

เฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร 2

A Co-Operative Framework For Document Composition II



นายวรานุสรณ์ สถาพรวงศ์
นาย อธิคม วงศ์วิโรพาร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เลขหม.....
เลขทะเบียน.....46165
วัน, เดือน, ปี.....20 ส.ค. 2546

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

เฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร 2

A Co-Operative Framework For Document Composition II



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร 2

A Co-Operative Framework For Document Composition II

ผู้จัดทำ

1. นายวรานุสรณ์ สถาพรวงศ์ รหัสประจำตัว 41014369
2. นายอติคม วงศ์วิโรพาร รหัสประจำตัว 41014505

อาจารย์ที่ปรึกษา

(ดร. วิศิษฐ์ หิรัญกิตติ)



เฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร 2

นาย วรานุสรณ์ สถาพรวงศ์ 41014369

นาย อธิคม วงศ์โรพาร 41014505

ดร.วิศิษฎ์ หิรัญกิตติ อาจารย์ที่ปรึกษา

ปีการศึกษา 2544

บทคัดย่อ

ระบบการประพันธ์เอกสารร่วมกันทางไกล คือระบบที่ผู้ใช้หลายๆบุคคล สามารถสร้าง แก้ไข เอกสารเดียวกันพร้อมกันจากสถานที่ต่างกันได้

ระบบนี้พัฒนาโดยใช้เทคโนโลยีทางเครือข่ายคอมพิวเตอร์ และ JAVA RMI และนำมาซึ่งปัญหาที่น่าสนใจในทาง Distributed Computing เช่น ปัญหาการแก้ไขข้อความเดียวกันโดยหลายผู้ใช้พร้อมกัน เป็นต้น

ปริญญานิพนธ์ฉบับนี้ ได้ศึกษาปัญหาที่พบในระบบการประพันธ์เอกสารร่วมกันทางไกล และนำเสนอการแก้ไข เพื่อเพิ่มประสิทธิภาพในการทำงานร่วมกันกับหลายผู้ใช้

Co-operative Framework for Document Composition II

Waranusorn Satapornwong

Atikom Wongvarolan

Dr.Visit Hirankitti Advisor

ABSTRACT

Co-operative Framework for Document Composition 2 is a network application working on internet which let people working together to achieve the same objective. This application allows co-operative teamworks to work together without the limit of places and time.

This thesis studies ,improves ,and developed from the thesis “Co-operative Framework for Document Composition” using advanced Java technology to improve, fix the problem found in the previous thesis and increase the performance of Co-operative editor.

สารบัญ

หน้าที่

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญตารางและรูปภาพ	VI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 หลักการและทฤษฎีการทำงานร่วมกัน (Theory of Cooperative Work)	3
2.1 บทนำ	3
2.2 การทำงานร่วมกัน (Cooperative Work)	3
2.3 การปรับปรุงการทำงาน (Tailoring)	4
2.4 คอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer Supported Cooperative Work หรือ CSCW)	4
2.4.1 ความหมายของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน	4
2.4.2 มิติของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer Supported Cooperative Work Dimensions หรือ CSCW Dimensions)	5
2.5 ตัวอย่างแนวความคิดและเทคโนโลยีเกี่ยวกับคอมพิวเตอร์ ที่สนับสนุนการทำงานร่วมกัน	5
2.6 สิ่งที่ต้องพิจารณาในการทำงานร่วมกัน	6
บทที่ 3 หลักการของเอเจนต์	7
3.1 บทนำ	7
3.2 คำนิยามเอเจนต์	7
3.3 การทำงานของเอเจนต์	8
3.4 แนวทางการสร้างเอเจนต์	9
3.5 การเรียนรู้ข้อมูลเอเจนต์	11
บทที่ 4 รีโมท เมธอด อินโวกชัน (REMOTE METHOD INVOCATION (RMI))	12
4.1 บทนำ	12

4.2 สถาปัตยกรรมของ RMI	12
บทที่ 5 XML (Extensible Markup Language)	14
5.1 บทนำ	14
5.2 ลักษณะของภาษา XML	14
5.3 องค์ประกอบย่อยของ XML	15
5.3.1 PI	15
5.3.2 DTD	15
5.3.3 Namespaces	16
5.3.4 XSL และ XSLT	16
5.3.5 Xpath	17
5.3.6 XQL	18
บทที่ 6 โครงสร้างและการทำงานของระบบ	19
6.1 การเลือกโปรแกรมประยุกต์และสิ่งแวดล้อมที่ใช้สร้าง โปรแกรมการทำงาน	19
6.2 ขั้นตอนการสร้าง โปรแกรมการทำงานร่วมกัน	19
6.3 โครงสร้างของโปรแกรมการทำงานร่วมกัน	19
6.4 อินเทอร์เฟซต่างๆที่ติดต่อกับผู้ใช้	21
6.5 โพรโตคอลที่ใช้ในการติดต่อกันระหว่างไคลเอนต์และเซิร์ฟเวอร์	21
6.6 การทำงานของโปรแกรมการทำงานร่วมกัน	22
6.6.1. การแสดงผลข้อความ	23
6.6.2 การเลือกบรรทัดเพื่อขอการทำงาน	24
6.6.3. การเซฟผลลัพธ์การทำงาน	25
บทที่ 7 การทดลองและวิเคราะห์ผลการทดลอง	27
7.1 บทนำ	27
7.2 การทดสอบและการใช้งานของ โปรแกรม	27
7.2.1 ส่วนประกอบของหน้าต่างการใช้งาน	27
7.3 ตัวอย่างการทำงานร่วมกันในการเลือกบรรทัด	28
7.4 ตัวอย่างการทำงานร่วมกันในการแสดงข้อความที่จอภาพ	30
7.5 ตัวอย่างการทำงานร่วมกันในการแสดงผลสถานะผู้ใช้	34
7.6 การเซฟผลการทำงาน	36
บทที่ 8 ปัญหาที่พบและแนวทางแก้ไข	38
8.1 สรุปการใช้โปรแกรม	38

8.2 ปัญหาที่พบ	38
8.3 แนวทางการปรับปรุงและพัฒนา	39
8.4 ข้อเสนอแนะ	39
บรรณานุกรม	41



สารบัญรูปภาพ

หน้าที่

สารบัญรูปภาพ	
รูปที่ 3.1 แสดงโมเดลการทำงานของเอเจนต์	8
รูปที่ 4.1 สถาปัตยกรรมของ RMI	12
รูปที่ 4.2 การสร้างรีโมทออบเจกต์ใน RMI	13
รูปที่ 5.1 ตัวอย่างไฟล์ XML	15
รูปที่ 5.2 ตัวอย่าง แสดงไฟล์XML	16
รูปที่ 5.3 ตัวอย่าง XSL สำหรับเอกสารXML	17
รูปที่ 5.4 ตัวอย่าง ไฟล์ HTML ที่ได้จาก XML และ XSL	17
รูปที่ 6.1 แสดงโครงสร้าง การทำงาน	20
รูปที่ 6.2 อินเทอร์เฟซติดต่อกับผู้ใช้	21
รูปที่ 6.3 แสดงการติดต่อกันระหว่างเซิร์ฟเวอร์กับไคลเอ็นต์	23
รูปที่ 6.4 แสดงขั้นตอนการแสดงผลข้อความ	24
รูปที่ 6.5 แสดงขั้นตอนการจัดการการคลิกข้อความในส่วนเซิร์ฟเวอร์	25
รูปที่ 6.6 แสดงขั้นตอนการแสดงผลข้อความ	26
รูปที่ 7.1 ตัวอย่างโปรแกรม	28
รูปที่ 7.2 เลือกบรรทัดที่ต้องการคลิก	28
รูปที่ 7.3 ขอล็อกบรรทัดที่ 1 ไม่สำเร็จ เพราะถูกล็อกอยู่โดยผู้ใช้อื่น	29
รูปที่ 7.4 การร้องขอล็อกสำเร็จ	29
รูปที่ 7.5 พิมพ์ข้อความลงในแท็กซ์ฟิลด์	30
รูปที่ 7.6 เลือกฟอนท์ให้ข้อความ	30
รูปที่ 7.7 เลือกรูปแบบให้ข้อความ	31
รูปที่ 7.8 เลือกสีให้ข้อความ	31
รูปที่ 7.9 เลือกขนาดให้ข้อความ	32
รูปที่ 7.10 เมื่อกดปุ่ม OK เพื่อแสดงผลที่จอภาพ	32
รูปที่ 7.11 ทำการยกเลิกบรรทัดที่ล็อกไว้	33
รูปที่ 7.12 ใส่ข้อความในบรรทัดที่ 6	33
รูปที่ 7.13 สถานะผู้ใช้ประกอบด้วย ข้อมูลผู้ใช้และข้อมูลการล็อก	34
รูปที่ 7.14 แสดงว่ามีผู้ใช้อยู่ 2 คน	35
รูปที่ 7.15 แสดงว่าผู้ใช้งานแรกล็อกบรรทัดที่ 1 อยู่ และเราล็อกบรรทัดที่ 6	35

รูปที่ 7.16 ทำการเซฟผลลัพธ์ที่ได้จากเอกสาร	36
รูปที่ 7.17 ใส่ชื่อไฟล์และนามสกุล xml ที่จะทำการเซฟ	36
รูปที่ 7.18 ผลที่ได้เมื่อแปลงจาก xml เป็น html	37



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในการทำงานต่างๆ ในชีวิตประจำวันนั้น เวลาและสถานที่ที่มีบทบาทสำคัญอย่างยิ่งที่เป็นข้อจำกัดในการทำกิจกรรมต่างๆ โดยเฉพาะในการทำงานร่วมกันเป็นทีมงานนั้นเป็นกิจกรรมที่ใช้เวลาสูงมาก เนื่องจากการเสียไปของเวลาในการเดินทางและการเตรียมงานเรื่องสถานที่ที่เป็นไปด้วยความยากลำบาก

โรงงานเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสารทำให้ผู้ใช้หลายๆคนสามารถทำงานร่วมกันได้บนเว็บ ทำให้สามารถกำจัดข้อจำกัดทางด้านเวลา คือประหยัดเวลาในการมาทำงานร่วมกัน อีกทั้งกำจัดข้อจำกัดทางด้านสถานที่ คือ ผู้ใช้อินเทอร์เน็ตที่อยู่ต่างสถานที่กันก็สามารถทำงานร่วมกันได้ กล่าวได้ว่าเป็นการนำเอาความก้าวหน้าของเทคโนโลยีทางคอมพิวเตอร์มาใช้ให้เกิดประโยชน์ที่สุด

เราได้สังเกตเห็นความสำคัญของโรงงานที่จะมีประโยชน์อย่างมากในอนาคต จึงได้คิดหาแนวทางแก้ไข ปรับปรุงและพัฒนาเพื่อประสิทธิภาพในการทำงาน โดยใช้ความสามารถทางเทคโนโลยีที่ปัจจุบันได้เจริญก้าวหน้ามาก มาใช้พัฒนาโรงงาน เช่น แอปเพล็ต (applet) เพื่อเพิ่มความสามารถในการติดต่อระหว่างแอปเพล็ตกับเซิร์ฟเล็ท รีโมท เมธอด อินโวลูชัน (remote method invocation) เพื่อเพิ่มความสามารถในการติดต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์ (Client/Server) เป็นต้น

แอปเพล็ตเหล่านี้ จะทำงานบนเว็บ โดยจะมีตัวแทน (Agent) ที่ช่วยในการจัดการ ด้านการทำงานเอกสารให้กับกลุ่มผู้ใช้งาน โดยผู้ใช้แต่ละคนจะมีส่วนร่วมในการสร้างเอกสารขึ้นมา ผู้ใช้แต่ละคนสามารถเพิ่มเติมและแก้ไขผลงาน ที่แสดงออกทางหน้าจอแสดงผลส่วนกลาง โดยต้องได้รับการรับรู้และยอมรับจากผู้ใช้อื่นๆ เพื่อที่จะพัฒนาโปรแกรมได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการทำงานของจาวา RMI (Remote Method Invocation) และการติดต่อระหว่างไคลเอนต์กับเซิร์ฟเวอร์โดยใช้ RMI
2. เพื่อศึกษาการทำงานของแอปเพล็ต (applet) และสามารถใช้อัปเพล็ตในการแสดงผลได้
3. เพื่อให้ผู้ใช้งานสามารถสร้างเอกสารร่วมกัน โดยผ่านเครือข่ายทางอินเทอร์เน็ต (Internet) ได้
4. เพื่อเพิ่มความสะดวกในการจัดทำเอกสารร่วมกันผ่านทางอินเทอร์เน็ต (Internet) โดยไม่ต้องเสียเวลาเดินทางเพื่อจะมาทำงานด้วยกัน อันจะเป็นการประหยัดเวลา และค่าใช้จ่ายที่จะต้องเสียไปกับการเดินทาง
5. เพื่อเป็นแนวทางในการพัฒนาอิดิเตอร์แก่ผู้ที่สนใจต่อไป

1.3 ขอบเขตของปริญญานิพนธ์

1. ศึกษาการทำงานของโครงงานเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร
2. รวบรวมข้อผิดพลาดและหาแนวทางแก้ไขเพื่อเพิ่มประสิทธิภาพในการทำงาน
3. ศึกษาและรวบรวมข้อมูลเกี่ยวกับการทำงานของเอเจนต์และภาษาการ โปรแกรมที่นำมาใช้ในการพัฒนาคือ ภาษาจาวา
4. ศึกษาและรวบรวมข้อมูลเกี่ยวกับการทำงานร่วมกันผ่านทางคอมพิวเตอร์ (Computer Support Cooperative Work) รวมทั้งการทำงานของระบบไคลเอนต์/เซิร์ฟเวอร์ (Client/Server)
5. ศึกษาและรวบรวมข้อมูลเกี่ยวกับ ไฮเพอร์เท็กซ์ ทรานสเฟอร์ โพรโตคอล (HyperText Transfer Protocol หรือ HTTP) ซึ่งเป็น โพรโตคอลที่ใช้ในการติดต่อสื่อสารผ่านเครือข่ายทางอินเทอร์เน็ต
6. ศึกษาและรวบรวมข้อมูลเกี่ยวกับการทำงานของ คอมมอน เกทเวย์ อินเตอร์เฟส (Common Gateway Interface หรือ CGI)
7. ศึกษาและรวบรวมข้อมูลเกี่ยวกับ รีโมท เมธอด อินโวกชัน (REMOTE METHOD INVOCATION)
8. ศึกษาและรวบรวมข้อมูลเกี่ยวกับเซิร์ฟเล็ตและการทำงานของเซิร์ฟเล็ตเพื่อนำมาประยุกต์ใช้ในการพัฒนาโครงงาน
9. ศึกษาและรวบรวมข้อมูลเกี่ยวกับแอปเพล็ตและการทำงานของแอปเพล็ตเพื่อการติดต่อกับเซิร์ฟเล็ต
10. ศึกษาและรวบรวมข้อมูลเกี่ยวกับ XML(Extensible Markup Language) เพื่อใช้ในการเก็บเอกสาร ได้อย่างมีประสิทธิภาพ
11. แก้ไขปรับปรุง โครงงานเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร

1.4 วิธีการดำเนินงาน

1. เข้าพบอาจารย์ที่ปรึกษาเพื่อรับฟังแนวความคิดและขอบเขตของการดำเนินงาน
2. ศึกษาหลักการการทำงานของ โครงงานเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร
3. ทดลองติดตั้ง โครงงานเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร
4. ศึกษาและทำความเข้าใจเกี่ยวกับระบบการทำงานแบบไคลเอนต์/เซิร์ฟเวอร์ (Client/Server), การเขียนโปรแกรมภาษาจาวา (Java Programming Language) , การทำงานของแอปเพล็ต , เซิร์ฟเล็ต ,การติดต่อระหว่างแอปเพล็ตและเซิร์ฟเล็ต
5. ทดลองเขียนแอปเพล็ตและเซิร์ฟเล็ตเพื่อทำความเข้าใจการติดต่อระหว่างแอปเพล็ตและเซิร์ฟเล็ต
6. ปรับปรุงโครงงานเฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสาร
7. รายงานความคืบหน้าให้อาจารย์ที่ปรึกษาทราบเป็นระยะและรับฟังคำแนะนำเพิ่มเติม
8. นำคำแนะนำและข้อคิดเห็นของอาจารย์ที่ปรึกษามาปรับปรุง แก้ไข จุดบกพร่องต่อไป
9. เตรียมเอกสารต่างๆที่ต้องใช้ เตรียมความพร้อมในการนำเสนอโครงงาน

บทที่ 2

หลักการและทฤษฎีการทำงานร่วมกัน (Theory of Cooperative Work)

2.1 บทนำ

คอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer Supported Co-operative Work (CSCW)) เป็นหัวข้องานวิจัยค้นคว้าใหม่ที่ใช้เทคโนโลยีสารสนเทศเพื่อทำให้เกิดกิจกรรมที่เกิดจากกลุ่มบุคคลเป็นไปได้โดยง่าย โดยเน้นไปทั้งรูปแบบธรรมชาติของการทำงานร่วมกันและทางที่เทคโนโลยีสารสนเทศสามารถเปลี่ยนแปลง,ขยาย,สนับสนุนรูปแบบของการทำงานร่วมกันghdfgdfgf

คอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกันเป็นการผสมผสานแนวคิดของวิศวกรรมซอฟต์แวร์ (Software Engineering) กับ การจัดการทำงานในแง่ของสังคมวิทยา (Sociology) และ คาบเกี่ยวกับสาขาต่างๆ เช่น ระบบจัดการข่าวสารข้อมูล (Management Information System) ,การพัฒนาระบบ (Systems Development) , ปฏิสัมพันธ์ระหว่างมนุษย์และคอมพิวเตอร์(Human-Computer Interaction (HCI)) และพฤติกรรมองค์กร(Organizational Behaviour) เริ่มมีการศึกษามาตั้งแต่ปี ค.ศ. 1993 จนถึงปัจจุบันได้มีการศึกษาและพัฒนาโดยลำดับ

2.2 การทำงานร่วมกัน (Coooperative Work)

การทำงานร่วมกัน โดยผ่านทางคอมพิวเตอร์นั้นเป็นการสร้างระบบข้อมูลข่าวสารที่ติดต่อกัน (Interactive Information System) ซึ่งเป็นการปรับให้เข้ากับยุคเทคโนโลยีข้อมูลข่าวสาร (information Technology) เพื่อตอบสนองความต้องการและเพิ่มความสะดวกให้กับผู้ใช้ในด้านการทำงานร่วมกัน โดยสิ่งที่สำคัญที่ควรระลึกถึงอยู่เสมอ คือ จะทำอะไรให้สามารถปรับปรุงให้เข้ากับความต้องการของผู้ใช้งานได้และง่ายต่อการใช้งาน โดยการพัฒนาจะมีส่วนเกี่ยวกับ การออกแบบการทำงานของตัวโปรแกรม (Design program) , การออกแบบหน้าจอติดต่อกับผู้ใช้งาน (User interface), การทำงานของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer support cooperative work), การทำงานของผู้ใช้งาน (User computing) , การจัดการต่างๆ เกี่ยวกับเว็บ (เช่น HTML, Dreamweaver ,Photoshop เป็นต้น) ,การจัดการฐานข้อมูล (Database), การจัดการเกี่ยวกับผู้ใช้งานหลายคน (Multiusers), การจัดการเกี่ยวกับซอฟต์แวร์ (ยกตัวอย่างเช่น การเลือกใช้ภาษาในการพัฒนา,กรรมวิธีและเทคนิคต่างๆ (Methods and Techniques) เป็นต้น) .และการจัดการหน้าต่างหลัก (ยกตัวอย่างเช่น การรีเฟรช(Refresh),การทำบัฟเฟอร์คู่ (Double buffer) เป็นต้น

2.3 การปรับปรุงการทำงาน (Tailoring)

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

ในการปรับปรุงการทำงานนั้นจะต้องมีการพิจารณาถึงความสัมพันธ์ต่างๆ ในการทำให้คอมพิวเตอร์แอปพลิเคชัน (Computer Application) มีประสิทธิภาพดีขึ้นและเหมาะสมกับสถานการณ์การทำงานของผู้ใช้ ลักษณะคล้ายกับในการพัฒนาระบบนั้นจะมีปัญหาเกิดขึ้นมากมายซึ่งไม่สามารถหาคำตอบได้ และบางครั้งจะไม่สามารถควบคุมปัจจัยต่างๆ ซึ่งสนับสนุนลักษณะเฉพาะของผู้ใช้แต่ละคนได้ แต่สามารถระบุการติดตามความสัมพันธ์ของการทำงานในอนาคตได้เพื่อหาว่าอะไรคือสาเหตุและการป้องกันเพื่อให้เกิดการปรับปรุงแอปพลิเคชันให้เป็นไปตามความต้องการและบรรลุผลสำเร็จ โดยจะต้องดูระดับการใช้งาน ความเหมาะสมและที่สำคัญเหนือสิ่งอื่นใดคือ 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, การสนับสนุนกิจกรรมต่างๆ ของผู้ใช้งาน รวมถึงเทคนิคต่างๆ ที่นำมาใช้ในการรองรับการเปลี่ยนแปลงและการแก้ปัญหา

2.4 คอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน (Computer Supported Cooperative Work หรือ CSCW)

2.4.1 ความหมายของคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน

คอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน(CSCW) เป็นการอ้างอิงความสนใจในการสนับสนุนการทำงานร่วมกันของผู้ใช้หลายๆคนกับระบบคอมพิวเตอร์และครอบคลุมถึงสิ่งใดๆที่กำกับคอมพิวเตอร์และสนับสนุนกิจกรรมต่างๆที่มีมากกว่า 1 คนมาเกี่ยวข้องทำงานร่วมกัน ซีเอสซีดับบิว(CSCW) ไม่ได้นิยามในเทอมเทคนิคที่นำมาประยุกต์ใช้เท่านั้นแต่มองในลักษณะของการวิจัยกลุ่มที่มีเป้าหมายในการออกแบบระบบแอปพลิเคชัน (Application System) ให้เหมาะสมกับการใช้งานร่วมกัน อาจกล่าวได้ว่าเป็นการสนับสนุนความ

ต้องการของการทำงานร่วมกัน (The support requirement of cooperative work) โดย ซีเอสซีดับบิว (CSCW) สามารถแก้ปัญหาการทำงานร่วมกันและสร้างความเข้าใจการทำงานให้กับผู้ใช้งานแต่ละคนได้ โดยเป้าหมายของ ซีเอสซีดับบิว (CSCW) คือ การค้นพบหนทางที่จะใช้เทคโนโลยีทางคอมพิวเตอร์ส่งเสริมกระบวนการทำงานเป็นกลุ่มร่วมกันที่สนับสนุนทั้งในมิติของเวลาและสถานที่ (Time and place dimensions)

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

มี 2 มิติ คือ

1. มิติทางด้านเวลา (Time Dimension) แบ่งเป็น

- ริงก์ไทม์ (Real time) : เกิดเมื่อการติดต่อ (Communication) เกิดขึ้นในเวลาเดียวกัน
- อะซิงโครนัส ไทม์ (Asynchronous time) : เกิดเมื่อการติดต่อเกิดในเวลาที่แตกต่างกัน

2. มิติทางด้านสถานที่ (Place Dimension) แบ่งเป็น

- ที่เดียวกัน (Same place): เมื่อการพบกันเกิดขึ้นเมื่อผู้ส่วนร่วมอยู่ ณ สถานที่เดียวกัน
- ต่างที่กัน (Different place): เมื่อการพบกันเกิดขึ้นเมื่อผู้มีส่วนร่วมอยู่ในทำเลที่ตั้งที่แตกต่างกันไป

2.5 ตัวอย่างแนวความคิดและเทคโนโลยีเกี่ยวกับคอมพิวเตอร์ที่สนับสนุนการทำงานร่วมกัน

1. วายซีวิส (Wysiwis) : เป็นการยึดมั่นในประโยคที่ว่า “อะไรที่คุณเห็นคือสิ่งที่ฉันเห็น” (What you see is what I see) ลักษณะคล้ายกับการที่คน 2 คน อยู่ที่บ้านของตนเองและดูรายการโทรทัศน์ได้เหมือนกันในเวลาเดียวกัน เทคโนโลยีได้ขยายแนวความคิดนี้และอนุญาตให้บุคคลปฏิบัติงานและติดต่อสื่อสารกันในสภาพแวดล้อมของวายซีวิส (Wysiwis environment)
2. การรวมส่วนประกอบที่แตกต่างกันไว้ด้วยกัน (Integration different tasks): เพื่อเป็นการลดกำแพงความแตกต่างระหว่างส่วนประกอบต่างๆของแอปพลิเคชันที่แตกต่างกันและทำให้อยู่ในลักษณะที่สามารถปรับปรุงเปลี่ยนแปลงได้ง่าย คล้ายกับการมีเครื่องมือชิ้นใหญ่เพียงชิ้นเดียวแต่สามารถยืดหยุ่นและปรับเปลี่ยนตามต้องการและความเหมาะสมได้ (One large flexible tool) มากกว่าที่มีเครื่องมือชิ้นเล็กๆหลายๆชิ้นแต่ทำงานเฉพาะด้านไม่สามารถปรับเปลี่ยนได้เลย
3. เลขานุการส่วนตัว (Personal secretary) : จะเป็นกระบวนการที่ทำตัวเสมือนเป็นเลขานุการคอยจัดการเรื่องต่างๆเช่น จะหยุดสิ่งที่ไม่จำเป็นและให้ทำสิ่งที่จำเป็นกว่าก่อนและช่วยเก็บข้อความ (Message) ต่างๆไว้ให้ตอนที่ไม่ว่างเป็นต้น
4. การจัดการเกี่ยวกับเวลา (Time management) : เป็นผลิตภัณฑ์กรุปแวร์ (Groupware Product) ที่ช่วยจัดการเกี่ยวกับปฏิทิน,การบันทึกประจำวัน, ตารางการทำงาน (Project schedules) ที่ช่วยจัดการบริหารเวลาในการทำงานให้มีประสิทธิภาพมากขึ้น

5. มัลติมีเดีย (Multimedia) : เกี่ยวกับเรื่องของกราฟิกส์(Graphics) และเสียงให้ความเป็นธรรมชาติในการติดต่อระหว่างกันของผู้ใช้งาน
6. โปรแกรมสำหรับผู้ใช้งาน (End user program) : จะสนใจเกี่ยวกับการทำให้ระบบมีความเป็นส่วนตัวของผู้ใช้มากขึ้นมีกรุปแวร์ ทูล (Groupware Tool) ที่สามารถยืดหยุ่นได้เพียงพอซึ่งเป็นมิตรกับผู้ใช้งาน (User friendly) ในการเปลี่ยนแปลงแก้ไขทูล (tool) ให้เหมาะสมกับความต้องการของผู้ใช้

2.6 สิ่งที่ต้องพิจารณาในการทำงานร่วมกัน

ในการทำงานร่วมกันผ่านทางคอมพิวเตอร์นั้นสิ่งที่ต้องพิจารณาถึงคือ

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. การทำงานเสมือนผู้ใช้งานเดี่ยว (Re-engineering single-user application) : เป็นการรีเ็นจิเนียริง (Re-engineering) ของกระบวนการการทำงานของแอปพลิเคชันที่มีการใช้งานของผู้ใช้เพียงคนเดียว (Non-collaborative applications) ไปเป็นการทำงานของแอปพลิเคชันที่มีผู้ใช้งานหลายคน (Multi-user collaborative applications) กล่าวคือ หมายความว่าในการทำงานของแอปพลิเคชันที่มีการทำงานร่วมกัน หากมีผู้ใช้งานเพียงคนเดียวก็จะสามารถทำงานได้เสมือนหนึ่งเป็นแอปพลิเคชันของการใช้งานของผู้ใช้เพียงคนเดียวเท่านั้นเอง

บทที่ 3

หลักการของเอเจนต์

3.1 บทนำ

อินฟอร์เมชันไฮเวย์ (Information highway) เป็นแนวทางใหม่ในการจัดการการทำงานและบริการของคอมพิวเตอร์ที่มีประสิทธิภาพ แต่เนื่องจากความซับซ้อนในกระบวนการนี้ทำให้ต้องการ การติดต่อกัน (Interaction) ระหว่างคอมพิวเตอร์ในรูปแบบใหม่ ซึ่งคอมพิวเตอร์จะต้องเป็นเหมือนผู้ช่วยของผู้ใช้ แนวความคิดในการใช้เอเจนต์มาช่วยในการติดต่อระหว่างผู้ใช้และคอมพิวเตอร์นั้นถูกคิดขึ้นโดยทศนะของ นิโคลัส นิโกรพอนต์ (Nicholas Negroponte) และ อัลัน เค (Alan Kay) มีบริษัทหลายแห่งยอมรับ และปรับเปลี่ยนมาใช้แนวคิดนี้ ซอฟต์แวร์เอเจนต์ประเภทต่างๆ เช่น เอเจนต์ที่ช่วยในการจัดการตารางการบริหารเวลา (Personal assistance about time scheduling agent), เอเจนต์ที่ช่วยในการจัดการการส่งเมล (Electronic mail handling agent), เอเจนต์ช่วยในการกรองข่าวสารอิเล็กทรอนิกส์ (Electric new filtering agent) และ เอเจนต์ในด้านบันเทิง (Selection of entertainment agent) ได้ถูกสร้างขึ้น แต่อย่างไรก็ตามเทคนิคที่ใช้สร้างเอเจนต์ที่มีอยู่ในปัจจุบันยังห่างไกลกับความต้องการในการสร้าง การติดต่อเสมือนมนุษย์ในขั้นสูง (High-level human-like interaction)

3.2 คำนิยามของเอเจนต์

เอเจนต์มีความหมายถึง บุคคลใดบุคคลหนึ่ง หรือ สิ่งใดสิ่งหนึ่ง ที่กระทำหน้าที่เป็นตัวแทนให้กับผู้ขอใช้บริการ เพื่อจัดการงานเฉพาะอย่างใดอย่างหนึ่งให้สำเร็จลุล่วงไปได้ด้วยดี

ในทางคอมพิวเตอร์ได้มีการพัฒนาโปรแกรมขึ้นเพื่อจัดการภาระต่างๆแทนผู้ใช้ให้สามารถทำงานได้อย่างสะดวก รวดเร็ว และลดความซับซ้อนที่เกิดขึ้นในงาน โปรแกรมที่ทำหน้าที่ช่วยเหลือ โปรแกรมหลักให้สามารถทำงานได้ดีนั้นถูกเรียกว่า โปรแกรมผู้ช่วย หรือ ซอฟต์แวร์เอเจนต์

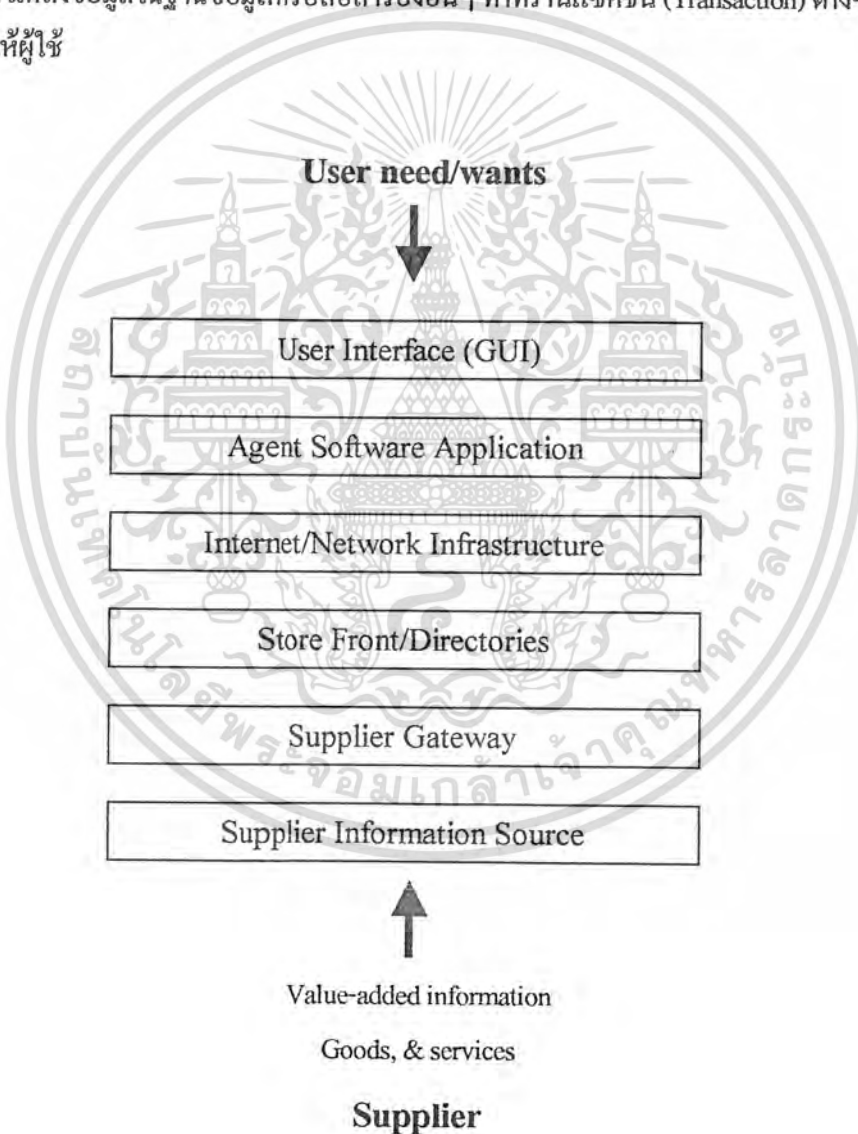
อาจจะกล่าวได้ว่าแอปพลิเคชันโปรแกรมเจนเนอเรชันที่ 4 เกือบทั้งหมดนั้นกระทำเป็นดังเช่นเอเจนต์ เพราะทุกวันนี้เราร้องขอการทำงานด้านคอมพิวเตอร์ผ่านทางซอฟต์แวร์ให้ทำงานต่างๆแทนเรา แต่หากมองให้ลึกแล้วจะทราบว่า ซอฟต์แวร์เอเจนต์ นั้นต่างจากแอปพลิเคชันอื่น โคนซอฟต์แวร์เอเจนต์จะต้องประกอบด้วย ลักษณะของการเปลี่ยนแปลงได้อัตโนมัติ , ความอิสระในการปกครองและควบคุมตนเอง และความสามารถในการติดต่อกับผู้ใช้ เมื่อเราเพิ่มความชาญฉลาดให้กับเอเจนต์ เอเจนต์จะได้รับความสามารถในการใช้เหตุผลที่ปรับเปลี่ยน ได้ตามความเหมาะสม ด้วยความสามารถนี้จะทำการ โพรเซสข้อมูลจากสิ่งแวดล้อมภายนอก เช่น เน็ตเวิร์ค ฐานข้อมูล หรือ ในอินเทอร์เน็ต สามารถทำได้

การเกิดขึ้นของเครือข่ายอินเทอร์เน็ต และ เครือข่ายเวิร์ลไวด์เว็บทำให้ความต้องการการใช้ซอฟต์แวร์เอเจนต์ชาญฉลาดเพิ่มสูงขึ้น การใช้งานเว็บเปลี่ยนจาก โมเดลแบบสแคเตอร์บราวซิง (Scattered Browsing)

ไปเป็น โมเดลการส่งข้อมูลแบบผ่านสื่อแบบพอยท์ทูพอยท์ (Point to Point) แนวโน้มนี้ช่วยเอเจนต์มีความเป็นธุรกิจทางการค้ามากขึ้น

3.3 การทำงานของเอเจนต์

โมเดลการทำงานกับเอเจนต์แสดงดังรูปที่ 2.1 ผู้ใช้ที่ต้องการจัดการงานต่างๆ ใช้กราฟิฟิคอินเทอร์เฟซที่ติดต่อกับซอฟต์แวร์แอปพลิเคชัน แอปพลิเคชันเหล่านี้จะทำหน้าที่เข้าไปหาข้อมูลในอินเทอร์เน็ตเพื่อไปถึงยังซัพพลายเออร์เกตเวย์ (Supplier gateway) ด้วยความช่วยเหลือของสตอร์ฟรอนต์ (Store front) ในที่สุดเอเจนต์จะสามารถเข้าถึง แหล่งข้อมูลในฐานข้อมูลหรือสื่อสำรองอื่นๆ ทำทรานแซกชัน (Transaction) ต่างๆ และ ส่งผลลัพธ์กลับไปให้ผู้ใช้งาน



รูปที่ 3.1 แสดงโมเดลการทำงานของเอเจนต์

- การสื่อสารระหว่างกันด้วยวิธีดั้งเดิม (Original Approach)

ใช้วิธีการติดต่อที่เรียกว่ารีโมทโพรซีเจอร์คอล (Remote Procedure Call (RPC)) โดย RPC จะมองการติดต่อระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ว่าเป็นการขอให้คอมพิวเตอร์เครื่องหนึ่งเรียกโพรซีเจอร์ของคอมพิวเตอร์อีกเครื่องหนึ่ง ข้อความที่มีการขนส่งในเน็ตเวิร์คจะเป็นการร้องขอหรือตอบรับการให้บริการ ข้อความร้องขอประกอบด้วยโพรซีเจอร์อาร์กิวเมนต์ (Procedure Argument) ข้อความตอบรับเป็นผลลัพธ์การทำงาน คอมพิวเตอร์ทั้งสองสื่อสารกันภายใต้โปรโตคอลเดียวกัน นิยมใช้ในระบบไคลเอ็นต์เซิร์ฟเวอร์ การทำงานแบบนี้จะเป็นแบบ 1 ผลลัพธ์ต่อ 1 คำสั่ง จึงทำให้ในเน็ตเวิร์คเต็มไปด้วยข้อความที่ส่งผ่านกันซึ่งอาจเกิดปัญหาข้อมูลผิดพลาดหรือสูญหาย ได้

- การสื่อสารระหว่างกันโดยใช้เอเจนต์ช่วย (Agent Enhanced Approach)

ใช้วิธีการติดต่อที่เรียกว่ารีโมทโปรแกรมมิ่ง (Remote Programming (RP)) โดย RP จะมองการติดต่อระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ว่าเป็นการที่คอมพิวเตอร์เครื่องหนึ่งส่งผ่านโพรซีเจอร์ที่จะทำงานไปให้ยังคอมพิวเตอร์อีกเครื่องหนึ่ง ข้อตกลงระหว่างคอมพิวเตอร์ทั้งสองอยู่ในรูปของภาษา กล่าวคือไคลเอ็นต์จะส่งเอเจนต์ไปยังเซิร์ฟเวอร์ เอเจนต์จะบรรจุโพรซีเจอร์ที่จะทำตามคำร้องขอของไคลเอ็นต์ ฉะนั้นไม่ว่าจะทำงานกี่ครั้งก็ต้องการการส่งข้อความที่เป็นเอเจนต์เพียงครั้งเดียว ข้อเด่นคือการติดต่อกันระหว่างสองเครื่องโดยที่ไม่ต้องผ่านเน็ตเวิร์คหากผ่านขั้นตอนของการขนส่งเอเจนต์แล้ว

ข้อได้เปรียบของ RP เปรียบเทียบกับ PRC

- ด้านประสิทธิภาพ กล่าวคือ ไคลเอ็นต์จะส่งเอเจนต์ไปยังเซิร์ฟเวอร์แทนที่จะส่งคำสั่งไปทำงาน โดยตรง การทำงานจะเป็นการทำงานภายในคอมพิวเตอร์ฝั่งเซิร์ฟเวอร์ แทนที่การทำงานระยะไกล ในเน็ตเวิร์คจึงมีความหนาแน่นของข้อมูลน้อยกว่า
- ด้านความยืดหยุ่น กล่าวคือ การใช้เอเจนต์ทำให้ผู้ผลิตไคลเอ็นต์ซอฟต์แวร์สามารถเพิ่มเติมฟังก์ชันจากที่มีอยู่ในเซิร์ฟเวอร์ซึ่ง RPC ทำไม่ได้เพราะเป็นเพียงการส่งคำสั่ง

3.4 แนวทางการสร้างเอเจนต์

คอมพิวเตอร์เข้ามามีบทบาทสำคัญในการช่วยลดเวลาในการทำงาน เพิ่มปริมาณกิจกรรมประจำวันให้สูงขึ้น ในขณะที่ยังมีผู้ใช้ที่ยังไม่รับการฝึกฝนในเรื่องการติดต่อกับคอมพิวเตอร์อยู่เป็นจำนวนมาก การติดต่อแบบไดเรกต์มานิปูเลชัน (Direct manipulation) ที่ให้ผู้ใช้เป็นผู้ริเริ่มสั่งงานและเป็นผู้คอยตรวจสอบเหตุการณ์ที่เกิดขึ้นนั้น จำเป็นที่จะต้องเปลี่ยนไป การติดต่อมีลักษณะแบบอินไดเรกต์มานเนทเมนต์ (Indirect management)

ได้ถูกนำมาใช้แทน มีลักษณะคือจะเป็นการทำกระบวนการร่วมกันของผู้ใช้และเอเจนต์ มีการสื่อสารกันในการทำงานและตรวจสอบเหตุการณ์ต่างๆ ระหว่างกันและกัน โดยที่เอเจนต์เป็นเหมือนผู้ช่วยส่วนบุคคลที่ร่วมทำงานไปด้วยกับผู้ใช้ และผู้ช่วยนั้นจะคอยเพิ่มประสิทธิภาพของตน โดยเรียนรู้จากความชอบ ความสนใจและสิ่งที่ทำจนเคยชินของผู้ใช้ และเรียนรู้จากเอเจนต์อื่นๆ ในชุมชนนั้น

แนวทางในการสร้างเอนต์สามารถทำได้หลายแนวทาง ดังนี้

แนวทางในการสร้างเอนต์แรกเรียกว่าเซมิออโตโนมัส (Semi autonomous) จะทำการรวบรวมกฎเกณฑ์ต่างๆ ของผู้ใช้ (User programmed rules) ที่ใช้ในการโพรเซสข้อมูลที่เกี่ยวข้องกับงานนั้นไว้ ตัวอย่างเช่น ในเอเจนต์ที่ช่วยในการจัดเรียงเมล ผู้ใช้สามารถกำหนดกฎในการโพรเซสเมลที่เข้ามาและ เรียงเข้าใส่โฟลเดอร์ต่างๆ กันได้ กฎเหล่านี้ทำให้เอเจนต์สามารถจัดการงานให้ผู้ใช้โดยที่ผู้ใช้ไม่ต้องร้องขอการบริการเลย ความยากในการใช้วิธีคือการจัดการให้เป็นมาตรฐานนั้นต้องการความเข้าใจในงานอย่างลึกซึ้ง โดยจะต้องผู้ริเริ่มในการสร้างและสอนความรู้แก่เอเจนต์ และต้องปรับปรุงตลอดเวลาหากความต้องการหรือลักษณะการทำงานนั้นเปลี่ยนแปลงไป แต่วิธีนี้จะสร้างความเชื่อถือในตัวเอเจนต์ได้มากเพราะผู้ใช้เป็นผู้กำหนดกฎและหรือทำการโปรแกรมขึ้นมาใช้งานเอง

แนวทางที่สองคือการใช้ฐานความรู้ (Knowledge based) วิธีนี้จะทำให้ข้อมูลเอนต์ด้วยโดเมนโมเดลและ โมเดลของผู้ใช้ ซึ่งจะเป็นการขยายความรู้ภูมิหลังในเรื่องนั้นๆ ให้กับเอเจนต์ เอเจนต์จะใช้ความรู้ภูมิหลังที่ได้รับวิเคราะห์ห่อออกมาเป็นแผนการและหนทางในการทำงาน ตัวอย่างเช่น เอเจนต์ที่ชื่อ “UCEgo” ที่ออกแบบมาเพื่อแก้ปัญหาบนระบบปฏิบัติการยูนิกซ์ UCEgo จะมีฐานความรู้ขนาดใหญ่เกี่ยวกับยูนิกซ์ ปัญหาของวิธีนี้คือ ความต้องการพื้นที่การทำงานขนาดใหญ่ทำให้สิ้นเปลืองขนาดของฐานข้อมูล ความยากลำบากในการวิเคราะห์หาข้อมูลจำนวนมากให้ได้ทั้งหมดเพราะข้อมูลบางอย่างอาจได้มาจากแอคชั่นที่คาดไม่ถึงของผู้ใช้ การค้นหาความรู้นี้อาจใช้วิศวกรเข้ามาช่วยวิเคราะห์ ผู้ใช้จึงมีความรู้ดีกว่าเอเจนต์เข้าใจยากและความคลุมไม่ได้ ผู้ใช้ไม่ทราบถึงข้อจำกัดและแนวทางที่เอเจนต์ทำงาน

แนวทางที่สามเป็นการสร้างเอนต์ที่ทำงานด้วยเทคนิคการเรียนรู้จากแมชีน (Machine learning) การทำงานโดยเทคนิคนี้ตั้งอยู่บนสมมติฐาน ที่ว่าต้องมีความสามารถในการทำการ โปรแกรมตัวเอง กล่าวคือ ในสถานะที่แน่นอน เอเจนต์ต้องสามารถรับรู้ความรู้ที่จำเป็นในการช่วยเหลือผู้ใช้และ / หรือ เอเจนต์อื่นๆ ด้วย โดยมีเงื่อนไขดังนี้

- (1) การใช้แอปพลิเคชันต่างๆ จะต้องมียุติกรรมที่กระทำซ้ำๆ เพื่อให้เอเจนต์ได้มีโอกาสในการเรียนรู้จากแอคชั่นที่เกิดขึ้น
- (2) สำหรับพฤติกรรมที่เกิดขึ้นจะต้องแตกต่างกันถ้าเป็นผู้ใช้คนละคนกัน วิธีการนี้จะมีลักษณะคล้ายว่าเอเจนต์เป็นผู้ช่วยผู้ใช้ ในตอนแรกเอเจนต์ยังไม่คุ้นเคยกับลักษณะนิสัยและความชอบ ความต้องการของผู้ใช้จึงไม่สามารถช่วยอะไรได้มาก พอเวลาผ่านไปผู้ช่วยคุ้นเคยกับแนวทางการทำงานและมีประสบการณ์มากขึ้นก็จะสามารถทำงานแทนได้ แนวทางนี้มีข้อได้เปรียบกว่าสอง

แนวทางก่อนคือ ต้องการการช่วยเหลือจากผู้ใช้และพัฒนาบ่อยลง สามารถปรับเปลี่ยนให้ใช้ได้กับผู้ใช้แต่ละคนได้

3.5 การเรียนรู้ข้อมูลของเอเจนต์

การเรียนรู้ข้อมูลของเอเจนต์ ทำได้ 4 ทางคือ

1. การจับจ้องการทำงานของผู้ใช้อย่างต่อเนื่อง เอเจนต์จะคอยดูการทำการกิจกรรมของผู้ใช้ในช่วงระยะเวลาหนึ่ง หากกฎเกณฑ์และรูปแบบที่เกิดขึ้นบ่อยๆ แล้วทำให้ส่วนนั้นกลายเป็นส่วนที่ทำงานอัตโนมัติโดยเอเจนต์แทนที่จะต้องทำซ้ำๆ โดยผู้ใช้ ตัวอย่างเช่น เมล์เอเจนต์สังเกตพบว่า ผู้ใช้มักเก็บเมลล์ที่ส่งมายังเมลล์ลิสต์หนึ่งเข้าไปในโฟลเดอร์หนึ่งอยู่เป็นประจำ เอเจนต์ก็อาจทำงานนี้แทนถ้ามีเมลล์ลิสต์ดังกล่าวในครั้งต่อไป
7. เรียนรู้โดยตรง หรือ โดยอ้อม จากข้อมูลป้อนกลับ (feed back) ของผู้ใช้ ข้อมูลป้อนกลับทางอ้อมเกิดขึ้นเมื่อผู้ใช้ไม่ได้ใจคำแนะนำของเอเจนต์แต่ไปทำเอกซันอย่างอื่นแทน ซึ่งเป็นการบอกเป็นนัยว่าผู้ใช้เปลี่ยนแปลงความต้องการในการทำกิจกรรมนั้นแล้ว เรียกข้อมูลป้อนกลับนี้ว่า ข้อมูลป้อนกลับในทางลบ เช่น การปฏิเสธ ไม่อ่านบทความที่เอเจนต์หาให้แต่ไปเปิดอ่านบทความอื่นที่เอเจนต์ไม่ได้หาให้
8. เรียนรู้จากตัวอย่างที่ผู้ใช้แสดงให้เห็น ผู้ใช้สามารถฝึกฝนเอเจนต์โดยกำหนดตัวอย่างจากสมมติฐานของเหตุการณ์และระบุว่าให้ทำอย่างไรกับกรณีนั้น ให้เอเจนต์บันทึกเอาไว้ ตัวอย่างเช่น ผู้ใช้สอนให้เอเจนต์ที่ช่วยกรองข่าวสารคัดเลือกรายบทความที่มีคำว่า KMITL มาเก็บไว้
- 4.. เรียนรู้จากการขอคำแนะนำจากเอเจนต์ ที่มีประสบการณ์มากกว่าที่ให้ความช่วยเหลือผู้ช่วยคนอื่นที่จัดการงานๆเดียวกันอยู่ โดยเอเจนต์ที่ขอคำแนะนำจะแสดงสถานะการณันั้นและถามว่าเอเจนต์อื่นในสถานะการณันั้นจะอย่างไร ตัวอย่างเช่น เมล์เอเจนต์ได้รับข้อความจากบุคคลสำคัญท่านหนึ่ง จึงถามเอเจนต์อื่นว่าควรทำอย่างไร เอเจนต์อื่นๆเห็นว่าเมลล์ที่ส่งมามีระดับความสำคัญสูง ให้แสดงผลทันที เอเจนต์นั้นก็แสดงผลเมลล์ดังกล่าวทันทีแม้ว่าผู้ใช้อาจจะไม่ได้ทำการตรวจสอบเมลล์อยู่ก็ตาม

บทที่ 4

รีโมท เมธอด อินโวล์เคชัน

(REMOTE METHOD INVOCATION (RMI))

4.1 บทนำ

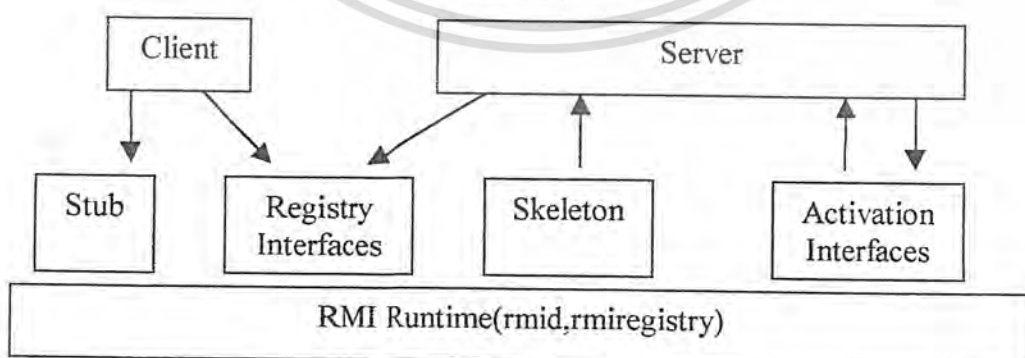
รีโมท เมธอด อินโวล์เคชัน เป็นเทคโนโลยีที่สามารถประมวลผลบนไคลเอนต์เพื่อร้องขอเมธอด (method) ต่างๆ ของออบเจกต์ที่ประมวลผลบนคอมพิวเตอร์ที่เป็นเซิร์ฟเวอร์

การใช้งาน RMI นั้นทั้งฝั่ง ไคลเอนต์และเซิร์ฟเวอร์ต้องถูกเขียนด้วยภาษาจาวา โดยการเรียกใช้รีโมทเมธอดนั้นจะแตกต่างจากการเรียกใช้ โคลลเมธอด ตรงที่โคลลเมธอดจะถูกจัดการโดยคลาสแต่รีโมทเมธอดนั้นจะถูกพิจารณาโดยส่วนติดต่อ(interfaces)

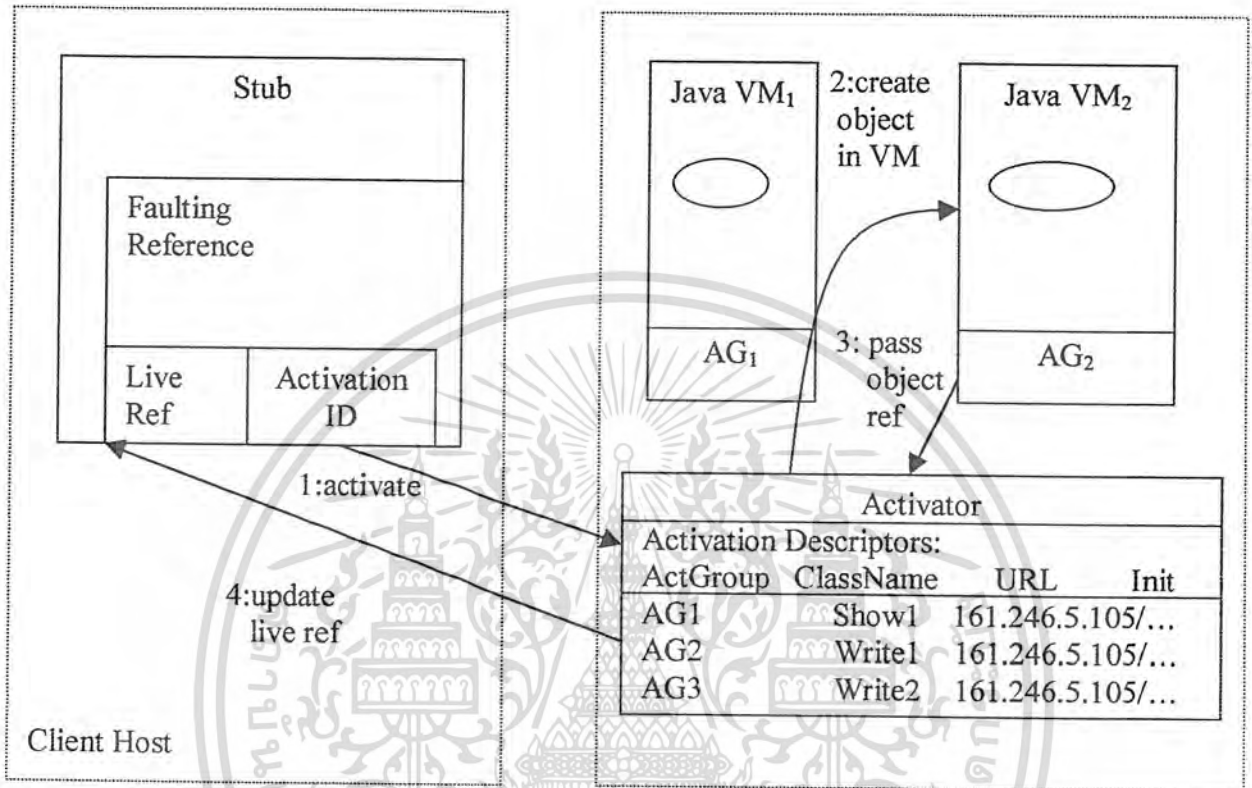
ในการที่จะสร้าง RMI นั้น ไคลเอนต์ออบเจกต์ (client object) ต้องมีส่วนอ้างอิง (references) ไปยังรีโมทออบเจกต์ (remote object) โดยส่วนอ้างอิงนี้จะมีรายละเอียดที่อยู่ของโฮสต์ที่รีโมทออบเจกต์นั้นอยู่เข้าไปด้วย ทั้งนี้ความแตกต่างระหว่างการอ้างอิงโคลลออบเจกต์และรีโมทออบเจกต์นั้นจะโปร่งใสต่อผู้พัฒนาไคลเอนต์ออบเจกต์

4.2 สถาปัตยกรรมของ RMI

ไคลเอนต์จะเริ่มการทำ RMI โดยโคลลเมธอดจะติดต่อไปยังสตับ (stub) โดยที่สตับจะรวบรวมรีโมทเมธอดจากรีโมทออบเจกต์ที่สามารถให้บริการได้บนเซิร์ฟเวอร์ จากนั้นไคลเอนต์รับเอาส่วนอ้างอิงของรีโมทออบเจกต์ ซึ่งอิมพลิเมนต์ในรีจิสตรี ที่ซึ่งเซิร์ฟเวอร์ออบเจกต์จะบันทึกส่วนอ้างอิงของตัวเองลงไป



รูปที่ 4.1 สถาปัตยกรรมของ RMI



รูปที่ 4.2 การสร้างรีโมทออบเจกต์ใน RMI

บทที่ 5

XML

(Extensible Markup Language)

5.1 บทนำ

เว็บแอปพลิเคชันในปัจจุบันนี้กำลังเผชิญกับปัญหาที่ไม่เคยมีใครคิดเมื่อ 10 ปีก่อน อันเนื่องมาจากพัฒนาระบบคอมพิวเตอร์แบบกระจาย (Distributed Computer System) ซึ่งคอมพิวเตอร์แต่ละเครื่องอาจมีระบบปฏิบัติการและแอปพลิเคชันที่ใช้งานแตกต่างกันไป ทำให้เกิดปัญหาคือไม่สามารถติดต่อสื่อสารหรือส่งข้อมูลโยกย้ายระหว่างระบบเพื่อนำไปใช้งานร่วมกันได้

XML (Extensible Markup Language) เป็นวิธีการสำหรับเก็บข้อมูลแบบโครงสร้าง (structured data) ลงในไฟล์ตัวอักษร โดยมีกฎเกณฑ์ เพื่อเป็นสื่อกลางในการแลกเปลี่ยนสำหรับระบบคอมพิวเตอร์ที่แตกต่างกัน

5.2 ลักษณะของภาษา XML

XML เป็นภาษาที่ไม่ได้มีการระบุชุดของแท็ก (tag set) และ ไวยากรณ์ของภาษา ทำให้ xml สามารถขยายได้ ใก้ดังต่อไปนี้ เป็นตัวอย่างของ XML ที่ใช้ในการระบุข้อมูลแบบโครงสร้าง “ห้องรับประทานอาหาร”

```
<?xml version="1.0"?>
<dining-room>
  <table type="round" wood="maple">
    <manufacturer> The Wood Shop </manufacturer>
    <price>$1999.99</price>
  </table>
  <chair wood="maple">
    <quantity>2</quantity>
    <quality>excellent</quality>
    <cushion included = "true">
      <color>blue</color>
    </cushion>
  </chair>
```

```

<chair wood="oak">
  <quantity>3</quantity>
  <quality>average</quality>
</chair>
</dining-room>

```

รูปที่ 5.1 ตัวอย่างไฟล์ XML

ข้อมูลโครงสร้างอื่นๆ ได้แก่ สเปรดชีต (spreadsheets), ข้อมูลการทำธุรกรรมทางการเงิน, ไฟล์เอกสารในรูปแบบต่างๆ ซึ่งถูกบันทึกลงสื่อต่างๆ อาจจะเป็นได้ทั้งรูปแบบไบนารี (Binary format) หรือ รูปแบบตัวอักษร (Text format) เมื่อถูกบันทึกในรูปแบบของ XML แล้วสามารถที่จะแลกเปลี่ยนกันระหว่างแอปพลิเคชันซึ่งสนับสนุน XML เพื่อแก้ไข อ่าน บันทึก เป็นข้อมูลของแอปพลิเคชันนั้นๆ ได้

5.3 องค์ประกอบย่อยของ XML

5.3.1 PI

PI ในเอกสาร XML คือคำสั่งเพื่อการประมวลผล (processing instruction) เป็นคำสั่งที่บอกแอปพลิเคชันเพื่อให้ทำการบางอย่าง แม้ว่า PI เป็นเพียงส่วนเล็กๆ ใน XML แต่ก็สำคัญมาก เนื่องจาก PI เป็นตัวที่ใช้แสดงคำสั่งต่อแอปพลิเคชันที่ใช้เอกสาร XML โดยมีรูปแบบคือ `<?target instructions?>` เช่น `<?xml version="1.0">` เป็นการบอกแอปพลิเคชันว่า xml ที่ใช้เป็นเวอร์ชัน 1.0

5.3.2 DTD

DTD คือ นิยามชนิดเอกสาร (Document type definition) ทำหน้าที่ตั้งชุดข้อบังคับสำหรับเอกสาร XML ซึ่งประกอบด้วยข้อบังคับในตัวเอกสาร XML และข้อบังคับภายนอกเอกสารที่ถูกอ้างอิงด้วย DTD เป็นส่วนที่บอกว่าเอกสาร XML จะถูกสร้างอย่างไร

เอกสาร XML จะถูกต้อง (Valid) ก็ต่อเมื่อทำตามข้อบังคับใน DTD ซึ่งสำคัญมากเมื่อมีการส่งข้อมูลระหว่างแอปพลิเคชัน โดยต้องมีการตกลงรูปแบบและไวยากรณ์ในแต่ละระบบเพื่อความเข้าใจร่วมกัน

ตัวอย่างเช่นแท็ก `<cushion>` สามารถปรากฏซ้อนอยู่ในแท็ก `<chair>` เท่านั้น ซึ่งสิ่งนี้ทำให้แอปพลิเคชันอื่นรู้วิธีประมวลผลไฟล์ที่ได้รับมาได้ DTD เป็นสิ่งที่เพิ่มความสามารถในการโยกย้าย (portability) ให้กับเอกสาร XML ไม่เพียงแต่ข้อมูลที่ยึดหยุ่นได้เท่านั้นแต่รวมถึงข้อมูลที่สามารถประมวลผลและตรวจสอบความถูกต้องได้โดยเครื่องมือ DTD ของเอกสารนั้น

5.3.3 Namespaces

เนมสเปซคือการจับคู่ระหว่างปริฟิกซ์(prefix) ของแต่ละข้อมูลกับ URI ใช้เพื่อการรับมือกับการซ้ำกันของเนมสเปซ และเพื่อระบุโครงสร้างข้อมูลที่อนุญาตพาสเซอร์ (parsers) ให้สามารถจัดการกับความซ้ำซ้อนได้ ยกตัวอย่างความซ้ำซ้อนเช่น เอกสาร XML หนึ่งอาจมีแท็ก <date> อยู่ ซึ่งอาจมีปรากฏอยู่ในเอกสาร XML อีกไฟล์หนึ่งได้เช่นกันทำให้เกิดความสับสนแก่ พาสเซอร์ว่าจะแปลความหมายอย่างไร

เนมสเปซจึงระบุกลไกเพื่อจัดคุณสมบัติของชื่อเหล่านี้ โดยใช้ URI เพื่อกำหนดคปริฟิกซ์ให้แก่ข้อมูลแต่ละตัวเช่น <program-one:date> และ <program-two:date> แทนที่แท็ก <date> ในแต่ละเอกสารทำให้พาสเซอร์ของ XML สามารถแยกความแตกต่างระหว่างเนมสเปซทั้งสองได้

เนมสเปซโดยมากมักจะปรากฏในเอกสาร XML แต่ยังสามารถถูกใช้ใน XSL ได้อีกด้วย

5.3.4 XSL และ XSLT

XSL คือ ภาษาสำหรับแผ่นกำหนดรูปแบบที่สามารถเพิ่มเติมได้ (Extensible Stylesheet Language) XSL ทำหน้าที่แปลงและ แปล ข้อมูล XML จากรูปแบบหนึ่งไปเป็นอีกรูปแบบหนึ่ง ตัวอย่างเช่น เอกสาร XML หนึ่งอาจต้องการแสดงผลในรูปแบบต่างๆเช่น HTML , PDF, และรูปแบบ Postscript ถ้าหากไม่มี XSL แล้ว เอกสาร XML นั้นต้องถูกทำสำเนาหลายชุดสำหรับแต่ละรูปแบบ จากนั้นแปลงเป็นรูปแบบต่างๆ แต่เนื่องจาก XSL เดิมมีกลไกสำหรับทำกระบวนการนี้ไว้แล้ว สำหรับแบ่งแยกข้อมูล และ การแสดงผล ถ้าหากเอกสาร XML ต้องการที่จะแปลงสำหรับการแสดงผลในรูปแบบอื่นแล้ว XSL คือคำตอบที่ดีที่สุด

```
<?xml version="1.0"?>
<?xml-stylesheet href="hello.xsl" type="text/xsl"?>

<!-- Here is a sample XML file -->

<page>
  <title>Test Page</title>
  <content>
    <paragraph>What you see is what you get!</paragraph>
  </content>
</page>
```

รูปที่ 5.2 ตัวอย่าง แสดงไฟล์XML

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="page">
```

```
<html>
  <head>
    <title>
      <cs:value-of select="title"/>
    </title>
  </head>
  <body bgcolor="#ffff">
    <cs:apply-templates/>
  </body>
</html>
</xsl:template>
```

```
<xsl:template match="paragraph">
  <p align="center">
    <i>
      <xsl:apply-templates/>
    </i>
  </p>
</xsl:template>
</xsl:stylesheet>
```

รูปที่ 5.3 ตัวอย่าง XSL สำหรับเอกสาร XML

```
<html>
  <head>
    <title> Testpage </title>
  </head>
  <body bgcolor="ffff">
    <p align="center">
      <i>
        What you see is what you get !
      </i>
    </p>
  </body>
</html>
```

รูปที่ 5.4 ตัวอย่าง ไฟล์ HTML ที่ได้จาก XML และ XSL

5.3.5 Xpath

Xpath (XML Path Language) เป็นข้อกำหนดที่ถูกใช้อย่างมากโดย XSL ข้อกำหนด Xpath ระบุว่า จะค้นหาข้อมูลที่เจาะจงในเอกสาร XML ได้อย่างไร ซึ่งทำได้โดยการระบุเจาะจงถึง โหนด (Node) ในเอกสาร XML คำว่า โหนด หมายถึง ข้อมูล XML ใดๆ ได้แก่ แอททริบิวต์, สมาชิก (Element), ข้อมูลที่เป็นตัวอักษร ตัวอย่างการใช้ Xpath ได้แก่

*[not(self::JavaXML:Title)]

Xpath นี้หาคำให้แก่สมาชิกทุกตัวของสมาชิกตัวปัจจุบัน โดยที่ ชื่อของสมาชิกต้องไม่เป็น JavaXML:Title สำหรับเอกสารส่วนนี้

```
<JavaXML:Book>
  <JavaXML:Title>Java and XML</JavaXML:Title>
  <JavaXML:Content>
    <!--Chapters go here -->
  </JavaXML:Content>
  <JavaXML:Copyright>&OreillyCopyright;</JavaXML:Copyright>
</JavaXML:Book>
```

5.3.6 XQL

XQL เป็นภาษาค้นคืนถูกออกแบบมาเพื่อให้รูปแบบเอกสาร XML สามารถแสดงผลการค้นคืนของฐานข้อมูล (Database) ได้โดยง่ายแม้ว่ายังไม่มีมาตรฐานอย่างเป็นทางการออกมาจาก W3C ก็ตามแต่ XQL ก็ได้รับความนิยมอย่างมากในความเป็นประโยชน์ ในการระบุเข้าถึงข้อมูลใน Database โดยผ่าน เอกสาร XML โครงสร้างของการค้นคืนระบุโดยอาศัยแนวคิดของ Xpath และผลลัพธ์ระบุโดยมาตรฐาน XML โดยมี XQL แท็ก

ยกตัวอย่างเช่นคำสั่ง XQL ต่อไปนี้ทำหน้าที่ค้นหาตาราง books ทั้งตาราง และส่งเรคอร์ดทุกเรคอร์ด ซึ่งมี “java” รวมอยู่ในชื่อเรื่อง

```
//book[title contains “Java”] (./authors)
เซตของผลลัพธ์จะเป็นดังนี้
<xql:result>
<book>
  <author name = “Richard Monson-Haefel” Location=“Minnesota” />
  <author name = “Jason Hunter” location = “california”/>
  <authorname = “William Crawford” location = “Massachusetts”/>
</book>
</xql:result>
```

บทที่ 6

โครงสร้างและการทำงานของระบบ

6.1 การเลือกโปรแกรมประยุกต์และสิ่งแวดล้อมที่ใช้สร้างโปรแกรมการทำงาน

งานวิจัยในโครงการนี้มีวัตถุประสงค์ที่สร้างระบบซึ่งสามารถให้ผู้ใช้แต่ละคนซึ่งอยู่ต่างสถานสามารถทำงานงานร่วมกันได้อย่างมีประสิทธิภาพ มีปัจจัยที่ต้องคำนึงถึงอยู่หลายประการ ได้แก่ โคลเอนต์ แอปพลิเคชัน การสื่อสารกันระหว่างคอมพิวเตอร์ โปรแกรมเซิร์ฟเวอร์

โคลเอนต์แอปพลิเคชัน เป็นแอปพลิเคชันที่เขียนด้วยภาษาจาวา โดยใช้หลักการของแอปเพล็ต เพื่อเพิ่มประสิทธิภาพในการสร้าง Graphic user interface(GUI) ที่ง่ายต่อการใช้งาน

โปรแกรมเซิร์ฟเวอร์ เป็นโปรแกรมที่เขียนด้วยภาษาจาวา โดยใช้หลักการ Distributed Systems สร้างรีโมท เมธอด อินโวนเคชัน เป็นเทคโนโลยีที่สามารถประมวลผลบนคอมพิวเตอร์ท้องถิ่นเพื่อร้องขอเมธอดต่างๆของออปเจกต์ที่อยู่ห่างไกล หรือพูดง่ายๆว่า RMI ช่วยให้เราสามารถเรียกใช้ออปเจกต์ระยะไกล จากหลายๆ JVM(Java Virtual Machine) โดยผ่าน LAN, WAN, หรือ WWW(World Wide Web) ได้ โปรแกรมการทำงานทั้งหมดจะถูกเก็บไว้ในฝั่ง

6.2 ขั้นตอนการสร้างโปรแกรมการทำงานร่วมกัน

1. สร้าง GUI ที่ง่ายต่อการใช้งาน
2. สร้างโคลเอนต์โปรแกรมด้วยภาษาจาวา ที่สามารถทำการติดต่อกับเซิร์ฟเวอร์ได้
3. สร้างเซิร์ฟเวอร์โปรแกรมที่สามารถจัดการการทำงานร่วมกันระหว่างผู้ใช้แต่ละคนให้มีความราบรื่น และถูกต้องตามลำดับก่อนหลัง และจัดความขัดแย้งของผู้ใช้แต่ละคน

6.3 โครงสร้างของโปรแกรมการทำงานร่วมกัน



6.1 แสดงโครงสร้างการทำงาน

6.4 อินเทอร์เฟซที่ติดต่อกับผู้ใช้



6.2 อินเทอร์เฟซที่ติดต่อกับผู้ใช้

ประกอบด้วย 4 ส่วน ดังนี้

- เมนูบาร์ ได้แก่ File และ Status
- TextField เป็นส่วนที่ให้ผู้ใช้ในการป้อนข้อความเพื่อแสดงผล
- ส่วนติดต่อกับเซิร์ฟเวอร์ ได้แก่ การเลือกบรรทัด การลบบรรทัด การยกเลิกการลบบรรทัด และการตกลงส่งข้อความ
- ส่วนเลือกลักษณะตัวอักษร ได้แก่ FONT STYLE COLOR และ SIZE

ในการประพันธ์เอกสาร เราสามารถเลือกกระทำบรรทัดใดบรรทัดหนึ่งได้เพียงบรรทัดเดียวต่อหนึ่งการทำงาน และไม่สามารถกระทำบรรทัดอื่นได้จนกว่าจะยกเลิกการขอกระทำบรรทัดที่ทำงานอยู่ โดยการขออนุญาตทำงานบรรทัดใดบรรทัดหนึ่งจะขึ้นอยู่กับเซิร์ฟเวอร์ที่เราร้องขอ เมื่อได้รับอนุญาตกระทำบรรทัดใดบรรทัดหนึ่งแล้ว จึงสามารถพิมพ์ข้อความที่ต้องการลงในเท็กซ์ฟิลด์และเลือกลักษณะตัวอักษรตามต้องการ หลังจากนั้นจึงส่งข้อความไปยังเซิร์ฟเวอร์ เซิร์ฟเวอร์ก็จะทำการส่งข้อความกลับไปให้ผู้ใช้งานทุกคน เพื่อแสดงผลที่หน้าต่างแสดงผล แล้วทางผู้ใช้งานต้องยกเลิกบรรทัดที่ทำงานอยู่เพื่อไปทำงานยังบรรทัดต่อไป เมื่อผู้ใช้งานการสิ้นสุดการทำงานจะสามารถเชฟผลลัพธ์ของการทำงานสุดท้าย

6.5 โพรโทคอลที่ใช้ในการติดต่อกันระหว่างไคลเอนต์และเซิร์ฟเวอร์

โพรโทคอลที่ใช้ในการติดต่อกันระหว่างไคลเอนต์และเซิร์ฟเวอร์ตลอดการทำงานนั้นเป็นโพรโทคอลที่ใช้ทางผู้จัดทำได้กำหนดขึ้นเอง โดยให้พื้นฐานอยู่บน TCP/IP รูปแบบข้างล่าง แสดงเท็กซ์ฟอร์มที่กำหนดขึ้นมา

<FONT="STYLE" "SIZE"><TYPE><COLOR>ข้อมูลทำการส่ง</COLOR></TYPE>

STYLE คือ รูปแบบตัวอักษร เช่น TimesRoman, Helvetica, Courier, Dialog เป็นต้น

SIZE คือ ขนาดตัวอักษรโดยแสดงเป็นตัวเลข

TYPE คือ ชนิดตัวอักษร เช่น Bold Italic เป็นต้น

COLOR คือ ลักษณะสีของตัวอักษร เช่น Red Yellow Black เป็นต้น

ตัวอย่างในการส่งแท็กฟอร์ม

```
<FONT="TimesRoman"26"><Bold><Black>This is my test.</Black></Bold></FONT>
```

จากแท็กฟอร์มข้างบนนี้สามารถแปลความหมายได้ ดังนี้

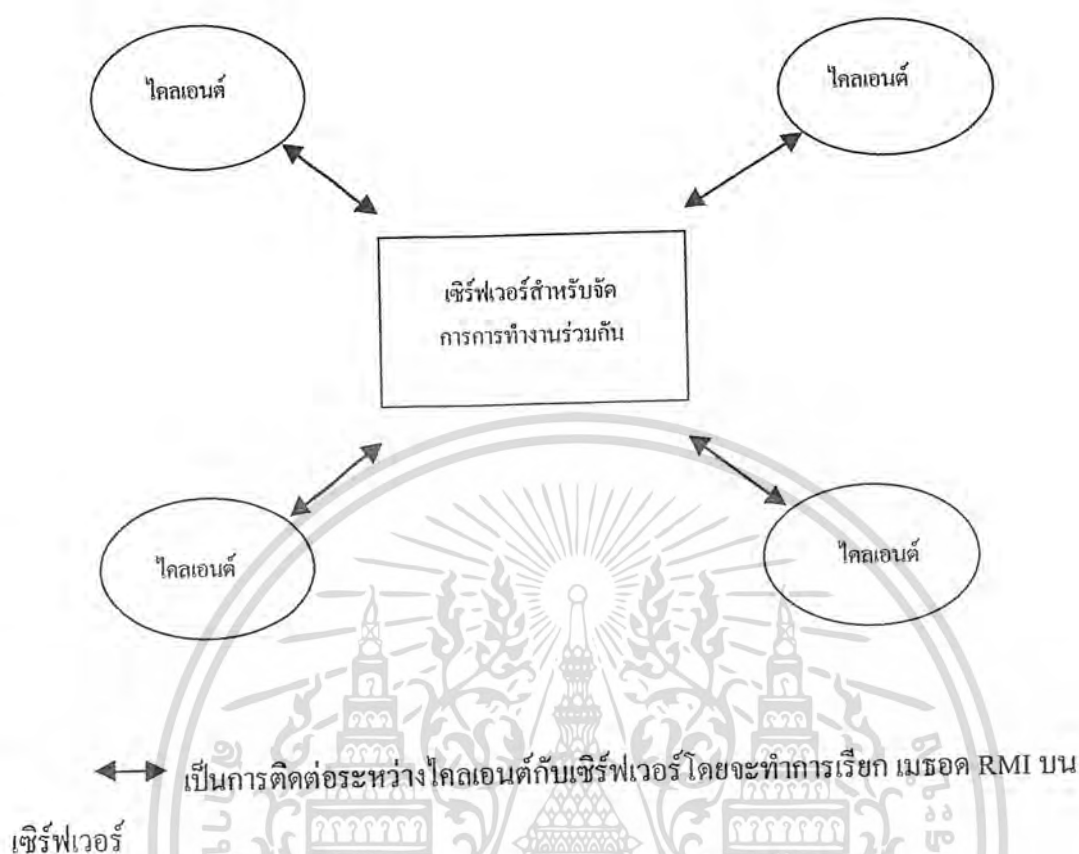
This is my test.

ในการส่งแท็กฟอร์มนั้นสามารถที่เดียวหลายๆแท็กได้ โดยโปรแกรมจะแปลความหมายได้เป็นข้อมูล ที่ติดกัน เราสามารถเพิ่มแท็ก
 ไว้หลังสุด หมายความว่าให้ขึ้นบรรทัดใหม่

ประโยชน์ที่ได้รับจากการใช้โฟโตครอลน์ คือ ที่ไคลเอนต์สามารถแสดงผลฟอร์มได้หลากหลายรูปแบบ จึงเป็นการช่วยเพิ่มประสิทธิภาพในการแสดงผลต่อผู้ใช้

6.6 การทำงานของโปรแกรมการทำงานร่วมกัน

การทำงานของโปรแกรมการทำงานร่วมกันในงานเอกสารนั้นจะเป็นการสื่อสารสองทาง ระหว่างไคลเอนต์และเซิร์ฟเวอร์ ภายใต้โพรโทคอล ดังที่มาแล้วข้างต้น ในการทำงานจะเป็นการใช้แหล่งข้อมูลกลางร่วมกันภายในกลุ่มการทำงานเดียวกัน ทำให้เห็นผลลัพธ์การทำงานเหมือนกันและพร้อมกัน เมื่อผู้ใช้ขออนุญาตเข้ามาทำงาน โปรแกรมจะเก็บบันทึกข้อมูลผู้ใช้และแยกเข้ากลุ่มการทำงาน ทำการสร้างอินสแตนซ์ของแอปพลิเคชันขึ้นมาอีกหนึ่งตัว ซึ่งจะมีการทำงานเป็นเซรค์และเป็นอิสระต่อกันระหว่างผู้ใช้แต่ละคน แอปพลิเคชันแต่ละตัวจะทำการรีเฟรชตัวเองเพื่อให้ได้รับข้อมูลที่ใหม่อยู่ตลอดเวลา ส่วนโปรแกรมในเซิร์ฟเวอร์จะทำหน้าที่ในการจัดการแก้ไขเปลี่ยนแปลงข้อมูลกลางของกลุ่มการทำงานทุกกลุ่มและส่งข้อมูลให้ผู้ใช้ในแต่ละการรีเฟรช



รูปที่ 6.3 แสดงการติดต่อกันระหว่างเซิร์ฟเวอร์กับไคลเอนต์

คุณสมบัติของโปรแกรมการทำงานร่วมกัน มีดังต่อไปนี้

6.6.1. การแสดงผลข้อความ

ผู้ใช้แต่ละคนในกลุ่มการทำงานสามารถพิมพ์ข้อความได้อย่างอิสระภายในเท็กซ์ฟิลด์(Text Field)ของตัวเอง ซึ่งส่วนเท็กซ์ฟิลด์ดังกล่าวจะเห็นได้เฉพาะเพียงผู้ใช้และยังไม่ถูกนำไปแสดงผลบนหน้าจอแสดงผลพลัฟ เมื่อทำการแก้ไขเป็นที่พอใจแล้ว ผู้ใช้จะส่งเท็กซ์ไปยังเซิร์ฟเวอร์ก่อน เพื่อทำการเก็บผลลงไฟล์ และเมื่อถึงเวลารีเฟรช เซิร์ฟเวอร์จะส่งเท็กซ์นี้ไปแสดงผลที่ทุกๆ ไคลเอนต์



รูปที่ 6.4 แสดงขั้นตอนการแสดงผลข้อความ

6.6.2. การลือกบรรทัดเพื่อขอการทำงาน

ก่อนการประพันธ์เอกสาร จำเป็นต้องมีการร้องขอการลือกบรรทัดไปยังเซิร์ฟเวอร์ จึงจะสามารถทำการสร้างหรือแก้ไขเอกสารในบรรทัดที่ร้องขอได้ เมื่อเสร็จสิ้นแล้วจำเป็นต้องทำการยกเลิกการลือก จึงจะสามารถกระทำกับบรรทัดอื่นต่อไปได้

ในส่วนเซิร์ฟเวอร์จะรับการร้องขอการลือกจากผู้หลายผู้ใช้ โดยมีหลักการพิจารณาคือ เมื่อผู้ใช้ร้องขอ บรรทัดที่ยังไม่มีผู้ใช้อื่นร้องขอก็จะได้บรรทัดนั้นไป เมื่อหลายผู้ใช้อื่นร้องขอบรรทัดเดียวกันก็จะให้กับผู้ใช้ที่ร้องขอก่อน

สาเหตุที่ให้มีการลือก ก็เพื่อที่จะจัดข้อขัดแย้งในการที่มีผู้ใช้ทำการแก้ไขข้อความเดียวกัน และสามารถแบ่งแยกการทำงานเพื่อให้การประพันธ์เอกสารรวดเร็วยิ่งขึ้น

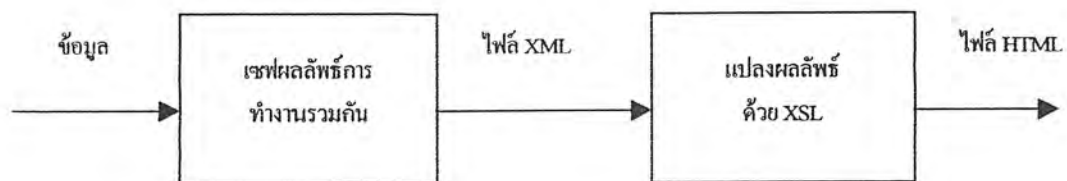


รูปที่ 6.5 แสดงขั้นตอนการจัดการการล็อกข้อความในส่วนเซิร์ฟเวอร์

6.6.3. การเซฟผลลัพธ์การทำงาน

ผลลัพธ์ของการทำงานจะถูกบันทึกผลให้อยู่ในรูปของภาษาริพรีเซ้นเตชัน คือ XML เมื่อผู้ใช้กดปุ่ม “Save” แอปพลิเคชันจะส่งข้อมูลไปบอกเซิร์ฟเวอร์ให้ทำการเขียนไฟล์นามสกุล XML ขึ้น

ข้อดีของการเก็บผลลัพธ์ในรูปของภาษา XML คือ สามารถใช้ในการแปลงไปเป็นภาษามาร์คอัพอื่นๆ ได้ง่าย เช่น แปลงจากภาษา XML ไปเป็นภาษา HTML หรือ แปลงจากภาษา XML ไปเป็นภาษา WML เป็นต้น ทั้งนี้ต้องอาศัยสไตลชีต(style sheet)ที่มีลักษณะเฉพาะสำหรับการแปลงไปสู่ภาษามาร์คอัพอื่นๆ



รูปที่ 6.6 แสดงการแปลงผลลัพธ์การทำงาน



บทที่ 7

การทดลองและวิเคราะห์ผลการทดลอง

7.1 บทนำ

ในบทนี้จะเป็นการแสดงการทดลองและผลการทดลองของการใช้งาน โปรแกรมการทำงานร่วมกันในงานเอกสาร และทำการวิเคราะห์ถึงผลของการทำงาน ปัญหาและแนวทางแก้ไขปัญหาที่เกิดขึ้น

7.2 การทดสอบและใช้งานโปรแกรม

7.2.1 ส่วนประกอบของหน้าต่างการใช้งาน

หน้าต่างการใช้งานประกอบด้วย 2 ส่วนหลักๆ คือ

1. ส่วนแสดงผล (Editor Client) เป็นส่วนที่ทุกผู้ใช้งานจะเห็นเหมือนกันในการประพันธ์เอกสารเดียวกัน
2. ส่วนติดต่อผู้ใช้ (User Interface) เป็นส่วนที่ให้ผู้ใช้งานป้อนคำสั่งในการประพันธ์เอกสารร่วมกัน ซึ่งมีลำดับการทำงานดังนี้ ผู้ใช้ต้องทำการเลือกบรรทัดที่ต้องการทำงาน จากนั้นจึงพิมพ์ข้อความที่ต้องการแสดงในส่วนแสดงผลในลงในText field. ทุกครั้งที่พิมพ์ข้อความเสร็จให้กด ENTER แล้วจึงเลือกคุณลักษณะของตัวอักษร สุดท้ายกดปุ่ม OK เมื่อต้องการทำงานบรรทัดอื่นต่อ ให้กดปุ่ม UNLOCK ก่อน แล้วจึงทำงานส่วนอื่นๆของเอกสารได้

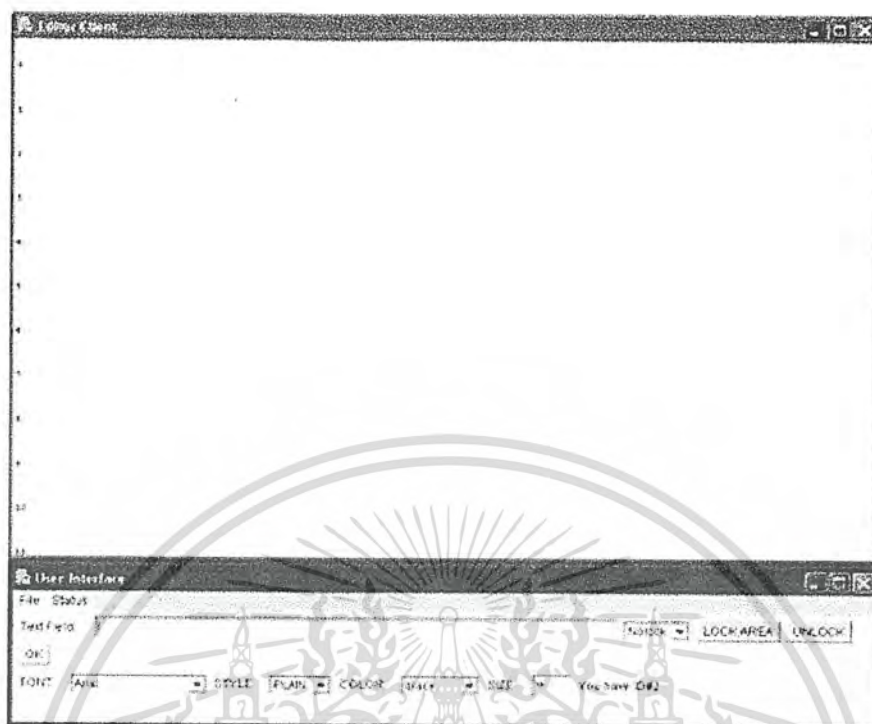
ถ้าไม่เลือกคุณลักษณะของตัวอักษร โปรแกรมจะกำหนดให้เอง ดังนี้

Font = Arial

Style = PLAIN

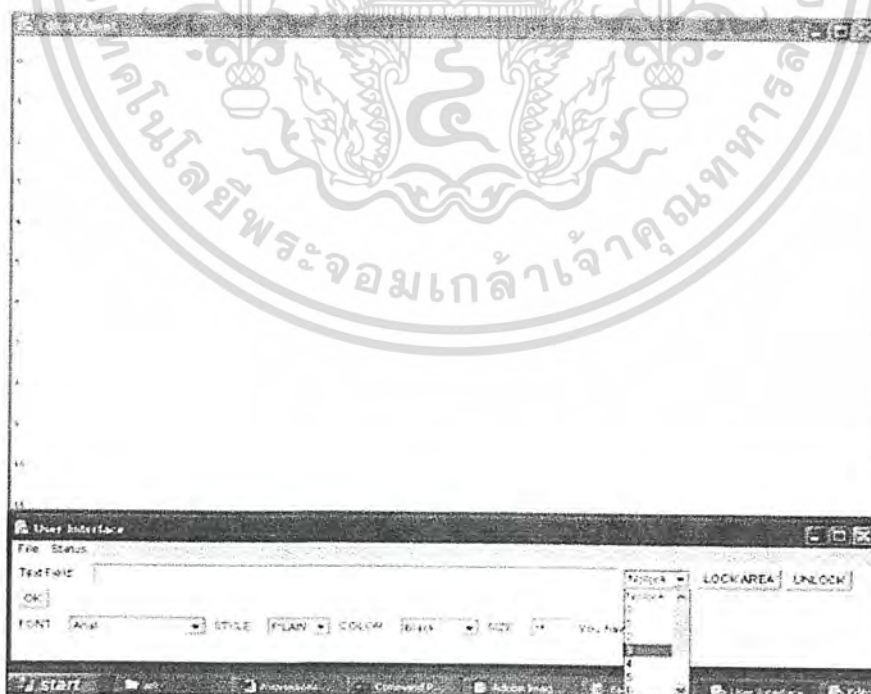
Color = black

Size = 14

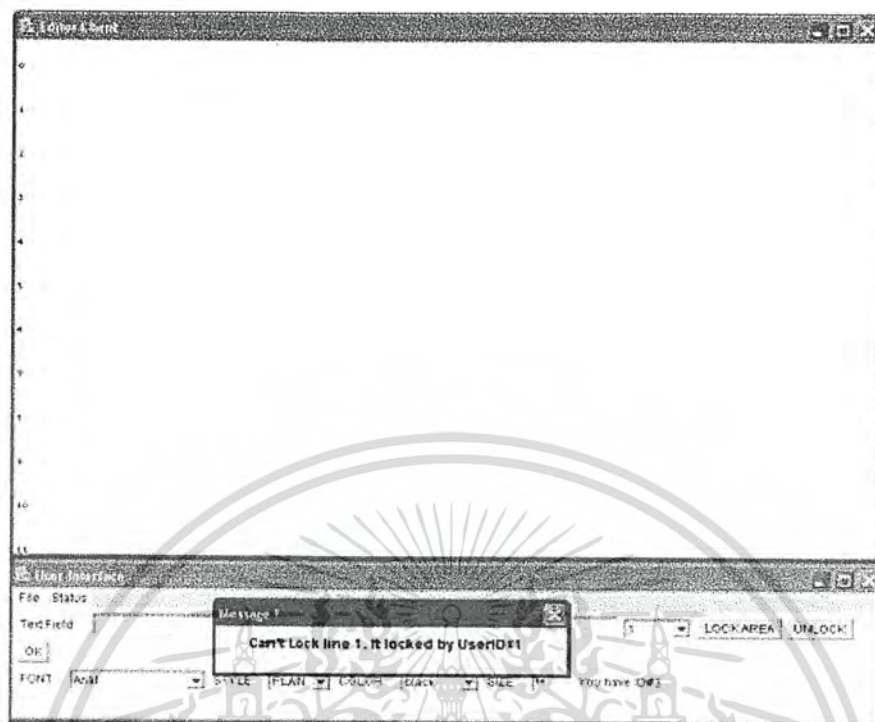


รูปที่ 7.1 ตัวอย่างโปรแกรม

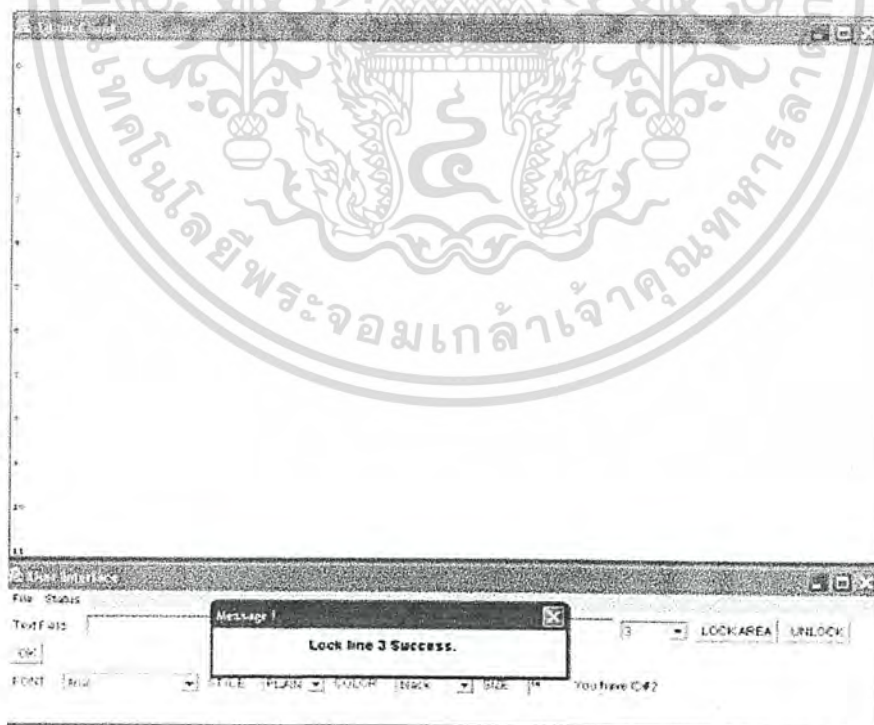
7.3 ตัวอย่างการทำงานร่วมกันในการเลือกบรรทัด



รูปที่ 7.2 เลือกบรรทัดที่ต้องการคลิก

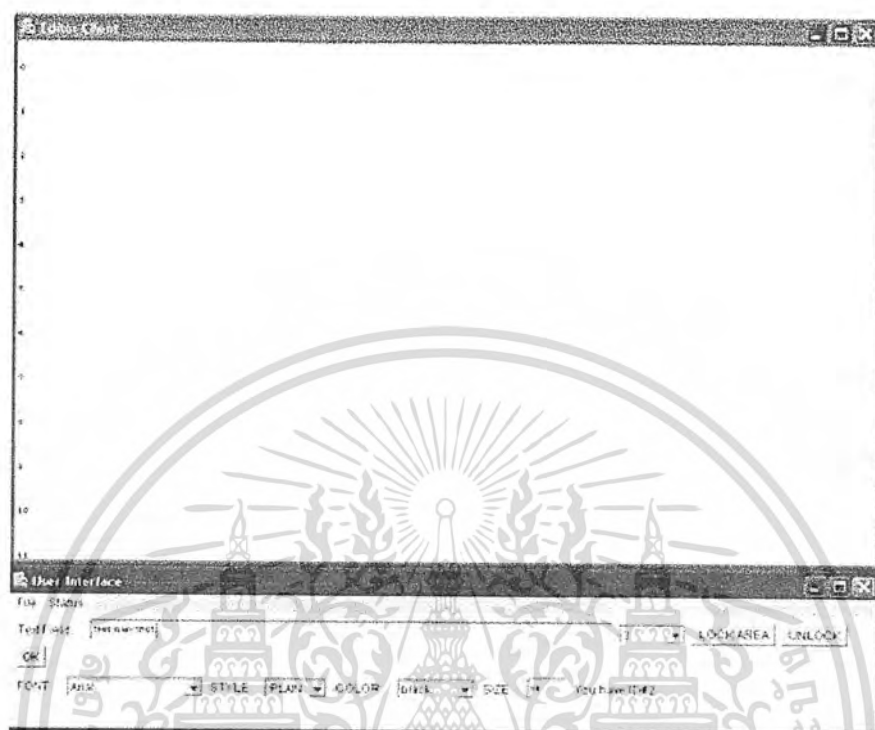


รูปที่ 7.3 ขอล็อกบรรทัดที่ไม่สำเร็จ เพราะถูกล็อกอยู่โดยผู้ใช้นอื่น



รูปที่ 7.4 การร้องขอล็อกสำเร็จ

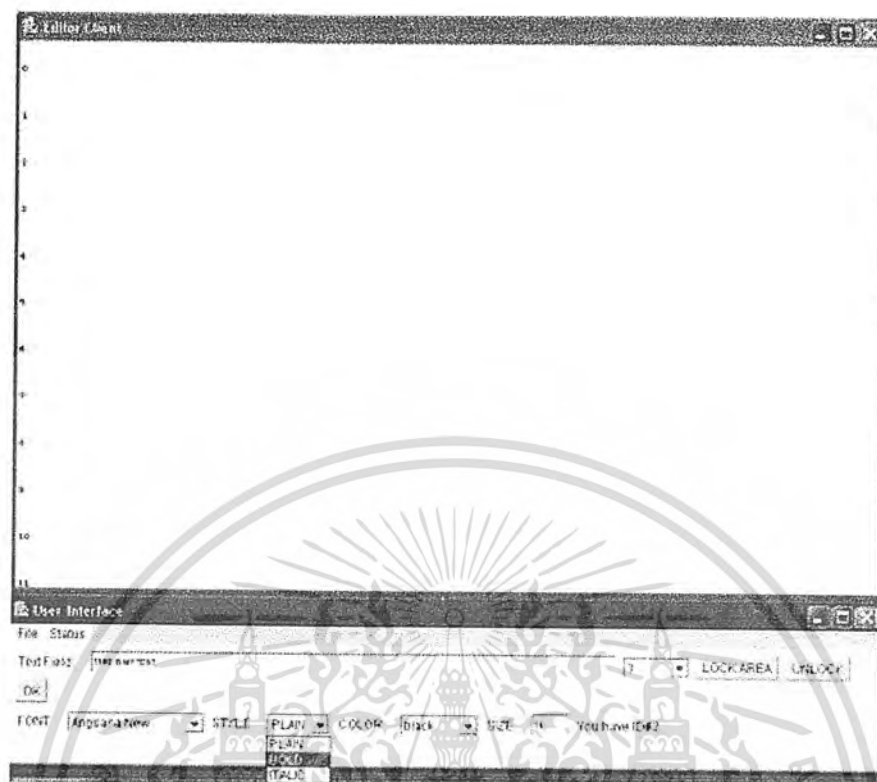
7.4 ตัวอย่างการทำงานร่วมกันในการแสดงข้อความที่จอภาพ



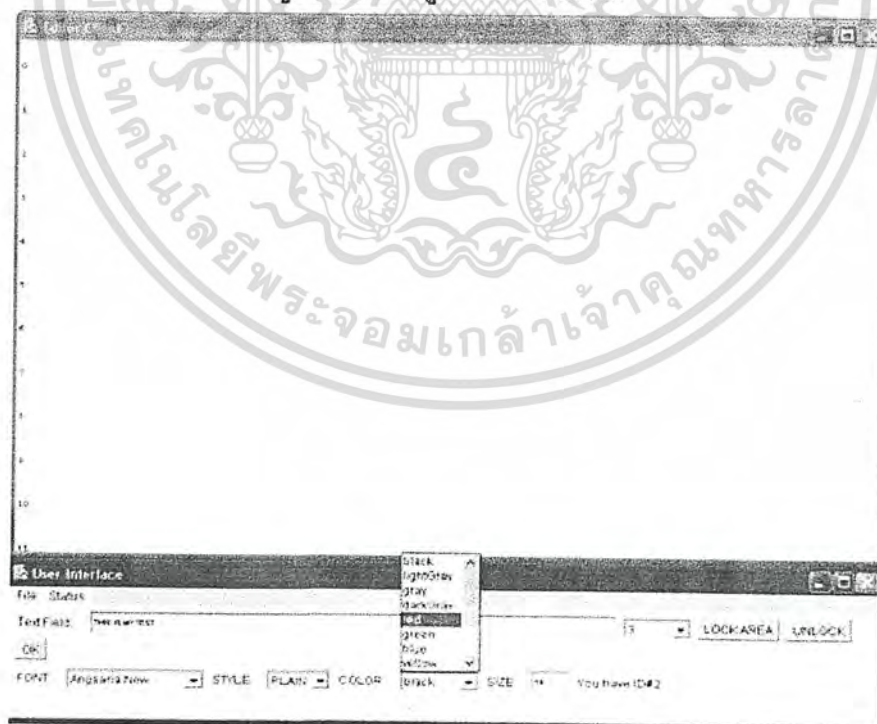
รูปที่ 7.5 พิมพ์ข้อความลงในเท็กซ์ฟิลด์



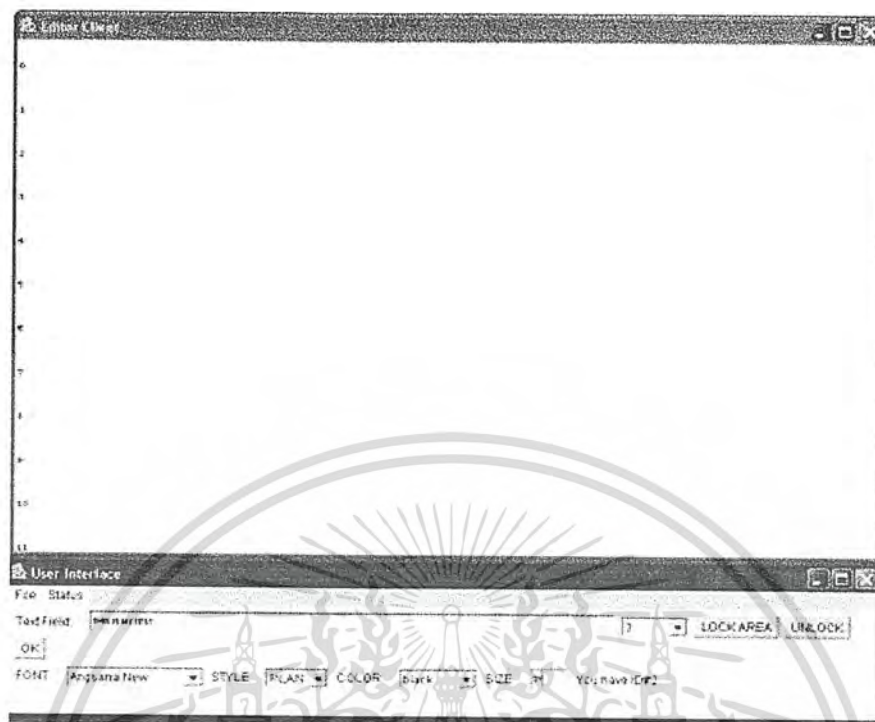
รูปที่ 7.6 เลือกฟอนท์ให้ข้อความ



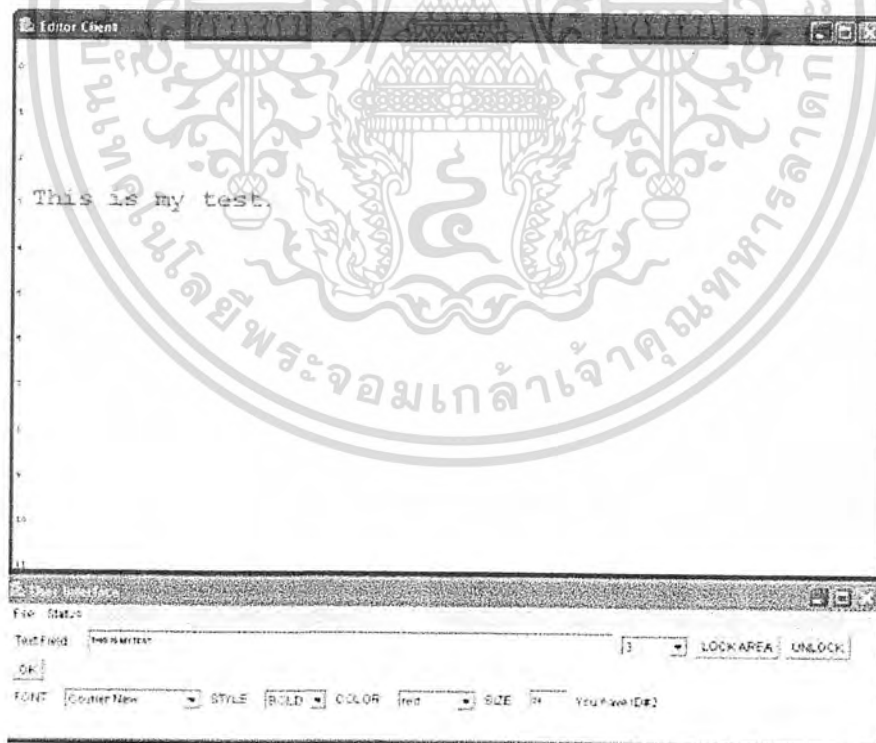
รูปที่ 7.7 เลือกรูปแบบให้ข้อความ



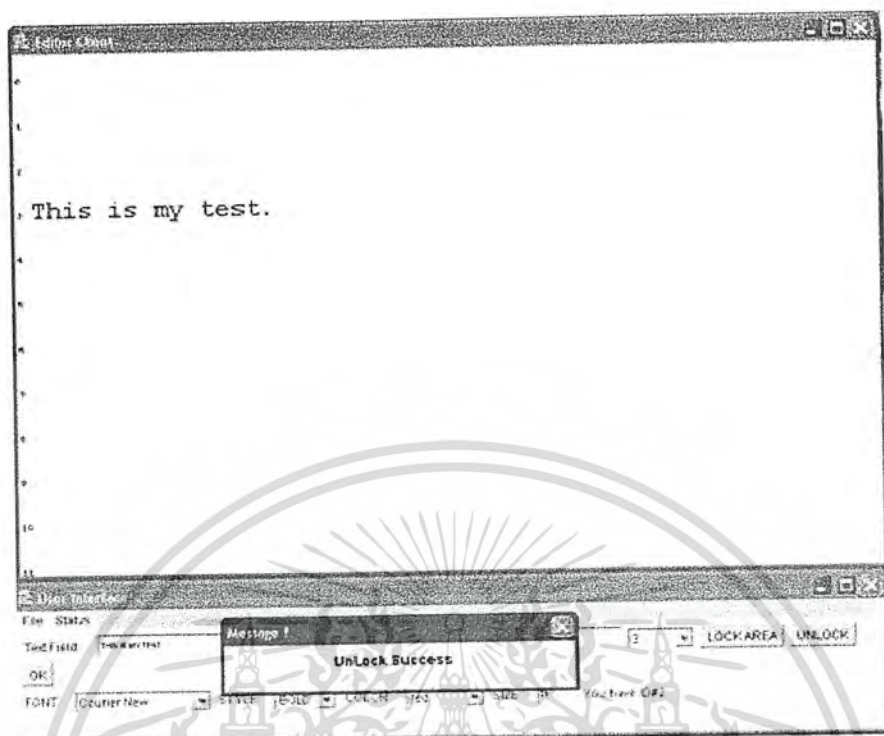
รูปที่ 7.8 เลือกสีให้ข้อความ



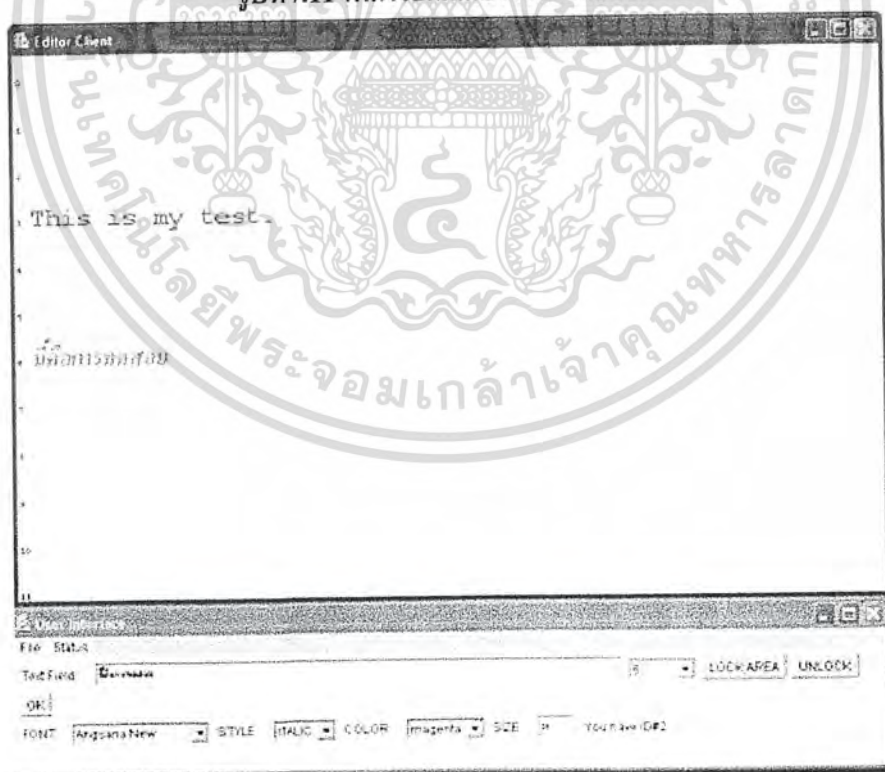
รูปที่ 7.9 เลือกขนาดให้ข้อความ



รูปที่ 7.10 เมื่อกดปุ่ม OK เพื่อแสดงผลที่จอภาพ



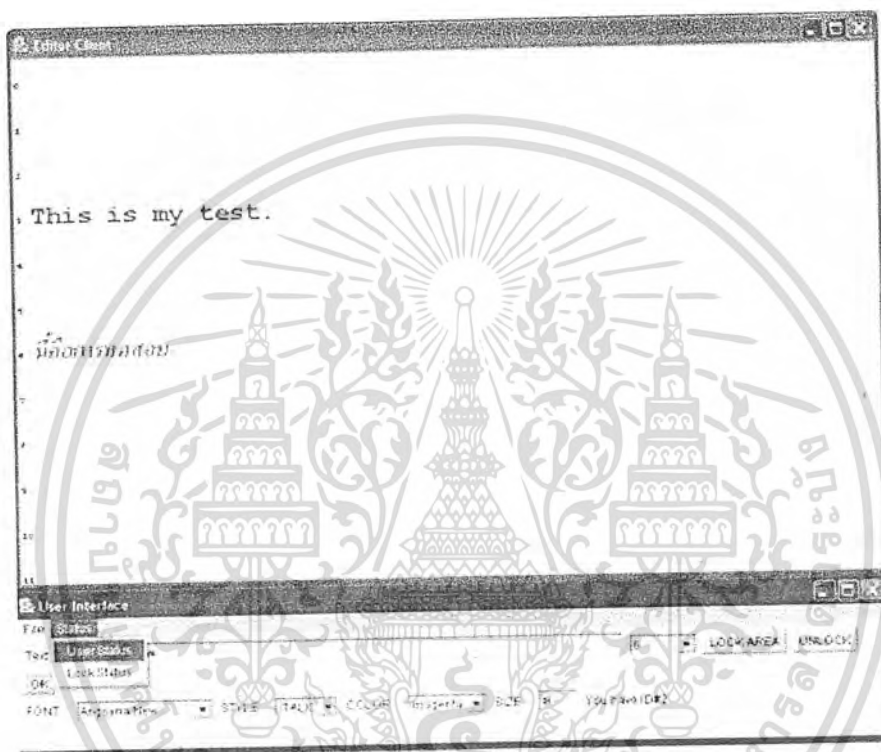
รูปที่ 7.11 ทำการยกเลิกบรรทัดที่ล็อกไว้



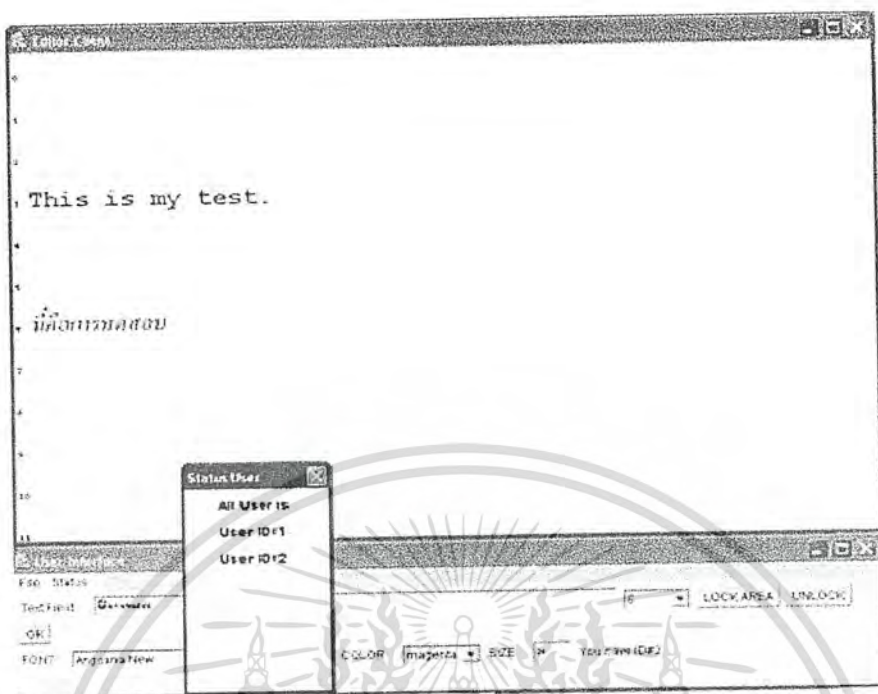
รูปที่ 7.12 ใส่ข้อความในบรรทัดที่ 6

7.4 ตัวอย่างการทำงานร่วมกันในการแสดงผลสถานะผู้ใช้

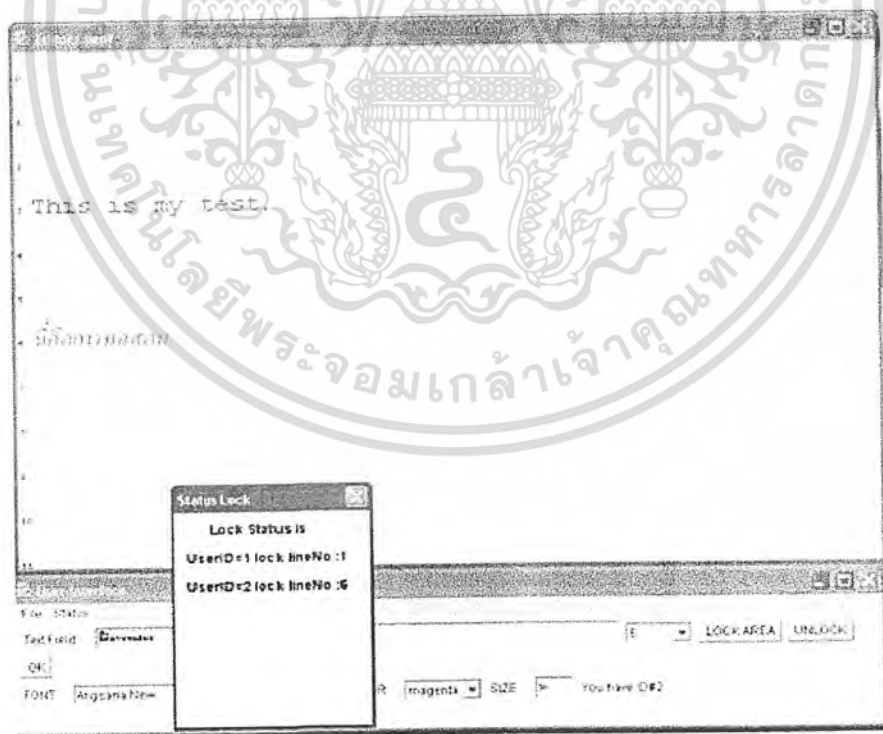
สถานะผู้ใช้งานจะให้ข้อมูลเกี่ยวกับผู้ใช้ทุกคนที่ทำงานเอกสารเดียวกัน เพื่อเป็นประโยชน์ในการแก้ไขปรับปรุงเอกสาร



รูปที่ 7.13 สถานะผู้ใช้ประกอบด้วย ข้อมูลผู้ใช้และข้อมูลการล็อก

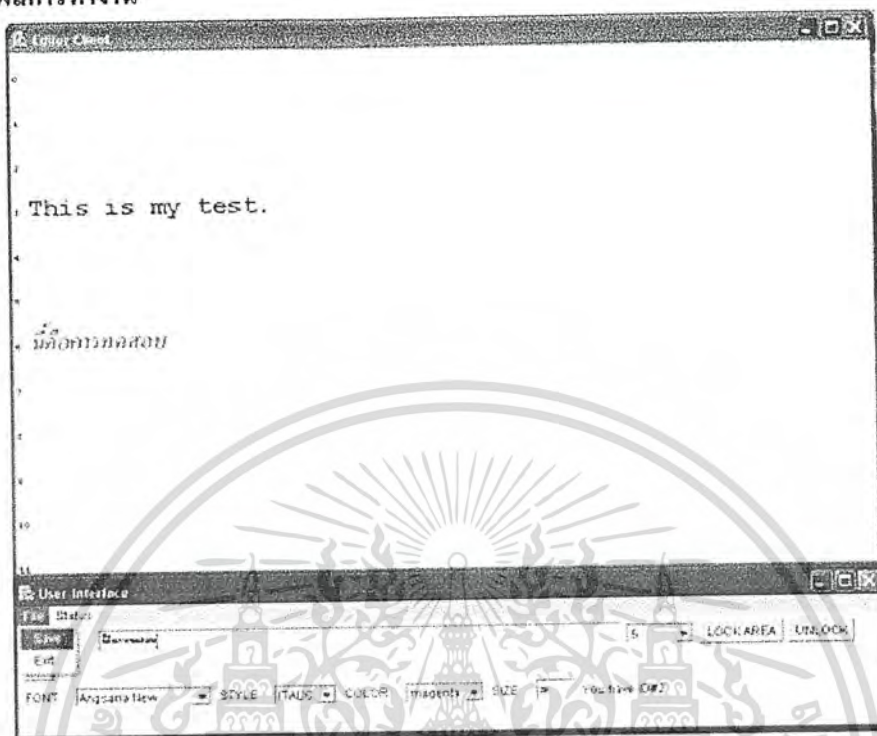


รูปที่ 7.14 แสดงว่ามีผู้ใช้อยู่ 2 คน

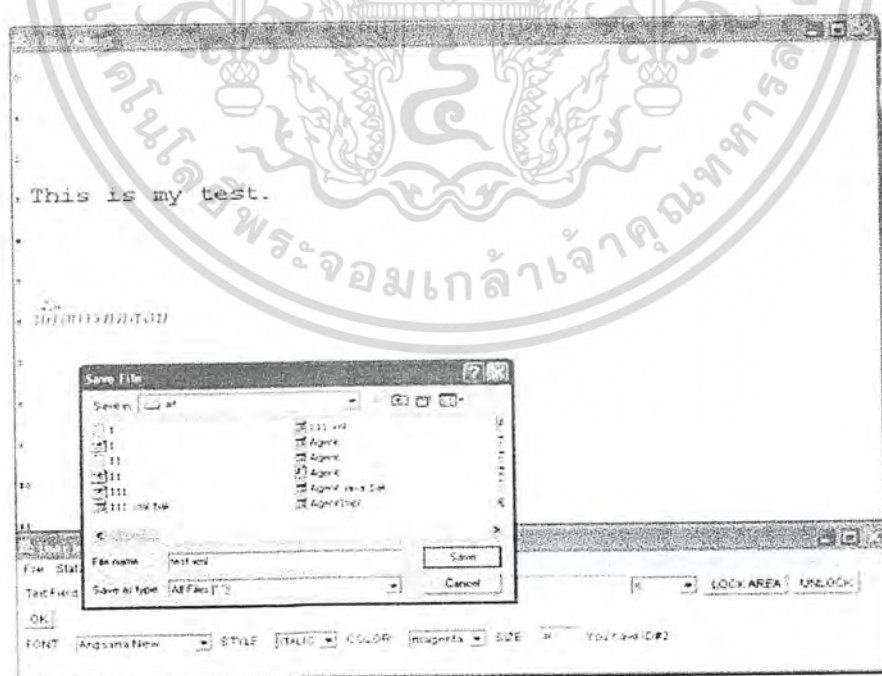


รูปที่ 7.15 แสดงว่าผู้ใช้คนแรกล็อกบรรทัดที่ 1 อยู่ และเราล็อกบรรทัดที่ 6

7.5 การเซฟผลการทำงาน



รูปที่ 7.16 ทำการเซฟผลลัพธ์ที่ได้จากเอกสาร



รูปที่ 7.17 ใส่ชื่อไฟล์และนามสกุล .xml เพื่อทำการเซฟ



This is my test.

ประเทศไทย



บทที่ 8

บทสรุปและวิจารณ์

8.1 สรุปการใช้โปรแกรม

การพัฒนาเฟรมเวิร์กสำหรับการทำงานร่วมกันในการประพันธ์เอกสารนั้น เป็นไปตามหลักการการทำงานร่วมกันผ่านทางคอมพิวเตอร์เพราะเป็นการจัดทำเอกสารร่วมกัน ผ่านเครือข่ายทางอินเทอร์เน็ต ทำให้ผู้ใช้งานสามารถจัดทำเอกสารร่วมกันแม้ไม่ได้อยู่ที่เดียวกันได้ จะเป็นการประหยัดเวลาในการทำงานและลดค่าใช้จ่ายได้บางส่วน ที่กล่าวมานี้เป็นวัตถุประสงค์ของโครงการนี้ ในการพัฒนามีการแก้ไขปรับปรุงเพื่อให้ตอบสนองความต้องการของผู้ใช้และขจัดปัญหาที่เกิดขึ้น ซึ่งเป็นไปตามหลักการของการแก้ไขปรับปรุงการทำงานการทำงานร่วมกัน (Tailoring) หากแต่ยังเป็นการพัฒนาเบื้องต้น เพื่อให้เห็นแนวทางที่เป็นไปได้ในการทำงานจริงว่าสามารถนำคอมพิวเตอร์มาทำงานร่วมกันในการประพันธ์เอกสารได้จริง ปัญหาต่างๆยังคงมีอยู่ เช่น การจัดการกับการทำงานที่มีผู้ใช้หลายคน (Multi User) ซึ่งเป็นการจัดการที่ยุ่งยาก รวมถึงการจัดการกับรูปแบบของเอกสารที่ยังคงไม่เป็นอิสระ หากได้รับการนำแนวความคิดและพัฒนาต่อเพื่อให้มีประสิทธิภาพเพิ่มมากขึ้น ซึ่งเป็นสิ่งที่ท้าทายอย่างมากในการพัฒนาโครงการนี้ให้ประสบผลสำเร็จ แต่ก็ต้องอาศัยเวลาในการศึกษาและพัฒนาอีกมาก เพื่อสามารถตอบสนองความต้องการของผู้ใช้ได้ในทุกๆด้าน

8.2 ปัญหาที่พบและยังไม่ได้แก้ไข

ปัญหาที่ยังไม่ได้แก้ไขหรืออาจกล่าวได้ว่าเป็นข้อเสียของการพัฒนาเฟรมเวิร์กสำหรับการทำงานร่วมกันในการประพันธ์เอกสารมีดังต่อไปนี้คือ

1. ยังไม่สามารถแสดงรูปภาพในหน้าจอส่วนแสดงผลได้ เนื่องจากส่วนแสดงผลต้องมีการกำหนดค่าของตัวอักษรแต่ละบรรทัดให้ตายตัว หากมีการแสดงผลรูปภาพด้วยจะทำให้ตำแหน่งข้อความคลาดเคลื่อนไป
2. หน้าจอแสดงผลยังมีการกระพริบอยู่อันเนื่องมาจากต้องมีการรีเฟรชทุกๆ 5 วินาที ให้มีการแสดงผลข้อความใหม่ทำให้หน้าตาที่เป็นส่วนป้อนข้อมูลจะเกิดอินแอคทีฟ(Inactive) ทำให้การป้อนข้อความเกิดความลำบากเนื่องจากต้องคอยคลิกที่ส่วนป้อนข้อความอยู่เสมอ
3. ตัวเอดิเตอร์ยังสามารถจัดการกับเอกสารได้เพียงหน้าเดียวเท่านั้น
4. ทุกครั้งที่ป้อนข้อความในเท็กซ์ฟิลด์ต้องมีการกด เ็นเทอร์ ทุกครั้ง มิฉะนั้นจะไม่สามารถแสดงผลได้อย่างถูกต้อง

8.3 ส่วนที่ปรับปรุงและพัฒนา

1. การจัดการเกี่ยวกับความขัดแย้ง (Conflict) : เราได้พัฒนาต่อมากมาย โดยของเดิมมัน จะมีการแจ้งให้ผู้ใช้ทราบว่าเกิดความขัดแย้งในการทำงานอย่างไร ซึ่งยังไม่ดีเพียงพอในการทำงานร่วมกัน เราจึงสร้างแนวทางแก้ไขที่สามารถจัดปัญหาความขัดแย้ง ดังนี้ สามารถให้ผู้ใช้ของพื้นที่ในส่วนที่ต้องทำงานในการแก้ไขเอกสาร ถ้ามีผู้ใช้ของพื้นที่เดียวกันมากกว่าหนึ่ง ผู้ใช้ที่ทำการร้องขอการจองก่อนจะได้รับอนุญาตกระทำกับเอกสารส่วนนี้ก่อน ดังนั้นผู้ใช้คนอื่นจะไม่สามารถมากระทำกับเอกสารส่วนนี้ได้เลย จนเซิร์ฟเวอร์จะอนุญาต แนวทางจะช่วยลดปัญหาความขัดแย้งให้หมดไปอย่างสิ้นเชิง

2. การจัดการเกี่ยวกับการพูดคุย (Chat) นอกเหนือจากหน้าจอหลักของการทำงาน การพัฒนาให้มีการพูดคุย (Chat) ได้จะทำให้การทำงานสะดวกขึ้นและลดปัญหาความขัดแย้งลงได้เพราะผู้ใช้งานมีการพูดคุยและสร้างความเข้าใจกันในการทำงาน อาจจะเป็นส่วนของรายละเอียดการทำเอกสาร, รูปแบบการทำเอกสาร และอื่นๆ อีกทั้งยังให้ความรู้สึกใกล้เคียงกับการทำงานพบปะมากกว่าการไม่มีการพูดคุย โดยในการพัฒนาที่ผ่านมาจะมีการแสดงรายชื่อของผู้ที่ใช้งาน (Online) เท่านั้น

3. การจัดการกับส่วนของการติดต่อกับผู้ใช้งาน (User Interface) : การพัฒนาทำได้โดยพัฒนาให้ส่วนที่ติดต่อกับผู้ใช้เป็นการติดต่อที่เป็นมิตรกับผู้ใช้ (User friendly) นั่นคือการออกแบบและพัฒนาให้ง่ายต่อการใช้งาน จอการใ้ใช้งานที่สวยงามเพิ่มมากขึ้นแล้วก็จะทำให้ผู้ใช้ ไม่รู้สึกว่าการใช้งานผ่านทางคอมพิวเตอร์ในการทำเอกสารงานร่วมกันนั้นเป็นเรื่องที่ไม่ยุ่งยากอีกต่อไป และมีความน่าสนใจในการใช้งานเพิ่มมากขึ้น

4. การใช้เทคโนโลยีใหม่ มาช่วยในการพัฒนาโปรแกรม : โดยอาศัยความสามารถของภาษาที่มีประสิทธิภาพเพิ่มขึ้น มาพัฒนา โคลเอนด์ ไซด์ แอปพลิเคชัน (side application Client) เพื่อเพิ่มประสิทธิภาพในการแสดงผล (มีหลายๆ ฟอนต์ มีหลายขนาดให้เลือก มีหลายสี) โดยที่ไม่สามารถทำได้ใน Servlet

5. นำเทคโนโลยี RMI เข้ามาแทน Servlet โดยที่ Local Class สามารถเรียกใช้ Remote Class เพื่อช่วยเพิ่มความสามารถของแอปพลิเคชันให้ขนาดเล็กลง

8.4 ข้อเสนอแนะ

1. การพัฒนาให้ตัวเอเจนต์มีความสามารถมากขึ้น เป็นการพัฒนาให้เฟรมเวิร์คสำหรับการทำงานร่วมกันในการประพันธ์เอกสารมีความสามารถเป็นเอเจนต์ที่มีประสิทธิภาพเข้ามาช่วยในการทำงานของผู้ใช้ และมีการปรับปรุงความสามารถในการเรียนรู้จากผู้ใช้ได้ เช่น การกำหนดเวลาที่จะรีเฟรช (Refresh) หน้าจอแสดงผลส่วนกลาง โดยหากสังเกตเห็นว่าผู้ใช้งานในกลุ่มทุกคนมีการกระทำที่ซ้ำกันไปทำให้ต้องแสดงผลเดิมบ่อยครั้ง ก็จะเปลี่ยนค่าการรีเฟรชหน้าจอให้มากขึ้น เป็นต้น แต่แนวทางนี้ก็จะเป็นแนวทางที่พัฒนาอยู่เพราะเป็นการทำให้เอเจนต์ฉลาด และมีการทำงานที่เป็นอัตโนมัติ ซึ่งเป็นเรื่องที่ยุ่งยากและซับซ้อน

2. สร้างแอปพลิเคชันให้มีความสามารถในการทำงานที่เป็น Word เต็มตัว โดยมีการแก้ไขเอกสารเป็นแบบ Real Time ซึ่งจะช่วยอำนวยความสะดวกในการสร้างเอกสารร่วมกันได้อย่างรวดเร็ว

3. ทำเป็นเว็บ โดยสามารถสร้างเอกสารแบบออนไลน์ ซึ่งจะก่อให้เกิดความแพร่หลายได้อย่างรวดเร็ว
ใน WWW(World Wide Web) บนอินเทอร์เน็ต



บรรณานุกรม

- [1] Elliotte Rusty Harold (2000): “Java Network Programming second edition”, O'Reilly Press.
- [2] Marko Boger. (2001) : “Java in distributed systems”, University of Hamburg, John Wiley, Sons, Ltd.
- [3] Wolfgang Emmerich. (2000): "*Engineering Distributed Objects*", London University, John Wiley & Sons, Ltd.
- [4] ดร.วีระศักดิ์ ชิงฉาว : “Fundamental of Java Programming Volume 1”, SUM Publishing Department ,SUM System Company Limited , 2541
- [5] Michael J.Young : “Step by Step XML “,สำนักพิมพ์สยามบ้าน.com , 2000

