

เฟรมเวิร์กสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร
Co-operative Framework for Document Composition



นางสาววราภรณ์ พงษ์ธานี
นายอินทชัย สุคนธมาน

เลขหม.....
เลขทะเบียน..... 42770
วัน, เดือน, ปี 10 ส.ย. 2545



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร
Co-operative Framework for Document Composition

โดย

นางสาววราภรณ์ พงษ์ธานี

นายอินทรชัย สุคนธมาน



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร

Co-operative Framework for Document Composition

ผู้จัดทำ

1. นายอินทรัชย์ สุคนธมาน รหัสประจำตัว 40011015
2. นางสาววราภรณ์ พงษ์ธานี รหัสประจำตัว 40010693



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร

นางสาววราภรณ์ พงษ์ธานี 40010693
 นายอินทรชัย สุคนธมาน 40011015
 ดร.วิศิษฎ์ หิรัญกิตติ อาจารย์ที่ปรึกษา
 ปีการศึกษา 2543

บทคัดย่อ

การนำเอาคอมพิวเตอร์มาช่วยในการทำงานด้านต่างๆร่วมกับเทคโนโลยีทางอินเทอร์เน็ตได้เพิ่มขึ้นอย่างรวดเร็วก่อให้เกิดการวิจัยและพัฒนาแอปพลิเคชันต่างๆที่นำเอาประโยชน์ของเครือข่ายเวิร์ลไวด์เว็บมาใช้สนับสนุนการทำงานร่วมกันของผู้ใช้ที่มีวัตถุประสงค์ในการทำงานร่วมกัน เพื่อให้การทำงานร่วมกันมีความสะดวกเพิ่มมากขึ้น อีกทั้งยังทำให้ข้อจำกัดทางด้านเวลาและสถานที่หมดไป ก่อให้เกิดประโยชน์อย่างมากแก่ผู้ใช้ที่ต้องการทำงานร่วมกัน

โครงการนี้ได้มีการศึกษาและทำการสร้างระบบเอเจนต์เบื้องต้นที่ทำงานเสมือนหนึ่งเป็นอิดเตอร์สำหรับประพันธ์เอกสารบนเว็บขึ้นโดยใช้ภาษาจาวาในการพัฒนาและได้มีการนำเอาแนวความคิดของการทำงานร่วมกันผ่านทางระบบเครือข่ายทางคอมพิวเตอร์มาใช้ ทำให้ผู้ใช้สามารถทำงานเอกสารร่วมกันบนเว็บได้ ด้วยระบบนี้ผู้ใช้แต่ละคนสามารถใช้คอมพิวเตอร์ของแต่ละคนเชื่อมต่อเข้ากับอินเทอร์เน็ตแล้วร่วมกันประพันธ์เอกสารเดียวกันหลายคนพร้อมกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Co-operative Framework for Document Composition

Waraporn Phongtanee

Intharachai Sukonthaman

Dr.Visit Hirankitti Advisor

ABSTRACT

This is a great deal of growing interest among researchers to develop a network application working on the internet which let people working together to achieve the same objective. This is a type of application which is call “Co-operative Framework for Document Composition”. This application allows co-operative teamworks to work together although they work in difference places and difference times.

This thesis studies a co-operative application which is a co-operative editor using Java networking technology. Using this editor, several users at difference places can work together in order to compose a single document.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี สืบเนื่องมาจากการได้รับความช่วยเหลือและความร่วมมือที่ดีจากบุคคลต่าง ๆ หลายฝ่าย บุคคลแรกที่มีส่วนอย่างมากในการทำให้ปริญญาานิพนธ์ฉบับนี้สำเร็จได้ก็คือ ดร. วิศิษฎ์ หิรัญกิตติ ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญาานิพนธ์ของผู้เขียน อาจารย์ให้ความเอาใจใส่ คำแนะนำและความช่วยเหลือเสมอมาตั้งแต่เริ่มต้นจนกระทั่งปริญญาานิพนธ์สำเร็จได้ด้วยดี จึงต้องขอขอบพระคุณเป็นอย่างยิ่ง

บุคคลฝ่ายที่สองที่ต้องขอขอบคุณเป็นอย่างมากก็คือ เพื่อน ๆ ทุกคนที่คอยช่วยเหลือ ให้คำแนะนำ และให้กำลังใจ ช่วยแก้ไขปัญหาร่วมกันตลอดมา รวมทั้งต้องขอขอบคุณห้อง Intelligence Communication Lab (ICL) ที่เอื้อเฟื้อสถานที่ในการทำปริญญาานิพนธ์นี้

สุดท้ายขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่และยังให้กำลังใจเอาใจใส่เสมอมาในทุกๆด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

วราภรณ์ พงษ์ธานี

อินทรชัย สุคนธมาน

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตารางและรูปภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 หลักการและทฤษฎี	4
2.1 หลักการทำงานร่วมกัน	4
2.1.1 บทนำ	4
2.1.2 การทำงานร่วมกันและการปรับปรุงการทำงาน (Cooperative work & Tailoring)	4
2.1.2.1 การทำงานร่วมกัน (Cooperative work)	4
2.1.2.2 การปรับปรุงการทำงาน (Tailoring)	5
2.1.3 คอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer Supported Cooperative Work(CSCW))	5
2.1.3.1 ความหมายของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน	5
2.1.3.2 มิติของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer Supported Cooperative Work Dimentions หรือ CSCW Dimentions)	6
2.1.3.3 ตัวอย่างแนวความคิดและเทคโนโลยีเกี่ยวกับคอมพิวเตอร์ ที่สนับสนุนการทำงานร่วมกัน	6
2.1.4 สิ่งที่ควรพิจารณาในการทำงานร่วมกัน	7
2.2 เอเจนต์(Agent)	8
2.2.1 บทนำ	8
2.2.2 คำนิยามของเอเจนต์	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3	การทำงานของเอเจนต์	9
2.2.4	แนวทางการสร้างเอเจนต์	11
2.2.5	การเรียนรู้ข้อมูลของเอเจนต์	12
2.3	ไฮเพอร์เท็กซ์ ทรานส์เฟอร์ โพรโตคอล (HyperText Transfer Protocol หรือ HTTP)	13
2.3.1	บทนำ	13
2.3.2	การทำงานของ เฮชทีทีพี (HTTP)	15
2.3.2.1	หลักการการทำงานและวิธีการของไคลเอ็นต์/เซิร์ฟเวอร์	15
2.3.2.2	การร้องขอบริการ (Request)	16
2.3.2.3	การตอบกลับ/การให้บริการ (Response/Reply)	17
2.3.2.4	การเชื่อมต่อระหว่างผู้ใช้บริการและผู้ให้บริการในระบบเครือข่าย	19
2.3.2.5	ประเภทของตัวกลาง (Intermedia)	20
2.3.3	เฮดเดอร์ (Header) ของการแสดงผล	21
2.3.3.1	การจัดการเฮดเดอร์ (Header) ของการแสดงผล	21
2.3.3.2	รายละเอียดเฮดเดอร์ (Header) ของเฮชทีทีพี (HTTP) โพรโตคอล	22
2.4	คอมมอน เกทเวย์ อินเตอร์เฟส (Common Gateway Interface หรือ CGI)	23
2.4.1	บทนำ	23
2.4.2	การทำงานของ ซีจีไอ (CGI)	24
2.4.3	การใช้งานฟอร์มแท็ก (Form Tag)	26
2.4.3.1	ฟอร์มแท็ก (Form Tag)	26
2.4.3.2	เมธอด (Method)	26
2.5	เซิร์ฟเล็ต (Servlet)	27
2.5.1	บทนำ	27
2.5.2	ข้อดีของเซิร์ฟเล็ต (Servlet)	29
2.5.3	ขั้นตอนการทำงานพื้นฐานของเซิร์ฟเล็ต (Basic Servlet Flow)	32
2.5.4	ความสัมพันธ์ระหว่าง เซิร์ฟเล็ต (Servlet) กับ แอปเพล็ต (Applet)	33
2.5.5	วงจรชีวิตของเซิร์ฟเล็ต (Java Servlet Life Cycle)	33
บทที่ 3	โครงสร้างและการทำงานของระบบ	35
3.1	การเลือกโปรแกรมประยุกต์และสิ่งแวดล้อมที่ใช้ในสร้างโปรแกรมทำงาน	35
3.2	ขั้นตอนการสร้างโปรแกรมการทำงานร่วมกัน	35
3.3	โครงสร้างของโปรแกรมการทำงานร่วมกัน	36
3.4	อินเทอร์เฟซต่างๆที่ติดต่อกับผู้ใช้	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5	ขั้นตอนการสร้างกลุ่มการทำงานใหม่	40
3.6	ขั้นตอนการลงทะเบียนผู้ใช้	41
3.7	ขั้นตอนการตรวจสอบการอนุญาตเข้าทำงาน	42
3.8	ฟังก์ชันการทำงานต่างๆ ของโปรแกรมการทำงานร่วมกัน	44
3.8.1	การอัปโหลดไฟล์ภาพไปยังเซิร์ฟเวอร์	44
3.8.2	การแสดงรายละเอียดของผู้ใช้หรือรายละเอียดของกลุ่มการทำงาน	44
3.8.3	การแสดงลิสต์ของผู้ใช้ทั้งหมด และ/หรือ ลิสต์ของกลุ่มการทำงานทั้งหมด	45
3.9	การทำงานของโปรแกรมการทำงานร่วมกัน	45
3.10	การแสดงผลลัพธ์ของการทำงาน	50
บทที่ 4	การทดลองและวิเคราะห์ผลการทดลอง	51
4.1	บทนำ	51
4.2	ส่วนประกอบและคำอธิบายโปรแกรมการทำงาน	51
4.3	การทดสอบและใช้งาน โปรแกรม	52
4.3.1	ตัวอย่างการสร้างกลุ่มการทำงานใหม่	52
4.3.2	ตัวอย่างการลงทะเบียนผู้ใช้งาน	53
4.3.3	ตัวอย่างการอัปโหลดไฟล์ภาพมายังเซิร์ฟเวอร์	54
4.3.4	ตัวอย่างการแสดงผลชื่อผู้ใช้ และ/หรือ กลุ่มการทำงานทั้งหมด	54
4.3.5	ตัวอย่างแสดงการขออนุญาตเข้าทำงาน	55
4.3.6	ตัวอย่างการทำงานร่วมกันในการเพิ่มข้อความ และรูปภาพและรูปทรงเรขาคณิต	56
4.3.7	ตัวอย่างการแก้ไข และลบอิลิเมนต์ที่ผลลัพธ์	59
4.3.8	ตัวอย่างซอร์สโค้ดภาษาXML และ ผลลัพธ์ภาษา HTML	61
4.4	ปัญหาและทางแก้ไข	62
4.4.1	ปัญหาจากการทำงานร่วมกันของผู้ใช้ภายในกลุ่มการทำงานเดียวกัน	62
4.4.1.1	การแสดงความเป็นเอกลักษณ์ของผู้ใช้แต่ละคน	62
4.4.1.2	ความสัมพันธ์ของระยะทางและลำดับก่อนหลัง	62
4.4.1.3	อิสระของผู้ใช้ในการจัดการงาน มีเคอร์เซอร์อิสระ	62
4.4.2	ปัญหาการจัดการการทำงานหลายกลุ่มการทำงานพร้อม ๆ กันของเซิร์ฟเล็ต	62
4.4.3	ปัญหาของการแสดงผลภาพ	62
4.4.3.1	ปัญหารูปที่ใช้จะต้องอยู่ที่เซิร์ฟเวอร์	62
4.4.3.2	ปัญหาขนาดภาพที่ใหญ่เกินไป	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทสรุปและวิจารณ์	64
5.1 บทสรุปและวิจารณ์	64
5.2 ปัญหาที่ยังไม่ได้แก้ไข	64
5.3 แนวทางการพัฒนา	65
ภาคผนวก ก.	68
บรรณานุกรม	87



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 แสดงอินเทอร์เฟซที่ติดต่อกับผู้ใช้ในการสร้างกลุ่มการทำงานใหม่	37
รูปที่ 3.3 แสดงอินเทอร์เฟซที่ติดต่อกับผู้ใช้ในการลงทะเบียนผู้ใช้	38
รูปที่ 3.4 แสดงอินเทอร์เฟซที่ติดต่อกับผู้ใช้ในการอนุญาตให้เข้าทำงาน	38
รูปที่ 3.5 แสดงอินเทอร์เฟซที่ติดต่อกับผู้ใช้ในการอัปโหลดไฟล์ภาพไปยังเซิร์ฟเวอร์ , แสดงรายละเอียดของผู้ใช้ หรือ รายละเอียดของกลุ่มการทำงาน , แสดงลิสต์ของผู้ใช้ทั้งหมด และ/หรือ ลิสต์ของกลุ่มการทำงานทั้งหมด	39
รูปที่ 3.6 แสดงไดอะแกรมของขั้นตอนการสร้างกลุ่มการทำงาน	40
รูปที่ 3.7 แสดงไดอะแกรมของขั้นตอนการลงทะเบียนผู้ใช้	42
รูปที่ 3.8 แสดงไดอะแกรมของขั้นตอนการตรวจสอบการขออนุญาตเข้าทำงาน	44
รูปที่ 3.9 แสดงการติดต่อกันระหว่างเซิร์ฟเวอร์กับแอปพลิเคชัน	45
รูปที่ 3.10 แสดงขั้นตอนการแสดงผลข้อความ	46
รูปที่ 3.11 แสดงขั้นตอนการแสดงผลรูปภาพและรูปทรงเรขาคณิต	47
รูปที่ 3.12 แสดงการทำงานของการทำงานแก้ไข และ ลบอิเลิเมนต์ผลลัพธ์	49
รูปที่ 3.13 แสดงการแปลงผลลัพธ์การทำงาน	50
รูปที่ 4.1 แสดงรูปของ โปรแกรมการทำงานร่วมกันในงานเอกสาร	52
รูปที่ 4.2 แสดงตัวอย่างการป้อนข้อมูลในการสร้างกลุ่มทำงานใหม่	52
รูปที่ 4.3 แสดงตัวอย่างการสร้างตารางเพื่อใช้ในการทำงานภายในกลุ่มการทำงาน	52
รูปที่ 4.4 แสดงตัวอย่างการบันทึกข้อมูลลงตารางกลุ่มการทำงาน	52
รูปที่ 4.5 แสดงตัวอย่างการป้อนข้อมูลการลงทะเบียนผู้ใช้	53
รูปที่ 4.6 แสดงตัวอย่างการบันทึกข้อมูลลงตารางผู้ใช้	53
รูปที่ 4.7 แสดงตัวอย่างการอัป โหลดไฟล์	54
รูปที่ 4.8 แสดงผลลัพธ์การอัป โหลดไฟล์	54
รูปที่ 4.9 แสดงตัวอย่างการเลือกการแสดงรายชื่อผู้ใช้และรายชื่อกลุ่มการทำงานทั้งหมด	54
รูปที่ 4.10 แสดงตัวอย่างรายชื่อผู้ใช้ รายชื่อกลุ่มการทำงานทั้งหมด	55
รูปที่ 4.11 แสดงตัวอย่างการป้อนข้อมูลการขออนุญาตเข้าทำงาน	55
รูปที่ 4.12 แสดงตัวอย่างการเพิ่มข้อความ โดยผู้ใช้ป้อนข้อมูลที่แท็กซ์แอเรีย แล้วกดปุ่ม Send Messages	56
รูปที่ 4.13 แสดงตัวอย่างผลลัพธ์การเพิ่มข้อมูล	56
รูปที่ 4.14 แสดงตัวอย่างการทำกรใส่ไฟล์รูปภาพ โดยผู้ใช้เลือกไฟล์จากไฟล์ไดอะล็อก	57
รูปที่ 4.15 แสดงผลลัพธ์การเพิ่มไฟล์ภาพ	57
รูปที่ 4.16 แสดงการเพิ่มรูปทรงทางเรขาคณิตโดยผู้ใช้กำหนดชนิด ขนาด และสี	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของรูปทรงเรขาคณิตและสามารถ Preview ก่อนทำการเพิ่มรูปทรงเรขาคณิตด้วยการกดปุ่ม ok	
รูปที่ 4.17 แสดงผลลัพธ์ของการเพิ่มรูปทรงเรขาคณิต	58
รูปที่ 4.18 แสดงการแก้ไขอิลิเมนต์ผลลัพธ์ด้วยการแทนที่ข้อความในแถบสว่างด้วยข้อความในเท็กซ์แอเรีย เมื่อผู้ใช้กดปุ่ม Replace	59
รูปที่ 4.19 แสดงผลลัพธ์การแก้ไขอิลิเมนต์ผลลัพธ์	59
รูปที่ 4.20 แสดงตัวอย่างการลบอิลิเมนต์ผลลัพธ์ โดยการที่ผู้ใช้กดปุ่ม Delete	60
รูปที่ 4.21 แสดงผลลัพธ์การลบอิลิเมนต์ผลลัพธ์	60
รูปที่ 4.22 แสดงตัวอย่างซอร์สโค้ดของการทำงานหลังจากผู้ใช้กดปุ่ม Save	61
รูปที่ 4.23 แสดงตัวอย่างผลลัพธ์ภาษา HTML ที่ได้จากการแปลงด้วย CFDC.xml	61



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในการทำงานต่างๆในชีวิตประจำวันนั้น เวลาและสถานที่ที่เป็นปัจจัยที่มีบทบาทสำคัญอย่างยิ่งที่เป็นข้อจำกัดในการทำกิจกรรมต่างๆ โดยเฉพาะในการทำงานร่วมกันเป็นทีมงานนั้นเป็นกิจกรรมที่ใช้เวลาสูงมาก เนื่องจากการเสียไปของเวลาในการเดินทางและการเตรียมการเรื่องสถานที่ที่เป็นไปด้วยความยากลำบาก การสร้างโปรแกรมที่ผู้ใช้หลายๆคนสามารถทำงานร่วมกันได้บนเว็บจึงเป็นแนวทางที่นำมาแก้ปัญหาดังกล่าวทำให้สามารถกำจัดข้อจำกัดทางด้านเวลา คือประหยัดเวลาในการมาทำงานร่วมกัน อีกทั้งกำจัดข้อจำกัดทางด้านสถานที่ คือ ผู้ใช้สามารถทำงานร่วมกันได้โดยไม่ต้องอยู่ ณ สถานที่เดียวกันแต่เป็นการทำงานร่วมกันผ่านเครือข่ายทางอินเทอร์เน็ตที่ผู้ใช้ที่อยู่ต่างสถานที่กันก็สามารถทำงานร่วมกันได้ กล่าวได้ว่าเป็นการนำเอาความก้าวหน้าของเทคโนโลยีทางคอมพิวเตอร์มาใช้เพื่อก่อให้เกิดประโยชน์ทางด้านการทำงานร่วมกัน

การพัฒนาให้มีการทำงานร่วมกันโดยผ่านเครือข่ายทางคอมพิวเตอร์หรือการที่ให้คอมพิวเตอร์เป็นตัวแทน (Agent) ในการทำงานของผู้ใช้งานนั้น ผู้เขียน โปรแกรม จะต้องมีการวางแผนและกำหนดนโยบายที่สะท้อนและบ่งบอกให้ผู้ใช้รู้ว่าจะสามารถใช้แอปพลิเคชัน (Application) นั้นในการติดต่อกับผู้ใช้คนอื่นๆได้อย่างไร โดยนโยบายนั้นสามารถเพิ่มเติม (Added), ปรับปรุงแก้ไข (Modify), และละทิ้ง (Abandon) ได้โดยพิจารณาตามความเหมาะสมของการใช้งานเพื่อให้การทำงานร่วมกันเป็นไปอย่างมีประสิทธิภาพ

โปรแกรมเอเจนต์ที่ทำงานบนเว็บนั้นจะเป็นตัวแทน (Agent) ที่ช่วยในการจัดการงานทางด้านการทำเอกสารให้กับกลุ่มผู้ใช้งาน โดยให้ผู้ใช้แต่ละคนจะมีส่วนร่วมในการสร้างเอกสารขึ้นมา ผู้ใช้แต่ละคนสามารถเพิ่มเติมและแก้ไขผลงานที่แสดงผลออกทางหน้าจอแสดงผลส่วนกลาง โดยต้องได้รับการรับรู้และยอมรับจากผู้ใช้คนอื่นๆ ในอนาคตพัฒนาการของโปรแกรมหากล่าวจะทำให้การทำงานร่วมกันเป็นไปอย่างมีประสิทธิภาพสูงได้

1.2 วัตถุประสงค์ของปริญญาณิพนธ์

ในการทำเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร (Co-operative Framework For Document Composition) หรืออาจจะเรียกเพื่อให้เข้าใจเพิ่มมากขึ้นได้ว่า เป็นเสมือนเอเจนต์ (Editor) ที่ทำงานร่วมกันบนเว็บนั้นมีวัตถุประสงค์ดังต่อไปนี้

1. เพื่อศึกษาการทำงานของเซิร์ฟเล็ต (เซิร์ฟเล็ต) และสามารถใช้อเจนต์ในการพัฒนาไดนามิกเว็บเพจ (Dynamic Web Page) ให้มีประสิทธิภาพเพิ่มมากขึ้นได้
2. เพื่อให้ผู้ใช้งานสามารถสร้างเอกสารร่วมกันโดยผ่านเครือข่ายทางอินเทอร์เน็ต (Internet) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เพื่อเพิ่มความสะดวกในการจัดทำเอกสารร่วมกัน จากเดิมที่ผู้ใช้งานต้องนัดกันเพื่อทำเอกสารรายงานร่วมกัน ก็สามารถจัดทำเอกสารร่วมกัน โดยผ่านเครือข่ายทางอินเทอร์เน็ตได้นั้นก็หมายความว่าไม่จำเป็นที่ผู้จัดทำเอกสารทุกคนต้องอยู่ ณ ที่เดียวกันจะอยู่ที่ไหนก็สามารถทำงานร่วมกันได้อีกทั้งยังประหยัดเวลาในการเดินทางมาเพื่อทำงานร่วมกันอีกด้วย
4. เพื่อเป็นแนวทางในการพัฒนาเว็บเพจแก่ผู้ที่สนใจที่จะพัฒนาเว็บเพจโดยใช้เทคโนโลยีแนวความคิดของเซิร์ฟเล็ทส์ต่อไป
5. เพื่อเป็นแนวทางในการพัฒนาแอปพลิเคชันที่สนับสนุนการทำงานทางด้านการทำเอกสารร่วมกันผ่านทางเครือข่ายอินเทอร์เน็ตสืบต่อไป

1.3 ขอบเขตของปริญญานิพนธ์

1. ศึกษาและรวบรวมข้อมูลเกี่ยวกับการทำงานของเอเจนต์และภาษาการโปรแกรมที่จะนำมาใช้ในการพัฒนา คือ ภาษาจาวา (Java)
2. ศึกษาและรวบรวมข้อมูลเกี่ยวกับการทำงานร่วมกันผ่านทางคอมพิวเตอร์ (Computer Support Cooperative Work) รวมทั้งการทำงานของระบบไคลเอ็นต์/เซิร์ฟเวอร์ (Client/Server)
3. ศึกษาและรวบรวมข้อมูลเกี่ยวกับ ไฮเพอร์เท็กซ์ ทรานส์เฟอร์ โพรโตคอล (HyperText Transfer Protocol หรือ HTTP) ซึ่งเป็น โพรโตคอลที่ใช้ในการติดต่อสื่อสารผ่านเครือข่ายทางอินเทอร์เน็ต
4. ศึกษาและรวบรวมข้อมูลเกี่ยวกับการทำงานของ คอมมอน เกทเวย์ อินเตอร์เฟซ (Common Gateway Interface หรือ CGI)
5. ศึกษาและรวบรวมข้อมูลเกี่ยวกับเซิร์ฟเล็ทส์และการทำงานของเซิร์ฟเล็ทส์เพื่อนำมาประยุกต์ใช้ในการพัฒนาโครงการ
6. พัฒนาโปรแกรมตัวอย่างเพื่อแสดงการทำงานร่วมกันด้านการประพันธ์เอกสารที่ทำงานผ่านเครือข่ายทางอินเทอร์เน็ต

1.4 วิธีการดำเนินงาน

1. เข้าพบอาจารย์ที่ปรึกษาเพื่อรับฟังแนวความคิดและขอบเขตของการดำเนินงาน
2. ค้นคว้าหาข้อมูลจากเอกสาร, เว็บ, และหนังสือต่างๆที่เกี่ยวข้อง
3. ศึกษาและทำความเข้าใจเกี่ยวกับระบบการทำงานแบบไคลเอ็นต์/เซิร์ฟเวอร์ (Client/Server), เซิร์ฟเล็ท (เซิร์ฟเล็ท), การทำงานของเซิร์ฟเล็ท, ภาษาจาวา (Java Programming Language) และการเขียนโปรแกรมโดยใช้ภาษาจาวาเพื่อเป็นแนวทางในการพัฒนาเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร (Co-operative Framework For Document Composition)
4. ตัดตั้งผลิตภัณฑ์ต่างๆที่จำเป็นต่อการใช้ในการพัฒนา อันประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จาวา ดีวีโพลเมนต์ คิต (Java Development Kit (JDK-1.3))
 - จาวา เว็บ เซิร์ฟเวอร์ (Java Web Server 2.0)
 - ออราเคิล 8.1 (Oracle 8.1)
5. ติดตั้งค่าต่างๆ (Configuration) ให้กับ จาวาเว็บเซิร์ฟเวอร์ (Java Web server 2.0)
 6. ทดลองเขียน โปรแกรมเซิร์ฟเล็ทงายๆเพื่อสร้างความเข้าใจในการเขียน โปรแกรม
 7. รันโปรแกรมและดูผลการทำงาน
 8. เริ่มพัฒนาโปรแกรมเพื่อสร้างเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร (Co-operative Framework For Document Composition)
 9. รันโปรแกรมและทดสอบผลการทำงานของเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร (Co-operative Framework For Document Composition)
 10. รายงานความคืบหน้าให้กับอาจารย์ที่ปรึกษาทราบเป็นระยะและรับฟังคำแนะนำเพิ่มเติม
 11. ปรับปรุงแก้ไขในกรณีที่เกิดปัญหาหรือผลการทำงานไม่เป็นไปตามต้องการ
 12. เตรียมรายงาน, เอกสารอธิบายโปรแกรม, คู่มือการใช้งานโปรแกรม, และปริญญานิพนธ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและทฤษฎี

2.1 หลักการทำงานร่วมกัน(Theory Of Cooperative Work)

2.1.1 บทนำ

ที่ผ่านมามีหลายการวิจัยที่ศึกษาเกี่ยวกับหนทางที่จะทำให้กลุ่มของผู้ใช้ (Users) สามารถทำงานร่วมกันได้ แม้ว่าการศึกษาวิจัยเริ่มต้นจะพยายามใช้หลักการทางวิทยาศาสตร์คอมพิวเตอร์ (Computer Science) ที่เริ่มต้นโดย ดักลาส เอ็นเจลบาร์ต (Douglas Engelbart) และมีการวิจัยเรื่อยมาเพื่อที่จะสนับสนุนกิจกรรมการทำงานร่วมกันของกลุ่มผู้ใช้งานผ่านทางคอมพิวเตอร์ โดยในกลางปี 1980 การวิจัยปรากฏออกมาภายใต้ชื่อของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer Supported Cooperative Work (CSCW)) ภายใต้หลักการทางวิทยาศาสตร์คอมพิวเตอร์ เป็นระยะเวลากว่า 10 ปี ที่ผ่านมาได้มีการค้นคว้าวิจัยอย่างกว้างขวางในการทำต้นแบบ (Phototype) และผลิตภัณฑ์ต่างๆเพื่อการติดต่อค้าขาย โดยพัฒนาจากระบบผู้ใช้งาน (End user) และขั้นตอนของการทำงาน (Workflow Program) เพื่อที่จะสร้างเครื่องมือ (Tool) และ แอปพลิเคชัน (Application) ในการปรับปรุงพัฒนา (Implementation) การทำงานร่วมกันให้มีประสิทธิภาพมากยิ่งขึ้น

2.1.2 การทำงานร่วมกันและการปรับปรุงการทำงาน (Cooperative work & Tailoring)

2.1.2.1 การทำงานร่วมกัน (Cooperative work)

งานทำงานร่วมกันโดยผ่านทางคอมพิวเตอร์นั้นเป็นการสร้างระบบข้อมูลข่าวสารที่ติดต่อกัน (Interactive Information system) ซึ่งเป็นการปรับให้เข้ากับยุคของเทคโนโลยีข้อมูลข่าวสาร (Information Technology) เพื่อตอบสนองความต้องการและเพิ่มความสะดวกให้กับผู้ใช้งานด้านการทำงานร่วมกัน โดยสิ่งสำคัญที่ควรระลึกถึงอยู่เสมอ คือ จะทำอย่างไรให้สามารถปรับให้เข้ากับความต้องการของผู้ใช้งานได้และง่ายต่อการใช้งาน โดยการพัฒนามีส่วนเกี่ยวข้องกับ การออกแบบการทำงานของตัวโปรแกรม (Design program), การออกแบบหน้าจอติดต่อกับผู้ใช้งาน (User interface), การทำงานของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer support cooperative work), การทำงานของผู้ใช้งาน (User computing), การจัดการต่างๆเกี่ยวกับเว็บ (ยกตัวอย่างเช่น HTML, Dreamweaver, Photoshop เป็นต้น), การจัดการกับฐานข้อมูล (Database), การจัดการเกี่ยวกับผู้ใช้งานหลายคน (Multiusers), การจัดการเกี่ยวกับซอร์ฟแวร์ (ยกตัวอย่างเช่น การเลือกใช้ภาษาในการพัฒนา, กรรมวิธีและเทคนิคต่างๆ (Methods and Techniques) เป็นต้น), และการจัดการหน้าจอหลัก (ยกตัวอย่างเช่น การรีเฟรช (Refresh), การทำบัฟเฟอร์คู่ (Double buffer) เป็นต้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.2 การปรับปรุงการทำงาน (Tailoring)

คำว่าเทลลอร์ริง (Tailoring) นั้นมีนิยามเป็น กิจกรรมของการแก้ไขเปลี่ยนแปลงคอมพิวเตอร์แอปพลิเคชัน (Computer Application) ภายใต้วงแวดล้อม (Context) ของการใช้งาน, เทลลอร์ริงเป็นการส่งเสริมการพัฒนาของแอปพลิเคชันในระหว่างการใช้งานและการดัดแปลงแอปพลิเคชันให้เหมาะสมกับความต้องการของผู้ใช้ ซึ่งอาจจะไม่ใช่สิ่งที่ออกแบบไว้ในตอนแรกก็ได้ เพราะมีการปรับปรุงในระหว่างการพัฒนาเพื่อให้เหมาะสมกับความต้องการของผู้ใช้, เทลลอร์ริงเป็นสมมติฐานที่ถูกต้องอย่างกว้างขวางในการนำไปเป็นคีย์ริควิเมนต์ (Key Requirement) สำหรับการออกแบบระบบการทำงานของกลุ่มผู้ใช้ (Groupware System) เป็นการกำหนดให้สามารถเปลี่ยนแปลงและปรับได้ตามสถานการณ์ที่เปลี่ยนไปเพื่อตอบสนองความต้องการของผู้ใช้

ในการปรับปรุงการทำงานนั้นจะต้องมีการพิจารณาถึงความสัมพันธ์ต่างๆ ในการทำให้คอมพิวเตอร์แอปพลิเคชัน (Computer Application) มีประสิทธิภาพดีขึ้นและเหมาะสมกับสถานการณ์การทำงานของผู้ใช้ ลักษณะคล้ายกับในการพัฒนาระบบนั้นจะมีปัญหาเกิดขึ้นมากมายซึ่งไม่สามารถหาคำตอบได้ และในบางครั้งจะไม่สามารถควบคุมปัจจัยต่างๆ ซึ่งสนับสนุนลักษณะเฉพาะของผู้ใช้แต่ละคนได้ แต่สามารถระบุการติดตามความสัมพันธ์ของการทำงานในอนาคตได้เพื่อหาว่าอะไรคือสาเหตุและการป้องกันเพื่อให้เกิดการปรับปรุงแอปพลิเคชันให้เป็นไปตามความต้องการและบรรลุผลสำเร็จ โดยจะต้องดูระดับการใช้งาน ความเหมาะสม และที่สำคัญเหนือสิ่งอื่นใดคือ การออกแบบการติดต่อกับผู้ใช้ (User interface) จะต้องออกแบบให้เหมาะสม มีการเลือกคำอุปมา (Metaphor) ที่เหมาะสมในการติดต่อสื่อสาร, การออกแบบแอปพลิเคชันและการเลือกการแสดงผล (Representation) ที่เหมาะสมและง่ายต่อการใช้งาน สิ่งเหล่านี้จะมีส่วนช่วยให้แอปพลิเคชันมีประสิทธิภาพเพิ่มมากขึ้นได้ โดยหลักการของเทลลอร์ริงนั้นสามารถใช้ได้ทั้งในการทำงานของผู้ใช้เพียงคนเดียว (Single user) และการทำงานของผู้ใช้หลายคน (Multiuser) ที่มีกิจกรรมการทำงานร่วมกัน (Group Activity) โดยจะต้องพิจารณาถึง การทำงานร่วมกันของผู้ใช้แต่ละคน, การทำงานกับส่วนกลาง, การป้องกันและการจัดการกับการทำงานที่มากเกินไป (Overload), ความเชื่อถือของผู้ใช้, การสนับสนุนกิจกรรมต่างๆของผู้ใช้งาน รวมถึงเทคนิคต่างๆที่นำมาใช้ในการรองรับการเปลี่ยนแปลงและการแก้ปัญหา

2.1.3 คอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer Supported Cooperative Work หรือ CSCW)

2.1.3.1 ความหมายของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน

คอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (CSCW) เป็นการอ้างอิงความสนใจในการสนับสนุนการทำงานร่วมกันของผู้ใช้หลายคนกับระบบคอมพิวเตอร์และครอบคลุมถึงสิ่งใดๆ ที่ทำกับคอมพิวเตอร์และสนับสนุนกิจกรรมต่างๆที่มีมากกว่า 1 คนมาเกี่ยวข้องทำงานร่วมกัน ซีเอสซีดับิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(CSCW) ไม่ได้นิยามในทอมเทคนิคที่นำมาประยุกต์ใช้เท่านั้นแต่มองในลักษณะของการวิจัยกลุ่มที่มีเป้าหมายในการออกแบบระบบแอปพลิเคชัน (Application System) ให้เหมาะสมกับการใช้งานร่วมกัน อาจกล่าวได้ว่าเป็นการสนับสนุนความต้องการของการทำงานร่วมกัน (The support requirement of cooperative work) โดย ซีเอสซีดับบิว (CSCW)สามารถแก้ปัญหาการทำงานร่วมกันและสร้างความเข้าใจในการทำงานให้กับผู้ใช้งานแต่ละคนได้ โดยเป้าหมายของซีเอสซีดับบิว (CSCW) คือ การค้นพบหนทางที่จะใช้เทคโนโลยีทางคอมพิวเตอร์ส่งเสริมกระบวนการทำงานเป็นกลุ่มร่วมกันที่สนับสนุนทั้งในมิติของเวลาและสถานที่ (Time and place dimensions)

2.1.3.2 มิติของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer Supported Cooperative Work Dimensions หรือ CSCW Dimensions)

มี 2 มิติ คือ

1. มิติทางด้านเวลา (Time Dimension) แบ่งเป็น
 - ริงไทม์ (Real time) : เกิดเมื่อการติดต่อ (Communication) เกิดขึ้นในเวลาเดียวกัน
 - อะซิงโครนัส ไทม์ (Asynchronous time) : เกิดเมื่อการติดต่อเกิดในเวลาที่แตกต่างกัน
2. มิติทางด้านสถานที่ (Place Dimension) แบ่งเป็น
 - ที่เดียวกัน (Same place) : เมื่อการพบกันเกิดขึ้นเมื่อผู้มีส่วนร่วมอยู่ ณ สถานที่เดียวกัน
 - ต่างที่กัน (Different place) : เมื่อการพบกันเกิดขึ้นเมื่อผู้มีส่วนร่วมอยู่ในทำเลที่ตั้งที่แตกต่างกันไป

2.1.3.3 ตัวอย่างแนวความคิดและเทคโนโลยีเกี่ยวกับคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน

1. วายซีวีส (Wysiwiw) : เป็นการยึดมั่นในประโยคที่ว่า “ อะไรที่คุณเห็นคือสิ่งที่ฉันเห็น ” (What you see is what I see) ลักษณะคล้ายกับการที่คน 2 คน อยู่ที่บ้านของตนเองและดูรายการโทรทัศน์ได้เหมือนกันในเวลาเดียวกัน เทคโนโลยีได้ขยายแนวความคิดนี้และอนุญาตให้บุคคลปฏิบัติงานและติดต่อสื่อสารกันในสภาพแวดล้อมของวายซีวีส (Wysiwiw environment)
2. การรวมส่วนประกอบที่แตกต่างกันไว้ด้วยกัน (Integration differrenct tasks) : เพื่อเป็นการลดกำแพงความแตกต่างระหว่างส่วน ประกอบต่างๆของแอปพลิเคชันที่แตกต่างกัน และทำให้อยู่ในลักษณะที่สามารถปรับปรุงเปลี่ยนแปลงได้ง่าย คล้ายกับการมีเครื่องมือชิ้นใหญ่เพียงชิ้นเดียวแต่สามารถยืดหยุ่นและปรับเปลี่ยนตามต้องการและความเหมาะสมได้ (One large flexible tool) มากกว่าที่มีเครื่องมือชิ้นเล็กๆหลายๆชิ้นแต่ทำงานเฉพาะด้านไม่สามารถปรับเปลี่ยนได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลขานุการส่วนตัว (Personal secretary) : จะเป็นกระบวนการที่ทำตัวเสมือนเป็นเลขานุการส่วนตัวคอยจัดการเรื่องต่างๆ เช่น จะหยุดสิ่งที่ไม่จำเป็นและให้ทำสิ่งที่จำเป็นกว่าก่อน และช่วยเก็บข้อความ (Message) ต่างๆ ไว้ให้ตอนที่ไม่มีอยู่ เป็นต้น
4. การจัดการเกี่ยวกับเวลา (Time management) : เป็นผลิตภัณฑ์กรุปแวร์ (Groupware Product) ที่ช่วยจัดการเกี่ยวกับปฏิทิน, การบันทึกประจำวัน, ตารางการทำงาน (Project schedules) ที่ช่วยจัดการบริหารเวลาในการทำงานให้มีประสิทธิภาพมากขึ้น
5. มัลติมีเดีย (Multimedia) : เกี่ยวกับเรื่องของกราฟิกส์ (Graphics) และเสียง ให้ความเป็นธรรมชาติในการติดต่อระหว่างกันของผู้ใช้งาน
6. โปรแกรมสำหรับผู้ใช้งาน (End user program) : จะสนใจเกี่ยวกับการทำให้ระบบมีความเป็นส่วนตัวของผู้ใช้มากขึ้นมีกรุปแวร์ ทูล (Groupware tool) ที่สามารถยืดหยุ่นได้เพียงพอซึ่งเป็นมิตรกับผู้ใช้ (User friendly) ในการเปลี่ยนแปลงแก้ไขทูล (tool) ให้เหมาะสมกับความต้องการของผู้ใช้

2.1.4 สิ่งที่ต้องพิจารณาในการทำงานร่วมกัน

ในการทำงานร่วมกันผ่านทางคอมพิวเตอร์นั้นสิ่งที่จะต้องพิจารณาถึง คือ

1. สเปคของการติดต่อกัน (Specification of group interaction) : การติดต่อกันระหว่างผู้ใช้งาน (User) ที่ใช้งานแอปพลิเคชันในการทำงานร่วมกัน ผู้เขียนโปรแกรมจะต้องสามารถอธิบายนโยบายวิธีการควบคุมในการเข้าถึง (Access) ข้อมูล, การจัดการเกี่ยวกับการใช้งานและโปรโตคอล (Protocol) ที่ใช้ในการใช้งานพื้นที่ส่วนกลางร่วมกัน
2. การอิมพลีเมนต์ภาษาที่ใช้ในการพัฒนา (Implement of cooperative language) : คือภาษาที่นำมาใช้ในการพัฒนาเพื่อใช้ในการควบคุมข้อมูล (Information) ต่างๆของผู้ใช้ที่มีส่วนร่วมในการทำงานร่วมกัน และการควบคุมการกระทำ (Actions) ต่างๆของผู้ใช้เหล่านั้น
3. การทำงานเสมือนผู้ใช้คนเดียว (Re-engineering single-user application) : เป็นการรีเอ็นจิเนียริง (Re-engineering) ของกระบวนการการทำงานของแอปพลิเคชันที่มีการใช้งานของผู้ใช้เพียงคนเดียว (Non-collaborative applications) ไปเป็นการทำงานของแอปพลิเคชันที่มีผู้ใช้งานหลายคน (Multi-user collaborative applications) กล่าวคือ หมายความว่าในการทำงานของแอปพลิเคชันที่มีการทำงานร่วมกัน หากมีผู้ใช้งานเพียงคนเดียวก็จะสามารถทำงานได้เสมือนหนึ่งเป็นแอปพลิเคชันของการใช้งานของผู้ใช้เพียงคนเดียวนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 เอเจนต์(Agent)

2.2.1 บทนำ

อินฟอร์เมชันไฮเวย์ (Information highway) เป็นแนวทางใหม่ในการจัดการการทำงานและบริการของคอมพิวเตอร์ที่มีประสิทธิภาพ แต่เนื่องจากความซับซ้อนในกระบวนการนี้ทำให้ต้องการ การติดต่อกัน (Interaction)ระหว่างคอมพิวเตอร์ในรูปแบบใหม่ ซึ่งคอมพิวเตอร์จะต้องเป็นเหมือนผู้ช่วยที่ของผู้ใช้ แนวความคิดในการใช้เอเจนต์มาช่วยในการติดต่อกันระหว่างผู้ใช้และคอมพิวเตอร์นั้นถูกคิดขึ้นโดยทัศนะของ นิโคลัส นีโกรพอนต์ (Nicholas Negroponte) และ อลัน เค (Alan Kay) มีบริษัทหลายแห่งยอมรับและปรับเปลี่ยนมาใช้แนวคิดนี้ ซอฟต์แวร์เอเจนต์ประเภทต่างๆ เช่น เอเจนต์ที่ช่วยในการจัดการตารางการบริหารเวลา (Personal assistance about time scheduling agent), เอเจนต์ที่ช่วยในการจัดการการส่งเมลล์(Electronic mail handling agent), เอเจนต์ช่วยในการกรองข่าวสารอิเล็กทรอนิกส์ (Electronic new filtering agent) และ เอเจนต์ในด้านบันเทิง (Selection of entertainment agent) ได้ถูกสร้างขึ้น แต่อย่างไรก็ตามเทคนิคที่ใช้สร้างเอเจนต์ที่มีอยู่ในปัจจุบันยังห่างไกลกับความต้องการในการสร้าง การติดต่อเสมือนมนุษย์ในขั้นสูง (High-level human-like interaction)

2.2.2 คำนิยามของเอเจนต์

เอเจนต์ มีความหมายถึง บุคคลใดบุคคลหนึ่ง หรือ สิ่งใดสิ่งหนึ่ง ที่กระทำหน้าที่เป็นตัวแทนให้กับผู้ขอใช้บริการ เพื่อจัดการงานเฉพาะอย่างใดอย่างหนึ่งให้สำเร็จลุล่วงไปได้ด้วยดี

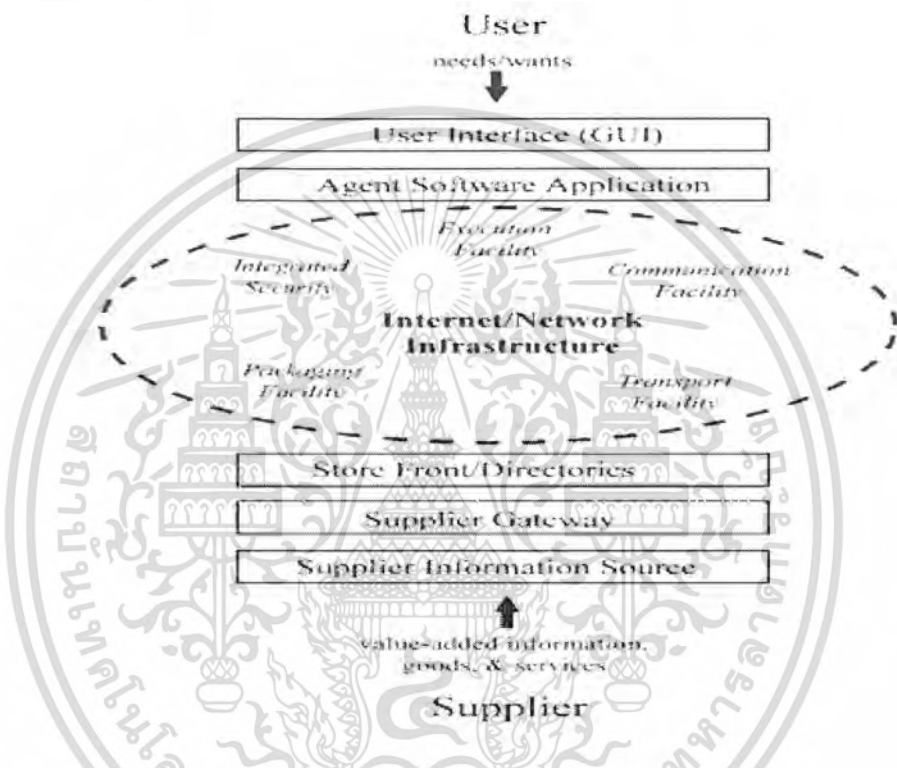
ในทางคอมพิวเตอร์ได้มีการพัฒนาโปรแกรมขึ้นเพื่อจัดการภาระต่างๆแทนผู้ใช้ให้สามารถทำงานได้อย่างสะดวก รวดเร็ว และลดความซับซ้อนที่เกิดขึ้นในงาน โปรแกรมที่ทำหน้าที่ช่วยเหลือโปรแกรมหลักให้สามารถทำงานได้ดีนั้นถูกเรียกว่า โปรแกรมผู้ช่วย หรือ ซอฟต์แวร์เอเจนต์

อาจจะกล่าวได้ว่าแอปพลิเคชันโปรแกรมเจนเนอเรชันที่4 เกือบทั้งหมดนั้นกระทำเป็นดังเช่นเอเจนต์ เพราะทุกวันนี้เราร้องขอการทำงานด้านคอมพิวเตอร์ผ่านทางซอฟต์แวร์ให้ทำงานต่างๆแทนเรา แต่หากมองให้ลึกแล้วจะทราบว่า ซอฟต์แวร์เอเจนต์ นั้นต่างจากแอปพลิเคชันอื่น โดยซอฟต์แวร์เอเจนต์จะต้องประกอบด้วยลักษณะของการเปลี่ยนแปลงได้อยู่เสมอ , ความอิสระในการปกครองและควบคุมตนเอง และความสามารถในการติดต่อกับผู้ใช้ เมื่อเราเพิ่มความชาญฉลาดให้กับเอเจนต์ เอเจนต์จะได้รับความสามารถของการใช้เหตุผลที่ปรับเปลี่ยนได้ตามความเหมาะสม ด้วยความสามารถนี้จะทำการโทรเสขข้อมูลจากสิ่งแวดล้อมภายนอก เช่น เน็ตเวิร์ค ฐานข้อมูล หรือ ในอินเทอร์เน็ต สามารถทำได้

การเกิดขึ้นของเครือข่ายอินเทอร์เน็ต และ เครือข่ายเวิลด์ไวด์เว็บทำให้ความต้องการการใช้ซอฟต์แวร์เอเจนต์ชาญฉลาดเพิ่มสูงขึ้น การใช้งานเว็บเปลี่ยนจากโมเดลแบบสแคเตอร์บราวซิง(Scattered Browsing)ไปเป็นโมเดลการส่งข้อมูลผ่านสื่อแบบพอยท์ทูพอยท์(Point to Point) แนวโน้มนี้ช่วยให้เอเจนต์มีความเป็นธุรกิจทางการค้ามากขึ้น

2.2.3 การทำงานของเอเจนต์

โมเดลการทำงานกับเอเจนต์แสดงดังรูปที่ 2.1 ผู้ใช้ที่ต้องการจัดการงานต่างๆใช้กราฟิกอินเตอร์เฟซที่ติดต่อกับซอฟต์แวร์แอปพลิเคชัน แอปพลิเคชันเหล่านี้จะทำหน้าที่เข้าไปหาข้อมูลในอินเทอร์เน็ตเพื่อไปถึงยังซัพพลายเออร์เกตเวย์ (Supplier gateway) ด้วยความช่วยเหลือของสตอร์ฟรอนต์ (Store front) ในที่สุดเอเจนต์จะสามารถเข้าถึง แหล่งข้อมูลในฐานข้อมูลหรือสื่อสำรองข้อมูลอื่นๆ ทำทรานส์แอ็กชัน (Transaction) ต่างๆและส่งผลลัพธ์กลับไปให้ผู้ใช้

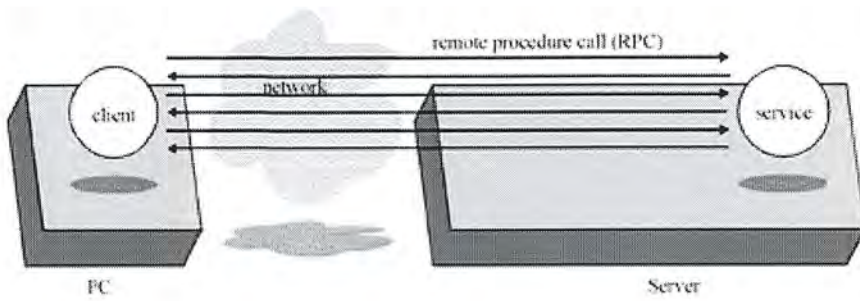


รูปที่ 2.1 แสดงโมเดลการทำงานของเอเจนต์

- การสื่อสารระหว่างกันด้วยวิธีดั้งเดิม (Original Approach)

ใช้วิธีการติดต่อที่เรียกว่ารีโมทโพรซีเจอร์คอล (Remote Procedure Call (RPC)) โดยRPCจะมองการติดต่อระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ว่าเป็นการยอมให้คอมพิวเตอร์เครื่องหนึ่งเรียกโพรซีเจอร์ของคอมพิวเตอร์อีกเครื่องหนึ่ง ข้อความที่มีการขนส่งในเน็ตเวิร์กจะเป็นการร้องขอหรือตอบรับการให้บริการ ข้อความร้องขอประกอบด้วยโพรซีเจอร์อาร์กิวเมนต์ (Procedure Argument) ข้อความตอบรับเป็นผลลัพธ์การทำงาน คอมพิวเตอร์ทั้งสองสื่อสารกันภายใต้โปรโตคอลเดียวกัน นิยมใช้ในระบบไคลเอ็นต์เซิร์ฟเวอร์ การทำงานแบบนี้จะเป็นแบบ 1 ผลลัพธ์ต่อ 1 คำสั่ง จึงทำให้ในเน็ตเวิร์กเต็มไปด้วยข้อความที่ส่งผ่านกันซึ่งอาจเกิดปัญหาข้อมูลผิดพลาดหรือสูญหายได้

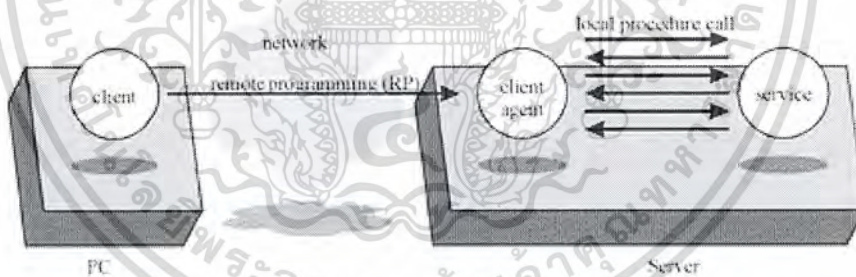
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการสื่อสารแบบรีโมทโพรซีเจอร์คอล (Remote Procedure Call)

- การสื่อสารระหว่างกัน โดยใช้เอเจนต์ช่วย (Agent Enhanced Approach)

ใช้วิธีการติดต่อที่เรียกว่ารีโมทโปรแกรมมิ่ง (Remote Programming (RP)) โดยRPจะมองการติดต่อระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ว่าเป็นการที่คอมพิวเตอร์เครื่องหนึ่งส่งผ่านโพรซีเจอร์ที่จะทำงานไปให้ยังคอมพิวเตอร์อีกเครื่องหนึ่ง ข้อตกลงระหว่างคอมพิวเตอร์ทั้งสองอยู่ในรูปของภาษา กล่าวคือไคลเอ็นต์จะส่งเอเจนต์ไปยังเซิร์ฟเวอร์ เอเจนต์จะบรรจุโพรซีเจอร์ที่จะทำตามคำร้องขอของไคลเอ็นต์ ฉะนั้นไม่ว่าจะทำงานกี่ครั้งก็ต้องการการส่งข้อความที่เป็นเอเจนต์เพียงครั้งเดียว ข้อเด่นคือการติดต่อกันระหว่างสองเครื่องโดยที่ไม่ต้องผ่านเน็ตเวิร์คหากผ่านขั้นตอนของการขนส่งเอเจนต์แล้ว



รูปที่ 2.3 แสดงการสื่อสารแบบรีโมทโปรแกรมมิ่ง Remote Programming

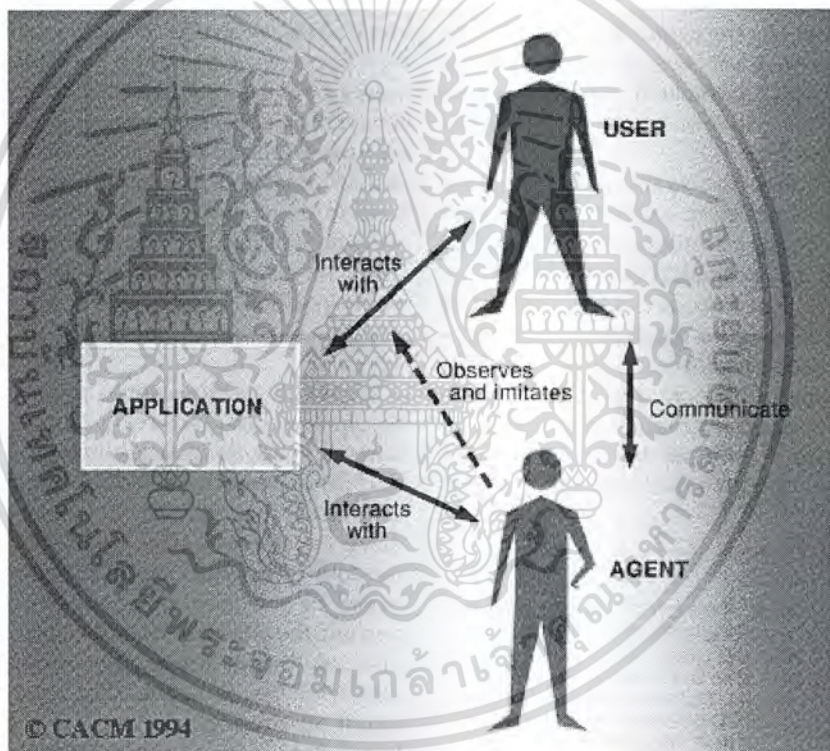
ข้อได้เปรียบของ RP เปรียบเทียบกับ PRC

- ด้านประสิทธิภาพ กล่าวคือ ไคลเอ็นต์จะส่งเอเจนต์ไปยังเซิร์ฟเวอร์แทนที่จะส่งคำสั่งไปทำงานโดยตรง การทำงานจะเป็นการทำงานภายในคอมพิวเตอร์ฝั่งเซิร์ฟเวอร์ แทนที่การทำงานระยะไกล ในเน็ตเวิร์คจึงมีความหนาแน่นของข้อมูลน้อยกว่า
- ด้านความยืดหยุ่น กล่าวคือ การใช้เอเจนต์ทำให้ผู้ผลิตไคลเอ็นต์ซอฟต์แวร์สามารถเพิ่มเติมฟังก์ชันจากที่มีอยู่ในเซิร์ฟเวอร์ซึ่งRPCทำไม่ได้เพราะเป็นเพียงการส่งคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 แนวทางการสร้างเอเจนต์

คอมพิวเตอร์เข้ามามีบทบาทสำคัญในการช่วยลดเวลาในการทำงาน เพิ่มปริมาณกิจกรรมประจำวันให้สูงขึ้น ในขณะที่ยังมีผู้ใช้ที่ยังไม่รับการฝึกฝนในเรื่องการติดต่อกับคอมพิวเตอร์อยู่เป็นจำนวนมาก การติดต่อแบบไดเรกต์มานิปูเลชัน (Direct manipulation) ที่ให้ผู้ใช้เป็นผู้ริเริ่มสั่งงานและเป็นผู้คอยตรวจสอบเหตุการณ์ที่เกิดขึ้นนั้น จำเป็นที่จะต้องเปลี่ยนไป การติดต่อมีลักษณะแบบอินไดเรกต์มานิเม้นต์ (Indirect management) ได้ถูกนำมาใช้แทน มีลักษณะคือจะเป็นการทำกระบวนการร่วมกันของผู้ใช้และเอเจนต์ มีการสื่อสารกันในการทำงานและตรวจสอบเหตุการณ์ต่างๆระหว่างกันและกัน โดยที่เอเจนต์เป็นเหมือนผู้ช่วยส่วนบุคคลที่ร่วมทำงานไปด้วยกับผู้ใช้ และผู้ช่วยนั้นจะคอยเพิ่มประสิทธิภาพของตนโดยเรียนรู้จากความชอบ ความสนใจ และ สิ่งที่ทำจนเคยชินของผู้ใช้ และเรียนรู้จากเอเจนต์อื่นๆในชุมชนนั้น



รูปที่ 2.4 แสดงการติดต่อกันระหว่างเอเจนต์, ผู้ใช้ และ แอปพลิเคชัน

แนวทางในการสร้างเอเจนต์สามารถทำได้หลายแนวทาง ดังนี้

แนวทางในการสร้างเอเจนต์แรกเรียกว่าเซมิออโตโนมัส (Semi autonomous) จะทำการรวบรวมกฎเกณฑ์ต่างๆของผู้ใช้ (User programmed rules) ที่ใช้ในการโปรเซสข้อมูลที่เกี่ยวข้องกับงานนั้นไว้ ตัวอย่างเช่น ในเอเจนต์ที่ช่วยในการจัดเรียงเมลล์ ผู้ใช้สามารถกำหนดกฎในการโปรเซสเมลล์ที่เข้ามาและเรียงเข้าไปโฟลเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างๆกันได้ กฎเหล่านี้ทำให้เอเจนต์สามารถจัดการงานให้ผู้ใช้โดยที่ผู้ใช้ไม่ต้องร้องขอการบริการเลย ความยากในการใช้วิธีนี้คือการจัดการให้เป็นมาตรฐานนั้นต้องการความเข้าใจในงานอย่างลึกซึ้ง โดยจะต้องเป็นผู้ริเริ่มในการสร้างและสอนความรู้แก่เอเจนต์ และต้องปรับปรุงตลอดเวลาหากความต้องการหรือลักษณะการทำงานนั้นเปลี่ยนแปลงไป แต่วิธีนี้จะสร้างความเชื่อถือในตัวเอเจนต์ได้มากเพราะผู้ใช้เป็นผู้กำหนดกฎและ/หรือทำการโปรแกรมขึ้นมาใช้งานเอง

แนวทางที่สองคือการใช้วิธีฐานความรู้ (Knowledge based) วิธีนี้จะทำการให้ข้อมูลเอเจนต์ด้วยโดเมนโมเดลและโมเดลของผู้ใช้ ซึ่งจะเป็นการขยายความรู้ภูมิหลังในเรื่องนั้นๆให้กับเอเจนต์ เอเจนต์จะใช้ความรู้ภูมิหลังที่ได้รับวิเคราะห์ออกมาเป็นแผนการและหนทางในการทำงาน ตัวอย่างเช่น เอเจนต์ที่ชื่อ "UCEgo" ที่ออกแบบมาเพื่อแก้ปัญหาบนระบบปฏิบัติการยูนิกซ์ UCEgoจะมีฐานความรู้ขนาดใหญ่เกี่ยวกับยูนิกซ์ ปัญหาของวิธีนี้คือ ความต้องการพื้นที่การทำงานขนาดใหญ่ทำให้สิ้นเปลืองขนาดของฐานข้อมูล ความยากลำบากในการวิเคราะห์หาข้อมูลจำนวนมากให้ได้ทั้งหมดเพราะข้อมูลบางอย่างอาจได้มาจากแอสซันที่คาดไม่ถึงของผู้ใช้ การค้นหาความรู้นี้อาจใช้วิศวกรเข้ามาช่วยวิเคราะห์ ผู้ใช้จึงมีความรู้สึกที่เอเจนต์เข้าใจยากและควบคุมไม่ได้ ผู้ใช้ไม่ทราบถึงข้อจำกัดและแนวทางที่เอเจนต์ทำงาน

แนวทางที่สามเป็นการสร้างเอเจนต์ที่ทำงานด้วยเทคนิคการเรียนรู้จากแมชชีน (Machine learning) การทำงานด้วยเทคนิคนี้ตั้งอยู่บนสมมติฐานที่ว่าต้องมีความสามารถในการทำการโปรแกรมตัวเอง กล่าวคือ ในสถานะที่แน่นอน เอเจนต์ต้องสามารถรับความรู้ที่จำเป็นในการช่วยเหลือผู้ใช้ โดยเอเจนต์ที่ดีจะเริ่มจากความรู้ภูมิหลังที่น้อยที่สุดเท่าที่จะเป็นไปได้ และเรียนรู้เพิ่มเติมจากผู้ใช้และ/หรือเอเจนต์อื่นๆด้วยตัวเอง โดยมีเงื่อนไขดังนี้ (1) การใช้แอปพลิเคชันต่างๆ จะต้องมีพฤติกรรมที่กระทำซ้ำๆเพื่อให้เอเจนต์ได้มีโอกาสในการเรียนรู้จากแอสซันที่เกิดขึ้น (2) สำหรับพฤติกรรมที่เกิดขึ้นจะต้องแตกต่างกันถ้าเป็นผู้ใช้คนละคนกัน วิธีการนี้จะมีลักษณะคล้ายว่าเอเจนต์เป็นผู้ช่วยผู้ใช้ ในตอนแรกเอเจนต์ยังไม่คุ้นเคยกับลักษณะนิสัยและความชอบ ความต้องการของผู้ใช้จึงไม่สามารถช่วยอะไรได้มาก พอเวลาผ่านไปผู้ช่วยคุ้นเคยกับแนวทางการทำงานและมีประสบการณ์มากขึ้นก็จะสามารถทำงานแทนได้ แนวทางนี้มีข้อได้เปรียบกว่าสองแนวทางก่อนคือ ต้องการความช่วยเหลือจากผู้ใช้และผู้พัฒนาน้อยลง สามารถปรับเปลี่ยนให้ใช้ได้กับผู้ใช้แต่ละคนได้

2.2.5 การเรียนรู้ข้อมูลของเอเจนต์

การเรียนรู้ข้อมูลของเอเจนต์ ทำได้ 4 ทางคือ

1. การจับจ้องการทำงานของผู้ใช้อย่างต่อเนื่อง เอเจนต์จะคอยดูการทำกิจกรรมของผู้ใช้ในช่วงระยะเวลา นาน หากกฎเกณฑ์และรูปแบบที่เกิดขึ้นบ่อยๆ แล้วทำให้ส่วนนั้นกลายเป็นส่วนที่ทำงานอัตโนมัติโดยเอเจนต์แทนที่จะต้องทำซ้ำๆ โดยผู้ใช้ ตัวอย่างเช่น เมล์เอเจนต์สังเกตเห็นว่า ผู้ใช้มักเก็บเมล์ที่ส่งมายังเมลล์ลิสต์หนึ่งเข้าไปในโฟลเดอร์หนึ่งอยู่เป็นประจำ เอเจนต์ก็อาจทำงานนี้แทนถ้ามีเมลล์ส่งมาที่เมลล์ลิสต์ดังกล่าวในครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

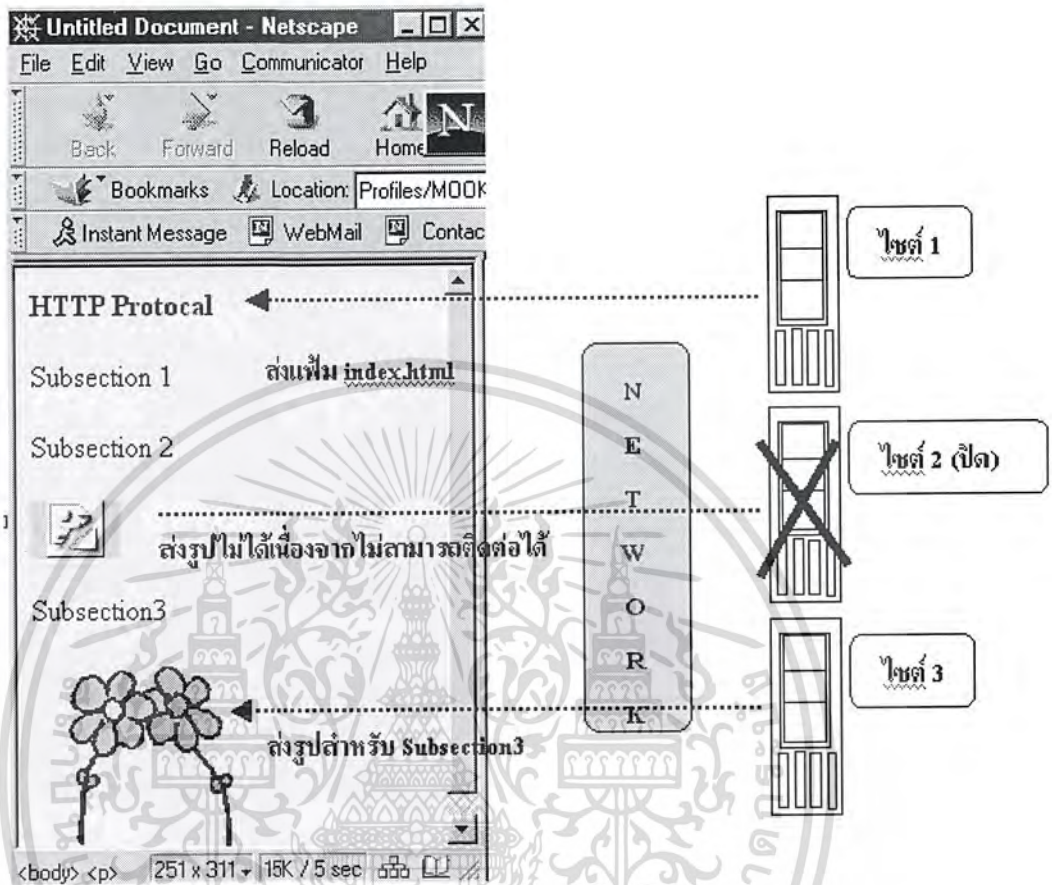
2. เรียนรู้โดยตรง หรือ โดยอ้อม จากข้อมูลป้อนกลับ(feed back)ของผู้ใช้ ข้อมูลป้อนกลับทางอ้อมเกิดขึ้นเมื่อผู้ใช้ไม่ได้ทำตามคำแนะนำของเอเจนต์แต่ไปทำแอชชันอย่างอื่นแทน ซึ่งเป็นการบอกเป็นนัยว่าผู้ใช้เปลี่ยนแปลงความต้องการในการทำกิจกรรมนั้นแล้ว เรียกข้อมูลป้อนกลับนี้ว่า ข้อมูลป้อนกลับในทางลบ เช่น การปฏิเสธไม่อ่านบทความที่เอเจนต์ทำให้แต่ไปเปิดอ่านบทความอื่นที่เอเจนต์ไม่ได้ทำให้
3. เรียนรู้จากตัวอย่างที่ผู้ใช้แสดงให้เห็น ผู้ใช้สามารถฝึกฝนเอเจนต์โดยการกำหนดตัวอย่างจากสมมติฐานของเหตุการณ์และระบุว่าให้ทำอะไรกับกรณีนั้น ให้เอเจนต์บันทึกเอาไว้ ตัวอย่างเช่น ผู้ใช้สอนให้เอเจนต์ที่ช่วยกรองข่าวสารคัดเลือกอาบทความที่มีคำว่า KMITL มาเก็บไว้
4. เรียนรู้จากการขอคำแนะนำจากเอเจนต์ที่มีประสบการณ์มากกว่าที่ให้ความช่วยเหลือผู้ช่วยคนอื่นที่จัดการงานๆเดียวกันอยู่ โดยเอเจนต์ที่ขอคำแนะนำจะแสดงสถานะการณ่นั้นและถามว่าเอเจนต์อื่นในสถานะการณ่นั้นจะอย่างไร ตัวอย่างเช่น เมล์เอเจนต์ได้รับข้อความจากบุคคลสำคัญท่านหนึ่ง จึงถามเอเจนต์อื่นว่าควรทำอะไร เอเจนต์อื่นๆเห็นว่าเมลล์ที่ส่งมามีระดับความสำคัญสูงให้แสดงผลทันที เอเจนต์นั้นก็แสดงผลเมลล์ดังกล่าวทันทีแม้ว่าผู้ใช้อาจจะไม่ได้ทำการตรวจสอบเมลล์อยู่ก็ตาม การเรียนรู้จากเอเจนต์อื่นนั้นนอกจากจะเรียนรู้อย่างเฉียบจากเอเจนต์ทั้งหมดในชุมชน อาจจะเลือกเรียนรู้จากเอเจนต์ที่มีความเชี่ยวชาญในด้านนั้นโดยเฉพาะก็ได้

2.3 ไฮเพอร์เท็กซ์ ทรานส์เฟอร์ โพรโตคอล (HyperText Transfer Protocol หรือ HTTP)

2.3.1 บทนำ

เฮชทีทีพี (HTTP) ย่อมาจาก ไฮเพอร์เท็กซ์ ทรานส์เฟอร์ โพรโตคอล (HyperText Transfer Protocol) เป็นโพรโตคอลเกี่ยวกับการจัดการเครือข่ายที่ใช้ในการติดต่อสื่อสาร และรับส่งข้อมูล ภายใต้ระบบเว็บซึ่งคือไฮเพอร์เท็กซ์ หรือ เว็บเพจ นั่นเอง กล่าวคือ เฮชทีทีพี เป็นข้อกำหนดหรือวิธีในการติดต่อสื่อสารหรือจัดการข้อมูลประเภทไฮเพอร์เท็กซ์ หรือมักเรียกว่าไฮเพอร์มีเดีย ทั้งนี้เพราะ ตัวรูปแบบเอกสารเปลี่ยนไปจากเดิมที่เป็นเอกสารแบบข้อความ (Text) เชื่อมโยงกันเป็นโครงข่าย แต่ปัจจุบันข้อมูลอาจเป็นได้ทั้งภาพ เสียง หรืออื่นๆ (พวกมัลติมีเดีย (Multimedia)) โดยรูปแบบจะเป็นแบบ “คอนเนคชัน โอเรียนเต็ด” (Connection Oriented) และการทำงานพื้นฐานจะมีรูปแบบเป็นลักษณะแบบ “ทรานส์แอ็กชัน โอเรียนเต็ด” (Transaction Oriented) คือ จะอาศัยหลักการง่ายๆของไคลเอนต์/เซิร์ฟเวอร์ (Client/Server) ในการร้องขอบริการ ซึ่งข้อมูลต่างๆที่เป็นส่วนประกอบของโฮมเพจที่ร้องขอบริการ เช่น ภาพ, เสียง, หรืออื่นๆ จะมีการเปิดการติดต่อใหม่เป็นอิสระแก่กัน ซึ่งจะขออธิบายจากตัวอย่าง ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงการเปิดการติดต่ออยู่ในการร้องขอโฮมเพจ

จากรูปแสดงการร้องขอข้อมูลของไคลเอ็นต์ ไปยังไซต์ 1, 2 และ 3 จะเห็นว่าข้อมูลที่ส่งกลับมาจากแต่ละไซต์ไม่ขึ้นแก่กันทั้ง 3 ไซต์ ไซต์ 1 และ 3 เว็บเซิร์ฟเวอร์มีการส่งข้อมูลกลับมายังไคลเอ็นต์ได้ แต่ไซต์ 2 ไม่มีข้อมูลกลับมา ซึ่งหลักการที่สำคัญในการทำงานเป็นเรื่องของการร้องขอของไคลเอ็นต์ และการตอบกลับของเซิร์ฟเวอร์นั้น ใช้หลักการของ ไคลเอ็นต์/เซิร์ฟเวอร์ ทั้งนี้เพราะเอชทีทีพี (HTTP) ก็เป็นโพรโตคอลที่ทำงานแบบไคลเอ็นต์/เซิร์ฟเวอร์ และด้วยเทคนิคกับวิธีการที่กล่าวข้างต้นจะเห็นว่าข้อดีคือ ทำให้งานแต่ละชิ้นเป็นอิสระต่อกัน ดังนั้นหากมีส่วนใดเสียหรือมีปัญหาในการติดต่อสื่อสารไม่ว่าจะด้วยสาเหตุใดก็ตามจะไม่ส่งผลกระทบต่ออีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การทำงานของ เชนทีทีพี (HTTP)

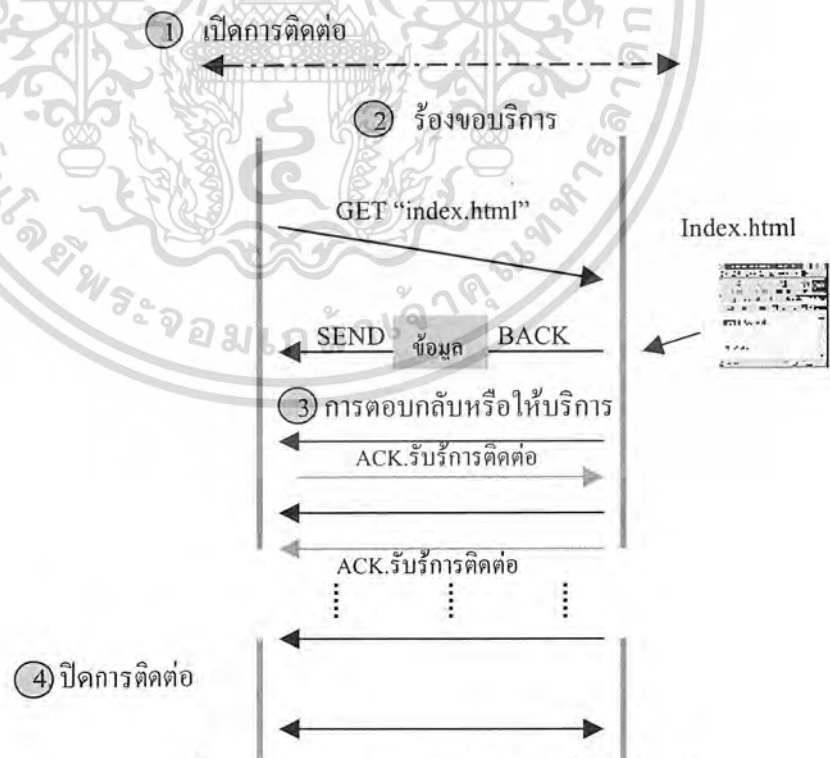
2.3.2.1 หลักการทำงานและวิธีการของไคลเอ็นต์/เซิร์ฟเวอร์

การทำงานของ เชนทีทีพี โพรโตคอล จะแบ่งออกเป็น 3 ส่วนหลักๆ คือ

1. ไคลเอ็นต์ (Client) : เป็นผู้ขอใช้บริการ
2. เซิร์ฟเวอร์ (Server) : เป็นผู้ให้บริการแก่ไคลเอ็นต์
3. ระบบเครือข่าย (Network)

ตามหลักการทำงานพื้นฐานของวิธีการไคลเอ็นต์/เซิร์ฟเวอร์ ฝั่งไคลเอ็นต์จะใช้งานบราวเซอร์โปรแกรม ติดต่อกับเว็บเซิร์ฟเวอร์ ซึ่งก็คือผู้ให้บริการเว็บ โดยโปรแกรมดังกล่าวจะเรียกว่า เชนทีทีพีดี (HTTPD ย่อมาจาก HyperText Transfer Protocol Daemon) ปกติแล้วจะทำงานที่พอร์ต (Port) 80 โดยมีขั้นตอนการทำงานดังนี้

1. การสร้างการติดต่อระหว่าง ไคลเอ็นต์ และ เซิร์ฟเวอร์ (Establish Connection)
2. การร้องขอบริการ (Request) คือ ไคลเอ็นต์ ร้องขอบริการไปยังเซิร์ฟเวอร์
3. การตอบกลับการให้บริการ (Response) คือ การที่เซิร์ฟเวอร์ ให้บริการซึ่งแล้วแต่ชนิดของการร้องขอบริการของไคลเอ็นต์
4. การปิดการติดต่อ (Terminate) คือ การยกเลิกการทำงานในกรณีที่มีการร้องขอบริการและการให้บริการเสร็จสิ้นสมบูรณ์

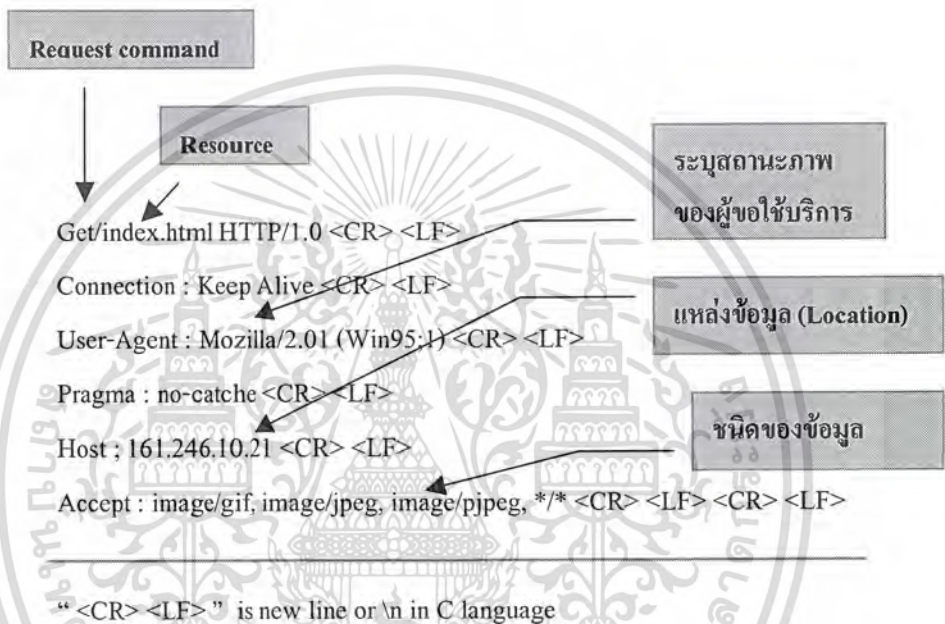


รูปที่ 2.6 แสดงการทำงานของไคลเอ็นต์/เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.2 การร้องขอบริการ (Request)

สำหรับเซชที่ทีพี (HTTP) จะมีคำสั่งหลักๆในการจัดการดังนี้ ออฟชั่น (OPTION), เฮด (HEAD), พูท (PUT), ดีลิต (DELETE), เทรซ (TRACE), เก็ท (GET) และ โปสต์ (POST) แต่คำสั่งหลักๆสำหรับผู้ขอใช้บริการมักใช้บ่อยคือ เก็ท (GET), โปสต์ (POST), และ เฮด (HEAD) โดยรายละเอียดการใช้คำสั่ง สำหรับการร้องขอนั้นจะมีการระบุรายละเอียดไว้ในตัวของ เซชที่ทีพี (HTTP) โพรโตคอล



รูปที่ 2.7 แสดงตัวอย่างการร้องขอบริการของ เซชที่ทีพี (HTTP)

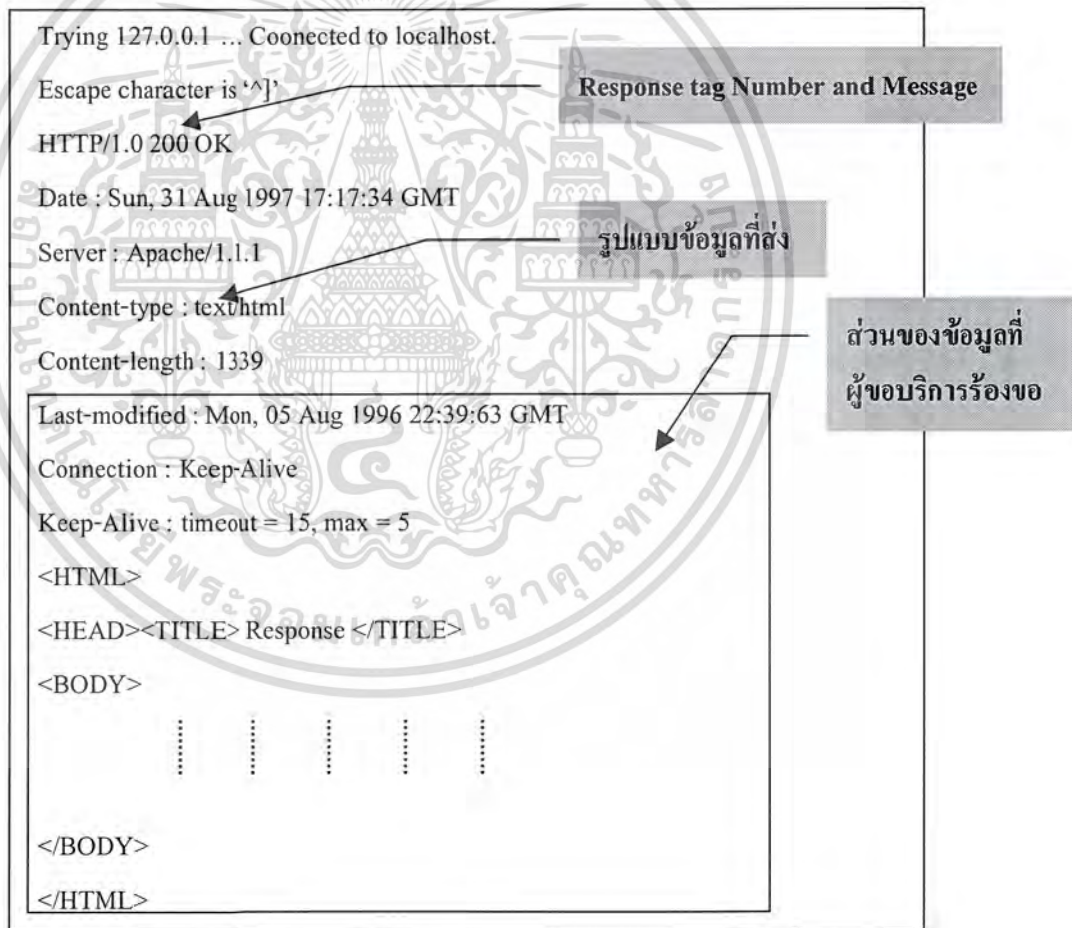
จากรูป การร้องขอบริการข้างต้นจะเป็นการเรียกขอโฮมเพจที่ชื่อ “index.html” โดยตรงนี้อาจจะระบุแบบส่วนขยาย URL ได้ เช่น /~s0010693/index.html เป็นต้น ส่วนสำคัญถัดมาของการร้องขอบริการ คือส่วนของ ยูสเซอร์ เอเจนต์ (User Agent) เป็รการระบุรายละเอียดเกี่ยวกับสถานะภาพของไคลเอ็นต์ เช่น โปรแกรมที่ใช้งานเป็นบราวเซอร์ของ เน็ตสเคป (Netscape) หรือ ไออี (IE) ใช้งานกับวินโดวส์ 95 (Windows 95) เป็นต้น

ส่วนต่อมาก็คือโฮสต์ (Host) ซึ่งจะเป็นไชด์ปลายทางและรายละเอียดนี้จะสัมพันธ์กับส่วนเพิ่มเติมของ ยูอาร์แอล (URL) ดังเช่น จากตัวอย่างระบุยูอาร์แอล (URL) เท่ากับ “<http://161.246.10.21/index.html>” หรือ “<http://www.kmitl.ac.th/index.html>” ในกรณีที่

สองจะมีการนำชื่อโฮสต์ “www.kmitl.ac.th” ไปแปลงเป็นค่า ไอพี แอดเดรส (IP Address) เท่ากับ “161.246.10.21” ค่าโฮสต์ = 161.246.10.21 และ เก็ท (GET) จะได้ค่าเท่ากับ /index.html

2.3.2.3 การตอบกลับ/การให้บริการ (Response/Reply)

ในการตอบกลับการให้บริการของ เอชทีทีพี (HTTP) นั้นจะมีรูปแบบในการใช้งานเหมือน โพรโตคอลอื่นๆ ตามหลักการของไคลเอ็นต์/เซิร์ฟเวอร์ทั่วไป คือ จะมีการส่งค่ากลับ นำด้วยหมายเลขซึ่งเรียกว่า “ เรสพอนส์ แท็กส์ นัมเบอร์ (Response Tags Number) หรือ สเตตัส โค้ด (Status Code)” และจะตามด้วยรายละเอียดข้อความซึ่งอธิบาย จากนั้นส่วนสุดท้ายที่อาจจะมีส่วนคือตัวของ ข้อมูลจริงๆ ในกรณีที่มีการขอข้อมูล เช่น จากการร้องขอข้างต้นเราจะได้รับข้อมูลที่เว็บเซิร์ฟเวอร์ หรือเซิร์ฟเวอร์ส่งกลับมาให้ดังนี้



รูปที่ 2.8 แสดงข้อมูลตอบกลับของเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลขการตอบกลับ (Response) จะเป็นค่ามาตรฐาน แต่ข้อมูลที่ตามมานั้นอาจจะไม่เหมือนกันก็ได้ ทั้งนี้แล้วแต่เนื้อหาของผลิตภัณฑ์ของเว็บเซิร์ฟเวอร์ ดังนั้นในการเขียนโปรแกรมประเภทไคลเอ็นต์/เซิร์ฟเวอร์ หรือหากจะเขียนบราวเซอร์โปรแกรม จะต้องอาศัยการตรวจสอบการทำงานของไคลเอ็นต์/เซิร์ฟเวอร์ จากหมายเลขดังกล่าว ไม่ควรใช้ข้อความเพราะอาจจะผิดพลาดได้

ตัวเลขทุกหลักของเรสปอนส์ แท็กส์ นัมเบอร์ (Response Tags Number) ล้วนมีความหมายเริ่มที่หลักแรก จะเป็นการแสดงว่ามีประสิทธิภาพของการร้องขอ และตอบกลับว่าเป็นเช่นไร เช่น “200 OK” หรือ “500 Error” เป็นต้น ตัวเลขหลักที่สองจะแจ้งให้ทราบถึงชนิดของการทำงาน เช่น หากเกิด Error เป็นชนิดไหน และหลักสุดท้ายเป็นส่วนย่อยในประเภทของการทำงานนั้นๆ เป็นการขยายในส่วนรายละเอียดของหลักที่สอง

Response Tags Number	รายละเอียด
100	Continue
101	Switching
200	OK
201	Created
202	Accepted
203	Non-Authoritative information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Move Temporarily
303	See Other
304	Not Modified
305	Use Proxy
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

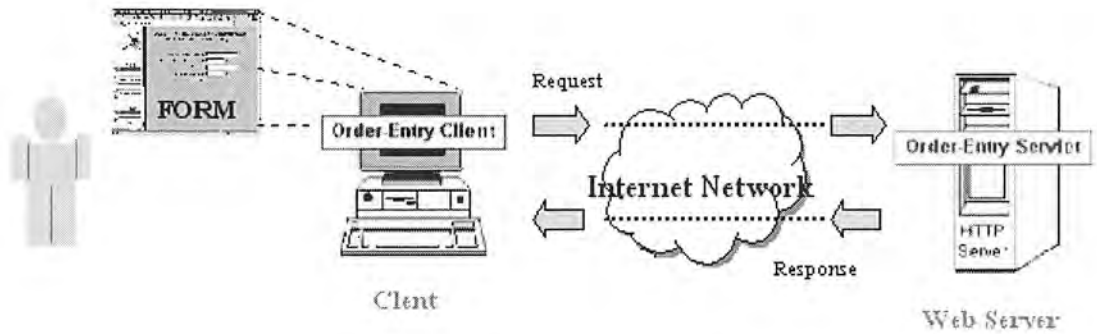
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Time-out
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request – URL Too Large
415	Unsupported Media Type
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Time-out
505	HTTP Version not supported

ตารางที่ 2-1 แสดงรายละเอียดของหมายเลขสถานะการทำงานของเฮททีทีพี (HTTP) โพรโทคอลที่ควรทราบ

2.3.2.4 การเชื่อมต่อระหว่างผู้ให้บริการและผู้ให้บริการในระบบเครือข่าย (Network)

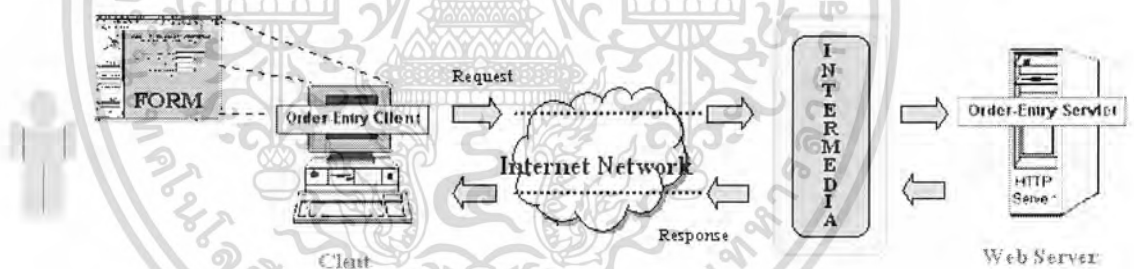
ในระบบเครือข่าย เวิร์ดไวด์เว็บ มีการเชื่อมต่อระหว่างผู้ให้บริการและผู้ให้บริการอยู่ 2 ลักษณะ คือ

1. การเชื่อมต่อโดยตรง : ลักษณะการทำงานคือ ผู้ขอใช้บริการจะติดต่อกับเซิร์ฟเวอร์ หรือ ผู้ให้บริการโดยตรง ซึ่งส่วนไคลเอ็นต์จะเรียกว่า ยูสเซอร์ เอเจนต์ (User Agent) โดยมี บราวเซอร์ทำหน้าที่นี้ให้ และส่วนเซิร์ฟเวอร์จะเรียกว่า “ออริจิน (Origin)” จะทำงานกับ บราวเซอร์หรือ ยูสเซอร์ เอเจนต์ นี้โดยตรง



รูปที่ 2.9 แสดงการเชื่อมต่อโดยตรง

2. การเชื่อมต่อผ่านตัวกลาง : ลักษณะการติดต่อแบบนี้ส่วนของ ยูสเซอร์ เอเจนต์ (User Agent) ไม่สามารถติดต่อกับ ออริจิน (Origin) ได้โดยตรง นั่นคือ ต้องติดต่อผ่านตัวกลางทุกครั้งที่มีการร้องขอบริการ และการตอบสนองก็ต้องผ่านตัวกลางเช่นกัน ดังนั้น การร้องขอหรือการตอบสนองจะมีลักษณะเหมือนลูกโซ่ โยงผ่านเป็นช่วงๆเรียกว่า รีเควสเชน /เรสพอนส์เชน (Request Chain / Response Chain)



รูปที่ 2.10 แสดงการเชื่อมต่อผ่านตัวกลาง

2.3.2.5 ประเภทของตัวกลาง (Intermedia)

ประเภทของตัวกลางตามข้อกำหนดของเฮชทีทีพี (HTTP) มี 3 ประเภท คือ

1. Tunnel : ทำหน้าที่เชื่อมต่อเท่านั้น อาจจะไว้เพื่อความประสงค์อะไรบางอย่าง แต่ตัวกลางนี้จะไม่มีหน้าที่หรืออำนาจในการเปลี่ยนแปลงข้อมูลที่วิ่งผ่าน
2. Proxy : ส่วนนี้สามารถปรับปรุงรายละเอียด มีการประยุกต์ใช้งานได้ทั้ง 2 ส่วน คือ โคลเอ็นต์และเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

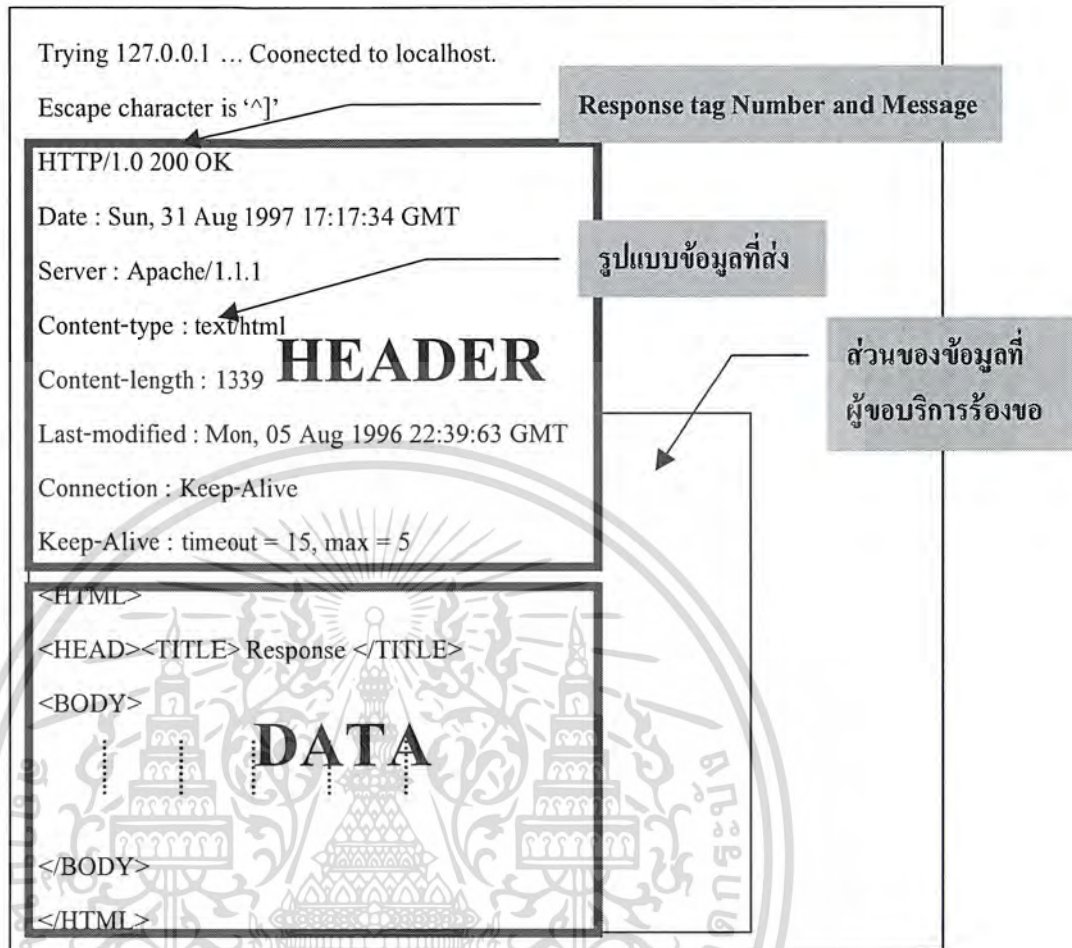
3. Gateway : ส่วนนี้มักทำหน้าที่เชื่อมต่อ ในกรณีที่ไม่สามารถติดต่อหรือใช้งานเชื่อมต่อกับตัวเว็บเซิร์ฟเวอร์ได้โดยตรง

2.3.3 เฮดเดอร์ (Header) ของการแสดงผล

2.3.3.1 การจัดการเฮดเดอร์ (Header) ของการแสดงผล

เฮชทีทีพี (HTTP) โพรโตคอลอาศัยหลักการของ ไมม์ (MIME-Multipurpose Internet Mail Extensions) ในการจัดการตัวรูปแบบข้อมูลที่ใช้ รับ-ส่งในระบบเครือข่าย โดยในการใช้งานเฮดเดอร์ หรือ ไมม์ เฮดเดอร์ (MIME Header) นั้นได้ระบุรายละเอียดไว้ว่าสามารถกำหนดรายละเอียดต่างๆ ที่ต้องการจะใช้งานหรือบ่งบอกถึงรูปแบบข้อมูล โดยแยกรายการต่างๆ ของรายละเอียดด้วย “<CR> <LF>” ซึ่งก็คือ “แคร่เรีย รีเทิร์น (Carriage Return)” และ “ไลน์ ฟีด (Line feed)” ตามลำดับ ในกรณีการโปรแกรมนั้นคือค่า “\n” หรือ “New Line” ใช้ในการแยกรายละเอียดของเฮดเดอร์ของผลลัพธ์นั้นๆ โดยใช้จำนวนหนึ่งชุด และเมื่อจบส่วนเฮดเดอร์ของผลลัพธ์จะใช้ “<CR> <LF> <CR> <LF>” จำนวนสองชุดต้องติดกัน

เมื่อเบราว์เซอร์โปรแกรมที่ฝั่งไคลเอ็นต์รับค่าจะตีความจาก “\n” (<CR> <LF>) ว่าส่วนหัวของรูปแบบระบุไว้อย่างไรและสิ้นสุดเฮดเดอร์ที่บรรทัดไหน จากนั้นส่วนที่จะตามมาก็คือ ตัวข้อมูลจริงๆ เบราวเซอร์จะนำค่าของเฮดเดอร์มาประมวลผลเพื่อที่จะแสดงข้อมูลที่ตามมาได้อย่างเหมาะสม



รูปที่ 2.11 ภาพแสดงส่วนเฮดเดอร์ และส่วนข้อมูล

2.3.3.2 รายละเอียดเฮดเดอร์ (Header) ของเฮททีทีพี (HTTP) โพรโทคอล

รายละเอียดเฮดเดอร์ทั้งหมดของเฮททีทีพี (HTTP) มีรายละเอียดดังต่อไปนี้

เฮดเดอร์ (Header)	รายละเอียด (Description)
Content-length	บอกขนาดของข้อมูลที่ส่งมาด้วยว่ามีขนาดเท่าไรหน่วยเป็น ไบต์ (byte)
Content-type	บอกชนิดของข้อมูลที่ส่งมาว่าเป็นแบบใด
Expires	บอกวันและเวลาสำหรับระบุว่าเอกสารจะหมดอายุเมื่อไร บราวเซอร์จะได้อ่านหรือทราบว่าควรรีโหลด (Reload) ใหม่หรือไม่เมื่อไหร่
Location	บอกให้เว็บเซิร์ฟเวอร์ทำการเปลี่ยนทิศทางส่งผลลัพธ์ (Redirection)
Pragma	จะเป็นการกำหนดเรื่องของแคช (Caching) ว่าจะใช้หรือไม่ใช้กับข้อมูลที่ส่งมานั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Status	บอกสถานะของการขอใช้บริการ
Refresh	เป็นการระบุให้โคลเอนต์ทำการรีโหลด ข้อมูลพิเศษใหม่
Set-Cookie	เป็นการระบุให้โคลเอนต์เก็บรายละเอียดเกี่ยวกับ ข้อมูลพิเศษ

ตารางที่ 2-2 ตารางแสดงรายละเอียดเซดเคอร์ของ เฮททีทีพี (HTTP) โพรโทคอล

หมายเหตุ Refresh และ Set-Cookie เป็นเซดเคอร์ที่ออกโดยไนเน็ตสเคป (Netscape) ซึ่ง ข้อมูลพิเศษ คือ ข้อมูลที่ไม่ถือว่าเป็นข้อมูลที่เกี่ยวข้องกับเอกสารหรือที่โคลเอนต์ต้องการ แต่เป็นข้อมูลที่ใช้งานบางอย่างพิเศษ ได้แก่ ข้อมูลที่เกี่ยวข้องกับวิธีการทำงานของคุกกี้ (Cookie)

2.4 คอมมอน เกทเวย์ อินเตอร์เฟส (Common Gateway Interface หรือ CGI)

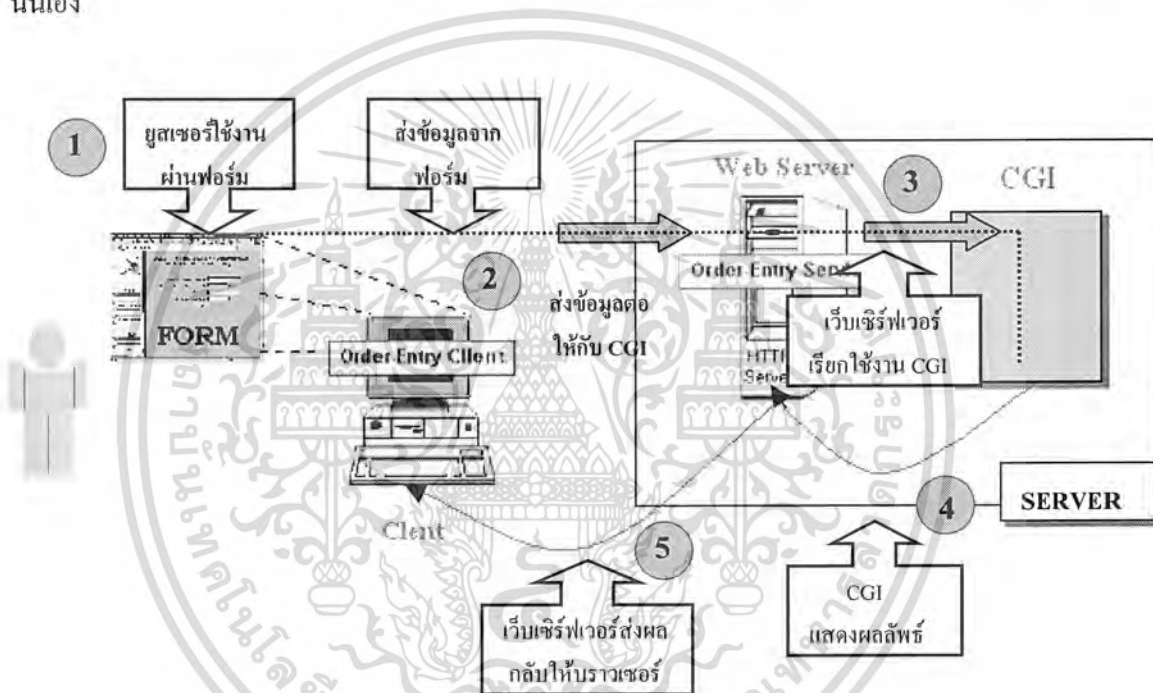
2.4.1 บทนำ

เมื่อเกิดระบบเครือข่ายเวิร์ลไวด์เว็บ ใช้งานจนเป็นที่นิยมดังเช่นปัจจุบัน หลายๆ ไซด์ (Web site) เริ่มต้องการนำเสนอข้อมูลภายในองค์กรที่เคยใช้งานกับ โปรแกรมประยุกต์ของตนภายในองค์กรผ่านเว็บเพจหรือโฮมเพจ (Homepage) ของตน จึงเกิดปัญหาว่าจะสามารถทำอย่างไร ทั้งนี้เพราะทั้งสองแอปพลิเคชันอยู่คนละส่วนกัน และวิธีการทำงานก็แตกต่างกันอย่างสิ้นเชิง ทางออกก็คือการพัฒนาแอปพลิเคชันในลักษณะเหมือนโปรแกรมประยุกต์ที่องค์กรใช้งานอยู่ โดยอาศัยหลักการของ ซีจีไอ (CGI-Common Gateway Interface) ในการพัฒนา แต่ที่ก็ยังเป็นเพียงแค่จุดเริ่มต้นของความต้องการเท่านั้น เพราะปัจจุบันจะเห็นได้ว่ามีแอปพลิเคชันหลากหลายรูปแบบบนระบบเว็บ เช่น การให้บริการส่งเพจ (Pager), การให้บริการค้นหา, การให้บริการให้ความช่วยเหลือแบบออนไลน์ เป็นต้น ซึ่งแอปพลิเคชันเหล่านี้เกิดจากความต้องการที่หลากหลาย และต่างความคิด รวมไปถึงวิสัยทัศน์ของแต่ละคนในการที่คิดประยุกต์ และร่วมสร้างกิจกรรมต่างๆ บนระบบเว็บ จนทำให้การใช้งานระบบเว็บนี้ กลายเป็นส่วนสำคัญหลักของเครือข่ายอินเทอร์เน็ตในปัจจุบันไปเสียแล้ว และด้วยความสามารถของหลักการ ซีจีไอ นี้เองทำให้หลายๆ องค์กรต้องการนำมาประยุกต์ใช้ในองค์กรจนเกิดคำที่ว่า “แอปพลิเคชันในอนาคต คือแอปพลิเคชันที่ใช้งานผ่านบราวเซอร์ หรือใช้งานภายใต้พื้นฐานเว็บ (Web-based หรือเรียกว่า Web-based Application)”

ซีจีไอ (CGI) คือ หลักการหรือวิธีการของการพัฒนาแอปพลิเคชัน ที่ทำหน้าที่เสมือนประตู (Gateway) เชื่อมโยงการติดต่อกับการทำงานอื่นๆ เพื่อให้เกิดการดำเนินงานที่หลากหลายในการใช้งาน โดยอาศัยพื้นฐานของระบบเว็บ หรือจะกล่าวได้ว่าทำงานควบคู่กับเว็บเซิร์ฟเวอร์ เพราะบราวเซอร์ไม่สามารถติดต่อส่วนอื่นๆ โดยตรงได้ เช่น จะติดต่อกับฐานข้อมูล เป็นต้น จำเป็นต้องติดต่อผ่านเว็บเซิร์ฟเวอร์ไปยังส่วนของ ซีจีไอ (CGI) โดยจะเรียกว่า “ซีจีไอ โปรแกรม” หรือ “ซีจีไอ แอปพลิเคชัน” หรือ “เว็บ แอปพลิเคชัน” ก็ได้ ด้วยเหตุนี้จึงจะเห็นว่าจริงแล้ว ซีจีไอ แอปพลิเคชัน หรือ แอปพลิเคชัน ที่พัฒนาตามแนวทาง ซีจีไอ เป็นแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภท เซิร์ฟเวอร์ แอปพลิเคชัน (Server Application) หรือ แอปพลิเคชันที่ทำงานฝั่งเซิร์ฟเวอร์ โดยมีส่วนที่มีหน้าที่ติดต่อกับ ผู้ขอใช้บริการ หรือ ไคลเอ็นต์ (Client) คือ เว็บเซิร์ฟเวอร์ และไคลเอ็นต์ใช้งานผ่านเว็บเบราว์เซอร์ ข้อดีของ เซิร์ฟเวอร์ แอปพลิเคชัน ที่เห็นได้ชัด คือ การปรับปรุงหรือเปลี่ยนเวอร์ชันจะทำได้ง่าย โดยไม่ต้องแจกจ่ายให้ผู้ใช้งานทุกครั้งแต่สามารถดูแลปรับปรุงได้ที่เซิร์ฟเวอร์โดยตรง เมื่อมีวิธีของ ซีจีไอ เกิดขึ้น ทำให้ได้เห็นรูปแบบของโฮมเพจเปลี่ยนไปจากเดิมที่เคยเป็นแค่ “สแตติก ไฮเพอร์มีเดีย ดอคูเมนต์(Static Hypermedia Dcument)” คือเอกสารที่แสดงโดยไม่มีเปลี่ยนแปลงไปเป็นเอกสารที่สามารถเปลี่ยนแปลงรูปแบบได้ ตลอดจนเห็นเป็นโฮมเพจที่สามารถโต้ตอบหรือเป็น อินเตอร์แอ็กทีฟ (Interactive) เหมือนส่วนของ อินเตอร์เฟซ (Interface) ของ ซีจีไอ แอปพลิเคชันที่แปรเปลี่ยนตลอดเหมือนกับการใช้งาน โปรแกรมประยุกต์นั่นเอง



รูปที่ 2.12 แสดงเบราว์เซอร์ติดต่อ CGI และรับผลลัพธ์จาก CGI ผ่านเว็บเซิร์ฟเวอร์

2.4.2 การทำงานของ ซีจีไอ (CGI)

การทำงานของ ซีจีไอ ก็ยังคงต้องอาศัยหลักการพื้นฐานของ ไคลเอ็นต์/เซิร์ฟเวอร์ โดยเว็บเซิร์ฟเวอร์จะเป็นผู้ติดต่อขอใช้บริการและรอรับผลลัพธ์ของ ซีจีไอ กลับมา แล้วส่งต่อให้กับยูสเซอร์ที่ใช้งานผ่านเบราว์เซอร์ โดยการทำงานเป็นดังนี้

- เมื่อเว็บเซิร์ฟเวอร์รับรายละเอียดมาตรวจสอบพบว่ามีคำสั่งเรียก ซีจีไอ ที่กำหนดในคอนฟิกไฟล์ ก็ จะทราบทันทีว่าจะต้องอ่านโปรแกรมที่ใดเรกทอรีใด ตามที่กำหนดในคอนฟิก ซึ่งไฟล์ที่ใช้งาน นั้นคือ ไฟล์ ซีจีไอ (cgi) โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หรืออาจจะทำการตรวจสอบชนิดของไฟล์ ถ้าพบว่านามสกุลเป็น “.cgi ”และพบในรายละเอียดคอนฟิกว่าเป็น ซีจีไอ แอปพลิเคชัน แสดงว่า การเรียกหรือขอใช้บริการครั้งนี้เป็นการเรียกรัน ซีจีไอ แอปพลิเคชัน เว็บเซิร์ฟเวอร์จะทำการเรียก ซีจีไอ ให้ทำงานและจะรอรับผลลัพธ์ของการทำงานด้วย
- เมื่อโปรแกรมหรือ ซีจีไอ สคริปต์ (CGI Script) ทำงานเสร็จสิ้นก็จะส่งผลลัพธ์ กลับมาให้เว็บเซิร์ฟเวอร์ จากนั้นเว็บเซิร์ฟเวอร์ก็จะทำการรับข้อมูลส่งกลับไปให้กับไคลเอ็นต์ การตอบหรือส่งผลลัพธ์ของ ซีจีไอ ให้กับเว็บเซิร์ฟเวอร์เพื่อคืนให้ไคลเอ็นต์นั้น จะมีรูปแบบของการส่งโดยทุกครั้งที่จะส่งกลับจะมีการระบุส่วนหัว (Header) ในการติดต่อสื่อสาร ว่าข้อมูลหรือรูปแบบข้อมูลที่ส่งตามมานั้น จะเป็นอย่างไร โดยวิธีการของ ไมม์ (MIME)

โดยวิธีการตอบหรือรูปแบบของการแจ้งส่วนหัวจะมีอยู่ 2 แบบ คือ

1. การตอบแบบเต็ม (Full Header) : แสดงได้ดังตัวอย่างต่อไปนี้

```
HTTP/1.0 200 OK
Date : Friday, 26 -February -99 10:10:00 GMT
Server : Apache/1.2
MIME-version : 1.0
Content-Type : text/html
Content-length : 12345
```

```
<HTML>
```

```
.....
```

```
.....
```

```
</HTML>
```

2. การตอบแบบย่อหรือแค่บางส่วน (Partial Header) : แสดงได้ดังตัวอย่างต่อไปนี้

```
Content-Type : text/html
```

```
<HTML>
```

```
.....
```

```
.....
```

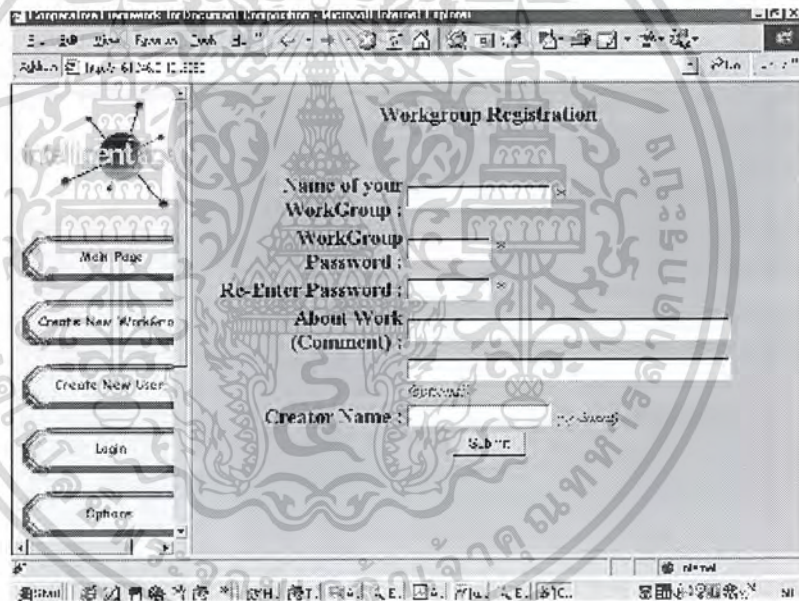
```
</HTML>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 การใช้งานฟอร์มแท็ก (Form Tag)

2.4.3.1 ฟอร์มแท็ก (Form Tag)

เซชที่เอ็มแอล (HTML) ที่สร้างโฮมเพจนั้นจะเป็นเสมือนส่วนอินเตอร์เฟซ (User Interface) ของโปรแกรม ที่ใช้งานผ่านบราวเซอร์ โปรแกรม (Browser) และ เซชที่เอ็มแอล (HTML) ได้ระบุวิธีการใช้งานหรือติดต่อกับ ซีจีไอ (CGI) โดยใช้แท็กชื่อว่า “<FORM>” แทนการเริ่มต้นการใช้งานฟอร์ม โดยมี “</FORM>” สำหรับระบุการสิ้นสุดการใช้งานของแต่ละฟอร์ม โดยแนวความคิดการทำงานเกิดจากต้องการสร้างแบบฟอร์ม เพื่อให้ยูสเซอร์ใส่ข้อมูล ใช้งานผ่านบราวเซอร์ โปรแกรม และเมื่อมีการใส่รายละเอียดแบบฟอร์มนั้นๆเรียบร้อยแล้วจะมีการคลิกปุ่ม (Submit) หรือกดปุ่มเพื่อแจ้งว่าเสร็จสิ้นการใส่ค่าแบบฟอร์ม จากนั้นก็ให้เรียกโปรแกรม ซีจีไอ ทำการประมวลผลโดยจะส่งค่าต่างๆ ในแบบฟอร์มนั้นให้กับ ซีจีไอ โปรแกรมด้วย



รูป 2.13 แสดงตัวอย่างของฟอร์มรับข้อมูล

2.4.3.2 เมธอด (Method)

เมธอด (Method) หมายถึง การระบุวิธีการใช้งาน หรือฟอร์มที่ต้องการใช้งานติดต่อกับ ซีจีไอ (CGI) โดยวิธีใด ซึ่งมีได้ 2 วิธีหลัก คือ เก็ท (GET), และ โปสต์ (POST) โดยในการใช้งานฟอร์มทุกครั้งต้องมีการ ระบุถึงวิธีการที่จะเรียกใช้ด้วยว่าเป็นแบบใด โดยการใช้งานทั้งเก็ทและโปสต์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการซั้บมิทฟอร์ม (Submit) ข้อมูลจะถูกส่งไปยังเซิร์ฟเวอร์ (Web Server) ทั้งสองวิธี แต่มีข้อแตกต่างกันดังนี้

1. เก้ท (GET) : การใช้งานเก้ท จะแสดงความต้องการในการขอข้อมูลเป็นหลัก เหมือนกับการขอบริการ (Request) การใช้งานก็่ทจะแสดงความต้องการในการร้องขอข้อมูล ดังนั้นอาจจะเป็นไปได้ ที่เมื่อเรียกใช้งานแล้วบราวเซอร์ตรวจสอบ ยูอาร์แอล (URL) แล้วเกิดพบว่าเคยได้เรียกใช้งานหรือเคยได้ทำการร้องขอไปแล้วอาจจะนำค่าโฮมเพจล่าสุดที่มีในแคช (Cache) มาให้แทน แทนที่จะมีการประมวลผลที่ ซีจีไอ โปรแกรม แล้ววางค่ามาให้ใหม่ (หรือกรณีที่บราวเซอร์ พยายามติดต่อเซิร์ฟเวอร์ แต่อาจมีปัญหาในการติดต่อ จะนำในแคชมาให้แทน) ดังนั้นหากเป็นบางแอปพลิเคชันการใช้เก้ททงไม่เหมาะสม เช่น ต้องการดูข้อมูลในฐานข้อมูลล่าสุด เป็นต้น แต่บางกรณีก็ไม่มีปัญหา เช่น ไซค์ให้บริการค้นหาข้อมูล (Search Engine) หากมีการเรียกใช้คำค้นหาเดิมที่เคยได้ค้นไปแล้วตรงนี้ถึงจะเอาค่าในแคชมาแสดงก็จะไม่มีผลกระทบต่อการใช้งาน นอกจากนี้เมื่อมีการเรียกใช้โดยวิธีเก้ท แล้วจะมีการปรับปรุงรายละเอียด ยูอาร์แอล (URL) โดยมีการนำค่าต่างๆ ที่เป็นค่าที่ผ่านมากับฟอร์มมาเพิ่มเติมให้กับยูอาร์แอล ตามข้อกำหนดที่สามารถทำได้ของหลักการยูอาร์แอล ซึ่งข้อดีตรงนี้ทำให้เราสามารถทราบได้ว่าการใช้งานฟอร์มก่อนหน้านี้มีค่าคีย์หรือตัวแปรอะไรบ้าง และมีค่าเท่ากับเท่าไร
2. โปสท (POST) : การใช้โปสทเป็นการระบุว่าต้องการส่งค่าให้ ดังนั้นบราวเซอร์จะเรียก ซีจีไอ (CGI) ประมวลผลใหม่ทุกครั้ง และซีจีไอจะต้องรับค่าไปทำใหม่ทุกครั้งด้วย รวมทั้งการส่งผลลัพธ์ให้บราวเซอร์ โปรแกรมจะทำใหม่ทุกครั้งเช่นกัน สรุป คือ การใช้โปสท บราวเซอร์จะไม่สามารถนำค่าจากการ ใช้งานเดิมในแคชมาแสดงอย่างเด็ดขาดไม่ว่ากรณีใดๆ ถึงแม้ว่ารายละเอียด ยูอาร์แอล (URL) จะเคยถูกเรียกใช้งานแล้วก็ตาม

2.5 เซิร์ฟเล็ท (Servlet)

2.5.1 บทนำ

จากเดิมที่เว็บเพจ(Web page) อยู่ในรูปแบบสแตติก เว็บ เพจ (Static web page) แต่มีอาจรองรับความต้องการของผู้ใช้อินเทอร์เน็ตได้ เนื่องจากผู้ใช้อินเทอร์เน็ตมีความต้องการให้เว็บเพจที่แสดงผลนั้นสามารถเปลี่ยนแปลงแก้ไขได้ ,สามารถโต้ตอบกับผู้ใช้ได้ ทำให้เกิดการเปลี่ยนแปลงจากสแตติก เว็บ เพจ (Static web page) ไปสู่ไดนามิก เว็บ เพจ (Dymamic web page) โดยเทคโนโลยีที่เกิดขึ้นเพื่อรองรับความต้องการเหล่านี้ก็คือ เซิร์ฟเวอร์ ไซด์ แอปพลิเคชัน (Server Side Application) โดยในระยะแรกนั้นเซิร์ฟเวอร์ ไซด์ แอปพลิเคชัน (Server Side Application) จะอยู่ในรูปแบบแนวความคิดของ ซีจีไอ (ซีจีไอ-Common Gateway Interface) ที่มีหลักการทำงานคือเว็บบราวเซอร์ (Web Browser) จะทำการส่งข้อมูล(Data)ที่เกิดจากการกระทำ(Action) ของผู้ใช้(User) เช่น การคลิกลิงค์หรือการกรอกแบบสอบถามไปยังเว็บเซิร์ฟเวอร์ (Web Server) โดยแทนที่เว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์ จะทำการส่งเพจที่เป็นสแตติก เว็บเพจ (Static web page) กลับมา เว็บเซิร์ฟเวอร์ จะทำการส่งข้อมูลเหล่านั้นต่อไปยังโปรแกรมซึ่งถูกจัดไว้ โปรแกรมดังกล่าวจะทำการประมวลผลข้อมูลที่ได้แล้วจะส่งผลกลับไปยังเว็บเซิร์ฟเวอร์ ซึ่งทางเว็บเซิร์ฟเวอร์ก็จะส่งผลที่ได้นี้กลับไปยังเว็บเบราว์เซอร์เพื่อแสดงผลการทำงานแก่ผู้ใช้ อีกทีหนึ่ง ตัวอย่างของเซิร์ฟเวอร์ ไซด์ แอปพลิเคชัน (Server Side Application) ที่เกิดจากการใช้แนวคิดของ CGI (ซีจีไอ) ยกตัวอย่างเช่น โปรแกรมนับจำนวนผู้เข้าชมเว็บเพจ(Counter), โปรแกรมการลงสมุดเยี่ยมชมเว็บเพจ(Guessbook), โปรแกรมการส่งจดหมายอิเล็กทรอนิกส์(Send mail) เป็นต้น โดย ซีจีไอ นั้นสามารถเขียนด้วยภาษาอะไรก็ได้ แต่ที่นิยมใช้มากที่สุดจะเป็นภาษาซี (C) และเพิร์ล (Perl) ซึ่งอาจจะเป็นเพราะว่า ซีจีไอ เป็นสิ่งที่มิมากับอินเทอร์เน็ตตั้งแต่ช่วงแรกๆ ดังนั้นไม่ว่าจะเป็นเว็บเซิร์ฟเวอร์ไหนก็ตามเซิร์ฟเวอร์ไซด์ แอปพลิเคชัน (Server Side Application) พื้นฐานที่ทางเว็บเซิร์ฟเวอร์เหล่านั้นจะต้องสนับสนุนก็คือ ซีจีไอ ซึ่งจุดนี้เองที่เป็นจุดเด่นทำให้ ซีจีไอ เป็นที่นิยมใช้กันอย่างกว้างขวาง แต่ต่อมาในภายหลังเทคโนโลยีทางอินเทอร์เน็ตที่ใช้แนวคิดของเซิร์ฟเล็ต (Servlet) ก็ได้กำเนิดขึ้นและมีแนวโน้มที่จะเป็นที่นิยมใช้กันแพร่หลาย เหตุผลสืบเนื่องมาจาก เซิร์ฟเล็ต มีคุณสมบัติและการทำงานที่มีประสิทธิภาพมากกว่า ซีจีไอ นั่นเอง

เซิร์ฟเล็ต เป็นเซิร์ฟเวอร์ ไซด์ แอปพลิเคชัน (Server Side Application) แบบหนึ่งจึงสามารถนำมาใช้แทน ซีจีไอสคริปต์ (ซีจีไอ สคริปต์) ได้ ภาษาที่นำมาใช้เขียน เซิร์ฟเล็ต คือ ภาษาจาวา(Java) โดยตัวภาษาจาวานั้นไม่เพียงแต่เป็นภาษาทางคอมพิวเตอร์เท่านั้นแต่ยังเป็นภาษาที่สามารถนำเอามาใช้จัดการกับระบบไคลเอนต์/เซิร์ฟเวอร์(Client/server) ได้ด้วย จึงมีประโยชน์อย่างยิ่งในการนำมาเขียนเซิร์ฟเล็ต โดยตัวของเซิร์ฟเล็ตเองมีโครงสร้างการให้บริการ(Service)เป็นแบบการร้องขอบริการและการตอบสนองการร้องขอบริการนั้น (Request – Response Paradigm) มุมมองของการติดต่อกันหรือการแก้ไขจะใช้เทคนิคการให้กำเนิดเว็บไดนามิก เว็บเพจ (Generate Dynamic Web Page) กล่าวคือ เซิร์ฟเล็ต เป็นเทคนิคที่นำมาใช้ในการสร้าง ไดนามิก เว็บเพจ (Dynamic web page) ที่มีประสิทธิภาพนั่นเอง นอกจากนี้เนื่องจากเซิร์ฟเล็ตเป็นโปรแกรมที่รัน(run)ภายในฝั่งเซิร์ฟเวอร์ ดังนั้น เซิร์ฟเล็ต จึงไม่ต้องการการเชื่อมต่อกับผู้ใช้(Graphic User Interface) โดยตรง ส่งผลให้ เซิร์ฟเล็ต ไม่มีในส่วนของความคิดต่อกับผู้ใช้ (Graphic User Interface) และ เซิร์ฟเล็ต สามารถเก็บฝัง(Embedded)ไว้ในหลายๆเซิร์ฟเวอร์ที่แตกต่างกันได้เพราะว่ามีในส่วนของ เซิร์ฟเล็ต เอพีไอ (Servlet API) ซึ่งเป็นสิ่งที่ใช้ช่วยในการเขียนเซิร์ฟเล็ต และแสดงให้เห็นว่า เซิร์ฟเล็ต ติดต่อกับเซิร์ฟเวอร์เมื่อไหร่และติดต่อกับอย่างไร สิ่งที่สำคัญคือ เซิร์ฟเล็ต เอพีไอ (Servlet API) จะต้องมียุทธศาสตร์ของฟังก์ชันการเรียกใช้บริการ(set of function calls) ซึ่งแสดงให้เห็นถึงการส่งข้อมูลข่าวสารให้ เซิร์ฟเวอร์ และการรับข้อมูลข่าวสารจากเซิร์ฟเวอร์ด้วย ทั้งนี้รวมทั้งจะต้องสามารถทำการส่งใหม่(Redirect)และแจ้งข้อมูลที่แสดงความผิดพลาด(Error message) ได้ด้วย

2.5.2 ข้อดีของเซิร์ฟเล็ต (Servlet)

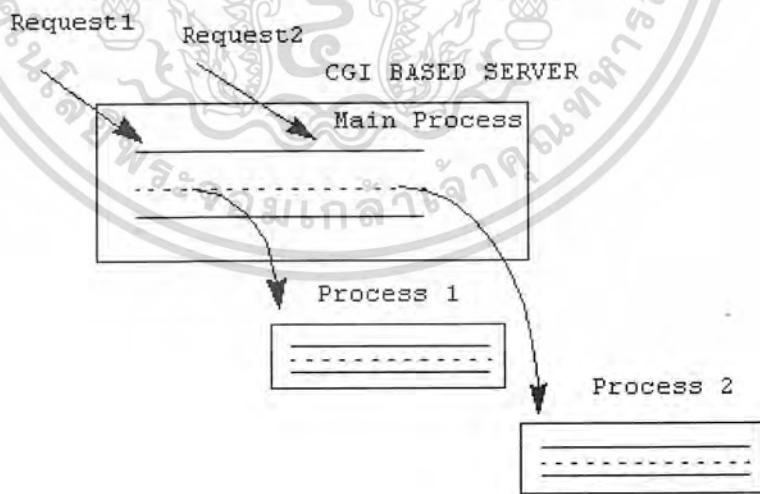
การที่สามารถนำ เซิร์ฟเล็ต มาใช้แทน ซีจีไอ(CGI-Common Gateway Interface) สคริปต์ ได้นั้น เหตุผลเพราะว่า เซิร์ฟเล็ต มีข้อดีที่เหนือกว่า ซีจีไอ หลายประการ โดยแสดงได้ดังต่อไปนี้

1. เซิร์ฟเล็ต มีความคงทน(Persistent)มากกว่า ซีจีไอ : เซิร์ฟเล็ตนั้นเมื่อต้องการใช้งานจะมีการโหลดเพียงครั้งเดียวเมื่อถูกเรียก โดยเว็บเซิร์ฟเวอร์(Web server)จะเป็นผู้ที่โหลดเซิร์ฟเล็ตขึ้นมา และจะไม่มีการโหลดอีกจนกว่า เซิร์ฟเล็ตจะถูกเปลี่ยนแปลงหรือแก้ไข และเมื่อมีการแก้ไขเซิร์ฟเล็ต แล้วสามารถรีโหลด(Reload) ได้ใหม่เลยโดยที่ไม่ต้องรีสตาร์ทเซิร์ฟเวอร์ (Restart Server) ใหม่ กล่าวคือ เซิร์ฟเล็ตสามารถรักษาการบริการ(Service)นั้นไปจนตลอดการร้องขอทั้งหมดได้(เช่น การติดต่อกับฐานข้อมูล(Database connection)จะมีการติดต่อเพียงครั้งเดียวและสามารถเรียกใช้ข้อมูลในฐานข้อมูลนั้นได้เมื่อมีการเรียกใช้เซิร์ฟเล็ตในเวลาถัดไปโดยที่ไม่ต้องมีการโหลดมาใหม่) ซึ่งตรงกันข้ามกับ ซีจีไอ สคริปต์ (CGI script) โดยสิ้นเชิง เพราะ ซีจีไอ สคริปต์ นั้นเมื่อมีการโหลดมาแล้วจะคงอยู่ในช่วงระยะเวลาหนึ่งเท่านั้น กล่าวคือในแต่ละครั้งที่มีการร้องขอ การร้องขอนั้นจะถูกส่งไปยัง ซีจีไอ สคริปต์ โดยเว็บเซิร์ฟเวอร์จะเป็นผู้โหลดและกระทำ (Execute) ตัวซีจีไอ สคริปต์ นั้นใหม่ทุกครั้ง และเมื่อซีจีไอ สคริปต์ นั้นทำงานเสร็จสมบูรณ์และส่งผลลัพธ์กลับไปยังผู้ร้องขอ(Client)เรียบร้อยแล้วก็จะถูกลบออกไปจากหน่วยความจำ(Memory)ทันที โดยทุกโปรแกรมจะต้องมีการติดตั้งค่าเริ่มต้นใหม่ทุกครั้ง(เช่น การติดต่อกับฐานข้อมูลจะมีการติดต่อและติดตั้งค่าเริ่มต้นใหม่ทุกครั้งที่ซีจีไอ สคริปต์ ถูกเรียกใช้)
2. เซิร์ฟเล็ต จะมีการทำงานที่เร็วกว่า ซีจีไอ : เนื่องจาก เซิร์ฟเล็ตมีการ โหลดเพียงครั้งเดียวเท่านั้นและจะอยู่ในหน่วยความจำโดยไม่ได้โดนลบออกไปจากหน่วยความจำเมื่อเซิร์ฟเล็ตทำงานเสร็จ เป็นผลให้เซิร์ฟเล็ตทำงานได้เร็วมาก
3. เซิร์ฟเล็ต เป็นรูปแบบอิสระ (Platform Independent) : เนื่องจากเซิร์ฟเล็ตใช้ภาษาจาวาในการเขียนจึงได้รับการสืบทอดข้อดีต่างๆของภาษาจาวามาด้วย นั่นคือได้รับคุณสมบัติรูปแบบอิสระ (Platform Independent) มาด้วยซึ่งเป็นคุณสมบัติที่ทำให้เซิร์ฟเล็ตสามารถเคลื่อนย้ายจากเซิร์ฟเวอร์หนึ่งไปยังเซิร์ฟเวอร์อื่นๆได้โดยที่ไม่ต้องกังวลเกี่ยวกับการทำงานภายในของเซิร์ฟเวอร์นั้นๆ ยกตัวอย่างเช่น ไม่ต้องกังวลเกี่ยวกับระบบปฏิบัติการ (Operating System(OS)), หรือชนิดของเซิร์ฟเวอร์ เลยกกล่าวได้ว่าเซิร์ฟเล็ตเป็น “การเขียนเพียงครั้งเดียว แต่สามารถรันที่ไหนก็ได้” (Write Once, Run Anywhere) คือ เขียนแล้วผ่านการคอมไพล์ (Compile) เพียงครั้งเดียวจากนั้นสามารถนำไปรันบนเซิร์ฟเวอร์ ใดๆ ก็ได้ต่างจากซีจีไอ สคริปต์ (CGI script) ที่เขียนโดยภาษาซี หรือภาษาเพิร์ล (Perl) ซึ่งโดยมากจะมีรูปแบบเซิร์ฟเวอร์ (Server Platform) ที่เฉพาะเจาะจง ทำให้ขอบเขตการนำไปใช้งานไม่สะดวกเหมือนเซิร์ฟเล็ต
4. เซิร์ฟเล็ต สามารถนำกลับมาใช้ใหม่ได้(Reusable) : ตัวภาษาที่ใช้เขียนซึ่งก็คือจาวานั้นเอง จาวาเป็นภาษาที่ใช้คอนเซ็ปของหลักการเชิงวัตถุ (Object Oriented) ในการเขียน โดยหลักการเชิงวัตถุนั้นสามารถลดความซ้ำซ้อนของโครงสร้างโปรแกรมรวมไปถึงการอำนวยความสะดวกในการนำกลับมาใช้ใหม่ (Reuse)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

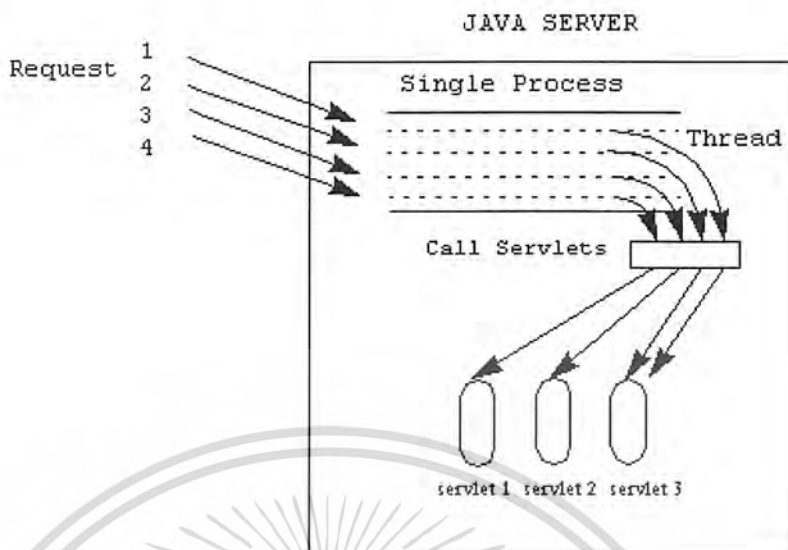
ส่วนของโปรแกรมที่เขียนไว้กลับมาใช้ใหม่ได้อีกด้วย ในขณะที่ ซีจีไอ นั้นเป็นการยากที่จะนำเอาโค้ด (Code) กลับมาใช้งานใหม่ได้

5. เซิร์ฟเล็ต สามารถขยายเพิ่มเติมได้(Extensible) : เพราะภาษาจาวาเป็นภาษาที่มีความแข็งแกร่ง (Robust) และสามารถขยายเพิ่มเติมได้ ตัวเซิร์ฟเล็ตซึ่งใช้ภาษาจาวาในการเขียนจึงมีความแข็งแกร่งและสามารถขยายเพิ่มเติมฟังก์ชันการทำงานต่างๆได้ตามความเหมาะสม
6. เซิร์ฟเล็ต มีความปลอดภัยสูง : เนื่องจากการขอเข้าใช้งาน เซิร์ฟเล็ต จากภายนอกนั้นสามารถทำได้โดยการติดต่อผ่าน เว็บเซิร์ฟเวอร์เท่านั้นเว็บเบราว์เซอร์ไม่ได้ติดต่อกับเซิร์ฟเล็ตโดยตรง จึงทำให้มีความปลอดภัยในการใช้งานสูงโดยเฉพาะอย่างยิ่งถ้าเซิร์ฟเวอร์มีระบบรักษาความปลอดภัยที่มีประสิทธิภาพแล้วละก็ จะทำให้เซิร์ฟเล็ตนั้นมีความปลอดภัยที่สูงตามไปด้วย
7. เซิร์ฟเล็ต ง่ายกว่า ซีจีไอ และเหมาะกับการสนับสนุน(Support)การสร้างเว็บ แอปพลิเคชัน (Web Application มากกว่า) : เนื่องจากปัญหาของซีจีไอ คือจะมีการสร้างกระบวนการ(process)ที่ซ้ำซ้อนมากจนเกินไปแต่เซิร์ฟเล็ตสามารถแก้ปัญหานี้ได้เพราะมีการทำงานในลักษณะที่เรียกว่าเธรด(Thread)เข้ามาช่วย
8. เซิร์ฟเล็ต สามารถให้บริการกับผู้ใช้(Client)ที่หลากหลายได้ในเวลาเดียวกัน : เซิร์ฟเล็ต มีการทำงานแบบขนาน(Parallel) โดยจะไม่สร้างกระบวนการใหม่(Process)ทุกครั้งที่มีการร้องขอ(Request)แต่จะรันทำงานขนานกันไปในเธรด (Thread)เดียวกัน เพราะเซิร์ฟเล็ตจะทำการสร้าง 1 เธรด (Thread) ต่อหนึ่งการร้องขอ (Request) ที่มาจากไคลเอนต์ (Client) ต่างจาก ซีจีไอ ซึ่งจะทำการสร้าง 1 โพรเซส (Process) ต่อหนึ่งการร้องขอ (Request) ทำให้เปลืองทรัพยากรมากกว่าและกระบวนการในการรันก็จะช้ากว่าด้วย ด้วยเหตุนี้จึงทำให้ เซิร์ฟเล็ตสามารถให้บริการกับผู้ใช้หลายๆคน(Multi-user)พร้อมกันได้



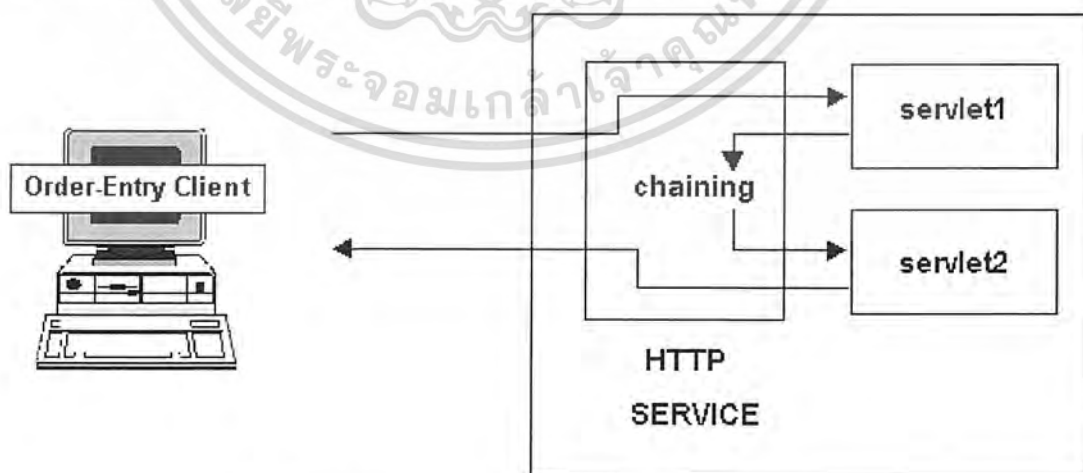
รูปที่ 2.14 แสดงการทำงานของ ซีจีไอ (CGI) ที่สร้าง 1 โพรเซส (Process) ต่อ 1 การร้องขอ (Request)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงการทำงานของเซิร์ฟเล็ต (Servlet) ที่สร้าง 1 เธรด (Thread) ต่อ 1 การร้องขอ (Request)

- เซิร์ฟเล็ต สามารถปฏิบัติงานเฉพาะ(Specific task)บางอย่างได้ โดยแต่ละ เซิร์ฟเล็ต สามารถสื่อสารหรือคุยกันได้โดยจะเรียกการสื่อสารระหว่าง เซิร์ฟเล็ต นี้ว่า “เซิร์ฟเล็ต เชนนิ่ง” (Servlet Chaining) และลักษณะการทำงานของ เซิร์ฟเล็ต เชนนิ่ง (Servlet Chaining) ก็คือ การนำเอาผลที่ได้ (Output) จากเซิร์ฟเล็ตตัวหนึ่งไปเป็นอินพุต (Input) ให้แก่ เซิร์ฟเล็ตตัวถัดไปเพื่อนำไปประมวลผลต่อไปได้ ซึ่งทำให้เซิร์ฟเล็ต มีกำลังและความแข็งแรงมากเนื่องจากตัวเซิร์ฟเล็ตเองไม่จำเป็นต้องรู้ถึงการทำงานทั้งหมดเพียงแต่รู้เฉพาะในส่วนที่ตนเองต้องรับผิดชอบเท่านั้น



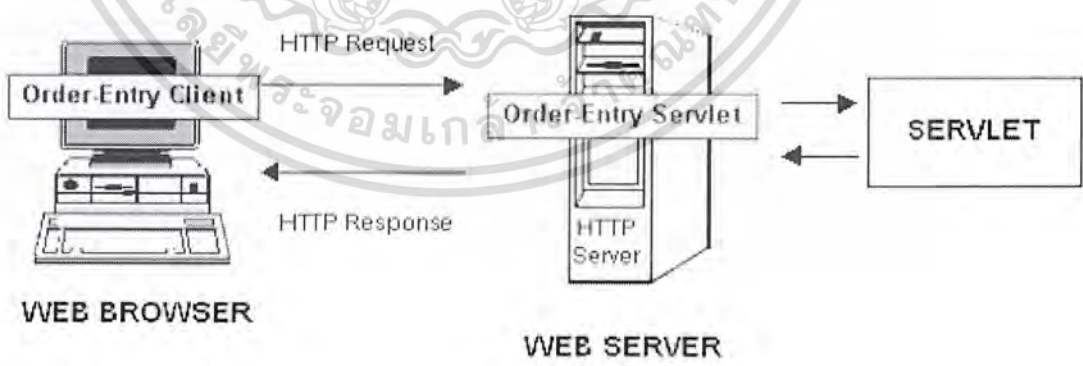
รูปที่ 2.16 แสดง เซิร์ฟเล็ต เชนนิ่ง (Servlet Chaining)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 ขั้นตอนการทำงานพื้นฐานของเซิร์ฟเล็ต (Basic Servlet Flow)

การติดต่อกันระหว่าง เว็บเบราว์เซอร์ (Web Browser) และ เว็บเซิร์ฟเวอร์ (Web Server) จะมีกระบวนการทำงานคือ เมื่อเซิร์ฟเล็ต (Servlet) ได้รับการร้องขอจากไคลเอนต์ (Client) ก็จะปฏิบัติตามการร้องขอนั้น และส่งผลลัพธ์กลับไปยัง ไคลเอนต์ที่ทำการร้องขอมาโดยสามารถแสดงขั้นตอนการทำงานพื้นฐานโดยใช้เซิร์ฟเล็ต ได้ดังนี้

1. ไคลเอนต์ ทำการร้องขอ(Request)ผ่านทาง เว็บเบราว์เซอร์
2. เว็บเบราว์เซอร์ จะทำการร้องขอผ่านทาง เอชทีทีพี โพรโตคอล (HTTP Protocol) ในระบบเครือข่าย(Network System)ไปยัง เว็บเซิร์ฟเวอร์
3. เว็บเซิร์ฟเวอร์ได้รับการร้องขอและส่งไปยังเซิร์ฟเล็ตโดยเว็บเซิร์ฟเวอร์จะตรวจสอบว่าเซิร์ฟเล็ตนั้นเคยถูกโหลดมาแล้วหรือยังถ้ายังก็จะทำการโหลดไปยัง จาวา เวอร์ชวล แมชีน (Java Virtual Machine) และ เอ็กซีคิวต์ เซิร์ฟเล็ต (Execute Servlet)
4. เซิร์ฟเล็ต รับการร้องขอ เอชทีทีพี (HTTP request)ที่เข้ามา และปฏิบัติตามกระบวนการทำการประมวลผลเพื่อตอบสนองการร้องขอนั้น
5. เซิร์ฟเล็ต ส่งผลลัพธ์ตอบสนอง(Response)กลับ ไปยัง เว็บเซิร์ฟเวอร์
6. เว็บเซิร์ฟเวอร์ ส่งผลลัพธ์การตอบสนองกลับไปยังเว็บเบราว์เซอร์ ผ่านระบบเครือข่าย(Network System)
7. เว็บเบราว์เซอร์ แสดงผลลัพธ์การตอบสนองแก่ ไคลเอนต์



รูปที่ 2.17 แสดงการติดต่อกันระหว่างเว็บเบราว์เซอร์ (Web browser) และเว็บเซิร์ฟเวอร์ (Web server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 ความสัมพันธ์ระหว่างเซิร์ฟเล็ต (Servlet) กับ แอปเพล็ต (Applet)

ทั้ง เซิร์ฟเล็ต และ แอปเพล็ต ต่างก็เป็นสิ่งสำคัญในระบบไคลเอนต์/เซิร์ฟเวอร์ (Client/Server) โดยเซิร์ฟเล็ต โดยมากนั้นก็จะมีลักษณะเหมือนหนึ่งเป็น แอปเพล็ต ที่อยู่ฝั่งของ เซิร์ฟเวอร์(เซิร์ฟเวอร์-Side แอปเพล็ต)นั่นเอง แต่ความแตกต่างระหว่าง เซิร์ฟเล็ต และ แอปเพล็ต นั้นก็คือ เซิร์ฟเล็ต จะรันภายใน เว็บเซิร์ฟเวอร์ ส่วน แอปเพล็ต นั้นจะรันอยู่ภายในบราวเซอร์ในการรันจาวาเซิร์ฟเล็ต สิ่งที่สำคัญและจำเป็นต้องมีคือจาวาเวอร์ชวล แมชีน (Java Virtual Machine(JVM)) ที่รันอยู่บนเซิร์ฟเวอร์ก็คล้ายๆกับการรันแอปเพล็ตในบราวเซอร์ โดยบราวเซอร์ที่สนับสนุน แอปเพล็ต จะต้องสามารถรันจี้วีเอ็ม(JVM)ได้ เช่นเดียวกันสำหรับเซิร์ฟเวอร์ ที่จะรันเซิร์ฟเล็ตก็ต้องสามารถรันจี้วีเอ็ม (JVM) ได้ด้วย

2.5.5 วงจรชีวิตของเซิร์ฟเล็ต (Java Servlet Life Cycle)

การทำงานในส่วนของ เซิร์ฟเล็ต นั้นจะมีวงจรการทำงาน หรือที่เรียกว่าวงจรชีวิตของเซิร์ฟเล็ต (Servlet life cycle) โดยจะมีรอบ (Cycle) ในการทำงานทั้งหมด 3 ขั้นตอนคือ เมธอดอินิต (init()) , เมธอดเซอร์วิซ (service()) และ เมธอดเดสทรอย (destroy())

1. เมธอดอินิต (init())

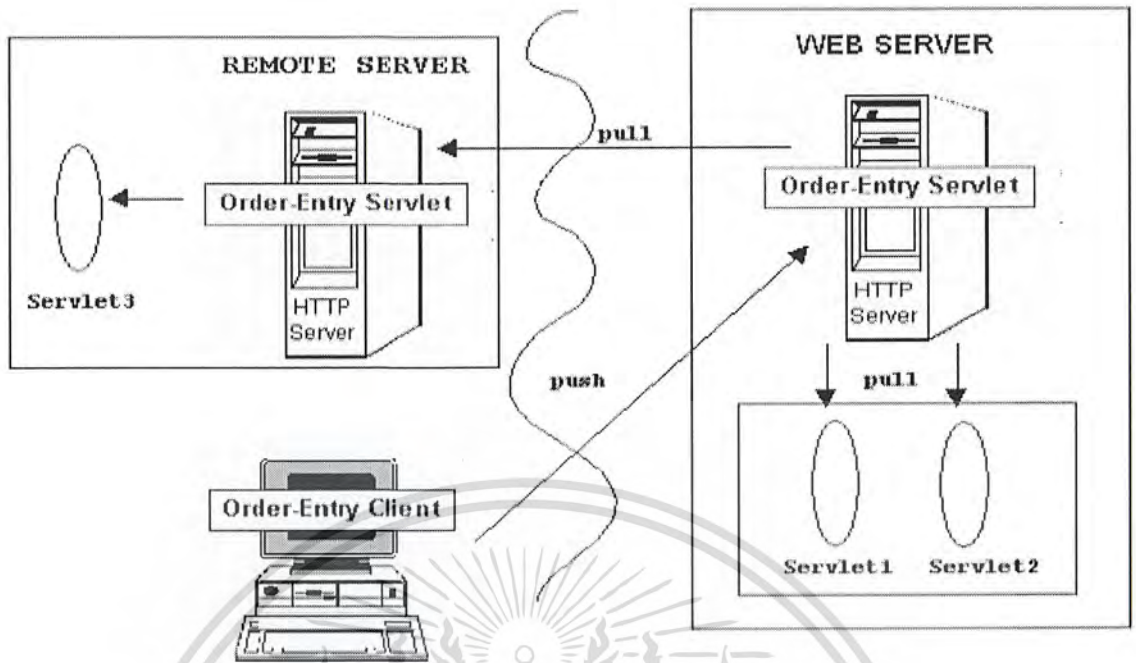
เมธอด นี้จะถูกใช้งานเมื่อมีการเรียกใช้งานเซิร์ฟเล็ต โดยจะเป็นส่วนที่สร้างคอนสตรัคเตอร์ (Constructure) สำหรับการเริ่มต้นการทำงานของโปรแกรม โดยการติดตั้งค่าเริ่มต้นต่างๆนี้จะรวมถึงการติดตั้งเพื่อเชื่อมต่อไปยังระบบเครือข่ายอื่นๆด้วย

2. เมธอด เซอร์วิซ (service())

จะถูกใช้งานหลังจากที่เมธอดอินิต (init()) ถูกเรียก โดยเมธอด เซอร์วิซ (service()) จะทำการแฮนเดิลการร้องขอ (Handles request) จากฝั่งไคลเอนต์ และทำงานตามเมธอดที่ได้แสดง(Define)ไว้ โดยในแต่ละการร้องขอจะให้หนึ่งการบริการ(service()) และถ้าหากมีหลายการร้องขอก็สามารถทำงานไปพร้อมๆกันได้(Concurrent)

3. เมธอดเดสทรอย (destroy())

เมื่อเซิร์ฟเล็ตประมวลผลการร้องขอจนเสร็จสิ้นสมบูรณ์แล้ว เว็บเซิร์ฟเวอร์ก็จะยกเลิก(Shut down)การใช้เซิร์ฟเล็ตนั้นโดยเรียกเมธอดเดสทรอย (destroy()) ขึ้นมาทำงาน



รูปที่ 2.18 แสดงการใช้งานเซิร์ฟเล็ต (Servlet) โดยสามารถเรียกใช้งานเซิร์ฟเล็ตที่อยู่บนเครื่องข่ายได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

โครงสร้างและการทำงานของระบบ

3.1 การเลือกโปรแกรมประยุกต์และสิ่งแวดล้อมที่ใช้ในสร้างโปรแกรมทำงาน

งานวิจัยในโครงการนี้มีวัตถุประสงค์ที่จะสร้างระบบซึ่งสามารถให้ผู้ใช้แต่ละคนซึ่งอยู่ต่างสถานที่กันสามารถมาทำงานร่วมกันได้อย่างมีประสิทธิภาพ มีปัจจัยที่ต้องคำนึงถึงอยู่หลายประการ ได้แก่ โพรโตคอลที่ใช้ในการสื่อสารกันระหว่างคอมพิวเตอร์ โปรแกรมประยุกต์ที่ใช้สร้างโปรแกรมทำงาน แหล่งสตอเรจข้อมูลกลาง และเว็บเซิร์ฟเวอร์ที่ใช้ในการเก็บโปรแกรมทำงาน

การกำหนดโพรโตคอลนั้น ทางผู้จัดทำได้สังเกตเห็นว่าเครือข่ายอินเทอร์เน็ตกำลังเป็นเครือข่ายที่ได้รับความนิยมอย่างกว้างขวางสำหรับผู้ใช้อินเทอร์เน็ตส่วนบุคคลทั่วไป ดังนั้นจึงอาศัยโพรโตคอลเซชที่ทีพี (HTTP) ซึ่งเป็นข้อกำหนดในการจัดรูปแบบข้อมูล การส่งและรับข้อมูล การรับและตอบสนองของคำสั่งต่าง ๆ ระหว่างเว็บเซิร์ฟเวอร์และบราวเซอร์ในการสร้างโปรแกรมทำงาน

ลักษณะของโปรแกรมจะเป็นโปรแกรมที่ทำงานในฝั่งเซิร์ฟเวอร์ที่จะคอยตอบสนองการร้องขอจากเว็บเพจของผู้ใช้แต่ละคน โปรแกรมประยุกต์ที่เลือกใช้คือ โปรแกรมเซิร์ฟเล็ต ซึ่งเป็นโปรแกรมซีจีไอที่ทำงานคล้ายกับการทำงานของแอปพลิเคชันแต่สามารถทำงานในสิ่งแวดล้อมของเว็บเซิร์ฟเวอร์ได้ โดยเขียนโปรแกรมด้วยพื้นฐานภาษาจาวาและมีการเพิ่มแพ็คเกจที่จำเป็นในการทำงานในฝั่งเซิร์ฟเวอร์เพิ่มเติม โปรแกรมทำงานทั้งหมดจะถูกเก็บไว้ในเว็บเซิร์ฟเวอร์ ซึ่งในงานวิจัยโครงการนี้ได้เลือกใช้จาวาเว็บเซิร์ฟเวอร์เป็นเว็บเซิร์ฟเวอร์ เนื่องจากความสะดวกในการติดตั้ง การทำการคอนฟิกูเรชัน และข้อมูลส่วนกลางจะเก็บไว้ในฐานข้อมูลแทนการเก็บลงไฟล์เพราะสามารถจัดเก็บและเข้าถึงข้อมูลได้ง่ายกว่า และสามารถรองรับขนาดข้อมูลจำนวนมากได้ เซิร์ฟเล็ตจะติดต่อกับฐานข้อมูลได้โดยตรงผ่านการเชื่อมต่อกันระหว่างไดร์เวอร์ของเซิร์ฟเล็ตและไดร์เวอร์ของระบบจัดการฐานข้อมูล (DBMS)

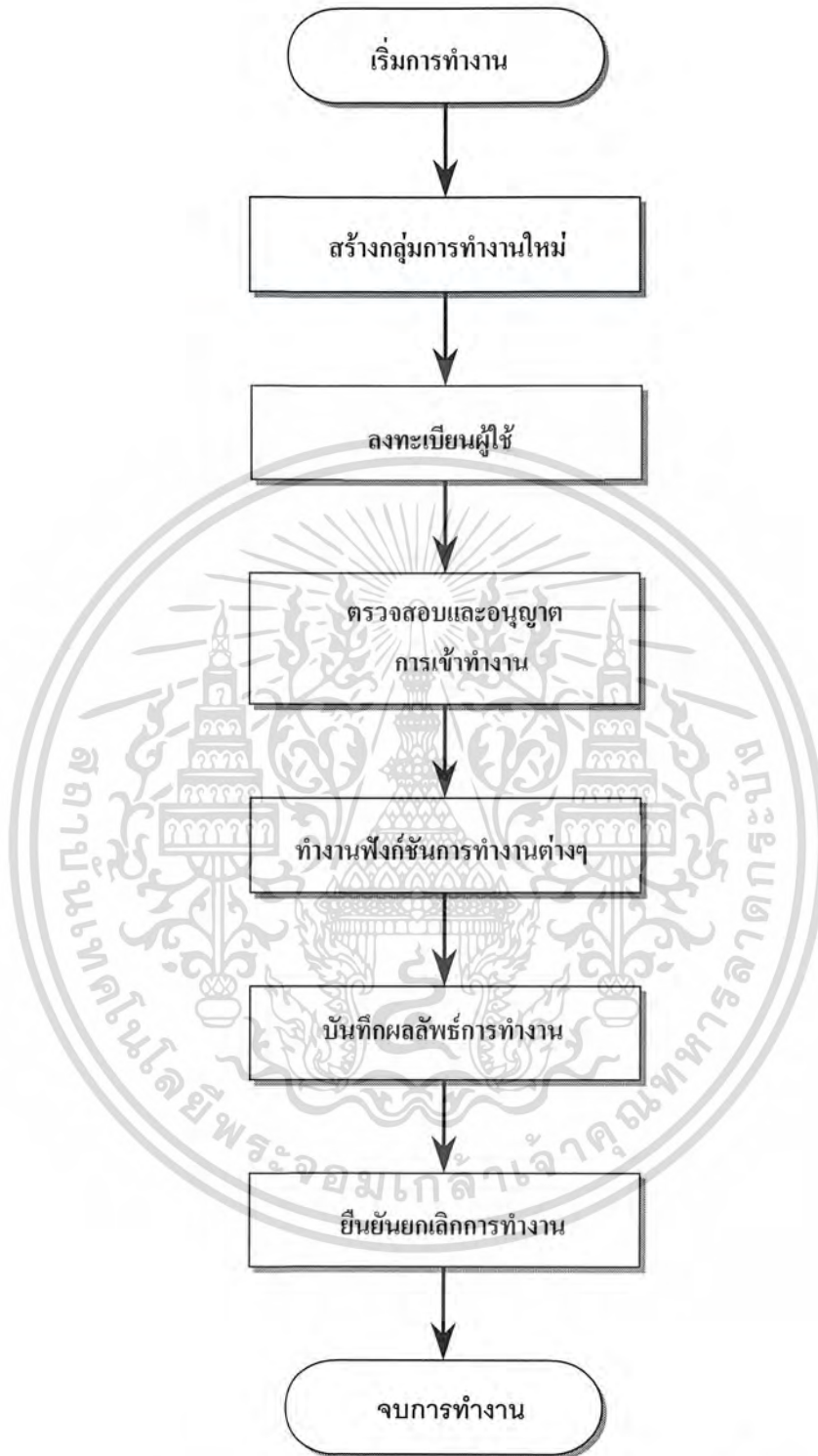
ส่วนการร้องขอการทำงานฝั่งไคลเอนต์ต่างๆของผู้ใช้นั้น กระทำด้วยการการส่งการร้องขอจากบราวเซอร์ โดยบราวเซอร์บางตัวอาจไม่รองรับการทำงานบางอย่างของเซิร์ฟเล็ต เช่น เทคนิค server push ไม่สามารถใช้กับบราวเซอร์ของ Internet Explorer ได้

3.2 ขั้นตอนการสร้างโปรแกรมการทำงานร่วมกัน

1. ทำการสร้างเว็บเพจด้วยภาษาฮทเอ็มแอล (HTML) สำหรับผู้ใช้ ที่สามารถส่งการร้องขอและทำการติดต่อกับเซิร์ฟเล็ตได้ รับข้อมูลจากผู้ใช้ผ่านทางแท็กฟอร์มและส่งค่าพารามิเตอร์ไปกับการร้องขอการทำงานแต่ละครั้ง
2. สร้างโปรแกรมที่ทำงานฝั่งเซิร์ฟเวอร์ในการจัดการการทำงานร่วมกันระหว่างผู้ใช้แต่ละคน ให้มีความราบรื่นและถูกต้องตามลำดับก่อนหลัง อีกทั้งให้มีความสามารถรองรับการทำงานหลายๆ กลุ่มการทำงานพร้อมกันได้
3. สร้างสเกิมาของตารางที่ใช้ในการเก็บข้อมูลส่วนกลางของผู้ใช้แต่ละกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โครงสร้างของโปรแกรมการทำงานร่วมกัน



รูปที่ 3.1 แสดงโครงสร้างการทำงานของโปรแกรมการทำงานร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 อินเทอร์เน็ตต่างๆ ที่ติดต่อกับผู้ใช้

การทำงานร่วมกันผ่านเครือข่ายเวิร์ลไวด์เว็บ ผู้ใช้แต่ละคนจะส่งข้อมูลผ่านแท็กฟอร์มของภาษา เซชทีเอ็มแอล(HTML) ซึ่งโปรแกรมทางฝั่งเซิร์ฟเวอร์จะได้รับข้อมูลโดยการอ่านค่าจากพารามิเตอร์ที่ส่ง มาให้ ดังนั้นการออกแบบอินเทอร์เน็ตที่ติดต่อกับผู้ใช้นั้นนอกจากจะความสะดวก และง่ายต่อการเข้าใจ นั้นยังต้องให้ครอบคลุมถึงข้อมูลทั้งหมดที่จำเป็นในการประมวลผลการทำงาน และกำหนดกฎในการรับ ข้อมูลอินพุท

อินเทอร์เน็ตที่ใช้ในการติดต่อกับผู้ใช้ของโปรแกรมการทำงานร่วมกัน มีดังต่อไปนี้

1. อินเทอร์เน็ตที่ติดต่อกับผู้ใช้ในการสร้างกลุ่มการทำงานใหม่

Workgroup Registration

Name of your WorkGroup : *

WorkGroup Password : *

Re-Enter Password : *

About Work (Comment) :
 (optional)

Creator Name : (optional)

รูปที่ 3.2 แสดงอินเทอร์เน็ตที่ติดต่อกับผู้ใช้ในการสร้างกลุ่มการทำงานใหม่

ประกอบด้วยฟิลด์ข้อมูล ชื่อของกลุ่มการทำงาน , รหัสผ่านของกลุ่มการทำงาน , รหัสผ่านยืนยัน ของกลุ่มการทำงาน , รายละเอียดอธิบายลักษณะของกลุ่มการทำงาน และ ชื่อผู้สร้างกลุ่มการทำงาน

ทำหน้าที่รับข้อมูลอินพุทที่เกี่ยวกับการสร้างกลุ่มการทำงานใหม่ขึ้นมา เมื่อกดปุ่ม“submit” ระบบจะนำพารามิเตอร์ต่างๆ ที่ได้รับมาทำการสร้างกลุ่มการทำงานใหม่ และบันทึกข้อมูลของกลุ่มการทำงานใหม่นั้นลงตารางกลุ่มการทำงาน

2. อินเทอร์เน็ตที่ติดต่อกับผู้ใช้ในการลงทะเบียนผู้ใช้

รูปที่ 3.3 แสดงอินเทอร์เน็ตที่ติดต่อกับผู้ใช้ในการลงทะเบียนผู้ใช้

ประกอบด้วย 2 ส่วน คือ กลุ่มของฟิลด์ข้อมูลที่เกี่ยวข้องกับข้อมูลส่วนตัวของผู้ใช้ได้แก่ ชื่อ-นามสกุล, ที่อยู่, ประเทศ, เพศ, ชื่ออีเมล, เบอร์โทรศัพท์, หมายเลขไอซีคิว และ วันเกิด กับ กลุ่มของฟิลด์ข้อมูลที่เกี่ยวข้องกับข้อมูลการเข้าใช้ ได้แก่ ชื่อยูสเซอร์ที่จะใช้ในการเข้าทำงาน, รหัสผ่านบังคับผู้ใช้ และ รหัสผ่านยืนยันบังคับผู้ใช้

ทำหน้าที่รับข้อมูลอินพุตที่เกี่ยวข้องกับการลงทะเบียนผู้ใช้ เมื่อกดปุ่ม“submit” ระบบจะบันทึกพารามิเตอร์ต่างๆ ที่ได้รับลงตารางผู้ใช้ที่ลงทะเบียนแล้ว

3. อินเทอร์เน็ตที่ติดต่อกับผู้ใช้ในการอนุญาตให้เข้าทำงาน

รูปที่ 3.4 แสดงอินเทอร์เน็ตที่ติดต่อกับผู้ใช้ในการอนุญาตให้เข้าทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วยฟิลด์ข้อมูล ชื่อของกลุ่มการทำงาน , รหัสผ่านของกลุ่มการทำงาน , ชื่อยูสเซอร์ และ รหัสผ่านบังคับผู้ใช้

ทำหน้าที่รับข้อมูลอินพุทเพื่อนำมาตรวจสอบว่ากลุ่มการทำงานและผู้ใช้มีอยู่จริงหรือไม่ และสามารถเข้าไปใช้ได้หรือไม่

4. อินเทอร์เฟซที่ติดต่อกับผู้ใช้ในการอัปโหลดไฟล์ภาพไปยังเซิร์ฟเวอร์, แสดงรายละเอียดของผู้ใช้ หรือ รายละเอียดของกลุ่มการทำงาน, แสดงลิสต์ของผู้ใช้ทั้งหมด และ/หรือ ลิสต์ของกลุ่มการทำงานทั้งหมด

Upload Image Files

Which file do you want to upload ?

Search for User or Workgroup Details

Search for : :

Display User or Workgroup Lists

Select Lists : User Lists Workgroup Lists

รูปที่ 3.5 แสดงอินเทอร์เฟซที่ติดต่อกับผู้ใช้ในการอัปโหลดไฟล์ภาพไปยังเซิร์ฟเวอร์, แสดงรายละเอียดของผู้ใช้ หรือ รายละเอียดของกลุ่มการทำงาน, แสดงลิสต์ของผู้ใช้ทั้งหมด และ/หรือ ลิสต์ของกลุ่มการทำงานทั้งหมด

ในการอัปโหลดไฟล์ภาพไปยังเซิร์ฟเวอร์ประกอบด้วยฟิลด์ข้อมูล ชื่อไฟล์ที่จะอัปโหลด โดยเมื่อกดปุ่ม “Browse” จะเกิดไฟล์ไดอะล็อกขึ้นให้เลือกไฟล์ที่ต้องการอัปโหลด และเมื่อกดปุ่ม “submit” ระบบจะสร้างไฟล์ภาพที่เหมือนกันกับไฟล์ภาพต้นทางขึ้นที่โพลเดอร์สำหรับเก็บภาพในเซิร์ฟเวอร์

ในการแสดงรายละเอียดของผู้ใช้ หรือ รายละเอียดของกลุ่มการทำงาน ประกอบด้วยฟิลด์ข้อมูล ประเภทของการค้นหา และ คีย์เวิร์ดการค้นหา เมื่อกดปุ่ม “submit” ระบบจะตรวจสอบประเภทการค้นหา และทำการค้นหาตามคีย์เวิร์ดที่ได้รับแล้วแสดงผลออกมาเป็นตารางแสดงรายละเอียดของผู้ใช้ หรือ กลุ่มการทำงาน

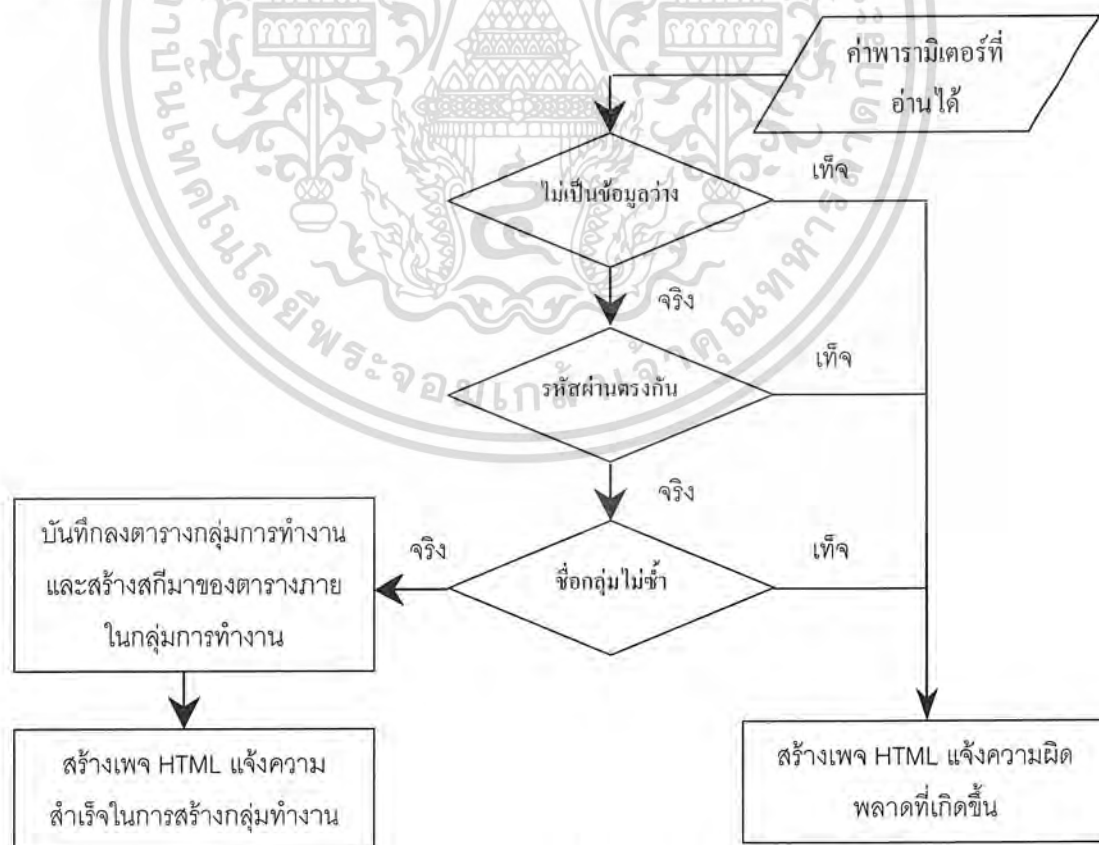
ในการแสดงลิสต์ของผู้ใช้ และ/หรือ ลิสต์ของกลุ่มการทำงาน เมื่อเลือกเช็คบ็อกซ์ที่ต้องการและกดปุ่ม “submit” ระบบจะแสดงรายชื่อของชื่อยูสเซอร์ทั้งหมด และ/หรือ แสดงรายชื่อของกลุ่มทำงานที่มีทั้งหมด

3.5 ขั้นตอนการสร้างกลุ่มการทำงานใหม่

ในการสร้างกลุ่มการทำงานขึ้นมาใหม่กลุ่มหนึ่งนั้น ผู้ใช้จะเลือก “Create New Workgroup” จากเฟรมเมนูด้านซ้ายมือ เซิร์ฟเล็ตชื่อ “CFMDC_createnewworkgroup” จะทำหน้าที่สร้างเพจเฮชทีเอ็มแอล (HTML) ขึ้นมาให้ผู้ใช้ป้อนข้อมูลเกี่ยวกับการสร้างกลุ่มการทำงานใหม่ แล้วเซิร์ฟเล็ตชื่อ “CFMDC_workgrouphandler” จะได้รับพารามิเตอร์มาจากการที่ผู้ใช้ป้อนข้อมูลอินพุตผ่านทางอินเทอร์เน็ตเพชด้วยเมธอด doPost() ข้อมูลที่ได้รับจะถูกนำมาพิจารณากับเงื่อนไขดังต่อไปนี้

- ฟیلด์ข้อมูลชื่อของกลุ่มการทำงานต้องไม่เป็นข้อมูลว่าง
- ฟیلด์ข้อมูลรหัสผ่านของกลุ่มการทำงานต้องไม่เป็นข้อมูลว่าง
- ข้อมูลภายในฟیلด์ข้อมูลรหัสผ่านของกลุ่มการทำงานและข้อมูลภายในฟیلด์ข้อมูลยืนยันรหัสผ่านของกลุ่มการทำงานต้องตรงกัน
- ฟیلด์ข้อมูลชื่อของกลุ่มการทำงานต้องไม่ซ้ำซ้อนกับชื่อกลุ่มการทำงานที่ได้สร้างขึ้นมาก่อนที่มีอยู่ในฐานข้อมูล

หากข้อมูลที่ผู้ใช้ป้อนผ่านทุกๆเงื่อนไขที่กำหนด เซิร์ฟเล็ตจะทำการบันทึกข้อมูลชื่อของกลุ่มการทำงาน, รหัสผ่านของกลุ่มการทำงาน, รายละเอียดอธิบายลักษณะของกลุ่มการทำงาน และ ชื่อผู้สร้างกลุ่มการทำงานลงตาราง “CFMDC_workgroup” หลังจากนั้นจะสร้างสกีมาของตารางที่จำเป็นในการทำงานของกลุ่มการทำงานนั้นขึ้นมา และสร้างเพจเฮชทีเอ็มแอลยืนยันการสร้างกลุ่มการทำงานใหม่ว่าเสร็จสมบูรณ์แล้ว แต่หากมีบางเงื่อนไขที่ไม่ตรงตามข้อกำหนดจะสร้างเพจเฮชทีเอ็มแอลที่แจ้งข้อผิดพลาดที่เกิดขึ้น



รูปที่ 3.6 แสดงไดอะแกรมของขั้นตอนการสร้างกลุ่มการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สกีมาของตารางที่จำเป็นในการทำงานร่วมกันมีดังนี้

1. Login Table ใช้เก็บข้อมูลของผู้ใช้ที่เข้ามาทำงานร่วมกันในขณะใดขณะหนึ่ง

ORDER_NO	USER_NAME	COLOR
----------	-----------	-------

2. Command Table ใช้เก็บข้อมูลคำสั่งต่างๆ ที่ทำให้เกิดการเปลี่ยนแปลงผลลัพธ์การทำงาน

ORDER_NO	COMMAND_TYPE	ATTRIBUTE	X_CO	Y_CO	CREATOR
----------	--------------	-----------	------	------	---------

3. Manage Table ใช้เก็บจำนวนคำสั่ง จุดเริ่มต้นและสิ้นสุดของผลลัพธ์ในแต่ละคำสั่ง

ORDER_NO	START_X	START_Y	END_X	END_Y
----------	---------	---------	-------	-------

4. Queue Table ใช้เก็บข้อมูลคำสั่งในการแก้ไข หรือ ลบ ผลลัพธ์ที่แสดงอยู่ที่หน้าจอการทำงานหลักเพื่อเข้าคิวรอการทำงาน

ORDER_NO	TYPE	ELEMENT	INFO	SECONDS	CREATOR
----------	------	---------	------	---------	---------

5. Conflict Table ใช้เก็บข้อมูลที่มีการเปลี่ยนแปลงแก้ไขอีลิเมนต์เดียวกันที่มีความซ้ำซ้อนกัน

ORDER_NO	TYPE	BEFORE	AFTER	CREATOR
----------	------	--------	-------	---------

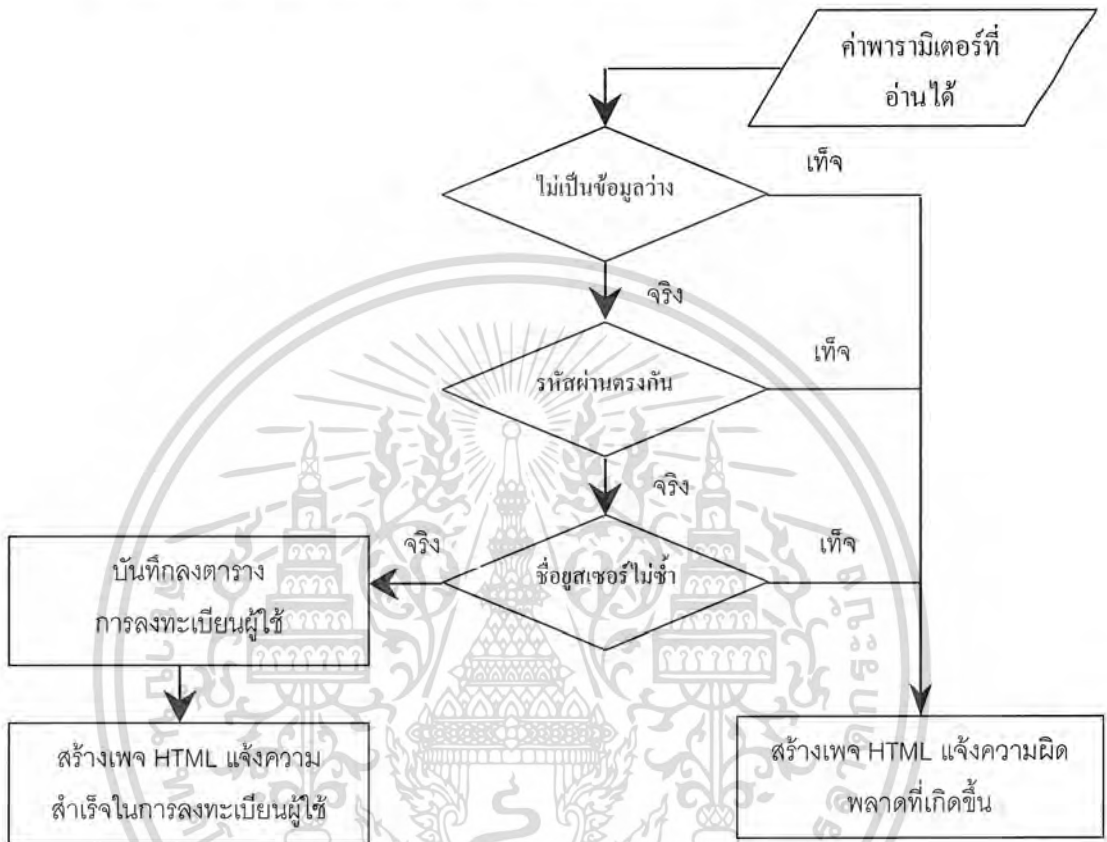
3.6 ขั้นตอนการลงทะเบียนผู้ใช้

การลงทะเบียนผู้ใช้จะทำเมื่อผู้ใช้ผู้นั้นได้เข้ามาใช้โปรแกรมการทำงานร่วมกันเป็นครั้งแรก ผู้ใช้จะเลือก“Create New User”จากเฟรมเมนูด้านซ้ายมือ เซิร์ฟเล็ตชื่อ “CFMDC_createnewuser”จะทำหน้าที่ในการสร้างเว็บเพจอินเทอร์เน็ตที่เชื่อมกับผู้ใช้ขึ้นมาเพื่อให้ผู้ใช้ป้อนข้อมูล จะมีเซิร์ฟเล็ตที่ชื่อ “CFMDC_userhandler”ทำหน้าที่อ่านค่าพารามิเตอร์ด้วยเมธอด doPost() มีการพิจารณาตรวจสอบเงื่อนไขดังนี้

- ฟิลด์ข้อมูลชื่อผู้สเซอร์ต้องไม่เป็นข้อมูลว่าง
- ฟิลด์ข้อมูลรหัสผ่านบ่งชี้ผู้ใช้ต้องไม่เป็นข้อมูลว่าง
- ข้อมูลภายในฟิลด์ข้อมูลรหัสผ่านบ่งชี้ผู้ใช้และข้อมูลภายในฟิลด์ข้อมูลยืนยันรหัสผ่านบ่งชี้ผู้ใช้ต้องตรงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟิลด์ข้อมูลชื่อผู้สเซอร์ต้องไม่ซ้ำซ้อนกับชื่อผู้สเซอร์ที่ได้ลงทะเบียนก่อนที่มีอยู่ในฐานข้อมูล หากข้อมูลที่ใช้ป้อนผ่านทุกๆเงื่อนไขที่กำหนด เซิร์ฟเวอร์จะทำการบันทึกรายละเอียดต่างๆที่ผู้ใช้ระบุมาลงตาราง “CFfDC_user” และสร้างเพจเฮลท์เอ็มแอล (HTML)แจ้งผลการลงทะเบียนผู้ใช้ว่าเสร็จสมบูรณ์แล้ว แต่หากมีบางเงื่อนไขบางประการที่ไม่ตรงตามข้อกำหนดจะสร้างเพจเฮลท์เอ็มแอล (HTML) ที่แจ้งข้อผิดพลาดที่เกิดขึ้นและไม่บันทึกข้อมูลลงตาราง



รูปที่ 3.7 แสดงไคอะแกรมของขั้นตอนการลงทะเบียนผู้ใช้

3.7 ขั้นตอนการตรวจสอบการอนุญาตเข้าทำงาน

เมื่อมีกลุ่มการทำงานและได้ทำการลงทะเบียนผู้ใช้เรียบร้อยแล้ว สามารถเข้าทำงานด้านการจัดการงานเอกสารได้โดยการทำขั้นตอนการตรวจสอบการอนุญาตเข้าทำงานซึ่งจะเป็นการตรวจสอบข้อมูลที่มีอยู่ในฐานข้อมูลว่าตรงกันกับข้อมูลที่ผู้ใช้ป้อนเข้ามาหรือไม่ เมื่อข้อมูลถูกต้องจะอนุญาตให้เข้าทำงานได้ต่อไป

วัตถุประสงค์ของการตรวจสอบและอนุญาต

1. เพื่อเป็นการสร้างความปลอดภัยในระดับหนึ่งให้กับงานเอกสารที่จัดทำ ป้องกันไม่ให้ผู้ใช้ที่ไม่ทราบรหัสผ่านของกลุ่มการทำงานเข้าไปทำงานโดยผลการได้
2. เพื่อเป็นการพิสูจน์ตนของผู้ใช้ด้วยการให้ผู้ใช้ระบุรหัสผ่านบ่งชี้ผู้ใช้ ซึ่งจะช่วยให้แน่ใจได้ว่าเป็นผู้ใช้คนนั้นๆ จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เพื่อที่จะสามารถระบุได้ว่าอิเลเมนต์ของข้อความที่ทำการเพิ่ม แก้ไข หรือ การลบอิเลเมนต์นั้น คัดสินใจมาจากผู้ใด
4. เพื่อจัดแยกผู้ใช้ให้เข้าทำงานตามกลุ่มการทำงานต่างๆ ได้อย่างถูกต้อง

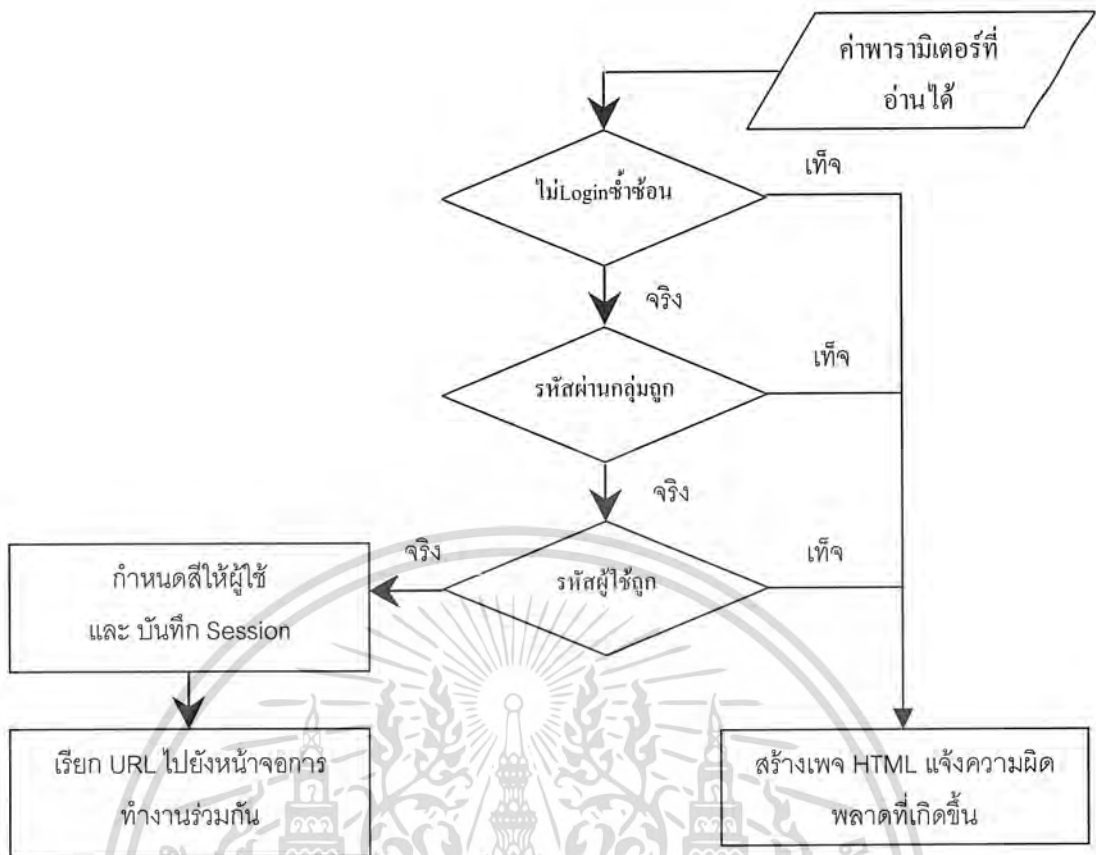
เซิร์ฟเล็ต“CFfDC_Loginhandler”จัดการการตรวจสอบการอนุญาตเข้าทำงานด้วยการอ่านค่าพารามิเตอร์ที่ได้รับมา นำไปตรวจสอบการมีอยู่ของข้อมูลในฐานข้อมูล ถ้าพบข้อมูลดังกล่าวในฐานข้อมูลจะอนุญาตให้เข้าทำงานได้ ทั้งนี้จะอนุญาตให้แต่ละผู้ใช้ทำการขอเข้าใช้งานได้ครั้งเดียวในขณะใดขณะหนึ่ง ผู้ใช้ไม่สามารถขอเข้าใช้ด้วยชื่อผู้ใช้เดียวกันได้หากมีการขอเข้าใช้งานด้วยชื่อดังกล่าวนี้ไปก่อนหน้านั้นและยังไม่ได้ออกจากการทำงาน

เงื่อนไขในการอนุญาตให้สามารถเข้าทำงานมีดังนี้

- พารามิเตอร์ชื่อของกลุ่มการทำงานถูกต้องตรงตามข้อมูลในฐานข้อมูลกลุ่มการทำงาน
- พารามิเตอร์ชื่อยูสเซอร์ถูกต้องตรงตามข้อมูลในฐานข้อมูลผู้ใช้
- พารามิเตอร์ชื่อยูสเซอร์ไม่ได้เข้าทำงานอยู่ก่อนแล้ว (สามารถสังเกตได้จากรายชื่อผู้ใช้ที่กำลังทำงานอยู่ใน Login Table)

ผู้ใช้แต่ละคนจะถูกกำหนดสิทธิ์ของข้อความที่ใช้ในการทำงานให้เป็นสิทธิ์ต่างๆตามลำดับก่อนหลังของการเข้าทำงานให้แตกต่างกันได้ทั้งหมด10สิทธิ์ ทำให้สามารถทราบได้ว่าการเปลี่ยนแปลงในการทำงานที่เกิดขึ้นแต่ละครั้งนั้นผู้ใช้คนใดเป็นผู้ทำ และนอกจากนี้เมื่อได้รับอนุญาตเข้าทำงานแล้วด้วยเทคนิค “session tracking” ของเซิร์ฟเล็ตทำให้ไม่ต้องทำการขอการเข้าทำงานอีกจนกว่าเซสชัน (session) นั้นจะถูกทำลายไปจากการปิดเว็บเบราว์เซอร์ หลักการก็จะมีการสร้างเซสชัน(session)ใหม่หากยังไม่มีเซสชัน (session)เกิดขึ้นและเมื่อมีการขออนุญาตเข้าทำงานจะใช้เมธอด putvalue()เขียนลงบนเซสชัน (session)

ขั้นตอนสุดท้ายของการตรวจสอบการขออนุญาตเข้าทำงาน เซิร์ฟเล็ตจะส่งการตอบสนองไปทำการเรียกURL (http://161.246.5.101:8080/servlet/CFfDC_servlet) ด้วยเมธอด sendRedirect() ผู้ใช้จะเข้าสู่หน้าจอการทำงานร่วมกันต่อไป



รูปที่ 3.8 แสดงไคอะแกรมของขั้นตอนการตรวจสอบการขออนุญาตเข้าทำงาน

3.8 ฟังก์ชันการทำงานต่างๆ ของโปรแกรมการทำงานร่วมกัน

3.8.1 การอัปโหลดไฟล์ภาพไปยังเซิร์ฟเวอร์

การอัปโหลดไฟล์ภาพเข้ามาไว้ยังโพลเดอร์ที่เก็บภาพภายในเว็บเซิร์ฟเวอร์เป็นการเตรียมความพร้อมในการที่จะแสดงผลรูปภาพในโปรแกรมการทำงานร่วมกัน รูปภาพที่จะนำมาแสดงผลในโปรแกรมการทำงานนั้นจะต้องถูกนำมาเก็บไว้ในเว็บเซิร์ฟเวอร์ก่อนเสมอ

แพ็คเกจที่ช่วยอำนวยความสะดวกในการจัดการเรื่องการขนย้ายไฟล์ประเภทมัลติมีเดีย ได้แก่ “com.oreilly.servlet.MultipartRequest” โดยเซิร์ฟเล็ต “CFIDC_upload” ได้ิมพอร์ตแพ็คเกจดังกล่าวมาใช้จัดการการอัปโหลดโดยไฟล์ภาพที่สามารถอัปโหลดมาได้ต้องมีขนาดไม่เกิน 5 เมกะไบต์ เมื่อผู้ใช้ต้องการอัปโหลดไฟล์ภาพจะสามารถทำได้โดยเลือกเมนู “Option” จากเฟรมเมนู จากนั้นทำการบราวส์ (browse) เลือกไฟล์ที่ต้องการและเมื่อกดปุ่ม “submit” ไฟล์ภาพจะถูกก๊อปปี้มาไว้ยังโพลเดอร์ “CFIDC_images” ที่ฝั่งเซิร์ฟเวอร์ และเซิร์ฟเล็ตจะแจ้งผลการอัปโหลดระบุชื่อไฟล์, ชนิดของไฟล์ และขนาดของไฟล์ที่อัปโหลดมาได้ให้กับผู้ใช้งาน

3.8.2 การแสดงรายละเอียดของผู้ใช้ หรือ รายละเอียดของกลุ่มการทำงาน

ในการแสดงรายละเอียดของผู้ใช้ หรือ รายละเอียดของกลุ่มการทำงาน เมื่อเลือก “Option” ทางเฟรมเมนูจะสังเกตเห็นฟอรม์ของการค้นหารายละเอียดผู้ใช้หรือกลุ่มการทำงานอยู่ตรงกลางหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บเพจ ซึ่งผู้ใช้จะเลือกว่าจะตรวจสอบรายละเอียดจากตารางผู้ใช้หรือตารางกลุ่มการทำงาน และกำหนด คีย์เวิร์ดในการค้นหา โดยคีย์เวิร์ดดังกล่าวจะเป็นชื่อของยูสเซอร์ถ้าเป็นการค้นหารายละเอียดผู้ใช้ และ คีย์เวิร์ดจะเป็นชื่อของกลุ่มการทำงานถ้าเป็นการค้นหารายละเอียดของกลุ่มการทำงาน

3.8.3 การแสดงลิสต์ของผู้ใช้ทั้งหมด และ/หรือ ลิสต์ของกลุ่มการทำงานทั้งหมด

การแสดงผลลิสต์ของผู้ใช้ทั้งหมดและ/หรือลิสต์ของกลุ่มการทำงานทั้งหมด ผู้ใช้จะเลือกOptionทาง เพรอมเมนูและเลือกเช็คบ็อกของฟอร์มที่อยู่ด้านล่างสุดของเว็บเพจ เมื่อกดปุ่ม “submit” ค่าพารามิเตอร์จะ ถูกส่ง ไปยังเซิร์ฟเว็ทที่ทำหน้าที่จัดการการแสดงผลลิสต์ที่ชื่อ “CFIDC_showlists”ซึ่งเซิร์ฟเว็ทนี้จะอ่านค่า สถานะของเช็คบ็อก ถ้าสถานะของเช็คบ็อกที่ระบุว่าจะให้แสดงผลลิสต์ของผู้ใช้ทั้งหมดถูกเช็ค เซิร์ฟเว็ทจะไป อ่านค่าจากฐานข้อมูลผู้ใช้งานมาแสดงผล ถ้าสถานะของเช็คบ็อกระบุว่าจะให้แสดงผลลิสต์ของกลุ่มการทำงานทั้งหมดถูกเช็ค เซิร์ฟเว็ทจะไปอ่านค่าจากฐานข้อมูลกลุ่มการทำงานมาแสดงผล

3.9 การทำงานของโปรแกรมการทำงานร่วมกัน

การทำงานของโปรแกรมการทำงานร่วมกันในงานเอกสารนั้นจะเป็นการสื่อสารสองทางระหว่าง เซิร์ฟเว็ทและแอปพลิเคชันภายใต้โพรโตคอลเฮททีที (HTTP) การทำงานจะเป็นการใช้แหล่งข้อมูลกลาง ร่วมกันภายในกลุ่มการทำงานเดียวกัน ทำให้เห็นผลลัพธ์การทำงานเหมือนกันและพร้อมๆกัน เมื่อมีผู้ใช้ ขออนุญาตเข้ามาทำงาน โปรแกรมจะเก็บบันทึกข้อมูลผู้ใช้และแยกเข้ากลุ่มการทำงาน ทำการสร้างอินสแตนซ์ของแอปพลิเคชันขึ้นมาอีกหนึ่งตัว ซึ่งจะมีการทำงานเป็นเซิร์ฟและไคลเอนต์ต่อกันระหว่างผู้ใช้แต่ละคน แอปพลิเคชันแต่ละตัวจะทำการรีเฟรชตัวเองเพื่อให้ได้รับข้อมูลที่ทันสมัยอยู่ตลอดเวลา ส่วนเซิร์ฟเว็ท จะทำหน้าที่ในการจัดการแก้ไขเปลี่ยนแปลงข้อมูลกลางของกลุ่มการทำงานทุกๆกลุ่ม



รูปที่ 3.9 แสดงการติดต่อกันระหว่างเซิร์ฟเว็ทกับแอปพลิเคชัน

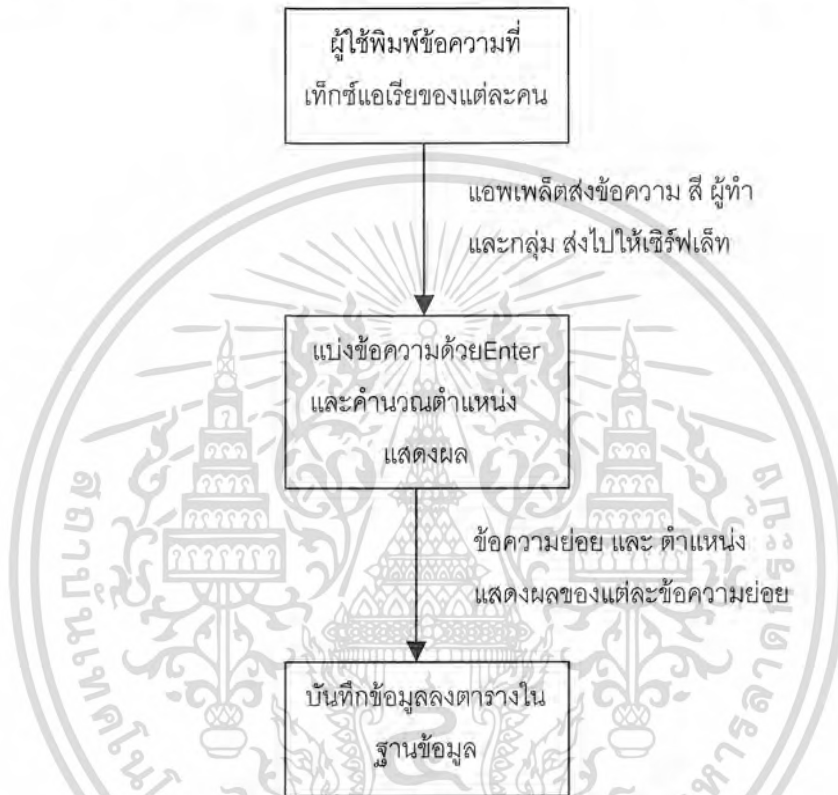
คุณสมบัติของโปรแกรมการทำงานร่วมกัน มีดังต่อไปนี้

1. การแสดงผลข้อความ

ผู้ใช้แต่ละคนในกลุ่มการทำงานสามารถพิมพ์ข้อความได้อย่างอิสระภายในเท็กซ์แเรีย (Text Area) ของตัวเอง ซึ่งส่วนเท็กซ์แเรียดังกล่าวจะเห็นได้เฉพาะเพียงผู้ใช้และยังไม่ถูกนำไปแสดงผลบน หน้าจอแสดงผลลิสต์ เมื่อทำการแก้ไขเพิ่มเติมจนเป็นที่พอใจแล้ว ผู้ใช้จะส่งไปแสดงผลยังหน้าจอแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ด้วยการกดปุ่ม “Send Messages” เมื่อแอปพลิเคชันตรวจสอบอีเวนต์ (event) และทราบว่ามีการกดปุ่มนี้ จะทำการส่งข้อมูลที่มีพารามิเตอร์ของข้อความที่พิมพ์, สีของข้อความ, ผู้พิมพ์, กลุ่มการทำงานที่สังกัด เมื่อเซิร์ฟเวอร์ได้รับข้อมูลมา จะทำการแบ่งข้อความออกเป็นข้อความย่อยๆ โดยแบ่งแยกด้วยเครื่องหมาย “Enter” แล้วคำนวณตำแหน่งแสดงผลของแต่ละข้อความย่อยนั้น คำนวณตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายของข้อความที่ส่งมา และเก็บบันทึกลงตาราง “Command Table” และ “Manage Table” ของกลุ่มการทำงาน เมื่อแอปพลิเคชันของผู้ใช้และแอปพลิเคชันอื่นๆ ทำการรีเฟรชจะได้รับข้อมูลใหม่เนื่องจากการรีเฟรชจะทำการอ่านข้อมูลในฐานข้อมูลมาประมวลผลและวาดลงบนหน้าจอแสดงผลลัพธ์



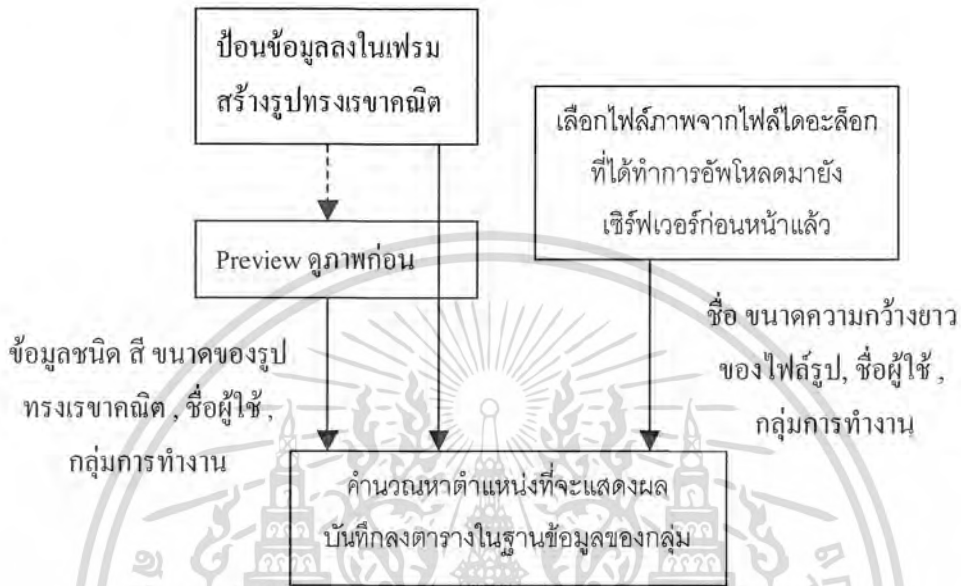
รูปที่ 3.10 แสดงขั้นตอนการแสดงผลข้อความ

2. การแสดงผลรูปภาพและรูปทรงเรขาคณิต

การแสดงผลรูปทรงเรขาคณิตสามารถแสดงได้อยู่ในรูปแบบ ได้แก่ สี เหลี่ยม วงกลม และ วงรี เมื่อผู้ใช้งานต้องการเพิ่มรูปภาพเข้าไปยังหน้าจอแสดงผลลัพธ์ จะกดปุ่ม “DrawObj” แอปพลิเคชันจะทำการสร้างเฟรมของการใส่รูปทรงเรขาคณิตขึ้น จากนั้นผู้ใช้งานจึงเลือกชนิดของรูปทรงเรขาคณิต, ขนาดความกว้างและความยาวของรูปทรงเรขาคณิต และ สีของรูปทรงเรขาคณิตที่ต้องการ หากต้องการดูตัวอย่างสามารถกดดูได้ที่ปุ่ม “preview” และเมื่อกดปุ่ม “ok” เฟรมจะทำการส่งข้อมูลที่มีพารามิเตอร์ของชนิด, ความกว้าง, ความยาว และสีของรูปทรงเรขาคณิตที่เลือก พร้อมทั้งข้อมูลชื่อเซิร์ฟเวอร์ของผู้ที่ทำการใส่รูปทรงเรขาคณิต และชื่อกลุ่มการทำงานไปยังเซิร์ฟเวอร์ หลังจากนั้นเซิร์ฟเวอร์จะทำการคำนวณตำแหน่งของรูปทรงเรขาคณิตในการแสดงผลและบันทึกลงตาราง “Command Table” และ “Manage Table” ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการแสดงผลรูปภาพจะมีขั้นตอนการทำงานในลักษณะเดียวกัน โดยมีเชื่อว่าไฟล์ภาพนั้นจะต้องได้รับการอัปเดตมาเก็บไว้ในโฟลเดอร์ “CFDC_images” ที่เซิร์ฟเวอร์เรียบร้อยแล้วก่อนที่จะมีการใส่รูปภาพดังกล่าวเข้าไปในหน้าจอแสดงผลลัพธ์ การใส่รูปภาพทำโดยการกดปุ่ม “Image” จากนั้นแอปพลิเคชันจะสร้างไฟล์ไอคอนขึ้นมาให้เลือกไฟล์ ไฟล์ที่ถูกเลือกจะถูกหาขนาดและความกว้าง แล้วส่งเป็นพารามิเตอร์ไปยังเซิร์ฟเวอร์เพื่อทำการคำนวณค่าและบันทึกลงตารางต่อไป



รูปที่ 3.11 แสดงขั้นตอนการแสดงผลรูปภาพและรูปทรงเรขาคณิต

3. การทำscrolling และการปรับเปลี่ยนตำแหน่งการแสดงผลที่จัดการโดยแอปพลิเคชัน

ค่าของข้อมูลที่บันทึกอยู่ในตารางข้อมูล ทั้งตาราง “Command Table” และตาราง “Manage Table” นั้นจะเก็บตำแหน่งที่จะแสดงผลลัพธ์ในขณะที่ไม่มีการทำการscrolling เพื่อให้สามารถทำการscrollได้แอปพลิเคชันจึงต้องมีอัลกอริทึมในการคำนวณตำแหน่งเมื่อมีการscrollเกิดขึ้น ซึ่งคิดเป็นสูตรทางคณิตศาสตร์ได้คือ

$$\text{ค่าของ โคออดิเนต Y ใหม่เมื่อทำการ scroll} = \text{ค่าโคออดิเนต Y ที่เก็บในตาราง} - (\text{ความห่างของบรรทัด} * (\text{จำนวนครั้งของการเลื่อน scroll bar ลงนับจากตำแหน่งบนสุด} - 1))$$

เมื่อผู้ใช้เลื่อนscroll bar ไปยังตำแหน่งหนึ่งค่าที่อ่านได้จากการติดตามอีเวนที่จะถูกนำไปคำนวณเพื่อปรับเปลี่ยนตำแหน่งของการแสดงผลลัพธ์ใหม่ และเนื่องจากการกระทำนี้เป็นการกระทำเฉพาะของแต่ละแอปพลิเคชันจึงไม่ส่งผลกระทบต่อแอปพลิเคชันของผู้ใช้คนอื่นๆภายในกลุ่มการทำงาน

และเช่นกันกับการแสดงผลลัพธ์ที่เป็นข้อความร่วมกับการแสดงผลรูปภาพและ/หรือรูปทรงเรขาคณิต เพื่อให้บรรทัดของการทำงานนั้นตรงกันจึงต้องมีการปรับเปลี่ยนโคออดิเนตแกน X ของข้อความให้มีระดับเดียวกับตำแหน่งโคออดิเนตแกน X ของรูปภาพและ/หรือรูปทรงทางเรขาคณิตที่มีขนาดความยาวมากที่สุด ณ บรรทัดนั้นด้วย

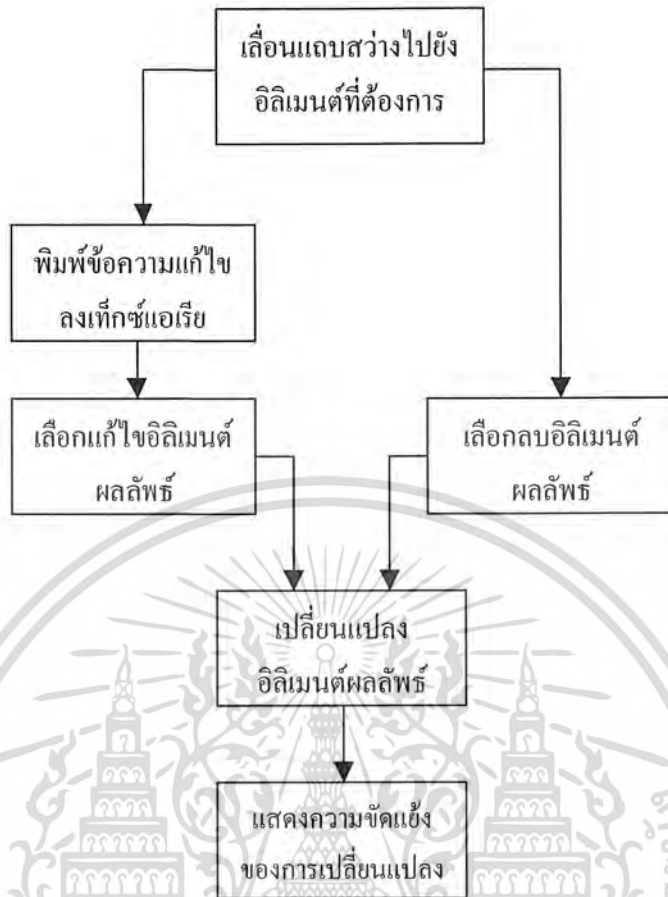
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การแก้ไขหรือลบอิลิเมนต์ของผลลัพธ์ที่แสดงผลอยู่ และการแสดงความขัดแย้งที่เกิดขึ้น ในการแก้ไขหรือลบอิลิเมนต์

ข้อความที่ถูกส่งไปแสดงยังหน้าจอแสดงผลในแต่ละครั้ง หรือ รูปภาพ หรือ รูปทรงทางเรขาคณิต จะถูกมองโดยโปรแกรมว่าเป็นอิลิเมนต์ของผลลัพธ์อิลิเมนต์หนึ่งซึ่งการทำการแก้ไข หรือ ทำการลบข้อมูลนั้นจะต้องทำคราวละหนึ่งอิลิเมนต์ เมื่อผู้ใช้ต้องการจะแก้ไขหรือลบอิลิเมนต์ผลลัพธ์ใดจะทำได้ด้วยการเลื่อนแถบสว่างสีเหลืองไปยังอิลิเมนต์ดังกล่าวด้วยปุ่ม <- สำหรับเลื่อนแถบสว่างไปทางซ้าย และ ปุ่ม -> เลื่อนแถบสว่างไปทางขวา เมื่อเลือกอิลิเมนต์ได้แล้ว หากผู้ใช้ต้องการแก้ไขผู้ใช้จะพิมพ์ข้อความที่จะแก้ไขลงในส่วนที่กซ์แเอเรียและกดปุ่ม “Replace” ซึ่งจะเป็นการแก้ไขในแบบการแทนที่ ส่วนหากต้องการลบอิลิเมนต์ผลลัพธ์สามารถกดปุ่ม “Delete” ได้เลย ข้อมูลของการขอการแก้ไขและลบอิลิเมนต์จะถูกนำไปเก็บไว้ในคิวเพื่อรอการแก้ไขหรือลบออกซึ่งคิวดังกล่าวนี้จะทำการส่งไปทำงานทุกเวลา 5 วินาที ข้อมูลของการแก้ไขหรือลบอิลิเมนต์ผลลัพธ์ที่จะส่งทุกครั้งจะบอกถึงชื่อผู้ส่ง, กลุ่มการทำงานและเวลาที่ส่งมา ข้อมูลที่มีความแตกต่างของเวลาไม่เกิน 5 วินาทีจะถูกรวบรวมจากคิวไปทำงาน สาเหตุที่ไม่ทำการแก้ไขหรือลบอิลิเมนต์ในทันทีเพื่อที่จะมีการป้องกันไม่ให้เกิดปัญหาความขัดแย้งที่เกิดขึ้นเนื่องจากการแก้ไขหรือลบอิลิเมนต์ผลลัพธ์เดียวกัน ข้อมูลที่รวบรวมได้จะถูกนำมาตรวจสอบความขัดแย้งที่เกิดขึ้นจากการแก้ไขหรือลบอิลิเมนต์เดียวกัน ด้วยเงื่อนไขดังต่อไปนี้

- ถ้าอิลิเมนต์ผลลัพธ์ใดมีการแก้ไข หรือ ลบอิลิเมนต์ผลลัพธ์เพียงครั้งเดียวอิลิเมนต์ดังกล่าวไม่เกิดความขัดแย้ง
- ถ้าอิลิเมนต์ผลลัพธ์ใดๆ มีทั้งการแก้ไขและการลบ อิลิเมนต์ผลลัพธ์ดังกล่าวเกิดความขัดแย้ง
- ถ้าอิลิเมนต์ผลลัพธ์ใดมีการแก้ไขอิลิเมนต์ผลลัพธ์มากกว่าหนึ่งครั้งหากข้อความที่แก้ไขไม่เหมือนกันถือว่าเกิดความขัดแย้งขึ้น
- ถ้าอิลิเมนต์ผลลัพธ์ใดๆ มีการลบอิลิเมนต์ผลลัพธ์เพียงอย่างเดียวมากกว่าหนึ่งครั้งจะไม่เกิดความขัดแย้งขึ้น

ความขัดแย้งที่เกิดขึ้นจะถูกบันทึกลงในตารางความขัดแย้งเพื่อแจ้งให้ผู้ใช้ทราบ ซึ่งผู้ใช้สามารถตรวจสอบความขัดแย้งได้โดยการกดปุ่ม “Conflict” หากยังไม่มีผู้ใช้ภายในกลุ่มการทำงานรับรู้ถึงความขัดแย้งที่เกิดขึ้น ระบบจะไม่ยอมให้มีการแก้ไขหรือลบอิลิเมนต์ผลลัพธ์อื่นๆเพิ่มเติม ซึ่งความขัดแย้งที่เกิดขึ้นผู้ใช้ภายในกลุ่มจะต้องหาข้อสรุปและทำการแก้ไขและลบอิลิเมนต์ดังกล่าวในคราวต่อไป



รูปที่ 3.12 แสดงการทำงานของกรแก้ไข และ ลบอิลิเมนต์ผลลัพธ์

5. การเซฟผลลัพธ์การทำงาน, การแสดงผู้ที่ใช้อยู่ในขณะนั้น และ การออกจากการทำงาน

ผลลัพธ์ของการทำงานจะถูกบันทึกผลให้อยู่ในรูปแบบของภาษาริพีรีเช่นเทซันคือXML เมื่อผู้ใช้กดปุ่ม "Save" แอปพลิเคชันจะส่งข้อมูลไปบอกเซิร์ฟเวอร์ให้ทำการเขียนไฟล์นามสกุลXMLขึ้น โดยใช้ชื่อไฟล์ตามชื่อกลุ่มการทำงานและข้อมูลในไฟล์นั้นกำหนดตามข้อมูลที่มีอยู่ในฐานข้อมูลของกลุ่มการทำงานนั้น

ส่วนการสังเกตผู้ที่ใช้งานอยู่ในขณะนั้นสามารถทำได้โดยการกดปุ่ม "Status" ซึ่งแอปพลิเคชันจะทำการสร้างเฟรมที่จะไปอ่านข้อมูลจากตาราง "Login Table" มาแสดงผลขึ้นแสดงให้ผู้ใช้ทราบ

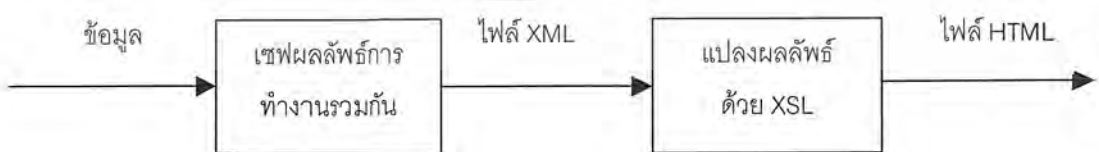
เมื่อผู้ใช้ต้องการออกจากการทำงานจะกดปุ่ม "Log out" ซึ่งแอปพลิเคชันจะลบข้อมูลออกจากตาราง "Login Table" และแสดงข้อความบอกว่าได้ออกจากโปรแกรมแล้ว ซึ่งการกดปุ่มดังกล่าวทุกครั้งเมื่อเลิกการทำงานมีความสำคัญมากเนื่องจากชื่อผู้สเซอร์เดิมจะไม่สามารถเข้ามาทำงานได้ใหม่อีกครั้งได้เพราะข้อมูลจะยังถูกเก็บไว้ในตาราง "Login Table" ดังนั้นระบบจะมองการขออนุญาตเข้าทำงานครั้งใหม่เป็นการเข้าทำงานที่ซ้ำซ้อน

3.10 การแสดงผลลัพธ์ของการทำงาน

ผลลัพธ์การทำงานของโปรแกรมการทำงานร่วมกันนี้จะแสดงผลด้วยภาษาXMLซึ่งประกอบด้วยแท็ก (Tags) ที่สำคัญคือ

- `<insert>...</insert>` เป็นแท็กที่แสดงถึงการเริ่มต้นและสิ้นสุดของผลลัพธ์การทำงาน
- `<text>...</text>` เป็นแท็กที่แสดงถึงการเขียนข้อความ
 - ประกอบด้วยแอตทริบิวต์
 - author ระบุชื่อผู้ทำการเขียนข้อความ
 - color ระบุสีของข้อความ
- `<image>...</image>` เป็นแท็กที่แสดงถึงการใส่ไฟล์ภาพและรูปทรงทางเรขาคณิต
 - ประกอบด้วยแอตทริบิวต์
 - author ระบุชื่อผู้ทำการใส่ไฟล์ภาพหรือรูปทรงทางเรขาคณิต
 - picname ระบุชื่อไฟล์ภาพหรือรูปทรงทางเรขาคณิต
 - width ระบุความกว้างของไฟล์ภาพหรือรูปทรงทางเรขาคณิต
 - height ระบุความยาวของไฟล์ภาพหรือรูปทรงทางเรขาคณิต
- `<newline/>` เป็นแท็กที่แสดงการขึ้นบรรทัดใหม่

ข้อดีของการเก็บผลลัพธ์ในรูปของภาษาXML คือ สามารถใช้ในการแปลงไปเป็นภาษามาร์คอัพอื่นๆ ได้ง่าย เช่น แปลงจากภาษาXML ไปเป็นภาษาHTML หรือ แปลงจากภาษาXML ไปเป็นภาษาWML เป็นต้น ทั้งนี้ต้องอาศัยสไตลชีต(style sheet) ที่มีลักษณะเฉพาะสำหรับการแปลงไปสู่ภาษามาร์คอัพอื่นๆ ซึ่งทางผู้จัดทำได้ทำการเขียนสไตลชีต(style sheet)ที่ชื่อ “CFMDC.xml” สำหรับแปลงไปสู่ภาษาเซชทีเอ็มแอล(HTML) เนื่องจากภาษาเซชทีเอ็มแอล(HTML)เป็นภาษาที่ได้รับความนิยม และสามารถแสดงผลผ่านทางบราวเซอร์ได้ซึ่งผู้ใช้แต่ละคนจะสามารถเซฟ(Save)ไปใช้ได้ทันที และโค้ดภาษาเซชทีเอ็มแอล(HTML)นั้นยังสามารถนำไปแก้ไขได้ง่าย ผู้ใช้จึงสามารถนำไปตัดแปลงให้มีลักษณะเฉพาะตัวของผู้ใช้เองได้ต่อไป



รูปที่ 3.13 แสดงการแปลงผลลัพธ์การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและวิเคราะห์ผลการทดลอง

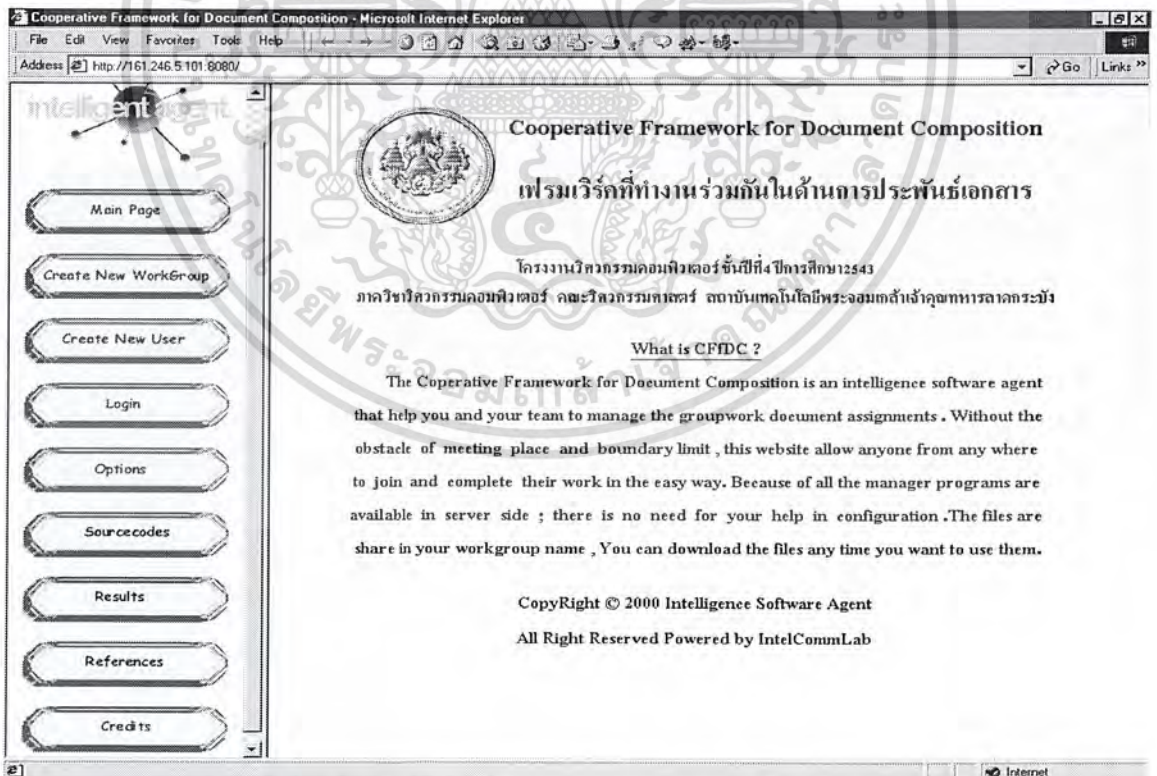
4.1 บทนำ

ในบทนี้จะเป็นการแสดงการทดลองและผลการทดลองของการใช้งานโปรแกรมการทำงานร่วมกันในงานเอกสาร และทำการวิเคราะห์ถึงผลของการทำงาน ปัญหาและแนวทางแก้ไขปัญหาที่เกิดขึ้น

4.2 ส่วนประกอบและคำอธิบายโปรแกรมการทำงาน

โปรแกรมการทำงานแบ่งออกเป็น 2 ส่วน ประกอบด้วย เฟรมเมนูที่อยู่ทางด้านซ้ายมือ และ เฟรมแสดงผลที่อยู่ทางด้านขวา

เฟรมเมนูจะบรรจุไอคอน(Icons)ที่เกี่ยวกับการทำงานทั้งหมดของโปรแกรมการทำงาน เมื่อผู้ใช้เลือกไอคอนใดๆ โปรแกรมจะทำการลิงค์ไปยังเพจที่จัดหน้าที่จัดการงานเหล่านั้น โดยแต่ละเว็บเพจจะถูกสร้างและควบคุมการทำงานด้วยเซิร์ฟเล็ต เฟรมแสดงผลมีหน้าที่เป็นอินเทอร์เฟซกับผู้ใช้ในการรับข้อมูลและแสดงผลลัพธ์ของการทำงาน



รูปที่ 4.1 แสดงรูปของโปรแกรมการทำงานร่วมกันในงานเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบและใช้งานโปรแกรม

4.3.1 ตัวอย่างการสร้างกลุ่มการทำงานใหม่

ผู้ใช้งานป้อนข้อมูลชื่อกลุ่มการทำงาน, รหัสผ่านกลุ่มการทำงาน, ยืนยันรหัสผ่านกลุ่มการทำงาน, รายละเอียดลักษณะงาน และชื่อผู้สร้างกลุ่มการทำงาน และกดปุ่มsubmit จะมีการสร้างตารางขึ้นเพื่อเก็บข้อมูลภายในกลุ่มการทำงาน และบันทึกข้อมูลลงตาราง“CFFDC_workgroup”เพื่อบอกถึงการเพิ่มขึ้นของกลุ่มทำงานใหม่

Workgroup Registration

Name of your WorkGroup : *

WorkGroup Password : *

Re-Enter Password : *

About Work (Comment) :

(optional)

Creator Name : (optional)

รูปที่ 4.2 แสดงตัวอย่างการป้อนข้อมูลในการสร้างกลุ่มทำงานใหม่

```
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID
ACCOUNT	TABLE	
BONUS	TABLE	
CFFDC_HOMWORK1_COMMANDTABLE	TABLE	
CFFDC_HOMWORK1_CONFLICTTABLE	TABLE	
CFFDC_HOMWORK1_LOGINTABLE	TABLE	
CFFDC_HOMWORK1_MANAGETABLE	TABLE	
CFFDC_HOMWORK1_QUEUETABLE	TABLE	
CFFDC_USER	TABLE	
CFFDC_WORKGROUP	TABLE	
DEPT	TABLE	
EMP	TABLE	
RECEIPT	TABLE	
SALGRADE	TABLE	
TELEPHONE	TABLE	

14 rows selected.

รูปที่ 4.3 แสดงตัวอย่างการสร้างตารางเพื่อใช้ในการทำงานภายในกลุ่มการทำงาน

```
SQL> select * from cffdc_workgroup;
```

ID_NO	WG_NAME	WG_PASSWOR	WG_COMMENT	CREATOR
1	Homework1	pw1	Essey about WAP technology	Intharachai Sukonthaman

รูปที่ 4.4 แสดงตัวอย่างการบันทึกข้อมูลลงตารางกลุ่มการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ตัวอย่างการลงทะเบียนผู้ใช้งาน

การลงทะเบียนผู้ใช้งาน ทำโดยผู้ใช้งานป้อนข้อมูลส่วนตัวของผู้ใช้ และ ชื่อยูสเซอร์ที่จะใช้ในขณะทำงาน รหัสผ่านของผู้ใช้ และ ยืนยันรหัสผ่านของผู้ใช้ แล้วกดปุ่ม “submit” ข้อมูลของผู้ใช้คนดังกล่าวจะถูกบันทึกลงตารางผู้ใช้งาน

รูปที่ 4.5 แสดงตัวอย่างการป้อนข้อมูลการลงทะเบียนผู้ใช้งาน

```
SQL> select * from cffdc_user;
```

ID_NO	USER_NAME	USER_PASSW	REAL_NAME	ADDRESS	COUNTRY	GENDER	EMAIL	PHONE	ICQ	BDAY
18	Intharachai	in	Intharachai Sukonthaman	158/11 Pattanakarn78 Pravate Bangkok10250	Thailand	male	intharachai@mailcity.com	3223268	41301747	16 may 1980

รูปที่ 4.6 แสดงตัวอย่างการบันทึกข้อมูลลงตารางผู้ใช้งาน

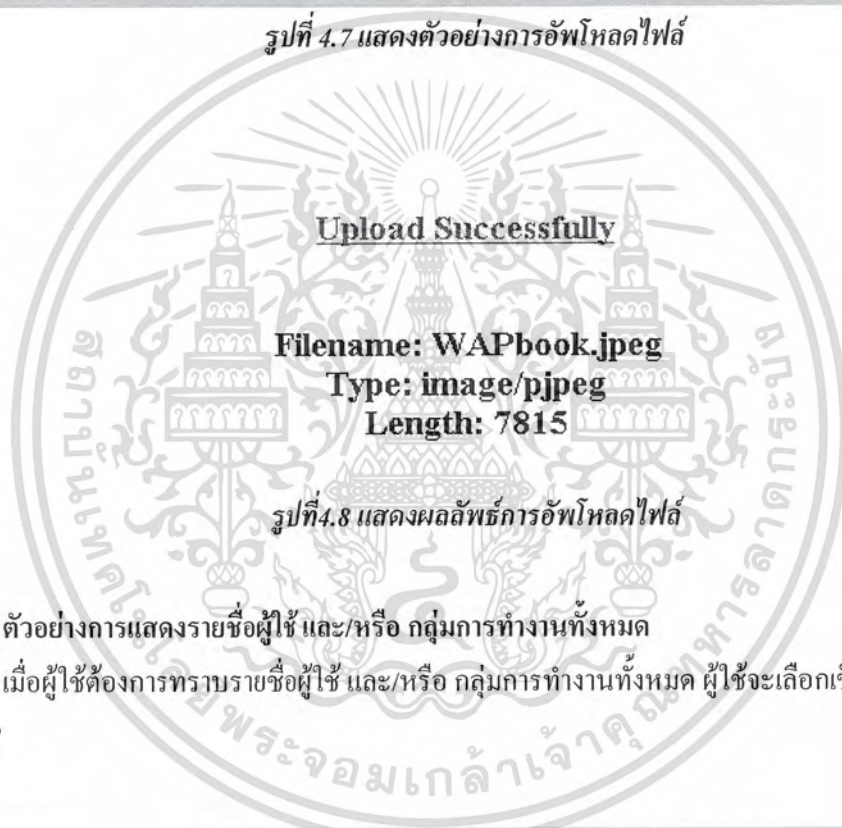
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ตัวอย่างการอัปโหลดไฟล์ภาพมายังเซิร์ฟเวอร์

การอัปโหลดไฟล์ภาพมายังเซิร์ฟเวอร์ ผู้ใช้จะทำการเลือกไฟล์ภาพจากไฟล์ไดอะล็อก(File Dialog) หากสามารถอัปโหลดมาได้ โปรแกรมจะแจ้งชื่อไฟล์ที่อัปโหลด, ชนิดของไฟล์ และ ขนาดของไฟล์



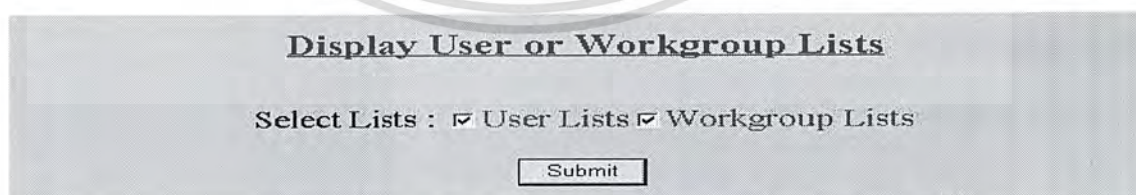
รูปที่ 4.7 แสดงตัวอย่างการอัปโหลดไฟล์



รูปที่ 4.8 แสดงผลลัพธ์การอัปโหลดไฟล์

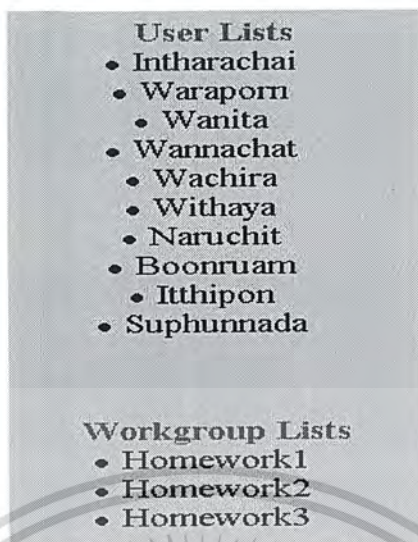
4.3.4 ตัวอย่างการแสดงรายชื่อผู้ใช้ และ/หรือ กลุ่มการทำงานทั้งหมด

เมื่อผู้ใช้ต้องการทราบรายชื่อผู้ใช้ และ/หรือ กลุ่มการทำงานทั้งหมด ผู้ใช้จะเลือกเช็บบ็อกแล้วกดปุ่ม “submit”



รูปที่ 4.9 แสดงตัวอย่างการเลือกการแสดงผลรายชื่อผู้ใช้และรายชื่อกลุ่มการทำงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงตัวอย่างรายชื่อผู้ใช้ และ รายชื่อกลุ่มการทำงานทั้งหมด

4.3.5 ตัวอย่างแสดงการขออนุญาตเข้าทำงาน

การขออนุญาตเข้าทำงาน ผู้ใช้จะกรอกข้อมูลชื่อกลุ่มการทำงาน, รหัสผ่านกลุ่มการทำงาน, ชื่อยูสเซอร์ และ รหัสผ่านบัญชีผู้ใช้ โปรแกรมทำการตรวจสอบความถูกต้อง หากข้อมูลถูกต้องจะเข้าสู่หน้าจอการทำงานร่วมกันต่อไป

The image shows a login dialog box with the following text and fields:

Please enter your UserName , WorkgroupName and all accurate Passwords to log in.

WorkGroup Name : ※

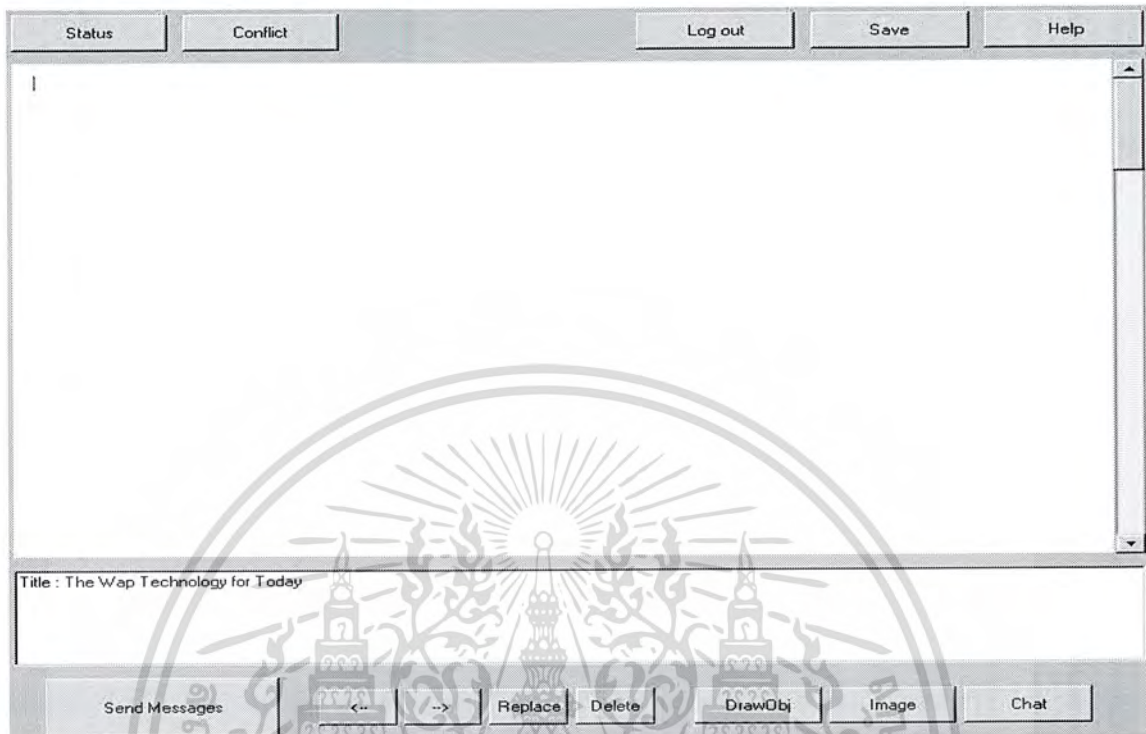
WorkGroup Password : ※

User Name : ※

User Password : ※

รูปที่ 4.11 แสดงตัวอย่างการป้อนข้อมูลการขออนุญาตเข้าทำงาน

4.3.6 ตัวอย่างการทำงานร่วมกันในการเพิ่มข้อความและรูปภาพและรูปทรงเรขาคณิต

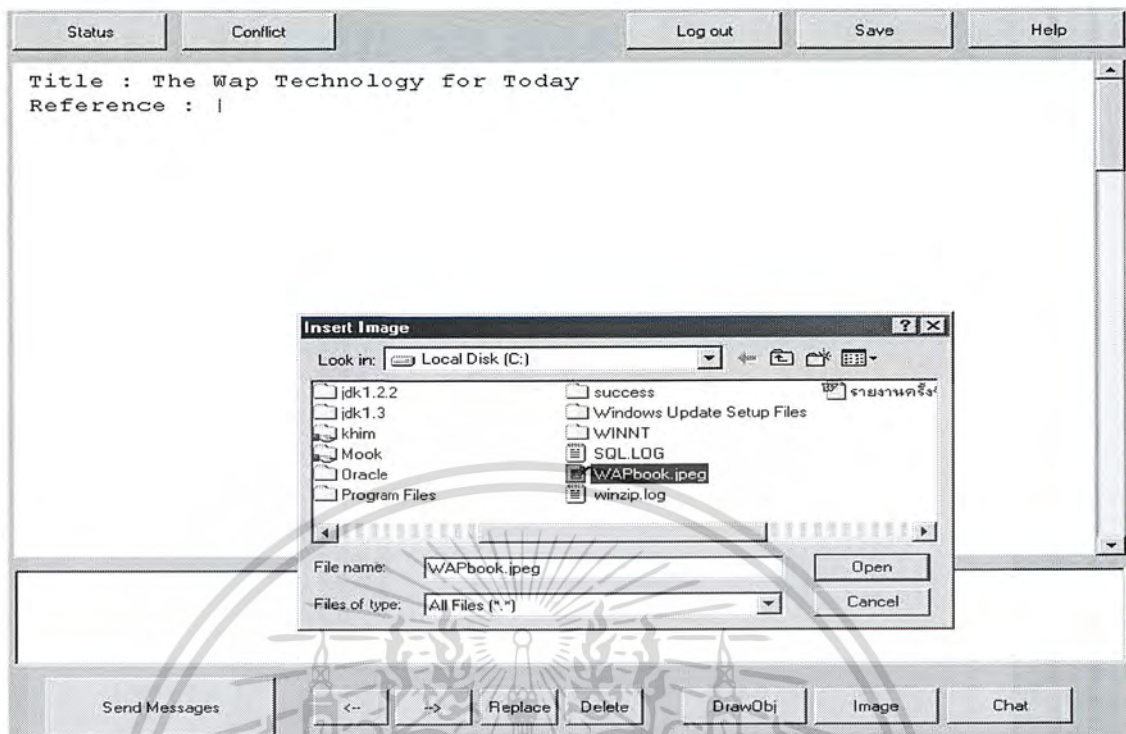


รูปที่ 4.12 แสดงตัวอย่างการเพิ่มข้อความ โดยผู้ใช้ป้อนข้อมูลที่เท็กซ์แอเรีย แล้วกดปุ่ม Send Messages

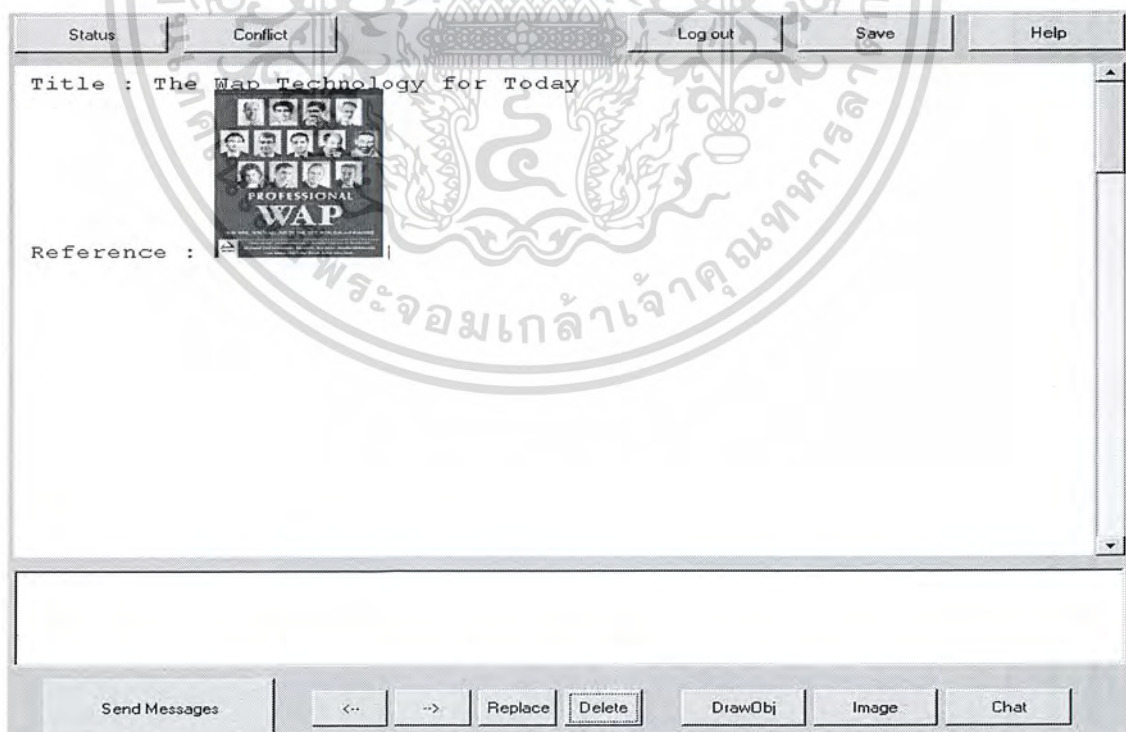


รูปที่ 4.13 แสดงตัวอย่างผลลัพธ์การเพิ่มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงตัวอย่างการทำการใส่ไฟล์รูปภาพ โดยผู้ใช้เลือกไฟล์จากไฟล์ไดอะล็อก

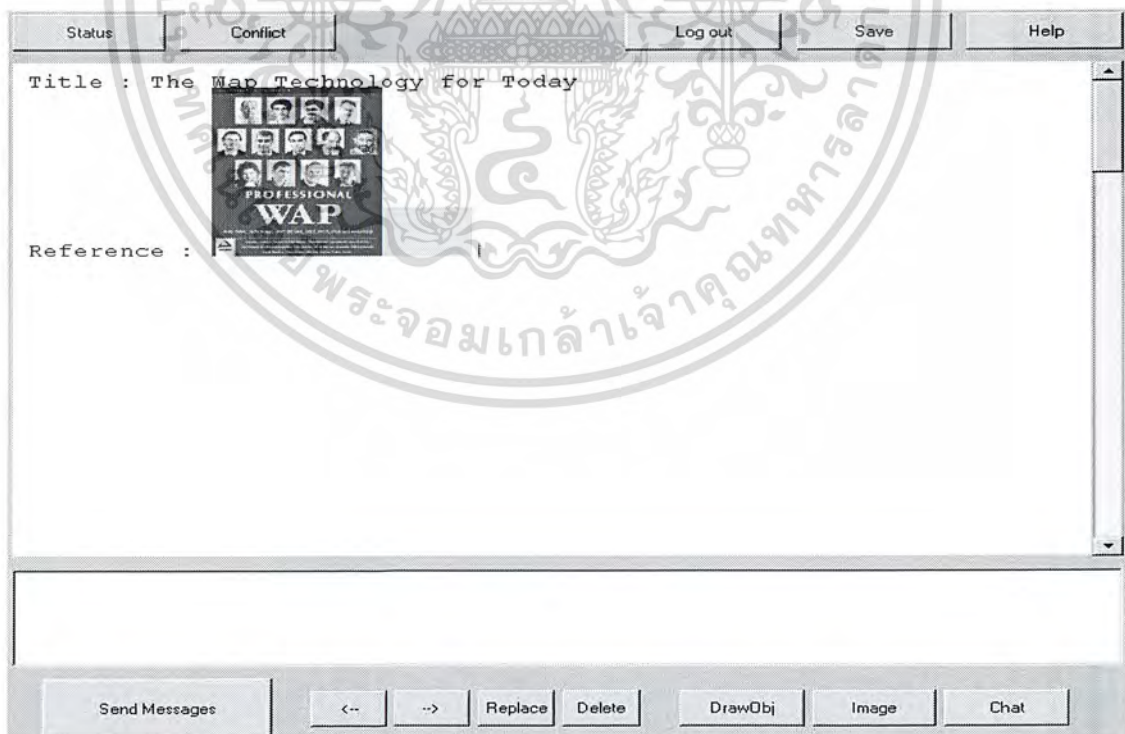


รูปที่ 4.15 แสดงผลลัพธ์การเพิ่มไฟล์ภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



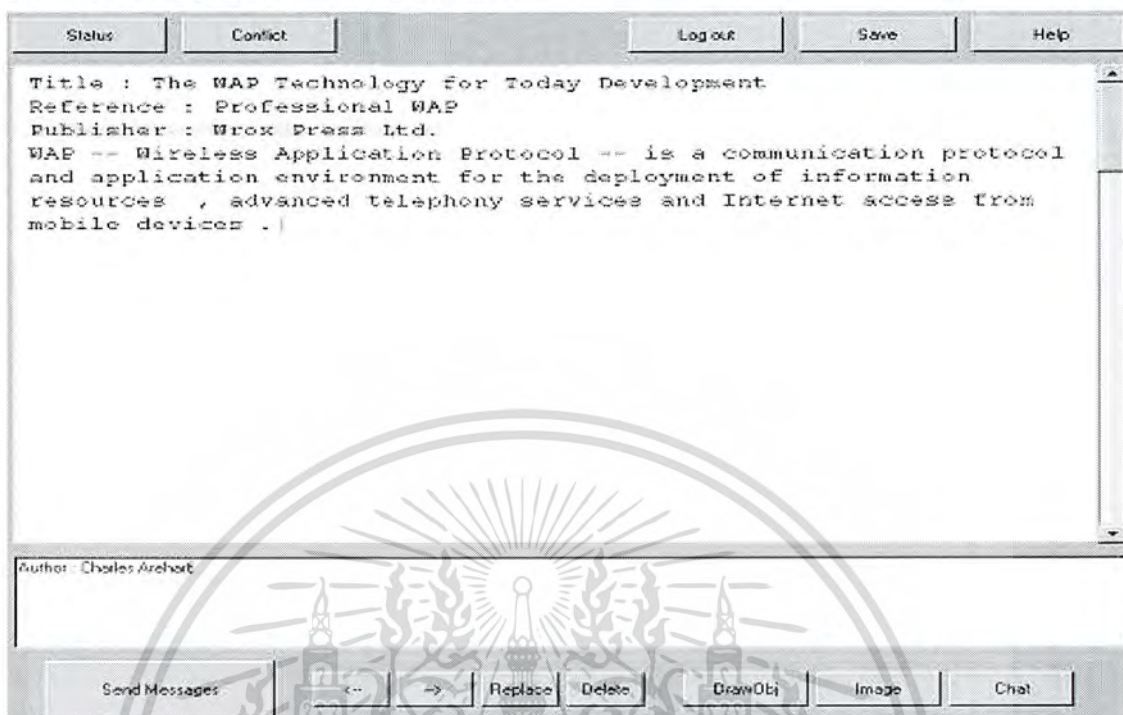
รูปที่ 4.16 แสดงการเพิ่มรูปทรงทางเรขาคณิต โดยผู้ใช้กำหนดชนิด ขนาด และสีของรูปทรงเรขาคณิตและสามารถPreview ก่อนทำการเพิ่มรูปทรงเรขาคณิตด้วยการกดปุ่ม Ok



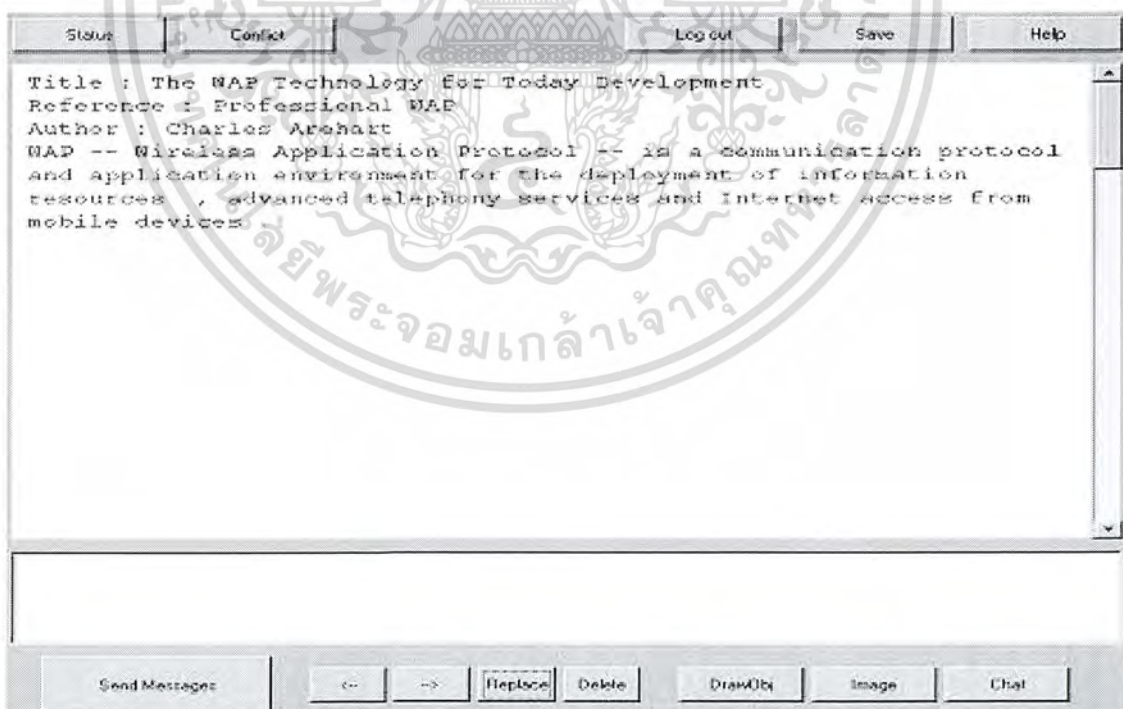
รูปที่ 4.17 แสดงผลลัพธ์ของการเพิ่มรูปทรงเรขาคณิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.7 ตัวอย่างการแก้ไข และลบอิลิเมนต์ผลลัพธ์

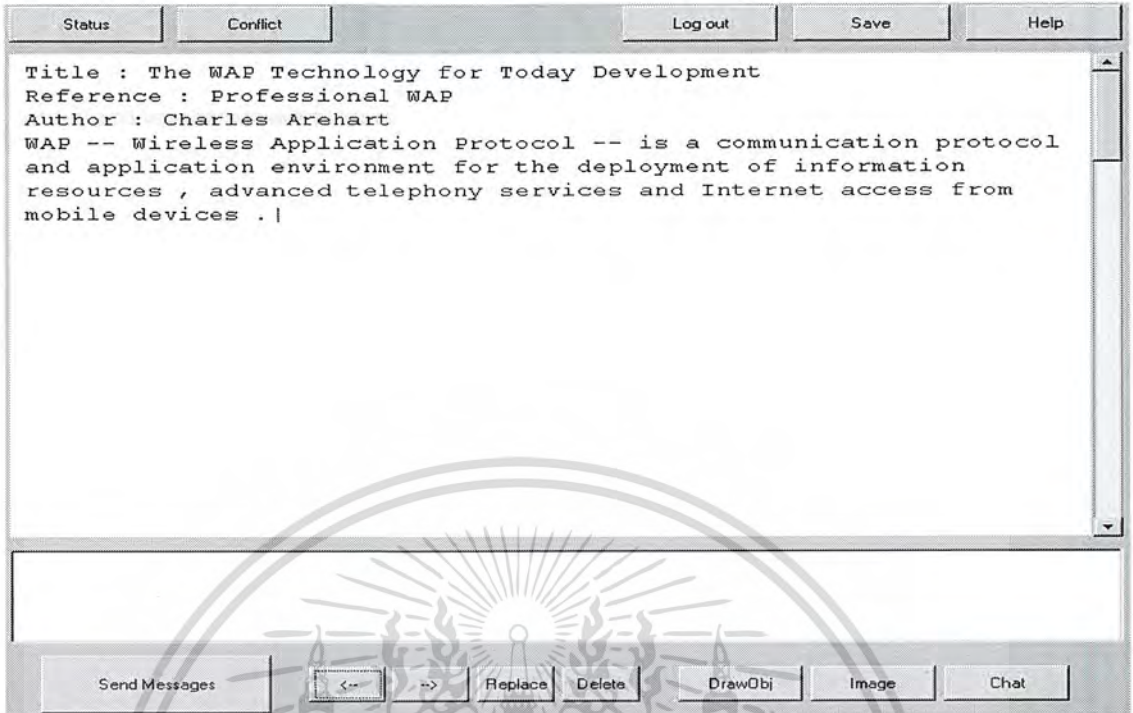


รูปที่ 4.18 แสดงการแก้ไขอิลิเมนต์ผลลัพธ์ด้วยการแทนที่ข้อความในแถบสว่าง
ด้วยข้อความในเท็กซ์แอเรีย เมื่อผู้ใช้คลิกปุ่ม Replace

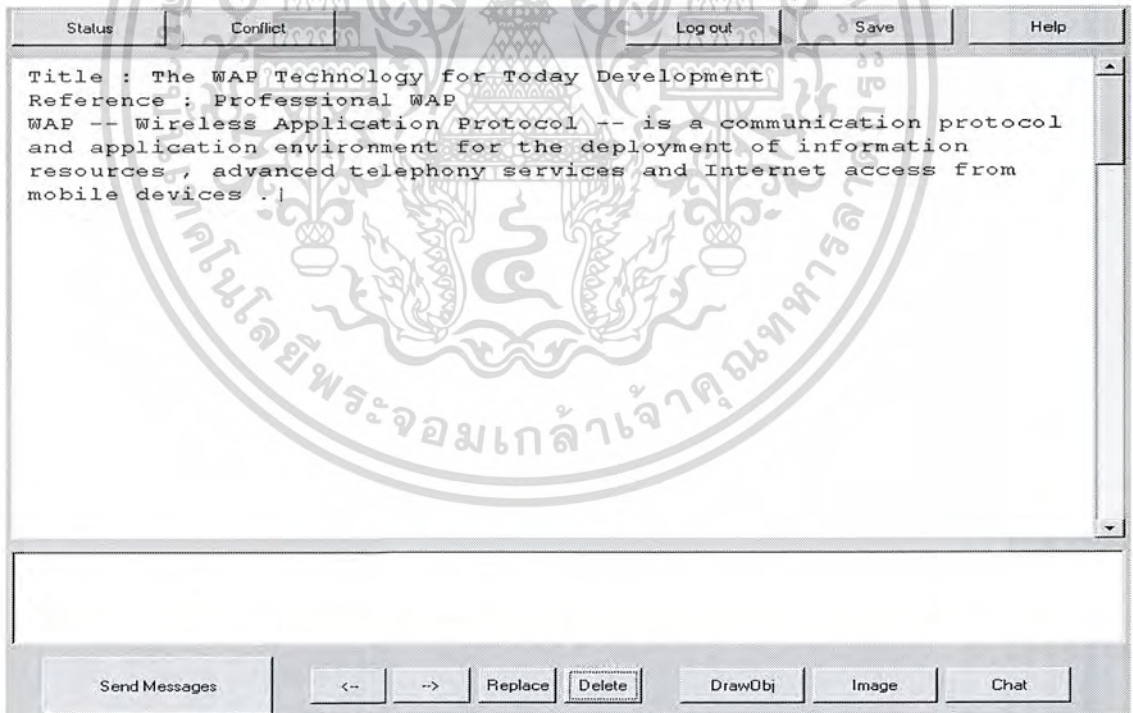


รูปที่ 4.19 แสดงผลลัพธ์การแก้ไขอิลิเมนต์ผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 แสดงตัวอย่างการลบอิลิเมนต์ผลลัพธ์ โดยการใช้ปุ่ม Delete

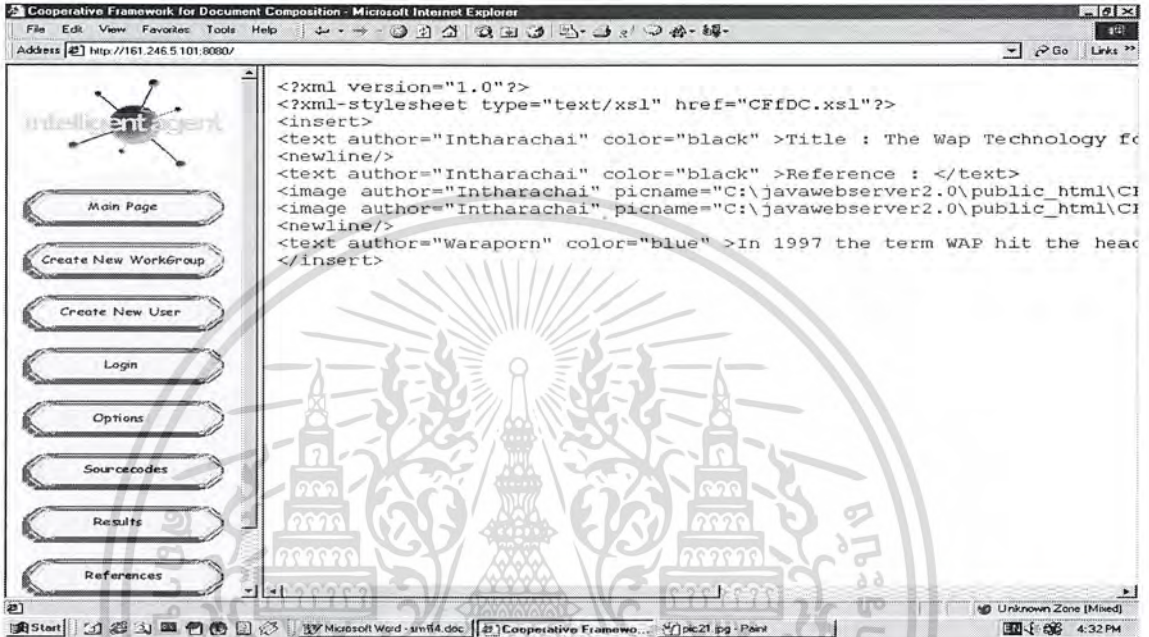


รูปที่ 4.21 แสดงผลลัพธ์การลบอิลิเมนต์ผลลัพธ์

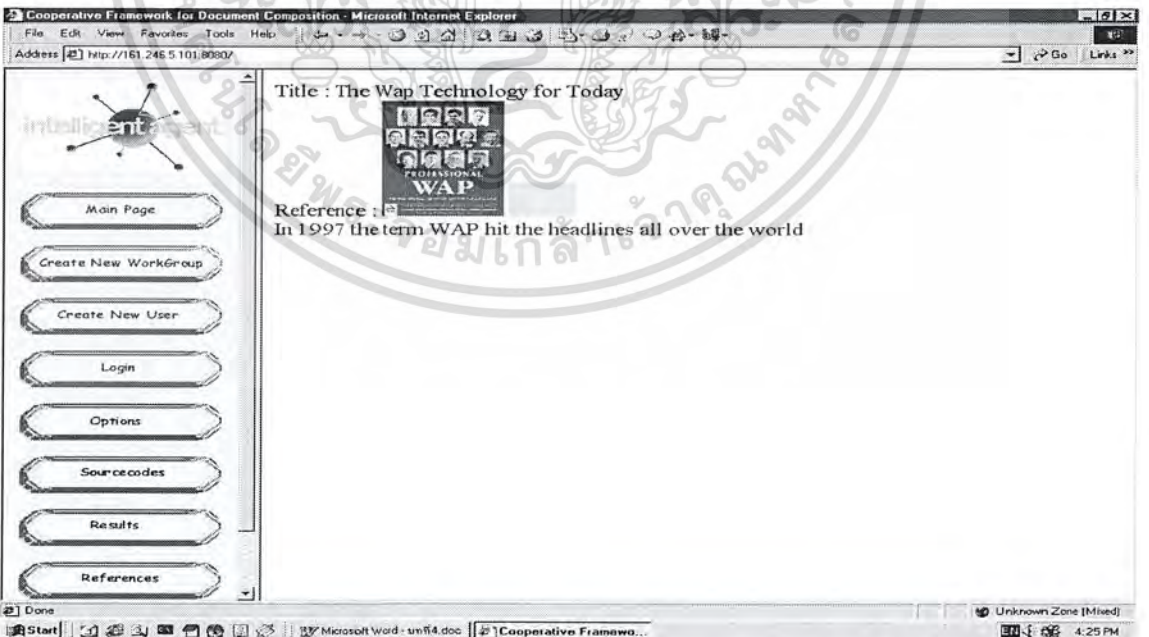
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.8 ตัวอย่างซอร์สโค้ดภาษาXML และ ผลลัพธ์ภาษา HTML

ผู้ใช้งานจะสามารถนำผลลัพธ์การทำงานร่วมกันไปใช้ได้สองทางคือ การเซฟเป็นภาษาXML หรือ การเซฟเป็นภาษาHTML ซึ่งซอร์สโค้ดภาษาXML ได้จากการสร้างโดยตรงจากเซิร์ฟเล็ต ส่วนภาษาHTML ได้จากการแปลงภาษาXML ด้วย xsl style sheet



รูปที่ 4.22 แสดงตัวอย่างซอร์สโค้ดของการทำงานหลังจากผู้ใช้ กดปุ่มSave



รูปที่ 4.23 แสดงตัวอย่างผลลัพธ์ภาษา HTML ที่ได้จากการแปลงด้วย CFfDC.xsl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ปัญหาที่เกิดขึ้นและทางแก้ไข

4.4.1 ปัญหาจากการทำงานร่วมกันของผู้ใช้ภายในกลุ่มการทำงานเดียวกัน

4.4.1.1 การแสดงความเป็นเอกลักษณ์ของผู้ใช้แต่ละคน

การทำงานร่วมกันจะไม่สามารถรับรู้ได้เลยว่าข้อความที่แสดงผลนั้นเกิดจากการกระทำของผู้ใช้คนใด หากไม่มีการแสดงความเป็นเอกลักษณ์ของผู้ใช้แต่ละคน โดยสิ่งที่จะแบ่งแยกผู้ใช้แต่ละคนได้ดีที่สุดนั้นคือ การกำหนดสีของข้อความให้กับผู้ใช้แต่ละคนแตกต่างกันไป โดยในขณะที่ทำการขออนุญาตเข้าใช้ โปรแกรมจะทำการกำหนดสีให้ผู้ใช้ด้วยการนำอันดับของการเข้าใช้มาคำนวณผลทางคณิตศาสตร์ ค่าที่ได้จะแทนด้วยสีของข้อความ การทำงานร่วมกันจึงสามารถทราบได้ว่าการเปลี่ยนแปลงผลลัพธ์การทำงานครั้งดังกล่าวใครเป็นผู้กระทำจากการสังเกตจากสีของข้อความ

4.4.1.2 ความสัมพันธ์ของระยะทางและลำดับก่อนหลัง

เนื่องจากโปรแกรมการทำงานร่วมกันในงานเอกสารนี้ทำงานภายใต้เครือข่ายเวิร์ลไวด์เว็บซึ่งทำให้ผู้ใช้สามารถทำงานร่วมกันได้จากทุกที่ ระยะทางที่ต่างกันของสถานที่ใช้งานและเว็บเซิร์ฟเวอร์นั้นส่งอาจส่งผลให้ลำดับของการเปลี่ยนแปลงข้อมูลผลลัพธ์ไม่ถูกต้องตามความเป็นจริง การแก้ปัญหาทำได้โดยการทำการซิงโครไนซ์(synchronize)เมธอดซึ่งเป็นคุณสมบัติหนึ่งในจาว่าที่จะทำให้การทำงานเป็นไปตามความสำคัญแบบก่อนหลัง โดยไม่เกี่ยวข้องกับระยะทาง

4.4.1.3 อีสระของผู้ใช้ในการจัดการงาน มีเคอร์เซอร์ (Cursor)อีสระ

ผู้ใช้แต่ละคนในกลุ่มการทำงานจะต้องมีอีสระในการทำงานเป็นของตนเอง ฉะนั้นการใช้เคอร์เซอร์กลางร่วมกันจะทำให้ความคล่องตัวในการทำงานลดลงเนื่องจากอาจเกิดการแย่งกันใช้เคอร์เซอร์ได้ ทางแก้ไขคือ ให้ผู้ใช้แต่ละคนสามารถควบคุมเคอร์เซอร์(แถบสว่างสี่เหลี่ยม)ได้อย่างอิสระ

4.4.2 ปัญหาการจัดการการทำงานหลายกลุ่มการทำงานพร้อม ๆ กันของเซิร์ฟเล็ต

เซิร์ฟเล็ตนั้นเป็นตัวกลางในการจัดการการทำงานร่วมกันของทุก ๆ กลุ่มการทำงาน ดังนั้นการกระทำแอคชัน(Action)ใดๆ จะต้องมีการแจ้งให้เซิร์ฟเล็ตรู้ว่าเป็นการกระทำกับกลุ่มการทำงานใด เมื่อเซิร์ฟเล็ตทราบว่ากระทำมาจากลุ่มการทำงานใดจะสามารถทำการตอบสนองการทำงานได้ถูกต้อง

4.4.3 ปัญหาของการแสดงผลภาพ

4.4.3.1 ปัญหารูปที่ใส่จะต้องอยู่ที่เซิร์ฟเวอร์

รูปภาพที่จะนำไปแสดงผลภาพที่หน้าจอแสดงผลนั้นต้องมีข้อจำกัดคือจะต้องอยู่ในโฟลเดอร์ที่อยู่ในเซิร์ฟเวอร์เท่านั้น หากไฟล์ภาพไม่อยู่ที่เซิร์ฟเวอร์ โปรแกรมจะหารูปภาพไม่เจอและไม่สามารถเรียกออกมาแสดงผลได้ อีกทั้งโปรแกรมจะไม่สามารถหาตำแหน่งความกว้าง ความยาวของภาพได้ทำให้ค่าที่อ่านได้เป็น -1 ซึ่งทำให้การคำนวณตำแหน่งผิดไป การทำงานครั้งต่อไปจึงคลาดเคลื่อนตามไปด้วย จึงเป็นข้อควรระวังอย่างยิ่ง

4.4.3.2 ปัญหาขนาดภาพที่ใหญ่เกินไป

ไฟล์ที่มีขนาดใหญ่เกินไปจะไม่สามารถแสดงผลได้เนื่องจากขนาดแอปเพล็ตที่กำหนดไว้มีขนาดเล็กกว่าไฟล์ภาพที่ต้องการแสดงผล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

5.1 บทสรุปและวิจารณ์

การพัฒนาเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสารนั้นเป็นไปตามหลักการการทำงานร่วมกันผ่านทางคอมพิวเตอร์เพราะเป็นการจัดทำเอกสารร่วมกันผ่านเครือข่ายทางอินเทอร์เน็ต (Internet) ทำให้ผู้ใช้งานสามารถจัดทำเอกสารร่วมกันแม้ไม่ได้อยู่ที่เดียวกันได้ และประหยัดเวลาในการทำงาน โดยที่กล่าวมานั้นเป็นข้อดีของการพัฒนาที่เป็นไปตามวัตถุประสงค์ที่วางไว้ และในการพัฒนาก็มีการแก้ไขปรับปรุงเพื่อให้สามารถตอบสนองความต้องการของผู้ใช้และขจัดปัญหาที่เกิดขึ้นระหว่างพัฒนาตลอดมาซึ่งเป็นไปตามหลักของการแก้ไขปรับปรุงการทำงานในการทำงานร่วมกัน (Tailoring) หากแต่การพัฒนาเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสารที่พัฒนามานั้นยังเป็นเพียงการพัฒนาเบื้องต้นเพื่อให้เห็นแนวทางในการพัฒนาที่เป็นไปได้ในการทำงานจริงว่าสามารถนำคอมพิวเตอร์มาช่วยเพิ่มประสิทธิภาพในการจัดทำเอกสารร่วมกันได้เพื่อรองรับความต้องการและเพิ่มความสะดวกให้กับผู้ใช้นั้น การจัดทำเอกสารยังคงเป็นเพียงการจัดทำง่าย ๆ และยังคงมีปัญหาอยู่ไม่น้อยในการจัดการกับการทำงานที่มีผู้ใช้หลายคน (Multi user) ซึ่งเป็นการจัดการที่ยุ่งยาก รวมถึงการจัดการกับรูปแบบของเอกสารที่ยังคงไม่เป็นอิสระ โดยหากได้รับการนำแนวความคิดและการพัฒนาไปพัฒนาต่อเพื่อให้มีประสิทธิภาพเพิ่มมากขึ้นแล้วคาดว่าในอนาคตก็มีความเป็นไปได้ที่จะมีตัวอิดีเตอร์บนเว็บที่สามารถอำนวยความสะดวกในการจัดทำเอกสารร่วมกันได้ แต่ก็ยังคงเป็นสิ่งที่ต้องอาศัยระยะเวลาในการศึกษาและพัฒนาอีกมากเพื่อให้สามารถตอบสนองความต้องการของผู้ใช้ได้ในทุกๆด้าน และในการที่จะทำให้ตัวอิดีเตอร์มีฟังก์ชัน (Function) ที่สนับสนุนการทำงานอย่างสมบูรณ์ก็ยังเป็นเรื่องที่ห่างไกลอยู่ไม่น้อยเลยทีเดียว

5.2 ปัญหาที่ยังไม่ได้แก้ไข

ปัญหาที่ยังไม่ได้แก้ไขหรืออาจกล่าวได้ว่าเป็นข้อเสียของการพัฒนาเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสารมีดังต่อไปนี้ คือ

1. การจัดการเกี่ยวกับการแทนที่รูป, การลบรูป ยังคงเป็นปัญหาอยู่ การลบรูปนั้นได้ทดลองแก้ปัญหาคือแล้ว หากแต่ให้ผลลัพธ์ที่ออกมาผิดไปจากความต้องการ เนื่องจากมีความยุ่งยากในการจัดการเกี่ยวกับตำแหน่งของรูปที่มีมาแตกต่างกันไป เมื่อลบแล้วเกิดการทำให้ค่า โคออดิเนต (co-ordinate) ผิดเพี้ยนไป
2. การจัดการเกี่ยวกับความซ้ำซ้อนของการแก้ไข และการลบที่ เอลิเมนต์ (Element) เดียวกัน นั่นก็คือปัญหาความขัดแย้งที่มีการกระทำเกิดขึ้นในเวลาเดียวกันนั่นเอง โดยในการพัฒนาสามารถรับรู้ได้ว่ามีความขัดแย้งและรายงานผลความขัดแย้งได้ หากแต่ยังไม่สามารถให้สิทธิอำนาจแก่ผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการจัดการเกี่ยวกับความขัดแย้งเพื่อก่อให้เกิดความยุติธรรมที่สุดได้ เนื่องจากเป็นเรื่องที่มีความซับซ้อนเพราะมีความเกี่ยวข้องกับการตัดสินใจบุคคลซึ่งยากจะคาดเดาเป็นผลให้อาจเกิดความขัดแย้งซ้อนกันหลายระดับได้ การจัดการจึงเป็นเรื่องที่ยาก

3. การจัดการเกี่ยวกับตัวอักษรในเรื่องของฟอนต์ (Font), ขนาด (Size), ตัวหนา (Bold), ตัวเอียง (Italic), ตัวขีดเส้นใต้ (Underline) เนื่องจากตัวโปรแกรมที่พัฒนายังไม่สนับสนุนลักษณะดังกล่าว และสาเหตุของการที่ยังไม่สามารถเปลี่ยนฟอนต์ได้ ก็คือ การจัดการเกี่ยวกับตำแหน่งและบรรทัดนั้นเกิดจากการจัดการเกี่ยวกับจำนวนตัวอักษรในบรรทัดโดยฟอนต์ที่ตั้งไว้จะมีขนาดของตัวอักษรเท่ากันแต่ถ้าหากเปลี่ยนฟอนต์จะทำให้ขนาดตัวอักษรไม่เท่ากันทำให้ไม่สามารถนับจำนวนตัวอักษรต่อหนึ่งบรรทัดได้ การจัดการต่างๆก็จะผิดเพี้ยนตามไปด้วยนั่นเอง
4. การจัดการเกี่ยวกับการใช้ภาษาไทย โดยทั่วไปแล้วแอปเพล็ต (Applet) จะสามารถจัดการเกี่ยวกับการใช้ภาษาไทยได้ แต่ในการพัฒนามีการทำในลักษณะของ ดับเบิล บัฟเฟอร์ (Double Buffer) เพื่อแก้ปัญหาการกระพริบและการทำลักษณะนี้ส่งผลให้ใช้ภาษาไทยไม่ได้ คาดว่าจะมาจากการที่ไม่สนับสนุน (Support) ฟอนต์ภาษาไทยนั่นเอง
5. การย้อน (Undo) และการ ีดู (Redo) ยังคงเป็นปัญหาอยู่โดยในช่วงแรกของการพัฒนาสามารถทำได้เนื่องจากการจัดการกับเคอร์เซอร์ (Cursor) ในการพัฒนาช่วงแรกนั้นยังไม่เป็นอิสระและจะอยู่ข้างท้ายสุดของข้อความเสมอทำให้สามารถทำการย้อนและรีดูได้เพราะมีการเก็บข้อมูลในฐานข้อมูลและการจัดการกับตำแหน่งไม่ยุ่งยาก แต่เมื่อมีการพัฒนาต่อมาให้เคอร์เซอร์ (Cursor) เป็นอิสระสามารถแทรกข้อมูล ณ ตำแหน่งที่ต้องการได้เป็นผลให้การทำการย้อนและรีดูเกิดปัญหาขึ้น หากจะทำการย้อนหรือรีดูจะต้องแก้ไขข้อมูลใหม่โดยการพิมพ์แทรกเข้าไปเอง
6. การเคาะแป้นเอนเทอร์ (Enter) ก่อนทำการคีย์ (Key) ข้อมูลในบางครั้งส่งผลก่อให้เกิดความผิดพลาดขึ้นได้ แต่เป็นในบางครั้งเท่านั้น ไม่ใช่ทุกครั้งที่มีการทำลักษณะเช่นนี้

5.3 แนวทางการพัฒนา

แนวทางการพัฒนามีดังต่อไปนี้

1. การจัดการเกี่ยวกับความขัดแย้ง (Conflict) : สามารถพัฒนาต่อไปได้เพื่อให้มีกระบวนการในการตรวจสอบความขัดแย้งที่มีประสิทธิภาพมากขึ้น เช่นนอกจากจะแจ้งให้ผู้ใช้ทราบว่าเกิดความขัดแย้งในการทำงานอย่างไร กล่าวคือ มีการกระทำที่เกิดขึ้นในเวลาเดียวกันของผู้ใช้คนใด อย่างไรแล้ว ก็สามารถพัฒนาให้ผู้ใช้มีอำนาจในการตัดสินใจเกี่ยวกับความขัดแย้งมากขึ้น เพื่อความเสมอภาคกันแก่ผู้ใช้ที่ทำงานร่วมกันทุกคนเท่าเทียมกัน โดยอาจจะทำในลักษณะของการป๊อปอัพ (Pop up) ไดอะล็อก (Dialog) ไปแจ้งให้กับผู้ใช้แต่ละคนรู้ว่ามีความขัดแย้งเกิดจากผู้ใดบ้างและให้ผู้ใช้ออกเสียงว่าจะเลือกให้สิทธิ์ผู้ใช้คนใดทำโดยการคลิกเลือก เป็นต้น หากผู้ใช้คนใดได้คะแนนโหวต (Vote) สูงสุดก็ให้ผู้ใช้คนนั้นได้ทำตามความต้องการคล้ายกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้สิทธิ์การใช้ทรัพยากรร่วมกันไปนั่นเอง แต่หากยังมีความขัดแย้งอยู่ เช่น มีคนได้คะแนน โหวตเท่ากัน ก็อาจจะต้องมีการโหวตใหม่ หรือมีการจัดการอย่างอื่นที่จะทำให้ได้ผลเป็นที่พอใจแก่ผู้ใช้งานทุกคน เป็นต้น โดยจะใช้หลักการในการพัฒนาอย่างไรรันนั้นก็ขึ้นกับการตัดสินใจของผู้พัฒนาต่อ แต่สิ่งที่ควรคำนึงถึงก็คือ ความพอใจของผู้ใช้งานเพราะโปรแกรมที่ทำงานเกี่ยวกับผู้ใช้งานหลายคน (Multiuser) ที่มีประสิทธิภาพที่ดีนั้นคือการตอบสนองความต้องการ (Requirement) ของผู้ใช้ได้ดีที่สุด แต่กระนั้นความต้องการของผู้ใช้ก็เป็นสิ่งที่สามารถเปลี่ยนแปลงได้ตลอดเวลาเพราะฉะนั้นจึงอาจกล่าวได้ว่าการจัดการเกี่ยวกับผู้ใช้งานหลายคน (Multiuser) เป็นสิ่งที่ยุ่งยากและซับซ้อน

2. การจัดการเกี่ยวกับการกำหนดคุณสมบัติอื่นๆ : การพัฒนาอาจเป็นไปในลักษณะที่ให้ผู้ใช้กำหนดค่าต่างๆในการใช้งานตามความต้องการของกลุ่มผู้ใช้ได้ เช่น ค่าของเวลาในการรีเฟรช (Refresh) หน้าจอหลักส่วนกลาง, ขนาดของคิว (Queue) ที่ใช้เก็บรายละเอียดเกี่ยวกับตัวผู้ใช้, จำนวนสมาชิกในกลุ่มที่จะเข้ามาทำงานอนุญาตให้มีมากที่สุดเท่าไร เพื่อป้องกันการดำเนินงานที่มีจำนวนคนมากเกินไปในงานบางอย่าง อาจเพื่อเป็นการลดจำนวนความขัดแย้ง เพื่อให้การทำงานเป็นไปอย่างราบรื่นและรวดเร็ว และอาจเพื่อป้องกันคนนอกกลุ่มที่อาจจะแอบแฝงมาดูข้อมูลของกลุ่ม เป็นต้น โดยก็ขึ้นอยู่กับแผนการพัฒนาว่าจะให้ผู้ใช้มีสิทธิ์ในการกำหนดเกี่ยวกับการจัดการเรื่องใดได้บ้าง แต่ไม่ควรให้ผู้ใช้มีอำนาจในการจัดการมากจนเกินไปเพราะอาจทำให้สูญเสียการควบคุมและอาจก่อให้เกิดความยุ่งยากในการควบคุมได้
3. การจัดการเกี่ยวกับความปลอดภัย (Security) : ตัวเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสารที่พัฒนามาแล้วนี้ป้องกันความปลอดภัยโดยอาศัยการใช้ ล็อกอิน (Login) และ รหัสผ่าน (Password) เพราะฉะนั้นแนวทางพัฒนาต่อไปในอนาคตนั้นอาจเป็นไปในลักษณะที่เข้มงวดเพิ่มมากขึ้นในการจัดการในเรื่องของความปลอดภัย เพื่อเป็นการป้องกันเอกสารในกรณีที่เอกสารนั้นเป็นเอกสารที่มีข้อมูลสำคัญที่ไม่อยากให้บุคคลอื่นล่วงรู้หรือเข้ามาละเมิดลิขสิทธิ์ โดยอาจจะจัดการเกี่ยวกับการกำหนดจำนวนสมาชิกในกลุ่มที่แน่นอนและอนุญาตให้จำนวนผู้ที่เข้ามาทำงานร่วมกันนั้นเป็นไปตามที่กำหนดไว้ ทั้งนี้เนื่องจากการป้องกันบุคคลนอกกลุ่มที่อาจจะผ่านกระบวนการรักษาความปลอดภัยระดับแรก (การ login) เข้ามาได้ โดยอาจจะมีการให้ป้อนข้อมูลผู้ใช้อีกครั้งเพื่อเป็นการยืนยันว่าเป็นบุคคลนั้นจริง ทั้งนี้อาจให้ผู้ใช้สามารถเลือกระดับความปลอดภัยได้ ในกรณีที่จัดทำเอกสารต่างๆไปได้ไม่ต้องผ่านขั้นตอนที่ยุ่งยากซับซ้อน และประหยัดทรัพยากรในการเก็บข้อมูลที่ไม่จำเป็นอีกด้วย แต่ทั้งหมดนี้ก็ขึ้นอยู่กับการออกแบบเพื่อให้สะดวกแก่ทั้งการควบคุมและมีความปลอดภัยเพียงพอสำหรับผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การจัดการเกี่ยวกับการพูดคุย (Chat)นอกเหนือจากหน้าจอหลักของการทำงาน : หากมีการพัฒนาให้มีการพูดคุย (Chat) ได้จะทำให้การทำงานสะดวกขึ้นและลดปัญหาความขัดแย้งลงได้ เพราะผู้ใช้งานมีการพูดคุยและสร้างความเข้าใจกันในการทำงานอาจจะเป็นส่วนหนึ่งของรายละเอียดการทำเอกสาร, รูปแบบการทำเอกสาร และอื่นๆ อีกทั้งยังให้ความรู้สึกใกล้เคียงกับการทำงานแบบพบปะมากกว่าการที่ไม่มีมีการพูดคุย โดยในการพัฒนาที่ผ่านมามีการแสดงรายชื่อของผู้ใช้ที่ใช้งาน (Online) อยู่ แนวทางพัฒนาก็อาจจะให้สามารถคลิกที่ชื่อผู้ใช้ที่ใช้งานอยู่และป๊อปอัพ (Pop up) เฟรมสำหรับการพูดคุยกับผู้ใช้ที่เลือกกว่าต้องการคุยด้วยในลักษณะที่คล้ายกับ ไอซีคิว (ICQ) แต่อาจจะมีการจัดการที่ยุ่งยากซับซ้อน หรือแนวทางพัฒนาง่ายๆก็อาจจะมีการป๊อปอัพ (Pop up) เฟรมสำหรับการพูดคุยรวมกันก็ได้ เป็นการลดความยุ่งยากแต่ก็ทำให้ผู้ใช้สามารถคุยกันได้เช่นกัน เป็นต้น
5. การพัฒนาให้ตัวเอเจนต์มีความสามารถมากขึ้น : การพัฒนาให้เฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสารมีความสามารถเป็นเอเจนต์ที่มีประสิทธิภาพเข้ามามีส่วนช่วยในการทำงานของผู้ใช้ และมีการปรับปรุงความสามารถในการเรียนรู้จากผู้ใช้ได้ เช่น การกำหนดเวลาที่จะรีเฟรช (Refresh) หน้าจอแสดงผลส่วนกลาง โดยหากสังเกตเห็นว่าผู้ใช้งานในกลุ่มทุกคนมีการกระทำ (Actions) ที่ซ้ำกันไปทำให้ต้องแสดงผลเดิมบ่อยครั้งก็จะเปลี่ยนค่าการรีเฟรชหน้าจอให้มากขึ้น เป็นต้น แต่แนวทางนี้ก็ยังเป็นแนวทางที่ค่อนข้างพัฒนาอยู่เพราะเป็นการทำให้เอเจนต์ฉลาด และมีการทำงานที่เป็นอัตโนมัติ ซึ่งเป็นเรื่องที่ยุ่งยากและซับซ้อน
6. การจัดการกับส่วนของการติดต่อกับผู้ใช้งาน (User interface) : การพัฒนาทำได้โดยพัฒนาให้ส่วนที่ติดต่อกับผู้ใช้เป็นการติดต่อที่เป็นมิตรกับผู้ใช้ (User friendly) นั่นคือการออกแบบและพัฒนาให้ง่ายต่อการใช้งาน โดยการพัฒนาที่ผ่านมาก็มีส่วนที่ติดต่อกับผู้ใช้ที่ไม่ซับซ้อนและใช้งานไม่ยาก หากแต่ถ้ามีการพัฒนาให้มีการใช้งานที่ง่ายขึ้น และหน้าจอการใช้งานที่สวยงามเพิ่มมากขึ้นแล้วก็จะทำให้ผู้ใช้ไม่รู้สึกรำคาญการใช้งานผ่านทางคอมพิวเตอร์ในการทำเอกสารงานร่วมกันนั้นเป็นเรื่องที่ไม่ยุ่งยากอีกต่อไป และมีความน่าสนใจในการใช้งานเพิ่มมากขึ้น เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ก.1 จาวา เว็บ เซิร์ฟเวอร์ 2.0 (JavaWebServer 2.0)

ก.1.1 บทนำ

เทคโนโลยีทางอินเทอร์เน็ตที่พัฒนาไปอย่างรวดเร็วคือก้าเน็คเซิร์ฟเล็ต (Servlet) ที่สามารถนำมาใช้แทนซีจีไอ (CGI-Common Gateway Interface) ได้ เนื่องจากเซิร์ฟเล็ต (Servlet) เป็น เซิร์ฟเวอร์ ไซด์ แอปพลิเคชัน (Server Side Application) ที่ทำงานบนฝั่งเซิร์ฟเวอร์ (Server) ดังนั้น เซิร์ฟเวอร์จึงเป็นสิ่งที่สำคัญที่ควรพิจารณาถึงเพื่อให้สามารถใช้งานเซิร์ฟเล็ต (Servlet) ได้อย่างมีประสิทธิภาพ

จาวา เว็บ เซิร์ฟเวอร์ 2.0 (Java Web Server 2.0) เป็นผลิตภัณฑ์(product)ที่ใช้เทคโนโลยีของเซิร์ฟเล็ตจึงมีประสิทธิภาพในการทำงานสูง สามารถเพิ่มความเร็วในการสื่อสารทางอินเทอร์เน็ตได้ โดยมีวิธีการที่ง่ายในการจัดการกับเว็บเพจ อีกทั้งจากการที่มีการรวมเอาความสามารถและความคล่องตัวของภาษาเขียนโปรแกรมภาษาจาวา(Java Programming Language)ไว้ ทำให้ Java Web server เป็นผลิตภัณฑ์ที่น่าสนใจตัวหนึ่งที่จะนำมาใช้ในการพัฒนาเว็บเพจ เพื่อรองรับความต้องการของการใช้อินเทอร์เน็ตที่เพิ่มมากขึ้นในปัจจุบัน

ก.1.2 ลักษณะเด่นและข้อดี (Key Feature) ของจาวา เว็บ เซิร์ฟเวอร์ (Java Web Server 2.0)

จาวา เว็บ เซิร์ฟเวอร์ 2.0 (Java Web Server 2.0) มีลักษณะเด่นและข้อดีต่างๆแสดงได้ดังต่อไปนี้

1. สนับสนุนเซิร์ฟเล็ต (Servlet Support) : จาวา เว็บ เซิร์ฟเวอร์ 2.0 (Java Web Server 2.0) สนับสนุนเทคโนโลยีของเซิร์ฟเล็ตที่มีความคงทน (Stable) และยืดหยุ่น (Flexible) มากกว่าซีจีไอ (CGI) ทำให้ผู้พัฒนาสามารถเชื่อมโยงเซิร์ฟเล็ตไปใช้กับแพลตฟอร์ม (Platform) ใดๆก็ได้ ทำให้สามารถพัฒนาความสามารถของการบริการ (Services) และแอปพลิเคชัน (Application) ต่างๆได้อย่างไม่จำกัด
2. เพจ คอมไพล์ชัน (Page Compilation) : ผู้พัฒนาเว็บสามารถเปลี่ยนแปลงเซิร์ฟเวอร์ (Server) ได้ โดยที่ไม่ต้องทำการรีคอมไพล์ (Recompile) โค้ดเซิร์ฟเล็ตใหม่ทำให้ประหยัดเวลาในการพัฒนา
3. ง่ายต่อการติดตั้ง(Installation) และง่ายต่อการดำเนินการ(Administration) : จาวา เว็บ เซิร์ฟเวอร์ 2.0 (Java Web Server 2.0) เป็นผลิตภัณฑ์ที่สามารถติดตั้งได้อย่างรวดเร็ว ง่ายต่อการดูแลรักษา และจัดการ มีการติดตั้งค่าคอนฟิกูเรชัน (Configuration) ต่างๆเพียงครั้งเดียว ไม่จำเป็นต้องติดตั้งค่าคอนฟิกูเรชัน (Configuration) ใหม่(Reconfigure)เมื่อเซิร์ฟเวอร์ล่ม (server down)
4. อยู่บนพื้นฐานความปลอดภัย : กล่าวคือ จาวา เว็บ เซิร์ฟเวอร์ 2.0 (Java Web Server 2.0) เป็นผลิตภัณฑ์ที่มีความปลอดภัยสูงนั่นเอง ไม่ว่าจะเป็นการใช้โลคอล เซิร์ฟเล็ต (Local Servlet) หรือรีโมท เซิร์ฟเล็ต (Remote Servlet) ก็สามารถมั่นใจได้ว่าจะได้รับข้อมูลที่ถูกต้องและเชื่อถือได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สามารถขยายเพิ่มเติมได้เป็นแพลตฟอร์ม (Platform) ที่เหมาะสมสำหรับธุรกิจต่างๆ : จาวา เว็บ เซิร์ฟเวอร์ 2.0 (Java Web Server 2.0) เป็นผลิตภัณฑ์ที่สามารถให้การปฏิบัติงาน(Performance) ต่างๆได้อย่างไม่จำกัด สามารถขยายและเพิ่มเติมความสามารถของเซิร์ฟเวอร์ (Server)ได้ตามความต้องการ

ก.1.3 สิ่งที่ต้องเตรียมก่อนการติดตั้ง (Prerequisites)

ฮาร์ดแวร์และระบบปฏิบัติการ (Hardware & Operating System)

- ขั้นต่ำ (Minimum):

Intel 486-based PC

32 MB RAM

Windows 95 OR Windows NT 4.0

- คำแนะนำ

Intel Pentium 166 (or higher)-based PC

64 MB RAM

Windows NT 4.0 with Service Pack 4

ในการติดตั้งต้องการเนื้อที่ร่วๆประมาณ 13 MB

โน้ต (Note) : สำหรับผู้ใช้งานบนระบบปฏิบัติการ วินโดว์ 95 (Windows 95) นั้นวินโดว์ 95 ไม่สนับสนุนเซิร์ฟเวอร์ แพลตฟอร์ม (Server Platform) ไมโครซอฟท์ (Microsoft) ไม่แนะนำให้ใช้วินโดว์ 95 (Windows 95) เป็นเซิร์ฟเวอร์

ก.1.4 ค่าดีฟอลต์ (Defaults) ของการติดตั้ง

- ไดรเรกทอรี : จะเป็น

C:\JavaWebServer2.0 (ตัวอย่างในวินโดว์(Windows))

โดยส่วนของไดเรกทอรี (Directory Path) ที่ติดตั้งจาวา เว็บ เซิร์ฟเวอร์ 2.0 (Java Web Server 2.0) จะถูกอ้างถึงเป็นเซิร์ฟเวอร์ รุท (server_root)

โน้ต (Note) : ไม่ควรติดตั้งลงในส่วนของไดเรกทอรี (Directory Path) ที่กำหนดเองไว้ก่อนเพราะอาจเกิดปัญหาเกี่ยวกับตัวแปรทางด้านสิ่งแวดล้อม (Environment variables) ที่ถูกใช้โดยจาวา เว็บ เซิร์ฟเวอร์ (Java Web Server) ได้ เช่น ติดตั้งลงใน

C:\Program Files\ เป็นต้น

- พอร์ต (Port) : เป็นดังนี้

Web Service 8080

Secure Service 7070

โดยค่า พอร์ต (Port) นี้สามารถเปลี่ยนแปลงได้ภายหลังการติดตั้งโดยใช้จาวา เว็บเซิร์ฟเวอร์ แอดมินิสเตรชัน ทูล (Java Web Server Administration Tool) โดย จาวา เว็บเซิร์ฟเวอร์ แอดมินิสเตรชัน ทูล (Java Web Server Administration Tool) สามารถเข้าถึงได้โดยใช้พอร์ต (Port) 9090

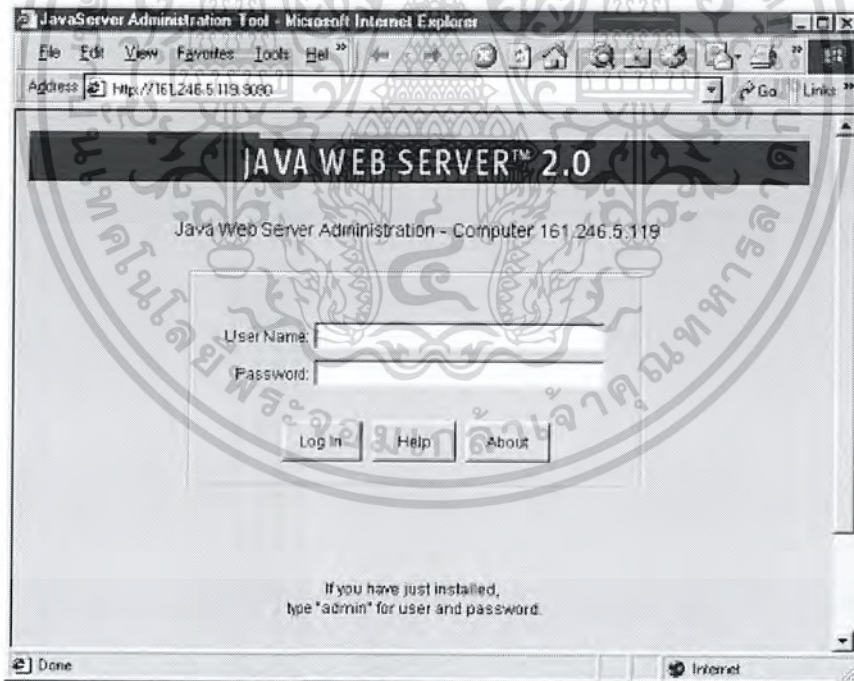
ก.1.5 การเซ็ทคลาส พาทซ์ (SET CLASSPATH)

การเซ็ทคลาส พาทซ์ (SET CLASSPATH) ใน ออโตอีเอ็กซีคิวทีฟ (AUTOEXEC.BAT) เพื่อให้สามารถใช้งาน จาวาเว็บเซิร์ฟเวอร์ ได้นั้นสามารถทำได้ดังนี้

```
SET CLASSPATH= c:\JavaWebServer2.0\lib\servlet.jar; c:\JavaWebServer2.0\lib\ssl.jar;
c:\JavaWebServer2.0\servlets\classes12.zip;
```

ก.1.6 ตัวอย่างการเซ็ทค่าคอนฟิกูเรชัน (Configurations) โดยใช้ จาวา เว็บเซิร์ฟเวอร์ แอดมินิสเตรชัน ทูล (Java Web Server Administration Tool)

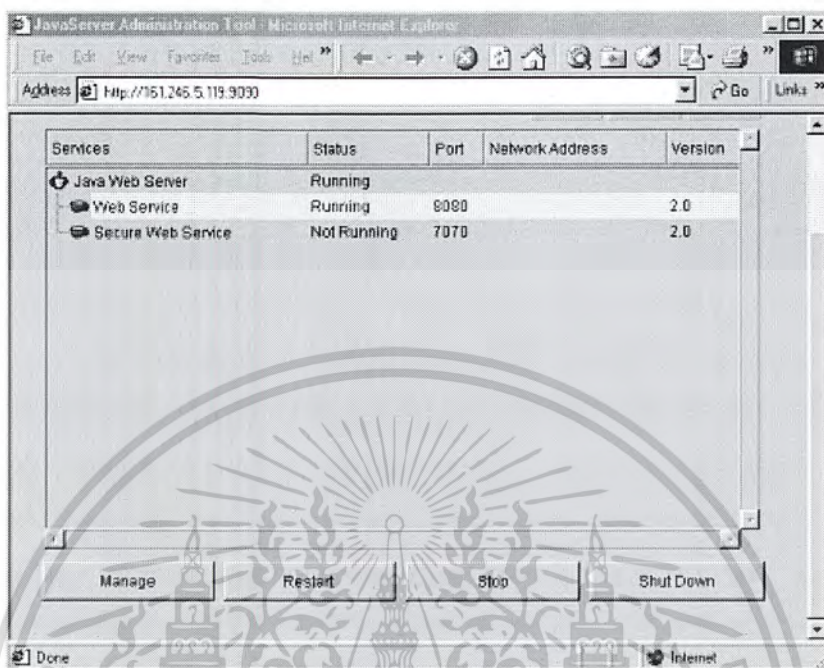
- การเข้าถึงจาวาเว็บเซิร์ฟเวอร์ แอดมินิสเตรชัน ทูล (Java Web Server Administration Tool) ผ่านทางพอร์ต (Port) 9090 (จากรูป ก.1 หมายเลข 161.246.5.119 คือ ไอพี แอดเดรส (IP Address) ของเซิร์ฟเวอร์ หรือจะใช้เป็นชื่อของเซิร์ฟเวอร์ก็ได้



รูปที่ ก.1 แสดงหน้าจอ ล็อกอิน (Login)

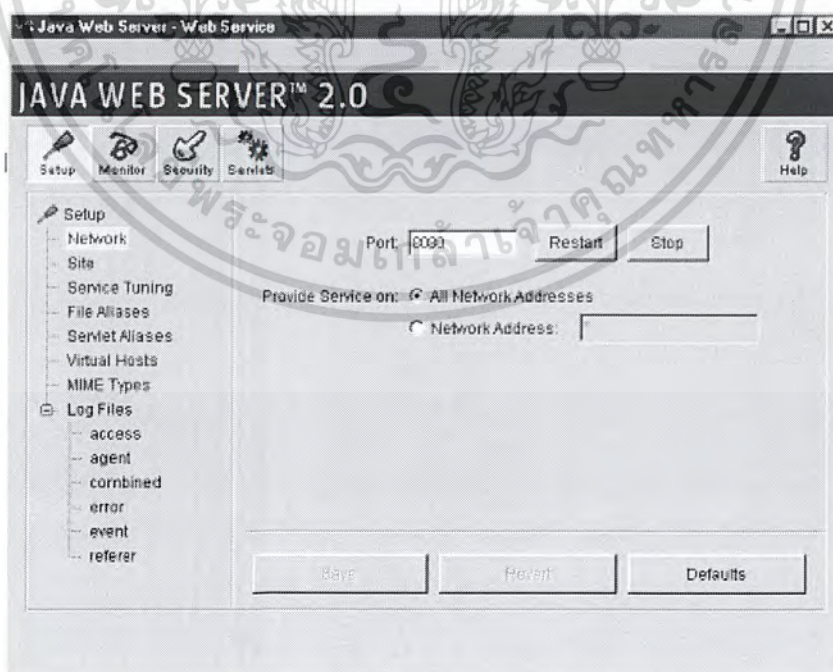
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อทำการลือคอิน (Login) โดยถ้าเพิ่งติดตั้งให้ ลือคอิน (Login) และรหัสผ่าน (Password) เป็น “admin” จะได้น้ำจอคังรูป ก.2



รูป ก.2 แสดงน้ำจอหลังการ ลือคอิน (Login)

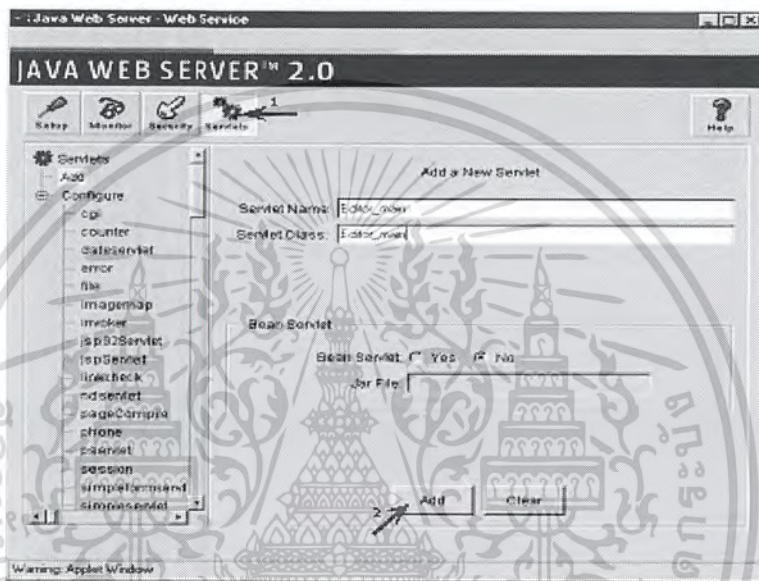
- จากรูป ก.2 กดที่ปุ่ม “Manage” จะได้น้ำจอแสดงคังรูป ก.3 เลือคที่ “Network” เพื่อติดตั้งค่า



รูป ก.3 แสดงการติดตั้งค่าเน็ตเวิร์ค (Network Setting)

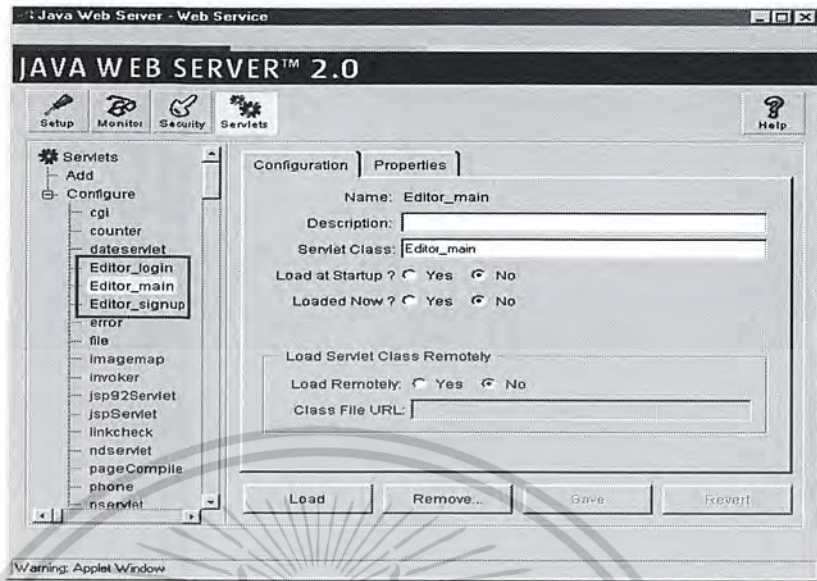
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากรูป ก.3 แสดงตัวอย่างการติดตั้งค่าต่างๆ โดยหากต้องการจะติดตั้งหรือเปลี่ยนแปลงค่าใดๆก็คลิกเลือกไปยังส่วนนั้น และติดตั้งค่าตามต้องการไม่ว่าจะเป็น “Site”, “Service Tuning”, “File Aliases”, หรือส่วนอื่นๆที่แสดงอยู่ในรูปก็สามารถทำได้
- จากตัวอย่างข้างต้นเป็นการติดตั้งในส่วนของ “Setup” โดยในส่วนของ “Monitor”, “Security”, และ “Servlets” ก็สามารถติดตั้งและแก้ไขเปลี่ยนแปลงค่าต่างๆได้ในลักษณะเดียวกัน โดยส่วนที่น่าสนใจ คือ การเพิ่ม (Add) เซิร์ฟเล็ตที่เขียนไปไว้ยังเซิร์ฟเวอร์เพื่อความสะดวกในการเรียกใช้งาน สามารถทำได้แสดงดังรูป ก.4 และ ก.5



รูป ก.4 แสดงการเพิ่มเซิร์ฟเล็ตไปยังเซิร์ฟเวอร์ โดยเลือกที่เซิร์ฟเล็ต (Servlet) และเลือกเพิ่ม (Add) จากนั้นพิมพ์ชื่อเซิร์ฟเล็ตและคลาสที่ต้องการเพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ก.5 แสดงเซิร์ฟเล็ต (Servlet) ที่เพิ่ม (Add) ลงบนเซิร์ฟเวอร์เรียบร้อยแล้ว

ก.2 จาวา เซิร์ฟเล็ต ดีเวลอปเมนต์ คิท (Java servlet Development Kit(JSDK))

จาวา เซิร์ฟเล็ต ดีเวลอปเมนต์ คิท (Java servlet Development Kit(JSDK)) เป็นเซิร์ฟเล็ต เอพีไอ (Servlet API) ที่ประกอบไปด้วยแพ็คเกจ (Package) ต่างๆรวมทั้ง javax.servlet และ javax.servlet.http ซึ่งบรรจุคลาส (Classes) และอินเทอร์เฟซ (Interface) ต่างๆที่จำเป็นต้องใช้ในการสร้างเซิร์ฟเล็ต โดยหลักการง่าย ๆ ในการสร้างเซิร์ฟเล็ต ก็คือการสร้างคลาสขึ้นมาคลาสหนึ่งโดยคลาสนั้นจะต้องทำการอิมพลีเมนต์ (Implement) ตัวอินเทอร์เฟซ (Interface) ที่ชื่อ javax.servlet.Servlet เพียงเท่านั้นก็จะได้เซิร์ฟเล็ตที่เป็นของตัวเองแล้ว โดยในการเขียนเซิร์ฟเล็ตอาจจะทำการอิมพลีเมนต์ (Implement) ตัว javax.servlet.Servlet interface ไม่ทางตรงก็ทางอ้อม เพื่อที่จะอำนวยความสะดวกให้กับนักพัฒนา ทางจาวาจึงได้มีการสร้างคลาสพื้นฐานที่ได้ทำการอิมพลีเมนต์ (Implement) ตัว javax.servlet.Servlet interface ขึ้นมาสองคลาสคือคลาส javax.servlet.GenericServlet และคลาส javax.servlet.http.HttpServlet (ซึ่งเป็นคลาสย่อย (Subclass) ของ javax.servlet.GenericServlet อีกทีหนึ่ง) ดังนั้นสิ่งที่นักพัฒนาจะต้องทำก็คือการนำเอาคลาสย่อย (Subclass) คลาสใดคลาสหนึ่งในสองคลาสนี้แล้วทำการโอเวอร์ไรด์ (Override) ฟังก์ชันที่ต้องการซึ่งโดยทั่วไปก็คือฟังก์ชันที่ชื่อ service() นั่นเอง นอกจากนี้ เจเอสดีเค (JSDK) ยังประกอบไปด้วยแพ็คเกจ sun.servlet ซึ่งเป็นแพ็คเกจที่ภายในประกอบไปด้วยคลาสที่เว็บเซิร์ฟเวอร์ (Web Server) จำเป็นต้องใช้ในการรันเซิร์ฟเล็ต

ตัวอย่างเมธอด(Method)ที่สำคัญที่ใช้ในการเขียน เซิร์ฟเล็ต แสดงได้ดังต่อไปนี้

- 1) เมธอดอินิต (init()) : เป็นเมธอดที่ใช้ในการติดตั้งค่าเริ่มต้นของการร้องขอ เมธอดนี้จะถูกใช้งานเมื่อมีการเรียกใช้งานเซิร์ฟเล็ต จากเซิร์ฟเวอร์ โดยจะมีการเรียกใช้งานครั้งแรกเพียงครั้งเดียวตลอดการทำงาน ของเซิร์ฟเล็ต
- 2) เมธอดเซอร์วิต (service()) : เมธอดนี้จะถูกเรียกใช้ในแต่ละครั้งที่ เซิร์ฟเล็ต ถูกเรียกหรือปลุก(Invoke)ขึ้นมาทำงานเป็นเมธอดที่ไว้รับและตอบสนองการร้องขอจาก Client
- 3) เมธอดเดสทรอย (destroy()) : เป็นเมธอดที่ใช้ในการกำจัด(Cleanup) เซิร์ฟเล็ต ภายหลังจากการถูกเรียกใช้งานเสร็จสิ้นแล้ว
- 4) เมธอดดูเก็ท (doGet ()) : เป็นเมธอดที่ใช้ในเซิร์ฟเล็ต อีกเมธอดหนึ่งที่มีความสำคัญ โดยจะมีการใช้งานเมื่อมีการเสนอเคิลการร้องขอแบบเก็ท (Handles GET Request) และจะเน้นการทำงานในด้านของการที่ไคลเอนต์รับข้อมูลจากเซิร์ฟเวอร์

ตัวอย่างการใช้งาน

```
public void doGet(HttpServletRequest req, HttpServletResponse res){}
```

- 5) เมธอดดูโพสต์ (doPost()) : เป็นเมธอดที่มีลักษณะการทำงานที่ตรงกันข้ามกับดูเก็ท (doGet) โดยจะเน้นการทำงานในส่วนของการส่งข้อมูลจากไคลเอนต์ไปยังเซิร์ฟเวอร์

ตัวอย่างการใช้งาน

```
public void doPost(HttpServletRequest req, HttpServletResponse res){}
```

ในการทำงานนั้นเซิร์ฟเล็ตจะผ่านค่าการร้องขอของเซิร์ฟเล็ต (ServletRequest) และ การตอบกลับของเซิร์ฟเล็ต (ServletResponse) เพื่อใช้ในการแสดงการร้องขอและการตอบสนองการร้องขอ โดย HTTPServlet จะเป็นคลาสที่อยู่ใน javax.servlet.http และ ตัวHTTPServlet จะให้การอิมพลิเม้นเทชัน (Implementation) แก่เมธอดเซอร์วิต (service())ในการจัดการกับการร้องขอแบบเก็ท (GET) และ โพสต์ (POST) และจากตัวอย่างการใช้งานจะเห็นว่าการส่งค่าพารามิเตอร์(Parameter)เพื่อให้เป็นค่าของ HttpServletRequestและ HttpServletResponse โดย

- HttpServletRequest : เป็นอินเทอร์เฟซ (Interface) ซึ่งให้เมธอดสำหรับการรับเอาค่า(get)ข้อมูลที่ได้จากคำร้องขอของไคลเอนต์ (Client) ยกตัวอย่างเช่น
 - เมธอดเก็ทพารามิเตอร์ (getParameter()) : จะคืน(return)ค่าของคีย์ (Key) หรือค่า(Value)ในคำร้องขอของไคลเอนต์ (Client)
- HttpServletResponse : ให้ค่าเอาท์พุท (Output stream) ซึ่งเซิร์ฟเล็ตสามารถเขียนเป็นเอชทีเอ็มแอล (HTML) กลับไปแสดงผลยังฝั่งไคลเอนต์ (Client) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.3 แพ็กเกจ (Package) อื่นๆ

```
package com.oreilly.servlet.MultipartRequest;
โดยรายละเอียดของแพ็กเกจ (Package) มีดังต่อไปนี้
// Copyright (C) 1998 by Jason Hunter <jhunter@acm.org>. All rights reserved.
// Use of this class is limited. Please see the LICENSE for more information.
```

```
package com.oreilly.servlet;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
import java.util.Hashtable;
```

```
import java.util.Vector;
```

```
import java.util.Enumeration;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import com.oreilly.servlet.multipart.MultipartParser;
```

```
import com.oreilly.servlet.multipart.Part;
```

```
import com.oreilly.servlet.multipart.FilePart;
```

```
import com.oreilly.servlet.multipart.ParamPart;
```

```
/**
```

- * A utility class to handle `<code>multipart/form-data</code>`
- * requests, the kind of requests that support file uploads. This class emulates the
- * interface of `<code>HttpServletRequest</code>`
- * , making it familiar to use.
- * It uses a "push" model where any incoming files are read and saved directly
- * to disk in the constructor. If you wish to have more flexibility, e.g.
- * write the files to a database, use the "pull" model
- * `<code>MultipartParser</code>`
- * instead.
- * `<p>`
- * This class can receive arbitrarily large files (up to an artificial limit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* you can set), and fairly efficiently too.

* It cannot handle nested data (multipart content within multipart content)

* or internationalized content (such as non Latin-1 filenames).

* <p>

* See the included upload.war

* for an example of how to use this class.

* <p>

* The full file upload specification is contained in experimental RFC 1867,

* available at

* http://www.ietf.org/rfc/rfc1867.txt.

*

* @see MultipartParser

*

* @author Jason Hunter

* @author Geoff Soutter

* @version 1.6, 00/07/21, redid internals to use MultipartParser,

* thanks to Geoff Soutter

* @version 1.5, 00/02/04, added auto MacBinary decoding for IE on Mac

* @version 1.4, 00/01/05, added getParameterValues(),

* WebSphere 2.x getContentType() workaround,

* stopped writing empty "unknown" file

* @version 1.3, 99/12/28, IE4 on Win98 lastIndexOf("boundary=") workaround

* @version 1.2, 99/12/20, IE4 on Mac readNextPart() workaround

* @version 1.1, 99/01/15, JSDK readLine() bug workaround

* @version 1.0, 98/09/18

*/

```
public class MultipartRequest {
```

```
    private static final int DEFAULT_MAX_POST_SIZE = 1024 * 1024; // 1 Meg
```

```
    private File dir;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private Hashtable parameters = new Hashtable(); // name - Vector of values
private Hashtable files = new Hashtable(); // name - UploadedFile

private MultipartParser parser;

/**
 * Constructs a new MultipartRequest to handle the specified request,
 * saving any uploaded files to the given directory, and limiting the
 * upload size to 1 Megabyte. If the content is too large, an
 * IOException is thrown. This constructor actually parses the
 * <tt>multipart/form-data</tt> and throws an IOException if there's any
 * problem reading or parsing the request.
 *
 * @param request the servlet request.
 * @param saveDirectory the directory in which to save any uploaded files.
 * @exception IOException if the uploaded content is larger than 1 Megabyte
 * or there's a problem reading or parsing the request.
 */
public MultipartRequest(HttpServletRequest request,
                        String saveDirectory) throws IOException {
    this(request, saveDirectory, DEFAULT_MAX_POST_SIZE);
}

/**
 * Constructs a new MultipartRequest to handle the specified request,
 * saving any uploaded files to the given directory, and limiting the
 * upload size to the specified length. If the content is too large, an
 * IOException is thrown. This constructor actually parses the
 * <tt>multipart/form-data</tt> and throws an IOException if there's any
 * problem reading or parsing the request.

```

```

*
* @param request the servlet request.
* @param saveDirectory the directory in which to save any uploaded files.
* @param maxPostSize the maximum size of the POST content.
* @exception IOException if the uploaded content is larger than
* <tt>maxPostSize</tt> or there's a problem reading or parsing the request.
*/

```

```

public MultipartRequest(HttpServletRequest request,
    String saveDirectory,
    int maxPostSize) throws IOException {
    // Sanity check values
    if (request == null)
        throw new IllegalArgumentException("request cannot be null");
    if (saveDirectory == null)
        throw new IllegalArgumentException("saveDirectory cannot be null");
    if (maxPostSize <= 0) {
        throw new IllegalArgumentException("maxPostSize must be positive");
    }

    // Save the dir
    dir = new File(saveDirectory);

    // Check saveDirectory is truly a directory
    if (!dir.isDirectory())
        throw new IllegalArgumentException("Not a directory: " + saveDirectory);

    // Check saveDirectory is writable
    if (!dir.canWrite())
        throw new IllegalArgumentException("Not writable: " + saveDirectory);

    // Parse the incoming multipart, storing files in the dir provided,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// and populate the meta objects which describe what we found
parser = new MultipartParser(request, maxPostSize);

Part part;
while ((part = parser.readNextPart()) != null) {
    String name = part.getName();
    if (part.isParam()) {
        // It's a parameter part, add it to the vector of values
        ParamPart paramPart = (ParamPart) part;
        String value = paramPart.getStringValue();
        //out.println("param; name="+ name + ", value=" + value ;
        Vector existingValues = (Vector)parameters.get(name);
        if (existingValues == null) {
            existingValues = new Vector();
            parameters.put(name, existingValues);
        }
        existingValues.addElement(value);
    }
    else if (part.isFile()) {
        // It's a file part
        FilePart filePart = (FilePart) part;
        String fileName = filePart.getFileName();
        if (fileName != null) {
            // The part actually contained a file
            filePart.writeTo(dir);
            files.put(name, new UploadedFile(
                dir.toString(), fileName, filePart.getContentType()));
        }
    }
    else {
        // The field did not contain a file
        files.put(name, new UploadedFile(null, null, null));
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    }
    }
}

/**
 * Constructor with an old signature, kept for backward compatibility.
 * Without this constructor, a servlet compiled against a previous version
 * of this class (pre 1.4) would have to be recompiled to link with this
 * version. This constructor supports the linking via the old signature.
 * Callers must simply be careful to pass in an HttpServletRequest.
 *
 */
public MultipartRequest(ServletRequest request,
    String saveDirectory) throws IOException {
    this((HttpServletRequest)request, saveDirectory);
}

/**
 * Constructor with an old signature, kept for backward compatibility.
 * Without this constructor, a servlet compiled against a previous version
 * of this class (pre 1.4) would have to be recompiled to link with this
 * version. This constructor supports the linking via the old signature.
 * Callers must simply be careful to pass in an HttpServletRequest.
 *
 */
public MultipartRequest(ServletRequest request,
    String saveDirectory,
    int maxPostSize) throws IOException {
    this((HttpServletRequest)request, saveDirectory, maxPostSize);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/**
 * Returns the names of all the parameters as an Enumeration of
 * Strings. It returns an empty Enumeration if there are no parameters.
 *
 * @return the names of all the parameters as an Enumeration of Strings.
 */
public Enumeration getParameterNames() {
    return parameters.keys();
}

/**
 * Returns the names of all the uploaded files as an Enumeration of
 * Strings. It returns an empty Enumeration if there are no uploaded
 * files. Each file name is the name specified by the form, not by
 * the user.
 *
 * @return the names of all the uploaded files as an Enumeration of Strings.
 */
public Enumeration getFileNames() {
    return files.keys();
}

/**
 * Returns the value of the named parameter as a String, or null if
 * the parameter was not sent or was sent without a value. The value
 * is guaranteed to be in its normal, decoded form. If the parameter
 * has multiple values, only the last one is returned (for backward
 * compatibility). For parameters with multiple values, it's possible
 * the last "value" may be null.
 *

```

```

* @param name the parameter name.
* @return the parameter value.
*/
public String getParameter(String name) {
    try {
        Vector values = (Vector)parameters.get(name);
        if (values == null || values.size() == 0) {
            return null;
        }
        String value = (String)values.elementAt(values.size() - 1);
        return value;
    }
    catch (Exception e) {
        return null;
    }
}

/**
* Returns the values of the named parameter as a String array, or null if
* the parameter was not sent. The array has one entry for each parameter
* field sent. If any field was sent without a value that entry is stored
* in the array as a null. The values are guaranteed to be in their
* normal, decoded form. A single value is returned as a one-element array.
*
* @param name the parameter name.
* @return the parameter values.
*/
public String[] getParameterValues(String name) {
    try {
        Vector values = (Vector)parameters.get(name);
        if (values == null || values.size() == 0) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return null;
    }

    String[] valuesArray = new String[values.size()];
    values.copyInto(valuesArray);
    return valuesArray;
}

catch (Exception e) {
    return null;
}
}

/**
 * Returns the filesystem name of the specified file, or null if the
 * file was not included in the upload. A filesystem name is the name
 * specified by the user. It is also the name under which the file is
 * actually saved.
 *
 * @param name the file name.
 * @return the filesystem name of the file.
 */
public String getFileName(String name) {
    try {
        UploadedFile file = (UploadedFile)files.get(name);
        return file.getFileName(); // may be null
    }
    catch (Exception e) {
        return null;
    }
}

/**

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

* Returns the content type of the specified file (as supplied by the
* client browser), or null if the file was not included in the upload.
*
* @param name the file name.
* @return the content type of the file.
*/

```

```

public String getContentType(String name) {
    try {
        UploadedFile file = (UploadedFile)files.get(name);
        return file.getContentType(); // may be null
    }
    catch (Exception e) {
        return null;
    }
}

```

```

/**
* Returns a File object for the specified file saved on the server's
* filesystem, or null if the file was not included in the upload.
*
* @param name the file name.
* @return a File object for the named file.
*/

```

```

public File getFile(String name) {
    try {
        UploadedFile file = (UploadedFile)files.get(name);
        return file.getFile(); // may be null
    }
    catch (Exception e) {
        return null;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

// A class to hold information about an uploaded file.
//
class UploadedFile {

    private String dir;
    private String filename;
    private String type;

    UploadedFile(String dir, String filename, String type) {
        this.dir = dir;
        this.filename = filename;
        this.type = type;
    }

    public String getContentType() {
        return type;
    }

    public String getFileSystemName() {
        return filename;
    }

    public File getFile() {
        if (dir == null || filename == null) {
            return null;
        }
        else {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
return new File(dir + File.separator + filename);  
}  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

หนังสืออ้างอิง

- [1] ดร.วีระศักดิ์ ชิงฉาวร : “*Fundamental of Java Programming Volume 1*”, SUM Publishing Department, SUM System Company Limited, 390 หน้า, 2541.
- [2] ดร.วีระศักดิ์ ชิงฉาวร : “*Fundamental of Java Programming Volume 1*”, Se-Education Public Company Limited, 319 หน้า, 2543.
- [3] กิตติ ภัคดีวัฒนกุล : “*Java ฉบับโปรแกรมเมอร์*”, บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด, 479 หน้า, 2542.
- [4] Jason Hunter with William Crawford : “*Java™ Servlet Programming*”, O’Reilly & Associates, Inc., 510 Pages, 1998.
- [5] Karl Moss : “*Java Servlets*”, McGraw-Hill Companies, 514 Pages, 1999.
- [6] นลองชัย จงประเสริฐพร, วรวิภา ท่าพระนา : “*CGI Web Programming*”, บริษัท จีรวัฒน์ เอ็กซ์เพรส จำกัด, 394 หน้า.
- [7] Mark Watson : “*Intelligent Java™ Applications for the Internet and Intranets*”, Morgan Kaufmann Publishers, 377 Pages, 1997.
- [8] Merlin Hughes, Michael Shoffner, Derek Hamner : “*Java Network Programming*”, Manning”, Publications Co., 807 Pages, 1999.
- [9] Michael J. Young : “*Step by Step XML ฉบับภาษาไทย*”, สำนักพิมพ์สามย่าน.com, 380 หน้า, 2000.
- [10] Joseph P. Bigus, Jennifer Bigus : “*Constructing intelligent Agent with Java™*”, Wiley Computer Publishing, 379 Pages, 1998.

เว็บไซต์อ้างอิง

- [1] “*Servlet*”, <http://www.servlet.com>.
- [2] “*Core Servlets and Java Server Pages™*”, <http://www.coreservlets.com>.
- [3] “*Core Servlets and Java Server Pages™*”, <http://www.archive.coreservlets.com>.
- [4] “*Java™ Servlet Technology*”, <http://java.sun.com/products/servlet>.
- [5] “*Java Web Server*”, <http://www.sun.com/software/jwebserver>.
- [6] “*A brief Introduction to Servlets and JSP*”,
<http://wdvl.internet.com/Authoring/Java/Servlets/intro.html>.
- [7] “*Tailorability and Cooperative Work*”, <http://www.ifi.uib.no/staff/anders/research>.
- [8] “*CSCW*”, <http://www.informatik.umu.se/~rwhit/CSCWGuide.html>.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [9] “CSCW”, <http://www.cti.dtu.dk/cscw/cscw.html>
- [10] “CSCW Directory”, <http://www.demon.co.uk/jrac/cscwdir.html>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้