณ์ในการวิเคราะห์และออกแบบระบบที่หลากหลาย



บรรณานุกรม

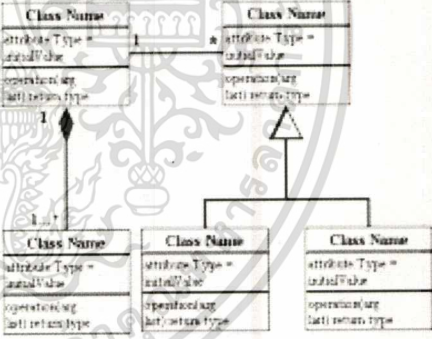
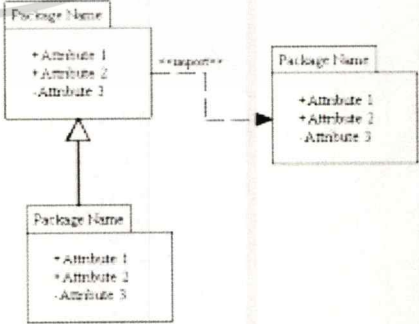
- กิตติ ภัคดีวัฒนะกุลและกิตติพงษ์ กลมกล่อม. 2544. **UML วิเคราะห์และออกแบบระบบเชิงวัตถุ**.
: เคทีพี คอมพ์ แอนด์ คอนซัลท์.
- เทพพิทักษ์ การุญบุญญานันท์และไพศาล เจริญพรสวัสดิ์และจุฬารัตน์ ต้นประเสริฐ. 2541.
“UML : ภาษาพิมพ์เขียวซอฟต์แวร์ (ตอนที่ 1)”. ไมโครคอมพิวเตอร์. ฉบับที่ 160 :
107-112.
- สมนึก คีรีโต. **แนวทางการพัฒนาระบบเชิงออปเจกต์**. ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะ
วิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.
- Hans-Erik Eriksson and Magnus Penker. 1998. **UML Toolkit**. : John Wiley & Sons.
- SmartDraw.com. 2001. **HOW TO DRAW UML DIAGRAMS**. [online]. Available :
<http://www.smartdraw.com/resources/centers/uml/uml.htm>.
- Terry Quatrani. 2000. **Visual modeling with Rational Rose 2000 and UML**. : Addison
Wesley.

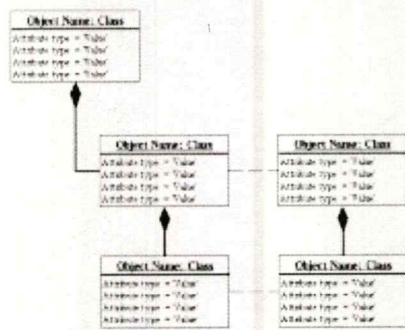
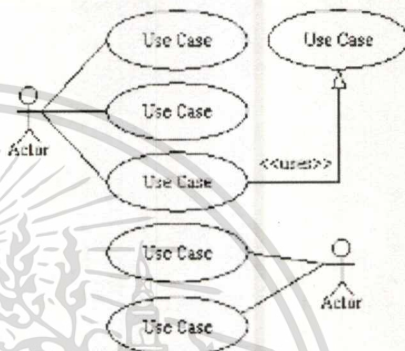
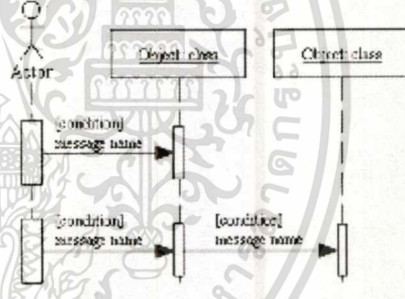
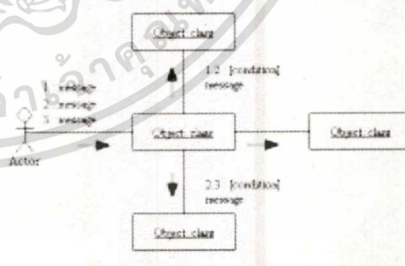
ภาคผนวก ก

สัญลักษณ์ต่าง ๆ ที่ใช้ในการวาดรูปสำหรับ UML (Unified Modeling Language) สัญลักษณ์ที่ใช้ในการวาดซึ่งนำข้อดีในการออกแบบของบุคคลที่มีชื่อเสียงด้าน Object-Oriented ทั้ง 3 ท่าน Grady Booch, James Rumbaugh, Ivar Jacobson นำมาผสมกัน

ชนิดของ UML Diagrams

UML ได้กำหนดให้มี 9 ชนิดของไดอะแกรม : Class (Package), Object, Use Case, Sequence, Collaboration, Statechart, Activity, Component และ deployment

ไดอะแกรม	รูปของไดอะแกรม
<p>1. Class Diagrams เป็น ไดอะแกรมหลักของ object oriented method รวมไปถึง UML ซึ่งจะอธิบายโครงสร้างแบบ static ของระบบ</p>	 <p>The diagram illustrates three types of class relationships: <ul style="list-style-type: none"> Association: Two class boxes connected by a solid line with open arrowheads at both ends. Multiplicity '1' is shown at each end. Inheritance: A solid line with an open arrowhead pointing to the superclass and a hollow triangle at the subclass end. Composition: A solid line with a hollow triangle at the whole end and a solid arrowhead at the part end. </p>
<p>Package Diagrams เป็นซัพเซตของ Class Diagrams แต่นักพัฒนาระบบบางครั้งต้องการที่จะแยกออกมา ซึ่ง Package Diagram จะช่วยจัดการอติเมนต์ของระบบที่มีความสัมพันธ์กันระหว่างกลุ่มให้ขึ้นต่อกันน้อยที่สุดระหว่าง packages</p>	 <p>The diagram illustrates two types of package relationships: <ul style="list-style-type: none"> Inheritance: A solid line with a hollow triangle pointing to the superclass package. Dependency: A dashed line with an open arrowhead pointing to the required package. </p>

<p>2. Object Diagram จะอธิบายถึงโครงสร้าง static ของระบบ ซึ่งสามารถใช้ทดสอบ Class Diagrams ได้ว่าถูกต้องหรือไม่</p>	 <p>The diagram shows a hierarchy of objects. At the top is an object labeled 'Object Name: Class' with four attributes, each set to 'Value'. Below it are two more objects of the same class, each with four 'Value' attributes. These two objects are connected to the top object by solid lines. Below these are two more objects of the same class, each with four 'Value' attributes. These two objects are connected to the two objects above them by solid lines.</p>
<p>3. Use Case Diagrams เป็นไดอะแกรมโมเดลที่มีบอกรหัสของระบบ โดยการใช้ actors และ use cases</p>	 <p>The diagram shows an actor (a stick figure) connected to five use cases (ovals). One use case is labeled '<<usecase>>'. Another use case is connected to a second actor. There are also some inheritance-like lines between use cases.</p>
<p>4. Sequence Diagrams จะอธิบายการโต้ตอบระหว่างคลาสโดยการแลกเปลี่ยนเมสเสจที่เกิดขึ้นตามเวลา</p>	 <p>The diagram shows an actor and two objects. The actor sends a 'conditional message name' to the first object. The first object then sends a 'conditional message name' to the second object. The second object sends a 'conditional message name' to the first object. There are also some dashed lines representing return messages.</p>
<p>5. Collaboration Diagrams นำเสนอการโต้ตอบระหว่าง Object ที่เป็นลำดับของเมสเสจ ไดอะแกรมนี้จะอธิบายโครงสร้างที่เป็น static และ dynamic ของระบบ</p>	 <p>The diagram shows an actor and three objects. The actor sends a 'message' to the first object. The first object sends a '1:2 Found() message' to the second object. The second object sends a '2:3 Found() message' to the third object. There are also some dashed lines representing return messages.</p>

<p>6. Statechart Diagrams อธิบายถึงพฤติกรรมที่เป็น dynamic ของระบบในการตอบสนองจากสิ่งเร้าภายนอก ซึ่งแต่ละ State จะถูกกระตุ้นโดยเหตุการณ์ต่าง ๆ</p>	<p>The diagram shows a state machine starting from an initial state (solid black circle) and ending at a final state (solid black circle with a white center). It consists of four states, each labeled 'State do/activity'. Transitions between states are triggered by 'Event (Guard) / Action'.</p>
<p>7. Activity Diagrams แสดงให้เห็น dynamic ของระบบโดยโมเดลที่เป็นการไหลของกิจกรรมหนึ่งไปยังกิจกรรมหนึ่ง โดยกิจกรรมจะแสดงถึงการทำงานกับคลาสในระบบโดยผลลัพธ์ถูกเปลี่ยนสถานะของระบบ ปกติแล้วไดอะแกรมถูกใช้กับ โมเดล workflow หรือ business processes และการทำงานภายใน</p>	<p>The diagram illustrates a flow of activities. It starts with an initial node leading to an 'Activity' node. This activity is associated with a ':Class'. The flow continues through several other 'Activity' nodes, each associated with a ':Class'. The process ends at a final node. Dashed lines indicate dependencies between activities and classes.</p>
<p>8. Component Diagrams ใช้อธิบายถึงฟังก์ชันของซอฟต์แวร์คอมพิวเตอร์ที่รวมเอาซอร์สโค้ด, รันไทม์โค้ด และไฟล์เอ็กซিকิวต์</p>	<p>The diagram shows three components, each represented by a rectangle with three horizontal lines on the left side. One component is at the top, one at the bottom left, and one at the bottom right. Solid arrows with open heads point from the bottom-left and bottom-right components to the top component, indicating dependencies. A dashed arrow points from the bottom-right component to the top component, indicating a provided interface.</p>
<p>9. Deployment Diagrams ไดอะแกรมได้อธิบายถึงฟิสิกัลริซอร์สในระบบรวมถึงคอมพิวเตอร์เน็ตและการเชื่อมต่อ</p>	<p>The diagram shows a hierarchical structure of nodes. At the top is a 'Node' labeled 'rater' containing three 'Component' boxes. Below it are two more 'Node' boxes, one labeled 'state' and another labeled 'rater' containing two 'Component' boxes. Lines connect the top node to the two bottom nodes, representing a deployment or distribution structure.</p>

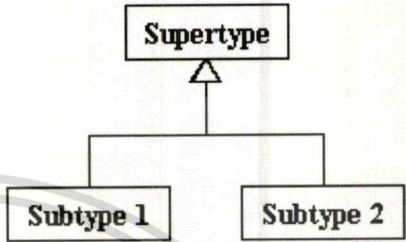

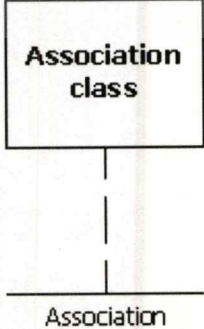
รายละเอียดของแต่ละคลาส

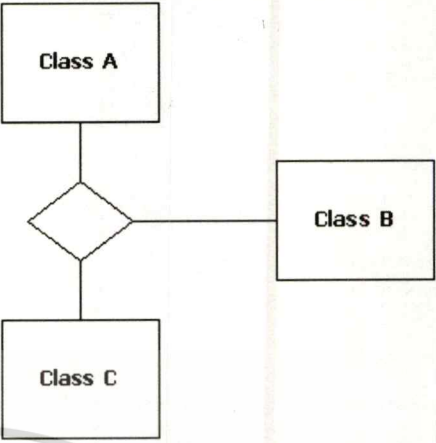
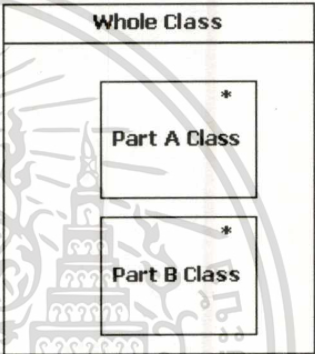




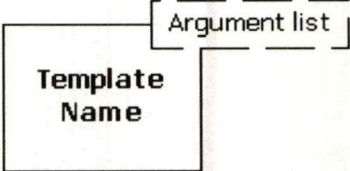
1. Class Diagram

สัญลักษณ์และการเขียนคลาสไดอะแกรม คลาสจะแสดง abstraction ของเอนติตี้ด้วยคุณสมบัติพื้นฐาน สามารถใช้แสดงความสัมพันธ์ระหว่างคลาสได้

สัญลักษณ์ที่ใช้ในคลาสไดอะแกรม	รูปลักษณะ
<p>1. Class (คลาส) แสดงเป็นรูปสี่เหลี่ยมที่แบ่งเป็นสามส่วน</p> <ul style="list-style-type: none"> - ส่วนที่ 1 ชื่อของคลาสอยู่ในส่วนแรกตรงกลางตัวอักษรตัว หนาและตัวใหญ่ - ส่วนที่ 2 แอตทริบิวต์ - ส่วนที่ 3 โอเปอเรชันในส่วนที่สาม 	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center;">Class Name</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p>attribute:Type = initialValue</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>operation(arg list):return type</p> </div>
<p>2. Active Class เป็นจุดเริ่มต้นการไหลของกิจกรรม ขณะที่ passive classes เก็บข้อมูลและให้บริการคลาสนั้น โดย active class จะแสดงโดยเส้นขอบที่หนา</p>	<div style="border: 3px double black; padding: 5px; text-align: center;"> <p>Active class</p> </div>
<p>3. Visibility</p> <p>ใช้ Visibility เป็นตัวมาร์กบอกเครื่องหมายว่าใครสามารถจะแอกเซสข้อมูลที่อยู่ในคลาสได้บ้าง โดย</p> <p><i>Private</i> จะซ่อนข้อมูลจากภายนอกคลาสนั้น</p> <p><i>Public</i> จะอนุญาตให้คลาสนั้นทั้งหมดสามารถมองเห็นได้</p> <p><i>Protected</i> จะอนุญาตให้คลาสลูกแอกเซสข้อมูลได้จากการ inherited จากคลาสนั้น</p>	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px;"> <p>+ <i>public</i></p> <p>- <i>private</i></p> <p># <i>protected</i></p> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Class Name</p> <p>- attribute</p> <p>- attribute</p> <p>+ operation</p> <p>+ operation</p> <p>+ operation</p> </div> </div>
<p>4. Associations</p> <p>Association จะแสดงความสัมพันธ์แบบสแตติกระหว่างคลาสโดยการแทนชื่อ association ข้างบนเส้น association โดยเติมลูกศรเพื่อแสดงทิศทางของความสัมพันธ์ และกฎจะอยู่ปลายทางของคลาสนั้นทั้งสอง</p>	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Class A</div> <div style="text-align: center;"> <p>name</p> </div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Class B</div> </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Class A</div> <div style="text-align: center;"> <p>role</p> <p>role</p> </div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Class B</div> </div>

<p>5. Multiplicity (Cardinality)</p> <p>การเขียนเงื่อนไขที่ปลายทางของ association สัญลักษณ์เหล่านี้จะแสดงถึงจำนวนของ instances ของคลาส ตัวอย่างบริษัท 1 บริษัท จะมีลูกจ้างตั้งแต่ 1 คนหรือมากกว่านั้นแต่จะทำงานกับบริษัทเดียวเท่านั้น</p>	<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 20px;"> <p>1 <i>no more than one</i></p> <p>0..1 <i>zero or one</i></p> <p>* <i>many</i></p> <p>0..* <i>zero or many</i></p> <p>1..* <i>one or many</i></p> </div> <div style="text-align: center;"> </div> </div>
<p>6. Constraint</p> <p>วาง constraint ไว้ภายในเครื่องหมายปีกกา</p>	<div style="text-align: center;"> <p><i>Simple Constraint</i></p> </div> <div style="text-align: center; margin-top: 20px;"> <p><i>Complex Constraint</i></p> </div>
<p>7. Composition และ Aggregation</p> <p>Composition คือชนิดเฉพาะของ aggregation ด้วยการแสดงความเป็นเจ้าของระหว่างคลาส A (whole คลาส) และ คลาส B (part คลาส) แสดงด้วยรูปข้าวหลามตัดทึบ และใช้รูปข้าวหลามตัดที่โปร่งเพื่อแสดงความสัมพันธ์ aggregation อย่างง่ายซึ่ง “whole” คลาสมีความสำคัญมากกว่า “part” คลาสแต่คลาสทั้ง</p>	<div style="text-align: center;"> </div>

<p>สองไม่ขึ้นต่อกัน รูปข้าวหลามตัดที่ปลาย ความสัมพันธ์ทั้ง composition และ aggregation ซึ่งไปยัง “whole” คลาสหรือ aggregate</p>	
<p>8. Generalization</p> <p>Generalization คือชื่ออีกอย่างของ inheritance หรือความสัมพันธ์ “is a” จะเป็น การอ้างถึงความสัมพันธ์ระหว่างคลาส 2 คลาส โดยที่คลาสหนึ่งเป็นเวอร์ชันเฉพาะของ คลาสอื่น ๆ ตัวอย่าง Honda เป็นประเภทหนึ่งของ รถ ดังนั้นคลาส Honda จะมี generalization relationship กับคลาส Car ตัวอย่าง จริงของการไล่ความแตกต่างระหว่าง inheritance และ aggregation จะสับสนถ้ามี ความสัมพันธ์แบบ aggregate (whole) สามารถ แอกเซสฟังก์ชัน PUBLIC ของ part คลาส ส่วน inheritance อนุญาตให้คลาสที่ inherit แอกเซสได้ทั้งฟังก์ชันที่เป็น PUBLIC และ PROTECTED ของซูเปอร์คลาส</p>	 <pre> classDiagram class Supertype class Subtype1 class Subtype2 Supertype < -- Subtype1 Supertype < -- Subtype2 </pre>
<p>9. Qualified association</p>	 <pre> classDiagram class Class class Qualifier Class --- Qualifier </pre>
<p>10. Association class</p>	 <pre> classDiagram class AssociationClass class Association AssociationClass --- Association </pre>

<p>11. Ternary association</p>	
<p>12. Aggregation (alternative)</p>	
<p>13. Abstract class</p>	
<p>14. Dependency</p>	
<p>15. Refinement</p>	
<p>16. interface</p>	 <p style="text-align: right;">Interface name</p>
<p>17. Parameterized class (templace)</p>	

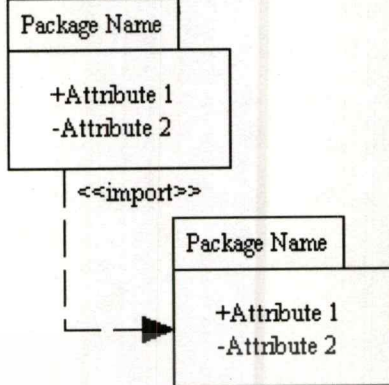
<p>18. Instantiated parameterized class</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p>Template Name <parameters></p> </div>
<p>19. Object</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;">Object Name:Class</p> <hr style="border: 0; border-top: 1px solid black; margin: 2px 0;"/> <p>attribute = value</p> </div>

Package Diagram

แพ็คเกจไดอะแกรมจะจัดการกับอิลิเมนต์ของระบบเข้าไปยังกลุ่มที่สัมพันธ์กัน โดยให้มีการขึ้นต่อกันน้อยที่สุด

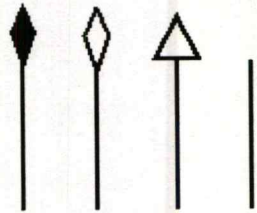
สัญลักษณ์ที่ใช้ในแพ็คเกจไดอะแกรม	รูปลักษณะ
<p>1. Packages ใช้ส่วนที่โผล่ออกมาจากแฟ้มให้เขียนชื่อของแพ็คเกจบนส่วนที่โผล่ออกมาหรือภายในแฟ้ม คล้าย ๆ กับคลาสและสามารถมีรายการของแอตทริบิวต์แพ็คเกจได้</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto 10px auto;"> <p>Package Name</p> </div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto 10px auto;"> <p>Package Name</p> </div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto;"> <p>+Attribute 1 +Attribute 2 -Attribute 3</p> </div> </div>
<p>2. Visibility มีตัวบอกเครื่องหมายว่าใครสามารถแอกเซสข้อมูลที่บรรจุภายในแพ็คเกจ</p> <p><i>Private</i> มีความหมายว่าแอตทริบิวต์หรือโอเปอเรชันไม่สามารถแอกเซสมาจากแพ็คเกจภายนอกได้</p> <p><i>Public</i> อนุญาตให้แอตทริบิวต์หรือโอเปอเรชันสามารถมองเห็นจากแพ็คเกจภายนอกได้</p> <p><i>Protected</i> ทำให้แอตทริบิวต์หรือโอเปอเรชันมองเห็นจากแพ็คเกจที่สืบทอดมาเท่านั้น</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p>+ <i>public</i> - <i>private</i> # <i>protected</i></p> </div>

3. Dependency กำหนดความสัมพันธ์ในส่วนที่เปลี่ยนแปลงจากแพ็คเกจหนึ่งจะมีผลต่ออีกแพ็คเกจหนึ่ง การ import คือประเภทของ dependency ที่ให้สิทธิ์แพ็คเกจหนึ่งสามารถแอกเซสเข้าไปยังอีกแพ็คเกจหนึ่ง



2. Object Diagram

ออบเจกต์ไคอะแกรมถูกเชื่อมต่อไปยังคลาสไคอะแกรม โดยออบเจกต์คือ instance ของคลาส ตัวออบเจกต์ไคอะแกรมสามารถถูกมองเห็นเป็น instance ของคลาสไคอะแกรม ออบเจกต์ไคอะแกรมอธิบายถึงโครงสร้างที่เป็นสแตติกของระบบ โดยถูกใช้เพื่อทดสอบความถูกต้องของคลาสไคอะแกรม

สัญลักษณ์ที่ใช้ในออบเจกต์ไคอะแกรม	รูปลักษณะ
<p>1. Object names แต่ละออบเจกต์ถูกแทนด้วยรูปสี่เหลี่ยมคางหมูซึ่งใส่ชื่อของออบเจกต์และคลาสของมันที่ขีดเส้นใต้และแยกกันด้วยเครื่องหมายโคลอน (:)</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p><u>Object name</u> : <u>Class</u></p> <p><i>Named object</i></p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p>: <u>Class</u></p> <p><i>Unnamed object</i></p> </div> <div style="border: 1px solid black; padding: 5px;"> <p><u>Object name</u> : <u>Class</u>::<u>Package</u></p> <p><i>Named object with path name</i></p> </div>
<p>2. Object attributed เหมือนกับคลาสสามารถมีรายการแอตทริบิวต์โดยแยกช่องออกมาต่างหาก อย่างไรก็ตามจะไม่เหมือนคลาสคือ object attributes ต้องมีค่าที่กำหนดไว้ก่อน</p>	<div style="border: 1px solid black; padding: 5px;"> <p><u>Object Name</u> : <u>Class</u></p> <p>Attribute type = 'Value'</p> <p>Attribute type = 'Value'</p> <p>Attribute type = 'Value'</p> <p>Attribute type = 'Value'</p> <p><i>Object with attributes</i></p> </div>
<p>3. Active object ออบเจกต์ที่ควบคุมการทำงานเรียกว่า active objects ซึ่งแสดงด้วยเส้นขอบที่หนา</p>	<div style="border: 3px double black; padding: 5px;"> <p><u>Object name</u> : <u>Class</u></p> <p><i>Active object</i></p> </div>
<p>4. Multiplicity แสดงถึงหลาย ๆ ออบเจกต์ด้วยสัญลักษณ์หนึ่งเดียวถ้าแอตทริบิวต์ของ individual ออบเจกต์ ไม่สำคัญ</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p><u>Object name</u> : <u>Class</u></p> </div> <p><i>Multiple objects</i></p>
<p>5. Links คือ instances ของ associations สามารถวาดลิงค์โดยการใช้เส้นในคลาสไคอะแกรม</p>	 <p style="text-align: center;"><i>Links</i></p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

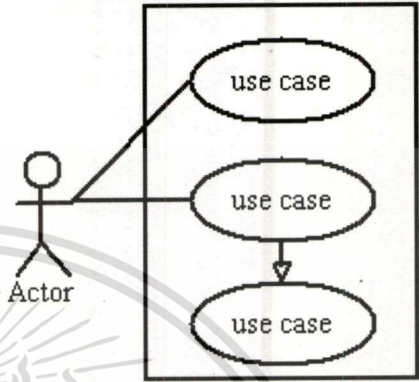

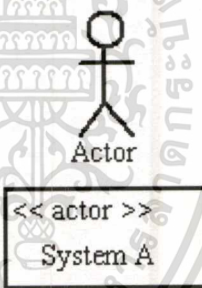
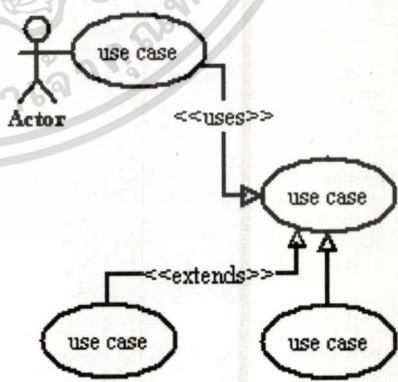
5. Self-linked ออกไปเจดต์ที่เป็น self-linked ตัว
 อย่างถ้าผู้ช่วยฝ่ายบริหาร มีหน้าที่เป็นผู้ช่วย
 ฝ่ายการตลาดด้วยและ 2 ตำแหน่งถูกเชื่อมต่
 ระหว่าง 2 คลาสจะเป็น self-linked

Object name : Class



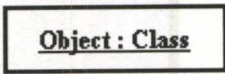
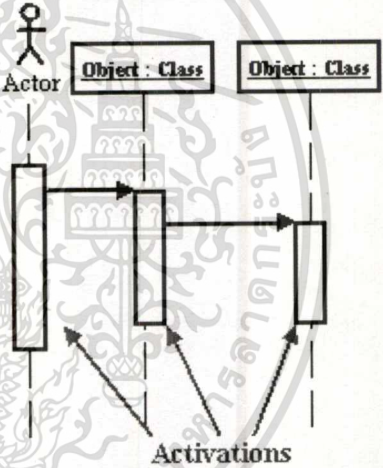
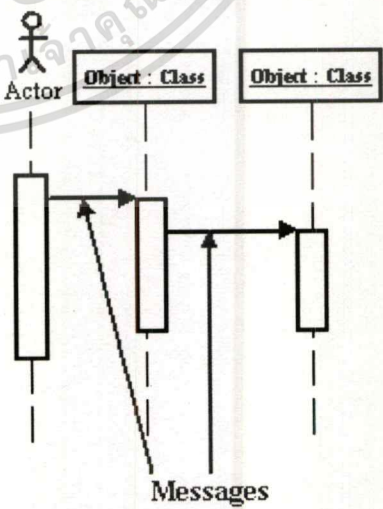
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



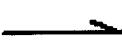


3. Use Case Diagram

สัญลักษณ์ที่ใช้ในยูสเคสไดอะแกรม	รูปลักษณะ
<p>1. System วาดขอบเขตของระบบ โดยการใช้สี่เหลี่ยมผืนผ้าที่บรรจุ use cases และวาง Actors ไว้ข้างนอกขอบเขตของระบบ</p>	
<p>2. Use Case วาด use cases โดยใช้วงรี ชื่ออยู่ในวงรีเป็นคำกริยาที่แทนฟังก์ชันของระบบ</p>	
<p>3. Actors คือยูสเซอร์ของระบบ เมื่อระบบหนึ่งคือ Actor ของอีกระบบหนึ่ง ชื่อ Actor System กับ Actor Stereotype</p>	
<p>4. Relationships แสดงความสัมพันธ์ระหว่าง actor และ use case ด้วยเส้นธรรมดา สำหรับความสัมพันธ์ระหว่าง use cases โดยใช้ชื่อ "uses" หรือ "extends" ความสัมพันธ์ "uses" แสดงให้เห็นว่า use case จำเป็นต้องมีการสั่งให้ทำงาน ความสัมพันธ์ "extends" แสดงให้เห็นว่ามีทางเลือกภายใต้ use case</p>	

4. Sequence Diagram

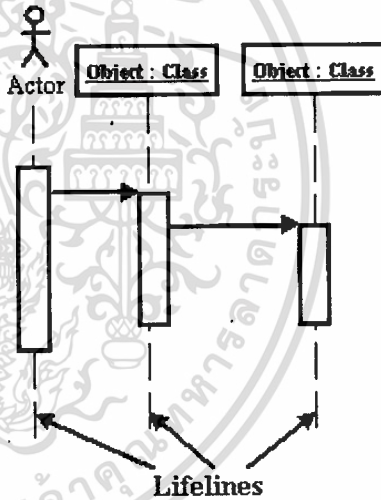
ซีเควนโคออร์ดิเนชันอธิบายถึงการโต้ตอบระหว่างคลาสในไอเท็มของการแลกเปลี่ยนเมสเสจตามช่วงเวลา รวมเอาออบเจกต์และการติดต่อสื่อสารระหว่างออบเจกต์เหล่านี้ ออบเจกต์ที่ถูกวาดบนแกนนอนของโคออร์ดิเนชันขณะที่เวลาอยู่แนวแกนตั้ง

สัญลักษณ์ที่ใช้ในยูสเคสโคออร์ดิเนชัน	รูปลักษณะ
<p>1. Class roles อธิบายถึงวิธีการที่ออบเจกต์จะกระทำในเนื้อหา ใช้ สัญลักษณ์ UML object เพื่อแสดง class roles แต่ไม่มีรายการออบเจกต์แอตทริบิวต์</p>	
<p>2. Activation of object ซึ่ง Activation boxes เสนอเวลาที่ออบเจกต์ที่ต้องการทำงานให้สมบูรณ์</p>	
<p>3. Messages คือลูกศรที่แสดงการติดต่อสื่อสารระหว่างออบเจกต์ ใช้ลูกศรแบบครึ่งแฉกแสดง asynchronous เมสเสจต่างๆ ถูกส่งจากออบเจกต์หนึ่งที่ไม่ต้องคอยการตอบสนองจากผู้รับก่อนทำงานต่อ</p>	

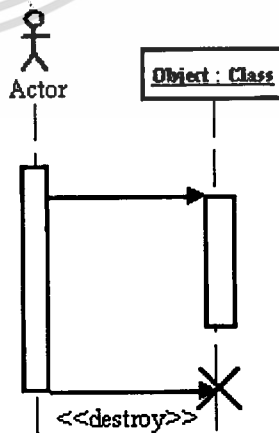
Arrow	Message type
	Simple
	Synchronous
	Asynchronous
	Balking
	Time out

Various message types for Sequence and Collaboration diagrams

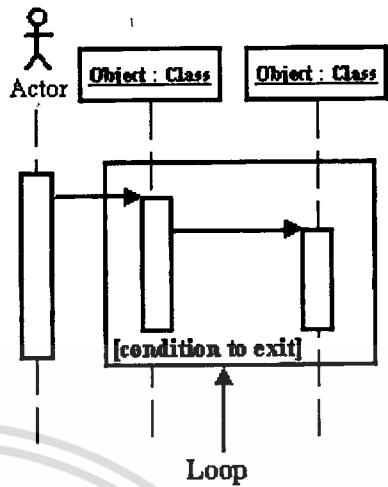
4. Lifelines คือเส้นประแนวตั้งที่แสดงถึงการมีอยู่ของออปเจกต์



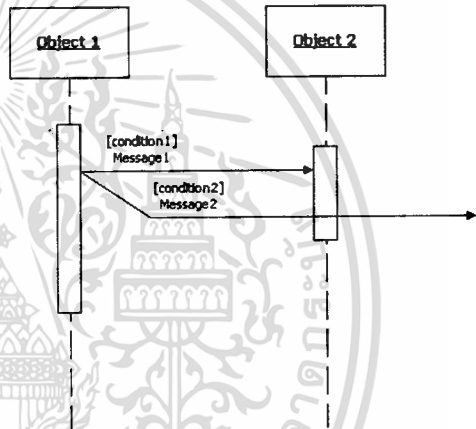
5. Destroying Objects ออปเจกต์สามารถถูกเทอร์มินेटก่อนโดยการใช้ชื่อ << destroy >> เพื่อชี้ไปยัง X



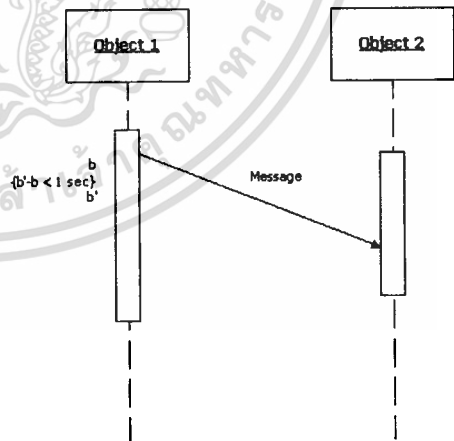
6. Loops การทำซ้ำหรือลูปภายในซีควนไดอะแกรมที่วาดเป็นสี่เหลี่ยมผืนผ้า โดยวางเงื่อนไขสำหรับลูปที่มีอยู่ด้านล่างซ้ายในเครื่องหมายก้ามปู []

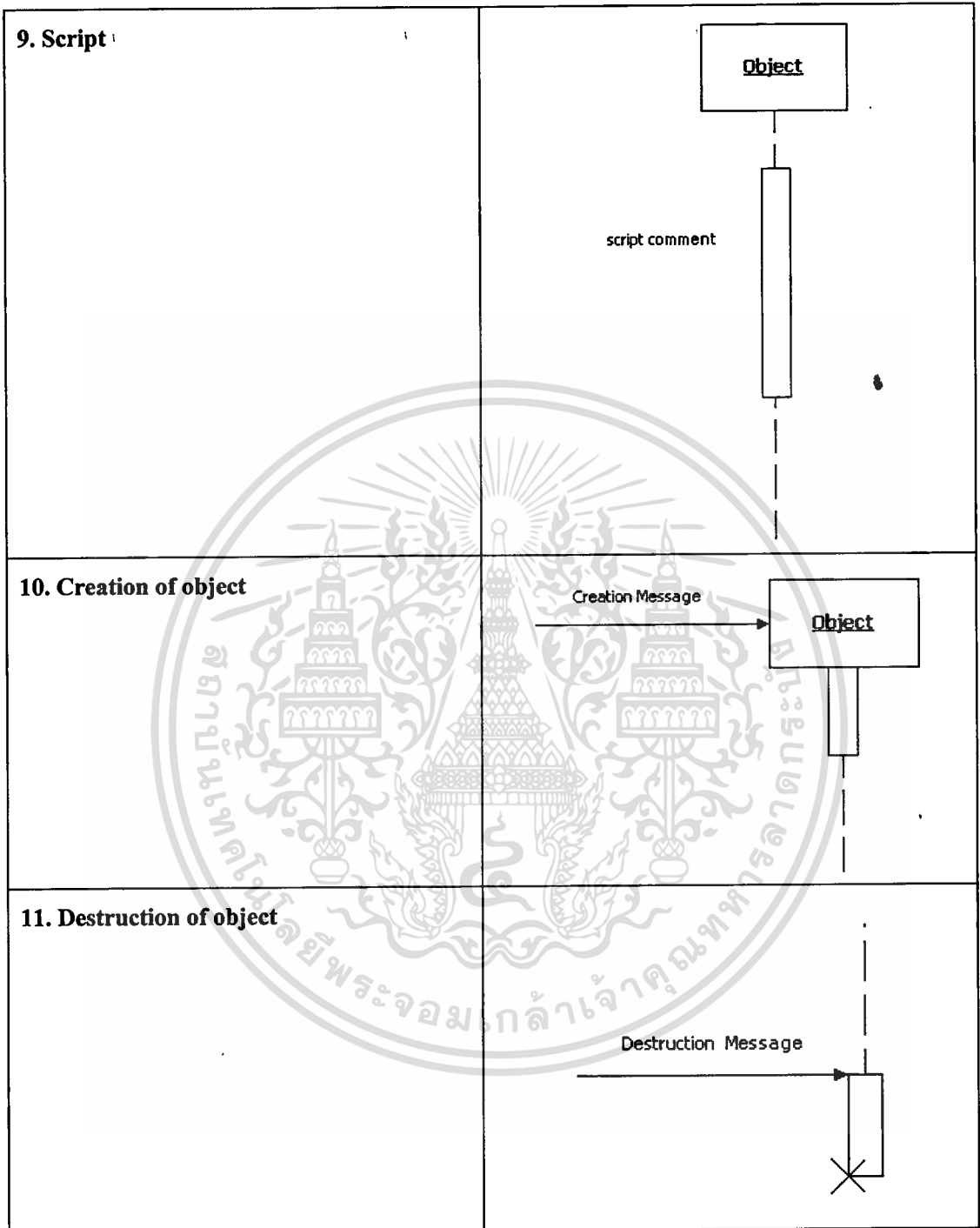


7. Condition

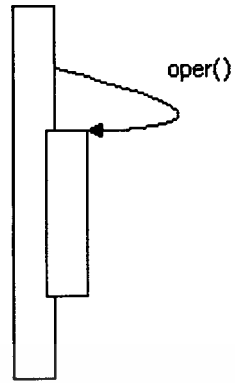


8. Message with transmission time



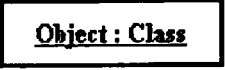
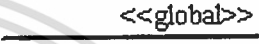
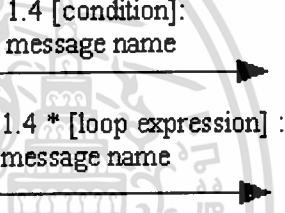
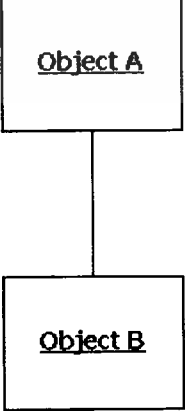
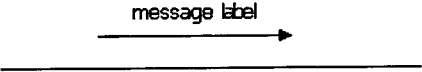


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. Recursion

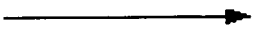
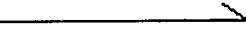
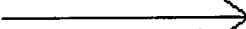
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Collaboration Diagram

สัญลักษณ์ที่ใช้ในคอลลาบอเรนไดอะแกรม	รูปลักษณะ
<p>1. Class roles อธิบายถึงวิธีการที่ออปเจกต์จะกระทำในเนื้อหา ใช้ สัญลักษณ์ UML object เพื่อแสดง class roles แต่ไม่มีรายการออปเจกต์แอตทริบิวต์</p>	
<p>2. Association roles จะอธิบายความสัมพันธ์ของแต่ละส่วน สามารถวาดรูปได้อย่างง่าย ๆ คือการใช้เส้นตรงมีชื่อกำกับ</p>	
<p>3. Messages ไม่เหมือนซีเควนไดอะแกรม, คอลลาบอเรนไดอะแกรมจะไม่มีวิธีที่ชัดเจนว่าแสดงถึงเวลาและหมายเลขของเมสเสจที่เป็นลำดับของการเอ็กซีคิวต์ ลำดับหมายเลขสามารถจะใช้เป็นระบบฐานสิบได้ ตัวอย่างเมสเสจที่ซ่อนอยู่ภายใต้เมสเสจแรกจะเป็นชื่อ 1.1, 1.2, 1.3 ไปเรื่อย ๆ เงื่อนไขของเมสเสจสเปคติจะวางอยู่ในเครื่องหมายก้ามปูตามด้วยลำดับของหมายเลข ใช้ * หลังลำดับของหมายเลขเป็นแสดงว่าเป็นมีการวนซ้ำ</p>	
<p>4. Link (between objects)</p>	
<p>5. Communication between linked objects</p>	


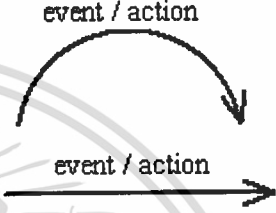


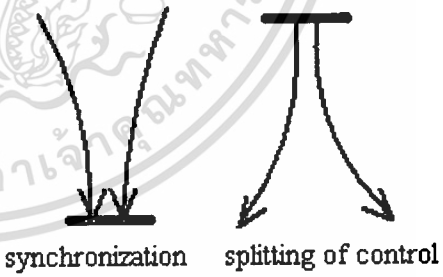
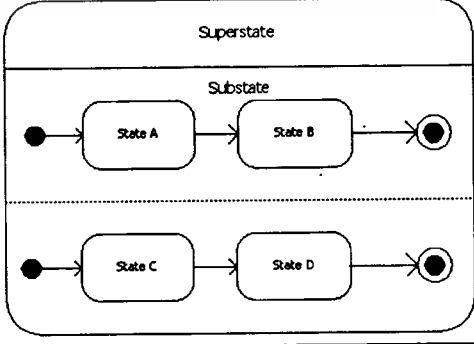
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

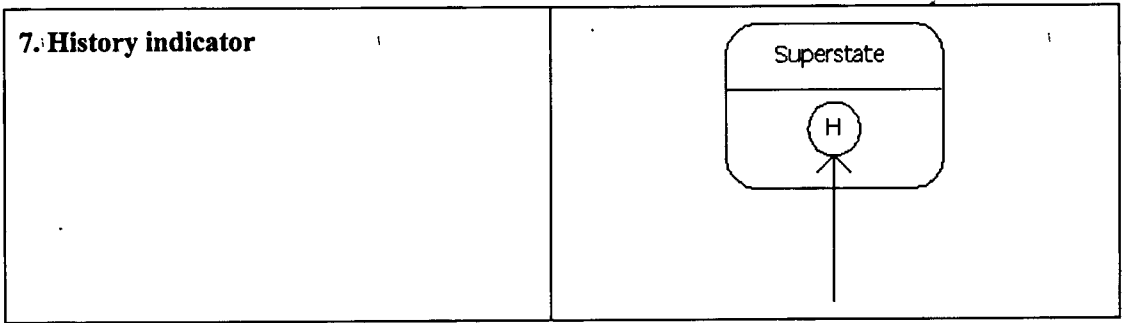
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Synchronous message	
7. Asynchronous message	
8. Simple message	
9. Creation and destruction of object	<div style="border: 1px solid black; padding: 5px; text-align: center;"> Object {new} or Object {destroyed} </div>
10. Active class or object	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <<ActiveClass>> ActiveClass Name </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> activeObject ActiveClass </div> </div>
11. Pattern (Parameterized collaboration)	<div style="border: 1px dashed black; border-radius: 50%; padding: 10px; text-align: center; width: fit-content; margin: auto;"> Pattern Name </div>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


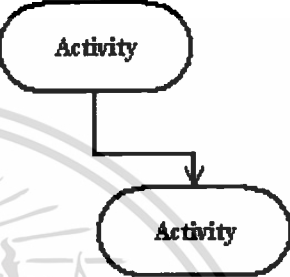
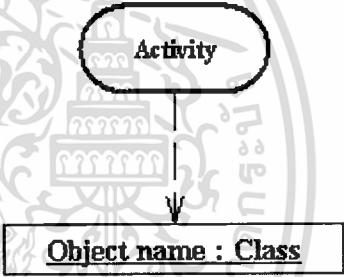


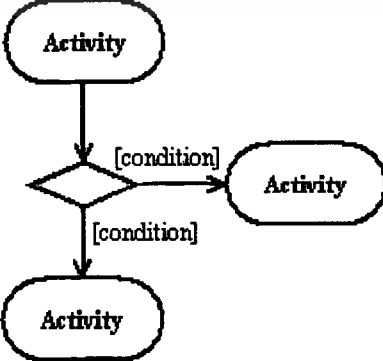
6. Statechart Diagram

สัญลักษณ์ที่ใช้ในสเตทชาร์ตไคอะแกรม	รูปลักษณะ
<p>1. State สเตทจะนำเสนอสถานะการมีชีวิตของออปเจกต์ ทำให้ง่ายจะแสดงสถานะวาดโดยสี่เหลี่ยมผืนผ้าขอบมน</p>	
<p>2. Transition เป็นลูกศรบอกเส้นทางระหว่างสเตทที่ต่างกันของออปเจกต์ ป้ายชื่อการเปลี่ยนแปลงจะบอกด้วยเหตุการณ์ที่จะกระตุ้นให้สเตทนั้นทำงาน</p>	
<p>3. Initial State เป็นรูปร่างกลมระบายสีเต็มพื้นตามด้วยลูกศรจะเป็นสเตทเริ่มต้นของออปเจกต์</p>	
<p>4. Final State ลูกศรชี้ไปยังวงกลมซ้อนกันอยู่โดยวงกลมด้านในจะระบายสีเต็มเป็นจุดสุดท้ายของออปเจกต์</p>	
<p>5. Synchronization and Splitting of Control เป็นรูปบาร์สั้น ๆ ด้วย 2 เส้นการเปลี่ยนแปลงเข้ามาเป็น synchronization of control และเป็นรูปบาร์สั้น ๆ มี 2 เส้นการเปลี่ยนแปลงออกมาเพื่อสร้างอีกหลายสเตท</p>	
<p>6. Substates</p>	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

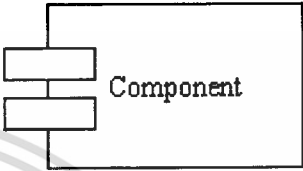

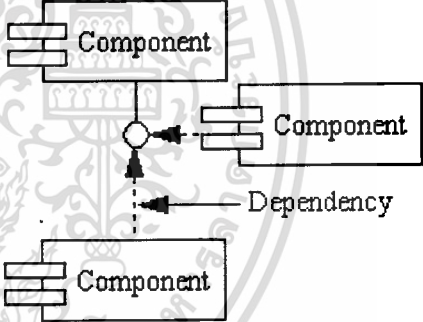
7. Activity Diagram

สัญลักษณ์ที่ใช้ในแอคทิวิตีไดอะแกรม	รูปลักษณะ
1. Action states นำเสนอการทำงานของออปเจกต์ โดยการวาดรูปสี่เหลี่ยมด้วยมุมโค้ง	
2. Action Flow ลูกศรแสดงการไหลความสัมพันธ์ระหว่างการทำงานของสเตท	
3. Object Flow อ่างถึงการสร้างและแก้ไขออปเจกต์โดยกิจกรรม ลูกศร object flow จากกิจกรรมไปยังออปเจกต์มีความหมายว่าสร้างหรือทำงานกับออปเจกต์นั้น	
4. Initial State รูปวงกลมระบายสีพื้นตามด้วยลูกศรเป็นจุดเริ่มต้น	
5. Final State ลูกศรชี้ไปยังวงกลมซ้อนกัน 2 วงวงในระบายสีพื้นเป็นสเตทการทำงานสุดท้าย	
6. Branching รูปข้าวหลามตัดแสดงถึงการตัดสินใจด้วยเส้นทางที่มีทางเลือกโดยจะมีป้ายชื่อบอกเงื่อนไข	

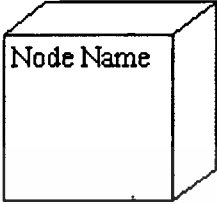
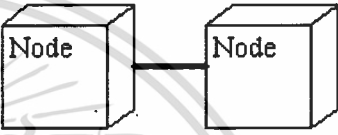
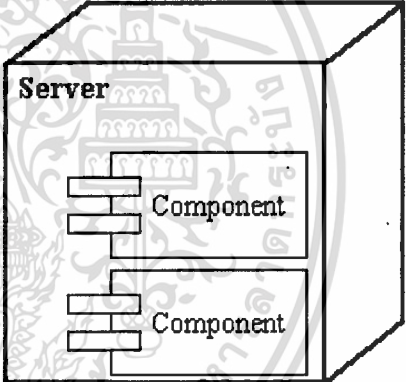

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<p>7. Synchronization เป็นรูปบาร์ช่วยแสดงการขนานของการเปลี่ยนแปลง Synchronization เรียกว่า forking และ joining</p>	
<p>8. Swimlanes กลุ่มของ Swimlanes มีความสัมพันธ์กับกิจกรรมในหนึ่งคอลัมน์</p>	
<p>9. Object as input or Output</p>	
<p>10. Signal sending</p>	
<p>11. Signal receiving</p>	

8. Component Diagram

สัญลักษณ์ที่ใช้ในคอมโพเนนต์ไดอะแกรม	รูปลักษณะ
<p>1. Component คอมโพเนนต์คือบล็อกการสร้างที่เป็นกายภาพของระบบ จะแสดงด้วยรูปสี่เหลี่ยมและมีแท็บ</p>	
<p>2. Interface จะเป็นการอธิบายว่ากลุ่มของการทำงานโดยจะใช้หรือสร้างจากคอมโพเนนต์</p>	
<p>3. Dependencies วัด dependencies ระหว่างคอมโพเนนต์โดยการวาดลูกศรเส้นประ</p>	

9. Deployment Diagram

ตัวอย่างที่ใช้ในศัพทมูลวิทยา	รูปลักษณะ
<p>1. Node โหนดคือทรัพยากรทางกายภาพที่เป็นคอมพิวเตอร์ที่ได้สำหรับเอ็กเซคิวต์</p>	
<p>2. Association จะอ้างถึงการเชื่อมต่อทางกายภาพระหว่างโหนด เช่น อินเทอร์เน็ต</p>	
<p>3. Component and Nodes วางคอมโพเนนต์ไว้ในโหนดที่เหมาะสมกับคอมโพเนนต์นั้น ๆ</p>	
<p>4. Dependency (from node to component or between components/objects)</p>	

ประวัติผู้เขียน

ชื่อ	นายนคร ไชยแสนชมภู
วันเดือนปีเกิด	06 กุมภาพันธ์ 2517
สถานที่เกิด	อำเภอเมือง จังหวัดพะเยา
ประวัติการศึกษา	วุฒิการศึกษาระดับปริญญาตรี วท.บ. (วิทยาการคอมพิวเตอร์ประยุกต์) สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ปี พ.ศ. 2538
ประวัติการทำงาน	ตำแหน่งนักคอมพิวเตอร์ 4 ฝ่ายบริหารและแผนงานด้านการใช้ไฟฟ้า การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย

