

โปรแกรมประยุกต์การบริการแบบกระจายด้วยเทคโนโลยี JINI
DISTRIBUTED SERVICE APPLICATION WITH JINI TECHNOLOGY



T 0 4 6 4 1 1



โดย

นาย กฤษดา

มานะสิทธิชัย

นาย ขวัญเมือง

ต้นประดิษฐ์

2/11/2546
กฤษดา
มานะสิทธิชัย

เลขหมู่.....
เลขทะเบียน..... 46411
วัน, เดือน, ปี..... 1 เม.ย. 2546

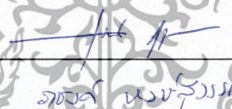
.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	โปรแกรมประยุกต์การบริการแบบกระจายด้วยเทคโนโลยี JINI
TITLE	Distributed Service Application with JINI Technology
นักศึกษา	นาย กฤษดา มานะสิทธิชัย รหัสประจำตัว 41014014 นาย ขวัญเมือง ตันประดิษฐ์ รหัสประจำตัว 41014051
อาจารย์ผู้ควบคุมปริญญานิพนธ์	อาจารย์ นภพินท์ อนันตรศิริชัย อาจารย์ ภูซังค์ หงษ์สุวรรณ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2544

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตร
วิศวกรรมศาสตรบัณฑิตคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

()
อาจารย์ผู้ควบคุมปริญญานิพนธ์

(_____)
อาจารย์ผู้ควบคุมปริญญานิพนธ์

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	โปรแกรมประยุกต์การบริการแบบกระจายด้วยเทคโนโลยี JINI
นักศึกษา	นายกฤษดา มานะสิทธิชัย รหัสประจำตัว 41014014 นายขวัญเมือง ดันประดิษฐ์ รหัสประจำตัว 41014051
อาจารย์ผู้ควบคุมปริญญานิพนธ์	อาจารย์ นภพินท์ อนันตรศิริชัย อาจารย์ ภูซังค์ หงษ์สุวรรณ
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

ในปัจจุบันระบบเครือข่ายได้ถูกพัฒนาไปอย่างมาก และมีความซับซ้อนมากขึ้น นี่เป็นเรื่องที่ยากเกินไปสำหรับผู้ทั่วไปในการใช้งานและดูแลรักษา ความต้องการของผู้ใช้ทั่วไปนั้นต้องการเข้าสู่เครือข่ายโดยง่าย และเป็นอัตโนมัติมากขึ้น

JINI คือชื่อของระบบการประมวลผลแบบกระจายแบบหนึ่งซึ่งแตกต่างจากระบบเครือข่ายทั่วไปตรงที่ JINI นั้นมีความสามารถที่เรียกว่า “เน็ตเวิร์กปลั๊กเอนด์เพลย์” ซึ่งความหมายก็คือ อุปกรณ์ต่าง ๆ ภายในเครือข่ายนั้นมีความสามารถในการถอดหรือเชื่อมต่อเข้ากับระบบแล้วสามารถทำงานได้ทันที โดยที่ไม่จำเป็นต้องมีการปรับแต่งใด ๆ กับระบบ โดยอุปกรณ์ต่าง ๆ ไม่ว่าจะเป็นฮาร์ดแวร์ หรือซอฟต์แวร์จะนำเสนอตนเองในรูปของการบริการบนเครือข่าย

ปริญญานิพนธ์ฉบับนี้ได้ทำการศึกษาถึงเทคโนโลยีของ JINI และนำความรู้ที่ได้มาประยุกต์ใช้โดยการพัฒนาระบบงานขึ้นมาเป็นระบบตัวอย่าง เพื่อแสดงให้เห็นถึงความสามารถของเทคโนโลยี JINI ระบบที่พัฒนามานั้นเป็นระบบเครือข่ายที่ได้ใช้เทคโนโลยี JINI มาช่วยในการติดต่อสื่อสาร และควบคุมการทำงานของโปรแกรมนำเสนอ โดยจำลองให้เครื่องฉายภาพเป็นอุปกรณ์อิเล็กทรอนิกส์ที่มาเชื่อมต่อกับระบบ JINI เครื่องฉายภาพเหล่านั้นจะถูกค้นหา และใช้งานได้อย่างอัตโนมัติ และผู้ใช้สามารถที่จะควบคุมการทำงานของโปรแกรม เช่น การเริ่มนำเสนอ การหยุดนำเสนอ เป็นต้น ผ่านเครือข่ายขณะนั้นได้อย่างง่ายดายโดยใช้เทคโนโลยี JINI

THESIS TITLE Distributed Service Application with JINI Technology

STUDENT Mr. Krisada Manasittichai No. 41014014
 Mr. Kwanmuang Tanpradit No. 41014051

ADVISOR Ms. Noppin Anantarasirichai
 Mr. Puchong Hongsuwan

COURSE Bachelor of Information Engineering

DEPARTMENT Information Engineering

YEAR 2001

ABSTRACT

Nowadays Network Technology grows up rapidly and makes itself more complex. By facing this problem, general users feel hard to deal with the network system or to maintain it. They need the system that is easy to use or automatic configuration system.

JINI Technology is a new Distributed Processing System which differs from other system that it has the ability called "Network Plug and Play", this means that the devices on the network can be added or removed without the configuration from the users and be ready immediately when it is plugged in. Those devices, both hardware and software are introduced as JINI Services.

This thesis discusses about JINI Technology for applying it to create a system which shows the efficiency of JINI Technology by demonstrating a presentation system. The projector will be represented as the service for users to discover automatically without configuration when it has plugged in and immediately ready for projecting the presentation. User can easily control the presentation such as playing or stopping through JINI Technology.

กิตติกรรมประกาศ

ความสำเร็จของปริยฐานิพนธ์ฉบับนี้จะไม่อาจเกิดขึ้นได้ หากขาดการเอาใจใส่ ดูแล ให้คำปรึกษา แนะนำทาง และความช่วยเหลือจากอาจารย์ที่ปรึกษาปริยฐานิพนธ์ผู้ซึ่งเล็งเห็นถึงความสำคัญของปริยฐานิพนธ์ อาจารย์ นภพินท์ อนันตรศิริชัย และ อาจารย์ ภูษงค์ หงษ์สุวรรณ ต้องขอขอบพระคุณเป็นอย่างยิ่ง

ขอขอบพระคุณบิดา มารดา ที่ทำให้ข้าพเจ้าได้มีวันแห่งความสำเร็จนี้ ซึ่งท่านได้ทำการทำนุ ดูแลอย่างดี และให้กำลังใจมาตลอดทำให้ปริยฐานิพนธ์นี้สำเร็จลุล่วงได้ด้วยกำลังใจอันเต็มเปี่ยม

ขอขอบคุณองค์การสื่อสารแห่งประเทศไทยที่เอื้อเฟื้อข้อมูลในการทำปริยฐานิพนธ์ในช่วงแรก ถึงแม้จะไม่ได้นำมาใช้ในการทำปริยฐานิพนธ์ต่อ เนื่องจากเหตุขัดข้องบางประการ

สุดท้ายนี้ ขอขอบคุณผู้ร่วมงานของข้าพเจ้า และเพื่อน ๆ ภาควิชากรรมสารสนเทศทุกคน สำหรับคำปรึกษาต่าง ๆ และร่วมกันฝ่าฟันอุปสรรคกันมาตั้งแต่เริ่มปริยฐานิพนธ์ จนถึงวันแห่งความสำเร็จนี้ ต้องขอขอบคุณอย่างยิ่ง

กฤษดา มานะสิทธิชัย
ขวัญเมือง ตันประดิษฐ์
ผู้จัดทำ

สารบัญ

บทที่	เรื่อง	หน้า
	ปกใน	ก
	หน้าอนุมัติ	ข
	บทคัดย่อภาษาไทย	ค
	บทคัดย่อภาษาอังกฤษ	ง
	กิตติกรรมประกาศ	จ
	สารบัญ	ฉ
	สารบัญรูป	ซ
1	บทนำ	
	1.1 ที่มาและความสำคัญ	1
	1.2 แนวความคิดของเทคโนโลยี JINI	1
	1.3 วัตถุประสงค์ของโครงการ	2
	1.4 ขอบเขตของโครงการ	2
	1.5 ประโยชน์ของโครงการ	3
2	เทคโนโลยี JINI	
	2.1 ภาพรวมของเทคโนโลยี JINI	4
	2.2 องค์ประกอบของเทคโนโลยี JINI	5
	2.3 เซอร์วิสของ JINI	6
	2.3.1 การลงทะเบียนเซอร์วิส	6
	2.3.2 เซอร์วิสพรอกซี	7
	2.4 ไคลเอ็นท์ของ JINI	8
3	ทฤษฎีที่ใช้ในโครงการ	
	3.1 การเขียนโปรแกรมแบบอับเจกต์ (Object Oriented Programming: OOP)	10
	3.1.1 คุณสมบัติของการเขียนโปรแกรมแบบอับเจกต์	10
	3.1.2 คลาส (Class) และ การส่งข้อความ (Message)	11
	3.2 ภาษาจาวา	11

บทที่	เรื่อง	หน้า
3.3	Java Remote Method Invocation : Java RMI	12
3.4	Java Swing	14
3.5	UML (Unified Modeling Language)	15
3.5.1	Class Diagram	15
3.5.2	Use Case Diagram	15
3.5.3	Activity Diagram	16
4	การออกแบบโปรแกรม	
4.1	ความต้องการของผู้ใช้ (User Requirement)	17
4.2	Class Diagram	17
4.3	Use Case Diagram	18
4.4	Activity Diagram	21
5	ผลการทดลอง	
5.1	ผลการทดลองที่ส่วนไคลเอนท์	33
5.2	ผลการทดลองที่ส่วนเซิร์ฟเวอร์	40
5.3	ผลการทดลองที่ส่วนเซิร์ฟเวอร์	41
5.3.1	ผลการรัน HTTP	41
5.3.2	ผลการรัน RMID	41
5.3.3	ผลการรัน reggie	41
6	สรุปและวิจารณ์	
6.1	วิเคราะห์ผลจากการออกแบบและสร้างระบบ	42
6.2	ปัญหาและอุปสรรค	42
6.3	ข้อดีและข้อเสียของระบบ	43
6.4	แนวทางในการพัฒนาต่อไป	44
6.5	สรุป	44
	บรรณานุกรม	45
	ภาคผนวก	
	คู่มือการใช้งาน	46

สารบัญรูป

รูปที่	ชื่อรูป	หน้า
1.1	แสดงตัวอย่างโครงสร้างระบบ	3
2.1	แสดงองค์ประกอบของเครือข่าย JINI	5
2.2	แสดงเซิร์ฟเวอร์ทำการค้นหาลูกอ๊พเซิร์ฟเวอร์	6
2.3	แสดงลูกอ๊พเซิร์ฟเวอร์ตอบสนองด้วยวิธีสทราออบเจกต์	6
2.4	แสดงเซิร์ฟเวอร์ส่งเซิร์ฟเวอร์ออบเจกต์ไปลงทะเบียนที่ลูกอ๊พเซิร์ฟเวอร์	7
2.5	แสดงพรอกซีเซิร์ฟเวอร์	7
2.6	แสดงไคลเอ็นท์ทำการค้นหาลูกอ๊พเซิร์ฟเวอร์	8
2.7	แสดงไคลเอ็นท์พบลูกอ๊พเซิร์ฟเวอร์และได้รับเซิร์ฟเวอร์ออบเจกต์	9
2.8	แสดงไคลเอ็นท์ติดต่อกับเซิร์ฟเวอร์ผ่านเซิร์ฟเวอร์ออบเจกต์โดยตรง	9
3.1	แสดงตัวอย่างการใช้ Java RMI	14
4.1	แสดง Class Diagram เบื้องต้น	17
4.2	แสดง High Level Use Case	18
4.3	แสดง Use Case ระดับที่ 1 ของ Presenter manage presentation data	19
4.4	แสดง Use Case ระดับที่ 1 ของ Client connect to network	19
4.5	แสดง Use Case ระดับที่ 1 ของ Projector connect to server	20
4.6	แสดง Use Case ระดับที่ 1 ของ Client use projector	20
4.7	แสดง Activity Diagram ของการสร้างไฟล์นำเสนอ (New)	21
4.8	แสดง Activity Diagram ของการแก้ไขไฟล์นำเสนอ (Edit)	22
4.9	แสดง Activity Diagram ของการลบไฟล์นำเสนอ (Delete)	23
4.10	แสดง Activity Diagram ของการเพิ่มรูปภาพในไฟล์นำเสนอ (Add)	24
4.11	แสดง Activity Diagram ของการลบไฟล์รูปภาพในไฟล์นำเสนอ (Remove)	25
4.12	แสดง Activity Diagram ของเลื่อนลำดับของรูปภาพขึ้น (Up)	26
4.13	แสดง Activity Diagram ของเลื่อนลำดับของรูปภาพลง (Down)	27
4.14	แสดง Activity Diagram ของการเปิดไฟล์นำเสนอ (Open)	28
4.15	แสดง Activity Diagram ของการเล่นไฟล์นำเสนอ (Play)	29
4.16	แสดง Activity Diagram ของการนำเสนอภาพถัดไป (Next)	30
4.17	แสดง Activity Diagram ของการนำเสนอภาพก่อนหน้า (Previous)	31
4.18	แสดง Activity Diagram ของการหยุดการนำเสนอภาพ (Stop)	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่	ชื่อรูป	หน้า
5.1	แสดงหน้าจอหลักหลังจากการทำการรันโปรแกรม MainWindow.java	33
5.2	แสดงเฟรมภายในที่เกิดขึ้นหลังจากการกดปุ่ม New	34
5.3	แสดงเฟรมที่เกิดขึ้นเมื่อกดปุ่ม Preview	34
5.4	แสดงเฟรมเกิดขึ้นหลังจากการกดปุ่ม Open	35
5.5	แสดงเฟรมภายในที่เกิดขึ้นหลังจากทำการเลือกไฟล์ที่จะเปิด	35
5.6	แสดงเฟรมภายในที่เกิดขึ้นหลังจากทำการเลือกไฟล์ที่จะแก้ไข	36
5.7	แสดงเฟรมแสดงเซอร์วิสที่สามารถเลือกใช้ได้	37
5.8	แสดงตัวอย่างของ Hastable	38
5.9	แสดงเฟรมภายในที่เกิดขึ้นเมื่อเลือกไฟล์ที่จะเล่น และปุ่ม Play ที่ตอบสนองขึ้นมา	39
5.10	แสดงภาพนำเสนอที่เริ่มแสดงภายในโอเปอร์เรชั่นวิวเมื่อกดปุ่ม Play	39
5.11	แสดงผลการลงทะเบียนของเซอร์วิส	40
5.12	แสดงเฟรมแสดงผลที่ส่วนเซอร์วิส	41
ผ.1	แสดงผลที่เกิดขึ้นเมื่อเซอร์วิสลงทะเบียนกับเซิร์ฟเวอร์	47
ผ.2	แสดงหน้าจอหลักของโปรแกรมนำเสนอ	49
ผ.3	แสดงเฟรมภายในของการ New Presentation	50
ผ.4	แสดงการเลือกสี	50
ผ.5	แสดงรูปแบบการนำเสนอแบบที่ 1	51
ผ.6	แสดงรูปแบบการนำเสนอแบบที่ 2	52
ผ.7	แสดงรูปแบบการนำเสนอแบบที่ 3	53
ผ.8	แสดงรูปแบบการนำเสนอแบบที่ 4	53
ผ.9	แสดงเฟรมรูปตัวอย่างที่เกิดขึ้นเมื่อกดปุ่ม Preview	54
ผ.10	แสดงเฟรมเลือกไฟล์นำเสนอเมื่อกดปุ่ม Open และปุ่ม Edit	55
ผ.11	แสดงเฟรมภายในของการ Open Presentation	55
ผ.12	แสดงเฟรมภายในของการ Edit Presentation	56
ผ.13	แสดงเฟรมรายการเซอร์วิสที่เลือกใช้ได้	57
ผ.14	แสดงเฟรมที่เกิดขึ้นเมื่อเลือกไฟล์นำเสนอแล้ว และแสดงปุ่ม Play ที่ตอบสนองขึ้นมา	57
ผ.15	แสดงโอเปอร์เรชั่นวิวที่แสดงภาพนำเสนอภาพแรกเมื่อกดปุ่ม Play	58
ผ.16	แสดงเฟรมนำเสนอที่เกิดขึ้นที่เซอร์วิส	59
ผ.17	แสดงปุ่ม Next และ ปุ่ม Stop	59
ผ.18	แสดงปุ่มปิดโปรแกรมทั้ง 2 ปุ่ม	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันระบบเครือข่ายกำลังเข้ามามีบทบาทในทุกที่ไม่ว่าจะเป็นที่บ้าน ที่ทำงาน ในสถานศึกษา หรือองค์กรต่าง ๆ นอกจากนี้ยังมีซอฟต์แวร์ อุปกรณ์ และแอปพลิเคชัน ในการจัดการกับเครือข่ายในรูปแบบกระจาย (Distributed Computing) มากขึ้น การลงทุนในเครือข่ายบรอดแบนด์สูงเพื่อการทำงานในระดับโลกก็มีอย่างต่อเนื่อง แต่ระบบเครือข่ายที่เป็นอยู่ในทุกวันนี้มีความซับซ้อนเกินไปสำหรับผู้ใช้ในการที่จะใช้งานและดูแล ผู้ทั่วไปต้องการเครือข่ายที่ใช้งานง่าย และเชื่อถือได้เพื่อการทำงานร่วมกันของคอมพิวเตอร์ในการแลกเปลี่ยนข้อมูลข่าวสาร บริการ ทางเครือข่าย

การประมวลผลแบบกระจาย เป็นเทคโนโลยีที่ช่วยให้โปรแกรมประยุกต์ต่าง ๆ ที่อยู่บนเครือข่ายสามารถทำงานร่วมกันได้สะดวกมากขึ้น งานบริการและโปรแกรมประยุกต์เหล่านั้นถูกมองเป็นออบเจกต์ โดยการที่ออบเจกต์ต่าง ๆ ในโปรแกรมเดียวกันจะทำงานร่วมกันหรือมีการติดต่อกันระหว่างออบเจกต์ด้วยกันนั้นไม่ได้มีความยุ่งยากซับซ้อนมากนักเพราะทำงานอยู่ในสภาพแวดล้อมเดียวกัน แต่ในระบบออบเจกต์แบบกระจาย (Distributed Object) นั้นออบเจกต์จะกระจายอยู่ทั่วไปในระบบเครือข่ายที่หลากหลายรูปแบบ สิ่งที่ยากลำบากคือการที่จะทำให้โปรแกรมประยุกต์ทั้งหลายนั้นสามารถติดต่อสื่อสารกันได้โดยไม่จำเป็นต้องทำงานอยู่ในสถานะแวดล้อมเดียวกันบนเครื่องเดียวกัน หรือบนระบบปฏิบัติการเดียวกัน

เทคโนโลยีจาวาได้มาทำให้การประมวลผลแบบกระจายนั้นเป็นเรื่องที่ง่ายขึ้น เนื่องจากตัวภาษาจาวาเองนั้นก็สามารถทำงานได้โดยไม่ต้องขึ้นอยู่กับแพลตฟอร์มอยู่แล้ว ซึ่งตรงกับความต้องการของการประมวลผลแบบกระจายโดยทางบริษัท Sun Microsystems ซึ่งเป็นผู้พัฒนาภาษาจาวา ก็ได้จัดสร้างเทคโนโลยีใหม่ขึ้นมาซึ่งมีความสามารถในการทำงานในลักษณะของการประมวลผลแบบกระจายไว้ในชื่อว่า “เทคโนโลยี JINI”

1.2 แนวความคิดของเทคโนโลยี JINI

เป็นเวลาหลายสิบปีมาแล้วที่ผู้นิยมในเทคโนโลยีพยายามสร้างแนวความคิด “บ้านอัจฉริยะ” ให้เป็นจริงขึ้นมา เหมือนกับที่เราเป็นในภาพยนตร์แนววิทยาศาสตร์ ที่อุปกรณ์เครื่องใช้ไฟฟ้าต่าง ๆ ล้วนแต่รู้หน้าที่ของมันเมื่อได้รับคำสั่ง โทรศัพท์จะติดต่อไปยังเลขหมายที่ต้องการได้เอง มีการค้นหาความผิดปกติในรถยนต์โดยอัตโนมัติ สามารถแสดงรายการอาหารที่ถูกเลือกพร้อมรายงานว่ายังขาดส่วนผสมใดบ้าง และสั่งซื้อไปยังร้านขายของชำในทันที แต่ที่จะเป็นมากกว่านั้นก็คือ เครื่องใช้ไฟฟ้าแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ละตัวสามารถหยิบยืมประสิทธิภาพของคอมพิวเตอร์มาใช้เมื่อจำเป็น และจะได้ผลตอบแทนในทันที เพราะอาศัยการทำงานของสัญญาณไร้สายจึงเป็นเครือข่ายที่กว้างขวางกว่าในปัจจุบันมากนัก และที่สำคัญคือจะไม่มีควมน่ารำคาญเหมือนดังเช่นอุปกรณ์ต่อเชื่อมคอมพิวเตอร์เดสก์ท็อป (Desktop) ในปัจจุบัน หรือถ้าจะลองจินตนาการดูว่า คุณได้ไปร่วมประชุมที่ออฟฟิศของลูกค้า ซึ่งคุณมีหน้าที่ในการนำเสนอข้อมูลให้กับลูกค้าเมื่อคุณต่อเครื่องโน้ตบุ๊กไปยังเครือข่ายที่มีอยู่ภายในห้อง เครื่องโน้ตบุ๊ก (Note book) ของคุณก็พบกับพรินเตอร์ (Printer), แฟกซ์ (FAX) และเครื่องฉายภาพ (Projector) ที่อยู่ในห้องได้ทันที และสามารถนำมาใช้ได้โดยไม่ต้องกำหนดค่าคอนฟิกูเรชัน (Configuration) ของคอมพิวเตอร์แต่อย่างใด หรือถ้าคุณลืมเอกสารอิเล็กทรอนิกส์ที่สำคัญต่อการประชุมไว้ที่ออฟฟิศของคุณ คุณก็จะสามารถเรียกเอกสารนี้จากเครือข่ายที่ออฟฟิศของคุณผ่านทางอินเทอร์เน็ตได้โดยปลอดภัย และสามารถสั่งพิมพ์ออกมาทางพรินเตอร์ที่อยู่ในห้องประชุมได้ทันที ไม่จำเป็นต้องพึ่งใครเวอร์ (Driver), ค่าคอนฟิกูเรชัน, ไม่มีปัญหาเรื่องความเข้ากันได้ (Compatibility) ทุก ๆ สิ่งจะทำหน้าที่เป็นบริการที่อยู่ในเครือข่าย เป็นเครือข่ายที่เกิดขึ้นได้โดยตัวมันเองจากอุปกรณ์เครือข่ายที่มีอยู่บนล้านชนิดนั่นเอง พลังประมวลผลของคอมพิวเตอร์กลายเป็นบริการหนึ่งที่คอมพิวเตอร์เครื่องอื่นซึ่งอยู่บนเครือข่ายสามารถหยิบยืมสมองหรือโปรเซสเซอร์ (Processor) ของเครื่องอื่นมาใช้ได้ตามหลักการของการประมวลผลแบบ นี้อีกแนวความคิดของเทคโนโลยี JINI

1.3 วัตถุประสงค์ของโครงการ

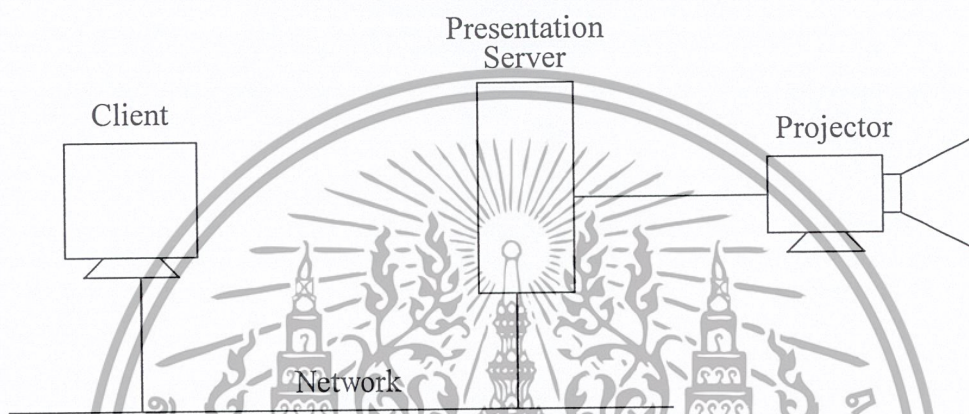
1. เพื่อศึกษาเทคโนโลยี JINI ว่าทำงานได้อย่างไร และสามารถนำไปประยุกต์ใช้กับงานประเภทใดได้บ้าง
2. ศึกษาเทคโนโลยีที่เกี่ยวข้อง เช่น Java RMI, การพัฒนาโปรแกรมประยุกต์ด้วยชุดพัฒนา Java 2 Micro Edition, การสร้างส่วนติดต่อกับผู้ใช้ด้วย Java Swing และการออกแบบซอฟต์แวร์ด้วยหลักการ UML (Unified Modeling Language)
3. นำเทคโนโลยี JINI มาประยุกต์ใช้โดยการสร้างระบบเครือข่ายจำลองของ JINI ขึ้นมาทำการสร้างโปรแกรมประยุกต์การแสดงผลภาพ ผ่านเครื่องฉายภาพ ซึ่งทำหน้าที่เป็นเซิร์ฟเวอร์ของ JINI โดยการสร้างนี้จะเน้นที่การแสดงให้เห็นถึงความง่าย และคุณสมบัติต่าง ๆ ของเทคโนโลยี JINI
4. เพื่อเป็นแนวทางในการพัฒนาระบบที่สามารถนำไปประยุกต์ใช้กับงานจริงได้ต่อไป

1.4 ขอบเขตของโครงการ

จากแนวความคิดของเทคโนโลยี JINI ดังที่กล่าวมาข้างต้น โครงการนี้จึงเป็นการศึกษาที่เกี่ยวข้องกับเทคโนโลยี JINI, หลักการทำงานของเทคโนโลยี JINI, การออกแบบเทคโนโลยี JINI โดยใช้หลักเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ UML, การออกแบบส่วนติดต่อกับผู้ใช้ และการสร้างโปรแกรมซึ่งเป็นโปรแกรมนำเสนอ ผ่านเครือข่าย โดยที่ผู้ใช้บริการจะสามารถควบคุมการนำเสนอภาพได้ โดยผ่านเครื่องคอมพิวเตอร์ที่เป็นไคลเอ็นท์นั่นเอง

ลักษณะของระบบนี้แสดงให้เห็นถึงข้อได้เปรียบของเทคโนโลยี JINI คือ เครื่องฉายภาพที่ทำหน้าที่เป็นเซิร์ฟเวอร์ของระบบนั้น สามารถนำมาเชื่อมต่อกับระบบ หรือถอดออกจากระบบได้ตลอดเวลา โดยที่เครื่องไคลเอ็นท์ที่ทำหน้าที่เป็นผู้ใช้บริการจะสามารถรับทราบได้ถึงการเปลี่ยนแปลงได้



รูปที่ 1.1 แสดงตัวอย่างโครงสร้างระบบ

1.5 ประโยชน์ของโครงการ

1. เข้าใจหลักการของเทคโนโลยี JINI และสามารถนำความรู้ที่ได้ ไปประยุกต์ใช้สร้างโปรแกรมประยุกต์การบริการแบบกระจายได้
2. เข้าใจหลักการเขียนโปรแกรมแบบอสังวัตถุ และการออกแบบซอฟต์แวร์ด้วย UML ซึ่งสามารถนำไปใช้ในการออกแบบซอฟต์แวร์ได้
3. เข้าใจหลักการเขียนโปรแกรมด้วยภาษาจาวา
4. โปรแกรมที่จัดสร้างขึ้นสามารถนำไปใช้ได้จริง เช่น สื่อการสอนต่าง ๆ เพิ่มความสะดวกมากขึ้นแก่ผู้ใช้
5. เป็นแนวทางใช้ในการพัฒนาเพื่อใช้ในการพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

เทคโนโลยี JINI

2.1 ภาพรวมของเทคโนโลยี JINI

JINI คือ ชื่อของระบบออกแบบแจกจ่ายตัวหนึ่งที่มีความสามารถที่เรียกว่า “Network Plug and Play” อุปกรณ์ต่าง ๆ ไม่ว่าจะเป็นซอฟต์แวร์หรือฮาร์ดแวร์สามารถเชื่อมต่อกับระบบแล้วทำการแจ้งให้ระบบทราบถึงการมาของตน ไคลเอนท์ (Client) ที่ต้องการจะเรียกใช้บริการ (Service) ต่าง ๆ บนเครือข่ายสามารถที่จะค้นหาบริการนั้นจากเครือข่ายแล้วทำการเรียกใช้บริการจากผู้ให้บริการนั้นได้ ซึ่งระบบนี้เหมาะกับเครือข่ายที่มีความเปลี่ยนแปลงบ่อย ตัวอย่างสถานการณ์ที่เหมาะสมกับการใช้เครือข่าย JINI คือ

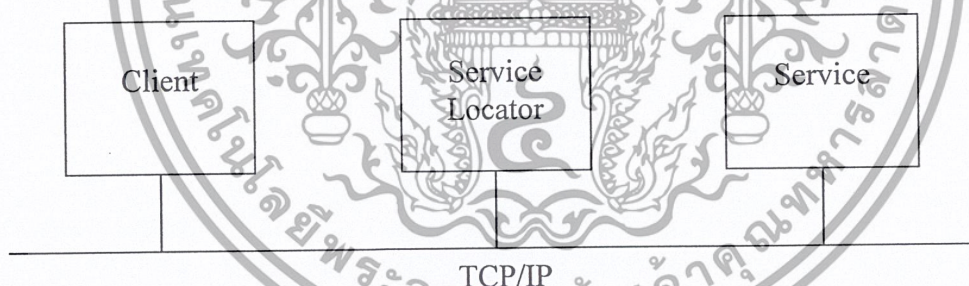
1. มีเครื่องพริเตอร์มาเชื่อมต่อกับเครือข่ายและได้แจ้งให้กับระบบรู้ถึงการเชื่อมต่อของตนและประสิทธิภาพของเครื่องทำได้ ซึ่งไคลเอนท์สามารถเรียกใช้เครื่องพริเตอร์เครื่องนี้ในการพิมพ์งานได้โดยไม่ต้องมีการปรับแต่งอะไรเลย
2. มีกล้องถ่ายภาพแบบดิจิทัลมาเชื่อมต่อกับเครือข่ายแล้วต้องการจะขอใช้บริการการพิมพ์จากเครื่องพริเตอร์สามารถขอใช้บริการได้เลยโดยไม่ต้องมีการติดตั้งไดรเวอร์ของเครื่องพริเตอร์
3. อุปกรณ์ใหม่ ๆ สามารถเพิ่มเติมเข้ามาในระบบได้โดยไม่ต้องมีการหยุดการทำงานของระบบและไม่ต้องมีการกลับมาปรับแต่งอะไรกับระบบเครือข่ายอีก
4. ผู้ให้บริการสามารถแจ้งกับระบบได้ถึงการเปลี่ยนแปลงสถานะบางอย่างของระบบ เช่น พริเตอร์กระดาษหมด โดยที่ไคลเอนท์สามารถตรวจสอบการเปลี่ยนแปลงของสถานะต่าง ๆ ได้

JINI คือระบบหรือกลุ่มของเซอร์วิสและไคลเอนท์ของเซอร์วิสที่มีการติดต่อสื่อสารกันด้วยโพรโตคอล JINI (JINI protocol) ซึ่งปรกติแล้วจะประกอบไปด้วย แอปพลิเคชันที่สร้างขึ้นมาด้วยภาษาจาวา (Java language) มีการติดต่อสื่อสารกันด้วยกระบวนการของ Java RMI แต่ถึงแม้ว่า JINI จะถูกสร้างขึ้นมาด้วยภาษาจาวาล้วน ๆ แต่ว่าไคลเอนท์หรือเซอร์วิสต่าง ๆ ที่มาเชื่อมต่อกับเครือข่ายของ JINI นั้นไม่จำเป็นต้องถูกสร้างขึ้นมาด้วยภาษาจาวาอย่างเดียว อาจจะถูกสร้างขึ้นมาด้วยวิธีการพื้นฐาน (Native method) แล้วห่อหุ้มด้วยจาวาออบเจกต์หรือแม้กระทั่งเป็นออบเจกต์ที่ถูกสร้างขึ้นด้วยภาษาอื่นก็สามารถทำได้ ทั้งหมดต้องการเพียงแค่การสื่อสารกันด้วย JINI โพรโตคอลเท่านั้น

2.2 องค์ประกอบของเทคโนโลยี JINI

เทคโนโลยี JINI เป็นประเภทหนึ่งจากหลาย ๆ สถาปัตยกรรมของระบบออบเจกต์แบบกระจายซึ่งรวมไปถึงระบบที่เป็นมาตรฐานอุตสาหกรรมอย่างเช่น คออร์บา (CORBA) แต่แตกต่างกันที่ JINI นั้นมีพื้นฐานมาจากภาษาจาวาซึ่งสามารถใช้คุณสมบัติต่าง ๆ ของภาษาจาวาได้อย่างครบถ้วน และยังมีเฟรมเวิร์ก (Frame work) ของภาษาจาวาตัวอื่น ๆ ที่ทาง Sun Microsystems ได้พัฒนาออกมาซึ่งมีส่วนที่คล้ายกับ JINI อยู่บ้าง เช่น Enterprise Java Beans ทำให้ JINI สามารถทำงานร่วมกับเฟรมเวิร์กอื่น ๆ ของ Sun Microsystems ได้ แต่จะไม่ขอล่าถึงในที่นี้ว่าทำได้อย่างไร

ในระบบของ JINI มีองค์ประกอบอยู่ 3 ส่วน ที่เป็นส่วนหลักสำคัญ ส่วนแรกก็คือเซิร์ฟเวอร์ ตัวอย่างของเซิร์ฟเวอร์ก็เช่น เครื่องฉายภาพ เครื่องปิ้งขนมปัง หรือแม้แต่เป็น ซอฟต์แวร์โปรแกรมเช่น โปรแกรมแปลงรูปแบบไฟล์ เป็นต้น ส่วนที่สองคือไคลเอ็นท์ซึ่งมาเรียกใช้บริการของเซิร์ฟเวอร์ต่าง ๆ ในระบบ และส่วนสุดท้ายคือ ลูคอัพเซิร์ฟเวอร์ (Lookup Service) ซึ่งทำหน้าที่เป็นตัวกลาง ระหว่างเซิร์ฟเวอร์และไคลเอ็นท์ และยังมีส่วนประกอบที่สำคัญอีกตัวหนึ่ง คือระบบเครือข่ายที่ทำการเชื่อมต่อองค์ประกอบสำคัญทั้งสามส่วนเข้าด้วยกัน และระบบเครือข่ายที่ใช้สำหรับ JINI นั้นปรกติแล้วจะทำงานอยู่บนเครือข่ายด้วย โพรโตคอล TCP/IP เทคโนโลยี JINI นั้นตามข้อกำหนดและหลักการนั้นสามารถทำงานได้โดยไม่ขึ้นกับเครือข่าย โพรโตคอล แต่ในปัจจุบัน เครือข่าย JINI ที่ถูกสร้างขึ้นนั้น ถูกทำขึ้นสำหรับ TCP/IP เท่านั้น ซึ่งเราสามารถสร้างเครือข่าย JINI สำหรับโพรโตคอลอื่น ๆ ก็สามารทำได้



รูปที่ 2.1 แสดงองค์ประกอบของเครือข่าย JINI

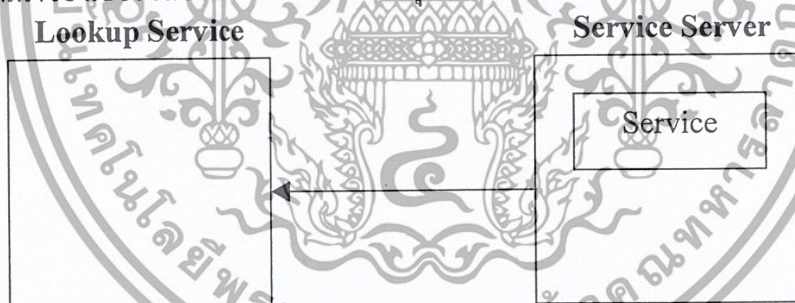
โค้ดของจาวา สามารถเคลื่อนย้ายระหว่างองค์ประกอบทั้ง 3 ตัวได้ด้วยกระบวนการออบเจกต์ซีเรียลไลซิง (Object serializing) และใช้จาวาซ็อกเก็ต (Java socket) ในการส่งและรับออบเจกต์มาจากเครือข่ายซึ่งออบเจกต์ที่อยู่ใน Java Virtual Machine หนึ่งสามารถเรียกใช้วิธีการของออบเจกต์ที่อยู่ในอีก Java Virtual Machine หนึ่งได้โดยกระบวนการของ Java RMI ถึงแม้ว่า JINI จะไม่จำเป็นต้องใช้ Java RMI แต่การเรียกใช้ Java RMI นั้นทำให้การทำงานของ JINI นั้นง่ายขึ้น

2.3 เซอร์วิสของ JINI

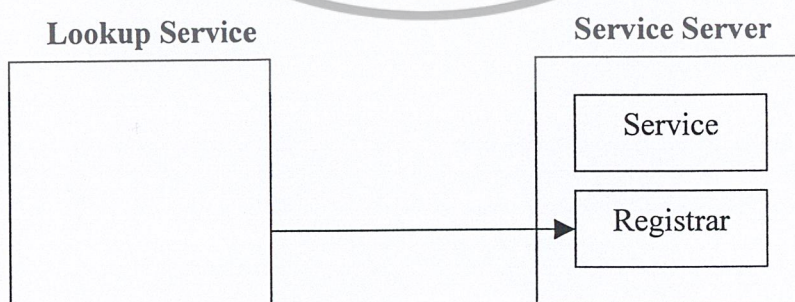
2.3.1 การลงทะเบียนเซอร์วิส

ตัวเซอร์วิสที่แท้จริงของระบบ JINI ก็คือออบเจกต์หรือกลุ่มของออบเจกต์ที่ถูกสร้างขึ้นมาภายในเซิร์ฟเวอร์นั่นเอง ซึ่งเซิร์ฟเวอร์จะทำหน้าที่ที่สำคัญสำหรับเซอร์วิส คือ การลงทะเบียนเซอร์วิสไปยังลูกอ็อปเซอร์วิส หรือเรียกได้อีกอย่างว่าเซอร์วิสโลเคเตอร์ (Service locator) ในกระบวนการนี้ เซิร์ฟเวอร์จะต้องทำการค้นหาลูกอ็อปเซอร์วิสเป็นอันดับแรก ซึ่งกระบวนการนี้สามารถทำได้ 2 วิธีคือ ถ้าหากเรารู้ตำแหน่ง (Address) ที่แน่นอน ของลูกอ็อปเซอร์วิส แล้วเซิร์ฟเวอร์สามารถใช้การทำ ยูนิคาสต์ (Unicast) ของ TCP ในการเชื่อมต่อโดยตรงไปยังลูกอ็อปเซอร์วิส แต่ถ้าหากไม่รู้วาลูกอ็อปเซอร์วิสทำงานอยู่ที่ใดในเครือข่ายเซิร์ฟเวอร์สามารถทำการส่ง UDP และทำการร้องขอแบบมัลติคาสต์ (Multicast request) ไปยังเครือข่าย เมื่อลูกอ็อปเซอร์วิสได้รับรีเควสต์ (Request) ตัวลูกอ็อปเซอร์วิสก็จะทำการส่งออบเจกต์กลับไปยังเซิร์ฟเวอร์ซึ่งออบเจกต์นี้เราเรียกว่า รีจิสทราออบเจกต์ (Registrar object) ที่ทำหน้าที่เป็นพรอกซี (Proxy) ไปยังลูกอ็อปเซอร์วิส การร้องขอสิ่งต่าง ๆ ไปยังลูกอ็อปเซอร์วิส จะทำผ่านออบเจกต์ตัวนี้ ซึ่งกระบวนการที่รีจิสทราออบเจกต์ทำการติดต่อกลับไปยังลูกอ็อปเซอร์วิส นั้นสามารถทำผ่านโพรโตคอลใดก็ได้ แต่วิธีที่ง่ายที่สุดวิธีหนึ่งก็คือการทำงานผ่านจาวาอาร์เอ็มไอ

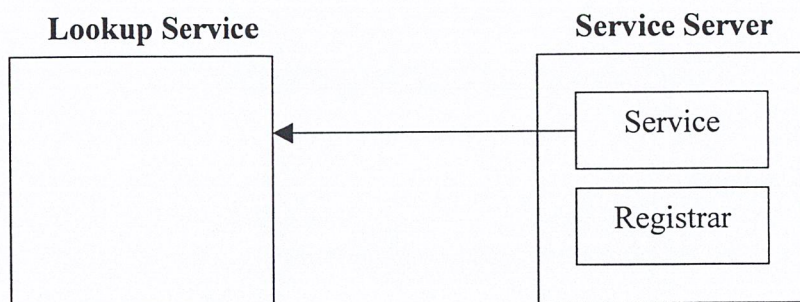
กระบวนการที่เซิร์ฟเวอร์จะทำการลงทะเบียนเซอร์วิสไปยังลูกอ็อปเซอร์วิส นั้น คือ เซิร์ฟเวอร์ จะทำการส่งสำเนาของเซอร์วิสออบเจกต์ไปเก็บไว้ที่ลูกอ็อปเซอร์วิส



รูปที่ 2.2 แสดงเซอร์วิสทำการค้นหาลูกอ็อปเซอร์วิส



รูปที่ 2.3 แสดงลูกอ็อปเซอร์วิสตอบรับด้วยรีจิสทราออบเจกต์



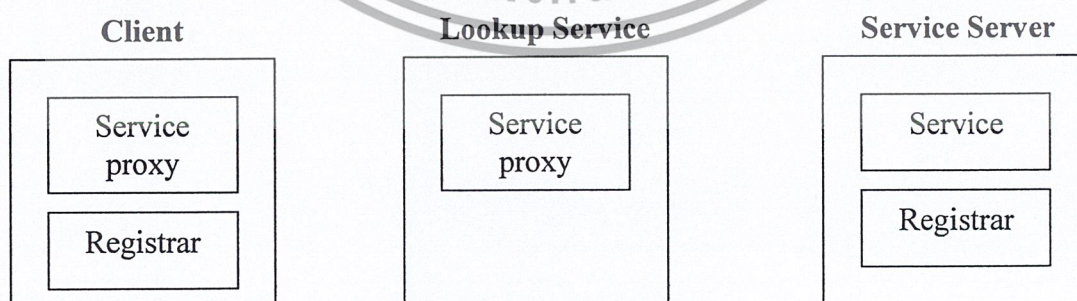
รูปที่ 2.4 แสดงเซอร์วิสส่งเซอร์วิสออบเจกต์ไปลงทะเบียนที่ลुकอัพเซอร์วิส

จากกระบวนการที่แสดงนี้เซอร์วิสออบเจกต์ตัวเดิมก็ยังทำงานอยู่ที่เซิร์ฟเวอร์ แต่เซอร์วิสออบเจกต์ที่เก็บอยู่ที่ลुकอัพเซอร์วิสนั้นเป็นแค่สำเนาของเซอร์วิสออบเจกต์ของเซิร์ฟเวอร์เท่านั้น ซึ่งไคลเอ็นท์ก็จะทำสำเนาของเซอร์วิสออบเจกต์นี้จากลुकอัพเซอร์วิส และทำการเรียกใช้บริการของเซอร์วิสผ่านทางเซอร์วิสออบเจกต์ซึ่งอยู่ที่ Java Virtual Machine ของตัวเอง

2.3.2 เซอร์วิสพรอกซี

ถ้าหากว่าเซอร์วิสนั้นเป็นฮาร์ดแวร์แทนที่จะเป็นซอฟต์แวร์ เช่น พรินเตอร์, เครื่องฉายภาพ, เครื่องปั๊มลมบ่ง โคลเอ็นท์ไม่สามารถคุยกับเซอร์วิสได้โดยตรงเนื่องจากเซอร์วิสนั้นเป็นฮาร์ดแวร์ ในกรณีนี้ตัวเซอร์วิสแทนที่จะส่งสำเนาของตัวเอง จะทำไม่ได้เนื่องจากตัวเซอร์วิสเองเป็นฮาร์ดแวร์ เซอร์วิสจึงส่งพรอกซีออบเจกต์ซึ่งทำหน้าที่ติดต่อกลับมายังเซอร์วิสไปแทน โดยอาจจะใช้ Java RMI ในการส่งการติดต่อ

แต่ก็อาจจะเกิดคำถามมาอีกว่าเนื่องจากพรอกซีออบเจกต์ไม่ใช่ตัวเซอร์วิสออบเจกต์เองพรอกซีออบเจกต์จะรู้ได้อย่างไรว่าตัวเซอร์วิสอยู่ที่ไหนใช้ลुकอัพเซอร์วิสเหมือนกันหรือเปล่า



รูปที่ 2.5 แสดงพรอกซีเซอร์วิส

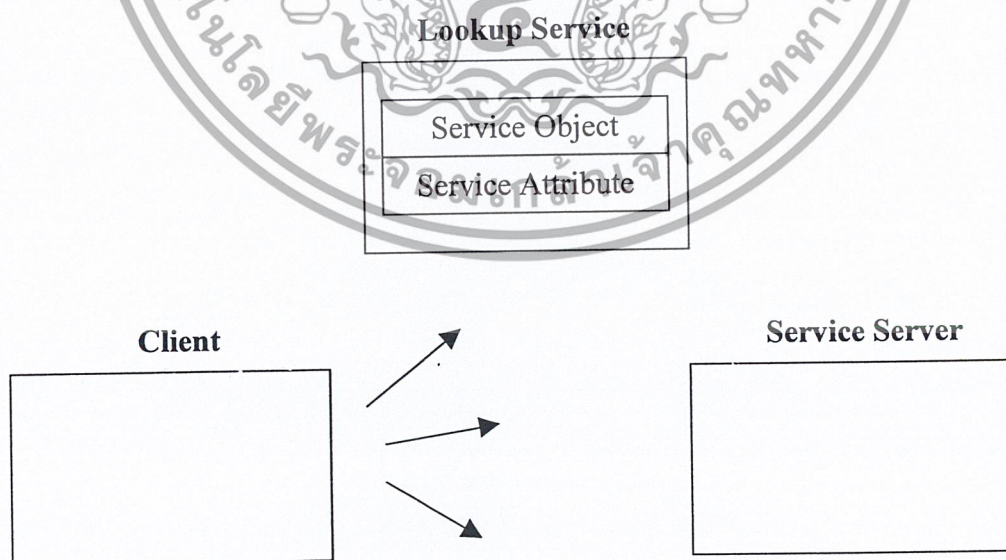
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ได้มีการกำหนดไว้เป็นวิธีที่ตายตัวว่าจะต้องทำอะไร แต่ก็มีหลายวิธีที่จะทำให้พร็อกซีออบเจกต์นั้นสามารถรู้จักกับเซอร์วิสของตัวเอง ตัวอย่างเช่น การใช้กระบวนการของ Java RMI ในการสร้างพร็อกซีออบเจกต์ก่อนที่จะส่งไปลงทะเบียนกับคลุคอัพเซอร์วิสก็ทำการใส่ตัวอ้างอิง (Reference) ของเซอร์วิสออบเจกต์ไปยังพร็อกซีออบเจกต์เพื่อให้พร็อกซีออบเจกต์สามารถมีตัวอ้างอิงกลับมายังเซอร์วิสได้ หรือแม้กระทั่งเซอร์วิสทำการใส่ค่าแอคเครสและพอร์ตของตนเองให้กับพร็อกซีออบเจกต์เพื่อให้พร็อกซีออบเจกต์สามารถสร้างชื่อเรียกกลับมายังเซอร์วิสอีกทีก็สามารถทำได้เช่นกัน

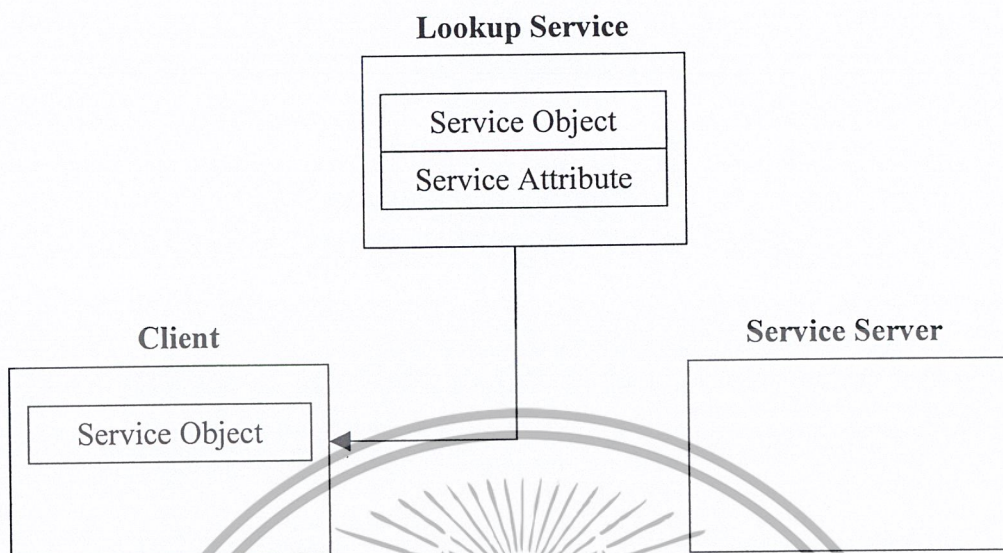
2.4 ไคลเอ็นท์ของ JINI

2.4.1 การค้นหาคลุคอัพเซอร์วิสของไคลเอ็นท์

กระบวนการที่ไคลเอ็นท์ทำการค้นหาคลุคอัพเซอร์วิสนั้นจะใช้กระบวนการเดียวกับเซอร์วิสเซิร์ฟเวอร์ในการค้นหาคลุคอัพเซอร์วิส โดยที่เมื่อไคลเอ็นต์สามารถค้นหาคลุคอัพเซอร์วิสได้แล้ว และต้องการจะเรียกใช้เซอร์วิสจากเครือข่าย JINI ไคลเอ็นต์จะต้องทำการค้นหาเซอร์วิสที่ต้องการจากคลุคอัพเซอร์วิสโดยไคลเอ็นต์ต้องทำการสอบถามเข้าไปที่คลุคอัพเซอร์วิสเพื่อค้นหาเซอร์วิสที่ต้องการ การค้นหาเซอร์วิสนั้น ไคลเอ็นต์จะใช้การสร้างเทมเพลต (Template) ในการค้นหาขึ้นแล้วทำการเรียกวิธีการ lookup () ไปยังคลุคอัพเซอร์วิสผ่านทางวิธีสทราออบเจกต์ที่ได้รับมาโดยส่งเทมเพลตในการค้นหาไปเป็นพารามิเตอร์ และเมื่อคลุคอัพเซอร์วิสค้นหาเซอร์วิสที่ไคลเอ็นต์ต้องการพบก็จะทำการส่งสำเนาของเซอร์วิสออบเจกต์ของเซอร์วิสนั้นกลับไปยังไคลเอ็นต์

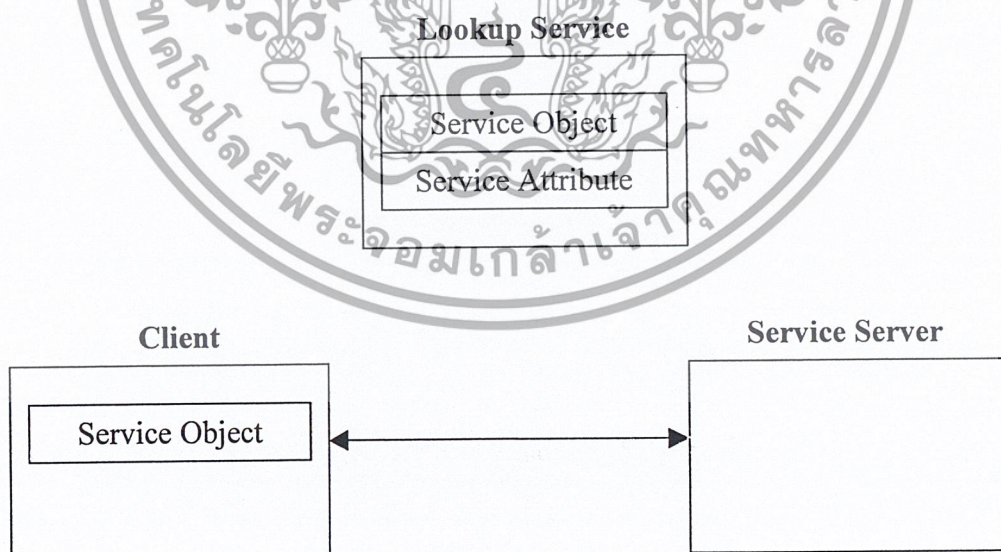


รูปที่ 2.6 แสดงไคลเอ็นท์ทำการค้นหาคลุคอัพเซอร์วิส



รูปที่ 2.7 แสดงไคลเอ็นท์พบลुकอัพเซอร์วิสและได้รับเซอร์วิสออบเจกต์

เมื่อไคลเอ็นท์ได้รับเซอร์วิสออบเจกต์แล้ว การที่ไคลเอ็นท์จะเรียกใช้บริการจากเซอร์วิสนั้น ไคลเอ็นท์สามารถทำได้โดยตรงจากการเรียกใช้ไปยังเซอร์วิสออบเจกต์ โดยที่กระบวนการเรียกใช้ เซอร์วิสนี้ไคลเอ็นท์จะทำงานโดยตรงกับเซอร์วิส โดยที่ไม่จำเป็นต้องเรียกผ่านไปยังลुकอัพเซอร์วิสอีกต่อไป



รูปที่ 2.8 แสดงไคลเอ็นท์ที่ติดต่อกับเซอร์วิสผ่านเซอร์วิสออบเจกต์โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ทฤษฎีที่ใช้ในโครงงาน

3.1 การเขียนโปรแกรมแบบอังกิววัตถุ (Object Oriented Programming: OOP)

การเขียนโปรแกรมแบบอังกิววัตถุ คือ มุมมองในการออกแบบซอฟต์แวร์ที่มององค์ประกอบ (Entity) ต่าง ๆ ในระบบเป็นออบเจกต์ (Object) ซึ่งมีคุณสมบัติและความสามารถเฉพาะตัว ที่สามารถทำหน้าที่ของแต่ละออบเจกต์นั้นได้ เพื่อที่จะทำงานร่วมกันให้บรรลุจุดประสงค์ของระบบที่ตั้งไว้ ส่วนออบเจกต์นั้น คือ การรวบรวมข้อมูลและฟังก์ชันที่ดำเนินการต่อข้อมูลนั้นเข้าไว้ด้วยกันเป็นหน่วยเดียวกัน โดยในการทำงานร่วมกันของออบเจกต์นั้นจะสื่อสารกันโดยส่งข้อมูลจากออบเจกต์หนึ่งไปยังอีกออบเจกต์หนึ่งได้โดยใช้การส่งข้อความ

การเขียนโปรแกรมแบบอังกิววัตถุ ต่างจากการเขียนโปรแกรมแบบโครงสร้าง (Structure) พอสมควรเนื่องจากแบบอังกิววัตถุ ผู้เขียนต้องมองโปรแกรมออกเป็นออบเจกต์ย่อย ๆ ที่มีที่เก็บข้อมูล มีฟังก์ชันการทำงานภายในออบเจกต์ แล้วนำเอาออบเจกต์เหล่านั้น มาผสมกันในอัตราที่พอเหมาะ จนเกิดเป็นโปรแกรมขึ้นใช้งาน แต่ถ้าเป็นแบบโครงสร้างแล้ว ผู้เขียนต้องมองโปรแกรมออกเป็นลักษณะการทำงานแบบที่การทำงานต้องมีรูปแบบที่แน่นอน เช่น แบบเรียงลำดับ แบบมีเงื่อนไข หรือแบบทำซ้ำไปมา แต่เราสามารถนำเอาความคิดแบบโครงสร้างไปใช้ร่วมกับแบบอังกิววัตถุได้ เพื่อทำให้ออบเจกต์ทำงานแบบมีโครงสร้างมากขึ้น

3.1.1 คุณสมบัติของการเขียนโปรแกรมแบบอังกิววัตถุ

1. การซ่อนรายละเอียด (Encapsulation) คือ คุณสมบัติในการห่อหุ้มข้อมูลที่อยู่ภายในออบเจกต์ด้วยวิธีการ (Method) ที่เตรียมไว้เพื่อให้จัดการกับข้อมูลนั้น เพื่อป้องกันไม่ให้ภายนอกสามารถเข้าถึงข้อมูลโดยตรง ทำให้มีความปลอดภัยและง่ายต่อการซ่อมบำรุง (Maintenance)

2. การสืบทอดคุณสมบัติ (Inheritance) คือ การสืบทอดความสามารถของออบเจกต์หนึ่งไปยังอีกออบเจกต์หนึ่ง เพื่อแทนที่จะประกาศคลาส (Class) หลาย ๆ คลาสที่มีคุณสมบัติความสามารถบางส่วนเหมือนกัน เราสามารถประกาศพารามิเตอร์ (Parent class) แล้วประกาศคลาสอื่น ๆ มาสืบทอดคุณสมบัติและความสามารถของคลาสนี้ได้ ทำให้คลาสนั้นสามารถนำกลับมาใช้ใหม่ได้

3. โพลิมอร์ฟิซึม (Polymorphism) คือ คุณสมบัติในการที่มีวิธีการหลาย ๆ วิธีการที่มีชื่อเดียวกัน แต่มีความสามารถต่างกัน ซึ่งการเลือกจะใช้วิธีการไหนก็ขึ้นอยู่กับพารามิเตอร์ (Parameter) ที่รับเข้ามา คุณสมบัตินี้ทำให้ซอฟต์แวร์มีความยืดหยุ่น

4. การแยกแยะเอกลักษณ์ (Abstraction) คือ วิธีการหนึ่งที่มนุษย์ใช้ในการมองสิ่งใด ๆ เพื่อลดความซับซ้อนของสิ่งนั้น ๆ โดยแยกแยะเอาเฉพาะสิ่งที่ผู้มองสนใจออกมา ซึ่งแน่นอนว่าถ้าผู้มองระบบเป็นคนละคนกันก็จะแยกแยะออกมาได้ต่างกัน การแยกแยะเอกลักษณ์เป็นการแสดงคุณลักษณะที่สำคัญของออบเจกต์ ซึ่งเป็นคุณลักษณะที่ทำให้ออบเจกต์นั้นแตกต่างจากออบเจกต์อื่น ๆ เช่น ออบเจกต์รถยนต์ ถ้าเป็นการแยกแยะเอกลักษณ์ของนักเดินทางก็จะมองว่ารถยนต์นั้นต้องพาผู้โดยสารไปยังปลายทางได้สะดวกรวดเร็ว แต่ถ้าเป็นการแยกแยะเอกลักษณ์ของช่างเครื่องก็จะมองว่ารถยนต์ประกอบด้วยเครื่องยนต์ต่าง ๆ มากมายที่ทำงานร่วมกัน

3.1.2 คลาส (Class) และการส่งข้อความ (Message)

คลาส คือ ต้นแบบ (Template) ของออบเจกต์ โดยประกอบไปด้วยองค์ประกอบพื้นฐานที่สำคัญสองส่วน คือ คาด้าเมมเบอร์ (Data member) และ เมมเบอร์ฟังก์ชัน (Member function)

คาด้าเมมเบอร์ หรือคุณลักษณะ (Attribute) หมายถึง ข้อมูลของออบเจกต์นั้น ๆ เป็นลักษณะเฉพาะตัวที่แสดงให้เห็นถึงความแตกต่างกันของวัตถุแต่ละชนิด เช่น สี, ขนาด, จำนวน ส่วนเมมเบอร์ฟังก์ชัน หรือวิธีการ จะหมายถึงการทำงาน (ตามโปรแกรม) (Processing code) ของออบเจกต์ เป็นส่วนที่จะบอกว่า คลาสนั้นทำอะไรได้บ้าง สามารถทำงานร่วมกับคลาสอื่น ๆ ได้อย่างไร และคลาสอื่น ๆ สามารถสั่งงานคลาสนั้นได้อย่างไร

การส่งข้อความ เป็นตัวกลางในการติดต่อสื่อสารกันระหว่างแต่ละออบเจกต์ โดยข้อมูลที่จะถูกส่งไปนั้นประกอบด้วย

- ออบเจกต์ปลายทาง
- ชื่อของวิธีการที่จะทำการติดต่อ
- พารามิเตอร์ต่าง ๆ ที่วิธีการต้องการ

3.2 ภาษาจาวา

ภาษาจาวาเป็นภาษาที่ใช้หลักการของการเขียนโปรแกรมแบบอ้างอิงวัตถุ ในการเขียนโปรแกรมแบบนี้ สามารถลดความซับซ้อนโครงสร้างของโปรแกรมรวมถึงการอำนวยความสะดวกในการนำส่วนประกอบต่าง ๆ ของโปรแกรมที่เขียนไว้แล้วมาใช้ใหม่ นอกจากนี้ภาษาจาวายังไม่ขึ้นกับแพลตฟอร์ม (Platform) ซึ่งจะช่วยให้เราสามารถพัฒนาระบบโดยใช้สภาพแวดล้อม (Environment) ของอะไรก็ได้ ซึ่งโดยทั่วไปมักแล้วจะเป็นสภาพแวดล้อมของวินโดว (Window environment)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาจาวาถูกพัฒนาจากบริษัท Sun Microsystems ซึ่งจัดให้เป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่ใช้หลักการออกแบบตัวภาษาด้วยวิธีเชิงวัตถุ และตัวภาษาถูกใช้เป็นเครื่องมือสำหรับพัฒนาโปรแกรมด้วยแนวคิดเชิงวัตถุ โดยตัวภาษามีลักษณะพิเศษดังนี้

- ความเข้ากันได้ (Portability) สามารถใช้งานในสภาพแวดล้อมที่แตกต่างโดยไม่ต้องมีการปรับแต่ง
- ความง่าย (Simple) ในการเขียนโปรแกรม
- ความคงสภาพ (Robust) ในการทำงาน มีโอกาสเกิดข้อผิดพลาดที่ไม่พึงประสงค์ได้น้อย
- ความปลอดภัย (Secure) การรองรับมาตรฐานความปลอดภัยในการใช้งานรูปแบบต่าง ๆ
- มีความสามารถในการประมวลผลแบบกระจาย (Distributed)
- วิธีการเชิงวัตถุ (Object Oriented) มีหลักการของแนวคิดเชิงวัตถุ ในการสร้างโปรแกรม

การสร้าง : ต้องเขียนคำสั่งตามรูปแบบของภาษาจาวาแล้วเก็บไว้ในไฟล์ที่เรียกว่าซอร์สโค้ด (Source code) จากนั้นนำไปคอมไพล์ (Compile) ไปเป็นไฟล์คลาส

การใช้งาน : ถ้าภาษาจาวาที่ใช้สร้างไฟล์คลาสเขียนในรูปแบบจาวาแอปพลิเคชัน (Java application) ให้ใช้งานบนเครื่องได้ทันทีผ่าน Java Virtual Machine เช่น ภาษาจาวาที่ใช้สร้างไฟล์คลาสเขียนในแบบ Java applet ก็ต้องสร้างเว็บ (Web) มาเรียกใช้งาน Java applet อีกครั้ง (2 ขั้นตอน) และเมื่อ Java applet ทำงาน Java Virtual Machine ที่อยู่บนบราวเซอร์ (Browser) ก็จะรัน Java applet ให้ทำงาน เป็นต้น

3.3 Java Remote Method Invocation : Java RMI

Java RMI เป็นข้อกำหนดทางด้านการสื่อสารระหว่างเครื่องคอมพิวเตอร์ ข้อหนึ่งที่สามารถนำไปใช้ประโยชน์ได้ในลักษณะเดียวกันกับ RPC (Remote Procedure Call) ในขณะที่ RPC ถูกนำไปใช้กับระบบซอฟต์แวร์ที่มีการโปรแกรมแบบโครงสร้าง แต่ Java RMI จะใช้กับระบบซอฟต์แวร์ที่มีการโปรแกรมแบบอ็อบเจกต์

ระบบแบบกระจาย (Distributed system) หรือ ระบบการประมวลผลแบบกระจาย มีประสิทธิภาพที่สูงในการติดต่อสื่อสารระหว่างกระบวนการ (Process) ซึ่งอยู่ต่างเครื่องกัน ภาษาจาวาสันับสนุนการใช้ซ็อกเก็ต ซึ่งเป็นกลไกการติดต่อสื่อสารระหว่างกระบวนการแบบพื้นฐานที่อยู่ในระดับดี เนื่องจากมีความยืดหยุ่น และมีประสิทธิภาพเพียงพอสำหรับการติดต่อสื่อสารที่ใช้กันทั่วไป แต่ซ็อกเก็ตมีความต้องการกฎการสื่อสาร (Protocol) ซึ่งมีการทำงานในลักษณะ ไคลเอ็นต์/เซิร์ฟเวอร์ (Client/Server) ที่จะจัดการการติดต่อสื่อสารในระดับแอปพลิเคชัน (Application) เพื่อที่จะเข้ารหัสและถอดรหัสข้อความ (Message) ที่จะใช้เพื่อการแลกเปลี่ยน ซึ่งการออกแบบกฎนี้ย่อมเป็นภาระของ

ผู้พัฒนาระบบซอฟต์แวร์ และกฎระเบียบที่ต้องพัฒนาขึ้นเองที่อาจเกิดข้อผิดพลาดขึ้นอันจะส่งผลกระทบต่อประสิทธิภาพของระบบที่นำกฎระเบียบนี้มาใช้ด้วย

RPC เป็นข้อกำหนดตัวหนึ่งที่มีความสามารถตรงความต้องการข้างต้น ซึ่งจะสร้างการติดต่อสื่อสารให้เกิดขึ้นได้ในลักษณะของการเรียกใช้โพรซีเจอร์ (Procedure call) แทนการส่งข้อความ หรือข้อความผ่านซ็อกเก็ตโดยตรง เมื่อเกิดความจำเป็นที่ต้องมีการติดต่อสื่อสารระหว่างออบเจกต์ในระดับโปรแกรมซึ่งออบเจกต์เหล่านี้มีอยู่ในแต่ละเครื่อง แต่เราไม่สามารถนำ RPC มาดัดแปลงใช้กับระบบแบบกระจายที่เป็นออบเจกต์โมเดล (Object model) ได้ ฉะนั้นระบบแบบกระจายที่เป็นออบเจกต์โมเดลจึงต้องการข้อกำหนดตัวใหม่อย่าง Java RMI

Java RMI ถูกพัฒนาขึ้นโดยทีมงานพัฒนาซอฟต์แวร์ของบริษัท Sun Microsystems ซึ่งออกแบบมาให้ทำงานได้ในสภาพแวดล้อมของจาวาเท่านั้น ในขณะที่รูปแบบการประมวลผลแบบกระจายตัวอื่น ๆ สามารถนำมาใช้กับออบเจกต์ของจาวาได้ แต่ไม่มีความสะดวก ไม่มีความราบรื่น ในการติดต่อสื่อสารระหว่างออบเจกต์ของระบบเหล่านั้นกับ ออบเจกต์ของจาวาเท่าที่ควร อย่างเช่น คอร์บา (CORBA) ที่สามารถจัดการกับสภาพแวดล้อมที่ต่างระบบกันได้ ซึ่งออบเจกต์ของคอร์บาจำเป็นต้องมีรูปแบบของภาษาที่เป็นกลาง ในทางกลับกัน Java RMI ซึ่งมีพื้นฐานมาจากภาษาจาวา จะทำให้ความสามารถของออบเจกต์ของจาวามาใช้ได้อย่างเต็มที่

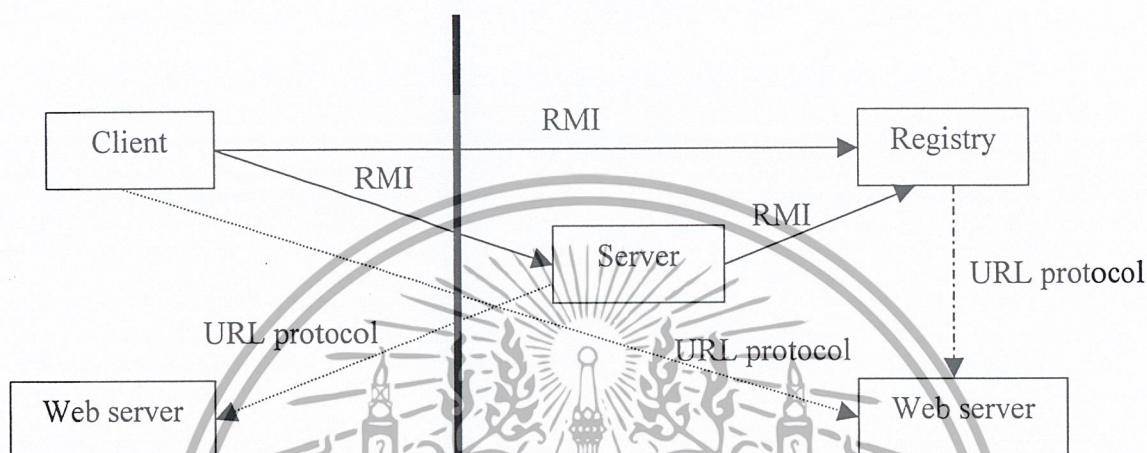
โปรแกรมที่ถูกพัฒนาขึ้นมาโดยใช้เทคโนโลยีของจาวาสามารถที่จะเรียกใช้ออบเจกต์ที่อยู่ต่างเครื่องได้โดยอาศัยค่าอ้างอิงของออบเจกต์เหล่านั้นซึ่งมันได้รับมา โดยการได้มาของค่าออบเจกต์นี้อาจได้มาจาก

- เข้าไปค้นหาจากบูทสทราปเนมมิงเซอร์วิส (Bootstrap naming service) ที่ระบบ Java RMI ได้จัดเตรียมไว้ให้
- ได้จากอาร์กิวเมนต์ (Argument) หรือ การส่งค่ากลับ (Return value) ที่ได้มาโดยการเรียกเมธอด

เครื่องทุกเครื่องสามารถเรียกใช้ออบเจกต์จากต่างเครื่องได้ โดยไม่มีการกำหนดสถานะไว้ว่าเครื่องนี้ต้องเป็นเซิร์ฟเวอร์หรือไคลเอ็นท์แต่เพียงอย่างเดียวตลอดเวลา และ Java RMI ยังใช้ออบเจกต์ซีเรียลไลเซชัน (Object serialization) เพื่อทำการมาร์แชล (Marshal) และการอันวาร์แชล (Unmarshal) ตัวพารามิเตอร์โดยไม่ตัดข้อมูลที่จะบอกชนิดของออบเจกต์ทำให้สามารถจัดการกับออบเจกต์ชนิดที่ผู้ใช้ (ผู้พัฒนาจาวาแอปพลิเคชัน) กำหนดขึ้นเองได้

จากรูป 3.1 เป็นตัวอย่างของการใช้ Java RMI ในระบบแบบกระจาย ที่มีการลงทะเบียน (Registry) เพื่อที่จะสามารถเข้าไปอ้างอิงรีโมทออบเจกต์ (Remote object) โดยเซิร์ฟเวอร์จะเป็นผู้เรียกตัวอ้างอิงเพื่อที่จะกำหนดชื่อให้กับรีโมทออบเจกต์ ส่วนตัวไคลเอ็นท์นั้นจะมองหารีโมทออบเจกต์โดยใช้ชื่อของตัวรีโมทออบเจกต์นั้นที่อยู่ในเซิร์ฟเวอร์ของการลงทะเบียน (Server's registry) จากเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นก็จะทำการเรียกขอวิธีการที่มีอยู่ในรีโมทออบเจกต์นั้น นอกจากนั้นในรูปยังแสดงถึงระบบของ Java RMI ที่มีการใช้เว็บเซิร์ฟเวอร์ (Web server) ในการโหลดไบต์โค้ด (Byte code) จากเซิร์ฟเวอร์ไปยังไคลเอ็นต์และจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์เมื่อมีการใช้ออบเจกต์นั้นอีกด้วย



รูปที่ 3.1 แสดงตัวอย่างการใช้ Java RMI

3.4 Java Swing

Java Swing เป็นส่วนหนึ่งของ JFC (Java Foundation Classes) ในแพลตฟอร์มของจาวา โดย JFC เป็นไลบรารี (Library) ที่ช่วยในการสร้างส่วนติดต่อกับผู้ใช้แบบกราฟฟิค (Graphic User Interface: GUI)

Java Swing ถูกเพิ่มเข้าไปในแพลตฟอร์มของจาวาครั้งแรกใน JDK 1.1 (Java Development Toolkit 1.1) ซึ่งก่อนหน้านี้ส่วนที่ใช้ในการสร้างส่วนติดต่อกับผู้ใช้ คือ AWT (Abstract Window Toolkit) ซึ่งยังคงรองรับอยู่ในแพลตฟอร์ม Java 2 โดยในชุดของแพ็คเกจ Java Swing จะมีส่วนที่คล้ายกับ AWT แต่จะขึ้นต้นคลาสด้วยตัวอักษร “J” เช่น คลาสปุ่ม (Button) ใน AWT จะมีชื่อว่า Button ส่วนคลาสปุ่มใน Java Swing จะมีชื่อว่า JButton เป็นต้น ส่วนของ AWT จะอยู่ในแพ็คเกจ java.awt และส่วนของ Java Swing จะอยู่ในแพ็คเกจ javax.swing

การคอมไพล์และการรัน Java Swing (Compiling and Running Java Swing) นั้นจะต้องใช้ JDK ของแพลตฟอร์ม Java 2 และถ้าต้องการการรับสวิงแอปเพลต (Swing applet) ในเว็บเบราว์เซอร์ จะต้องมีการติดตั้งปลั๊กอิน (Java Plug-in) ในเครื่องที่ใช้ด้วย

3.5 UML (Unified Modeling Language)

UML คือ รูปแบบหนึ่งของภาษาหรือมาตรฐานที่ถูกกำหนดขึ้นเพื่อใช้ในการอธิบายถึงโครงสร้าง การทำงาน หรือความสัมพันธ์ของระบบที่มีความซับซ้อนนั่นเอง เป็นผลงานร่วมมือของ Grady Booch, James Rumbaugh และ Ivar Jacobson ซึ่งเป็นผู้เชี่ยวชาญระดับแนวหน้าของโลกในด้านการเขียนโปรแกรมแบบอ้างอิงวัตถุ โดยการเข้าร่วมงานใน Rational Software Cooperation ในปี ค.ศ. 1994

UML เป็นภาษาการออกแบบซอฟต์แวร์ในรูปแบบภาพวาด ซึ่งใช้ในการประกอบกันเป็นไดอะแกรม (Diagram) โดยมีกฎในการประกอบกันของส่วนต่าง (Element) ไดอะแกรมจะแสดงถึงมุมมองต่าง ๆ (Multiple view) ของระบบ ซึ่งจะรวมเรียกว่าโมเดล (Model) โมเดล UML จะบ่งบอกถึงรายละเอียดของระบบ แต่จะไม่ระบุถึงวิธีการในการพัฒนาระบบ (Implementation) ซึ่งการออกแบบนั้นจะประกอบไปด้วยการออกแบบสิ่งต่าง ๆ ดังต่อไปนี้

3.5.1 Class Diagram

เป็นการแสดงถึงแนวความคิดของขอบเขตของปัญหาใน UML การแสดงถึงกลุ่มของโครงสร้างของไดอะแกรม ซึ่งจะไม่มีการอธิบายถึงการทำงานต่าง ๆ แต่จะมีการแสดงความสัมพันธ์กันของออบเจกต์ต่าง ๆ ในระบบจะประกอบด้วยคุณสมบัติ และวิธีการ หรือคุณลักษณะ ออบเจกต์ที่คุณสมบัติและวิธีการเดียวกันจะรวมเรียกว่าคลาส หรือกล่าวในอีกนัยหนึ่ง คือ คลาสจะเป็นต้นแบบหรือพิมพ์เขียวของออบเจกต์ และในทางกลับกันออบเจกต์ก็คือ อินสแตนซ์ (Instance) ของคลาส

ขั้นตอนการเขียนคลาสไดอะแกรมเบื้องต้น

1. กำหนดคลาสต่าง ๆ ที่ควรมีในระบบ
2. กำหนดคุณสมบัติและวิธีการการทำงานของแต่ละคลาสที่คลาสนั้น ๆ สามารถทำได้
3. ทำการวาดคลาสไดอะแกรมตามที่กำหนด
4. เชื่อมโยงความสัมพันธ์ (Associate) ระหว่างคลาสต่าง ๆ

3.5.2 Use Case Diagram

Use Case Diagram จะอธิบายถึงกิจกรรมของระบบจากมุมมองของผู้ใช้ ในแง่ของนักพัฒนาระบบ เป็นการอธิบายขั้นตอนการทำงานของระบบโดยใช้ภาพประกอบถึงการทำงานของระบบ (Use Case), ผู้กระทำ และความสัมพันธ์ระหว่าง Use Case กับผู้กระทำ ซึ่ง Use Case จะแสดงด้วยวงรี ส่วนผู้กระทำ (Actor) จะแสดงโดยใช้รูปคน และเส้นตรงจะเป็นส่วนติดต่อระหว่างผู้กระทำกับ Use Case Diagram นี้มีความสำคัญมากในจุดเริ่มต้นในการกำหนดความต้องการของระบบ โดยพิจารณาจากมุมมองของผู้ใช้ ซึ่งจะช่วยให้ได้ระบบที่ตรงกับความต้องการและสามารถนำไปใช้งานได้จริง

ขั้นตอนในการเขียน Use Case

1. ทำการกำหนดหน้าที่การทำงานของระบบ ,ทำการกำหนดขอบเขต, กำหนดผู้กระทำ และ Use Case และลำดับขั้นตอนการทำงานของ Use Case
2. เขียนผู้กระทำ (Actor)
3. เขียน Use Case
4. เชื่อมโยงความสัมพันธ์ระหว่างผู้กระทำ กับ Use Case
5. หาก Use Case ใดที่มีขั้นตอนรายละเอียดใน Use Case นั้นมาก ก็สามารถเขียน Use Case นั้นในระดับที่ต่ำลงไปได้อีก โดยจะต้องแสดงความสัมพันธ์ของ Use Case ใหม่ที่เกิดขึ้น กับ Use Case เก่าด้วย

3.5.3 Activity Diagram

กิจกรรมที่เกิดขึ้นตาม Use Case หรือเกิดจากพฤติกรรมของออบเจกต์เองตามปกติ เป็นลำดับของกิจกรรม สามารถเขียนอยู่ในรูปของ Activity Diagram ซึ่งจะอธิบายการทำงานโดยละเอียดของ Use Case นั้น ๆ ว่าจะต้องทำการส่งค่าอะไร ไปที่ขั้นตอนไหนก่อนหลังเป็นลำดับ ซึ่งไดอะแกรมนี้จะมีส่วนช่วยในการออกแบบหน้าจอที่ไว้ติดต่อกับผู้ใช้ และรูปแบบฐานข้อมูลแบบคร่าว ๆ หรืออาจช่วยให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้

ขั้นตอนในการเขียน Activity Diagram

1. นำ Use Case มาวิเคราะห์หาขั้นตอนในการใช้ ออกแบบวิธีการส่ง และวิธีการเก็บข้อมูล
2. แบ่งส่วนของระบบเป็นส่วน ๆ เช่น ส่วนติดต่อกับผู้ใช้ ส่วนฐานข้อมูล ส่วนเซิร์ฟเวอร์ หรือไคลเอ็นท์ ขึ้นอยู่กับความเหมาะสม
3. เขียนขั้นตอนต่าง ๆ ทั้งหมด
4. เขียนความสัมพันธ์แบบลำดับขั้น
5. เขียนกำกับสิ่งที่ต้องส่งค่าให้

บทที่ 4

การออกแบบโปรแกรม

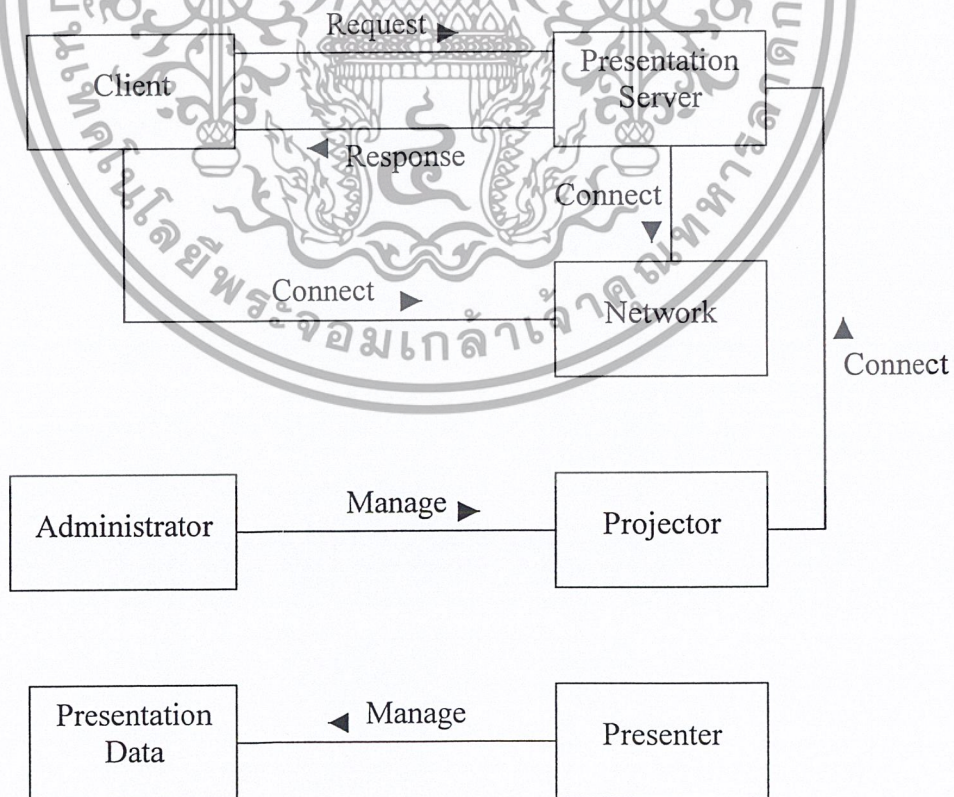
4.1 ความต้องการของผู้ใช้ (User Requirement)

จากการรวบรวมความต้องการของผู้ใช้ เมื่อต้องการใช้โปรแกรมนำเสนอ สามารถสรุปได้ดังต่อไปนี้

- การนำเสนองาน ถ้าหากสามารถผ่านระบบเครือข่ายได้น่าจะสะดวกมากขึ้น
- ต้องการกำจัดปัญหาที่เกี่ยวกับกับตั้งค่าคอนฟิกูเรชันต่าง ๆ เมื่อทำการเชื่อมต่อเครื่องคอมพิวเตอร์ที่มีงานนำเสนอ กับระบบเครือข่าย และสามารถใช้เครื่องฉายภาพที่ต่ออยู่กับระบบเครือข่ายได้ทันที
- หากส่งได้ทั้งภาพและเสียงจะดีมาก
- ให้โปรแกรมช่วยนำเสนอจัดการทุกอย่างกับค่าแอดเดรสต่าง ๆ

4.2 Class Diagram

สามารถเขียน Class Diagram เบื้องต้น (Initial Class Diagram) ตามความต้องการของผู้ใช้ได้ดังนี้

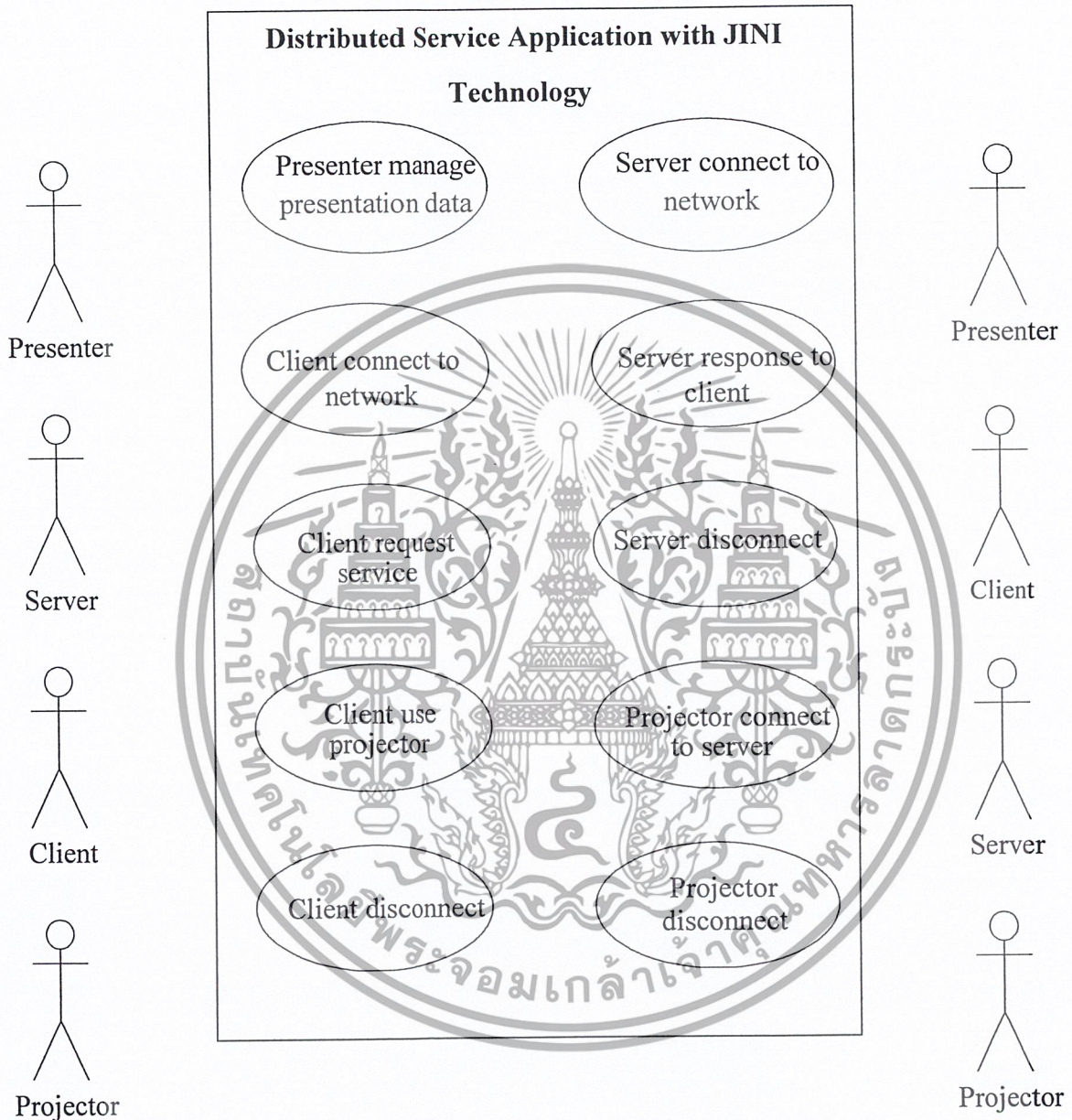


รูปที่ 4.1 แสดง Class Diagram เบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

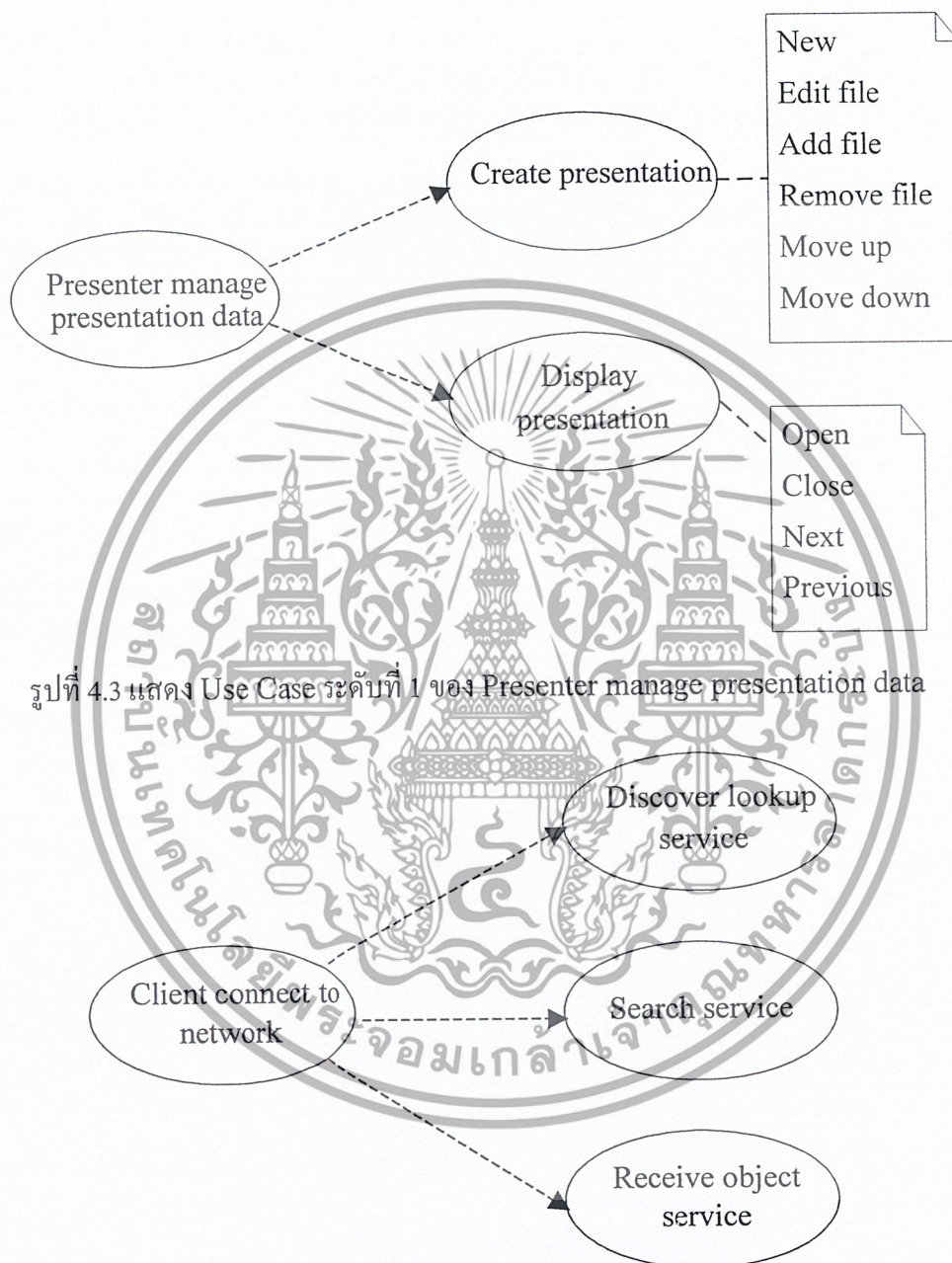
4.3 Use Case Diagram

สามารถเขียน Use Case ระดับบน (High Level Use Case) ได้ดังนี้



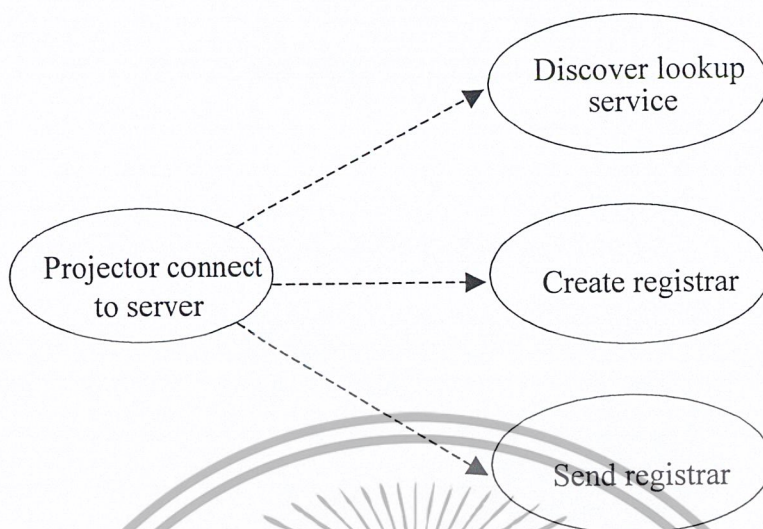
รูปที่ 4.2 แสดง High Level Use Case

หลังจากทำการวิเคราะห์เลือกเฉพาะ Use Case ที่เกี่ยวข้อง จะได้ Use Case ในระดับที่ลึก ลงไป และอธิบายการทำงานคร่าว ๆ ของแต่ละ Use Case ได้เป็น Use Case ระดับที่ 1 (Level 1 Use Case) ดังต่อไปนี้

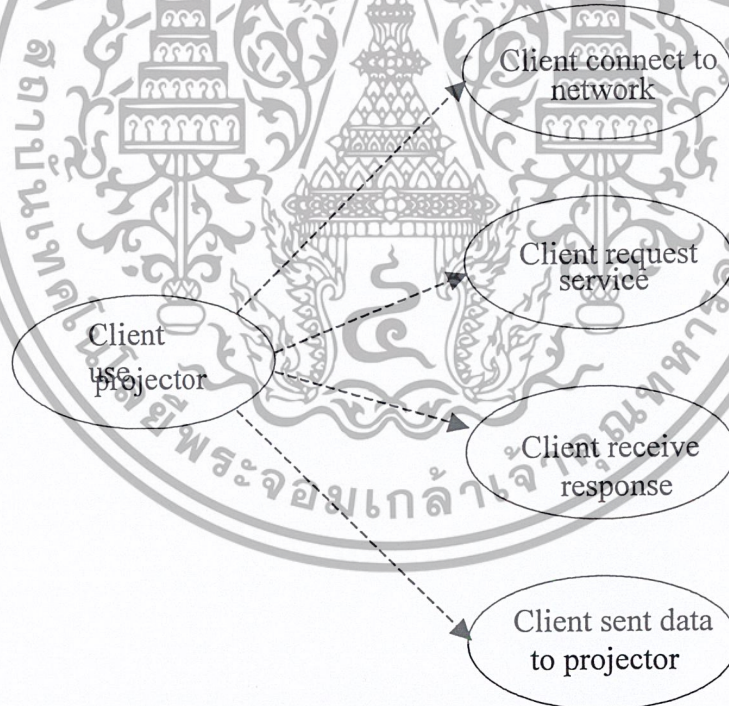


รูปที่ 4.3 แสดง Use Case ระดับที่ 1 ของ Presenter manage presentation data

รูปที่ 4.4 แสดง Use Case ระดับที่ 1 ของ Client connect to network



รูปที่ 4.5 แสดง Use Case ระดับที่ 1 ของ Projector connect to server

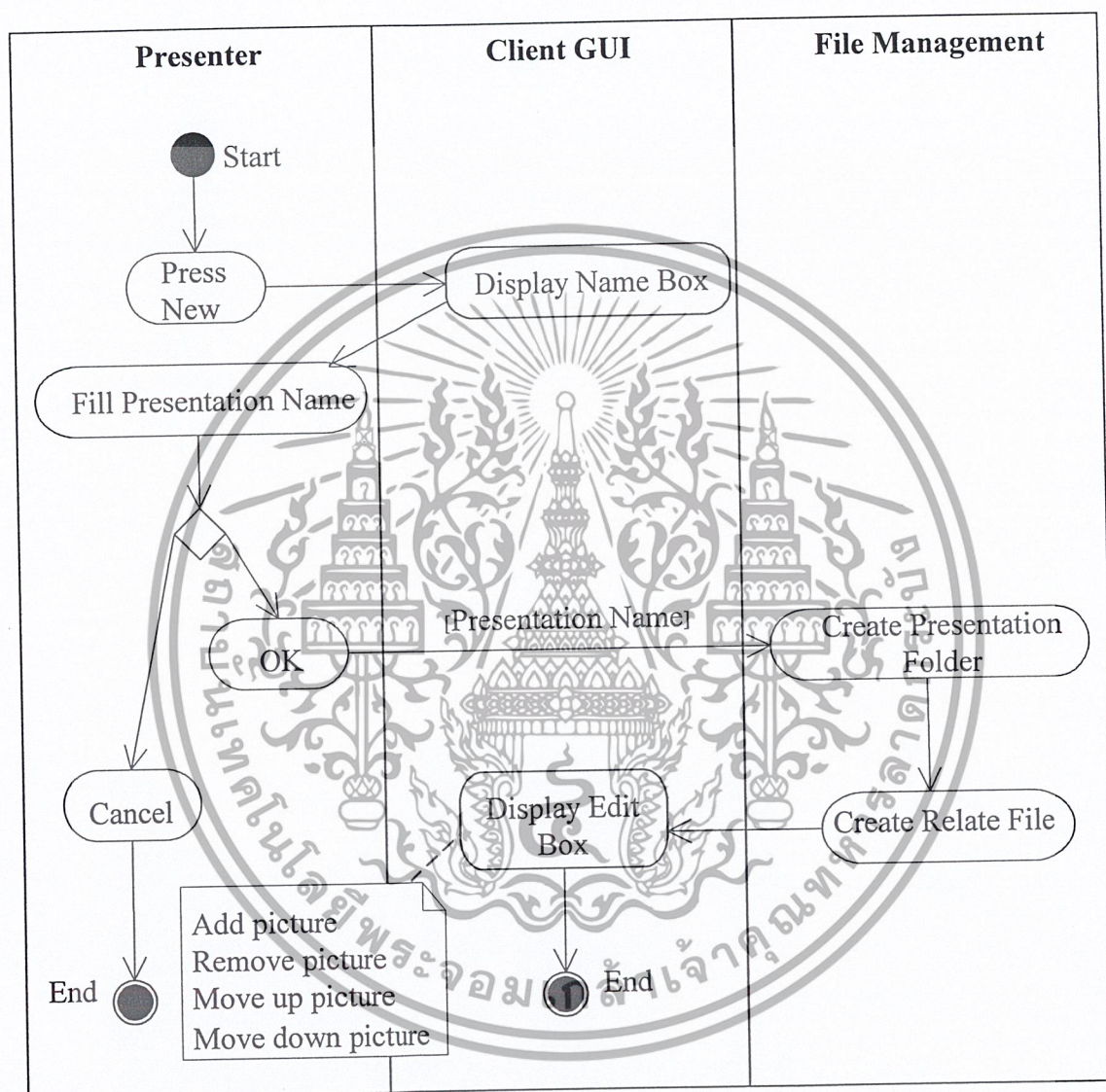


รูปที่ 4.6 แสดง Use Case ระดับที่ 1 ของ Client use projector

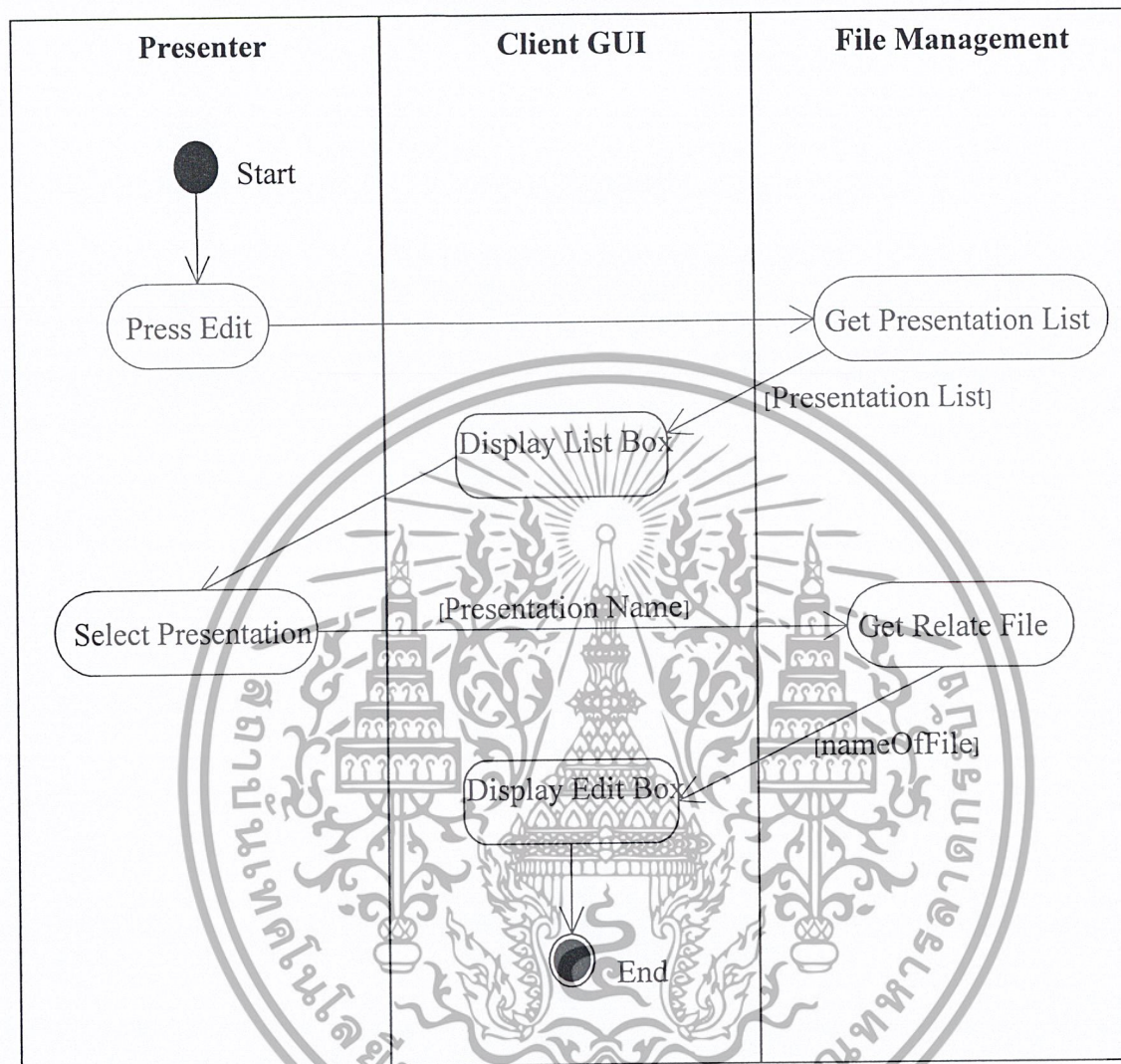
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 Activity Diagram

จาก Use Case การจัดการกับการนำเสนอ (Manage presentation) มีขั้นตอนละเอียดมากมายซึ่งสามารถนำมาเขียน Activity Diagram ได้ดังต่อไปนี้

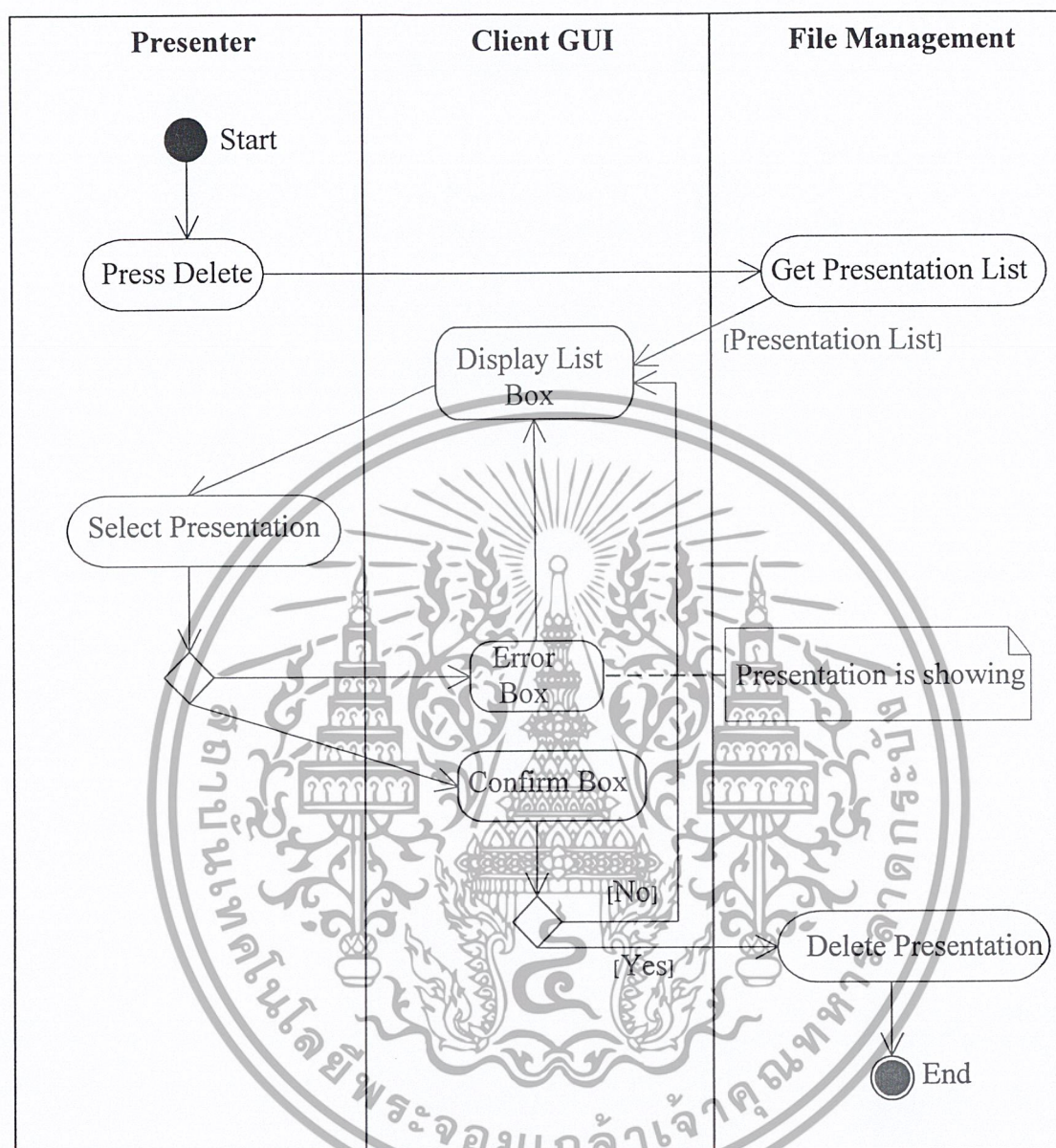


รูปที่ 4.7 แสดง Activity Diagram ของการสร้างไฟล์นำเสนอ (New)



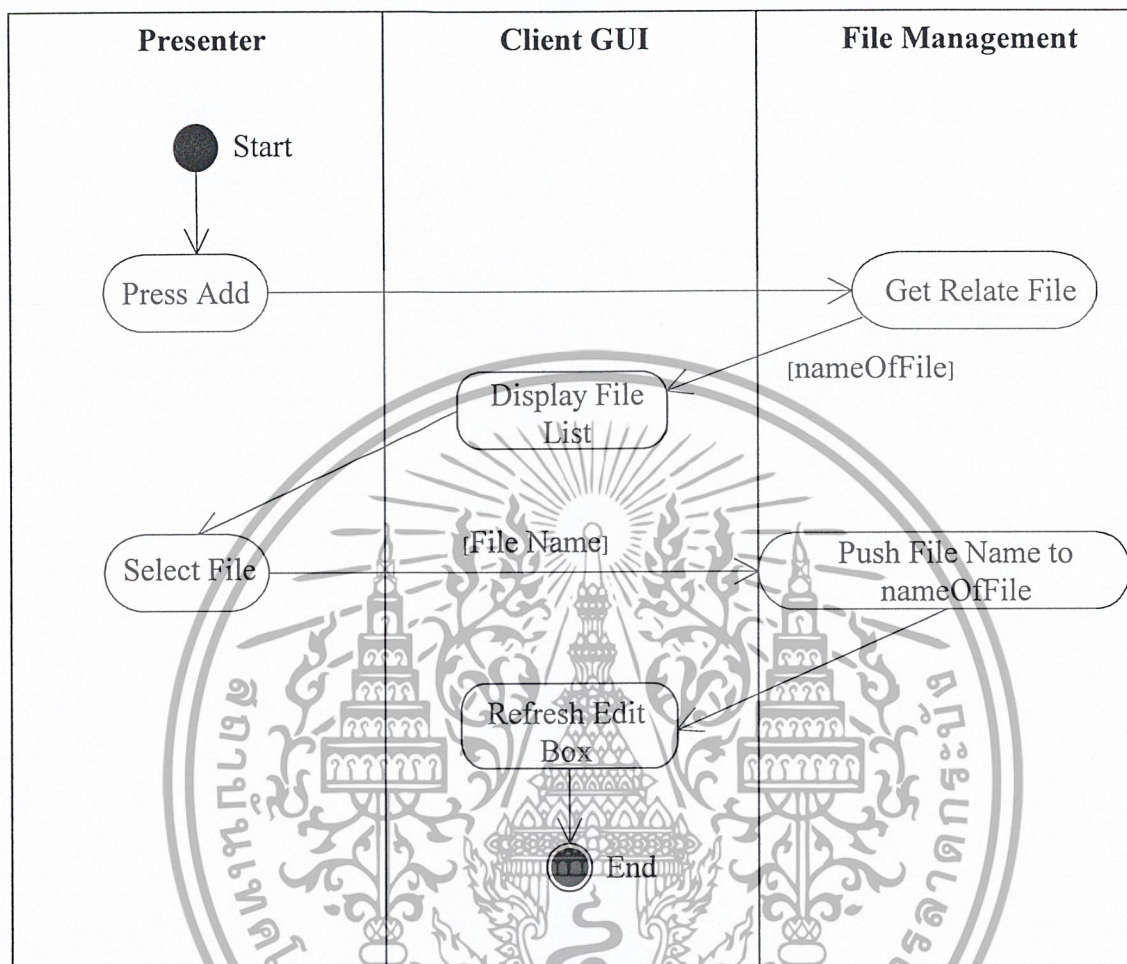
รูปที่ 4.8 แสดง Activity Diagram ของการแก้ไขไฟล์นำเสนอ (Edit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

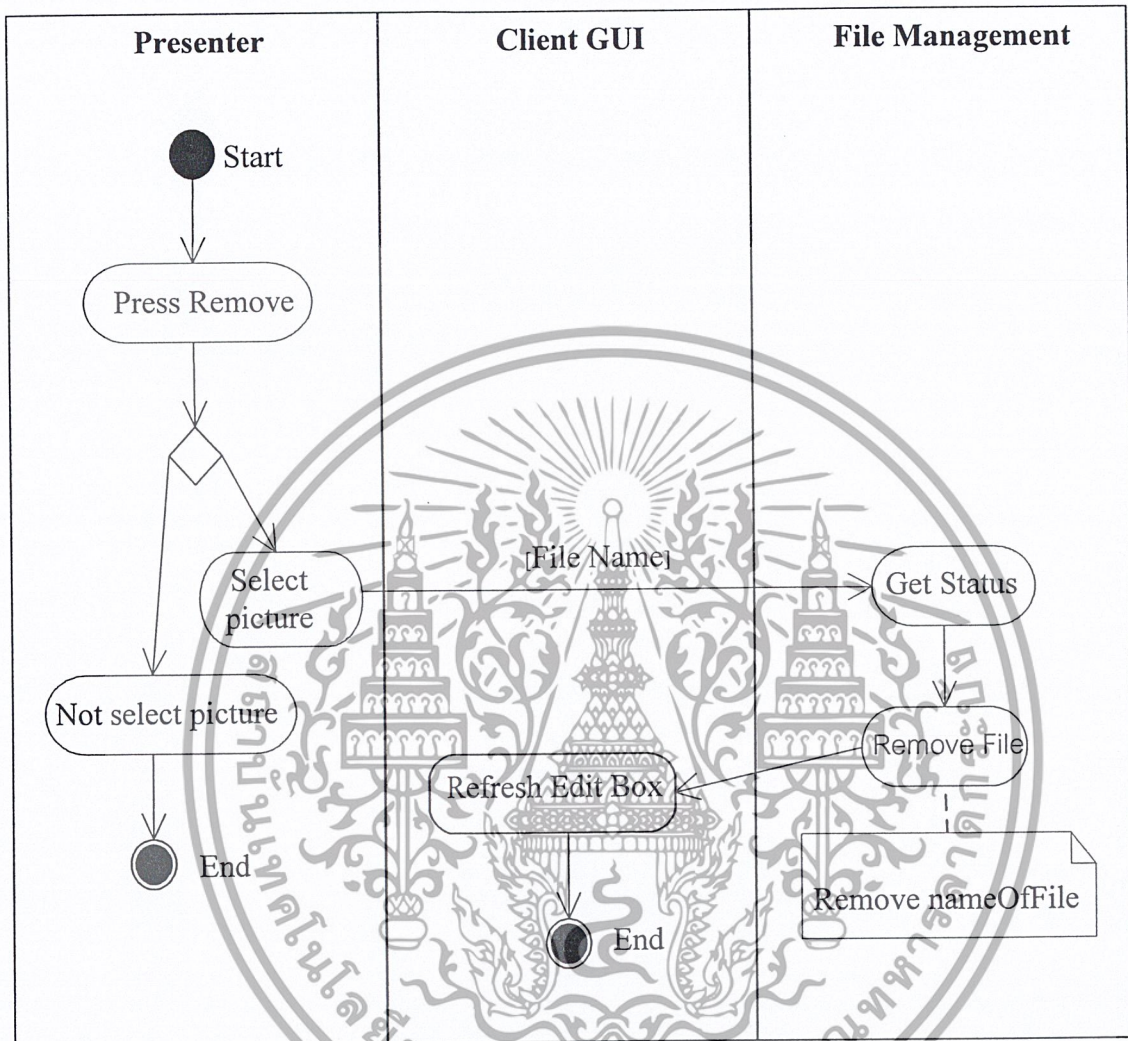


รูปที่ 4.9 แสดง Activity Diagram ของการลบไฟล์นำเสนอ (Delete)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

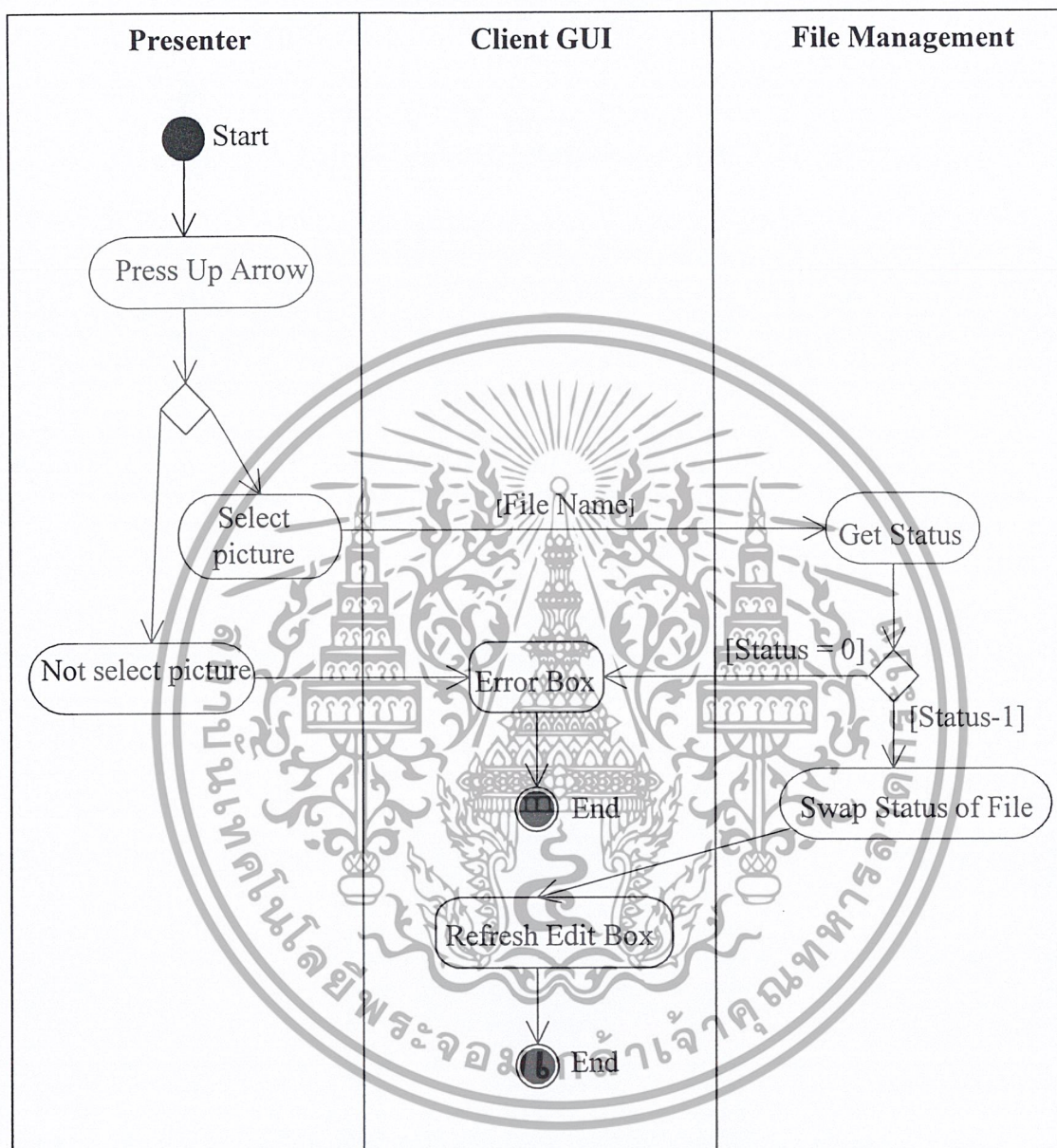


รูปที่ 4.10 แสดง Activity Diagram ของการเพิ่มรูปภาพในไฟล์นำเสนอ (Add)



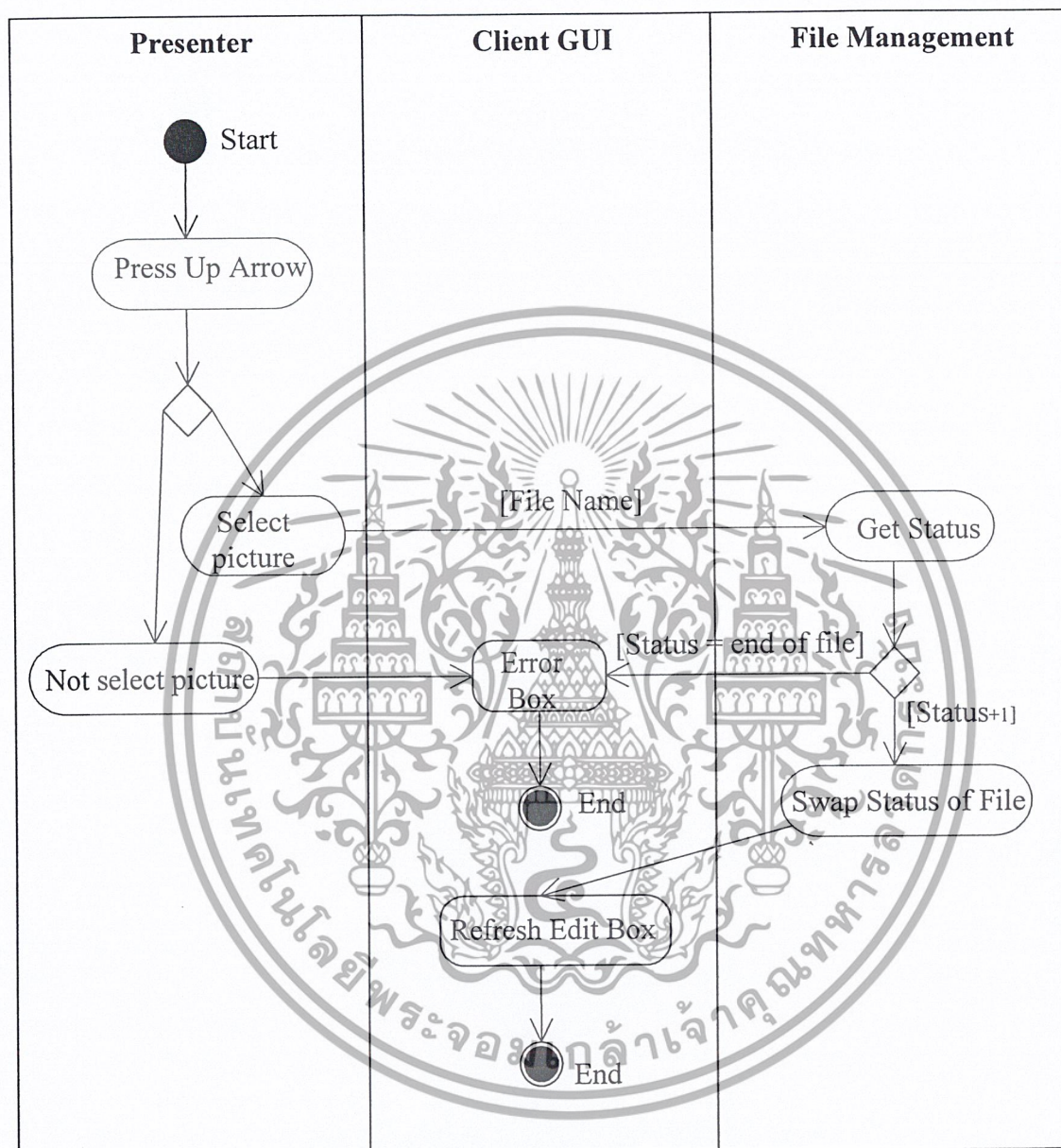
รูปที่ 4.11 แสดง Activity Diagram ของการลบไฟล์รูปภาพในไฟล์นำเสนอ (Remove)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



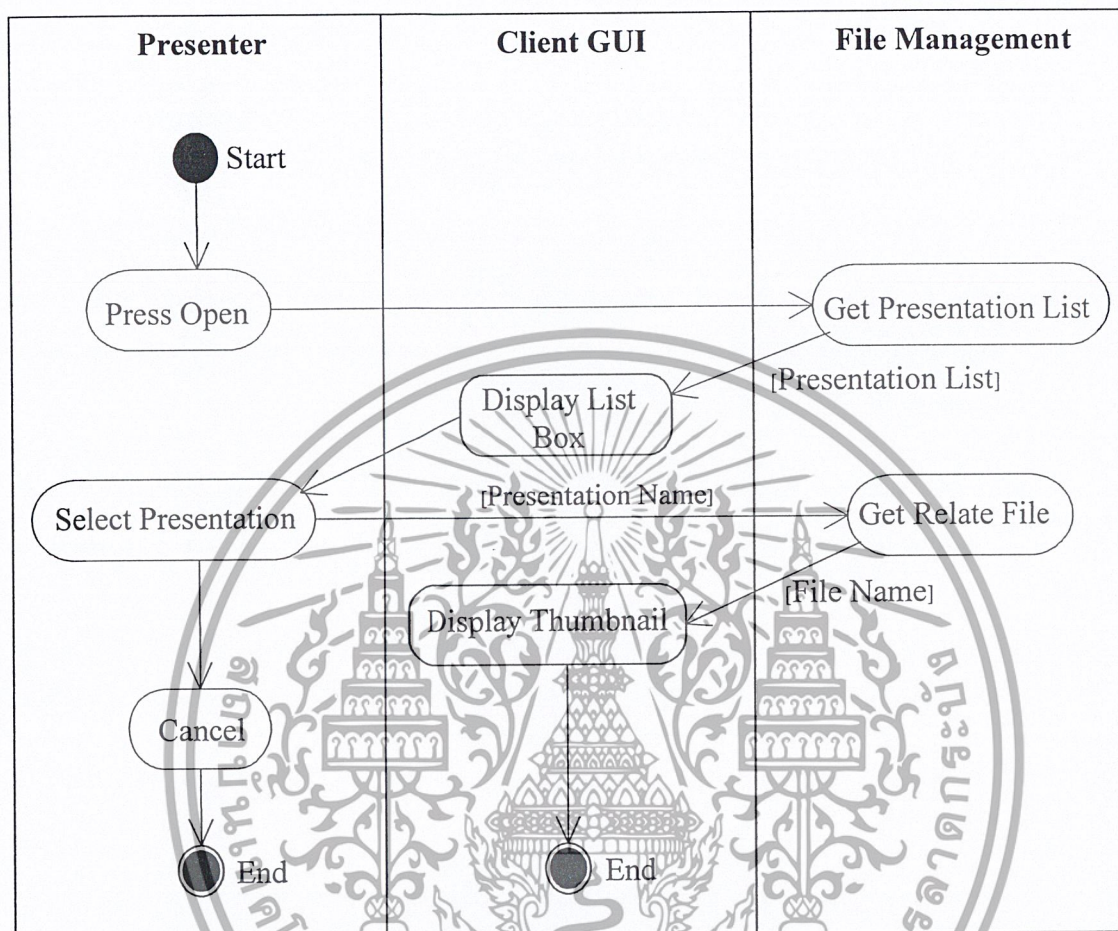
รูปที่ 4.12 แสดง Activity Diagram ของเลื่อนลำดับของรูปภาพขึ้น (Up)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

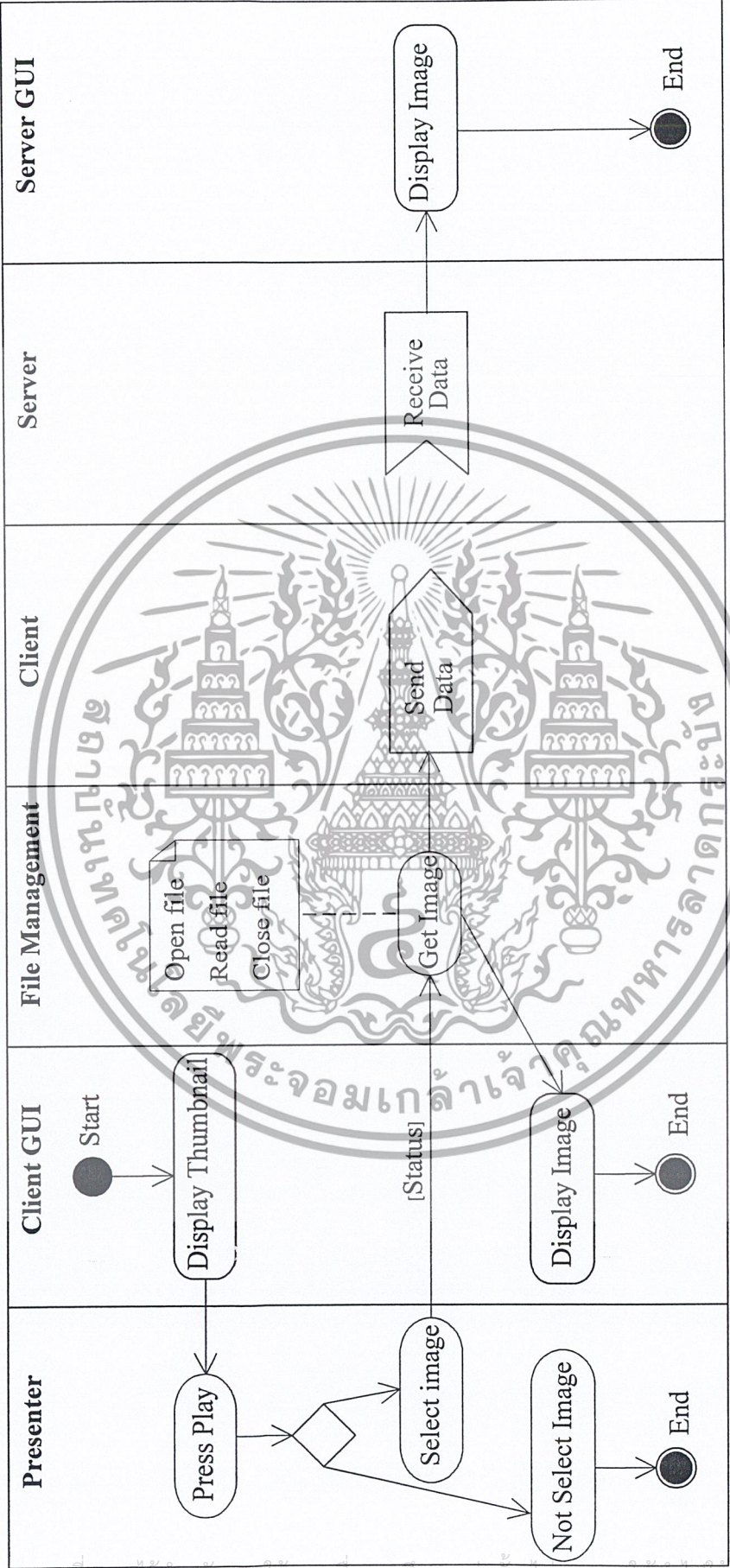


รูปที่ 4.13 แสดง Activity Diagram ของเลื่อนลำดับของรูปภาพลง (Down)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดง Activity Diagram ของการเปิดไฟล์นำเสนอ (Open)

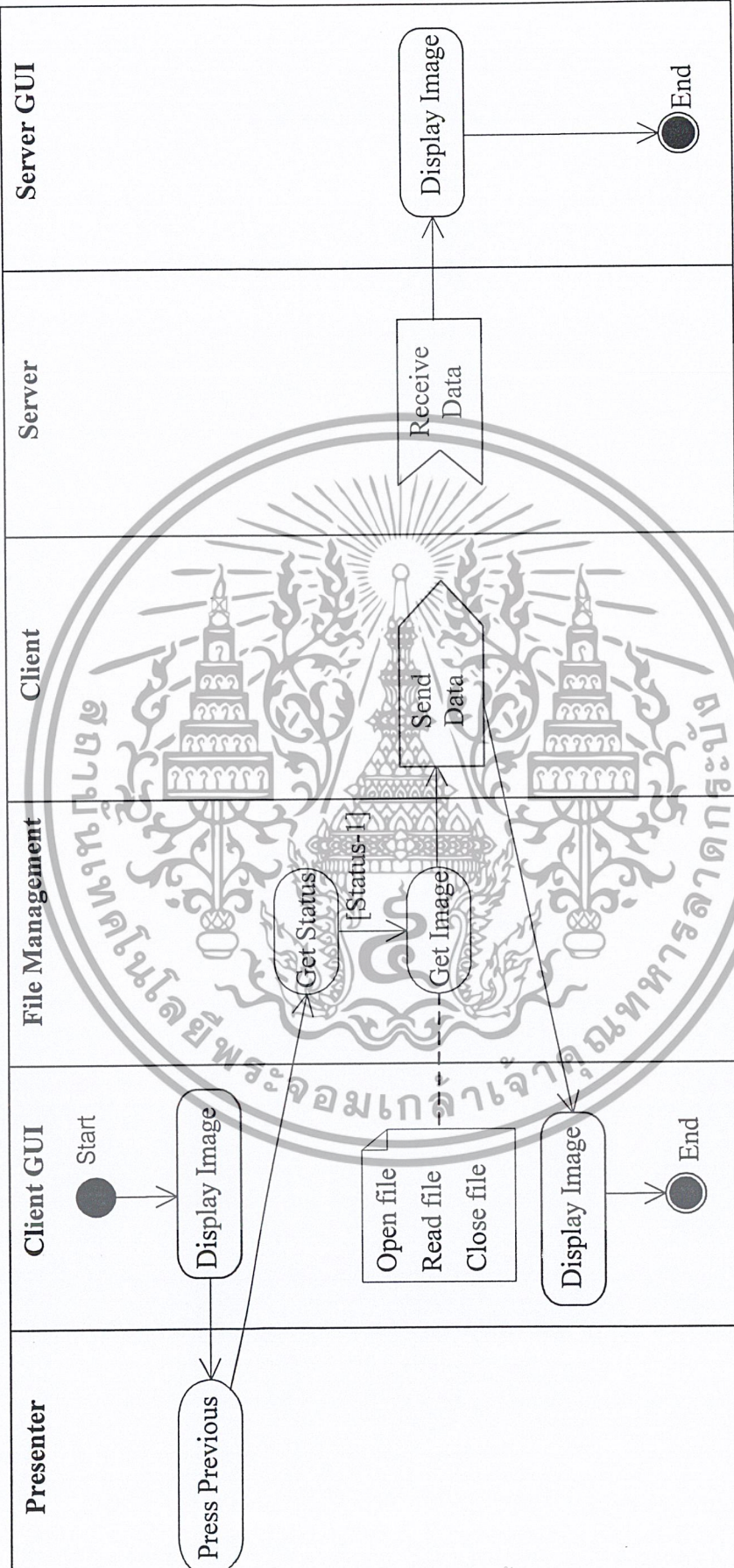


รูปที่ 4.15 แสดง Activity Diagram ของการเล่น ไฟล์นำเสนอ (Play)

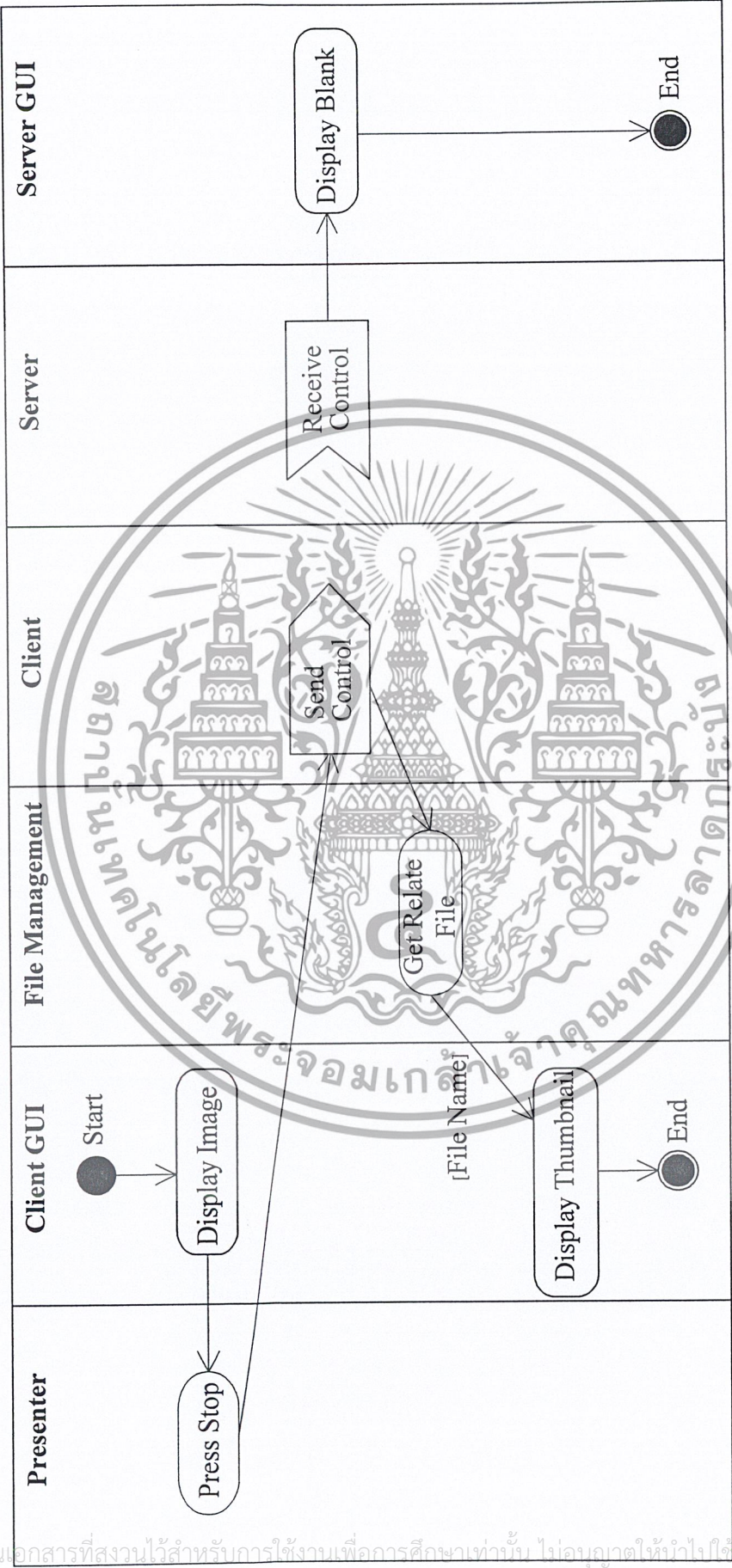


รูปที่ 4.16 แสดง Activity Diagram ของการนำเสนอภาพถัดไป (Next)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 แสดง Activity Diagram ของการนำเสนอภาพก่อนหน้า (Previous)



รูปที่ 4.18 แสดง Activity Diagram ของการหยุดการนำเสนอภาพ (Stop)

บทที่ 5

ผลการทดลอง

5.1 ผลการทดลองที่ส่วนไคลเอ็นต์

เมื่อทำการเขียนโปรแกรมตามที่ได้ออกแบบมา ทำการคอมไพล์ (Compile) ไฟล์ที่เกี่ยวข้องทุกไฟล์และทำการรัน (Run) โปรแกรม MainWindow.java จะเกิดหน้าจอหลักของโปรแกรมนำเสนอขึ้นดังรูป



รูปที่ 5.1 แสดงหน้าจอหลักหลังจากการทำการรันโปรแกรม MainWindow.java

เมื่อกดปุ่ม New จะเกิดเฟรมภายใน (Internal Frame) บนโอเปอเรชันวิว (Operation View) ซึ่งใช้สำหรับการสร้างไฟล์นำเสนอได้ เพื่อให้ผู้ใช้ใส่ข้อมูลของไฟล์นำเสนอต่าง ๆ ไม่ว่าจะเป็น ชื่อไฟล์นำเสนอ เนื้อหาของการนำเสนอ รูปแบบของเนื้อหานำเสนอ ฉากหลัง (Background) และการรูปแบบการเปลี่ยนหน้านำเสนอ และยังสามารถทำการบันทึกข้อมูลที่ได้กรอกไว้แล้ว เป็นไฟล์นำเสนอเพื่อใช้งานครั้งต่อไปได้ โดยมีลักษณะดังรูป 5.2 และสามารถทำการดูตัวอย่างในแต่ละหน้า โดยเลือกหน้าแล้วทำการกดปุ่ม Preview ซึ่งสามารถทำได้ทั้งในเฟรม New, เฟรม Open, เฟรม Edit ดังรูป 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

New Presentation

Presentation
 Presentation Name:
 Page:
 Current Page: 2

Background
 Gradient:

Template & Contents
 Template:
 Title:
 Size:
 Effect:
 Page Change Effect
 Close out Effect:

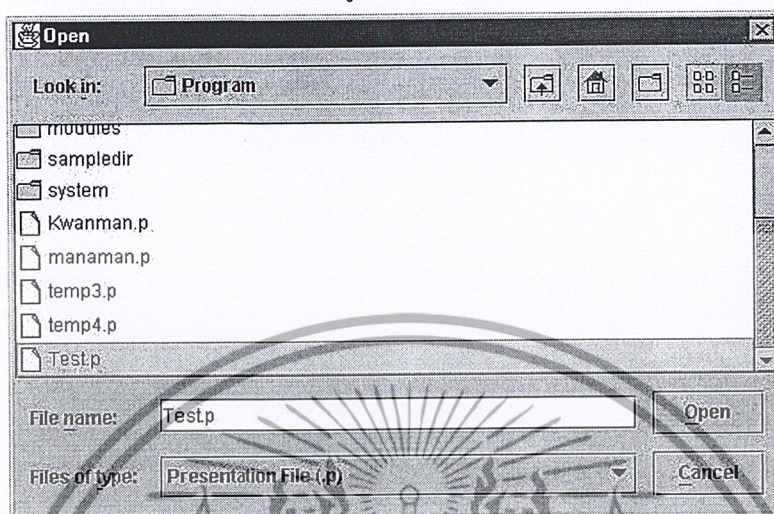
รูปที่ 5.2 แสดงเฟรมภายในที่เกิดขึ้นหลังจากการกดปุ่ม New



รูปที่ 5.3 แสดงเฟรมที่เกิดขึ้นเมื่อกดปุ่ม Preview

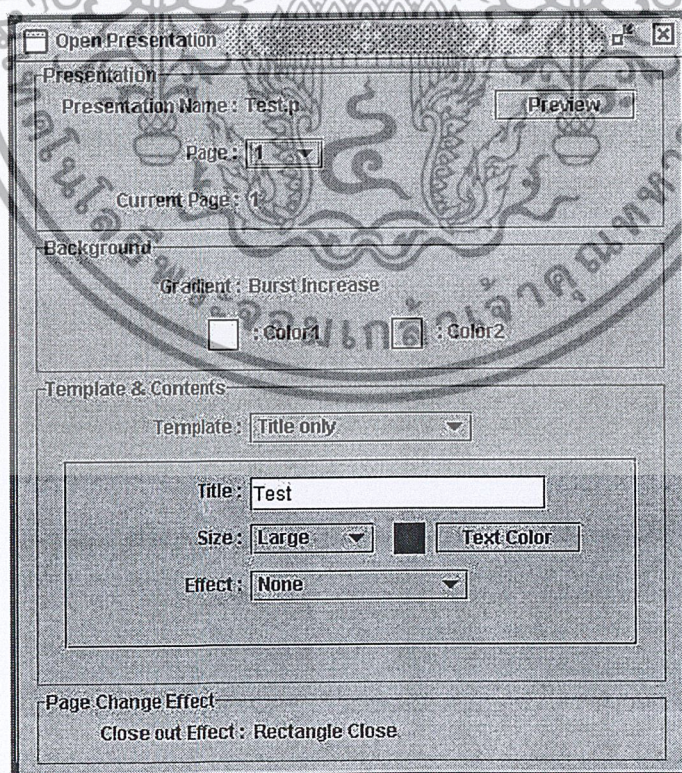
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกดปุ่ม Open จะเกิดเฟรมสำหรับการเลือกไฟล์นำเสนอที่จะเปิดขึ้น และเพื่อความสะดวกของผู้ใช้ ผู้ใช้สามารถเลือกเฉพาะไฟล์นำเสนอได้โดยทำการเลือกที่ชนิดของไฟล์เป็น .p ซึ่งเป็นชนิดของไฟล์นำเสนอ ตรงส่วนด้านล่างของเฟรม ดังรูป



รูปที่ 5.4 แสดงเฟรมเกิดขึ้นหลังจากการกดปุ่ม Open

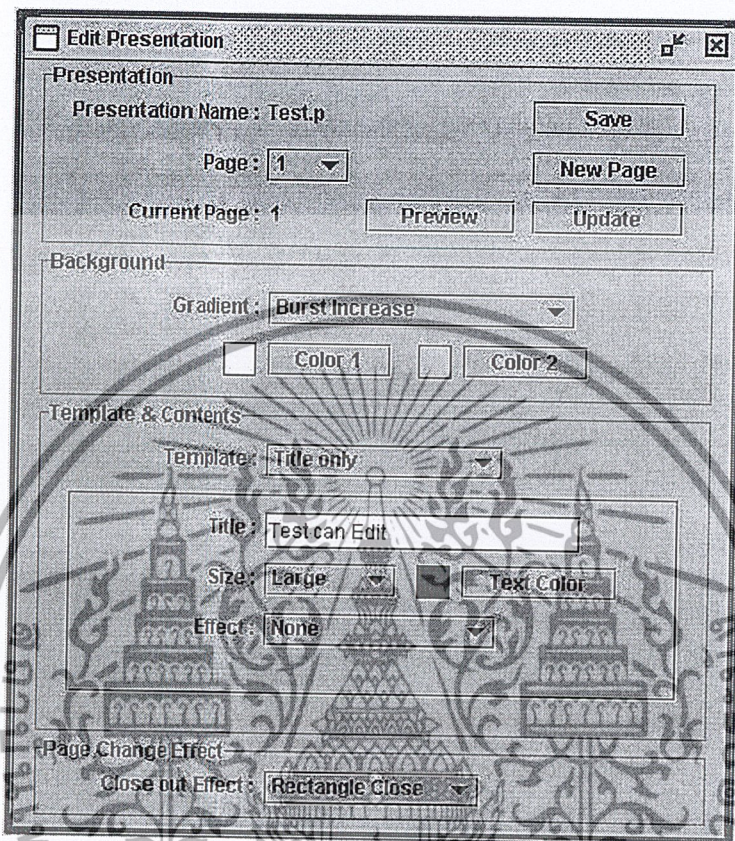
เมื่อทำการเลือกไฟล์ และทำการกดปุ่ม Open แล้วจะเกิดเป็นเฟรมภายในจันทันบนโอเปอร์เรชั่นวิว ดังรูป 5.5 โดยจะมีข้อมูลเก่าที่ทำไว้ก่อนหน้านี้แสดงอยู่ซึ่งสามารถเลือกดูหน้าต่างๆได้



รูปที่ 5.5 แสดงเฟรมภายในที่เกิดขึ้นหลังจากทำการเลือกไฟล์ที่จะเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกดปุ่ม Edit จะเกิดเฟรมเช่นเดียวกับเมื่อกดปุ่ม Open ดังรูปที่ 5.4 เพื่อใช้สำหรับเลือกไฟล์ที่จะทำการแก้ไขเมื่อเลือกไฟล์ที่จะแก้ไขแล้ว และกดปุ่ม Open ก็จะมีเฟรมภายในที่โอเปอเรชั่นวิวเพื่อใช้สำหรับการแก้ไข ก็จะสามารถแก้ไขข้อมูลต่าง ๆ ได้ ดังรูปที่ 5.6



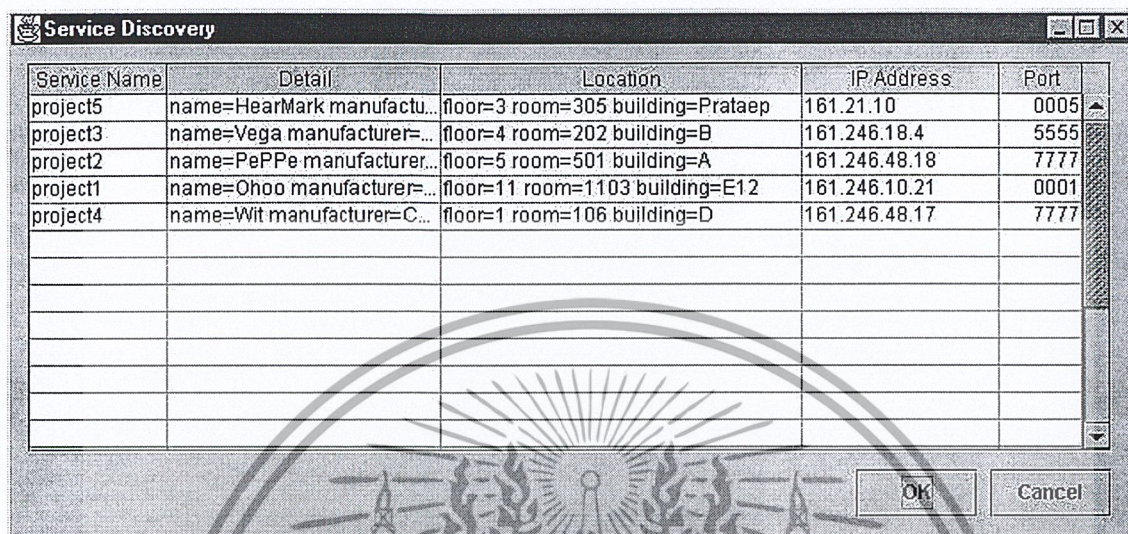
รูปที่ 5.6 แสดงเฟรมภายในที่เกิดขึ้นหลังจากทำการเลือกไฟล์ที่จะแก้ไข

เมื่อกดปุ่ม JINI จะเกิดการจัดการตามหลักของเทคโนโลยี JINI ดังนี้

1. ไคลเอ็นท์ค้นหา Lookup Locator โดยใช้การ Multicast ซึ่งจะได้เซอร์วิสพรอกซีของ Lookup Locator กลับมา
2. เมื่อไคลเอ็นท์หา Lookup Locator พบ ก็จะทำการค้นหาเซอร์วิสทั้งหมด โดยการหาเซอร์วิสพรอกซีแล้วแสดงเซอร์วิสที่ค้นพบทั้งหมดดังรูป 5.7
3. ผู้ใช้ทำการเลือกเซอร์วิสที่ต้องการ จากนั้นโปรแกรมจะทำการเชื่อมต่อเพื่อให้สามารถใช้เซอร์วิสนั้นๆ ได้
4. โปรแกรมจะทำการ Unicast ไปยังเซอร์วิสนั้นๆ ตามหลักการของเทคโนโลยี JINI เพื่อให้ไคลเอ็นท์สามารถใช้เซอร์วิสนั้นได้โดยตรง
5. เมื่อเลือกไฟล์นำเสนอแล้วปุ่ม Play โปรแกรมจะทำการส่ง Hashtable ทั้งหมดที่เก็บไฟล์นำเสนอตัวอย่างดังรูป 5.8 ไปยังเซอร์วิสเพื่อให้เซอร์วิสแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. การควบคุมไฟล์นำเสนอ นั้น โปรแกรมที่ไคลเอ็นท์จะทำการส่ง Message ไปยังเซิร์ฟเวอร์ เช่น จะแจ้งว่าได้ทำการกดปุ่ม Next แล้ว เป็นต้น



The screenshot shows a window titled "Service Discovery" with a table containing the following data:

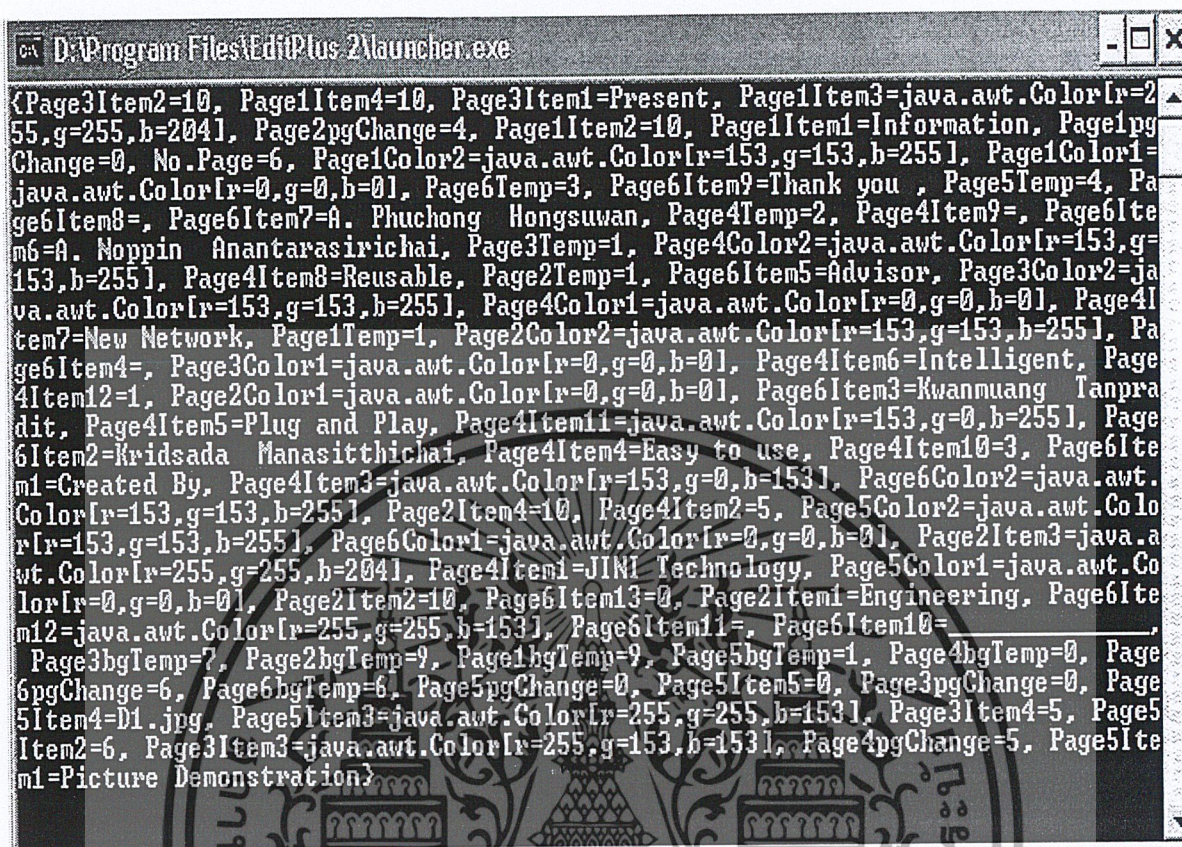
Service Name	Detail	Location	IP Address	Port
project5	name=HearMark manufactu...	floor=3 room=305 building=Prataep	161.21.10	0005
project3	name=Vega manufacturer=...	floor=4 room=202 building=B	161.246.18.4	5555
project2	name=PePpe manufacturer=...	floor=5 room=501 building=A	161.246.48.18	7777
project1	name=Ohoo manufacturer=...	floor=11 room=1103 building=E12	161.246.10.21	0001
project4	name=Wit manufacturer=C...	floor=1 room=106 building=D	161.246.48.17	7777

At the bottom of the window are "OK" and "Cancel" buttons.

รูปที่ 5.7 แสดงเฟรมแสดงเซิร์ฟเวอร์ที่สามารถเลือกใช้ได้

ตารางที่แสดงมีรายละเอียดดังต่อไปนี้

- Service Name บอกถึงชื่อของเซิร์ฟเวอร์ที่สามารถให้บริการได้
- Detail บ่งบอกถึงรายละเอียดต่าง ๆ ของเซิร์ฟเวอร์นั้น ๆ เช่น ชื่อของเจ้าของ โรงงานที่ผลิต หมายเลขประจำตัวของเครื่อง เป็นต้น
- Location บอกถึงตำแหน่งปัจจุบันของเซิร์ฟเวอร์นั้น ๆ
- IP Address บอกถึงหมายเลข Address ของเซิร์ฟเวอร์นั้น
- Port บ่งบอกหมายเลขช่องที่เซิร์ฟเวอร์นั้น ๆ ให้บริการ



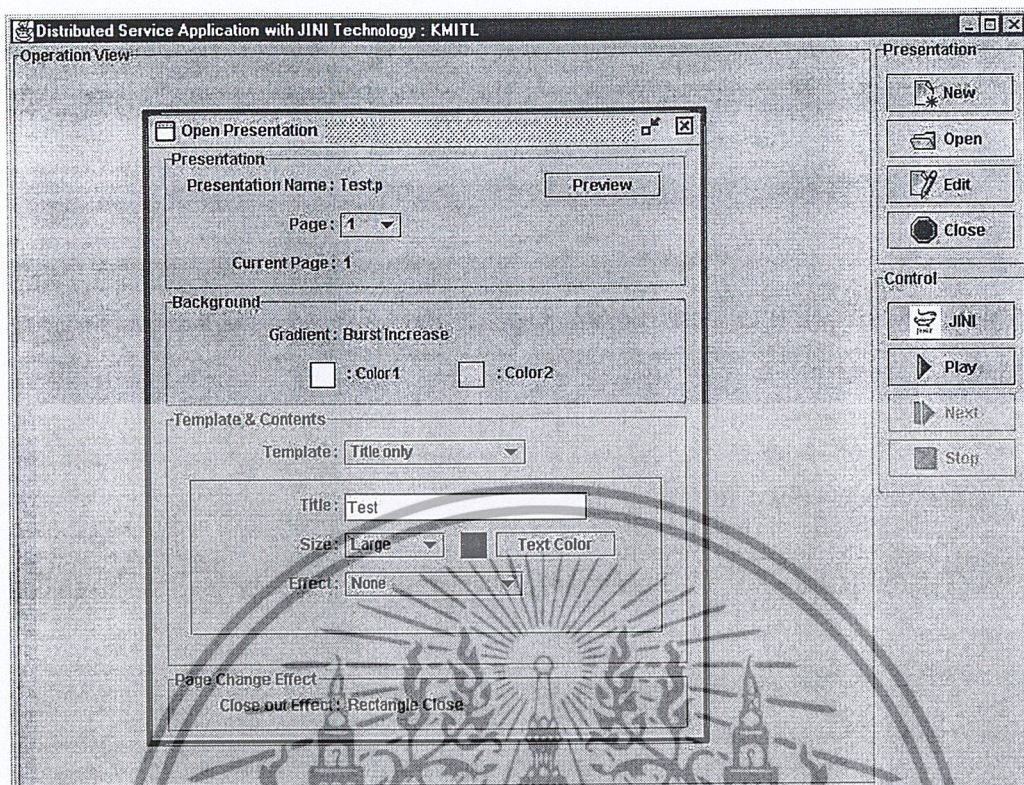
```

{Page3Item2=10, Page1Item4=10, Page3Item1=Present, Page1Item3=java.awt.Color[r=255,g=255,b=204], Page2pgChange=4, Page1Item2=10, Page1Item1=Information, Page1pgChange=0, No.Page=6, Page1Color2=java.awt.Color[r=153,g=153,b=255], Page1Color1=java.awt.Color[r=0,g=0,b=0], Page6Temp=3, Page6Item9=Thank you , Page5Temp=4, Page6Item8=, Page6Item7=A. Phuchong Hongsuwan, Page4Temp=2, Page4Item9=, Page6Item6=A. Noppin Anantarasirichai, Page3Temp=1, Page4Color2=java.awt.Color[r=153,g=153,b=255], Page4Item8=Reusable, Page2Temp=1, Page6Item5=Advisor, Page3Color2=java.awt.Color[r=153,g=153,b=255], Page4Color1=java.awt.Color[r=0,g=0,b=0], Page4Item7=New Network, Page1Temp=1, Page2Color2=java.awt.Color[r=153,g=153,b=255], Page6Item4=, Page3Color1=java.awt.Color[r=0,g=0,b=0], Page4Item6=Intelligent, Page4Item12=1, Page2Color1=java.awt.Color[r=0,g=0,b=0], Page6Item3=Kwanmuang Tanpradit, Page4Item5=Plug and Play, Page4Item11=java.awt.Color[r=153,g=0,b=255], Page6Item2=Kridsada Manasitthichai, Page4Item4=Easy to use, Page4Item10=3, Page6Item1=Created By, Page4Item3=java.awt.Color[r=153,g=0,b=153], Page6Color2=java.awt.Color[r=153,g=153,b=255], Page2Item4=10, Page4Item2=5, Page5Color2=java.awt.Color[r=153,g=153,b=255], Page6Color1=java.awt.Color[r=0,g=0,b=0], Page2Item3=java.awt.Color[r=255,g=255,b=204], Page4Item1=JINI Technology, Page5Color1=java.awt.Color[r=0,g=0,b=0], Page2Item2=10, Page6Item13=0, Page2Item1=Engineering, Page6Item12=java.awt.Color[r=255,g=255,b=153], Page6Item11=, Page6Item10=, Page3bgTemp=7, Page2bgTemp=9, Page1bgTemp=9, Page5bgTemp=1, Page4bgTemp=0, Page6pgChange=6, Page6bgTemp=6, Page5pgChange=0, Page5Item5=0, Page3pgChange=0, Page5Item4=D1.jpg, Page5Item3=java.awt.Color[r=255,g=255,b=153], Page3Item4=5, Page5Item2=6, Page3Item3=java.awt.Color[r=255,g=153,b=153], Page4pgChange=5, Page5Item1=Picture Demonstration}

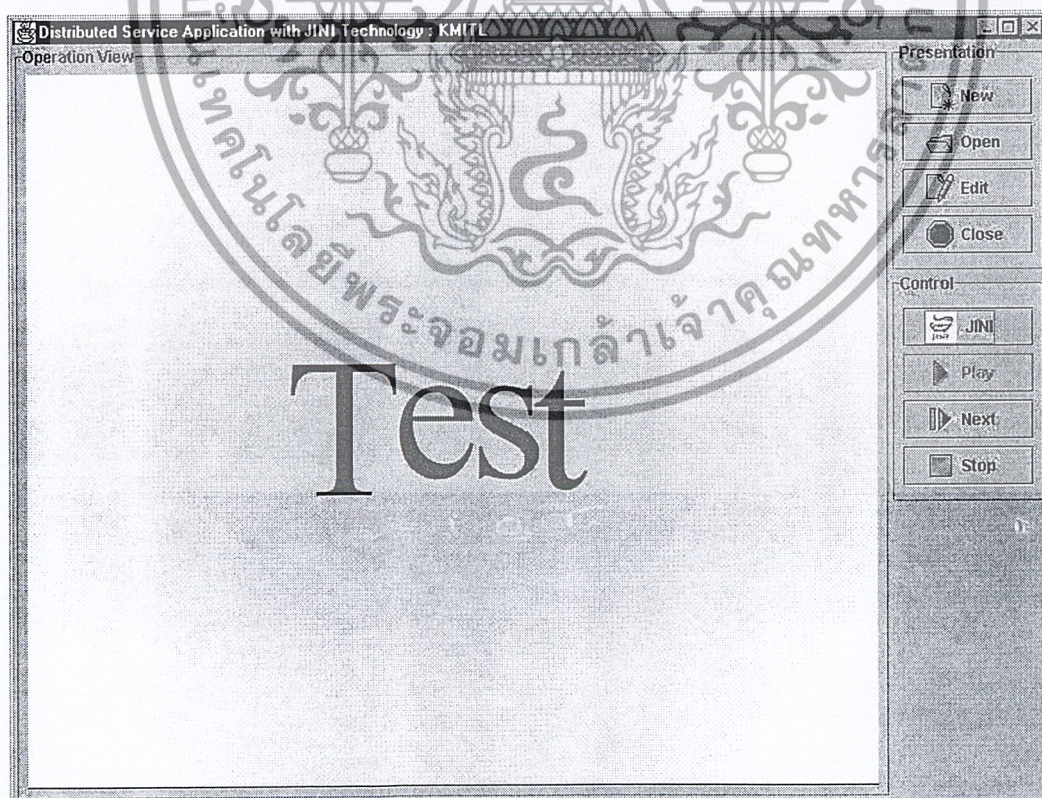
```

รูปที่ 5.8 แสดงตัวอย่างของ Hashtable

หลังจากทำการสร้างไฟล์นำเสนอแล้ว ผู้ใช้ก็สามารถเล่นไฟล์นั้นได้อย่างต่อเนื่องได้โดยการกดปุ่ม Play ซึ่งปุ่ม Play จะสามารถใช้ได้ก็ต่อเมื่อทำการเปิดไฟล์นำเสนอ ก่อน ผู้ใช้สามารถควบคุมการเล่นภาพนำเสนอได้โดยการกดปุ่ม Next เพื่อใช้เล่นภาพต่อไป หรือปุ่ม Stop เพื่อหยุดการเล่นภาพนำเสนอ ดังรูปที่ 5.9 และ 5.10 ตามลำดับ



รูปที่ 5.9 แสดงเฟรมภายในที่เกิดขึ้นเมื่อเลือกไฟล์ที่จะเล่น และปุ่ม Play ที่ตอบสนองขึ้นมา

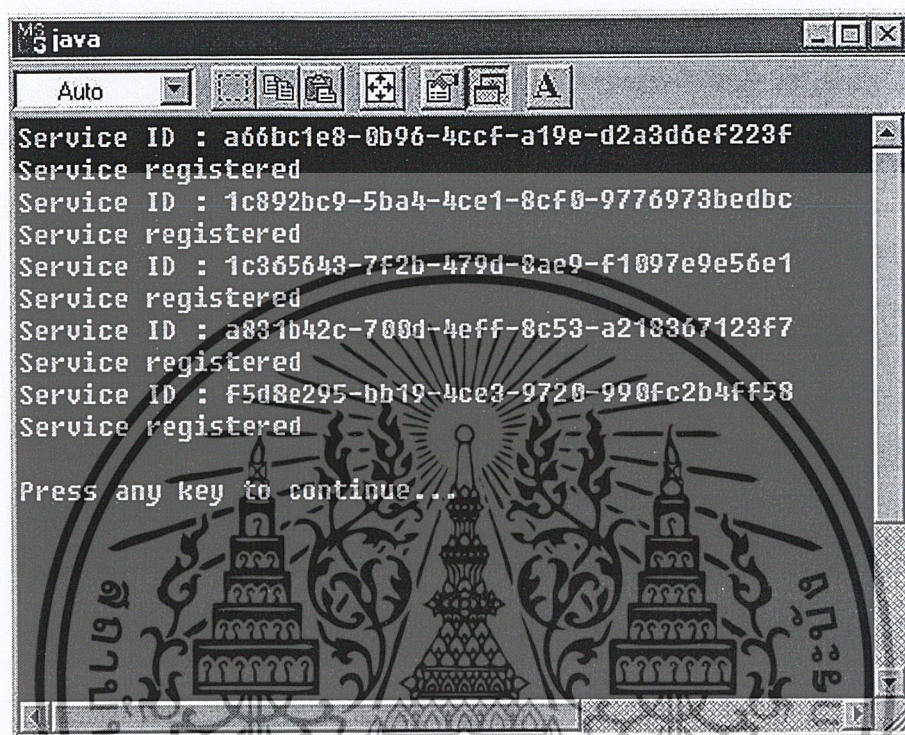


รูปที่ 5.10 แสดงภาพนำเสนอที่เริ่มแสดงภายในโอเพอร์เรชั่นวิวเมื่อกดปุ่ม Play

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

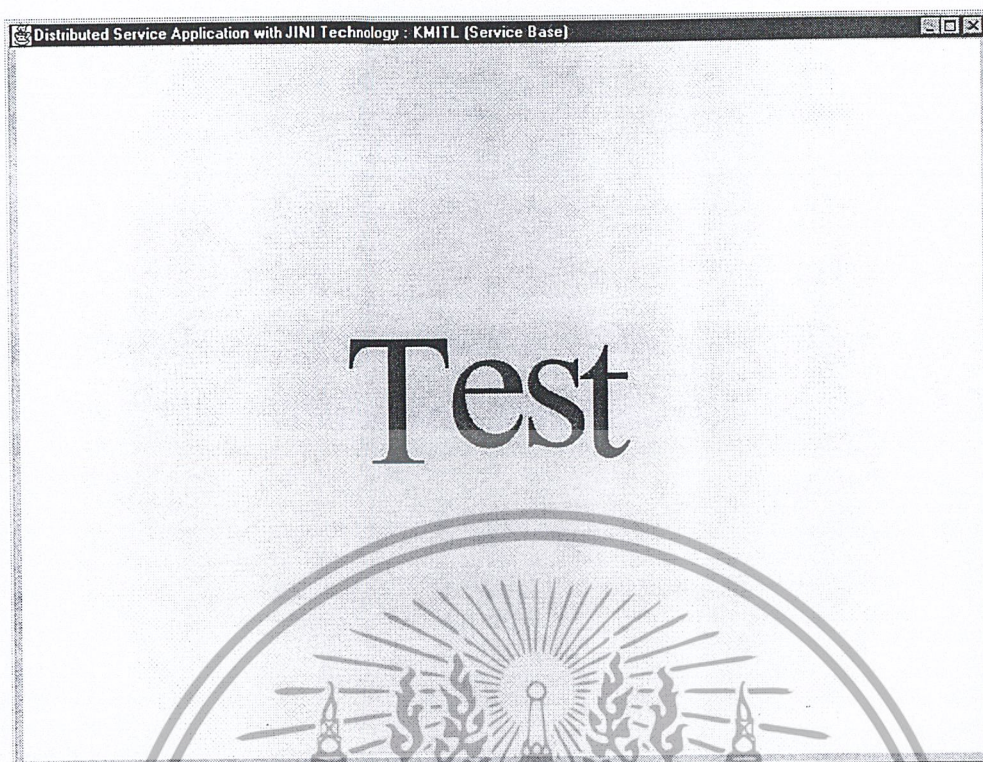
5.2 ผลการทดลองที่ส่วนเซอร์วิส

เซอร์วิสที่ให้บริการ จะต้องทำการลงทะเบียน (Register) กับ Lookup Locator ก่อน โดยเกิดผลแสดงดังรูป 5.11



รูปที่ 5.11 แสดงผลการลงทะเบียนของเซอร์วิส

หลังจากที่ผู้ใช้เลือกเซอร์วิสที่ต้องการ เปิดไฟล์นำเสนอ และทำการกดปุ่ม Play ที่ส่วนเซอร์วิส จะตอบสนองด้วยการแสดงเฟรมนำเสนอแบบเต็มจอภาพ เพื่อให้เครื่องฉายภาพแสดงภาพได้อย่างเต็มที่ โดยผลที่เกิดขึ้นจะมีลักษณะดังรูปที่ 5.12 การแสดงเฟรมนำเสนอนี้จะเกิดขึ้นได้เมื่อผู้ใช้เลือกเซอร์วิส แล้ว เปิดไฟล์นำเสนอ และกดปุ่ม Play เท่านั้น



รูปที่ 5.12 แสดงเฟรมแสดงผลที่ส่วนเซิร์ฟเวอร์

5.3 ผลการทดลองที่ส่วนเซิร์ฟเวอร์

ตามเทคโนโลยี JINI แล้วที่ส่วนเซิร์ฟเวอร์จะไม่แสดงผลใดๆ ให้เห็น โดยเมื่อตั้งค่าเริ่มต้นให้เป็นเซิร์ฟเวอร์แล้วก็สามารถใช้งานเป็น Service Locator ได้ทันที โดยทำการรันคำสั่งที่ DOS Prompt ดังนี้

5.3.1 ผลการรัน HTTP

`java -jar ./jini1_2/lib/tools.jar -port 8080 -dir /usr/java/jini1_2/lib/ &` จะเป็นการรันกระบวนการ HTTP ซึ่งเป็นการกำหนดการสื่อสารโดยใช้รูปแบบ HTTP โดยผ่านเทคโนโลยี Java RMI

5.3.2 ผลการรัน RMID

`rmid -J-Djava.security.policy=/usr/java/jini1_2/policy/policy.all &` จะเป็นการรัน RMID เพื่อให้สามารถใช้ Java RMI ได้

5.3.3 ผลการรัน reggie

`java -jar /usr/java/jini1_2/lib/reggie.jar http://ftp.ite.kmitl.ac.th:8080/reggie-dl.jar/usr/java/jini1_2/policy/policy.all tmp/reggie2_log public &` จะเป็นการรัน reggie ซึ่งเป็น Service Locator ที่ไว้ใช้เก็บเซิร์ฟเวอร์ โดย Service Locator นี้ถูก

กำหนดเงื่อนไข และจัดสร้างขึ้นโดย บริษัท Sun Microsystems

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปและวิจารณ์

6.1 วิเคราะห์ผลจากการออกแบบและสร้างระบบ

ผลของการสร้างระบบจำลองเครือข่ายของ JINI จากจุดประสงค์ของโครงการนี้ว่าด้วยการแสดงให้เห็นถึงความง่ายและคุณสมบัติต่าง ๆ ของ JINI แต่จากระบบที่สร้างขึ้น พบว่าการให้บริการเครื่องฉายภาพ หรือ เซอร์วิสของระบบนั้น ยังไม่สามารถสร้างเซอร์วิสได้ตามความสามารถทั้งหมดของ JINI ดังที่กล่าวได้ในเอกสารของ JINI เนื่องจากเครื่องมือที่ใช้ให้บริการเซอร์วิสนั้นยังไม่มีระบบประมวลผลโดยใช้จาวา ทำให้ไม่สามารถรองรับความสามารถของ JINI ทั้งหมดได้ ระบบที่สร้างขึ้นนั้น แสดงความสามารถของ JINI ได้ไม่เต็มที่ และยังคงมีจุดอ่อนในบางจุด เช่น การควบคุมเทรด (Thread) ส่วนของเทคโนโลยีในการลีสซิง (Leasing) ซึ่งเป็นการตรวจสอบการถอดออกของบริการ เป็นต้น

เซอร์วิสที่สร้างขึ้นนี้ไม่สามารถสร้างให้ใช้ความสามารถของ JINI ได้อย่างครบถ้วน แต่เลือกมาเพียงบางส่วนเพื่อให้เหมาะสมกับตัวเซอร์วิสเท่านั้น ซึ่งคุณสมบัติบางอย่างนั้น ก็ไม่มีความจำเป็นที่จะต้องใช้ และบางส่วนก็ไม่สามารถพัฒนาได้ทันตามกำหนด

การพัฒนาเซอร์วิสให้สามารถใช้งานได้ในสถานการณ์จริงนั้น ยังคงต้องรอให้เทคโนโลยีที่เกี่ยวข้อง เช่น การเพิ่มเติมระบบประมวลผลแบบจาวาในเครื่องมือที่จะให้เป็นเซอร์วิส เป็นต้น เพราะถ้าขาดเทคโนโลยีเหล่านี้ JINI ก็อาจไม่สามารถอำนวยความสะดวกให้ได้เต็มที่ และอาจจะทำให้ระบบนั้นยุ่งยากขึ้นด้วย

6.2 ปัญหาและอุปสรรค

สามารถสรุปปัญหาที่เกิดจากการทำโครงการนี้ได้ดังนี้

1. เนื่องจากเทคโนโลยี JINI เป็นเทคโนโลยีใหม่ แะยังมีข้อมูลมาก จึงต้องใช้เวลาในการศึกษาค่อนข้างมาก
2. การออกแบบด้วยหลักการ UML ค่อนข้างยุ่งยาก และมีขั้นตอนมากมาย ผู้ออกแบบไม่สามารถออกแบบได้อย่างครอบคลุม ทำให้ขั้นตอนการเขียนโปรแกรมผิดเพี้ยนไปจากการออกแบบ และการกลับไปปรับปรุงส่วนที่ออกแบบจะทำได้ยาก
3. ตัวโปรแกรมนำเสนอเน้นใช้การสร้างภาพด้วยจาวาทูดี (Java2D) ซึ่งเป็นระบบใหม่เช่นกัน มีข้อมูลให้ศึกษามากมาย จึงต้องใช้เวลาในการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การควบคุมการนำเสนอ นั้นสามารถทำได้ยากเนื่องจากตอนออกแบบนั้นใช้ Thread (Thread) ในการควบคุมให้ภาพนั้นเล่นไปอย่างต่อเนื่อง ซึ่งการควบคุม Thread นั้นทำได้ยากในภาษาจาวา ทำให้เกิดปัญหาคือไม่สามารถเล่นภาพย้อนกลับได้
5. การประมวลผลด้วยภาษาจาวานั้นค่อนข้างช้า ทำให้ภาพนำเสนอที่แสดงจะขึ้นอยู่กับสมรรถนะของเครื่องคอมพิวเตอร์ หากเครื่องคอมพิวเตอร์เครื่องนั้นมีสมรรถนะเครื่องที่ดี ภาพที่ได้ก็จะไม่เกิดการกระตุก หรือติดขัด

6.3 ข้อดีและข้อเสียของระบบ

ข้อดี

- ระบบนี้ออกแบบด้วยหลักการเขียนโปรแกรมแบบอ้างอิงวัตถุ ทำให้สามารถนำส่วนต่าง ๆ มาใช้ใหม่ (Reusable) ได้ง่าย และได้ระบบที่มีประสิทธิภาพ
- เป็นระบบประมวลผลแบบกระจาย ออบเจกต์ต่าง ๆ สามารถทำงานร่วมกันได้ ถึงแม้ว่าจะอยู่ต่างแพลตฟอร์มและต่างสถานที่
- มีการติดต่อกันโดยอัตโนมัติระหว่างผู้ใช้และเซิร์ฟเวอร์ โดยที่ผู้ใช้ไม่จำเป็นต้องรู้ตำแหน่งที่อยู่ของเซิร์ฟเวอร์มาก่อน
- ผู้ใช้ไม่จำเป็นต้องรู้จักตัวเซิร์ฟเวอร์มาก่อน เนื่องจากผู้ใช้ติดต่อกับเซิร์ฟเวอร์ผ่านทางส่วนติดต่อกับผู้ใช้ จึงไม่จำเป็นต้องรู้ถึงรายละเอียดวิธีการสร้างเซิร์ฟเวอร์
- ง่ายต่อผู้ใช้ เนื่องจากมีส่วนติดต่อกับผู้ใช้ที่ค่อนข้างชัดเจน ในการที่มีเซิร์ฟเวอร์อยู่หลายรูปแบบ และผู้ใช้ไม่จำเป็นต้องเปลี่ยนหรือปรับแต่งระบบของตนเอง ในกรณีที่มีเซิร์ฟเวอร์ใหม่เกิดขึ้น
- การใช้ภาษาจาวาในการเขียนโปรแกรม มีข้อมูลให้ศึกษาค่อนข้างมาก และมีไลบรารี (Library) ให้เลือกใช้เข้ากับระบบอย่างมากมาย

ข้อเสีย

- ทำงานได้ช้าหากเปรียบเทียบกับระบบแบบเดียวกันที่สร้างจากภาษาอื่นเช่น ภาษาซี (C Language)
- เทคโนโลยีถูกจำกัดด้วยภาษาจาวา การจะพัฒนาระบบของ JINI ด้วยภาษาอื่นนั้นสามารถทำได้ แต่ว่ายากกว่า
- การออกแบบระบบด้วยหลักการเขียนโปรแกรมแบบอ้างอิงวัตถุ ซึ่งทำได้ยาก หากออกแบบไม่ครอบคลุมการใช้ จะทำให้ได้ผลผิดเพี้ยน และแก้ไขส่วนที่ออกแบบได้ยาก
- เทคโนโลยีบางอย่างควบคุมได้ยากเช่น การควบคุม Thread (Thread)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 แนวทางในการพัฒนาต่อไป

1. ส่วนติดต่อกับผู้ใช้สามารถพัฒนาให้ครอบคลุมการใช้งานมากยิ่งขึ้น เช่น การส่งเสียงภาพเข้าไปด้วยได้ หรือ การจัดวางรูปแบบข้อมูลนำเสนอแบบใช้กราฟฟิค ยูสเซอร์ อินเตอร์เฟส (Graphic User Interface : GUI)
2. สามารถพัฒนาระบบการสื่อสารระหว่างอุปกรณ์ กับเซิร์ฟเวอร์ให้ดียิ่งขึ้น เช่น ใช้การสื่อสารแบบไร้สายด้วยคลื่นวิทยุ
3. สามารถพัฒนาให้เทคโนโลยี JINI มีความสมบูรณ์มากขึ้น เช่น มีระบบลิขสิทธิ์ ทำให้รู้ว่าเซอร์วิสมีสถานะเป็นอย่างไร เป็นต้น
4. อาจเพิ่มเซอร์วิสที่จะให้บริการนอกจากเครื่องฉายภาพเพียงอย่างเดียว อาจให้มีเซอร์วิสในการพิมพ์ผ่านทางพรินเตอร์ได้ด้วย

6.5 สรุป

ตัวอย่างของระบบที่สร้างขึ้นนั้นก็มີประสิทธิภาพในระดับหนึ่ง แสดงให้เห็นประโยชน์ของเทคโนโลยี JINI ที่สามารถนำไปพัฒนาต่อให้เกิดความสมบูรณ์มากขึ้น หรือพัฒนาใช้กับระบบอื่นได้ ขึ้นอยู่กับผู้พัฒนาระบบว่าจะนำไปประยุกต์กับสิ่งใดเพื่อให้เกิดประโยชน์สูงสุด

ถึงแม้ว่าเทคโนโลยี JINI จะเป็นเทคโนโลยีใหม่ แต่ถ้าหากดูถึงรายละเอียดภายในแล้ว เทคโนโลยีที่ใช้ก็เป็นแบบพื้นฐาน มิได้มีความแปลกใหม่เกินไป หากศึกษาอย่างถี่ถ้วน ออกแบบระบบที่ต้องการอย่างครอบคลุม ก็อาจนำเทคโนโลยี JINI มาใช้ได้อย่างสมบูรณ์ เกิดประโยชน์ต่อชีวิตประจำวัน และอาจนำไปพัฒนาในเชิงพาณิชย์ได้

บรรณานุกรม

เอกสารอ้างอิง

1. Joseph Schmuller, UML in 24 Hours, G. C. Jain for Techmedia, 1999
2. ดร.วีระศักดิ์ ชิงदार, JAVA Programming Volume 1, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), กรุงเทพฯ 2543
3. ดร.วีระศักดิ์ ชิงदार, JAVA Programming Volume 2, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), กรุงเทพฯ 2543
4. H.M. Deitel & P.J. Deitel, JAVA™ How To Program, Third Edition, Prentice-Hall, Inc. 1999
5. อ.รุ่งโรจน์ โพนคำและ อ. ปราณี มณีรัตน์, Advanced Java Programming ฉบับสร้างแอปพลิเคชันด้วยจาวาสวิงซ์, บริษัท ซัคเซส มีเดีย จำกัด 2537

เว็บไซต์อ้างอิง

1. <http://java.sun.com/>
2. <http://java.sun.com/j2se/1.3/>
3. <http://java.sun.com/products/java-media/2d/index.html/>
4. <http://www.sun.com/jini/>
5. <http://www.sun.com/forte/ffj/>
6. <http://www.cswl.com/whiteppr/tutorials/jini.html>

ภาคผนวก

คู่มือการใช้งาน

1. ติดตั้ง JAVA Development Kit หรือ JDK บนเครื่องไคลเอ็นท์, เซิร์ฟเวอร์, เซอร์วิส

JDK เป็นเครื่องมือสำหรับการพัฒนาภาษาจาวาของบริษัทซัน ไมโครซิสเต็ม ซึ่งใช้เป็นเครื่องมือหลักในการพัฒนาโครงการนี้ JDK มี 2 ประเภท

- 1.1 JAVA™ 2 SDK, Standard Edition, เวอร์ชัน 1.3.1_01 ในชุดโปรแกรมประกอบด้วย 2 ส่วน คือ ส่วน Development System ที่ใช้สำหรับการแปลภาษาจาวา และส่วนของ Runtime Environment ที่ใช้ในการทำงานกับโปรแกรมจาวา

- 1.2 JAVA™ 2 Runtime Environment, เวอร์ชัน 1.3.1_01 จะต่างกับชุด Standard Edition ตรงที่มีเฉพาะส่วนของ Runtime Environment เท่านั้น ทำให้ตัวโปรแกรมติดตั้งนี้มีขนาดเล็กกว่ามาก

ตัวโปรแกรม JDK เวอร์ชัน 1.3.1_01 นั้นสามารถดาวน์โหลดได้จากเว็บไซต์ <http://java.sun.com> ส่วนการติดตั้งจะไม่แตกต่างจากการติดตั้งโปรแกรมทั่วในระบบวินโดวส์ เพราะ JDK มีโปรแกรมติดตั้งที่ทำงานอยู่บนวินโดวส์แล้ว ข้อควรระวัง คือ หากมีการติดตั้ง JDK นี้ในเวอร์ชันอื่นๆ ควรทำการถอนโปรแกรมเวอร์ชันนั้น ๆ ออกจากระบบให้เรียบร้อยก่อนที่จะทำการติดตั้งเวอร์ชันที่ใหม่กว่าลงไป

2. การติดตั้ง Path ของระบบ

เนื่องด้วย JDK เป็นโปรแกรมพัฒนาที่ทำงานบนดอส (DOS) เป็นหลัก ดังนั้นการเรียกใช้โปรแกรมจึงต้องผ่านตัวรับคำสั่ง (Command Line) การปรับแต่งตัวแปร Path จะสร้างความสะดวกให้กับผู้ใช้ให้สามารถรันโปรแกรมจาวาได้

- 2.1 ระบบวินโดวส์ 95/98 สามารถปรับแต่งตัวแปร Path ได้โดยเข้าไปที่ MS-DOS Prompt โดยให้อยู่ที่ C:\ แล้วพิมพ์ edit autoexec.bat จะเกิดหน้าต่างเพื่อทำการแก้ไข ให้พิมพ์เพิ่มลงไปด้วยดังนี้

PATH = %PATH%;C:\jdk1.3.1_01\bin

แล้วทำการเซฟไฟล์ไว้ ก่อนการใช้งานจะต้องทำการเปิดเครื่องใหม่ก่อน

- 2.2 ระบบวินโดวส์เอ็นที (Window NT) จะมีลักษณะการปรับแต่งโดยใช้ค่าเช่นเดียวกัน แต่การแก้ไขให้ใช้ System Properties ซึ่งอยู่ใน Control Panel => System เลือก Environment

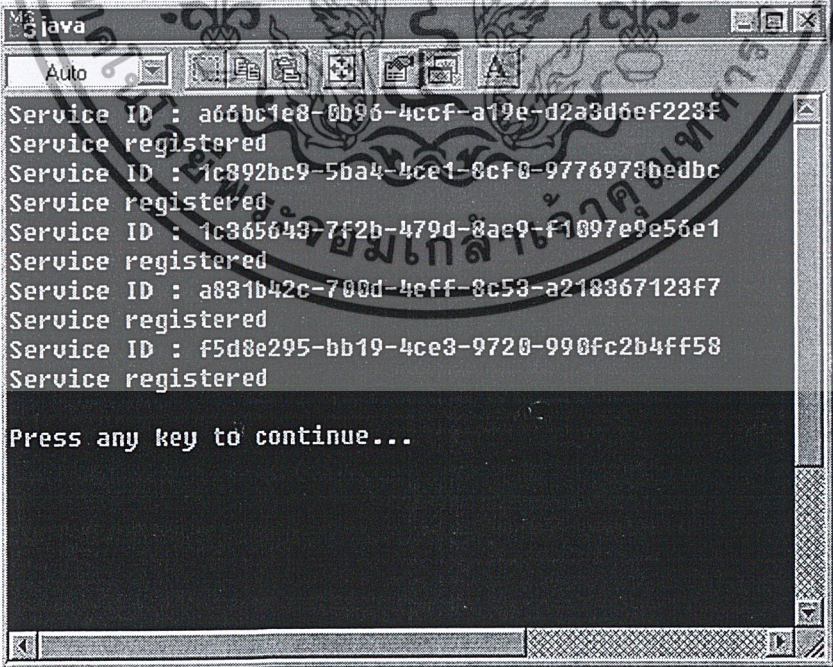
3. ทำการติดตั้งเซิร์ฟเวอร์ หรือ Service Locator โดยเครื่องที่จะเป็นเซิร์ฟเวอร์นั้นจะต้องทำการดาวน์โหลดโปรแกรม JINI จาก <http://www.sun.com/jini/> และทำการขยายไฟล์ที่ออกมาจากนั้นที่ DOS Prompt ทำการรันคำสั่งดังนี้

3.1 `java -jar ./jini1_2/lib/tools.jar -port 8080 -dir /usr/java/jini1_2/lib/ &` จะเป็นการรันกระบวนการ HTTP ซึ่งเป็นการกำหนดการสื่อสารโดยใช้รูปแบบ HTTP โดยผ่านเทคโนโลยี Java RMI

3.2 `rmid -J-Djava.security.policy=/usr/java/jini1_2/policy/policy.all &` จะเป็นการรัน RMID เพื่อให้สามารถใช้ Java RMI ได้

3.3 `java -jar /usr/java/jini1_2/lib/reggie.jar http://ftp.ite.kmitl.ac.th:8080/ reggie-dl.jar/usr/java/jini1_2/policy/policy.all tmp/reggie2_log public &` จะเป็นการรัน reggie ซึ่งเป็น Service Locator ที่ไว้ใช้เก็บเซอร์วิส โดย Service Locator นี้ถูกกำหนดเงื่อนไข และจัดสร้างขึ้นโดยบริษัท Sun Microsystems
ขั้นตอนนี้จะไม่แสดงผลใดๆให้เห็น

4. ที่ส่วนเซอร์วิสจะต้องทำการลงทะเบียนกับเซิร์ฟเวอร์ซึ่งตามทฤษฎีแล้ว เทคโนโลยี JINI จะจัดการให้โดยอัตโนมัติ หากเครื่องเซอร์วิสมีการประมวลผลโดยใช้ JINI แต่ในปัจจุบันยังไม่มีเซอร์วิสใดที่บรรจส่วนประมวลผลเช่นนั้น ฉะนั้นจะต้องทำการรันเองโดยทำการรันโปรแกรม `helpServiceToRegister.java` ซึ่งจะเกิดผลดังรูปที่ 1



```

java
Auto
Service ID : a66bc1e8-0b96-4ccf-a19e-d2a3d6ef223f
Service registered
Service ID : 1c892bc9-5ba4-4ce1-8cf8-9776973bedbc
Service registered
Service ID : 1c365643-7f2b-479d-8ae9-f1097e9e56e1
Service registered
Service ID : a831b42c-700d-4eff-8c53-a218367123f7
Service registered
Service ID : f5d8e295-bb19-4ce3-9720-990fc2b4ff58
Service registered

Press any key to continue...
  
```

รูปที่ ผ.1 แสดงผลที่เกิดขึ้นเมื่อเซอร์วิสลงทะเบียนกับเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

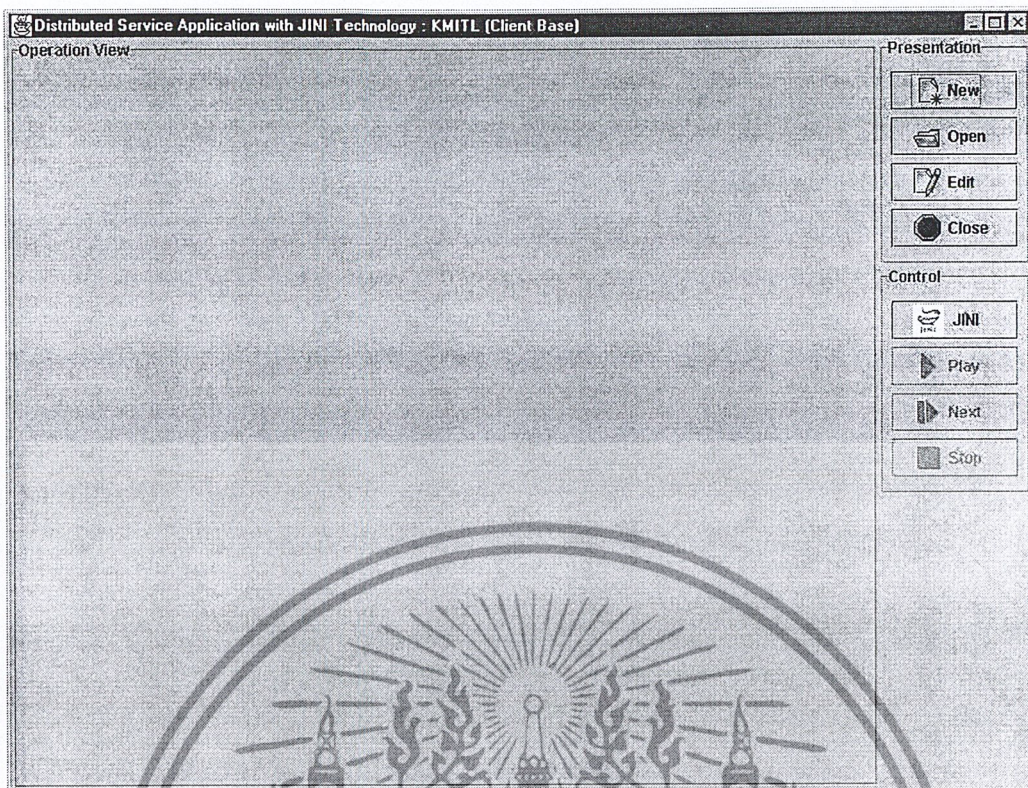
5. ข้อมูลเบื้องต้นของโปรแกรมนำเสนอ และการคอมไพล์ (Compile)

ไฟล์ทั้งหมดของโปรแกรมจะมีดังต่อไปนี้

- | | |
|--------------------|------------------------------|
| - MainWindow.java | - Buffer.java |
| - NewPresent.java | - UnicastRegistrar.java |
| - OpenPresent.java | - Servicewindow.java |
| - EditPresent.java | - ServiceRegister.java |
| - Presentaion.java | - Service.java |
| - Template.java | - Server.java |
| - Template1.java | - MulticastRegister.java |
| - Template2.java | - helpServiceToRegister.java |
| - Template3.java | - Discovery.java |
| - Template4.java | - Client.java |
| - BackGround.java | - Preview.java |
| - PageChange.java | - PreviewFrame.java |
| - Filter.java | - DemoImages.java |

ตรวจสอบดูว่ามีไฟล์ครบหรือไม่ จากนั้นทำการคอมไพล์ไฟล์ทุกตัวโดยใช้คำสั่ง `javac` ซึ่งเป็น JAVA Compiler ใน JDK เช่นหากต้องการคอมไพล์ไฟล์ `MainWindow.java` ให้พิมพ์ที่ DOS Prompt ว่า `javac MainWindow.java` โดยที่ Prompt ปัจจุบันที่อยู่จะต้องอยู่ที่เดียวกับที่ไฟล์ที่จะคอมไพล์นั้นเซฟไว้ ซึ่งหากคอมไพล์ผ่านก็จะเกิดไฟล์ `.class` ขึ้น

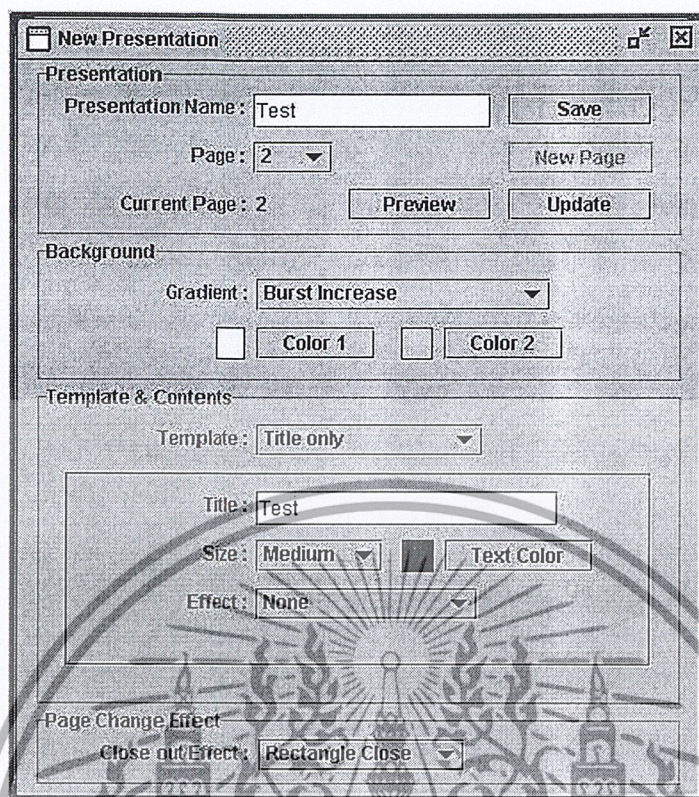
6. ทำการรันโปรแกรม MainWindow.java โดยใช้คำสั่ง `java` มีลักษณะเช่นเดียวกับการคอมไพล์ แต่เปลี่ยนจากคำสั่ง `javac` เป็น `java` ตามด้วยชื่อของไฟล์ที่จะรันโดยไม่ต้องเติมนามสกุลของไฟล์ เช่น `java MainWindow` จะเกิดหน้าจอหลักของโปรแกรมนำเสนอขึ้นดังรูปที่ 2



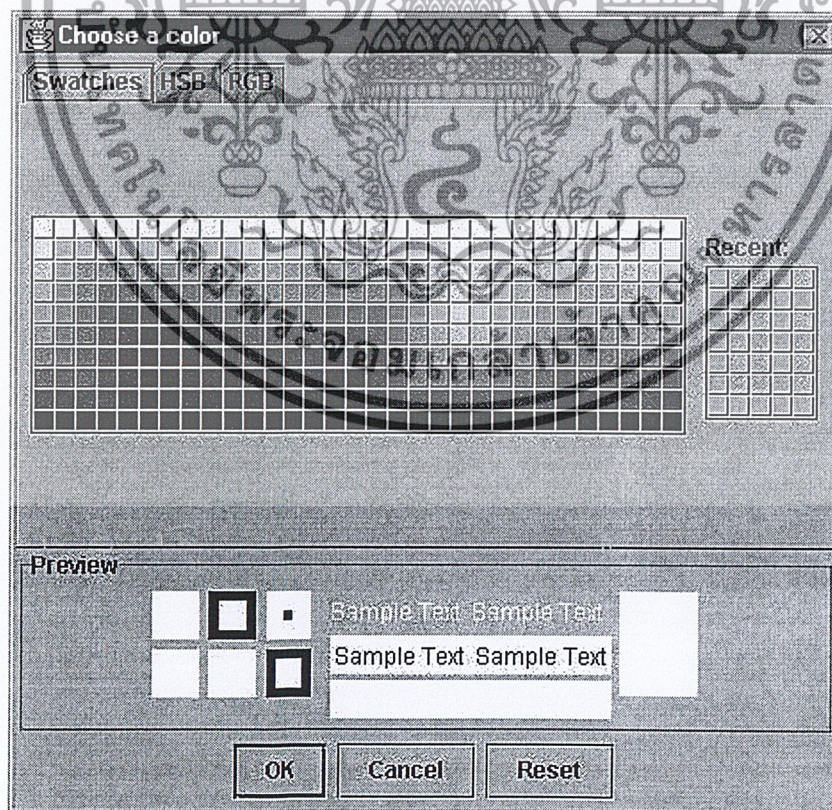
รูปที่ ผ.2 แสดงหน้าจอหลักของโปรแกรมนำเสนอ

7. การสร้างไฟล์นำเสนอสามารถทำได้โดยการกดปุ่ม New ซึ่งจะเกิดเฟรมภายในขึ้นที่โอเปอร์เรชั่นวิว แล้วทำการกดปุ่ม New Page ที่อยู่บนเฟรมภายในนั้น จะปรากฏ ComboBox ให้เลือกรูปแบบของไฟล์นำเสนอซึ่งรูปที่ 3 โดยส่วนต่างๆ ในหน้า New Presentation นั้นอธิบายได้ดังนี้
- Presentation Name เป็นส่วนที่ใส่เดิมชื่อไฟล์ที่จะเซฟ
 - Page จะเป็นที่เกี่ยวข้องเลขหน้าทั้งหมด โดยสามารถเลือกดูหน้าที่ต้องการได้
 - Current Page จะบอกสถานะว่าอยู่ที่หน้าปัจจุบันใด
 - ปุ่ม Save มีไว้สำหรับเซฟข้อมูลที่ได้ทำการกรอกลงไป
 - ปุ่ม New Page มีไว้สำหรับสร้างหน้าใหม่ของไฟล์นำเสนอ
 - ปุ่ม Update มีไว้สำหรับเซฟข้อมูลหน้านั้นๆ
 - ปุ่ม Preview มีไว้สำหรับดูตัวอย่างของข้อมูลที่ใส่ลงไป
 - Gradient จะเป็นที่เกี่ยวข้องรูปแบบของฉากหลัง
 - Color 1, Color 2 จะเป็นสีของฉากหลังโดยหากเลือกฉากหลังเป็น None ก็จะไม่มีการไล่ระดับสีเป็น Color 2 นอกเหนือจากนั้นจะมีการไล่ระดับสีทั้งหมด จาก Color1 ไป Color2 โดยสามารถเลือกรูปแบบสีได้ตามที่ต้องการดังรูปที่ 4 หากไม่ทำการเลือกจะตั้งเป็นสีขาวไว้
 - Close Out Effect สามารถเลือกรูปแบบการเปลี่ยนหน้านำเสนอได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



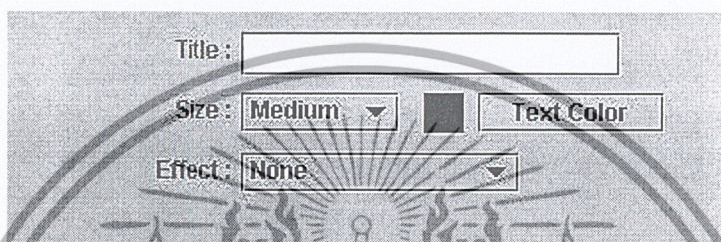
รูปที่ ผ.3 แสดงเฟรมภายในของการ New Presentation



รูปที่ ผ.4 แสดงการเลือกสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

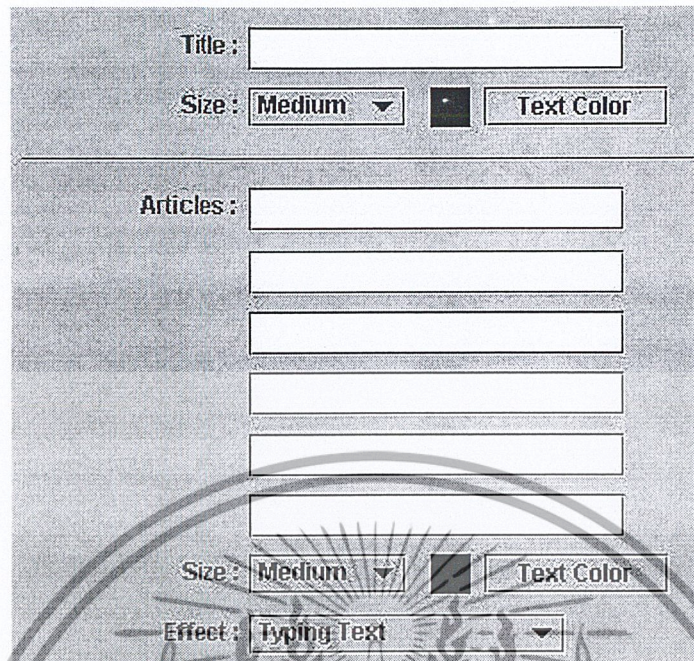
- ส่วน Template จะมีรูปแบบให้ใช้งาน 4 รูปแบบดังนี้
แบบที่ 1 Title Only (รูปที่ 5)
 - สามารถเติมตัวอักษรที่จะแสดงได้ที่ Title
 - สามารถเลือกขนาดที่ Size ได้ ซึ่งมีหลายขนาด
 - ที่ปุ่ม Text Color สามารถเลือกสีของตัวอักษรได้เช่นเดียวกับการเลือกสีจากหลังดังรูปที่ 4 หากไม่ทำการเลือกจะตั้งเป็นสีดำไว้
 - ตรงส่วน Effect สามารถเลือกรูปแบบของตัวอักษรที่จะแสดงออกมาได้



รูปที่ ๕.5 แสดงรูปแบบการนำเสนอแบบที่ 1

แบบที่ 2 Title & Articles (รูปที่ 6)

- สามารถเติมตัวอักษรที่จะแสดงได้ที่ Title
- สามารถเลือกขนาดที่ Size ได้ ซึ่งมีหลายขนาด
- ที่ปุ่ม Text Color สามารถเลือกสีของตัวอักษรได้เช่นเดียวกับการเลือกสีจากหลังดังรูปที่ 4 หากไม่ทำการเลือกจะตั้งเป็นสีดำไว้
- ส่วน Articles จะมีไว้สำหรับการใส่หัวข้อต่าง ๆ ตามที่ต้องการสามารถเติมได้ 6 หัวข้อ และสามารถเลือกขนาด สีได้เช่นเดียวกับ Title
- ตรงส่วน Effect สามารถเลือกรูปแบบของตัวอักษรที่จะแสดงออกมาได้



รูปที่ ๗.6 แสดงรูปแบบการนำเสนอแบบที่ 2

แบบที่ 3 Title & List (FontFixed) รูปที่ 7

- สามารถเติมตัวอักษรที่จะแสดงได้ที่ Title
- ส่วน List จะมีไว้สำหรับการใส่รายการต่าง ๆ ตามที่ต้องการสามารถเติมได้ 10 รายการ โดยจะเลื่อนขึ้นมาจากส่วน Title ก่อน
- ที่ปุ่ม Text Color สามารถเลือกสีของตัวอักษรได้เช่นเดียวกับการเลือกสีจากหลังดังรูปที่ 4 หากไม่ทำการเลือกจะตั้งเป็นสีดำไว้
- ตรงส่วน Effect สามารถเลือกรูปแบบของตัวอักษรที่จะแสดงออกมาได้

Title:

List:

-
-
-
-
-
-
-
-
-
-

Effect: Slide up only ▾

รูปที่ ผ.7 แสดงรูปแบบการนำเสนอแบบที่ 3

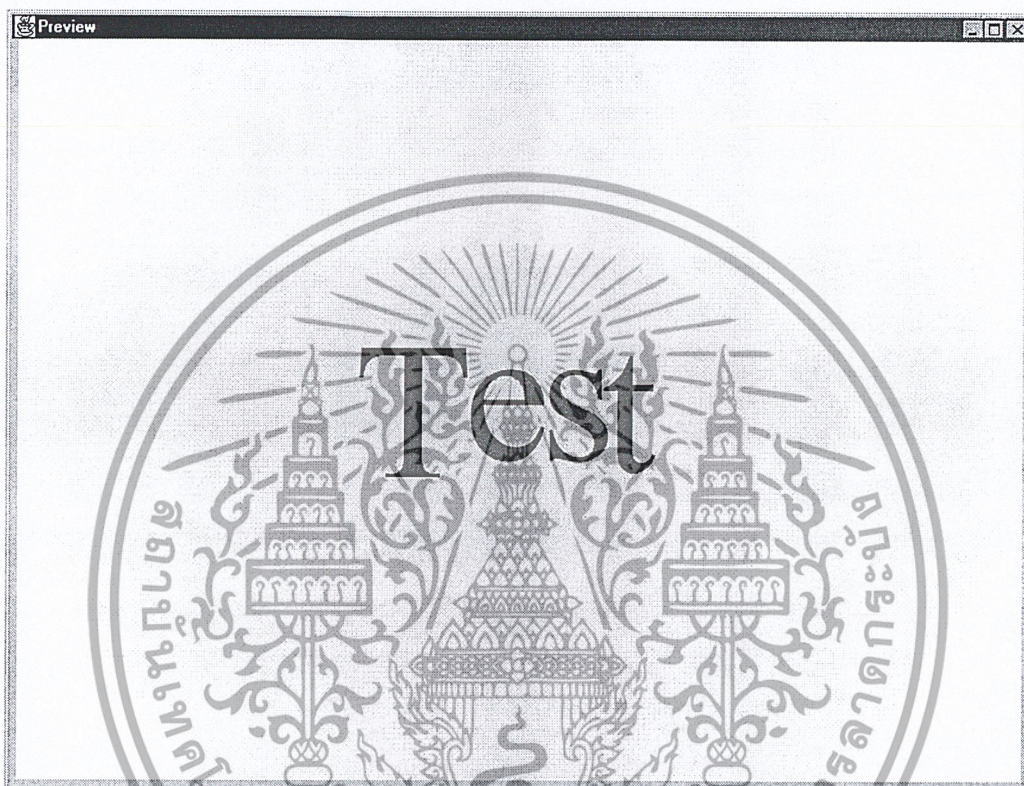
แบบที่ 4 Title & Picture (รูปที่ 8)

- สามารถเติมตัวอักษรที่จะแสดงได้ที่ Title
- สามารถเลือกขนาดที่ Size ได้ ซึ่งมีหลายขนาด
- ที่ปุ่ม Text Color สามารถเลือกสีของตัวอักษรได้เช่นเดียวกับการเลือกสีฉากหลังดังรูปที่ 4 หากไม่ทำการเลือกจะตั้งเป็นสีดำไว้
- ที่ Picture สามารถเลือกรูปที่จะนำมาแสดงได้
- ตรงส่วน Effect สามารถเลือกรูปแบบของตัวอักษรที่จะแสดงออกมาได้

Title :			
Size :	Medium	☐	Text Color
Picture :		Browse	
Effect :	No Rectangle		

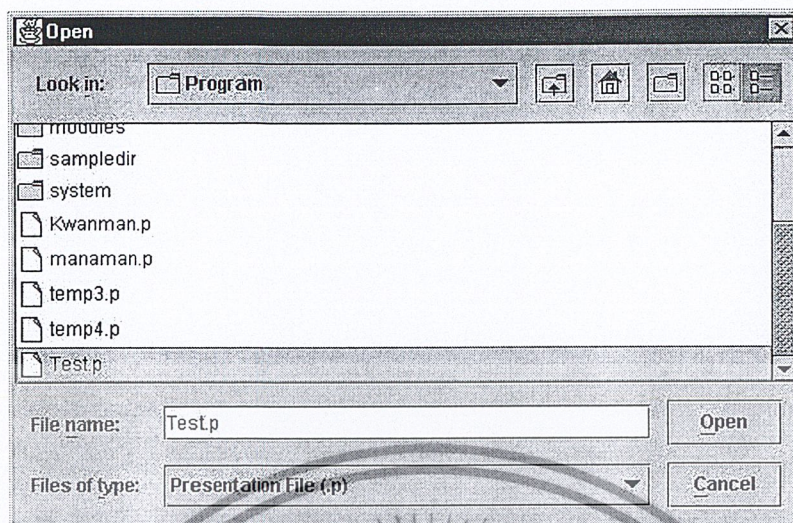
รูปที่ ผ.8 แสดงรูปแบบการนำเสนอแบบที่ 4

8. ลำดับการสร้างไฟล์นำเสนอจะเริ่มจาก การกดปุ่ม New Page => เลือก จากหลัง, Template, รูปแบบการเปลี่ยนภาพ => กรอกข้อมูลต่างๆ => กดปุ่ม Update หากต้องการหน้าถัดไปก็ปุ่ม New Page ทำตามขั้นตอนเช่นเดิม โดยในแต่ละหน้าสามารถดูตัวอย่างได้ โดยกดปุ่ม Preview ซึ่งจะขึ้นเป็นเฟรมดังรูปที่ 9 โดยการดูตัวอย่างนี้สามารถทำได้ในทุกเฟรมภายในของการ New, Open, Edit



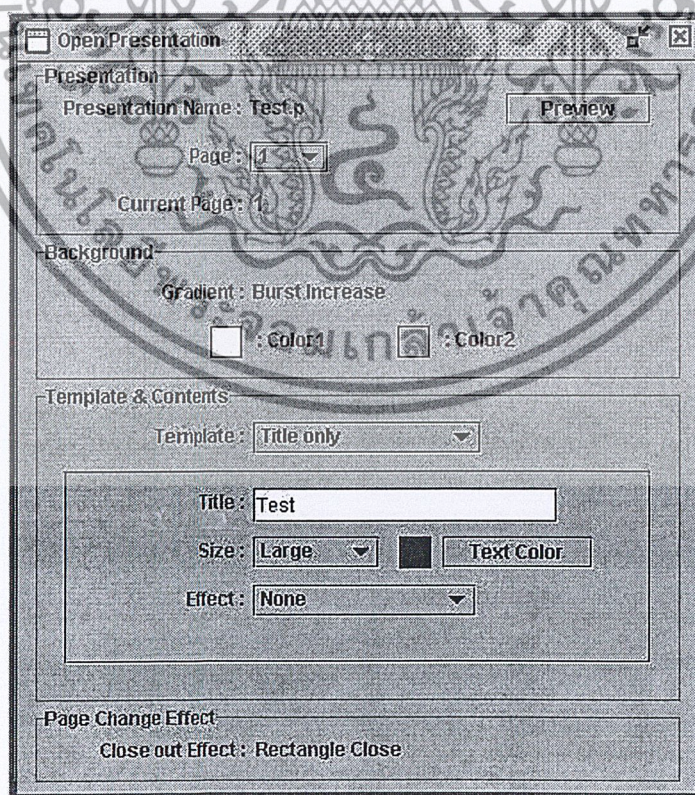
รูปที่ ๙.๙ แสดงเฟรมรูปตัวอย่างที่เกิดขึ้นเมื่อกดปุ่ม Preview

9. จากหน้าจอหลักที่ปุ่ม Open ใช้ในการเปิดไฟล์นำเสนอที่สร้างไว้แล้ว โดยเมื่อกดที่ปุ่มนี้จะเกิดเฟรมที่ใช้สำหรับเลือกไฟล์ดังรูปที่ 10 และเพื่อความสะดวกในการค้นหาไฟล์ที่ต้องการสามารถเลือกชนิดของไฟล์ทั้งหมดเป็นไฟล์ .p ซึ่งเป็นไฟล์นำเสนอ ตรงส่วน Files of type ด้านล่างของเฟรม เมื่อเลือกแล้วก็กดปุ่ม Open



รูปที่ ผ.10 แสดงเฟรมเลือกไฟล์นำเสนอเมื่อกดปุ่ม Open และปุ่ม Edit

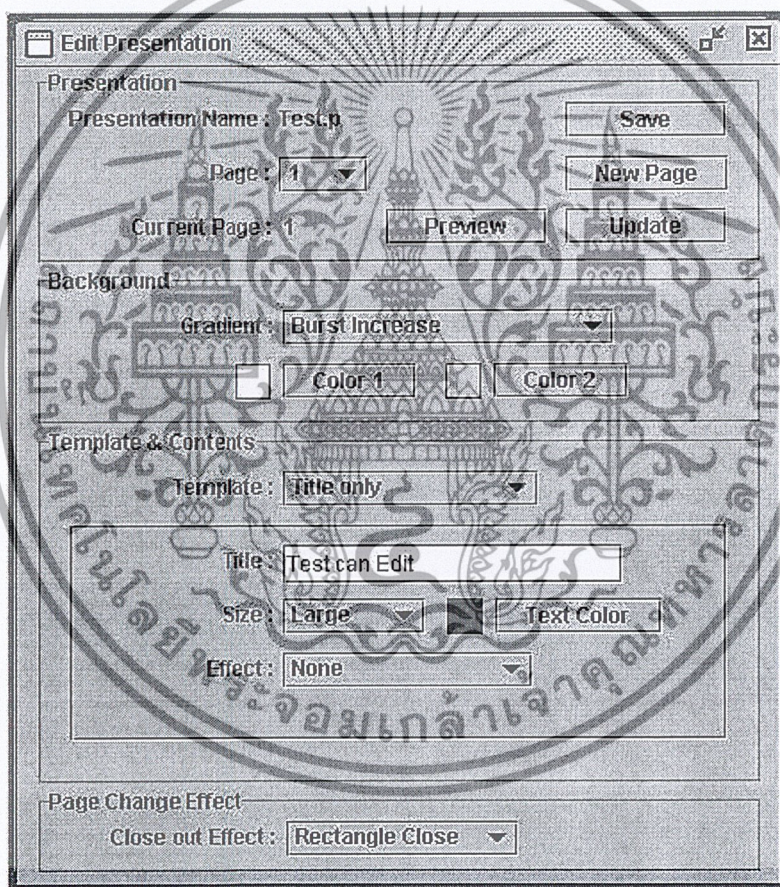
10. เมื่อเลือกไฟล์แล้วที่โอเพอร์เรชันวิว ก็จะเกิดเฟรมภายในของการ Open Presentation ขึ้น โดยส่วนต่างๆ ภายในเฟรมนั้นจะมีลักษณะคล้ายกับการ New Presentation แต่จะไม่สามารถแก้ไขใดๆ ได้ ซึ่งส่วน Open Presentation นี้ไว้ใช้สำหรับเปิดไฟล์, ทำการตรวจสอบก่อนการเล่นไฟล์นำเสนอ และสามารถเลือกหน้าที่จะ Preview ได้ดังรูปที่ 11



รูปที่ ผ.11 แสดงเฟรมภายในของการ Open Presentation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

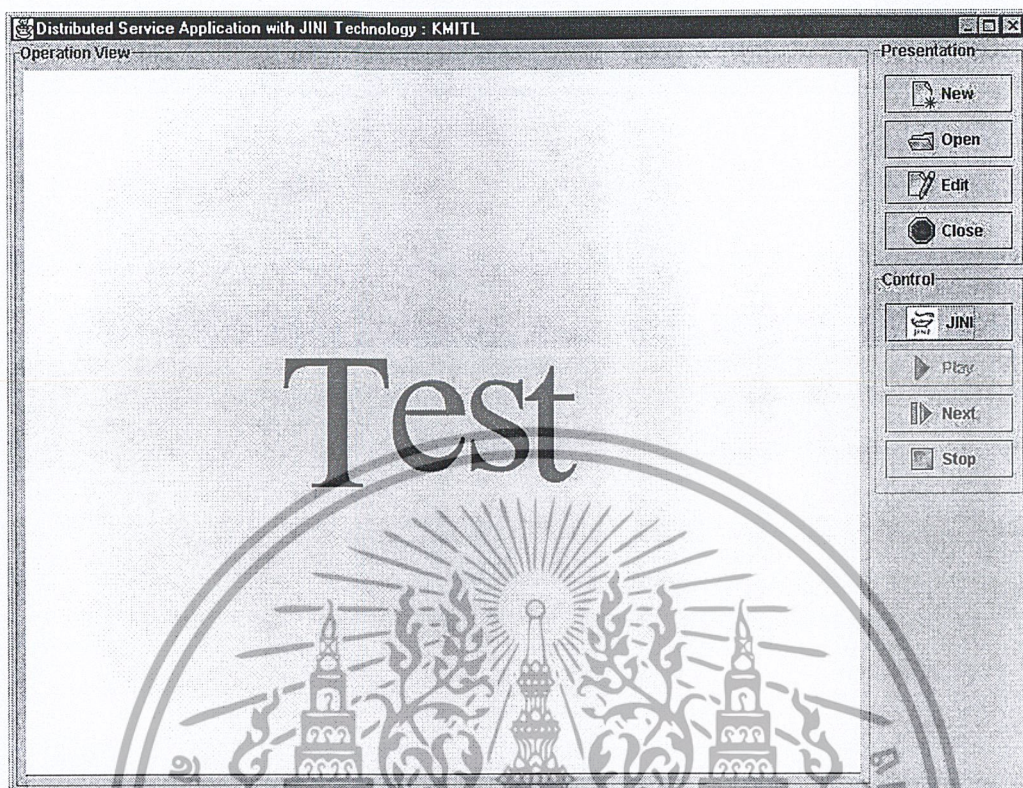
11. จากหน้าจอหลักที่ปุ่ม Edit ใช้ในการแก้ไขไฟล์นำเสนอที่สร้างไว้แล้วโดยเมื่อกดที่ปุ่มนี้จะเกิดเฟรมที่ใช้สำหรับเลือกไฟล์ดังรูปที่ 9 เช่นเดียวกันกับการ Open Presentation และเพื่อความสะดวกในการค้นหาไฟล์ที่ต้องการสามารถเลือกชนิดของไฟล์ทั้งหมดเป็นไฟล์ .p ซึ่งเป็นไฟล์นำเสนอ ตรงส่วน Files of type ด้านล่างของเฟรม เมื่อเลือกแล้วก็กดปุ่ม Open
12. เมื่อเลือกไฟล์แล้วที่โอเพอร์เรชันวิว ก็จะเกิดเฟรมภายในของการ Edit Presentation ขึ้น ดังรูปที่ 12 โดยส่วนต่าง ๆ ภายในเฟรมนั้นจะมีลักษณะคล้ายกับการ New Presentation โดยสามารถแก้ไขสิ่งต่าง ๆ ได้ โดยเลือกหน้าที่ต้องการจะแก้ไขจาก Page เมื่อแก้ไขข้อมูลต่าง ๆ แล้วต้องกดปุ่ม Update ทุกครั้ง หลังจากแก้ไขครบทุกหน้าแล้วก็สามารถเซฟไฟล์นั้น ๆ ได้โดยกดปุ่ม Save เฟรมภายใน Edit ก็จะหายไป



รูปที่ ผ.12 แสดงเฟรมภายในของการ Edit Presentation

13. ก่อนเล่นไฟล์นำเสนอจะต้องทำการกดปุ่ม JINI เพื่อเลือกเซอร์วิสที่จะเรียกใช้ โดยจะขึ้นเฟรมของรายการเซอร์วิสที่ใช้ได้ทั้งหมดดังรูปที่ 13

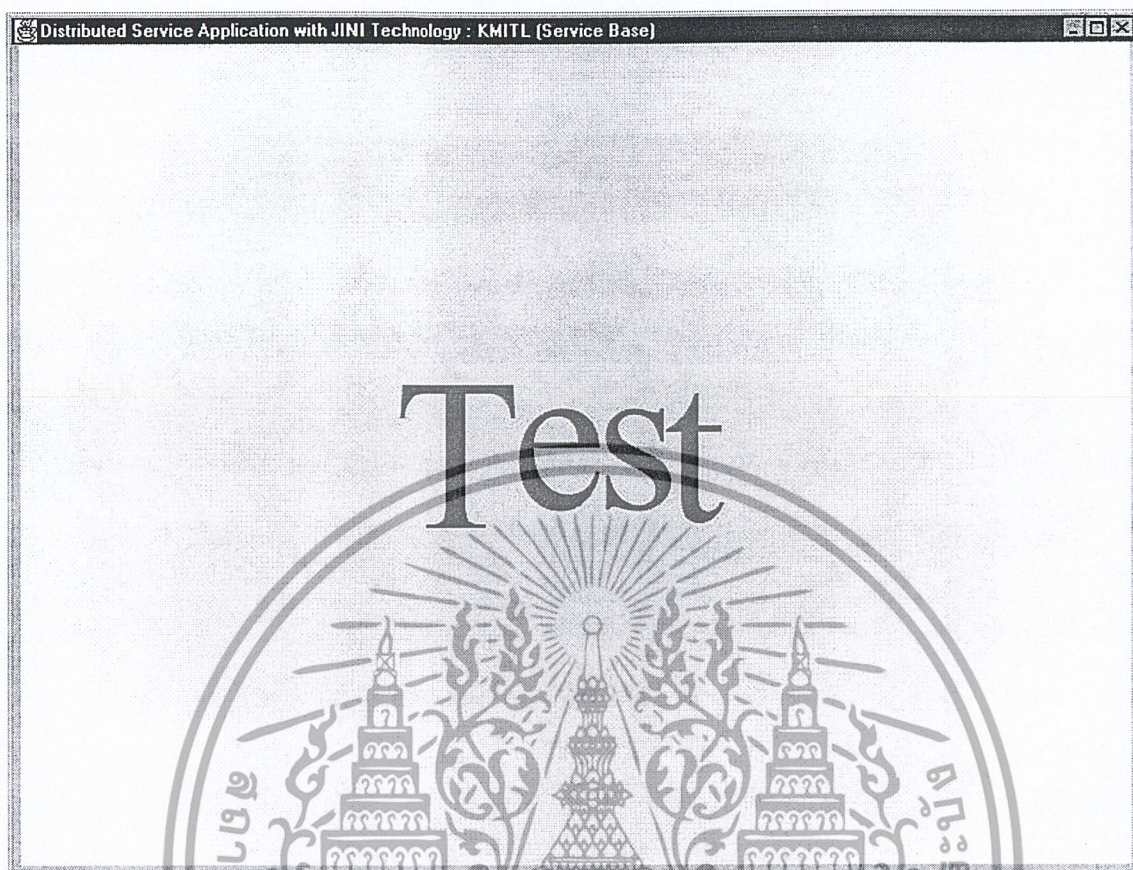
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ผ.15 แสดงโอเพอร์เรชันวิวที่แสดงภาพนำเสนอภาพแรกเมื่อกดปุ่ม Play

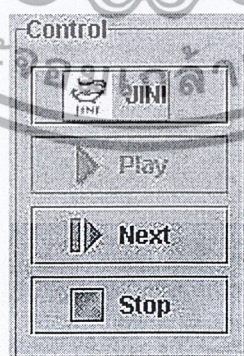
และที่ส่วนเซอร์วิสจะเกิดเฟรมนำเสนอที่มีขนาดเต็มหน้าจอขึ้นโดยอัตโนมัติดังรูปที่ 16 ซึ่งจะเกิดเฟรมนี้ที่ส่วนของเซอร์วิสได้ผู้ใช้งานจะต้องทำการเลือกเซอร์วิสก่อนโดยการกดปุ่ม JINI แล้วทำการเปิดไฟล์และกดปุ่ม Play เท่านั้น หากไม่ทำตามขั้นตอนนี้ ก็การประมวลผลต่างๆ จะอยู่ที่ส่วนไคลเอนท์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ผ.16 แสดงเฟรมนำเสนอที่เกิดขึ้นที่เซอร์วิส

15. สามารถควบคุมไฟล์นำเสนอได้ คือ การเล่นภาพต่อไป (Next) และการหยุดการนำเสนอ (Stop) โดยสามารถควบคุมจากปุ่ม Next และ ปุ่ม Stop ตามรูปที่ 17 ที่หน้าจอหลักได้



รูปที่ ผ.17 แสดงปุ่ม Next และ ปุ่ม Stop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16. หลังจากสิ้นสุดการใช้งานแล้วสามารถปิดโปรแกรมได้โดยการกดปุ่ม Close ที่หน้าจอหลัก หรือ ปุ่มกากบาทที่มุมขวาบน ดังรูปที่ 18



รูปที่ ผ.18 แสดงปุ่มปิดโปรแกรมทั้ง 2 ปุ่ม

