

บราวเซอร์ที่สามารถโปรแกรมได้โดยใช้ภาษาไจธอน  
PROGRAMMABLE BROWSER USING JYTHON



นายณวัฒน์ แก้วคำ  
นายบุญทวี สันติศรีวารณ์

5

เลขหมู่.....  
เลขทะเบียน..... 46170  
วัน, เดือน, ปี 20 ส.ค. 2546

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

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

1111111111

บราวเซอร์ที่สามารถโปรแกรมได้โดยใช้ภาษาไจธอน  
PROGRAMMABLE BROWSER USING JYTHON



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2544

ปริญญาโทปีการศึกษา 2544

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง บราวเซอร์ที่สามารถโปรแกรมได้โดยใช้ภาษาไพธอน

Programmable Browser using Jython

ผู้จัดทำ

1. นายชนวัฒน์ แก้วคำ รหัสประจำตัว 41014172
2. นายบุญทวี สันติศรีวารินทร์ รหัสประจำตัว 41014234



..... อาจารย์ที่ปรึกษา

(ดร. วิศิษฎ์ หิรัญกิตติ)

## บราวเซอร์ที่สามารถโปรแกรมได้โดยใช้ภาษาไจซอน

นายชนวัฒน์ แก้วคำ	รหัส 41014172
นายบุญทวี สันติศรีวารณณ์	รหัส 410141234
ดร. วิศิษฐ์ ธีรฤกิตติ	อาจารย์ที่ปรึกษา
	ปีการศึกษา 2544

### บทคัดย่อ

ปฏิญานิพนธ์นี้นำเสนอบราวเซอร์ที่สามารถโปรแกรมได้ ซึ่งเป็นแนวคิดที่นำไปสู่การสร้างเอนจินต์ที่ชาญฉลาด บราวเซอร์ที่สามารถโปรแกรมได้สามารถติดต่อสื่อสารกับโปรโตคอลต่างๆบนอินเทอร์เน็ตแบบกึ่งอัตโนมัติ ซึ่งลำดับการติดต่อสื่อสารได้ถูกโปรแกรมไว้ล่วงหน้าโดยผู้ใช้งานทางคำสั่งภาษาสคริปต์ ตัวอย่างเช่น สั่งให้เปิดเว็บบราวเซอร์หลายๆตัวขึ้นมาทำงานพร้อมกันตามยูอาร์แอลที่กำหนด เรียกใช้บริการเว็บเซอร์วิสต่างๆ เช่น การส่งข้อความสั้นไปยังโทรศัพท์มือถือ มากไปกว่านั้นเอนจินต์ยังสามารถทำงานตอบสนองต่อเหตุการณ์ที่ผู้ใช้งานกำหนด เช่น เมื่อมีอีเมลใหม่เข้ามาในเมลบ็อกซ์ให้แจ้งผู้ใช้ที่หน้าจอคอมพิวเตอร์ สำหรับการสั่งงานเอนจินต์นั้นผู้ใช้งานสามารถป้อนความรู้เข้าสู่ฐานความรู้ของเอนจินต์ในรูปแบบของการทำงานตามลำดับและในรูปแบบของเหตุการณ์ → การทำงาน

คำสั่งภาษาสคริปต์ถูกออกแบบให้ใช้ไวยากรณ์เช่นเดียวกับภาษาไจซอนซึ่งเป็นภาษาที่ง่ายและยืดหยุ่นเหมาะสำหรับนำมาใช้พัฒนาเอนจินต์ จึงทำให้ภาษาสคริปต์ที่ใช้สั่งงานเอนจินต์มีความสามารถเช่นเดียวกับภาษาไจซอน นอกจากนี้ยังได้เพิ่มเติมไวยากรณ์ที่ใช้ในการรันคำสั่งพร้อมๆกันแบบขนาน และไวยากรณ์ที่ใช้เพิ่มความรู้อื่นในรูปแบบของเหตุการณ์ → การทำงาน ซึ่งเป็นภาษาตรรกศาสตร์ ให้กับเอนจินต์เพื่อให้สามารถโต้ตอบกับเหตุการณ์ต่างๆได้

## PROGRAMMABLE BROWSER USING JYTHON

Thanawat Kaewka	41014172
Boontawee Suntisrivaraporn	41014234
Dr. Visit Hirankitti	Advisor

### ABSTRACT

This project proposes Programmable Browser which can lead to a development of an Intelligent agent. Programmable Browser can communicate through the internet semiautomatically, once a communication sequence has been programmed by the user via its scripting language. for example, to open multiple browsers at specific URLs, to enquire web services such as sending short messages to cellular phone, etc. Moreover, the agent can response to the events such as informing the user when a new mail has arrived. To program agent, user can assert new knowledge in the form of scripts and knowledge in the form of Event  $\rightarrow$  Action.

The scripting language's syntax of this agent is same as that of Jython language, which is a language that is very flexible and easy for developing agent. In addition, we also extend the syntax to facilitate parallel script and event  $\rightarrow$  action script, the latter is based on the language of logic.

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้คงจะไม่สามารถสำเร็จลุล่วงไปได้ ถ้าไม่ได้รับความช่วยเหลือจากบุคคลหลายท่าน บุคคลแรกที่ต้องกล่าวถึง และเป็นบุคคลที่มีส่วนสำคัญต่อปริญญาบัตรนี้มากที่สุด คือ ดร. วิศิษฎ์ หิรัญกิตติ อาจารย์ที่ปรึกษาปริญญาบัตร ซึ่งอาจารย์ได้ให้คำปรึกษา แนะนำ และเอาใจใส่ในการทำปริญญาบัตรนี้อย่างสม่ำเสมอ หลายครั้งที่อาจารย์มีธุระมากมายแต่ก็ยังเสียสละเวลามาก และแนะนำการทำงาน ต้องขอขอบพระคุณเป็นอย่างสูง

นอกจากนี้ต้องขอขอบคุณอาจารย์ทุกๆท่าน หนังสือทุกๆเล่มที่ได้ประสิทธิ์ประสาทความรู้ และให้แนวคิดต่างๆแก่ผู้จัดทำ รวมทั้งเพื่อนๆทุกคนที่เป็นเพื่อนคุย เพื่อนเล่น ให้คำแนะนำ และช่วยเหลือในสิ่งต่างๆด้วยดีเสมอมา

และสุดท้ายต้องขอขอบพระคุณบุคคลที่มีความสำคัญต่อผู้จัดทำมากที่สุดที่ทำให้ผู้จัดทำวันนี้ได้ คือ บิดา และมารดาอันเป็นที่เคารพ และรักยิ่งซึ่งได้อบรมเลี้ยงดูมาเป็นอย่างดี พร้อมทั้งให้โอกาสทางการศึกษาที่ดีที่สุด และคอยให้กำลังใจห่วงใยเสมอมา ผู้จัดทำระลึกในพระคุณ และขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ธนวัฒน์ แก้วคำ  
บุญทวี สันติศรีวารินทร์

## สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูปภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ที่มาของโครงการ	1
1.2 วัตถุประสงค์	1
1.3 เป้าหมาย	2
1.4 ขอบเขตของโครงการ	2
1.5 วิธีการดำเนินงาน	2
บทที่ 2 ศึกษาและวิจารณ์ข้อมูลเก่า	3
2.1 โปรแกรมเว็บเบราว์เซอร์ (Web Browser) ทั่วไปในปัจจุบัน	3
2.2 เบราวเซอร์ที่สามารถโปรแกรมได้ (Programmable Browser)	3
2.3 เบราวเซอร์ที่สามารถโปรแกรมได้เวอร์ชันภาษาไพธอน	4
บทที่ 3 หลักการพื้นฐาน	5
3.1 หลักการพื้นฐานของเอเจนต์	5
3.1.1 ความหมายของเอเจนต์	5
3.1.2 การแบ่งประเภทของเอเจนต์	5
3.1.3 ลำดับชั้นของเอเจนต์ (Agents Hierarchy)	6
3.1.4 เอเจนต์ที่ชาญฉลาดด้วยภาษาไพธอน	7
3.2 หลักการพื้นฐานของเว็บเบราว์เซอร์	8
3.2.1 โปรแกรมเว็บเบราว์เซอร์	8
3.2.2 หลักการทำงานของโปรแกรมเว็บเบราว์เซอร์	9
3.3 หลักการพื้นฐานของอินเทอร์เน็ต	10
3.3.1 ที่มาและความหมายของอินเทอร์เน็ต และเว็บ	10
3.3.2 โพรโทคอลที่ซีพีไอพี (TCP/IP Protocol)	10
3.3.3 ยูอาร์แอล (URL)	13
3.3.4 การเขียนโปรแกรมเน็ตเวิร์คโดยภาษาไพธอน	14
3.4 เอชทีทีพี (HTTP)	16

## สารบัญ (ต่อ)

	หน้าที่
3.4.1 โครงสร้างการร้องขอเอชทีทีพี (HTTP Requests)	17
3.4.2 โครงสร้างการตอบรับเอชทีทีพี (HTTP Response)	18
3.5 ซีจีไอ (Common Gateway Interface)	19
3.6 เอสเอ็มทีพีโปรโตคอล (Simple Mail Transfer Protocol)	20
3.6.1 สถาปัตยกรรมระบบเมลล์	20
3.6.2 ยูนิกซ์เมลล์	21
3.6.3 รูปแบบจดหมาย	22
3.6.4 ป๊อป และไอแมป (POP and IMAP Protocol)	22
3.7 โปรโตคอลเอฟทีพี (File Transfer Protocol)	23
บทที่ 4 ภาษาไพธอน	26
4.1 คุณสมบัติของภาษาไพธอน	26
4.2 ชนิดของข้อมูล	28
4.3 ตัวกระทำ (Operators)	30
4.4 คำสงวน (Reserved words)	32
4.5 สโคปของโค้ด (Code block)	32
4.6 คอมเมนต์ (Comment)	33
4.7 การพิมพ์ statement เดียวในหลายบรรทัด (Continuation)	33
4.8 การพิมพ์หลาย statement ในบรรทัดเดียว	33
4.9 การกำหนดค่าให้ตัวแปร (Variable assignment)	33
4.9.1 การกำหนดค่าหนึ่งค่าให้กับหลายตัวแปร (Multiple assignment)	33
4.9.2 การกำหนดค่าหลายค่าให้กับหลายตัวแปร	34
4.10 คำสั่งลูป (Control flow)	34
4.10.1 คำสั่ง IF	34
4.10.2 คำสั่ง WHILE	35
4.10.3 คำสั่ง FOR	35
4.11 การจัดการหน่วยความจำ (Memory management)	36
4.12 การใช้งานภาษาไพธอน	37
4.13 ภาษาไพธอน (Jython : Python in Java)	39
บทที่ 5 หลักการของบราวเซอร์ที่สามารถโปรแกรมได้	41
5.1 ส่วนติดต่อกับผู้ใช้ (Graphical User Interface)	44
5.2 โพรไฟล์ (Profile)	47

## สารบัญ (ต่อ)

	หน้าที่
5.3 บีลีฟเซต (Belief set)	49
5.3.1 เนม (Name)	50
5.3.2 โปรแกรม (Program)	50
5.3.3 อีเวนต์ – แอ็กชัน (Event - action)	51
5.4 ส่วนจัดการอีเวนต์ (Event handler)	53
5.4.1 ตัวจัดการไทม์อีเวนต์ (Time-Event Handler)	53
5.4.2 ตัวจัดการเมลอีเวนต์ (Mail-Event Handler)	54
5.4.3 ตัวจัดการอีเวนต์อื่นๆ (Other-Events Handler)	54
5.5 อินเทอร์พรีเตอร์แบบมัลติเธรด (Multithreading interpreter)	54
5.6 โมดูลที่ใช้ติดต่อผ่านอินเทอร์เน็ต (Communication module)	55
5.6.1 เว็บเบราว์เซอร์ (Web browser)	55
5.6.2 เอฟทีพีไคลเอนต์ (FTP Client)	57
5.6.3 เมลไคลเอนต์ (Mail Client)	57
5.6.4 เอสเอ็มเอสไคลเอนต์ (SMS Client)	60
5.7 โมดูลเวลา	62
5.8 ระบบช่วยเหลือผู้ใช้งาน (Intelligent Script Editor)	64
บทที่ 6 การทดลองใช้โปรแกรม	65
บทที่ 7 บทสรุป	72
7.1 สรุป	72
7.2 วิจัย	73
ภาคผนวก คู่มือการใช้งานโปรแกรมเบราว์เซอร์ที่สามารถโปรแกรมได้ เวอร์ชัน 2.1	74
บรรณานุกรม	91

## สารบัญตาราง

	หน้าที่
ตารางที่ 3-1 แสดงคุณสมบัติต่างๆ ของเอเจนต์	6
ตารางที่ 3-2 TCP/IP Layers และตัวอย่าง โปรโตคอล	11
ตารางที่ 3-3 องค์ประกอบของยูอาร์แอล (URL Component)	14
ตารางที่ 3-4 เลขที่ที่พีเอ็มธอด (HTTP method)	18
ตารางที่ 3-5 สถานะการตอบรับ ( HTTP Status Code Categories)	19
ตารางที่ 4-1 ตารางเปรียบเทียบคุณสมบัติต่างๆ ของภาษาไพธอนกับภาษาอื่นๆ	28
ตารางที่ 5.1 ตารางแสดง PWB Built in Object	50
ตารางที่ 5.2 ตารางแสดง PWB Built in Function	51
ตารางที่ 5.3 ตารางแสดง PWB Built in Class	51



## สารบัญรูปภาพ

	หน้าที่
รูปที่ 3-1 รูปแสดงลำดับชั้นของเอเจนต์	7
รูปที่ 3-2 รูปแสดงการทำงานของเว็บเบราว์เซอร์	10
รูปที่ 3-3 รูปแสดง TCP/IP protocol for WWW	11
รูปที่ 3-4 รูปแสดงการติดต่อถึงชั้นแอปพลิเคชัน โดยใช้พอร์ต	13
รูปที่ 3-5 รูปแสดงโครงสร้างของยูอาร์แอล (URL)	13
รูปที่ 3-6 รูปแสดงโครงสร้างของ HTTP Request	17
รูปที่ 3-7 รูปแสดงโครงสร้างของ HTTP Response	18
รูปที่ 3.8 รูปแสดงการร้องขอข้อมูลผ่านซีจีไอ	20
รูปที่ 3.9 รูปแสดงสถาปัตยกรรมทั่วไปของระบบเมลล์ในทีซีพี/ไอพี	20
รูปที่ 3.10 รูปแสดงรูปแบบของยูนิคซ์เมลล์	22
รูปที่ 3.11 รูปแสดงแบบจำลองการทำงานของเอฟทีพี	24
รูปที่ 3.12 รูปแสดงการสร้างส่วนเชื่อมโยงข้อมูล	25
รูปที่ 3-1 แสดงการทำงานของเอเจนต์ในโหมดสคริปต์ (Script Mode)	19
รูปที่ 3-2 แสดงการทำงานของเอเจนต์ในโหมดอีเวนต์ (Event Mode)	19
รูปที่ 4-1 รูปแสดงโปรแกรมไพธอน IDLE	38
รูปที่ 4-2 รูปแสดงโปรแกรมไพธอนคอมมานด์ไลน์	38
รูปที่ 4-3 รูปแสดง โปรแกรม pythonwin	39
รูปที่ 4-4 รูปแสดงโปรแกรมไจรอนคอนโซล	40
รูปที่ 5.1 รูปแสดงโครงสร้างทั้งหมดของเว็บเอเจนต์	42
รูปที่ 5.2 รูปแสดงหน้าต่างแก้ไขไพรไฟล์ของผู้ใช้	45
รูปที่ 5.3 รูปแสดงหน้าต่างเขียนโปรแกรม	46
รูปที่ 5.4 รูปแสดงหน้าต่างรายการอีเวนต์	46
รูปที่ 5.5 รูปแสดงหน้าต่างโต้ตอบและแสดงผล	47
รูปที่ 5.6 รูปแสดงโครงสร้างของออบเจกต์ “my”	48
รูปที่ 5.7 รูปแสดงการใช้งาน “my” ออบเจกต์บนหน้าต่างเขียนสคริปต์	49
รูปที่ 5.8 รูปแสดงการใช้งาน “my” ออบเจกต์บน input dialog อื่นๆ	49
รูปที่ 5.9 รูปแสดงการเปลี่ยนสถานะของรายการอีเวนต์-แอ็กชัน	53
รูปที่ 5.10 รูปแสดงโครงสร้างของเว็บเบราว์เซอร์	56
รูปที่ 5.11 รูปแสดงหน้าต่างของเว็บเบราว์เซอร์	56
รูปที่ 5.12 รูปแสดงหน้าต่างของโปรแกรมเอฟทีพี	57
รูปที่ 5.13 รูปแสดงหน้าต่างโปรแกรมเมลล์ (รับเมลล์)	59
รูปที่ 5.14 รูปแสดงหน้าต่างโปรแกรมเมลล์ (ส่งเมลล์)	60

## สารบัญรูปภาพ

	หน้าที่
รูปที่ 5.15 รูปแสดงเว็บเพจที่ใช้ส่งข้อความสั้น	61
รูปที่ 5.16 รูปแสดงหน้าต่างของ โปรแกรมส่งข้อความมือถือ	62
รูปที่ 5.17 รูปแสดงการทำงานของตัวตั้งเวลา	63
รูปที่ 5.18 รูปแสดงระบบช่วยเหลือผู้ใช้งาน	64
รูปที่ 6.1 ตัวอย่างสกริปต์ที่ 1	65
รูปที่ 6.2 ผลลัพธ์จากตัวอย่างที่ 1	65
รูปที่ 6.3 ตัวอย่างสกริปต์ที่ 2	66
รูปที่ 6.4 ผลลัพธ์จากตัวอย่างที่ 2	66
รูปที่ 6.5 ตัวอย่างสกริปต์ที่ 3	67
รูปที่ 6.6 ผลลัพธ์จากตัวอย่างที่ 3	67
รูปที่ 6.7 ตัวอย่างสกริปต์ที่ 4	68
รูปที่ 6.8 ผลลัพธ์จากตัวอย่างที่ 4	68
รูปที่ 6.9 ตัวอย่างสกริปต์ที่ 5	69
รูปที่ 6.10 ผลลัพธ์จากตัวอย่างที่ 5	69
รูปที่ 6.11 ตัวอย่างสกริปต์ที่ 6	70
รูปที่ 6.12 ผลลัพธ์จากตัวอย่างที่ 6	70
รูปที่ 6.13 ตัวอย่างสกริปต์ที่ 7	71
รูปที่ 6.14 ผลลัพธ์จากตัวอย่างที่ 7	71

# บทที่ 1

## บทนำ

### 1.1 ที่มาของโครงการ

ในปัจจุบันเครือข่ายอินเทอร์เน็ตได้ทวีความสำคัญต่อการดำรงชีวิตของมนุษย์อย่างชัดเจนเนื่องจากเป็นแหล่งในการเก็บรวบรวมข้อมูล และข่าวสารต่างมากมาย การให้บริการข้อมูลบนเครือข่ายอินเทอร์เน็ตได้มีการนำเสนอในรูปแบบต่างๆกันออกไป เช่น บริการข้อมูลเพื่อการศึกษา โดยเฉพาะบริการข้อมูลเพื่อประโยชน์ในการประชาสัมพันธ์ รวมทั้งบริการข้อมูลเพื่อประโยชน์ในเชิงธุรกิจด้วย การบริการข้อมูลจึงเป็นประโยชน์ และได้รับความสนใจจากผู้ใช้ในทุกระดับชั้น จึงทำให้ขนาดของแหล่งเก็บข้อมูลใหญ่ขึ้นเพื่อการครอบคลุมการบริการ และสามารถตอบสนองความต้องการในการใช้บริการข้อมูลซึ่งแตกต่างกันไปได้อย่างทั่วถึง

ผู้ใช้สามารถใช้บริการเว็บเบราว์เซอร์ (Web Browser) โดยการกรอกยูอาร์แอล (URL) ที่ผู้ใช้งานต้องการให้แก่เบราว์เซอร์ ซึ่งเบราว์เซอร์จะทำการติดต่อกับเซิร์ฟเวอร์ (Server) เพื่อค้นหาเว็บเพจ (Web Page) ที่ต้องการ และสุดท้ายเบราว์เซอร์จะแสดงเว็บเพจเหล่านั้นให้ผู้ใช้งานได้เห็น

จากการทำงานของเบราว์เซอร์ในลักษณะข้างต้น จะเห็นว่าการใช้งานเบราว์เซอร์นั้นเหมาะที่จะใช้ในงานที่ผู้ใช้งานต้องสามารถรับ และตอบสนองบริการซึ่งแสดงออกทางเบราว์เซอร์ได้ จึงเป็นเรื่องที่ไม่สะดวกนักถ้าผู้ใช้งานมีเวลาน้อย แต่ต้องการเรียกใช้บริการทางเครือข่ายหลายอย่างและ ต้องรอรับการให้บริการเป็นเวลานาน

ทางคณะผู้จัดทำโครงการ จึงพัฒนาเว็บเบราว์เซอร์ให้มีความสามารถมากขึ้น คือทำให้เบราว์เซอร์เป็นเบราว์เซอร์ที่สามารถถูกโปรแกรมได้โดยผู้ใช้งาน (Programmable Web Browser) โดยมีการนำเทคโนโลยีเอเจนต์ (Agent Technology) มาใช้เพื่ออำนวยความสะดวกให้แก่ผู้ใช้งาน ผู้ใช้งานสามารถสั่งงานผ่านทางเอเจนต์ได้โดยที่เอเจนต์จะทำหน้าที่ควบคุมการทำงานของเบราว์เซอร์อีกทีหนึ่ง และนอกจากเว็บเบราว์เซอร์แล้วก็ควรจะมีความสามารถในการใช้งานอินเทอร์เน็ตโปรโตคอลอื่นๆอีกด้วย เช่น บริการโอนถ่ายไฟล์ (File Transfer Protocol) และบริการอีเมล

### 1.2 วัตถุประสงค์

1. เพื่อพัฒนาเบราว์เซอร์ที่สามารถโปรแกรมได้ให้มีความสามารถมากขึ้นจากโครงการเก่าและแก้ไขข้อบกพร่อง
2. เพื่อศึกษาลักษณะการทำงานของเอเจนต์ และประยุกต์การใช้งาน
3. เพื่อศึกษาและประยุกต์การทำงานของเว็บเบราว์เซอร์ และ โปรโตคอลอื่นๆ

### 1.3 เป้าหมาย

1. เพื่อสร้างโปรแกรมเอเจนต์ที่มีฐานความรู้เป็นของตนเอง และสามารถถูกโปรแกรมโดยผู้ใช้ได้
2. เพื่อสร้างโปรแกรมประยุกต์ที่ใช้ติดต่อสื่อสารบนเครือข่ายอินเทอร์เน็ต
3. โปรแกรมเอเจนต์สามารถควบคุมโปรแกรมประยุกต์ที่ใช้ติดต่อสื่อสารบนเครือข่ายอินเทอร์เน็ตได้ รวมทั้งมีฟังก์ชันที่สามารถทำงาน และอำนวยความสะดวกให้กับผู้ใช้

### 1.4 ขอบเขตของโครงการงาน

1. สร้างโปรแกรมเอเจนต์ต้นแบบที่สามารถทำงานได้โดยผู้ใช้ และมีฐานความรู้เป็นของตนเอง
2. สร้างบราวเซอร์ต้นแบบที่ทำงานได้ใกล้เคียงกับบราวเซอร์ที่มีอยู่ในปัจจุบัน
3. สร้างโปรแกรมประยุกต์อื่นๆ ที่เป็นที่นิยมใช้บนเครือข่ายอินเทอร์เน็ต เช่น โปรแกรมโอนถ่ายไฟล์ และโปรแกรมอีเมลล์
4. ผู้ใช้สามารถสั่งงานบราวเซอร์และโปรแกรมประยุกต์ได้ด้วยภาษาสคริปต์ผ่านทางโปรแกรม เอเจนต์
5. เอเจนต์สามารถพัฒนาต่อให้มีความสามารถมากขึ้นได้

### 1.5 วิธีการดำเนินงาน

การทำงานแบ่งออกเป็น 3 ส่วน ดังนี้

1. รวบรวมและศึกษาข้อมูลที่สำคัญ ดังต่อไปนี้
  - 1.1 ลักษณะทั่วไปของเอเจนต์
  - 1.2 หลักการทำงานที่สำคัญของเอเจนต์
  - 1.3 ข้อมูลทั่วไปของเอเจนต์
  - 1.4 ข้อมูลทั่วไปของบราวเซอร์
  - 1.5 หลักการทำงานของบราวเซอร์
  - 1.6 หลักการทำงานของซีจีไอ
  - 1.7 หลักการทำงานของเอฟทีพี
  - 1.8 หลักการทำงานของป๊อป 3
  - 1.9 หลักการทำงานของเอสเอ็มทีพี
  - 1.10 ภาษาไพธอน
  - 1.11 ภาษาจาวา
  - 1.12 ภาษาไจซอน
2. วิเคราะห์ข้อมูลที่ได้รวบรวมมาทั้งหมด เพื่อกำหนดขอบเขตและลักษณะของชิ้นงาน
3. พัฒนาโครงการงาน

## บทที่ 2

### ศึกษาและวิจารณ์ข้อมูลเก่า (Survey)

#### 2.1 โปรแกรมเว็บเบราว์เซอร์ (Web Browser) ทั่วไปในปัจจุบัน

ในปัจจุบันได้มีบริษัทผู้ผลิตหลายรายที่ผลิตเว็บเบราว์เซอร์ ออกสู่ท้องตลาด เช่น Aweb II, Hot Java, Lynx, Netscape และที่โด่งดังและได้รับความนิยมในการใช้งานเป็นอย่างมากคือ Internet Explorer แต่โปรแกรมเบราว์เซอร์ดังกล่าวยังไม่ได้ตอบสนองความต้องการของผู้ใช้อย่างเต็มที่

*ข้อบกพร่อง*

1. ผู้ใช้งานต้องคอยโต้ตอบกับเว็บเบราว์เซอร์ในการใช้งานอยู่ตลอด ไม่สามารถสั่งให้เว็บเบราว์เซอร์ทำงานอย่างอัตโนมัติได้
2. การใช้บริการบางอย่าง เช่น ส่งอีเมล หรือส่งข้อความสั้นต้องเสียเวลาในการลิงก์และโหลดเว็บเพจหลายครั้งเพื่อที่จะไปกรอกแบบฟอร์ม
3. ในกรณีที่ผู้ใช้ต้องการเปิดเว็บเพจหลายๆเพจพร้อมกันผู้ใช้ต้องเปิดเว็บเบราว์เซอร์ขึ้นมาทีละหน้าต่าง และกรอกยูอาร์แอลให้กับเว็บเบราว์เซอร์เรื่อยๆทีละหน้าต่าง

#### 2.2 เบราว์เซอร์ที่สามารถโปรแกรมได้ (Programmable Browser)

จากข้อบกพร่องของโปรแกรมเว็บเบราว์เซอร์ทั่วไปในปัจจุบันดังกล่าว จึงก่อให้เกิดโครงการนี้ขึ้นมา โครงการนี้เป็นปริญญาานิพนธ์ในปีการศึกษา 2543 โดยมีวัตถุประสงค์คือ เพื่อประยุกต์การทำงานของเอเจนต์เข้ากับการทำงานของเบราว์เซอร์ ทำให้เว็บเบราว์เซอร์มีความสามารถที่มากขึ้น โดยมีเป้าหมายคือ สร้างโปรแกรมเว็บเบราว์เซอร์ที่ผู้ใช้งานสามารถโปรแกรมได้ด้วยคำสั่งภาษาสคริปต์

*ลักษณะของโปรแกรมพอสังเขป*

**โปรแกรมเอเจนต์**

เป็นตัวแทนของผู้ใช้ที่ร้องขอบริการต่างๆ จากเซิร์ฟเวอร์ ผ่านคำสั่งสคริปต์ โดยมีความสามารถดังต่อไปนี้

- 1) เอเจนต์ทำงานแบบเรครด เพื่อการทำงานที่มีประสิทธิภาพและสนับสนุนคุณสมบัติของเอเจนต์ในเรื่องความเป็นอัตโนมัติ
- 2) เอเจนต์ทำงานที่เกี่ยวข้องกับเวลาได้ โดยสามารถตั้งเวลาการทำงานได้โดยใช้ภาษาสคริปต์

**การส่งและรับข่าวผ่าน NNTP**

เอเจนต์เป็นผู้ร้องขอบริการจากนิวส์เซิร์ฟเวอร์ (News Server) และนำข้อมูลข่าวสารที่ได้มาประมวลผล และยังสามารถส่งข้อมูลเหล่านี้ให้กับเอเจนต์ตัวอื่นๆโดยผ่านนิวส์เซิร์ฟเวอร์ (News Server) ได้อีกด้วย

## การทำงานผ่านโปรแกรมเทลเน็ต (Telnet)

ผู้ใช้งานสามารถส่งงานบราวเซอร์จากระยะไกลผ่านโปรแกรมเทลเน็ตได้ โดยโปรแกรมบราวเซอร์เมื่อเปิดขึ้นมาจะสร้างเซิร์ฟเวอร์เฉพาะแต่ละเครื่องทันที โดยทำหน้าที่จัดการการเชื่อมต่อจากเทลเน็ต (Telnet Handler)

## ภาษาสคริปต์

เป็นภาษาที่สร้างขึ้นมาจากโดยมีความสามารถพื้นฐาน if, if-else, while, repeat..until และมีคำสั่งที่ใช้ในการส่งงานบราวเซอร์ และจัดการเรื่องซีจีไอ

### ข้อเสีย

- โปรแกรมบราวเซอร์ไม่มีความสามารถที่เพียงพอจะนำไปใช้งานได้จริง เนื่องจากเป็นบราวเซอร์ที่ไม่สามารถแสดงสถานการณ์ทำงานของตนเอง ไม่สามารถท่องเว็บไปมาได้สะดวก (ไม่มี Navigator bar) ไม่สามารถจกจ่ายการยูอาร์แอลเก่า
- โปรแกรมไม่รองรับการทำงานของไฟล์ทรานสเฟอร์โปรโตคอล และอีเมล ซึ่งเป็นองค์ประกอบที่มีความสำคัญไม่น้อย และเป็นที่ยอมรับในปัจจุบัน
- โปรแกรมหลักเอเจนต์ ใช้งานได้ยาก
- ส่วนของภาษาสคริปต์ยังไม่มีความสมบูรณ์ มีความสามารถไม่มาก และไม่ค่อยมีความหลากหลาย
- ฟังก์ชันอำนวยความสะดวกมีไม่มากและชื่อฟังก์ชันยากต่อการจดจำ ยังไม่มีเครื่องมือช่วยอำนวยความสะดวกในการเขียนสคริปต์แก่ผู้ใช้ที่ไม่มีความรู้ด้านการเขียนโปรแกรม

## 2.3 บราวเซอร์ที่สามารถโปรแกรมได้เวอร์ชันภาษาไจธอน

### (Programmable Browser using Jython)

ทั้งข้อบกพร่องของโปรแกรมเว็บบราวเซอร์ต่างๆไป และของโปรแกรมเมเบิลบราวเซอร์ เป็นแรงผลักดันให้ผู้พัฒนาได้พัฒนาโปรแกรมเมเบิลบราวเซอร์เวอร์ชันใหม่ต่อเนื่องจากของเก่าออกมา โดยเป้าหมายของโครงการนี้คือ

- ✓ เว็บบราวเซอร์ต้องมีความสามารถใกล้เคียงถึงเทียบเท่าโปรแกรมเว็บบราวเซอร์ตามต่างๆไปในปัจจุบัน คือต้องมีความสามารถจัดการประวัติการใช้งาน สามารถจัดการรายการยูอาร์แอลโปรด และต้องสามารถรายงานสถานการณ์ทำงานได้อย่างถูกต้อง
- ✓ เพิ่มเติมความสามารถในเรื่อง ไฟล์ทรานสเฟอร์โปรโตคอล และอีเมล
- ✓ ออกแบบโปรแกรมหลักเอเจนต์ให้มีการใช้งานได้ง่ายมากยิ่งขึ้น : และมีระบบช่วยเหลือผู้ใช้งานมากขึ้น เช่น สคริปต์เอคิเตอร์ที่ชาญฉลาด, ระบบโพรไฟล์ผู้ใช้
- ✓ ภาษาสคริปต์ที่ใช้ต้องมีความสมบูรณ์, มีความสามารถที่หลากหลาย และใช้งานได้ง่าย จะต้องมีฟังก์ชันไลบรารี (Built-in Script Utilities) มากๆเพื่อการทำงานที่ครอบคลุม

## บทที่ 3

### หลักการพื้นฐาน

#### 3.1 หลักการพื้นฐานของเอเจนต์ (Agents)

##### 3.1.1 ความหมายของเอเจนต์

เอเจนต์ในมุมมองของผู้ใช้ คือ โปรแกรมที่ช่วยงานผู้ใช้และทำงานที่เป็นประโยชน์กับผู้ใช้  
เอเจนต์ในความหมายของระบบ คือ โปรแกรมที่มีลักษณะดังนี้

- อาศัยอยู่ในเอนิวิตซ์ชันเอนไวรอนเมนต์ (Execution environment)
- มีคุณสมบัติดังต่อไปนี้
  - รีแอ็กทีฟ (Reactive) รับรู้การเปลี่ยนแปลงของสิ่งแวดล้อมและตอบสนองต่อการเปลี่ยนแปลงของสิ่งแวดล้อม
  - ออโตโนมิตี (Autonomous) ควบคุมการทำงานของตัวเองได้สามารถทำกิจกรรมต่างๆ ได้อย่างอิสระโดยไม่ขึ้นกับผู้ใช้ หรือสิ่งกระตุ้นภายนอก
  - มีเป้าหมายแน่นอน (Goal-driven)
  - ทำงานอย่างต่อเนื่อง (Continuous)
- และอาจจะมีคุณสมบัติบางอย่างดังนี้
  - สามารถติดต่อกับเอเจนต์ตัวอื่นได้ (Communicative)
  - สามารถเคลื่อนย้ายจากที่หนึ่งไปยังอีกที่หนึ่งได้ แม้ว่าจะอยู่คนละแพลตฟอร์มกัน (Mobile or Portable)
  - สามารถเรียนรู้และเปลี่ยนแปลงพฤติกรรมได้ขึ้นอยู่กับประสบการณ์ (Adaptive)

##### 3.1.2 การแบ่งประเภทของเอเจนต์

เอเจนต์อาจจะแบ่งประเภทได้ตามคุณสมบัติดังกล่าวนี้ หรือจะแบ่งประเภทของเอเจนต์ด้วยวิธีอื่นก็ได้ เช่น เราอาจแบ่งเอเจนต์ตามลักษณะงานที่เอเจนต์ทำ ตัวอย่างเช่น เอเจนต์ที่รวบรวมข้อมูลหรือเอเจนต์ที่ทำหน้าที่กรองจดหมายอิเล็กทรอนิกส์ เป็นต้น

Brustoloni แบ่งเอเจนต์ออกเป็น 3 ประเภท คือ เรกกูเลชันเอเจนต์ (Regulation agents), แพลนนิ่งเอเจนต์ (Planning agents) และ อแด็ปทีฟเอเจนต์ (Adaptive agents)

- Regulation agents คือเอเจนต์ที่ตอบสนองต่อการเปลี่ยนแปลงของสภาพแวดล้อม และเป็นเอเจนต์ที่มีเป้าหมายว่าจะทำอะไรอย่างแน่นอน
- Planning agents คือเอเจนต์ที่เลือกวิธีการทำงานที่เหมาะสมที่สุด โดยใช้ทฤษฎีของวิชาปัญญาประดิษฐ์ หรืออาจใช้อัลกอริทึมแบบสุ่ม (Randomizing Agent)

- Adaptive agents คือเอเจนต์ที่สามารถเรียนรู้ระบบ และเปลี่ยนแปลงการทำงานของตัวเองได้ หรืออาจเรียกว่า เอเจนต์ที่สามารถเรียนรู้ได้ (Learning Agent)

Property	Other names	Meaning
Reactive	Sensing and acting	ตอบสนองต่อการเปลี่ยนแปลงของสิ่งแวดล้อม
Autonomous		สามารถทำงานได้ด้วยตัวเอง
Goal-oriented	Pro-active purposeful	มีเป้าหมายในการทำงาน
Temporally continuous		ทำงานอย่างต่อเนื่อง
Communicative	Socially able	ติดต่อกับเอเจนต์อื่นๆ หรือผู้ใช้
Learning	Adaptive	เรียนรู้และเปลี่ยนแปลงการทำงานของตัวเองตามประสบการณ์
Mobile		สามารถเคลื่อนย้ายตัวเองจากที่หนึ่งไปอีกที่หนึ่ง
Flexible		มีความยืดหยุ่น

ตารางที่ 3-1 แสดงคุณสมบัติต่างๆ ของเอเจนต์

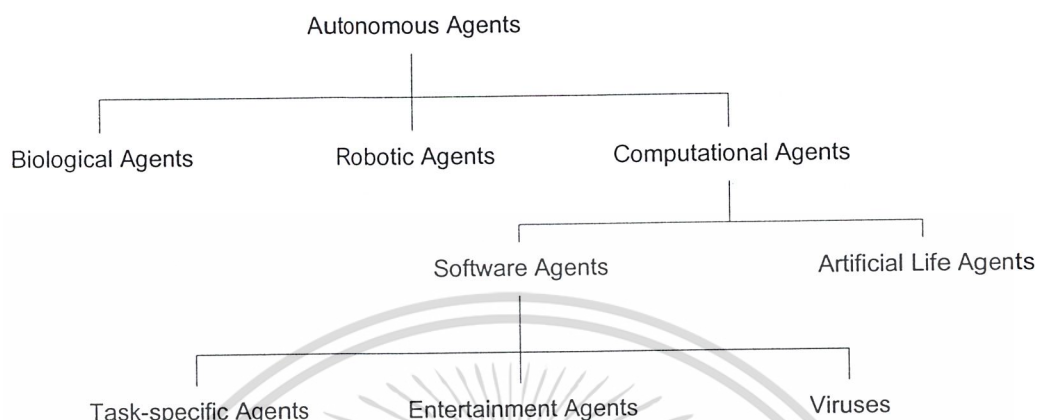
### 3.1.3 ลำดับชั้นของเอเจนต์ (Agents Hierarchy)

เราสามารถเปรียบเทียบความสัมพันธ์ระหว่างเอเจนต์ชนิดต่างๆ กับโมเดลทางชีววิทยาได้ เช่น ลำดับชั้นของมนุษย์เป็นดังนี้

- ❖ Kingdom = Animal
- ❖ Phylum = Chordata
- ❖ Class = Mammalia
- ❖ Order = Primate
- ❖ Family = Pongidae
- ❖ Subfamily = Hominidae
- ❖ Genus = Homo
- ❖ Species = Sapiens

เราสามารถจัดลำดับชั้นของเอเจนต์ให้อยู่ในลักษณะนี้ได้เช่นกัน โดยในระดับ Kingdom เราสามารถแบ่งเอเจนต์ออกเป็น 3 ชนิดดังนี้ ไปโอโลจิกอลเอเจนต์ (Biological agents), โรโบติกเอเจนต์ (Robotic agents) และคอมพิวเตอร์เอเจนต์ (Computational agents) โดยเอเจนต์ที่เราสนใจอยู่ในกลุ่มของคอมพิวเตอร์เอเจนต์ ในระดับ Phylum เราสามารถแบ่งคอมพิวเตอร์เอเจนต์ได้เป็น 2 ประเภท คือ ซอฟต์แวร์เอเจนต์ (Software agents) และอาทิฟิเชียลไลฟ์เอเจนต์ (Artificial life agents) เราสามารถแบ่งซอฟต์แวร์เอเจนต์ได้เป็น 3 ประเภท คือ ทาส์กสเปซิฟิกเอเจนต์ (Task-specific agents), เอน

เตอร์เทนเมนต์เอเจนต์ (Entertainment agents) และไวรัสคอมพิวเตอร์ (Computer viruses) ซึ่งลำดับชั้นทั้งหมดที่กล่าวมาสามารถแสดงเป็นแผนภาพได้ดัง รูปที่ 3-1



รูปที่ 3-1 รูปแสดงลำดับชั้นของเอเจนต์

#### 3.1.4 เอเจนต์ที่ชาญฉลาดด้วยภาษาไพธอน (Python-Implemented Intelligent Agent)

ภาษาไพธอนเป็นภาษาที่มีความชาญฉลาดในตัว แต่ยังคงความเป็นภาษาที่มีโครงสร้าง คุณสมบัติต่างๆ ของภาษาไพธอนที่สนับสนุนอินเทลลิเจนต์เอเจนต์แอปพลิเคชัน (Intelligent Agent Application) มีดังนี้ ความเป็นอัตโนมัติ (Autonomous), ความฉลาด (Intelligence), ความสามารถเคลื่อนที่ได้ (Mobile) และความสามารถติดต่อกับเอเจนต์ตัวอื่น (Communicative)

##### ความเป็นอัตโนมัติ

โปรแกรมที่สามารถทำงานโดยอัตโนมัติจะทำงานอยู่ตลอดเวลา พร้อมทั้งจะตอบสนองต่อผู้ใช้หรือการกระตุ้นจากสิ่งแวดล้อมได้ทุกเมื่อ คำถามที่เกิดขึ้นก็คือ เอเจนต์จะรับรู้ได้อย่างไรว่ามีการเปลี่ยนแปลงในสิ่งแวดล้อม วิธีการหนึ่งที่ทำให้เอเจนต์รับรู้สิ่งแวดล้อมได้ตลอดเวลาอาจใช้วิธีการส่งอีเวนต์ (event) มายังเอเจนต์ โดยถ้ามองในมุมมองการออกแบบเชิงวัตถุ (Object-oriented design) อีเวนต์ที่เอเจนต์ได้รับก็คือ การเรียกเมธอด (method) ของเอเจนต์ หรือการส่งเมสเสจ (message) ให้กับเอเจนต์นั่นเอง

นอกจากนี้เอเจนต์ยังสามารถตรวจสอบ คอยรับรู้การเปลี่ยนแปลงของสิ่งแวดล้อมได้โดยตรง โดยวิธีการนี้เอเจนต์จำเป็นต้องแตกโปรเซสออกมา เพื่อทำหน้าที่ตรวจสอบและรวบรวมข้อมูลไปพร้อมๆ กับโปรเซสอื่นๆ

ภาษาไพธอนสนับสนุนการทำงานทั้งแบบหลายโปรเซส (multi-process) หรือหลายเธรด (multi-thread) ขึ้นกับความเหมาะสมของแพลตฟอร์ม เช่น การแตกโปรเซสสามารถทำได้ง่ายภายในระบบปฏิบัติการยูนิกซ์ แต่ไม่เหมาะกับระบบปฏิบัติการวินโดวส์

## ความฉลาด

ในการสร้างโปรแกรมปัญญาประดิษฐ์ (Artificial Intelligent Applications) นั้นมีปัจจัยหลักๆ 2 อย่าง คือ

- Knowledge representation คือการแทนฐานความรู้ให้อยู่ในรูปแบบที่สามารถ ประมวลผลได้ และง่ายต่อการเก็บ
- Algorithm คือวิธีการหรือกระบวนการแก้ปัญหา

ภาษาไพธอนสนับสนุน Knowledge representation โดยใช้ if-then rule และ built-in type Dictionary ซึ่งเป็น โครงสร้างข้อมูลที่เหมาะสมกับการเก็บฐานความรู้

นอกจากนี้ภาษาไพธอนเป็นภาษาที่มีไวยากรณ์ง่าย มีเหตุผล และไม่ฟุ่มเฟือยทำให้การเขียนโปรแกรมสามารถคิดถึงปัญหาที่ต้องการได้อย่างเต็มที่ ไม่ต้องมาคำนึงถึงตัวภาษา

## ความสามารถเคลื่อนที่ได้

ซอร์สโค้ด (Sourcecode) และไบต์โค้ด (Bytes code) ของภาษาไพธอนที่คอมไพล์แล้ว สามารถนำไปใช้งานในระบบที่ต่าง แพลตฟอร์มได้

## ความสามารถติดต่อกับเอเจนต์ตัวอื่น

ภาษาไพธอนสามารถเข้าถึงข้อมูลของระบบได้อย่างง่ายดาย รวมทั้งความสามารถในการ เรียกใช้คำสั่งต่างๆ บน DOS สำหรับระบบปฏิบัติการวินโดวส์ หรือบน Shell สำหรับระบบปฏิบัติการยูนิกซ์ อีกทั้งความสามารถในการเรียกโปรแกรมอื่นๆ

นอกจากนี้สำหรับระบบปฏิบัติการวินโดวส์ ไพธอนมีไลบรารีสำหรับการติดต่อกับ Windows COM (Component Object Model) ซึ่งความสามารถต่างๆเหล่านี้สนับสนุนความสามารถติดต่อกับเอเจนต์อื่นได้เป็นอย่างดี

## 3.2 หลักการพื้นฐานของเว็บเบราว์เซอร์

### 3.2.1 โปรแกรมเว็บเบราว์เซอร์

โปรแกรมเว็บเบราว์เซอร์ คือ โปรแกรมแสดงเว็บเพจ เราสามารถแบ่งเบราว์เซอร์ตามลักษณะการแสดงผลได้ 2 ประเภท ดังนี้

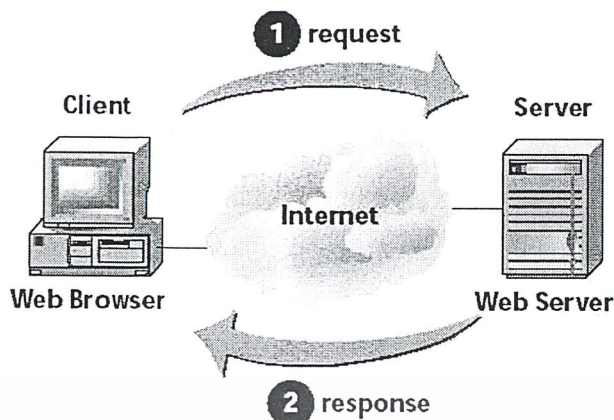
1. *เท็กซ์บราวเซอร์ (Text Browser)* เป็นเบราว์เซอร์ที่มีมาก่อน แต่ในปัจจุบันมีใช้กันในการศึกษาและวิจัย ไม่เป็นที่นิยมเนื่องจากแสดงผลเป็นตัวอักษรอย่างเดียว
2. *กราฟฟิคคอลลบราวเซอร์ (Graphical Browser)* เป็นเบราว์เซอร์ที่ถูกพัฒนาขึ้นมาทีหลัง เบราวเซอร์ประเภทนี้สามารถแสดงรูปภาพได้ และในปัจจุบันเบราว์เซอร์ได้มีการนำความสามารถในการแสดงผลในรูปแบบสื่อผสม (Multimedia) คือมีทั้งภาพและเสียง

### ตัวอย่างบราวเซอร์และรายละเอียด

- ❖ **โมเสก (Mosaic)** เป็นการฟีกคอลบราวเซอร์ตัวแรกที่มีการแสดงในลักษณะของรูปภาพ สร้างในปี 1993 โดย Mark Andreessen เป็นผู้ริเริ่มสร้างบราวเซอร์ตัวนี้ร่วมกับ National Center for Supercomputing Application (NCSA) ใช้ได้บนระบบปฏิบัติการวินโดวส์ 3.1.x (Windows v 3.1.x), วินโดวส์95 (Windows 95) และวินโดวส์เอ็นที (Windows NT)
- ❖ **เน็ตสเคป (Netscape)** เป็นบราวเซอร์ที่แสดงด้วยรูปภาพ และเป็นที่ยอมรับโดยทั่วไป พัฒนาโดยบริษัทเน็ตสเคป คอมมิวนิเคชัน (Netscape Communications) บราวเซอร์ของเน็ตสเคปเรียกว่า นาวิกเตอร์ (Navigator) ซึ่งถูกพัฒนาในปี 1995 โดยมี Mark Andreessen เป็นหัวหน้าในการพัฒนา
- ❖ **ไมโครซอฟท์ อินเทอร์เน็ต เอ็กซ์พลอเรอร์ (Microsoft Internet Explorer)** เป็นบราวเซอร์ที่แสดงรูปภาพด้วยเช่นกัน
- ❖ **ลินซ์ (Lynx)** เป็นบราวเซอร์ที่แสดงด้วยตัวอักษร (Text Browser) ซึ่งพัฒนาโดยมหาวิทยาลัยแคนซัส (University of Kansas) โดยใช้บนยูนิกซ์เวิร์กสเตชัน (UNIX workstation)
- ❖ **โอเปรา (Opera)** เป็นบราวเซอร์ที่มีข้อดีกว่านาวิกเตอร์และไมโครซอฟท์ อินเทอร์เน็ต เอ็กซ์พลอเรอร์ คือ เป็นบราวเซอร์ที่มีขนาดเล็ก ใช้เวลาในการโหลดน้อย โอเปราถูกพัฒนาโดยบริษัทนอร์วีเจียน ซอฟต์แวร์ (Norwegian software company) ซึ่งสามารถใช้ได้บนระบบปฏิบัติการวินโดวส์ และโอเอสทู (OS/2)
- ❖ **ฮอตจาวา (HotJava)** เป็นบราวเซอร์ที่แสดงด้วยรูปภาพ โดยมีบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems) เป็นผู้พัฒนา ในเวอร์ชันใหม่ล่าสุดมีการใช้คอมโพเนนต์ (Component) ของจาวาบีนส์ (JavaBean) ด้วย

### 3.2.2 หลักการทำงานของโปรแกรมเว็บบราวเซอร์

เว็บบราวเซอร์หรืออาจเรียกได้ว่าเป็นเอเจนต์ของผู้ใช้ เป็นโปรแกรมที่ทำงานอยู่บนพื้นฐานของโปรโตคอลเอชทีทีพี (HTTP : Hypertext Transfer Protocol) โดยหน้าที่หลักๆของโปรแกรมคือ ร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์ (Web Server) โดยข้อมูลที่ร้องขอเป็นข้อมูลเอชทีเอ็มแอล (HTML : Hypertext Markup Language) จากนั้นโปรแกรมเว็บบราวเซอร์จะอินเตอร์พรีต (Interprete) ข้อมูลที่ได้รับมาและแสดงผล โดยการอ้างอิงระหว่างเอกสารเอชทีเอ็มแอลจะใช้ไฮเพอร์ลิงก์ (Hyperlink)



รูปที่ 3-2 รูปแสดงการทำงานของเว็บเบราว์เซอร์

จากรูป เริ่มต้นเว็บเบราว์เซอร์เป็นฝ่ายคิดต่อไปหาเว็บเซิร์ฟเวอร์ จากนั้นเว็บเซิร์ฟเวอร์จะตอบกลับมาพร้อมกับข้อมูลที่ถูกร้องขอ

### 3.3 หลักการพื้นฐานของอินเทอร์เน็ต

#### 3.3.1 ที่มาและความหมายของอินเทอร์เน็ต และเว็บ

อินเทอร์เน็ต (Internet) เป็นชื่อเครือข่ายคอมพิวเตอร์ที่แผ่ครอบคลุมทั่วโลก พัฒนาต่อมาจาก ARPAnet ซึ่งแต่เดิมเป็นเครือข่ายทางการทหารของสหรัฐอเมริกาโดยเงินทุนของ ARPA (Advanced Research Projects Agency)

เวิลด์ไวด์เว็บ (World Wide Web) เป็นบริการหนึ่งบนอินเทอร์เน็ต ที่ให้ข้อมูลข่าวสารในรูปแบบเอกสารที่เอ็มแอล ผ่านโปรโตคอลเอชทีทีพี ทั้งโปรโตคอลเอชทีทีพี และเวิลด์ไวด์เว็บ รวมทั้งเว็บเบราว์เซอร์ตัวแรก ถูกออกแบบและพัฒนาขึ้นโดย Tim Berners-Lee และ Robert Cailliau

#### 3.3.2 โปรโตคอลทีซีพีไอพี (TCP/IP Protocol)

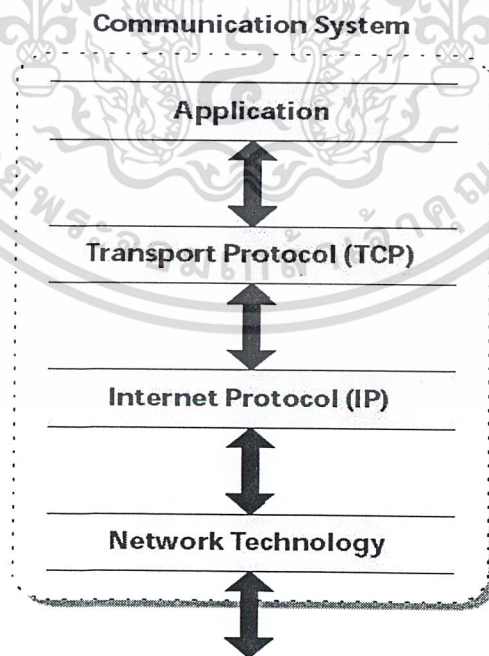
หลักการงานของเน็ตเวิร์คโปรโตคอลใดๆ จะมีการทำงานที่แบ่งออกเป็นชั้นๆ การทำงานทั้งหมดของโปรโตคอลจะประกอบด้วยหลายๆเลเยอร์ (layer) ซึ่งนำมาวางซ้อนทับกันได้ออกมาในรูปแบบที่เรียกกันว่าโปรโตคอลสแต็ก (protocol stack) แต่ละชั้นก็จะมีหน้าที่การทำงานที่ชัดเจน และไม่เกี่ยวข้องกัน

เน็ตเวิร์คโปรโตคอลมาตรฐาน คือ โอเอสไอโมเดล (OSI : Open System Interconnection) ส่วนเน็ตเวิร์คโปรโตคอลที่ใช้กันอย่างแพร่หลาย และเป็นมาตรฐานบนเครือข่ายอินเทอร์เน็ต คือ ทีซีพีไอพีโปรโตคอล (TCP/IP Protocol) รายละเอียดแสดงดัง ตารางที่ 3-2

Layer	Name	Responsibility	Protocol
4	Application Layer	consist of applications and processes that use the network	Telnet, FTP, SMTP, HTTP, DNS, RIP, NFS
3	Transport Layer	provides end-to-end data delivery services	TCP, UDP
2	Internet Layer	defines the datagram and handles the routing of data	IP, ICMP
1	Network Access Layer	consist of routines for accessing physical networks	Ethernet, FDDI, Frame Relay

ตารางที่ 3-2 TCP/IP Layers และตัวอย่างโปรโตคอล

เฉพาะส่วนโปรคอลลที่ใช้กับเว็บมี อินเทอร์เน็ตโปรโตคอล (Internet Protocol หรือ IP), ทรานส์มิชชั่นคอนโทรลโปรโตคอล (Transmission Control Protocol หรือ TCP), ไฮเพอร์เท็กซ์ทรานส์เฟอร์โปรโตคอล (Hypertext Transfer Protocol หรือ HTTP) ดังแสดงใน รูปที่ 3-3



รูปที่ 3-3 รูปแสดง TCP/IP protocol for WWW

### อินเทอร์เน็ตโปรโตคอล (Internet Protocol)

ถือได้ว่าเป็นหัวใจของทีซีพีไอพีโปรโตคอล หน้าที่สำคัญของโปรโตคอลนี้คือ

- สร้างคำดาแกรม (defining the datagram)
- หาเส้นทางเพื่อส่งคำดาแกรมไปยังปลายทาง (routing)
- แบ่งและประกอบคำดาแกรม (fragmenting and reassembling)

อินเทอร์เน็ตโปรโตคอลเป็นโปรโตคอลแบบคอนเนกชันเลส (connectionless) นั่นคือมันจะส่งข้อมูลออกไปโดยไม่คำนึงว่าจะไปถึงปลายทางหรือไม่ นอกจากนี้ยังไม่มีตรวจสอบความถูกต้องของข้อมูลด้วย

### ทรานส์มิสชันคอนโทรลโปรโตคอล (Transmission Control Protocol)

เป็นโปรโตคอลแบบคอนเนกชันออเรียนเต็ด (connection-oriented) และมีการตรวจสอบความถูกต้องของข้อมูลที่ได้รับด้วย โดยเมื่อสองโปรแกรมต้องการจะติดต่อกัน การเชื่อมต่อ (connection) จะต้องถูกสร้างขึ้นมาก่อน และส่งข้อมูลไปมาผ่านการเชื่อมต่อที่สร้างขึ้นมา ทรานส์มิสชันคอนโทรลโปรโตคอลรับประกันว่าข้อมูลถูกส่งผ่านจากปลายทางด้านหนึ่งไปยังปลายทางอีกด้านหนึ่ง (end-to-end communication)

ทรานส์มิสชันคอนโทรลโปรโตคอล จัดให้มีช่องทางการเชื่อมต่อแบบหนึ่งต่อหนึ่ง (point-to-point channel) สำหรับแอปพลิเคชันต่างๆ ที่ต้องการการติดต่อกันที่เชื่อถือได้ เช่น HTTP, FTP, Telnet

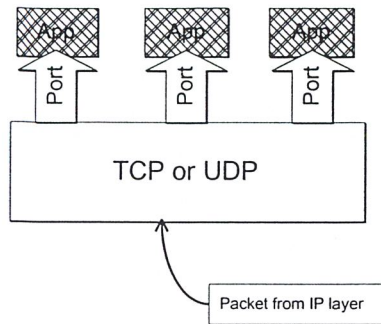
### ยูสเซอร์ดาตาแกรมโปรโตคอล (User Datagram Protocol หรือ UDP)

เป็นโปรโตคอลแบบคอนเนกชันเลส (connectionless) จึงไม่มีการรับประกันการส่งข้อมูลถึงปลายทาง และไม่มีการตรวจสอบความถูกต้องของข้อมูล ยูสเซอร์ดาตาแกรมโปรโตคอล จะส่งข้อมูลเป็นแพ็กเกจออกไปอย่างอิสระ ทำให้มีความเร็วในการทำงานสูง เหมาะที่จะใช้กับงานที่ส่งข้อมูลขนาดเล็กและส่งบ่อยๆ

### พอร์ตและซอกเก็ต (Port and Socket)

โดยทั่วไปคอมพิวเตอร์เครื่องหนึ่งจะมีการเชื่อมต่อทางกายภาพไปยังเครือข่ายเพียงการเชื่อมต่อเดียว ทุกๆ ข้อมูลที่ส่งออกไปจะมุ่งไปยังปลายทางการเชื่อมต่อ แต่อย่างไรก็ตามข้อมูลอาจจะถูกส่งไปให้แอปพลิเคชันต่างๆกันที่ทำงานอยู่ที่เครื่องคอมพิวเตอร์ปลายทาง การส่งข้อมูลให้แยกแยะแอปพลิเคชันให้ถูกต้องจะใช้ หมายเลขพอร์ตในการกำหนด

การส่งข้อมูลในระดับแอปพลิเคชัน จะอาศัยทีซีพี หรือไม่ก็ยูดีพีในการส่งข้อมูลผ่านลงไปยังไอพี โดยหมายเลขไอพี (IP address) รับประกันว่าข้อมูลจะถูกส่งไปถึงเครื่องปลายทางที่ต้องการ และหมายเลขพอร์ต (Port number) รับประกันข้อมูลที่จะถูกส่งให้กับแอปพลิเคชันที่ต้องการ



รูปที่ 3-4 รูปแสดงการติดต่อถึงชั้นแอปพลิเคชันโดยใช้พอร์ต

หมายเลขไอพีพร้อมกับหมายเลขพอร์ตเมื่อนำมารวมกันแล้ว จะเรียกว่าซอกเก็ต (Socket) โดยหมายเลขซอกเก็ตจะระบุถึงโปรเซส บนเครื่องคอมพิวเตอร์ใดๆ ที่เชื่อมต่อกับเครือข่าย เช่น โปรแกรมเว็บเซิร์ฟเวอร์บนเครื่องเจ้าคุณจะมีหมายเลขซอกเก็ตคือ 161.246.10.21.80

### 3.3.3 ยูอาร์แอล (URL)

ยูอาร์แอล (URL : Uniform Resource Locator) คือตัวอ้างอิง (reference) ไปยังรีซอร์ส (resource) บนอินเทอร์เน็ต ยูอาร์แอลเป็นข้อความที่มีรูปแบบที่ใช้อธิบายว่าจะพบรีซอร์สบนอินเทอร์เน็ตได้อย่างไร โดยยูอาร์แอลประกอบด้วยข้อมูลมากมายโดยมีโครงสร้างดัง รูปที่ 3-5 รายละเอียดและการใช้งานของแต่ละส่วนประกอบแสดงดัง ตาราง 3-3

[http://guest:secret@www.ietf.org:80/html.charters/wg-dir.html?sess=1#Applications\\_Area](http://guest:secret@www.ietf.org:80/html.charters/wg-dir.html?sess=1#Applications_Area)

```

protocol ... http
username ..... guest
password ..... secret
host ..... www.ietf.org
port ..... 80
path ..... /html.charters
file ..... wg-dir.html
query ..... sess=1
fragment ..... Applications_Area

```

รูปที่ 3-5 รูปแสดงโครงสร้างของยูอาร์แอล (URL)

Component	Use
protocol	Identifies the application protocol needed to access the resource
username	If the protocol supports the concept of user names, this provides a user name that has access to the resource
password	The password associated with the user name
host	The communication system that has the resource
port	The TCP port that the application protocols should use to access the resource; many protocols have an implied TCP port (for HTTP that port is 80)
path	The path through a hierarchical organization under which the resource is located, often a file system's directory structure or equivalent
file	The resource itself
query	Additional information about the resource or the client
fragment	a particular location within a resource

ตารางที่ 3-3 องค์ประกอบของยูอาร์แอล (URL Component)

### 3.3.4 การเขียนโปรแกรมเน็ตเวิร์กโดยภาษาไพธอน (Network Programming in Python)

ภาษาไพธอนมีโมดูลที่ทำหน้าที่เกี่ยวกับเน็ตเวิร์ก อยู่หลายโมดูล ตั้งแต่ระดับล่างอย่างเช่นซอกเก็ต (Socket), หรือสูงขึ้นมาในระดับแอปพลิเคชัน โพรโทคอลเช่น เอฟทีพี (FTP) และเอชทีทีพี (HTTP) หรือแม้แต่การเข้าถึงข้อมูลบนอินเทอร์เน็ตในรูปแบบยูอาร์แอล (URL) ทำให้โปรแกรมเมอร์สามารถเลือกใช้ได้ตามความเหมาะสม ถึงแม้การใช้โมดูลในระดับสูงเช่น URL จะมีความสะดวกมาก แต่บางครั้งเราก็จำเป็นต้องเขียนในระดับล่างๆ เช่น Socket เพื่อได้งานที่ตรงตามต้องการจริง นอกจากนี้แล้วพื้นฐานการเขียนโปรแกรมซอกเก็ต (Socket Programming) เป็นพื้นฐานที่สำคัญในการเขียนโปรแกรมเน็ตเวิร์ก (Network Programming)

#### การเขียนโปรแกรมซอกเก็ต (Socket Programming)

ภาษาไพธอนมีโมดูลซอกเก็ต (Socket Module) ที่ใช้ในการสร้างซอกเก็ต, เชื่อมต่อคอนเน็คชัน, รับส่งข้อมูล และการปิดการเชื่อมต่อ

ตัวอย่างต่อไปนี้เป็นชุดโค๊ดที่ใช้ในการสร้าง TCP Server และ TCP Client โดยซอกเก็ต

```

ss = socket()           # create server socket
ss.bind()              # bind socket to address
ss.listen()            # listen for connections
inf_loop :             # server infinite loop
    cs = ss.accept()   # accept client connection
    comm_loop :        # communication loop
        cs.recv()|cs.send() # receive / send
        cs.close()     # close client socket
    ss.close()         # close server socket

```

```

cs = socket()           # create client socket
cs.connect()           # attempt sever connection
comm_loop :           # communication loop
    cs.send()|cs.recv() # send / receive
cs.close()            # close client socket

```

หลักการการทำงานของ TCP Server คือ สร้าง socket ขึ้นมา และผูก socket นี้เข้ากับ socket address (pair of IP and Port) ที่ต้องการ จากนั้นจะเข้าสู่ forever loop ซึ่งคอยรับการเชื่อมต่อจาก client, ให้บริการ และยุติการเชื่อมต่อ

หลักการการทำงานของ TCP Client คือ สร้าง socket ขึ้นมาและพยายามเชื่อมต่อกับ TCP server เมื่อเชื่อมต่อได้ก็จะรับส่งข้อมูล และยุติการเชื่อมต่อเมื่อเสร็จสิ้น

ตัวอย่างต่อไปนี้เป็นชุดโคโอดที่ใช้ในการสร้าง UDP Server และ UDP Client โดยชอกเก็จ

```

ss = socket()           # create server socket
ss.bind()              # bind socket to address
inf_loop :             # server infinite loop
    cs = ss.recvfrom|ss.sendto # receive / send
    ss.close()         # close server socket

```

```

cs = socket()           # create server socket
comm_loop :           # communication loop
    cs.sendto()|cs.recvfrom() # receive / send
cs.close()            # close server socket

```

การทำงานของ UDP Server และ UDP Client จะคล้ายกับ TCP Server และ TCP Client ตามลำดับ แต่ทว่าไม่ต้องมีการสร้างการเชื่อมต่อขึ้นมาก่อน จึงมีความยุ่งยากน้อยกว่า

### โมดูลยูอาร์แอล (URL Module)

โมดูลนี้อยู่ในระดับที่สูงที่สุด มีความง่ายต่อการใช้งานมาก โดยมีการมอดูอาร์แอลต่างๆ ให้เหมือนไฟล์หนึ่งๆ (file-like) ทำให้โปรแกรมเมอร์ไม่ต้องยุ่งกับโปรโตคอล และการเชื่อมต่อ โดยสามารถนำมาประยุกต์ใช้ได้ทั้ง เอฟทีพี (FTP) และเอชทีทีพี (HTTP)

ตัวอย่างต่อไปนี้เป็นกรเรียกเว็บเพจของสถาบันฯ ออกมาแสดงผลในรูปแบบเอกสารเอชทีเอ็มแอล (HTML)

```

remoteAddress = 'http://kmitl.ac.th'
remoteFile = urllib.urlopen(remoteAddress)
remoteData = remoteFile.readlines()
remoteFile.close()
for line in remoteData :
    print line

```

โดยการทำงานของโปรแกรมคือมีการเรียก urllib.urlopen ซึ่งซ่อนความยุ่งยากไว้ และส่งค่าออกมาเป็นออบเจกต์ที่คล้ายไฟล์ (file-like object) โดยออบเจกต์ที่ได้มานี้มีเมธอดต่างๆ เหมือนไฟล์ออบเจกต์จึงสามารถอ่านได้ และเมื่ออ่านข้อมูลเสร็จก็เรียก close เพื่อปิดการเชื่อมต่อ

### 3.4 เอชทีทีพี (HTTP)

ย่อมาจาก ไฮเปอร์เท็กซ์ทรานส์เฟอร์โพรโตคอล (Hypertext Transfer Protocol) คือโพรโตคอล (protocol) สำหรับเครือข่ายที่ใช้ส่งไฟล์และข้อมูลต่างๆ บนเว็ลด์ไวด์เว็บ (World Wide Web) ไม่ว่ามันจะเป็นเอกสารเอชทีเอ็มแอล (HTML) ไฟล์รูปภาพ หรือไฟล์อะไรก็ตาม โปรแกรมบราวเซอร์ คือ เอชทีทีพี-ไคลเอ็นต์ (HTTP-Client) เพราะว่ามันส่งคำร้องต่างๆ ไปยัง เอชทีทีพี-เซิร์ฟเวอร์ (HTTP-Server) ซึ่งก็คือโปรแกรมเว็บเซิร์ฟเวอร์ ที่ทำหน้าที่ให้บริการไคลเอ็นต์ โดยคอยส่งการตอบสนองตามที่ไคลเอ็นต์ต้องการ การติดต่อระหว่าง เอชทีทีพี-ไคลเอ็นต์ กับ เอชทีทีพี-เซิร์ฟเวอร์ จะกระทำผ่านพอร์ต (Port) มาตรฐานของเอชทีทีพี คือ พอร์ตหมายเลข 80

#### โครงสร้างของเอชทีทีพี

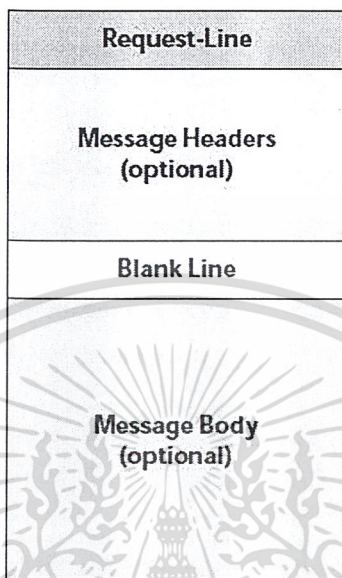
เอชทีทีพี ก็เหมือนกับโพรโตคอลอื่นๆ ที่ใช้รูปแบบไคลเอ็นต์-เซิร์ฟเวอร์ : เอชทีทีพี-ไคลเอ็นต์สร้างการเชื่อมต่อไปยัง เอชทีทีพี-เซิร์ฟเวอร์ แล้วส่งการร้องขอบริการไปยังเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ได้รับการร้องขอ ก็จะส่ง การตอบรับกลับไปยังไคลเอ็นต์ ซึ่งโดยปกติแล้วก็จะส่งกลับไปด้วยข้อมูลที่ไคลเอ็นต์ร้องขอมา เช่น ถ้าไคลเอ็นต์ต้องการไฟล์ index.html ก็จะส่งการร้องขอไปยังเซิร์ฟเวอร์ แล้วเซิร์ฟเวอร์ก็จะตอบสนองกลับมาพร้อมกลับไฟล์ index.html ที่ไคลเอ็นต์ต้องการ หลังจากตอบสนองความต้องการของไคลเอ็นต์เสร็จแล้ว เซิร์ฟเวอร์ก็จะปิดการเชื่อมต่อ ซึ่งการทำงานในลักษณะนี้ทำให้เอชทีทีพีกลายเป็นโพรโตคอลที่ไม่มีสถานะ (Stateless)

รูปแบบ การร้องขอ (request) และ การตอบรับ (response) อยู่ในรูปแบบที่คล้ายกัน ซึ่งเป็นภาษาอังกฤษทั้งคู่ ประกอบด้วย :

- ❖ บรรทัดเริ่มต้น
- ❖ บรรทัดเฮดเดอร์จะมีหลายบรรทัดหรือไม่ก็ได้
- ❖ บรรทัดว่าง (ทั้ง LF และ CR)
- ❖ บรรทัดข้อมูลจะมีหรือไม่ก็ได้

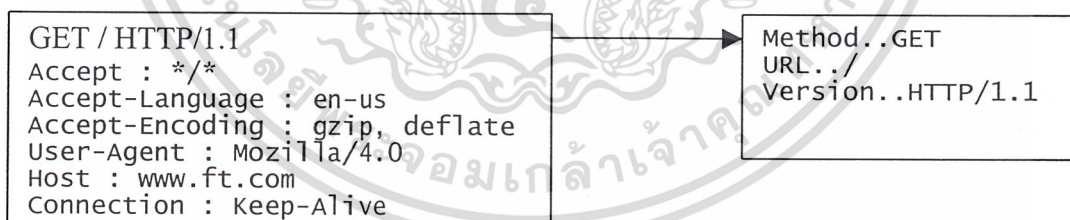
### 3.4.1 โครงสร้างการร้องขอเอชทีทีพี (HTTP Requests)

โครงสร้างของ HTTP Request แสดงดัง รูปที่ 3-6



รูปที่ 3-6 รูปแสดงโครงสร้างของ HTTP Request

HTTP Request-Line ประกอบด้วย เมธอด (method), ยูอาร์แอล (URL), เอชทีทีพีเวอร์ชัน (HTTP version) ดังตัวอย่างต่อไปนี้



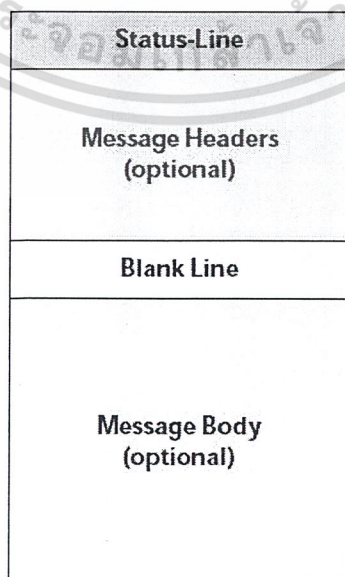
ส่วนของเมธอด ตามโพรโตคอลเอชทีทีพีนั้นมี 8 วิธีดังแสดงใน ตารางที่ 3-4 แต่ที่ใช้กันมาก และเป็นที่ยอมรับจะมีอยู่ 3 ตัวคือ GET, POST และ HEAD

Method	Server Support	Use
CONNECT	Optional	Asks server to establish a tunnel
DELETE	Optional	Asks server to delete the indicated resource
GET	Required	Asks server to return requested resource
HEAD	Required	Asks server to reply as if it were going to return the requested resource, but not to include the resource itself in the response
OPTIONS	Optional	Asks server to indicate the options it supports for the indicated resource
POST	Optional	Asks server to pass the message body to the indicated resource
PUT	Optional	Asks server to accept the message body as the indicated resource
TRACE	Optional	Asks server simply to respond to the request

ตารางที่ 3-4 เชนที่ที่ที่เมธอด (HTTP method)

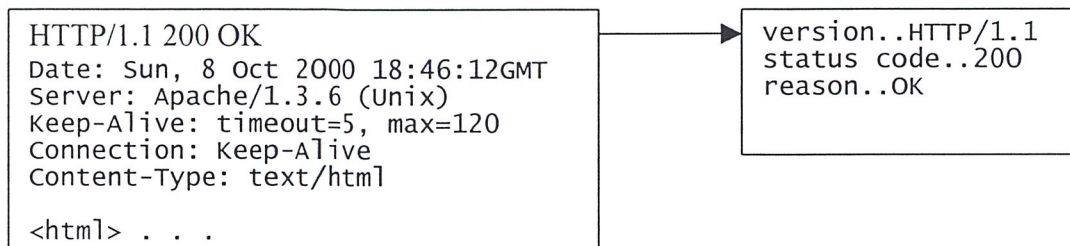
### 3.4.2 โครงสร้างการตอบรับเชนที่ที่ที่ (HTTP Response)

โครงสร้างของ HTTP Response แสดงดัง รูปที่ 3-7



รูปที่ 3-7 รูปแสดงโครงสร้างของ HTTP Response

HTTP Status-Line ประกอบด้วย เวอร์ชัน (version), สถานะ (status code), เหตุผล (reason) ดังตัวอย่างต่อไปนี้ รหัสสถานะและความหมายอย่างคร่าวๆ สามารถแสดงได้ดังตารางที่ 3-5



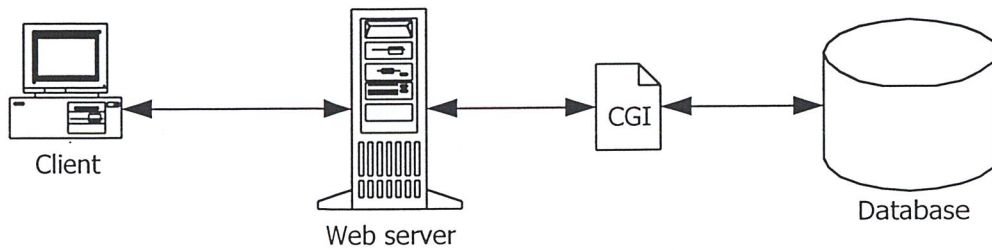
Status Code	Meaning
100-199	Informational; the result is not yet available
200-299	Success
300-399	Redirection
400-499	Client error; the request contained an error
500-599	Server error; the server failed to act on a request

ตารางที่ 3-5 สถานะการตอบรับ (HTTP Status Code Categories)

### 3.5 ซีจีไอ (Common Gateway Interface)

ซีจีไอ คือ หลักการของการพัฒนาโปรแกรมประยุกต์ ที่ทำหน้าที่เสมือนประตู (Gateway) เชื่อมโยงติดต่อกับการทำงานอื่นๆ เพื่อให้เกิดการทำงานที่หลากหลาย โดยอาศัยพื้นฐานของระบบเครือข่าย หรืออาจกล่าวได้ว่าเป็นการพัฒนาโปรแกรมประยุกต์เพื่อทำงานร่วมกับเว็บเซิร์ฟเวอร์ โปรแกรมที่พัฒนาขึ้นโดยใช้แนวคิดของซีจีไอ นั้นจะเป็นโปรแกรมที่ทำงานบนฝั่งเซิร์ฟเวอร์ การเกิดขึ้นของเทคโนโลยีซีจีไอนี้ทำให้โปรแกรมบนระบบเครือข่ายมีการพัฒนาไปจากเดิมอย่างมาก เนื่องจากสามารถพัฒนาโปรแกรมให้โต้ตอบกับผู้ใช้ได้

การทำงานของซีจีไอใช้หลักการของไคลเอ็นต์ - เซิร์ฟเวอร์ โดยเว็บเซิร์ฟเวอร์จะเป็นผู้ติดต่อขอใช้บริการและรอผลของซีจีไอกลับมา จากนั้นจึงส่งต่อไปให้กับผู้ใช้ที่ฝั่งไคลเอ็นต์ ลักษณะการทำงานของซีจีไอต้องอาศัยการประมวลผลที่เซิร์ฟเวอร์ แล้วสร้างคำตอบออกมาในรูปแบบของเอชทีเอ็มแอล จากนั้นจึงส่งเนื้อหากลับไปให้ไคลเอ็นต์ เซิร์ฟเวอร์ใดที่ยอมรับให้มีการรันซีจีไอได้จึงต้องทำงานหนักกว่าเซิร์ฟเวอร์ที่ให้บริการเอกสารเอชทีเอ็มแอลเพียงอย่างเดียว แนวคิดการทำงานของซีจีไอจะเป็นแบบรวมศูนย์ (Centralize) งานทุกอย่างต้องเข้ามารันที่เซิร์ฟเวอร์หมดไคลเอ็นต์เพียงแต่ทำหน้าที่ส่งคำร้องขอ และรอรับผลการดำเนินงานเท่านั้น ซึ่งขัดกับหลักการงานแบบกระจายศูนย์ (Distribute) ที่พยายามลดการเข้าไปรันงานในเซิร์ฟเวอร์ และให้ไคลเอ็นต์ทำงานให้มากที่สุด



รูปที่ 3.8 รูปแสดงการร้องขอข้อมูลผ่านซีจีไอ

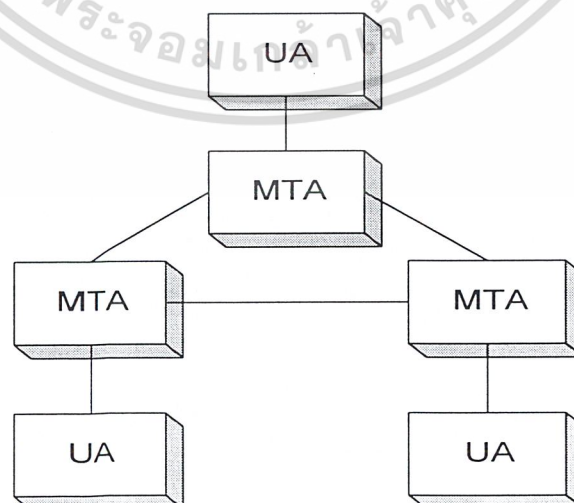
### 3.6 เอสเอ็มทีพีโปรโตคอล (Simple Mail Transfer Protocol)

#### 3.6.1 สถาปัตยกรรมระบบเมลล์

ไปรษณีย์อิเล็กทรอนิกส์ ซึ่งนิยมเรียกกันสั้นๆว่า อีเมล (e-mail) เป็นโปรแกรมประยุกต์ที่แพร่หลายในระบบเครือข่าย ประโยชน์ของอีเมลส์ช่วยให้ผู้ใช้ส่งและรับข้อความข้ามเครือข่ายได้ ทีซีพี/ไอพี มีโปรโตคอลสนับสนุนการรับส่งเมลล์หลายโปรโตคอล แต่โปรโตคอลที่นิยมใช้ในอินเทอร์เน็ตคือ เอสเอ็มทีพี หน้าที่ของเอสเอ็มทีพีคือ กำหนดกรรมวิธีและรูปแบบการนำส่งข้อความระหว่างผู้รับและผู้ส่ง เอสเอ็มทีพีอาศัยทีซีพีเพื่อลำเลียงจดหมายผ่านพอร์ตเบอร์ 25

ระบบเมลล์ที่ใช้ในทีซีพี/ไอพี มีองค์ประกอบ 2 ส่วน คือ ยูเอ (User Agent) และเอ็มทีเอ (Mail Transfer Agent) ทั้งยูเอและเอ็มทีเอเป็นชื่อที่นำมาจากระบบ X.400 ซึ่งเป็นมาตรฐานนานาชาติกำหนดการนำส่งเมลล์

ยูเอเป็นโปรแกรมติดต่อกับผู้ใช้ และอำนวยความสะดวกให้ผู้ใช้ เขียน แก้ไขและส่งจดหมาย รวมทั้งการเปิดอ่านจดหมายที่ได้รับ และจัดเก็บจดหมายเพื่อนำมาใช้ในภายหลัง ส่วนเอ็มทีเอทำหน้าที่หาเส้นทางและส่งจดหมายไปถึงปลายทาง การติดต่อระหว่างเอ็มทีเอใช้พอร์ตเบอร์ 25 โดยแลกเปลี่ยนข้อมูลตามรูปแบบที่กำหนดในอาร์เอฟซี 822 รูปที่ 3.9 แสดงถึงสถาปัตยกรรมทั่วไปของระบบเมลล์



รูปที่ 3.9 รูปแสดงสถาปัตยกรรมทั่วไปของระบบเมลล์ในทีซีพี/ไอพี

การจัดแบ่งออกเป็นยูเอและเอ็มทีเอ็มที่ดีคือ แยกงานของทั้งสองส่วนให้เป็นอิสระจากกัน หน้าที่ของยูเอเน้นการทำงานกับผู้ใช้เพื่อให้ผู้ใช้สามารถอ่านเขียนจดหมายได้อย่างสะดวกโดยไม่ต้องยุ่งเกี่ยวกับการทำงานระดับล่างของโปรโตคอล ส่วนเอ็มทีเอ็มทำงานตามเอสเอ็มทีพี เช่นการตรวจสอบความถูกต้องของแอดเดรสของผู้รับและผู้ส่ง รวมทั้งการหาเส้นทางและนำส่งจดหมายไปยังปลายทาง

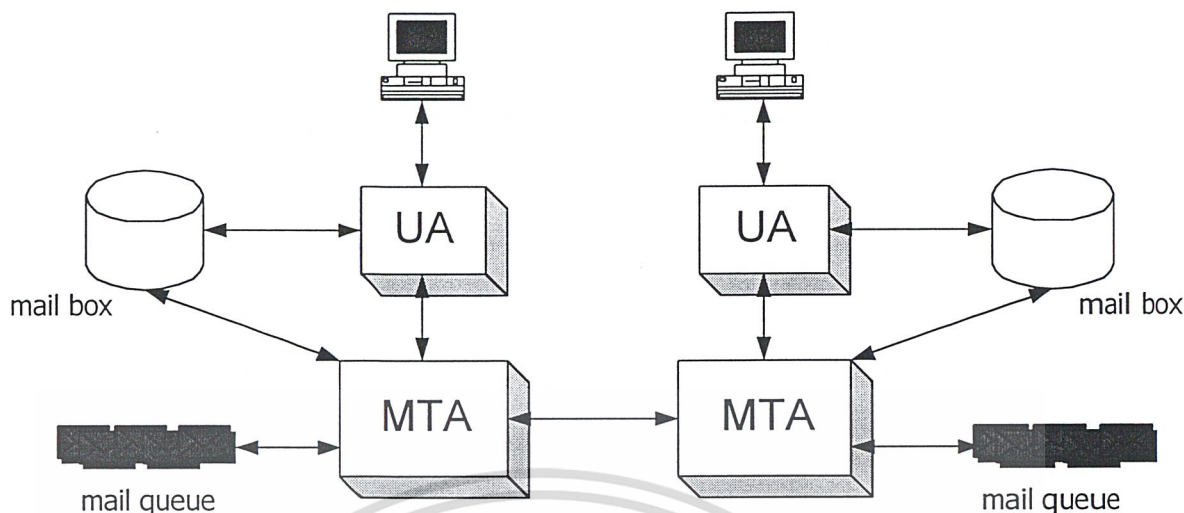
เมื่อผู้ใช้ส่งจดหมาย หน้าที่ของยูเอคือส่งจดหมายไปยังเอ็มทีเอ็มเพื่อให้เอ็มทีเอ็มนำส่งต่อไป เอ็มทีเอ็มต้นทางอาจติดต่อกับเอ็มทีเอ็มปลายทางโดยตรง หรือใช้วิธีรีเลย์ (relay) โดยส่งต่อเป็นทอดๆ จากเอ็มทีเอ็มต้นทางไปยังเอ็มทีเอ็มระหว่างทางซึ่งจะเก็บเมลไว้ และนำส่งต่อตามจังหวะเวลาที่เหมาะสมจนกระทั่งเมลไปถึงปลายทาง ระบบเมลที่ใช้ส่งต่อเป็นทอดๆนี้เรียกว่า ระบบเก็บและส่งต่อ(store-and-forward-system) เมลล์ีเลย์ประจำโดเมนหนึ่งๆเรียกว่า ตัวแลกเปลี่ยนเมลล์ (mail exchanger)

การใช้เมลล์ีเลย์มีข้อดีหลายประการเช่น

- ผู้ใช้งานพีซีทั่วไปที่ไม่มีเอ็มทีเอ็มก็ไม่ได้เปิดเครื่องใช้งานอยู่ตลอดเวลา เมื่อมีเมลเข้ามาจำเป็นต้องอาศัยเมลล์ีเลย์เป็นตัวเก็บพักเมลไว้จนกว่าจะเปิดใช้พีซีเพื่อขอถ่ายเมลล์มาจากเมลล์ีเลย์
- เครื่องข่ายในหลายองค์กรใช้เมลล์ีเลย์ทำหน้าที่ติดต่อกับเครื่องข่ายภายนอก เมลล์ีเลย์อาจเป็นจุดเดียวที่อนุญาตให้รับส่งเมลล์โดยตรงกับภายนอกได้ โดยมีระบบไฟร์วอลล์ห้ามเครื่องอื่นภายในเครือข่ายรับส่งเมลล์โดยตรง เพื่อที่องค์กรจะได้สร้างระบบเมลล์ศูนย์กลางและจะได้ไม่ให้ชื่อเครื่องอื่นภายในเครือข่ายแพร่ออกไปภายนอก
- การติดตั้งเอ็มทีเอ็มอย่างเช่น sendmail ในยูนิกซ์มีความซับซ้อน ผู้ดูแลระบบบางแห่งจะไม่ติดตั้งเอ็มทีเอ็มอะไรๆไปแต่ให้ใช้บริการผ่านเมลล์ีเลย์แทน

### 3.6.2 ยูนิกซ์เมลล์

ยูนิกซ์เมลล์ีเลย์มีรูปแบบการทำงานเช่นเดียวกับที่ได้กล่าวไปแล้ว เมื่อยูเอนำส่งเมลล์ ยูเออาจนำส่งเมลล์ออกไปทันทีหรือเอาเก็บพัก (spool) ไว้ในหน่วยความจำสำรองก่อนเพื่อรอการนำส่ง (เช่นใน /var/spool/mqueue) การเก็บพักช่วยให้เอ็มทีเอ็มเจ็ดลำดับ และบริหารการนำส่งเมลล์ได้อย่างเป็นระบบ โดยปกติแล้วเอ็มทีเอ็มจะนำส่งเมลล์ไปยังปลายทางในทันทีแต่ถ้าการส่งล้มเหลว เอ็มทีเอ็มจะจัดเก็บเมลล์ไว้ในคิวเพื่อรอส่งใหม่ และควรทดลองส่งซ้ำเป็นช่วงอย่างน้อยช่วงละ 30 นาทีจนกว่าจะส่งได้หรือยกเลิกการส่งเพราะปัญหาปลายทางไม่สามารถรับเมลล์ได้มักเกิดขึ้นเพียงช่วงระยะเวลาสั้นๆเท่านั้น กรณียกเลิกการส่งควรพยายามส่งมาแล้วไม่น้อยกว่า 4-5 วัน



รูปที่ 3.10 รูปแสดงรูปแบบของยูนิคซ์เมลล์

รูปที่ 3.10 แสดง โครงสร้างของยูนิคซ์เมลล์ ผู้ส่งเรียกใช้เมลล์ผ่านทางยูเอ เช่น pine หรือ mail และ เอ็มทีเอฝ่ายส่งติดต่อกับเอ็มทีเอฝ่ายรับผ่านที่ซีพีพอร์ตเบอร์ 25 จดหมายที่ไปถึงปลายทางจะถูกเก็บอยู่ในตู้ไปรษณีย์ประจำตัวผู้ใช้ หรือเมลล์บ็อกซ์ (mail box) ในยูนิคซ์เก็บเมลล์บ็อกซ์ในรูปเพิ่มข้อมูล ผู้ใช้แต่ละรายจะมีแฟ้มนี้เป็นของตนเอง (มักอยู่ใน /var/mail หรือ /var/spool/mail)

### 3.6.3 รูปแบบจดหมาย

โครงสร้างของเมลล์ประกอบด้วยส่วนสำคัญ 2 ส่วน คือ ส่วนหัวจดหมาย และส่วนเนื้อความจดหมาย หัวจดหมายซึ่งอยู่ส่วนต้นประกอบด้วยข้อความแสดงข่าวสารเกี่ยวกับการรับส่งจดหมาย ส่วนเนื้อความคือข้อความที่ได้รับซึ่งจะอยู่ต่อจากหัวจดหมาย เอสเอ็มทีพีกำหนดรูปแบบของหัวจดหมายไว้เพื่อให้ใช้เป็นหลักมาตรฐาน หัวจดหมายแต่ละฉบับอาจมีรายละเอียดมากน้อยต่างกันไป

ตัวอย่างต่อไปนี้ เป็นจดหมายที่มีหัวจดหมายสั้นๆ ที่มีรายละเอียดบ่งบอกว่าใครเป็นผู้ส่ง (From:) เมื่อเวลาใด (Date:) ส่งถึงใคร (To:) ใครบ้างที่ได้รับสำเนาจดหมาย (Cc:) และหัวเรื่องจดหมาย (Subject:)

```
From: "Thanawat Kaewka" <kaewka_thanawat@hotmail.com>
To: s1014172@diamond.ce.kmitl.ac.th
Subject: hello
Date: Fri, 08 Mar 2002 06:49:44 +0000
```

### 3.6.4 ป๊อป และไอแมป์ (POP and IMAP Protocol)

โดยปกติแล้วเจ้าของบัญชีสามารถอ่านจดหมายโดยล็อกอินเข้าที่เซิร์ฟเวอร์ที่เก็บจดหมายอยู่ แล้วใช้โปรแกรมที่ใช้อ่านจดหมาย เช่น โปรแกรม mail เปิดอ่านจดหมายหรืออีกวิธีหนึ่ง คือการเข้าใช้จดหมายแบบรีโมต

การเข้าใช้จดหมายแบบรีโมต เป็นการติดต่อขออ่านจดหมายวิธีหนึ่งที่ผู้ขอบริการ ไม่จำเป็นต้องล็อกอินเข้าเซิร์ฟเวอร์ที่เก็บจดหมายแต่สามารถอ่านจดหมายได้จากเครื่องใดก็ได้ที่มีโปรแกรมสนับสนุนโปรโตคอลอ่านจดหมายแบบรีโมต

ข้อดีของระบบนี้คือ จำนวนความสะกดกรณีเครื่องไคลเอ็นต์ไม่มีโปรแกรมเทเลเน็ตเพื่อล็อกอินเข้าเซิร์ฟเวอร์ อีกทั้งระบบจะนำจดหมายมาเก็บที่ไคลเอ็นต์โดยไม่จำเป็นต้องเชื่อมกับเซิร์ฟเวอร์ตลอดเวลาที่อ่านจดหมาย รูปแบบนี้มีประโยชน์อย่างมากสำหรับผู้ใช้อินเทอร์เน็ตแบบจ่ายค่าบริการตามระยะเวลาการใช้งาน

โปรโตคอลที่ใช้ในการเข้าถึงจดหมายจากระยะไกลมี 2 โปรโตคอลซึ่งนิยมใช้อย่างแพร่หลายได้แก่ ป๊อป (POP : Post Office Protocol) และไอแม็ป (IMAP : Internet Message Access Protocol) ทั้งป๊อปและไอแม็ปต่างก็เป็นบริการสำคัญที่ใช้แพร่หลายในอินเทอร์เน็ต ที่เซิร์ฟเวอร์ส่วนใหญ่มีให้บริการ

ป๊อปเป็นโปรโตคอลที่เก่ากว่า ส่วนไอแม็ปเป็นโปรโตคอลที่พัฒนาขึ้นมาภายหลังและเพิ่มเติมขีดความสามารถในการทำงานให้มากขึ้น ไอแม็ปรุ่นล่าสุดได้แก่ IMAP4 ส่วนป๊อปคือ POP3

### ป๊อป (POP)

ป๊อปได้รับการออกแบบมาให้ทำงานแบบออฟไลน์และทำงานโดยใช้ที่ซีพียูพอร์ตเบอร์ 110 การดำเนินการใดๆกับจดหมายจะเกิดกับจดหมายที่อ่านมาเก็บไว้ที่ไคลเอ็นต์เท่านั้น โดยสรุปแล้วหน้าที่หลักของ ป๊อปคือ การส่งถ่ายจดหมายจากเซิร์ฟเวอร์มายังไคลเอ็นต์ นอกจากนี้ป๊อปยังสามารถลบจดหมายที่เก็บไว้ที่เซิร์ฟเวอร์ได้อีกด้วย

### ไอแม็ป (IMAP)

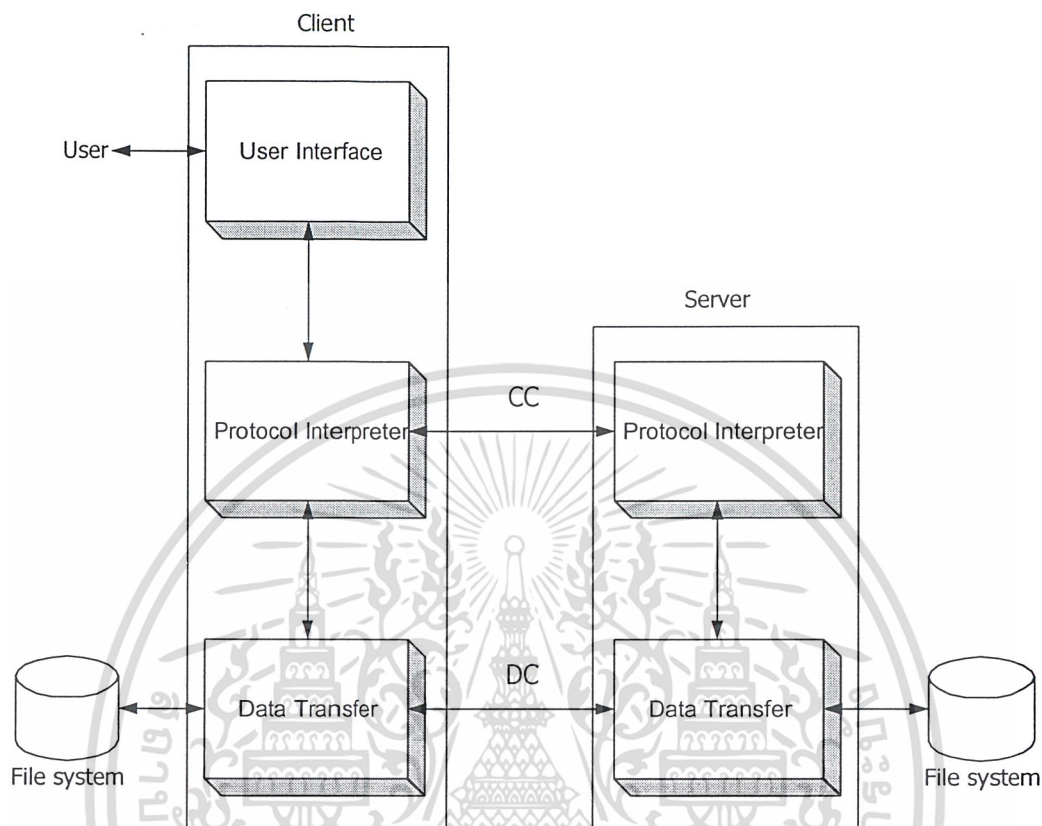
ไอแม็ปมีความสามารถในการเข้าถึงแบบออนไลน์หรือเป็นแบบโต้ตอบกับเซิร์ฟเวอร์ ผู้ใช้สามารถดึงเฉพาะหัวข้อของจดหมาย หัวจดหมาย หรือค้นหาจดหมายที่ตรงตามความต้องการ และสามารถเปลี่ยนแปลงสถานะจดหมายที่เซิร์ฟเวอร์ได้ ซึ่งโดยสรุปก็คือ ไอแม็ปได้รับการออกแบบมาให้เข้าถึงจดหมายจากระยะไกลเหมือนกับการเข้าถึงจดหมายจากภายในเครื่องของผู้ใช้ ไอแม็ปทำงานโดยใช้ที่ซีพียูพอร์ตเบอร์ 143

### 3.7 โปรโตคอลเอฟทีพี (File Transfer Protocol)

อาร์เอฟซี 959 นิยามแบบจำลองการทำงานของเอฟทีพีซึ่งทำงานตามแบบไคลเอ็นต์เซิร์ฟเวอร์ เอฟทีพีเป็นโปรโตคอลที่ค่อนข้างซับซ้อน เนื่องจากการสถาปนาที่ซีพีขึ้นสองส่วนคือ ส่วนรับส่งคำสั่งและส่วนถ่ายโอนข้อมูล

เอฟทีพีแต่ละด้านจะมีตัวแปลโปรโตคอล (Protocol Interpreter: PI) ต่อเชื่อมผ่านส่วนเชื่อมต่อควบคุม (Control Connection: CC) ซึ่งเป็นช่องทางส่งคำสั่ง ตัวแปลโปรโตคอลมีหน้าที่แปลคำสั่งและดำเนินการตามคำสั่งนั้น ส่วนเชื่อมต่อที่สองคือ ส่วนเชื่อมต่อข้อมูล (Data Connection: DC) เป็นช่องทาง

ส่งข้อมูลซึ่งอยู่ภายใต้การดูแลของ โมดูลถ่ายโอนข้อมูล(Data Transfer : DT) แบบจำลองเอพีทีพีแสดงได้ ดังรูปที่ 3.11



รูปที่ 3.11 รูปแสดงแบบจำลองการทำงานของเอพีทีพี

หากพิจารณาเพิ่ม /etc/services ในยูนิกซ์ จะพบว่าเอพีทีพีใช้พอร์ต 21 ส่งผ่านคำสั่งควบคุม และพอร์ต 20 ส่งข้อมูล ส่วนเชื่อมโยงควบคุมนั้นจะคงอยู่ตลอดการเชื่อมต่อ ในขณะที่ส่วนเชื่อมโยงข้อมูลจะเกิดขึ้นเมื่อต้องถ่ายโอน และปิดลงเมื่อเสร็จสิ้นการถ่ายโอน แต่ครั้งของการถ่ายโอนจะมีการสร้างส่วนเชื่อมโยงข้อมูลขึ้นใหม่ทุกครั้ง

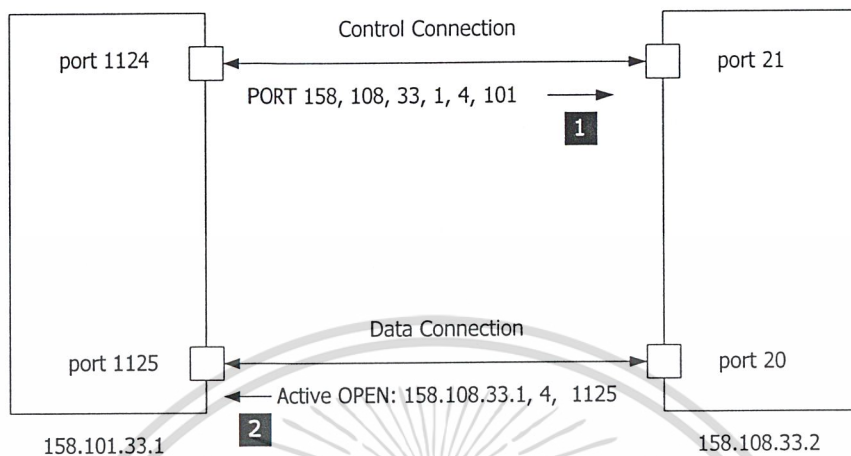
การสร้างส่วนเชื่อมโยงข้อมูลอยู่ภายใต้การควบคุมของไคลเอ็นต์ เพราะไคลเอ็นต์ต้องส่งคำสั่งเพื่อขอถ่ายโอน ไคลเอ็นต์จะจัดเตรียมพอร์ตเพื่อรอสถาปนาส่วนเชื่อมโยงข้อมูล และแจ้งไปยังเซิร์ฟเวอร์ (เปิดแบบพาสซีฟ) ด้วยคำสั่ง PORT ผ่านส่วนเชื่อมโยงควบคุมเมื่อเซิร์ฟเวอร์ได้รับคำสั่งแล้วก็จะสถาปนาที่ซีพีไปยังไคลเอ็นต์ตามพอร์ตที่ได้รับ

รูปที่ 3.12 แสดงการสร้างส่วนเชื่อมโยงข้อมูลระหว่างไคลเอ็นต์ และเซิร์ฟเวอร์ ในที่นี้สมมติให้พอร์ตประจำส่วนเชื่อมโยงควบคุมของไคลเอ็นต์คือ 1124 และเตรียมพอร์ต 1125 รอไว้สำหรับส่วนเชื่อมโยงข้อมูล

ไคลเอ็นต์จะขอเปิดส่วนเชื่อมโยงข้อมูลตามตำแหน่ง 1 โดยส่งรหัสคำสั่ง PORT ตามด้วยอาร์กิวเมนต์หกตัวแยกด้วยจุลภาคคือไอพีแอดเดรส (158, 108, 33, 1) และหมายเลขพอร์ต 4, 101 ซึ่งแสดงถึงพอร์ต 4, 101 ซึ่งหมายถึงพอร์ต 1125 (เลขพอร์ตเป็นรหัส 16 บิต 2 ชุดติดกัน ดังนั้นตัวเลข 4, 101 คือ 1125)

ต่อจากนั้นเซิร์ฟเวอร์จะสถาปนาที่ซีพีจากพอร์ต 20 ไปยังไคลเอ็นต์ที่พอร์ต 1125 ตามตำแหน่ง

ที่ 2



รูปที่ 3.12 รูปแสดงการสร้างส่วนเชื่อมโยงข้อมูล



## บทที่ 4

### ภาษาไพธอน (Jython)

ไพธอนคือการรวมกันของ 2 ภาษา ได้แก่ ภาษาจาวา และภาษาไพธอน ซึ่งทั้ง 2 ภาษาได้รับความนิยมด้วยเหตุผลที่ต่างกันคือ จาวามีคลาสไลบรารี และเอกสารต่างๆสนับสนุนอยู่มากมายรวมทั้งจาวาในปัจจุบันเป็นที่นิยมจนมีบางองค์กรสร้างโปรแกรมประยุกต์ภายในองค์กรทั้งหมดด้วยภาษาจาวา ส่วนภาษาไพธอนนั้นมีข้อดีที่ความยืดหยุ่น และความเร็วในการพัฒนาโปรแกรมประยุกต์

ไพธอนเป็นภาษาที่ใช้เขียนโปรแกรมโดยสมบูรณ์ ไม่ใช่ java translator และไม่ใช่ python compiler เลยทีเดียวแต่เป็นการสร้างภาษาไพธอนในจาวาแพลตฟอร์ม ภาษาไพธอนได้รับไลบรารีโมดูลมาจากไพธอน(ไพธอนโดยปกติหมายถึงซีไพธอน) เกือบทั้งหมด และที่สำคัญคือภาษาไพธอนไม่เหมือนภาษาระดับสูงทั่วไปคือภาษาไพธอนอนุญาตให้ผู้ใช้เข้าถึงทุกอย่างของภาษาที่เขียนมัน ซึ่งหมายถึงภาษาไพธอนสามารถใช้ คลาสไลบรารีทั้งหมดของจาวาได้

ภาษาไพธอนเป็นภาษาที่มีการทำงานแบบอินเทอร์พรีเตอร์ มีความยืดหยุ่นสูง สามารถนำมาใช้งานได้หลากหลาย มีความสามารถในการจัดการกับสายอักขระ และความสามารถในการใช้คำสั่ง exec ซึ่งสามารถทำการประมวลผลคำสั่งได้ในขณะรันไทม์

#### 4.1 คุณสมบัติของภาษาไพธอน

ภาษาไพธอนมีคุณสมบัติ และมีความสามารถหลายประการดังนี้

- **High-level** ภาษาไพธอนมีโครงสร้างข้อมูลระดับสูงซึ่งช่วยลดระยะเวลาในการพัฒนาโปรแกรมลง เช่น Python's list(Resizable arrays) และ Dictionary(Hash table) ซึ่งการสร้างสิ่งเหล่านี้ในภาษาอื่นๆ เช่น ในภาษา C ทำได้ยากเนื่องจากความจำเป็นที่ต้องใช้โครงสร้างข้อมูล และ pointer ในภาษา C ข้อดีของการมีโครงสร้างข้อมูลระดับสูงคือ ลดระยะเวลาในการพัฒนาโปรแกรม, ขนาดของโปรแกรม และทำให้โปรแกรมอ่านง่ายซึ่งจะส่งผลให้การ debug โปรแกรมทำได้โดยง่าย
- **Object-oriented** ภาษาไพธอนถูกสร้างมาเป็นภาษาเชิงวัตถุ(Object Oriented Language) เช่นเดียวกับ C++ และ JAVA และสามารถสร้างคลาสย่อยจากภาษาทั้งสองได้อีกด้วย จึงเหมาะแก่การเป็นภาษาสคริปต์ให้กับภาษาทั้งสอง
- **Scalable** ภาษาไพธอนมีการเขียนซอร์สโค้ดที่เข้าใจง่าย มีโครงสร้างระดับสูง และสามารถสร้างแพ็คเกจที่บรรจุหลายๆคอมโพเน้น ซึ่งสิ่งเหล่านี้สนับสนุนการออกแบบโปรแกรมเป็นบล็อกของโปรแกรม และเมื่อต้องการจะขยายโปรแกรมโครงสร้างที่เป็นโมดูล และ pluggable ของไพธอนทำให้สามารถจัดการขยายได้โดยง่าย

- Extensible** การเขียนโปรแกรมในภาษาไพธอนสามารถแบ่งซอร์สโค้ดออกเป็นโมดูลย่อย ซึ่งโมดูลย่อยเหล่านี้ และโมดูลใน standard library ของ ไพธอน ผู้เขียนโปรแกรมสามารถเรียกมาใช้ได้ นอกจากนี้ยังสามารถห่อ(wrapping) lower-level code โดยใช้ไพธอน interface ซึ่งทำให้สามารถสร้าง และ import โมดูลที่คอมไพล์แล้วเข้ามาใช้งานได้ โดยวิธีการเรียกใช้งานยังคงเหมือนโมดูลปกติในไพธอนแต่ประสิทธิภาพในการทำงานจะดีขึ้น ซึ่งความสามารถนี้ทำให้ไพธอนสามารถ extend โมดูลจากภาษาอื่นได้ด้วย เช่น ภาษาJava สำหรับ JPython
- Portable** ภาษาไพธอนถูกพัฒนาขึ้นมาจากภาษา C โปรแกรมที่เขียนจากภาษาไพธอน จะสามารถใช้ได้กับระบบปฏิบัติการทั่วไป โดยตัวแปลภาษาจะแปลคำสั่งให้เป็น bytecode ที่ระบบปฏิบัติการนั้นเข้าใจ
- Easy-to-learn** ภาษาไพธอนมีคีย์เวิร์ดน้อย มีโครงสร้างภาษาที่เรียบง่าย และมี syntax ที่เข้าใจง่ายทำให้สามารถเรียนรู้ได้ง่าย และรวดเร็ว
- Easy-to-read** syntax ของภาษาไพธอนไม่มีการใช้สัญลักษณ์ที่มักพบในภาษาอื่นๆ ที่ใช้สำหรับในการเข้าถึงตัวแปร, การกำหนดสโคปของ Code block และการทำ pattern matching เช่น dollar sign( \$ ), semicolons( ; ), tildes( ~ ) ซึ่งการที่ภาษาไพธอนหลีกเลี่ยงการกระทำดังกล่าวทำให้ซอร์สโค้ดของภาษาไพธอนสามารถอ่านทำความเข้าใจได้ง่าย
- Easy-to-maintain** เนื่องจากภาษาไพธอนมีคุณสมบัติ easy-to-learn และeasy-to-read ทำให้เกิดข้อดีคือ มีความซับซ้อนน้อยลง จึงทำให้การบำรุงรักษา และการพัฒนาต่อทำได้โดยง่าย
- Robust** ภาษาไพธอนมีการจัดการข้อผิดพลาดที่ดี โดยเมื่อไพธอนเกิดมีการทำงานที่ผิดพลาดขึ้น ก็จะมี stack trace ซึ่งจะระบุว่าเกิดข้อผิดพลาดขึ้นที่ไหน และอย่างไรบ้าง และภาษาไพธอนยังมีความสามารถในการให้ผู้เขียนโปรแกรมเพื่อจัดการกับข้อผิดพลาด(Exception handler) อีกด้วย
- Effective as a Rapid Prototyping Tool** การเขียนโปรแกรมบนระบบใดๆโดยใช้ภาษาไพธอนเพียงอย่างเดียวสามารถทำได้ ถึงแม้ว่า compiled language อื่นๆ ก็สามารถทำได้เช่นกัน แต่ไพธอนมี library จำนวนมากให้ใช้ทำให้ลดระยะเวลาในการพัฒนาได้ ซึ่ง library ที่มีในตัวแปลภาษาอยู่แล้ว เช่น networking, Internet/Web/CGI, graphics and graphical user interface (GUI) development(Tkinter), imaging(PIL), numerical computation and analysis(Numpy), database access, hypertext(HTML,XML,SGML)
- A Memory Manger** ข้อเสียที่สำคัญของการเขียนโปรแกรมในภาษา C และC++ คือ การที่ผู้เขียนโปรแกรมต้องรับผิดชอบในการจัดการกับหน่วยความจำเอง แต่ในภาษาไพธอนตัวแปลภาษาจะทำหน้าที่นี้ให้โดยอัตโนมัติ ทำให้ผู้เขียนโปรแกรมไม่ต้องมากังวลกับการจัดการหน่วยความจำ ซึ่งส่งผลดีคือ ทำให้โปรแกรมมีข้อผิดพลาดน้อยลง และระยะเวลาในการพัฒนาน้อยลง
- Interpreted and (Byte-) Compiled** ไพธอนเป็นภาษาที่มีการทำงานแบบInterpreter ทำให้การทำงานช้ากว่าภาษาที่ใช้การ compile เนื่องจากการประมวลผลคำสั่งไม่ได้ทำในรูปแบบของภาษา

เครื่องของระบบ(system's native binary language) แต่ภาษาไพธอนก็สามารถทำ byte-compiled ได้ซึ่งผลจากการ compile จะอยู่ในรูปแบบที่ใกล้เคียงกับภาษาเครื่องทำให้ประสิทธิภาพการทำงานดีขึ้น และยังมีความเป็นภาษาที่มีการทำงานแบบ Interpreter อยู่

	Execution Speed	Coding Speed	Object-Oriented	GUI Coding	Dev Environment	Suitability for large tasks	Libraries available
Python	Fair	Excellent	Excellent	Good	Fair	Excellent	Good
Perl	Fair	Excellent	Fair	Good	Fair	Fair	Excellent
Vis.Basic	Good	Excellent	Fair	Excellent	Excellent	Poor	Fair
C	Excellent	Poor	Poor	n/a	Excellent	Good	Good
C++	Excellent	Fair	Excellent	n/a	Excellent	Excellent	Good
Java	Fair	Good	Excellent	Good	Excellent	Excellent	Good

ตารางที่ 4-1 ตารางเปรียบเทียบคุณสมบัติต่างๆ ของภาษาไพธอนกับภาษาอื่นๆ

#### 4.2 ชนิดของข้อมูล

ชนิดของข้อมูลในภาษาไพธอนจะอยู่ในอ็อบเจ็กต์ที่ชื่อว่า ไพธอนอ็อบเจ็กต์ การกำหนดชนิดของข้อมูลให้กับตัวแปรในภาษาไพธอนจะถูกกระทำโดยอัตโนมัติ ผู้เขียนโปรแกรมสามารถเปลี่ยนแปลงตัวแปรหนึ่งให้เก็บค่าที่ต่างไปได้โดยไม่ต้องสนใจชนิดของตัวแปรที่เก็บอยู่เดิม ชนิดของข้อมูลทั่วไปได้แก่

##### ตัวเลข (Numeric)

ข้อมูลชนิดตัวเลข ได้แก่

- เลขฐานสิบ (Integer) คือเลขที่ประกอบไปด้วยเลข 0-9 ไม่ว่าจะป็นจำนวนบวก หรือจำนวนลบแต่ต้องไม่มีจุดทศนิยม ตัวอย่างเช่น 1, -3, 8784 ฯลฯ
- เลขฐานแปดคือ ตัวเลขที่ขึ้นต้นด้วย 0 เช่น 0715 ฯลฯ
- เลขฐานสิบหกคือ ตัวเลขที่ขึ้นต้นด้วย 0x เช่น 0x9FAC, 0x12A ฯลฯ
- เลขทศนิยม(Floats) เช่น 3.2, -45e12 ฯลฯ
- เลขจำนวนเต็มขนาดยาว(Long integer) คือตัวเลขที่ลงท้ายด้วยตัว L หรือ l เช่น 4890549579L, 5l ฯลฯ
- เลขจำนวนเชิงซ้อน(Complex number) เช่น 9j, 5i-4j, 45i ฯลฯ

##### จำนวนทางตรรกะ (Boolean)

โดยปกติจำนวนทางตรรกะจะมีค่าเท็จ หรือจริง สำหรับภาษาไพธอนจะแทนค่าเท็จด้วย เลขศูนย์, โครงสร้างเปล่า หรือ None value เช่น 0, [], {}, (), None และแทนค่าจริงด้วย ค่าใดๆที่ไม่ใช่ศูนย์, โครงสร้างที่มีข้อมูลอยู่ เช่น 1, [5], (7,8,98,347), “xyz”

### สายอักขระ (String)

ข้อมูลชนิดสายอักขระ คือ ข้อมูลชนิดของตัวอักษร เช่น ‘This is a string.’, “This is another string” สายอักขระนี้จะอยู่ภายในสัญลักษณ์ Single quoted หรือ Double quoted ก็ได้ แต่ต้องเหมือนกันทั้งหัว และท้าย หากใช้ Single quoted ในสายอักขระจะสามารถมี Double quoted ได้ และในทำนองเดียวกัน หากใช้ Double quoted ก็จะสามารถมี Single quoted ในสายอักขระได้เช่นเดียวกัน

ในข้อมูลชนิดสายอักขระนั้นจะมีอักขระพิเศษอยู่ด้วย ได้แก่

\n = Newline    \' = Single quoted \b = Backspace

\t = Tab    \” = Double quoted    \f = Formfeed

\\ = Backslash    \a = Bell    \r = Carriage return

\v = Vertical tab

สำหรับการใช้ ‘’ หรือ “” นั้นจะเป็นสายอักขระที่ หากมีการขึ้นบรรทัดใหม่ในสายอักขระ หรือมีการย่อหน้าจะทำการเปลี่ยนให้เป็นตัวอักขระพิเศษ

### ลิสต์ (Lists)

ข้อมูลชนิดลิสต์นี้จะเป็นการนำเอาข้อมูลชนิดอื่นๆมาเก็บไว้ ข้อมูลชนิดนี้ โดยจะมีลักษณะคล้ายกับข้อมูลชนิดอาร์เรย์(Array) แต่จะต่างกันตรงที่ข้อมูลที่เก็บในลิสต์ไม่จำเป็นต้องเก็บข้อมูลชนิดเดียวกัน ข้อมูลที่สามารถเก็บในลิสต์ได้แก่ ตัวเลข, ตัวอักษร, สายอักขระ หรือแม้แต่จะเป็นลิสต์ด้วยตัวเองก็ได้ การเขียนอ้างอิงถึงข้อมูลชนิดลิสต์ทำได้ดังนี้

list1 = []

เป็นการสร้างลิสต์เปล่า

list2 = [1,“two”,[3,4]]

เป็นการสร้างลิสต์ที่ประกอบด้วยสมาชิก 3 ตัว โดยตัวแรก

เป็น ตัวเลข, ตัวที่สองเป็นสายอักขระ, ตัวที่สามเป็น

ลิสต์ที่มีสมาชิกเป็นตัวเลขสองตัว

การอ้างอิงสมาชิกในลิสต์ทำได้ดังนี้

list2[0]

จะได้ผลลัพธ์เป็น 1

list2[1:2]

จะได้ผลลัพธ์เป็น “two”

list2[1:]

จะได้ผลลัพธ์เป็น [“two”,[3,4]]

list2[-1]

จะได้ผลลัพธ์เป็น [3,4]

### ทUPLE (Tuples)

ข้อมูลชนิดทUPLE มีลักษณะเหมือนข้อมูลชนิดลิสต์ แต่จะต่างกันตรงที่ไม่สามารถแก้ไขข้อมูลของสมาชิกภายในทUPLE ได้ ทUPLE จึงเหมาะแก่การใช้เป็นข้อมูลอ้างอิง การกำหนดทUPLE มีวิธีการดังนี้

```
tuple1 = ()
```

เป็นการสร้างทUPLE เปล่า

```
tuple2 = ("one",2,"three",4)
```

การจัดการกับทUPLE นั้นมีวิธีการเช่นเดียวกับการจัดการกับลิสต์

### ดิกชันนารี (Dictionary)

ข้อมูลชนิดดิกชันนารีจะประกอบไปด้วยคีย์(Keys) และค่าที่เก็บ(Values) ในการอ้างอิงข้อมูลประเภทดิกชันนารีทำได้ดังนี้

dic[key]	จะให้ผลลัพธ์เป็นอ็อบเจกต์ที่ถูกเก็บ โดยคีย์นั้น
dic.has_key(key)	จะส่งค่ากลับมาเป็น 1 เมื่อมีคีย์ที่ระบุอยู่ใน dic และ 0 เมื่อไม่มีคีย์ที่ระบุอยู่ใน dic
dic.keys()	ส่งค่ากลับมาเป็นลิสต์ของคีย์ต่างๆที่มีอยู่ใน dic
dic.values()	ส่งค่ากลับมาเป็นลิสต์ของค่าที่เก็บต่างๆที่มีอยู่ใน dic
dic.clear()	ทำการลบทุกค่าที่มีอยู่ใน dic

โดยปกติข้อมูลชนิดลิสต์จะสามารถเรียงลำดับข้อมูลได้ แต่ข้อมูลชนิดดิกชันนารีนั้นจะไม่มี การเรียงลำดับข้อมูล การอ้างอิงทำได้โดยผ่านทางคีย์เท่านั้น

### ข้อมูลเปล่า (None)

ข้อมูลชนิดนี้ใช้ในการระบุค่าเริ่มต้นของตัวแปร หรือแสดงว่าไม่มีข้อมูล ซึ่งในการเปรียบเทียบค่า นั้น None จะมีค่าเท่ากับ None เท่านั้น

## 4.3 ตัวกระทำ (Operators)

ตัวกระทำต่างๆแบ่งออกเป็น 4 ประเภทคือ ตัวกระทำทางตรรกะ, ตัวกระทำทางการเปรียบเทียบ, ตัวกระทำทางบิตไวด์ และตัวกระทำทางคณิตศาสตร์

### ตัวกระทำทางตรรกะ (Logical operators)

ตัวกระทำทางตรรกะมีทั้งหมด 3 ตัวด้วยกันคือ and, or, not โดยทั้ง 3 ตัวกระทำนี้สามารถใช้งาน ได้ด้วยการเขียน and,or และ not โดยตรง

### ตัวกระทำทางการเปรียบเทียบ (Comparison operators)

ตัวกระทำที่ใช้เปรียบเทียบค่าต่างๆมีดังต่อไปนี้

<	น้อยกว่า	>	มากกว่า
<=	น้อยกว่าหรือ เท่ากับ	>=	มากกว่าหรือ เท่ากับ
=	เท่ากับ	<>,!=	ไม่เท่ากับ(สามารถใช้ได้ทั้ง 2 แบบ)

in	$x \text{ in } y$ หมายความว่า มีค่า $x$ เป็นสมาชิกใน $y$ หรือไม่
not in	ให้ผลตรงข้ามกับ in
is	$x \text{ is } y$ หมายความว่า $x$ เป็นสิ่งเดียวกับ $y$ ซึ่งไม่เหมือนกับ $==$
is not	ให้ผลตรงข้ามกับ is

### ตัวกระทำทางบิตไวด์ (Bitwise operators)

<<	<code>integer&lt;&lt;n</code> ส่งค่ากลับเป็นตัวเลขที่เกิดจากการเลื่อนบิตของ <code>integer</code> ไปทางซ้าย (Shift left) ไปเป็นจำนวน $n$ บิต
>>	ทำงานเช่นเดียวกับ << แต่เลื่อนไปทางขวา (Shift right)
&	<code>m&amp;n</code> ผลลัพธ์คือการ and กัน ในรูปตัวเลขฐานสองของ $m$ และ $n$
	<code>m n</code> ผลลัพธ์คือการ or กัน ในรูปตัวเลขฐานสองของ $m$ และ $n$
^	<code>m^n</code> ผลลัพธ์คือการ xor กัน ในรูปตัวเลขฐานสองของ $m$ และ $n$
~	<code>~m</code> ผลลัพธ์คือการ not ในรูปตัวเลขฐานสองของ $m$

### ตัวกระทำทางคณิตศาสตร์ (Arithmetic-Style operators)

ตัวกระทำทางคณิตศาสตร์นี้สามารถใช้งานไม่เพียงกับตัวเลขเท่านั้น แต่ยังสามารถใช้งานกับข้อมูลชนิดอื่นๆ เช่น สายอักขระได้ ตัวกระทำดังกล่าวได้แก่

+	หากใช้กับตัวเลขจะเป็นการบวกกัน แต่หากใช้กับสายอักขระจะเป็นการนำมาเรียงต่อกัน และหากเป็นลิสต์ หรือทUPLE จะเป็นกรรวมสมาชิกของสองลิสต์เข้าเป็นลิสต์เดียว หรือสมาชิกของสองทUPLE เป็นทUPLE เดียว
-	ใช้กับตัวเลขเพื่อทำการลบกัน
*	หากใช้กับตัวเลขจะเป็นการคูณกัน แต่หากใช้กับสายอักขระ, ลิสต์ หรือทUPLE จะเป็นกรเพิ่มจำนวนของข้อมูลนั้น เช่น <code>'M'*5</code> จะได้ <code>'MMMMM'</code> เป็นต้น
/	ใช้กับตัวเลขเพื่อทำการหาร โดยหากใช้กับจำนวนเต็มผลที่ได้ก็จะเป็นจำนวนเต็ม เศษที่ได้จะถูกปัดลง ตัวกระทำนี้เหมือนกับ <code>div</code> ในภาษา Pascal
**	ใช้กับตัวเลขเท่านั้นเป็นการยกกำลัง
%	หากใช้กับตัวเลขจะเป็นการหาเศษจากการหาร เหมือนการใช้คำสั่ง <code>mod</code> ในภาษา Pascal หากใช้กับสายอักขระ จะเป็นการบอกรูปแบบในการพิมพ์

### ลำดับของตัวกระทำ (Precedence)

ลำดับของตัวกระทำมีลำดับดังนี้

or

and

not

<, <=, ==, >=, >, !=, <>, is, in, not, not in

|

^

&

<<, >>

+, -

\*, /, %

\*\*

unary+, unary-, unary~

#### 4.4 คำสงวน (Reserved words)

คำสงวนในภาษาไพธอนแบ่งเป็น คีย์เวิร์ด(Keywords) และบิวท์อินฟังก์ชัน(Built-in function) คีย์เวิร์ด ได้แก่

'and', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'exec', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'not', 'or', 'pass', 'print', 'raise', 'return', 'try', 'while'

บิวท์อินฟังก์ชันได้แก่

'\_\_import\_\_', 'abs', 'apply', 'buffer', 'callable', 'chr', 'cmp', 'coerce', 'compile', 'complex', 'delattr', 'dir', 'divmod', 'eval', 'execfile', 'filter', 'float', 'getattr', 'globals', 'hasattr', 'hash', 'hex', 'id', 'input', 'int', 'intern', 'isins', 'issubclass', 'len', 'list', 'locals', 'long', 'map', 'max', 'min', 'oct', 'open', 'ord', 'pow', 'range', 'raw\_input', 'reduce', 'reload', 'repr', 'round', 'setattr', 'slide', 'str', 'tuple', 'type', 'vars', 'xrange'

#### 4.5 สโคปของโค้ด (Code block)

การกำหนดสโคปของซอร์สโค้ดใช้การย่อหน้า(Indentation) แทนสัญลักษณ์ ซึ่งต่างจากภาษาอื่นๆ เช่น ภาษาC และ Java ใช้วงเล็บปีกกา และภาษา Pascal ใช้ begin end เป็นตัวกำหนดสโคปของซอร์สโค้ด

การที่ไพธอนใช้การย่อหน้าในการกำหนดสโคปของซอร์สโค้ดทำให้ช่วยลดความยุ่งยาก และข้อผิดพลาดที่มักเกิดขึ้นจากการลืมพิมพ์สัญลักษณ์ที่ใช้ในการกำหนดสโคป และข้อดีอีกอย่างหนึ่งคือ ทำให้

ซอร์สโค้ดอ่านง่ายเนื่องจากทุกครั้งที่ขึ้นสโคปใหม่จะต้องทำการย่อหน้าซึ่งทำให้ซอร์สโค้ดมีความเป็นระเบียบ

#### 4.6 คอมเมนต์ (Comment)

สำหรับการคอมเมนต์ นั้นจะใช้ '#' เป็นตัวระบุ โดยข้อความที่อยู่หลัง '#' ไปจนสุดบรรทัดจะไม่ถูกประมวลผลโดยตัวแปลภาษาไพธอน

#### 4.7 การพิมพ์ statement เดียวในหลายบรรทัด (Continuation)

เนื่องจาก statement ในภาษาไพธอนจะถูกแบ่งโดยการขึ้นบรรทัดใหม่(NEWLINE) ดังนั้นหนึ่ง statement จะต้องใช้หนึ่งบรรทัด แต่สามารถใช้เครื่องหมาย backslash(\) ในการพิมพ์ statement เดียวในหลายบรรทัดได้ดังนี้

```
if (weather_is_hot == 1) and \
    (available == 1):
    goto_the_beach()
```

#### 4.8 การพิมพ์หลาย statement ในบรรทัดเดียว

ใช้เครื่องหมาย semicolon(;) ในการพิมพ์หลาย statement ในบรรทัดเดียว เช่น

```
import sys; x = 'hello'; sys.stdout.write(x+'\n')
```

#### 4.9 การกำหนดค่าให้ตัวแปร (Variable assignment)

การกำหนดค่าให้ตัวแปรใช้เครื่องหมายเท่ากับ(=) ซึ่งการกำหนดค่าให้ตัวแปรนั้นไม่ได้เป็นการนำไปใส่ให้ตัวแปรจริงๆ แต่เป็นการกำหนดให้ตัวแปรอ้างอิงไปยังไพธอนอ็อบเจกต์ที่มีค่านั้นอยู่ โดยไพธอนอ็อบเจกต์นั้นอาจถูกสร้างขึ้นมาใหม่ตอนที่กำหนดค่า หรือเป็นไพธอนอ็อบเจกต์ที่มีอยู่แล้วจากตัวแปรอื่นก็ได้

##### 4.9.1 การกำหนดค่าหนึ่งค่าให้กับหลายตัวแปร (Multiple assignment)

```
x = y = z = 1
```

ตัวอย่างด้านบนเป็นการกำหนดค่าหนึ่งค่าให้กับหลายตัวแปร โดยอ็อบเจกต์ชนิดจำนวนเต็มที่มีค่า 1 จะถูกสร้างขึ้น และตัวแปร x, y, z จะถูกกำหนดให้อ้างอิงไปยังอ็อบเจกต์นี้ ซึ่งนี่คือขั้นตอนการกำหนดค่าหนึ่งอ็อบเจกต์ให้กับหลายตัวแปร

##### 4.9.2 การกำหนดค่าหลายค่าให้กับหลายตัวแปร

```
x, y, z = 1, 2, 'three'
```

ตัวอย่างด้านบนเป็นการกำหนดค่าหลายค่าให้กับหลายตัวแปรโดยผลลัพธ์ที่ได้คือ x, y และ z มีค่าเท่ากับ 1, 2 และ 'three' ตามลำดับ การกำหนดค่าแบบนี้ทั้งสองข้างของเครื่องหมายเท่ากับคือทาบเปิด ซึ่งตัวอย่างด้านบนสามารถเขียนได้อีกอย่างคือ  $(x, y, z) = (1, 2, \text{'three'})$

การกำหนดค่าโดยใช้วิธีนี้ยังมีประโยชน์อีกอย่างในกรณีที่ต้องการสลับค่าตัวแปรซึ่งสามารถเขียนเพียง statement เดียวก็สามารถสลับค่าตัวแปรได้แล้ว เช่น

$x, y = y, x$       หรือ       $(x, y) = (y, x)$

แต่ถ้าใช้ภาษาอื่น เช่น ภาษา C ต้องเขียนดังนี้

temp = x

x = y

y = temp

#### 4.10 คำสั่งควบคุมการทำงาน (Control flow)

คำสั่งลูปเป็นคำสั่งที่ใช้ในการควบคุมลำดับการประมวลผลของคำสั่งต่างๆ โดยให้ประมวลผลวนรอบไปเรื่อยๆจนกว่าจะถึงเงื่อนไขที่กำหนด คำสั่งลูปต่างๆได้แก่ IF, WHILE, FOR

##### 4.10.1 คำสั่ง IF

รูปแบบของคำสั่ง IF มีดังนี้

If <EXPRESSION>:

<STATEMENT>

elif <EXPRESSION>:

<STATEMENT>

else:

<STATEMENT>

โดย <EXPRESSION> นั้นหมายถึงเงื่อนไขในการหยุดลูป ส่วน <STATEMENT> คือคำสั่งที่อยู่ในลูป ตัวอย่างของการใช้คำสั่ง IF เช่น

if x == 0:

print 'x equal 0'

elif x > 0:

print 'x is positive number'

else:

print 'x is negative number'

##### 4.10.2 คำสั่ง WHILE

คำสั่ง WHILE นั้นมีการตรวจสอบเงื่อนไขก่อนการทำคำสั่งในลูปทุกครั้ง โดยมีรูปแบบคือ

```
while <EXPRESSION> :
    <STATEMENT>
```

ตัวอย่างคำสั่ง WHILE เช่น

```
while 1:
    Print 'Infinite loop'
```

#### 4.10.3 คำสั่ง FOR

คำสั่ง FOR เป็นคำสั่งที่จะวนรอบการทำงานจนครบจำนวนที่กำหนด รูปแบบของคำสั่งคือ

```
For <VARIABLE> in <RANGE>:
```

```
    <STATEMENT>
```

โดย <RANGE> คือตัวแปรที่เป็นลิสต์ หรือคำสั่งเฉพาะบางคำสั่ง ส่วน <VARIABLE> คือตัวแปรที่เก็บค่าของสมาชิกทุกตัวของ<RANGE> การทำงานของคำสั่ง FOR จะทำงานวนรอบเป็นจำนวนเท่ากับจำนวนสมาชิกของ <RANGE> ตัวอย่างเช่น

```
list1 = [1,'a',[b,c]]
```

```
for var1 in list1:
```

```
    print var1
```

จากซอร์สโค้ดข้างต้น ได้ผลลัพธ์ดังนี้

```
1
```

```
'a'
```

```
[b,c]
```

การทำงานของคำสั่ง FOR ข้างต้นจะทำงานเป็นจำนวน 3 ครั้ง เท่ากับจำนวนสมาชิกใน list1

```
for i in range(10):
```

```
    print i
```

คำสั่ง range(n) จะส่งค่ากลับเป็นตัวเลขตั้งแต่ 0 จนถึง n-1 ผลลัพธ์ที่ได้คือ

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

#### 4.11 การจัดการหน่วยความจำ (Memory management)

ลักษณะเด่นของตัวแปร และการจัดการหน่วยความจำที่ผู้ใช้งานสามารถสังเกตได้มีดังนี้

- ตัวแปรจะถูกประกาศตอนให้ค่าครั้งแรก
- ไม่มีการกำหนดชนิดของตัวแปร
- ผู้เขียน โปรแกรมไม่ต้องเขียน โปรแกรมจัดการกับหน่วยความจำ
- ชื่อตัวแปรที่เคยประกาศใช้ไปแล้วสามารถนำมาใช้ใหม่ได้ถึงแม้ว่าจะมีชนิดของตัวแปรที่ต่างกัน
- สามารถใช้คำสั่ง del เพื่อทำการปลดปล่อยหน่วยความจำอย่างเปิดเผยในการเขียนโค้ดได้

#### การประกาศตัวแปร (Variable Declarations)

ตัวแปรจะถูกประกาศตอนให้ค่าครั้งแรกด้วยโอเปอเรเตอร์เท่ากับ (Assignment operation) โดยไม่ต้องมีการกำหนดชนิดของตัวแปร

#### การจองหน่วยความจำ (Memory Allocation)

ผู้เขียน โปรแกรมไม่ต้องจัดการเกี่ยวกับการจอง และการปลดปล่อยหน่วยความจำ เนื่องจากไพธอนจะจัดการกับการทำงานในระดับต่ำ (Lower-level) เช่นนี้ให้อยู่แล้ว

#### การรวบรวมหน่วยความจำที่ไม่ได้ใช้ (Garbage Collection)

หน่วยความจำที่ไม่ได้ใช้ ระบบจะมีกลไกในการรวบรวมหน่วยความจำเหล่านี้กลับมาใช้ใหม่ โดยที่ผู้เขียน โปรแกรมไม่ต้องเขียน โค้ดในส่วนนี้เนื่องจากไพธอนจะใช้การนับการอ้างอิง (Reference counting) ในการตัดสินใจว่าหน่วยความจำส่วนไหนที่ไม่ได้ใช้อีกต่อไปอีกแล้ว

#### การนับการอ้างอิง (Reference counting)

ไพธอนจะมีตัวแปรที่ใช้สำหรับนับการอ้างอิงที่กระทำไปยังไพธอนอ็อบเจ็กต์ โดยค่าเริ่มต้นของตัวแปรนี้คือ หนึ่ง เมื่อตอนประกาศตัวแปร และเมื่อไพธอนอ็อบเจ็กต์ถูกอ้างอิงอีก หรือถูกใช้เป็นอาร์กิวเมนต์ในการเรียกใช้ฟังก์ชัน หรือเมธอด ค่านี้ก็จะถูกเพิ่มขึ้นอีกหนึ่ง และเมื่อไรที่ค่านี้มีค่าเป็นศูนย์ก็จะหมายถึงไม่มีการอ้างอิงอ็อบเจ็กต์นี้อีกแล้ว และหน่วยความจำในส่วนนี้ก็จะถูกนำกลับมาใช้งานใหม่

```
foo1 = 'foobar' # reference count = 1
```

```
foo2 = foo1 # reference count = 2
```

```
Check_value(foo1) # reference count = 3 ตอนเรียกใช้ และหลังจากใช้เสร็จก็จะกลับมาเท่าเดิม
```

จากตัวอย่างด้านบนเมื่อประกาศตัวแปรครั้งแรก reference count ของสายอักขระ 'foobar' จะมีค่าเท่ากับหนึ่ง และเมื่อถูกอ้างอิงโดย foo2 ค่า reference count ก็จะถูกเพิ่มเป็นสอง และในตอนที่เรียกใช้ฟังก์ชันก็ จะถูกเพิ่มเป็นสาม แต่พอจบจากฟังก์ชันก็จะถูกลดไปเป็นสอง และถ้าใช้คำสั่ง del กับ foo2 ค่า reference count ก็จะถูกลดเป็นหนึ่ง และเมื่อ foo1 หลุดออกจาก scope ค่า reference count ก็จะถูกลดลงเหลือศูนย์ และหน่วยความจำในส่วนนั้นก็จะถูกนำกลับมาใช้งานใหม่

### คำสั่ง del

คำสั่ง del มีรูปแบบดังนี้

del obj

ตัวอย่างเช่น del foo2 จะทำให้เกิดผลสองอย่างดังนี้คือ

1. ลบชื่อ foo2 ออกจาก namespace
2. ลดค่า reference count ของอ็อบเจ็กต์ที่มันอ้างอิง ('foobar') ลงหนึ่ง

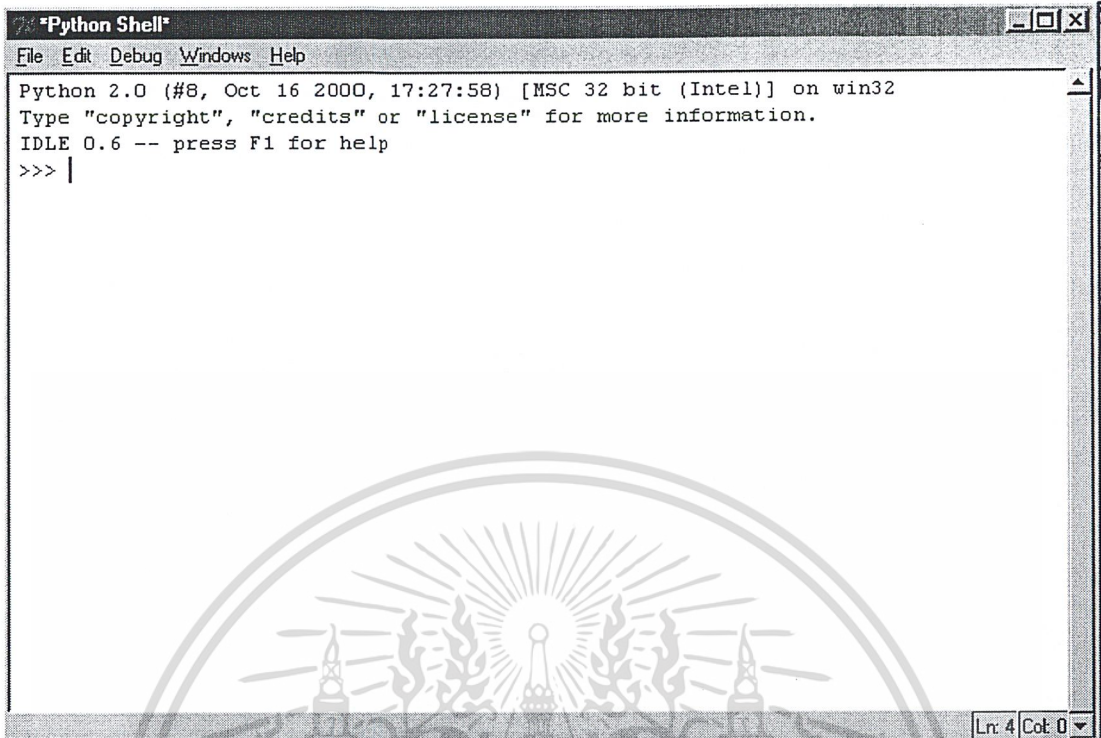
### การลดค่า reference count

ค่า reference count จะถูกลดลงเมื่อเกิดเหตุการณ์ต่อไปนี้

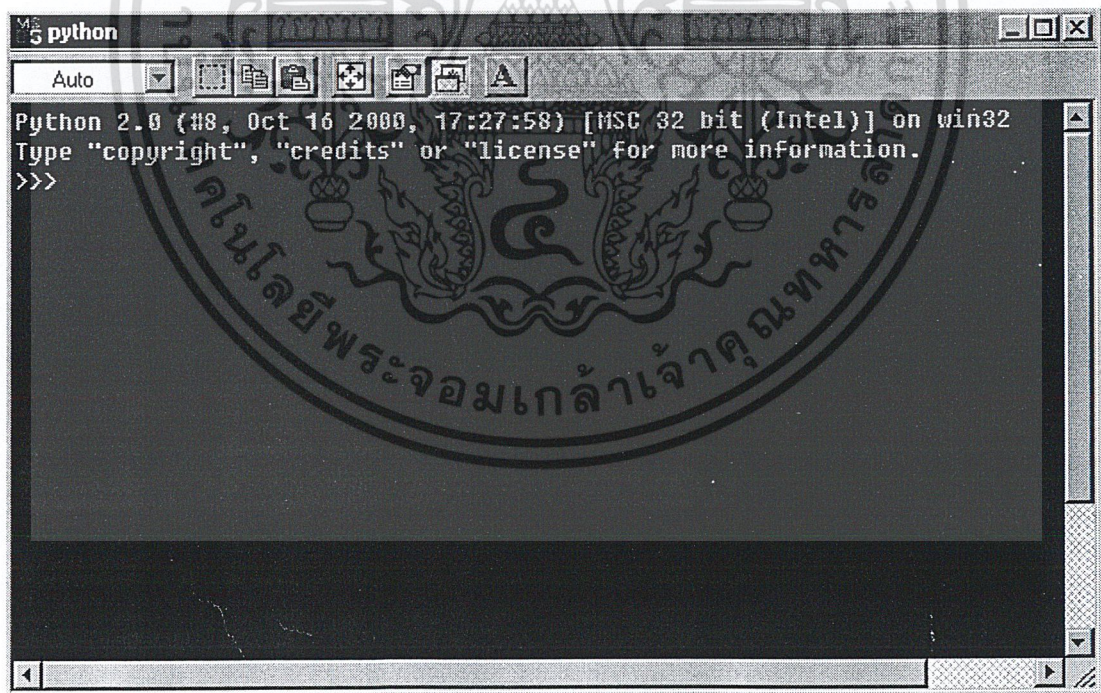
1. ใช้คำสั่ง del
2. ตัวแปรถูกเปลี่ยนการอ้างอิงจากอ็อบเจ็กต์เดิมไปยังอ็อบเจ็กต์ใหม่ ซึ่งอ็อบเจ็กต์เดิมนั้นก็จะถูกลดค่า reference count ลงหนึ่ง
3. หลุดออกจาก scope

## 4.12 การใช้งานภาษาไพธอน

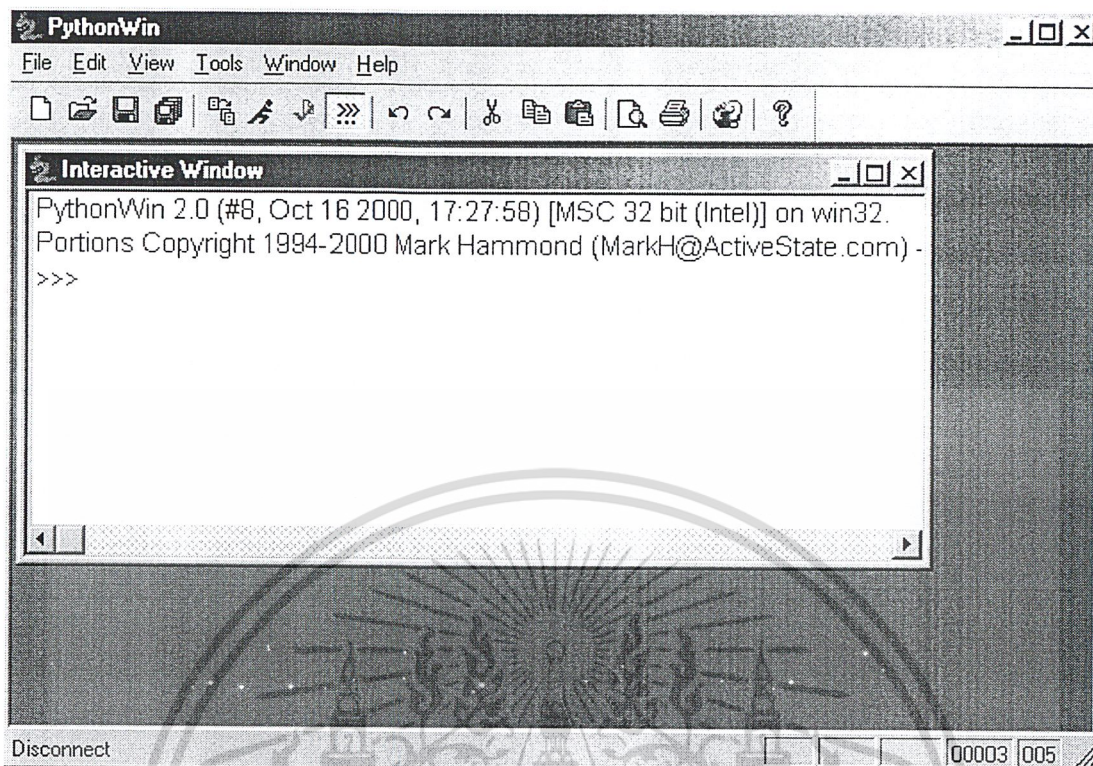
ภาษาไพธอนที่โครงการเลือกใช้คือภาษาไพธอนเวอร์ชัน 2.0 ในชุดของโปรแกรมจะประกอบไปด้วย คู่มือการใช้งานเป็นเอกสาร HTML, และโปรแกรมหน้าจอโต้ตอบซึ่งมีให้เลือกใช้ 2 อย่างคือ IDLE ซึ่งเป็นไพธอน GUI และไพธอนคอมมานด์ไลน์ซึ่งเป็นหน้าจอโต้ตอบอยู่บน DOS แต่ในการเขียนโปรแกรมจริงนั้นทางผู้จัดทำได้ใช้ pythonwin ซึ่งเป็นหน้าจอโต้ตอบที่ทำงานได้อย่างสมบูรณ์แบบบนวินโดวส์ซึ่งต้องทำการติดตั้งเพิ่มเติมเข้าไปในภายหลัง



รูปที่ 4-1 รูปแสดงโปรแกรมไพธอน IDLE



รูปที่ 4-2 รูปแสดงโปรแกรมไพธอนคอนโซล



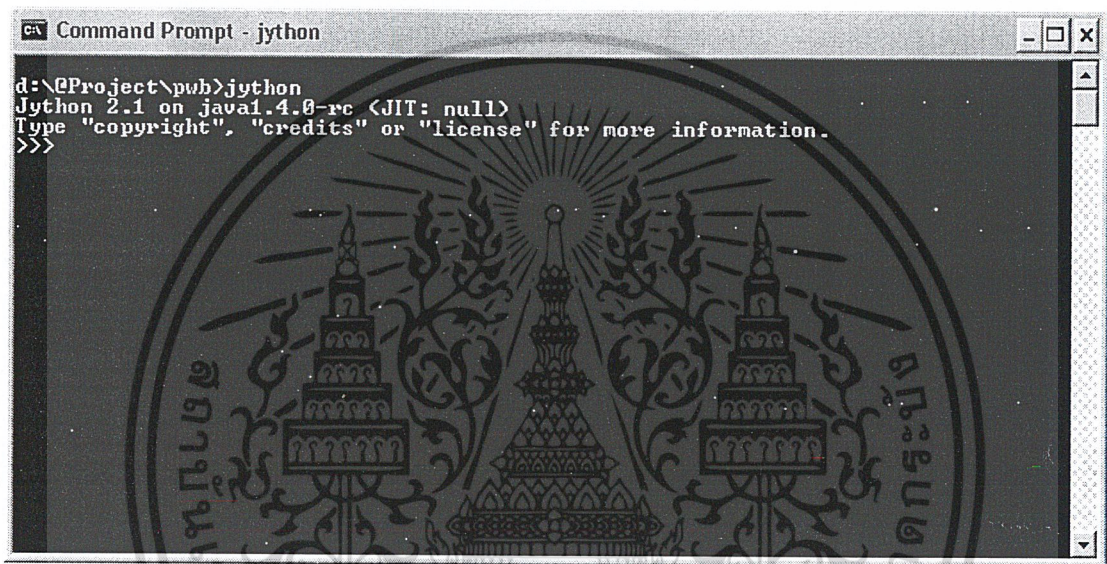
รูปที่ 4-3 รูปแสดงโปรแกรม pythonwin

#### 4.13 ภาษาไจรอน(Jython : Python in Java)

ภาษาไจรอนมีจุดเด่นที่มันดึงเอาเฉพาะคุณสมบัติที่ดีของทั้งจาวา และไพธอนมา ซึ่งลักษณะ และข้อดีที่สำคัญของภาษาไจรอนมีดังนี้

- Seamless Access to Java Clases ภาษาไจรอนสามารถใช้คลาสไลบรารีทั้งหมดของจาวาได้โดยตรง โดยการอิมพอร์ตเข้ามาใช้ได้โดยเช่นเดียวกับไพธอน โมดูล
- Dynamic type ไม่มีการประกาศชนิดของข้อมูลในภาษาไจรอน เพราะชนิดข้อมูลจะถูกกำหนดในขณะรันไทม์ นอกจากนั้นตัวแปรใดๆในภาษาไจรอนจะเป็นออบเจ็กต์ซึ่งมีคุณสมบัติพอลิมอร์ฟิก (Polymorphic) คือสามารถทำงานเป็นชนิดข้อมูลใดๆก็ได้ดังนั้นจึงสามารถสร้างอินสแตนซ์ของคลาสในภาษาจาวาได้โดยไม่ต้องประกาศไทป์ ทั้งคุณสมบัติไดนามิกไทป์ และพอลิมอร์ฟิกจะช่วยลดจำนวนโค้ด และลดความซับซ้อนในการเขียนโปรแกรม
- Introspection and Dynamic execution ภาษาไจรอนมีบิวท์อินฟังก์ชันซึ่งช่วยในการหาข้อมูลของออบเจ็กต์ และไดนามิกเอ็กซีคิวชันคือความสามารถในการเอ็กซีคิวโค้ดที่ถูกสร้างขึ้นในขณะรันไทม์
- First-Class function คือความสามารถในการพาสชื่อฟังก์ชันเหมือนกับการพาสตัวแปร ซึ่งเฟิร์ทคลาสฟังก์ชันจะมีประโยชน์ในการเขียนโค้ดเพื่อจัดการกับอีเวนต์ต่างๆในการเขียนยูเซอร์อินเตอร์เฟส

- Write once run anywhere เนื่องจากภาษาไพธอนถูกเขียนมาจากภาษาจาวา และภาษาไพธอนถูกคอมไพล์เป็นจาวาไบนารีโค้ด ดังนั้นมันจึงสามารถรันบนJVM ใดๆได้โดยที่จะต้องติดตั้งไพธอนหรือสร้างจาวาไคลว์ไฟล์ที่มี jython.jar อยู่ด้วย
- Code clarity เป็นข้อดีที่สำคัญของภาษาไพธอน ทำให้โค้ดกระจ่างไม่มีการใช้เครื่องหมายวรรคตอนมากเกินไป เช่น โค้ดบล็อกจะใช้การย่อหน้า และการขึ้นบรรทัดใหม่หมายถึงสเตทเมนต์ใหม่ ซึ่งสิ่งเหล่านี้ทำให้โค้ดเป็นระเบียบ ง่ายต่อการแก้ไข และพัฒนาต่อ
- Unicode an internationalization ภาษาไพธอนใช้รูปแบบของยูนิโคดเหมือนกับภาษาจาวา



รูปที่ 4-4 รูปแสดงโปรแกรมไพธอนก่อนโซล

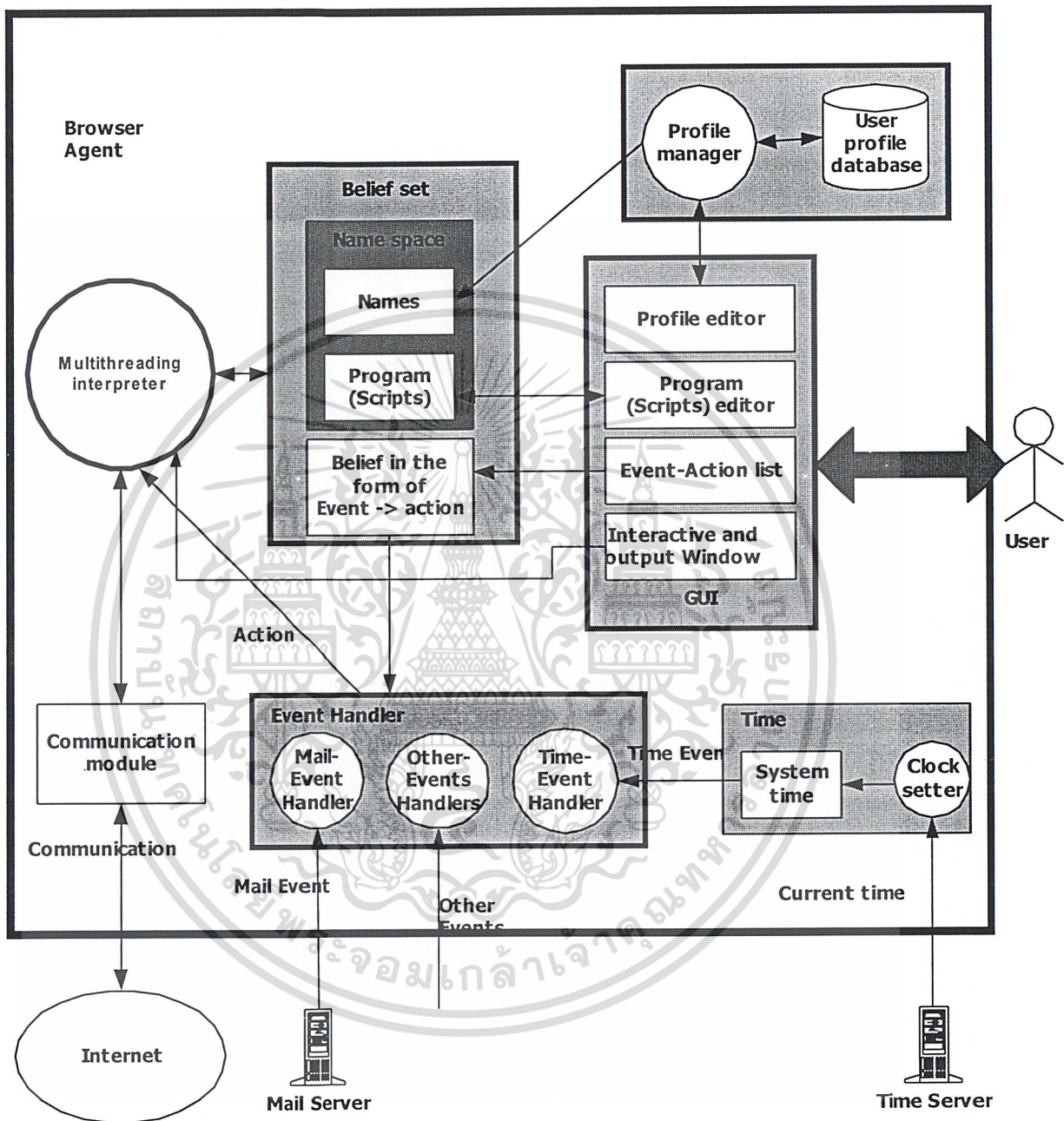
## บทที่ 5

### หลักการของบราวเซอร์ที่สามารถโปรแกรมได้

หลักการของบราวเซอร์ที่สามารถโปรแกรมได้ คือโปรแกรมเอเจนต์จะต้องทำงานตามที่ผู้ใช้งาน และจะต้องสามารถตอบสนองต่อสิ่งแวดล้อมตามที่ผู้ใช้งานกำหนดได้ ซึ่งการทำงานตามที่ผู้ใช้งานสามารถทำได้โดยให้ผู้ใช้งานเป็นคำสั่งสคริปต์แล้วให้โปรแกรมเอเจนต์ประมวลผลสคริปต์นั้น ส่วนการตอบสนองต่อสิ่งแวดล้อมนั้นผู้ใช้งานสามารถทำได้โดยการกำหนดอีเวนต์ และแอ็กชันโดยเป็นการกำหนดว่าเมื่ออีเวนต์เกิดขึ้นแล้วจะให้ทำแอ็กชันอะไร ซึ่งในการทำงานเอเจนต์จะมีฐานความรู้เป็นของตนเองซึ่งผู้ใช้งานสามารถเพิ่มเติมความรู้ใหม่ๆเข้าไปให้เอเจนต์ได้

การที่จะทำให้โปรแกรมเอเจนต์มีความสามารถดังกล่าวจำเป็นต้องใช้ภาษาที่มีความยืดหยุ่นสูง สามารถทำการประมวลผลได้อย่างไดนามิกได้ในขณะรันไทม์ มีไวยากรณ์ที่เรียบง่าย และสามารถที่จะจัดการควบคุมเนมสเปซ (Name space) ได้เป็นอย่างดีซึ่งโปรแกรมภาษาไพธอนมีความสามารถดังกล่าว และเนื่องจากภาษาจาวาเป็นที่นิยมใช้อย่างแพร่หลายในปัจจุบันและมีคลาสไลบรารีสนับสนุนอยู่จำนวนมากจึงเป็นประโยชน์อย่างมากในการพัฒนาโปรแกรมที่ใช้ในการติดต่อกับอินเทอร์เน็ต รวมทั้งจาวายังมีเครื่องมือที่ใช้ในการพัฒนาส่วนติดต่อกับผู้ใช้ที่มีประสิทธิภาพสูงอย่างสวิงให้ใช้งานอีกด้วย ดังนั้นการใช้ภาษาไพธอนซึ่งใช้ไวยากรณ์ที่เรียบง่ายของไพธอน และมีคลาสไลบรารีทั้งหมดของจาวาจึงเป็นการเหมาะสมที่สุดในการที่จะนำภาษานี้มาใช้ในการพัฒนาโปรแกรมเอเจนต์

## โครงสร้างและการทำงานของโปรแกรมเอเจนต์



รูปที่ 5.1 รูปแสดงโครงสร้างทั้งหมดของเว็บเอเจนต์

รูปที่ 5.1 แสดงโครงสร้างหลักของโปรแกรมเอเจนต์ซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้ คือ

1) ส่วนติดต่อกับผู้ใช้ (Graphical User Interface) — คือส่วนที่รับข้อมูลจากผู้ใช้เข้ามาเก็บหรือประมวลผลภายในเอเจนต์ซึ่งส่วนติดต่อกับผู้ใช้มีอยู่ทั้งหมด 4 ส่วนด้วยกันคือ

- หน้าต่างแก้ไขโพรไฟล์ของผู้ใช้ (Profile Dialog) – มีไว้สำหรับให้ผู้ใช้แก้ไขข้อมูลของตนเองที่เก็บขยายในเอเจนต์ เช่น ชื่อผู้ใช้ รหัสผ่าน อีเมลล์ของผู้ใช้ สคริปต์เริ่มต้นเมื่อรันสคริปต์ (Start up script)
- หน้าต่างเขียนสคริปต์ (Script editor) – มีไว้สำหรับให้ผู้ใช้เขียน แก้ไข เปิด และเซฟสคริปต์ โดยการกดปุ่มรันจะเป็นการรันสคริปต์ในหน้าต่างนี้ และนำข้อมูลที่ใช้สร้างขึ้นมา เช่น ฟังก์ชัน คลาส และ เนมต่างๆมาเก็บไว้ในเนมสเปซของเอเจนต์
- หน้าต่างรายการอีเวนต์ (Event-action list) – เป็นหน้าต่างที่มีไว้สำหรับตรวจสอบรายการของอีเวนต์ – แอ็กชันต่างๆ ที่อยู่ในเอเจนต์ว่ามีสถานะเป็นอย่างไร และยังสามารถลบ แก้ไข และเพิ่มรายการได้ผ่านทางเมนูหลัก
- หน้าต่างโต้ตอบ และแสดงผล (Interactive and output window) – เป็นหน้าต่างที่มีไว้สำหรับใส่คำสั่งทีละคำสั่งเพื่อให้เอเจนต์ประมวลผลโดยตรง และยังมีไว้สำหรับแสดงผล output ต่างๆของโปรแกรมเอเจนต์อีกด้วย

2) ส่วนโพรไฟล์ (Profile) คือส่วนที่ทำหน้าที่ในการเก็บข้อมูลลงในฐานข้อมูล ดึงข้อมูลของผู้ใช้ขึ้นมาแสดง และนำข้อมูลต่างๆที่ผู้ใช้กำหนดมาสร้างเป็นเนมภายในเนมสเปซของผู้ใช้แต่ละคน รวมทั้งยังเก็บค่าการดีฟอลต์ของเอเจนต์ไว้อีกด้วย ส่วนโพรไฟล์ประกอบด้วย 2 ส่วนคือ

- ฐานข้อมูลของผู้ใช้ (User profile database) – ใช้เก็บข้อมูลโพรไฟล์ต่างๆของผู้ใช้แต่ละคน เช่น ชื่อผู้ใช้ รหัสผ่าน อีเมลล์ของผู้ใช้ และค่าเซตคั้งต่างๆ
- ส่วนจัดการ โพรไฟล์ (Profile manager) - ทำหน้าที่ในการเก็บข้อมูลลงในฐานข้อมูล ดึงข้อมูลของผู้ใช้ขึ้นมาแสดง และนำข้อมูลต่างๆที่ผู้ใช้กำหนดมาสร้างเป็นเนมภายในเนมสเปซ

3) บีลีฟเซต (Belief set) คือ ส่วนข้อมูล หรือความรู้ของเอเจนต์ อันได้แก่ เนม และสคริปต์ซึ่งจะมีชื่ออยู่ในเนมสเปซของเอเจนต์ และอีเวนต์ – แอ็กชันที่เมื่อไรก็ตามที่อีเวนต์เกิดขึ้นแล้วเอเจนต์จะทำงานตามแอ็กชันนั้น บีลีฟเซตประกอบด้วย 3 ส่วน คือ

- เนม (Name) - คือชื่อที่ผู้ใช้อ้างอิงถึงข้อมูลต่างๆภายในเอเจนต์ซึ่งเนมในมุมมองของผู้ใช้ก็คือตัวแปรที่ใช้อ้างอิงถึงข้อมูลต่างๆนั่นเอง โดยที่ผู้ใช้สามารถเพิ่มเติมเนมเข้าไปเป็นฐานความรู้ให้แก่เอเจนต์ได้
- สคริปต์ (Script) – คือส่วนของโปรแกรมที่เอเจนต์มีอยู่ทั้งในรูปแบบของฟังก์ชัน และ คลาส ผู้ใช้สามารถเพิ่มเติมโปรแกรมเข้าไปเป็นฐานความรู้ให้แก่เอเจนต์ได้
- อีเวนต์ – แอ็กชัน (Event - action) – คือฐานความรู้ในอีกรูปแบบหนึ่งมีความหมายว่าเมื่ออีเวนต์เกิดขึ้นแล้วจะให้ทำแอ็กชันอะไร โดยมันจะถูกเก็บอยู่ในฐานความรู้ของเอเจนต์และถูกตรวจสอบอีเวนต์อยู่เสมอจนเมื่ออีเวนต์เป็นจริงแล้วก็จะทำแอ็กชัน และรายงานผลของแอ็กชันว่าเป็นเช่นไร

4) ส่วนจัดการอีเวนต์ (Event handler) คือ ส่วนที่ใช้ในการตรวจสอบอีเวนต์ในบีลีฟของเอเจนต์ ประกอบด้วย 3 ส่วนดังนี้

- ตัวจัดการอีเวนต์เวลา (Time-Event handler) - คือส่วนที่ทำหน้าที่ตรวจสอบอีเวนต์เวลาทั้งหมดในฐานความรู้โดยจะเปรียบเทียบกับเวลาของเอเจนต์ โดยเมื่อถึงเวลาที่กำหนด (กำหนดได้ละเอียดสุดถึงหน่วยนาทิจ) ก็จะทำแอ็กชันที่กำหนด
- ตัวจัดการอีเวนต์เมลล์ (Mail-Event handler) - คือส่วนที่ทำหน้าที่ตรวจสอบอีเวนต์เมลล์ทั้งหมดในฐานความรู้โดยเมื่อมีอีเมลฉบับใหม่เข้ามาในอีเมลแอ็กเคานท์ก็จะทำแอ็กชันที่กำหนด
- ตัวจัดการอีเวนต์อื่น ๆ (Other-Events handler) - คือส่วนที่ทำหน้าที่ตรวจสอบอีเวนต์ใดๆ ที่ผู้ใช้กำหนดขึ้นเอง ในรูปแบบของเอ็กเพรสชัน (Expression) ที่สามารถส่งค่ากลับมาในรูปแบบของถูก หรือผิด(true or false) ได้

5) อินเทอร์เน็ตแบบมัลติเธรด (Multithreading interpreter) ทำหน้าที่ในการประมวลผลโปรแกรมแบบขนานโดยไม่ต้องเสียเวลารอให้โปรแกรมปัจจุบันประมวลผลเสร็จก็สามารถประมวลผลโปรแกรมต่อไปได้ซึ่งการทำการประมวลผลเช่นนี้จะมีประโยชน์อย่างมากเนื่องจากโมดูลที่ใช้ติดต่อกับอินเทอร์เน็ตจำเป็นต้องเสียเวลาในการรอรับข้อมูลตอบกลับจากเซิร์ฟเวอร์ดังนั้นการทำงานแบบมัลติเธรดจะสามารถทำให้เอเจนต์สามารถทำงานอย่างอื่นไปได้โดยไม่ต้องเสียเวลาไปโดยเปล่าประโยชน์

6) โมดูลที่ใช้ติดต่อผ่านอินเทอร์เน็ต (Communication module) คือ โมดูลที่ทำหน้าที่ติดต่อขอใช้บริการที่สำคัญบนอินเทอร์เน็ต ประกอบด้วย โมดูลต่างๆ ดังนี้

- เว็บเบราว์เซอร์ (Web browser) - ใช้ในการดูเอกสารเอชทีเอ็มแอล โดยกรอกยูอาร์แอลให้กับเว็บเบราว์เซอร์จากนั้นเว็บเบราว์เซอร์จะร้องขอข้อมูลมาให้และแสดงผลที่หน้าจอ
- เอฟทีพีไคลเอนต์ (FTP Client) - ใช้ในการอัปโหลด(Upload) และดาวน์โหลด(Download)ไฟล์ระหว่างไคลเอนต์กับเอฟทีพีเซิร์ฟ
- เมลล์ไคลเอนต์ (Mail Client) - ใช้ในการดาวน์โหลดจดหมายจากเมลล์บ็อกซ์บนเซิร์ฟเวอร์มาไว้ที่ไคลเอนต์ผ่านโปรโตคอลป็อป 3 และใช้ในการส่งจดหมายผ่านโปรโตคอลเอสเอ็มทีพี
- เอสเอ็มเอสไคลเอนต์ (SMS Client) - ใช้ในการส่งข้อความสั้นไปยังโทรศัพท์มือถือโดยผ่านเครือข่ายของเอไอเอส

7) เวลา (Time) คือเวลาที่เอเจนต์ใช้อ้างอิงในการทำงานโดยผู้ใช้สามารถเลือกได้ว่าจะให้เวลานี้ถูกตั้งค่าใหม่ทุกครั้งเมื่อเริ่มใช้งานโปรแกรมเอเจนต์ (โดยอ้างอิงจากเครื่องเซิร์ฟเวอร์) หรือไม่ก็ได้ โดยหากมีการใช้เวลาจากเซิร์ฟเวอร์จะมีการเซตโดย ตัวตั้งค่าเวลา (Time Synchronizer) ซึ่งใช้ในการเซตเวลาให้กับเวลาระบบโดยที่ส่วนนี้จะทำหน้าที่ในการร้องขอข้อมูลเวลาจากเซิร์ฟเวอร์ แล้วนำมาเซตให้กับเวลาระบบ

### 5.1 ส่วนติดต่อกับผู้ใช้ (Graphical User Interface)

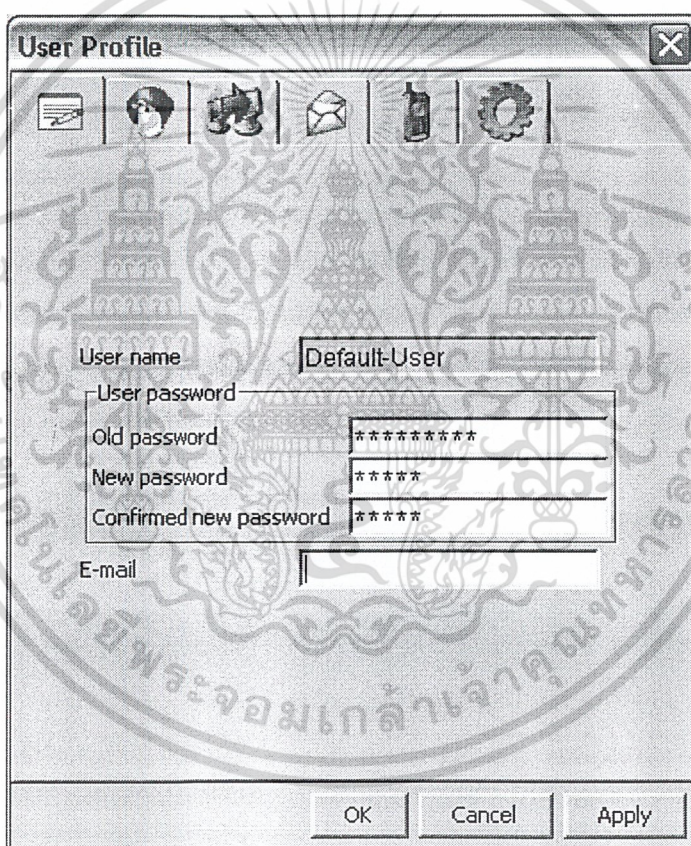
คือส่วนที่รับข้อมูลจากผู้ใช้เข้ามาเก็บหรือประมวลผลภายในเอเจนต์และเนื่องจากใช้ภาษาไพธอนจึงสามารถใช้สวิง (Swing) ซึ่งเป็นคลาสไลบรารีของภาษาจาวาในการทำส่วนติดต่อกับผู้ใช้ซึ่งมีความ

สามารถสูงกว่าที่เคอเดอร์เฟส (Tkinter) ของภาษาไพธอนเป็นอย่างมาก  
กับผู้ใช้มีอยู่ทั้งหมด 4 ส่วนด้วยกันคือ

ส่วนติดต่อ

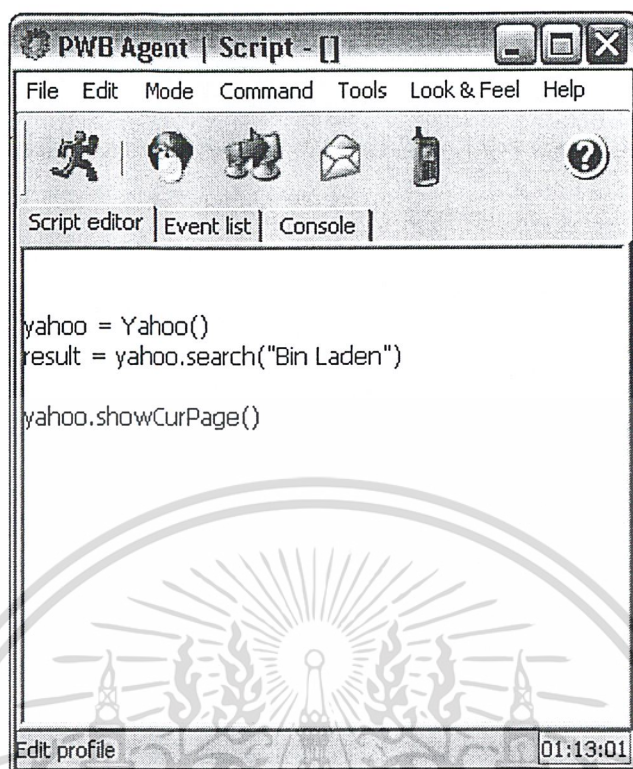
I. หน้าต่างแก้ไขโปรไฟล์ของผู้ใช้ (Profile dialog) โดยสามารถกำหนดสิ่งต่างๆได้ดังนี้คือ

- ❖ ข้อมูลของผู้ใช้
  - ข้อมูลรายการโปรดคยูอาร์แอล (Favorites) ที่ใช้ในเว็บเบราว์เซอร์
- ❖ ข้อมูลที่ใช้ในการทำงานของเอฟทีพี
  - ข้อมูลที่ใช้ในการจัดการเมล์
- ❖ ข้อมูลที่ใช้ในการทำงานของเอสเอ็มเอสโคลเอ็นด์
  - ข้อมูลการตั้งค่าเอเจนต์



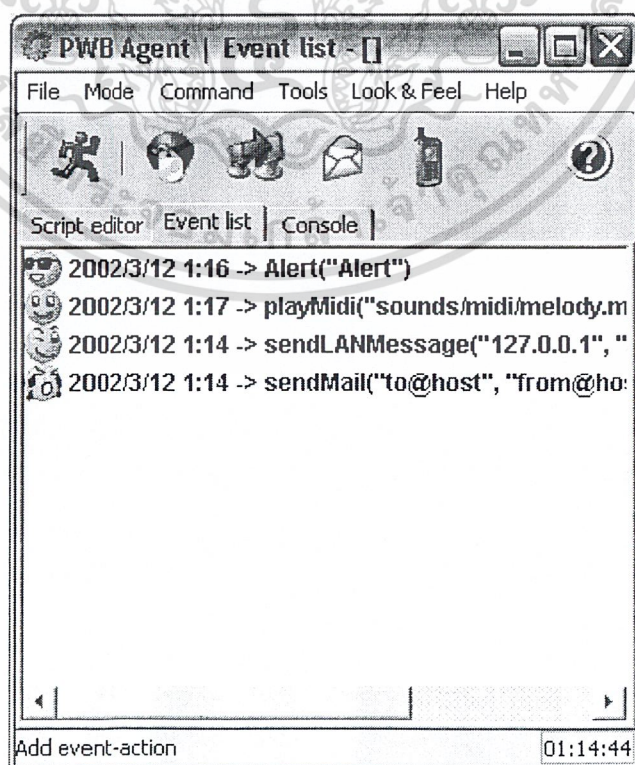
รูปที่ 5.2 รูปแสดงหน้าต่างแก้ไขโปรไฟล์ของผู้ใช้

II. หน้าต่างเขียนสคริปต์ (Script editor) — มีไว้สำหรับให้ผู้ใช้เขียน แก้ไข เปิด และเซฟสคริปต์ โดยการกดปุ่มรันจะเป็นการรันสคริปต์ในหน้าต่างนี้ และนำข้อมูลที่ผู้ใช้สร้างขึ้น เช่น ฟังก์ชัน คลาส และ เนมต่างๆมาเก็บไว้ในเนมสเปซของเอเจนต์ซึ่งเป็นการทำให้เอเจนต์รู้จักกับผู้ใช้



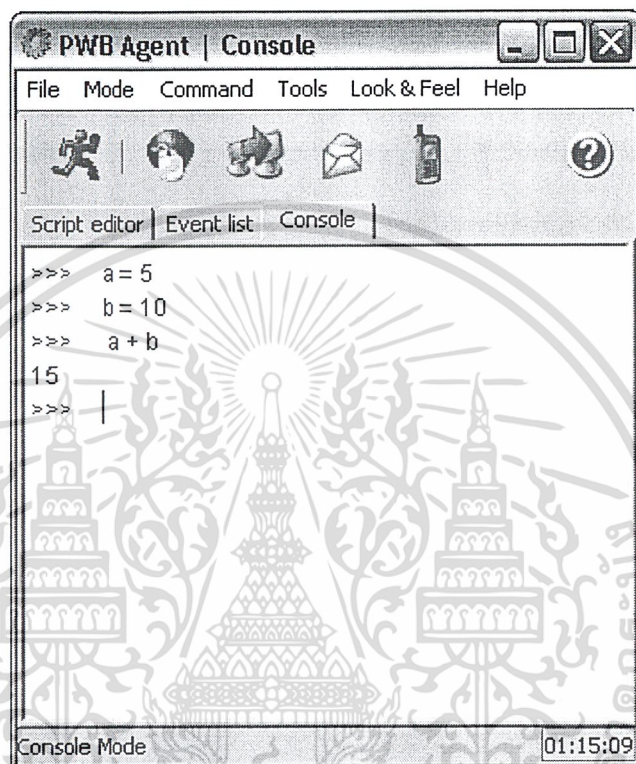
รูปที่ 5.3 รูปแสดงหน้าต่างเขียนสคริปต์

- III. หน้าต่างรายการอีเวนต์ (*Event – action list*) – เป็นหน้าต่างที่มีไว้สำหรับตรวจสอบรายการของอีเวนต์ – แอ็กชันต่างๆ ที่อยู่ในเอเจนต์ว่ามีสถานะเป็นอย่างไร และยังสามารถลบ แก้ไข และเพิ่มรายการได้ผ่านทางเมนูย่อยที่อยู่ในเมนูหลัก



รูปที่ 5.4 รูปแสดงหน้าต่างรายการอีเวนต์

- IV. หน้าต่างโต้ตอบและแสดงผล (Interactive and output window) – เป็นหน้าต่างที่มีไว้สำหรับใส่คำสั่งที่ละคำสั่งเพื่อให้เอเจนต์ประมวลผลโดยตรง และยังมีไว้สำหรับแสดงผล output ต่างๆ ของโปรแกรมเอเจนต์อีกด้วย



รูปที่ 5.5 รูปแสดงหน้าต่างโต้ตอบและแสดงผล

## 5.2 โพรไฟล์ (Profile)

คือส่วนที่ทำหน้าที่ในการเก็บข้อมูลลงในฐานข้อมูล คึงข้อมูลของผู้ใช้ขึ้นมาแสดง และนำข้อมูลต่างๆ ที่ผู้ใช้กำหนดมาสร้างเป็นเมนูภายในเมนูสเปคของผู้ใช้แต่ละคน รวมทั้งยังเก็บค่าการดีฟอลต์ของเอเจนต์ไว้อีกด้วย ส่วนโพรไฟล์ประกอบด้วย 2 ส่วนคือ

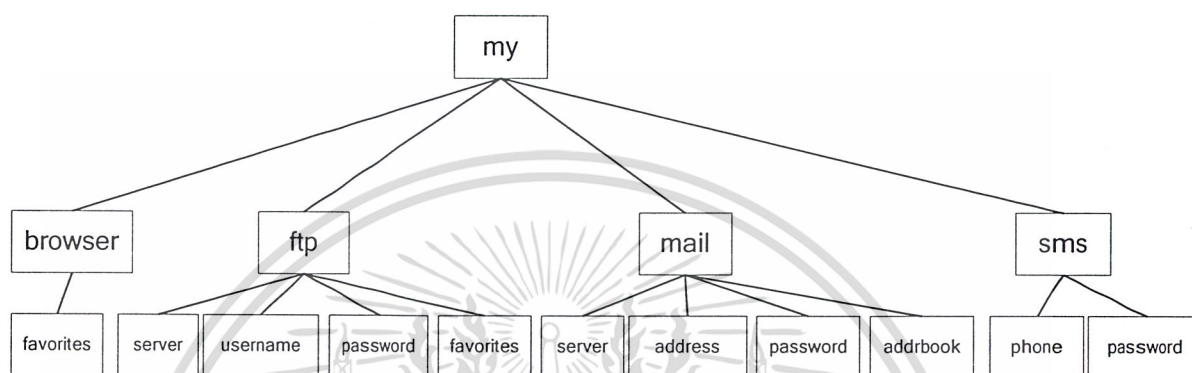
1. ฐานข้อมูลของผู้ใช้ (User profile database) – ใช้เก็บข้อมูลโพรไฟล์ต่างๆ ของผู้ใช้แต่ละคน เช่น ชื่อผู้ใช้ รหัสผ่าน อีเมลของผู้ใช้ และค่าเซตคั้งต่างๆ ผู้ใช้ทุกคนจะมีไคเรกทอรีที่เก็บโพรไฟล์ของตนอยู่ในไคเรกทอรี user-profiles ซึ่งไคเรกทอรีของแต่ละผู้จะมีข้อมูลโพรไฟล์คั้งนี้ คือ ftp.profile, mail.profile, sms.profile, user.profile และข้อมูลอื่นๆ อีกคั้งนี้ คือ cache, favourites, ftp\_fav, mailbook, phonebook, default.script, default.list และ settings

2. ส่วนจัดการโพรไฟล์ (Profile manager) - ทำหน้าที่ในการเก็บข้อมูลลงในฐานข้อมูล คึงข้อมูลของผู้ใช้ขึ้นมาแสดง และนำข้อมูลต่างๆ ที่ผู้ใช้กำหนดมาสร้างเป็นเมนูภายในเมนูสเปค

## การเก็บข้อมูลลงในฐานข้อมูล

โปรแกรมเอเจนต์ได้มีการนำเอาระบบยูสเซอร์โพรไฟล์โดยใช้ระบบไดเรกทอรีมาใช้ ซึ่งก่อนเข้าโปรแกรมต้องมีการระบุชื่อผู้ใช้งานและรหัสผ่าน เพื่อดึงเอาค่ารายละเอียดของผู้ใช้มา การมียูสเซอร์โพรไฟล์นี้มีประโยชน์เป็นอย่างมากต่อการทำงานของผู้ใช้ผ่าน Built-in object “my”

โครงสร้างของ my object มีดังรูปที่ 5.6



รูปที่ 5.6 รูปแสดงโครงสร้างของออบเจกต์ “my”

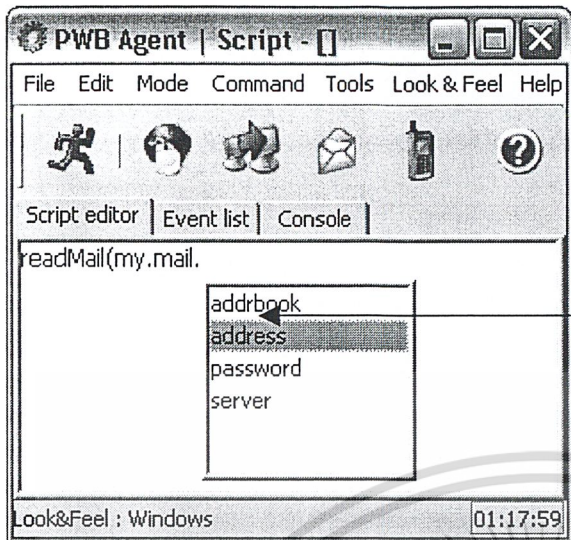
ตัวอย่างการใช้งานเป็นดังนี้

```

my.browser.favorite["KMITL"] # return "www.kmitl.ac.th"
my.ftp.server                 # return "diamond.ce.kmitl.ac.th"
my.ftp.username               # return "s1014234"
my.mail.server                 # return "chaokhun.kmitl.ac.th"
my.mail.addrbook              # return python-dictionary
my.mail.addrbook["M"]        # return "s1014172@cd.kmitl.ac.th"
  
```

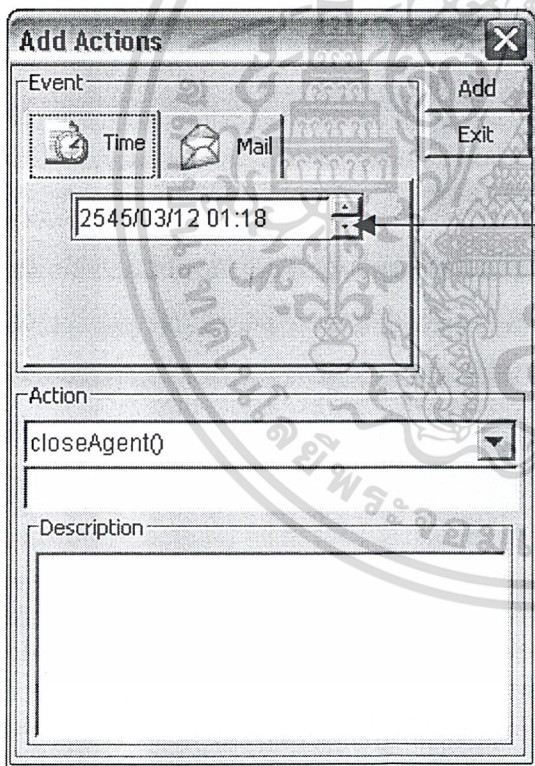
“my” object เมื่อร่วมกับระบบช่วยเหลือผู้ใช้ไม่เพียงแต่ช่วยผู้ใช้ในเรื่องการจดจำหรือปัญหาการเขียนชื่อยาวแล้ว มันยังช่วยในเรื่องการป้อนค่ารหัสผ่าน ทั้งบน dialog form และ ในคำสั่งสคริปต์ด้วย ดังรูปที่ 5.7 ประโยชน์อีกอย่างของการใช้ “my” object คือ เราสามารถเขียนสคริปต์ครั้งเดียวโดยสามารถใช้ได้กับทุกผู้ใช้งานโดยไม่ต้องเปลี่ยนแปลงโค้ดของสคริปต์

วิธีทำให้การกรอกรายละเอียดผู้ใช้เป็น “my” object ทำได้โดยใช้ความสามารถของภาษาไพธอนเรื่อง eval โดยตรวจสอบดูก่อนว่า สิ่งที่ใช้ป้อนเข้ามามีความหมายหรือเปล่า ถ้ามีก็ให้ใช้ค่าตามความหมายนั้น แต่ถ้าไม่มีก็ใช้ค่าเดิม ซึ่งจากรูปที่ 5.8 จะเห็นว่าเมื่อเรากรอก my.mail.server เข้าไป โปรแกรมจะมองเป็นเป็น ค่า chaokhun.kmitl.ac.th แทนที่จะเป็น my.mail.server



สคริปต์ที่เขียนขึ้นมา  
สามารถนำไปใช้ได้กับทุกผู้  
ใช้งาน โดยสามารถทำงาน  
ได้ถูกต้องโดยไม่ต้อง  
เปลี่ยนแปลงโค้ด อีกทั้งยัง  
ปลอดภัยเรื่องรหัสผ่าน

รูปที่ 5.7 รูปแสดงการใช้งาน “my” ออบเจกต์บนหน้าต่างเขียนสคริปต์



หากกรอกเป็น  
diamond.ce.knitl.ac.th โปรแกรม  
ก็จะใช้ค่านั้นในการทำงาน แต่หาก  
ป้อนเป็น my.mail.server  
โปรแกรมจะใช้ค่า  
chaokhun.kmitl.ac.th ในการ  
ทำงานแทน  
และเช่นเดียวกันกับ  
username และ password

รูปที่ 5.8 รูปแสดงการใช้งาน “my” ออบเจกต์บน input dialog อื่นๆ

5.3 บีลีฟเซต (Belief set)

คือ ส่วนข้อมูล หรือฐานความรู้ของเอเจนต์ อันได้แก่ เนม และสคริปต์ซึ่งจะมีชื่ออยู่ในเนม สเปส ของเอเจนต์ และอีเวนต์ – แอ็กชันที่เมื่อ ไรก็ตามที่อีเวนต์เกิดขึ้นแล้วเอเจนต์จะทำงานตามแอ็กชันนั้น บีลีฟเซตประกอบด้วย 3 ส่วน คือ เนม, สคริปต์, และอีเวนต์-แอ็กชัน

### 5.3.1 เนม (Name)

คือชื่อที่ผู้ใช้อ้างถึงข้อมูลต่างๆภายในเอเจนต์ซึ่งเนมในมุมมองของผู้ใช้ก็คือตัวแปรที่ใช้อ้างถึงข้อมูลต่าง ๆ นั้นเอง โดยที่ผู้ใช้สามารถเพิ่มเติมเนมเข้าไปเป็นฐานความรู้ให้แก่เอเจนต์ได้ บีลีเฟดในส่วนี้ถูกเก็บอยู่ใน เนมสเปส (Name space)

โปรแกรมมีเนมที่สร้างขึ้นไว้ให้แล้ว (PWB Built-in Name and Object) ดูจากตารางที่ 5.1

Built-in object	Description
AGENT	object that refer to current agent object
my.browser.favorites	dictionary of favorite websites
my.ftp.server	default FTP server name
my.ftp.username	default FTP username
my.ftp.password	default FTP password
my.ftp.favorites	dictionary of favorite FTP servers
my.mail.server	default Mail server
my.mail.address	default Mail address
my.mail.password	default Mail password
my.mail.addrbook	dictionary of e-mail addresses
my.sms.username	default SMS username
my.sms.password	default SMS password

ตารางที่ 5.1 ตารางแสดง PWB Built in Object

### 5.3.2 สคริปต์ (Script)

คือส่วนของโปรแกรมที่เอเจนต์มีอยู่ทั้งในรูปแบบของฟังก์ชัน และคลาส ผู้ใช้สามารถเพิ่มเติมสคริปต์เข้าไปเป็นฐานความรู้ให้แก่เอเจนต์ได้ โดยการประกาศฟังก์ชันหรือคลาส ตามหลักไวยากรณ์ภาษาไพธอน ส่วนของสคริปต์นี้จะเก็บอยู่ในที่เดียวกันกับเนม คือ เนมสเปส

โปรแกรมมีฟังก์ชันและคลาสที่สร้างขึ้นไว้ให้แล้ว (PWB Built-in Function and Class) ดูจากตารางที่ 5.2 และ 5.3 ตามลำดับ

Function name	Parameters	Description
openWebPage	url = ที่อยู่อินเตอร์เน็ต	เปิดหน้าต่าง PWB – Browser พร้อมกับดึงข้อเว็บเพจ ที่ระบุที่อยู่โดย url มาแสดงผล
uploadFile	server = ที่อยู่ของไฟล์ทรานสเฟอร์เซิร์ฟเวอร์ userName = ชื่อผู้ใช้งาน passwd = รหัสผ่าน localFile = ไฟล์ต้นทางที่เครื่องผู้ใช้ remotePath = ที่อยู่ปลายทางที่เครื่องเซิร์ฟเวอร์	เชื่อมต่อเข้าสู่ FTP server และถ่ายโอนข้อมูลจากเครื่องผู้ใช้งานไปยัง server
downloadFile	server = ที่อยู่ของไฟล์ทรานสเฟอร์เซิร์ฟเวอร์ userName = ชื่อผู้ใช้งาน passwd = รหัสผ่าน remoteFile = ไฟล์ต้นทางที่เครื่องเซิร์ฟเวอร์ localPath = ที่อยู่ปลายทางที่เครื่องผู้ใช้	เชื่อมต่อเข้าสู่ FTP server และถ่ายโอนข้อมูลจาก server มายังเครื่องผู้ใช้งาน

sendMail	toAddress = ที่อยู่เมลผู้รับ fromAddr = ที่อยู่เมลผู้ส่ง subject = เรื่อง body = เนื้อข้อความ	ส่ง e-mail ไปยังผู้รับปลายทาง
checkMail	server = เมลเซิร์ฟเวอร์ username = ชื่อผู้ใช้ password = รหัสผ่านผู้ใช้	ตรวจสอบดูที่ mail server ว่ามี e-mail อยู่ใน mailbox เป็นจำนวนกี่ฉบับ
readMail	server = เมลเซิร์ฟเวอร์ username = ชื่อผู้ใช้ password = รหัสผ่านผู้ใช้	เปิดหน้าต่าง PWB – Mail Client พร้อมกับดึง e-mail มาแสดงผล
sendSMS	telNo = หมายเลขโทรศัพท์ password = รหัสผ่าน message = ข้อความ	ส่ง short message ไปยัง cellular phone
playWav	pathfile = ที่อยู่ไฟล์	เล่นไฟล์เพลง นามสกุล *.wav
playMidi	pathfile = ที่อยู่ไฟล์	เล่นไฟล์เพลง นามสกุล *.midi
showMessage	message = ข้อความที่แสดง	แสดงข้อความโดย showMessageDialog
sendLANMessage	destIP = ชื่อเครื่องหรือ IP เครื่องปลายทาง message = ข้อความ	ส่งข้อความไปยังเครื่องคอมพิวเตอร์ปลายทาง ภายในเครือข่าย LAN เดียวกัน
startFile	pathfile = ที่อยู่ไฟล์	
shutdownComputer	[mode] = shutdown or restart	ปิดระบบปฏิบัติการ เมื่อ mode = 0 restart เมื่อ mode = 1
runCMD	command	รันคำสั่งระบบ
Alert	msg = ข้อความที่ใช้เตือน [func] = ฟังก์ชันที่จะเรียกเมื่อกดปุ่มเตือน [delay] = ระยะเวลาการเตือน	popup ข้อความเตือนผู้ใช้งาน และสามารถใส่ function เข้าไปได้โดยจะถูกเรียกเมื่อ กดปุ่มบน Alert popup

ตารางที่ 5.2 ตารางแสดง PWB Built in Function

Class name	Description
WebSite	Abstract class for any website
KMITL	WebSite-inherited class for www.kmitl.ac.th
Yahoo	WebSite-inherited class for www.yahoo.com
Weather	WebSite-inherited class for www.thaimet.tmd.go.th

ตารางที่ 5.3 ตารางแสดง PWB Built in Class

### 5.3.3 อีเวนต์ – แอ็กชัน (Event - action)

คือฐานความรู้ในอีกรูปแบบหนึ่งที่มีความหมายว่าเมื่ออีเวนต์เกิดขึ้นแล้วจะให้ทำแอ็กชันอะไร โดยมันจะถูกเก็บอยู่ในฐานความรู้ของเอเจนต์และถูกตรวจสอบโดยส่วนจัดการอีเวนต์อยู่เสมอจนเมื่ออีเวนต์เป็นจริงแล้วก็จะทำแอ็กชัน และรายงานผลของแอ็กชันว่าเป็นเช่นไร โครงสร้างของอีเวนต์-แอ็กชันถูกเก็บไว้เป็นลิสต์มีโครงสร้าง ดังนี้

[Event type, Flag, Event, Action]

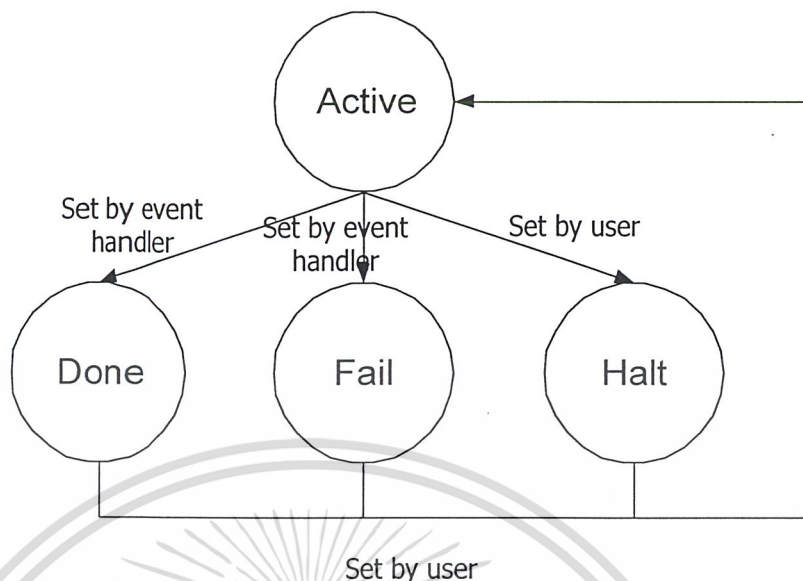
**Event type** คือ ชนิดของเหตุการณ์ซึ่งส่วนจัดการอีเวนต์จะใช้ข้อมูลนี้เป็นตัวแยกจัดการอีเวนต์  
Event type มีอยู่ 3 ชนิด คือ

- time - ใช้ระบุว่ารายการนี้เป็นอีเวนต์ชนิด time ซึ่งเมื่อเวลาที่กำหนดไว้ในรายการมีค่าเท่ากับเวลาปัจจุบัน(สามารถตั้งได้มากที่สุดถึงหน่วยนาทีก็นจะทำแอ็กชัน
- mail - ใช้ระบุว่ารายการนี้เป็นอีเวนต์ชนิด mail ซึ่งเมื่อมีเมลล์ฉบับใหม่เข้ามาในเมลล์บ็อกซ์ของเราที่เมลล์เซอร์ฟเวอร์ที่ระบุไว้ก็จะทำแอ็กชัน
- others - ใช้ระบุว่ารายการนี้เป็นอีเวนต์ชนิด others ซึ่งเมื่อทำการประมวลผลอีเวนต์(ในรูปแบบเอ็กเพรสชัน) แล้วได้ค่าเป็นจริงก็จะทำแอ็กชัน แต่ถ้าได้ค่าเป็นเท็จก็จะคอยตรวจสอบต่อไป

**Flag** คือ ค่าที่ใช้ระบุสถานะของรายการนั้นๆว่าปัจจุบันมีสถานะอย่างไร ซึ่งค่านี้จะถูกใช้โดยส่วนจัดการอีเวนต์ และถูกนำมาแสดงผลให้ผู้ใช้ดู และสามารถแก้ไขได้ Flag มีอยู่ 4 ชนิดคือ

- active - ใช้ระบุว่ารายการนี้ยังทำงานอยู่ และได้รับการตรวจสอบจากส่วนจัดการอีเวนต์อยู่อย่างสม่ำเสมอ
- done - ใช้ระบุว่ารายการนี้ได้กระทำแอ็กชันเสร็จเรียบร้อยแล้ว และไม่พบข้อผิดพลาดในการทำงานแต่อย่างใดและ ส่วนจัดการอีเวนต์จะไม่ตรวจสอบรายการนี้อีกต่อไปจนกว่าผู้ใช้จะทำการเปลี่ยนสถานะให้เป็น active
- fail - ใช้ระบุว่ารายการนี้ได้กระทำแอ็กชันแล้วแต่ว่า พบข้อผิดพลาดในการทำงานและส่วนจัดการอีเวนต์จะไม่ตรวจสอบรายการนี้อีกต่อไปจนกว่าผู้ใช้จะทำการเปลี่ยนสถานะให้เป็น active
- halt - ใช้ระบุว่ารายการนี้จะไม่ถูกตรวจสอบโดย ส่วนจัดการอีเวนต์ จนกว่าผู้ใช้จะทำการเปลี่ยนสถานะให้เป็นให้เป็น active สถานะนี้จะถูกตั้งให้กับทุกรายการเมื่อทำการเซฟลิสต์ของรายการลงในไฟล์

โดยสรุป Flag มีไว้สำหรับบอกสถานะของแต่ละรายการ โดย active หมายถึงรายการถูกตรวจสอบโดยส่วนจัดการอีเวนต์ ส่วน done, fail และ halt จะไม่ถูกตรวจสอบโดยส่วนจัดการอีเวนต์ แต่ที่ต้อต้องมีก็เพื่อไว้รายงานสถานะการทำงานให้ผู้ใช้ทราบ สถานะของรายการแสดงได้ดังรูปที่ 5.9



รูปที่ 5.9 รูปแสดงการเปลี่ยนสถานะของรายการอีเวนต์-แอ็กชัน

**Event** คือ ค่าที่ใช้เก็บอีเวนต์เพื่อให้ส่วนจัดการอีเวนต์ทำหน้าที่ตรวจสอบโดยค่าของอีเวนต์มี 3 รูปแบบตามจำนวนชนิดของอีเวนต์คือ

- Time – รูปแบบของ time event คือ ปี.ศ./เดือน/วันที่ ชั่วโมง:นาที เช่น 2002/3/8 23:22
- Mail – รูปแบบของ mail event คือ user@server:password เช่น s1014172@ce.kmitl.ac.th :Abc123
- Others – รูปแบบของ others คือ เอ็กเพรสชันใดๆ ที่สามารถส่งกลับค่ามาเป็นถูกหรือผิด เช่น a and b, x, sendMail() and upload()

**Action** คือ ค่าที่ใช้เก็บชื่อฟังก์ชันที่จะให้ประมวลผลเมื่ออีเวนต์เกิดขึ้นซึ่งโปรแกรมเอเจนต์มียูทิลิตี้ฟังก์ชันที่สำคัญมาให้แล้ว และผู้ใช้ก็สามารถสร้างฟังก์ชันของตัวเองขึ้นมาใช้งานได้ รูปแบบของแอ็กชันก็คือรูปแบบของการเรียกฟังก์ชันทั่วไป เช่น sendMail(), Alert('hello')

#### 5.4 ส่วนจัดการอีเวนต์ (Event handler)

คือ ส่วนที่ใช้ในการตรวจสอบอีเวนต์ในบิลีฟของเอเจนต์โดยหลักการคือจะต้องให้ส่วนจัดการอีเวนต์รันเป็นเธรดอยู่ตลอดเวลา และคอยตรวจสอบอีเวนต์ที่เป็นหน้าที่ของตนว่ามีอีเวนต์ไหนเกิดขึ้นแล้วบ้างซึ่งเมื่อพบก็จะส่งแอ็กชันไปประมวลผลที่อินเทอร์พรีเตอร์แบบมัลติเธรด (Multithreading interpreter) ส่วนจัดการอีเวนต์ประกอบด้วย 3 ส่วนดังนี้

##### 5.4.1 ตัวจัดการอีเวนต์เวลา (Time-Event Handler)

คือส่วนที่ทำหน้าที่ตรวจสอบอีเวนต์เวลาทั้งหมดในฐานความรู้โดยจะเปรียบเทียบกับเวลาของเอเจนต์ โดยเมื่อถึงเวลาที่กำหนด(กำหนดได้ละเอียดสุดถึงหน่วยนาฬิกา) ก็จะทำแอ็กชันที่กำหนด

หลักการทำงาน คือ จะตรวจสอบรายการอีเวนต์-แอ็กชันทุกรายการ โดยเมื่อพบรายการที่มีอีเวนต์เป็นชนิด time และยังมีสถานะ active อยู่ ก็จะทำการเปรียบเทียบเวลาในรายการนั้นว่าเท่ากับเวลาปัจจุบันของเอเจนต์หรือไม่โดยถ้าเวลาเท่าก็จะส่งแอ็กชันไปประมวลผลที่อินเตอร์พรีดเตอร์แบบมัลติเธรด (Multithreading interpreter) ซึ่งถ้าทำได้สำเร็จรายการนั้นก็จะมีสถานะเป็น done แต่ถ้ามีข้อผิดพลาดเกิดขึ้นก็จะเปลี่ยนสถานะของรายการนั้นให้เป็น fail

#### 5.4.2 ตัวจัดการอีเวนต์เวลา (Mail-Event Handler)

คือส่วนที่ทำหน้าที่ตรวจสอบอีเวนต์เมลล์ทั้งหมดในฐานความรู้โดยเมื่อมีอีเมลล์ฉบับใหม่เข้ามาในอีเมลล์แอ็กเคานท์ก็จะทำแอ็กชันที่กำหนด

หลักการทำงาน คือ จะตรวจสอบรายการอีเวนต์-แอ็กชันทุกรายการ โดยเมื่อพบรายการที่มีอีเวนต์เป็นชนิด mail และยังมีสถานะ active อยู่ ก็จะดูว่ารายการนี้มีค่าจำนวนเมลล์ปัจจุบันแล้วหรือยังซึ่งในครั้งแรกจะยังไม่มีการติดต่อกับเมลล์เซิร์ฟเวอร์เพื่อขอข้อมูลจำนวนเมลล์มาเก็บไว้หลังจากนั้นในทุกๆช่วงเวลาที่กำหนดก็จะไปร้องขอจำนวนเมลล์จากเซิร์ฟเวอร์มาใหม่ซึ่งถ้าจำนวนเมลล์มากขึ้นก็จะส่งแอ็กชันไปประมวลผลที่อินเตอร์พรีดเตอร์แบบมัลติเธรด (Multithreading interpreter) ซึ่งถ้าทำได้สำเร็จรายการนั้นก็จะมีสถานะเป็น done แต่ถ้ามีข้อผิดพลาดเกิดขึ้นก็จะเปลี่ยนสถานะของรายการนั้นให้เป็น fail

#### 5.4.3 ตัวจัดการอีเวนต์อื่นๆ (Other-Events Handler)

คือส่วนที่ทำหน้าที่ตรวจสอบอีเวนต์ใดๆ ที่ผู้ใช้กำหนดขึ้นเอง ในรูปแบบของเอ็กเพรสชัน (Expression) ที่สามารถส่งค่ากลับมาในรูปแบบของถูก หรือผิด (true or false) ได้โดยเมื่อเอ็กเพรสชันส่งค่ากลับมาเป็นจริงก็จะทำแอ็กชันที่กำหนด

หลักการทำงาน คือ จะตรวจสอบรายการอีเวนต์-แอ็กชันทุกรายการ โดยเมื่อพบรายการที่มีอีเวนต์เป็นชนิด others และยังมีสถานะ active อยู่ ก็จะทำการประมวลผลเอ็กเพรสชันในรายการนั้นดูว่าค่าที่ส่งกลับมามีค่าเป็นจริง หรือเปล่า ถ้ามีค่าเป็นจริง ก็จะส่งแอ็กชันไปประมวลผลที่อินเตอร์พรีดเตอร์แบบมัลติเธรด (Multithreading interpreter) ซึ่งถ้าทำได้สำเร็จรายการนั้นก็จะมีสถานะเป็น done แต่ถ้ามีข้อผิดพลาดเกิดขึ้นก็จะเปลี่ยนสถานะของรายการนั้นให้เป็น fail แต่ถ้ามีค่าเป็นเท็จก็จะผ่านไปโดยไม่ทำแอ็กชัน และจะคอยตรวจสอบต่อไปเรื่อยๆ จนกว่าอีเวนต์จะเป็นจริงแล้วจึงทำแอ็กชัน

### 5.5 อินเตอร์พรีดเตอร์แบบมัลติเธรด (Multithreading interpreter)

ทำหน้าที่ในการประมวลผลโปรแกรมแบบขนาน โดยไม่ต้องเสียเวลารอให้โปรแกรมปัจจุบันประมวลผลเสร็จก็สามารถประมวลผลโปรแกรมต่อไปได้ซึ่งการทำการประมวลผลเช่นนี้จะมีประโยชน์อย่างมากเนื่องจากโมดูลที่ใช้ติดต่อกับอินเตอร์เน็ตจำเป็นต้องเสียเวลาในการรอรับข้อมูลตอบกลับจากเซิร์ฟเวอร์ดังนั้นการทำงานแบบมัลติเธรดจะสามารถทำให้เอเจนต์สามารถทำงานอย่างอื่นไปได้โดยไม่ต้องเสียเวลาไปโดยเปล่าประโยชน์

โดยโครงสร้างแล้วหน่วยประมวลผลจะแบ่ง สคริปต์ออกเป็น 3 ส่วนคือ

- Sequential script เป็นสคริปต์ที่ทำงานอย่างต่อเนื่องคำสั่งต่อคำสั่ง เหมาะสมกับงานที่ต้องอาศัยความขึ้นต่อกันของคำสั่งก่อนหน้า
- Parallel script เป็นสคริปต์ที่แต่ละคำสั่งจะทำงานไปพร้อมๆกัน (Concurrency) อย่างอิสระ โดยหน่วยประมวลผลจะต้องแตกเรดออกมาเพื่อรองรับการทำงานของแต่ละคำสั่ง
- Event-action script เป็นสคริปต์ในส่วนใหญ่ที่ใช้กำหนด อีเวนต์-แอ็กชัน โดยจะถูกเพิ่มเข้าไปในรายการอีเวนต์ทันที

การแบ่งแยกสคริปต์ทั้ง สามส่วน สามารถทำได้โดยใช้ XML tag ดังนี้

```

<sequential>
### sequential part ###
</sequential>

<parallel>
### parallel part ####
</parallel>

<event>
### event-action part ###
</event>

```

## 5.6 โมดูลที่ใช้ติดต่อผ่านอินเทอร์เน็ต (Communication module)

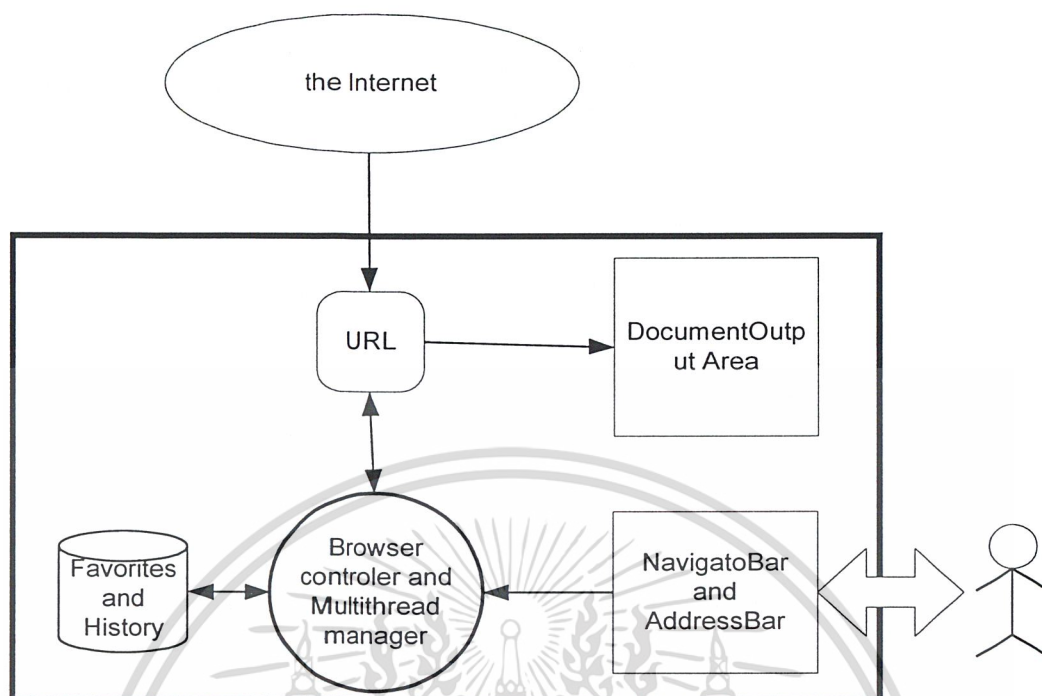
คือโมดูลที่ทำหน้าที่ติดต่อขอใช้บริการที่สำคัญๆบนอินเทอร์เน็ต ประกอบด้วยโมดูลต่างๆ ดังนี้

### 5.6.1 เว็บเบราว์เซอร์(Web browser)

เว็บเบราว์เซอร์ (PWB – Browser) เป็นตัวแสดงเอกสารเอชทีเอ็มแอลซึ่งต้องอาศัยหลักการพาร์สเซอร์ตัวภาษาแล้วจึงแสดงผลข้อมูลตามรูปแบบที่ได้ ซึ่งในส่วนใหญ่สามารถจัดการได้โดยไลบรารีของจาวา สิ่งที่เราต้องจัดการเพิ่มเติมมีดังนี้

- การทำงานแบบ มัลติเธรด เพื่อให้ขณะที่โหลดข้อมูลผ่านเอชทีทีพีโปรโตคอล ตัวโปรแกรมยังสามารถตอบสนองต่อผู้ใช้งาน
- การจัดการเรื่องรายการยูอาร์แอลเก่า (History) เพื่อให้ผู้ใช้สามารถท่องเว็บไปมาได้สะดวก และเรื่องของรายการโปรด (Favorites) ของผู้ใช้แต่ละคน โดยทั้งสองสามารถทำได้โดยใช้ Python Dictionary

รูปที่ 5.10 แสดงรูปของโครงสร้างของโปรแกรม และ รูปที่ 5.11 แสดงรูปของโปรแกรมเว็บเบราว์เซอร์



รูปที่ 5.10 รูปแสดงโครงสร้างของเว็บเบราว์เซอร์



รูปที่ 5.11 รูปแสดงหน้าต่างของเว็บเบราว์เซอร์

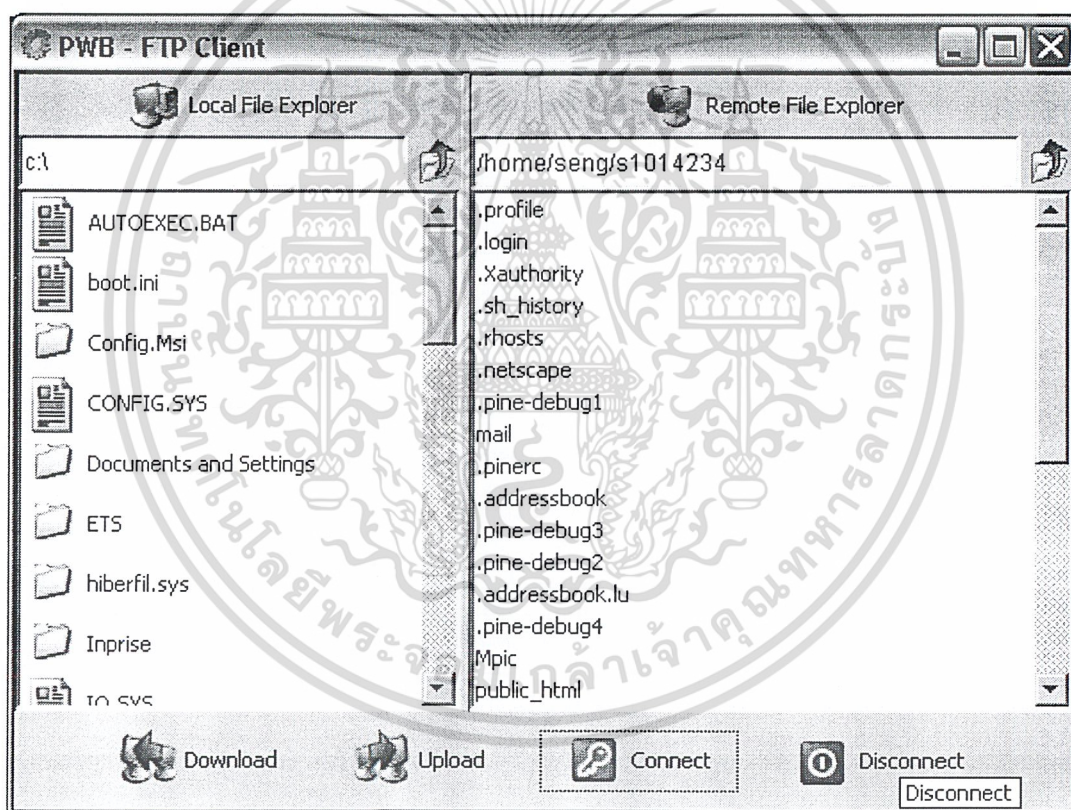
### 5.6.2 เอฟทีพีไคลเอนต์ (FTP Client)

ใช้ในการอัปโหลด (Upload) และดาวน์โหลด (Download) ไฟล์ระหว่างไคลเอนต์กับเอฟทีพีเซิร์ฟเวอร์

โมดูลที่ใช้ในการติดต่อและส่งไฟล์ของไพธอนคือ `ftplib` โดยสามารถใช้ได้ดังนี้  
`ftplib.FTP(host, user, passwd)` จะได้ออบเจกต์ที่สามารถใช้ติดต่อกับเซิร์ฟเวอร์ได้  
`retrbinary(command, method, size)` ใช้สำหรับส่งคำสั่งไป โดย `command` จะเป็นไปได้ดังนี้

- RETR remoteFile เพื่อ get ไฟล์
- STOR localFile เพื่อ put ไฟล์

รูปที่ 5.12 แสดงรูปของโปรแกรมเอฟทีพีไคลเอนต์



รูปที่ 5.12 รูปแสดงหน้าต่างของโปรแกรมเอฟทีพี

### 5.6.3 เมล์ไคลเอนต์ (Mail Client)

ใช้ในการดาวน์โหลดจดหมายจากเมลบ็อกซ์บนเซิร์ฟเวอร์มาไว้ที่ไคลเอนต์ผ่านโปรโตคอลป๊อป 3 และใช้ในการส่งจดหมายผ่านโปรโตคอลเอสเอ็มทีพี

## ดาวน์โหลดเมลล์

โมดูลที่สำคัญที่ใช้ในการดาวน์โหลดจดหมายจากเมลล์บ็อกซ์ได้แก่ poplib ซึ่ง โมดูลนี้มีคลาส POP3 อยู่ ซึ่งคลาสนี้สามารถติดต่อไปยังป๊อป 3 เซิร์ฟเวอร์ได้ และเมื่อได้ ออบเจกต์ของคลาสนี้มาแล้วก็จะสามารถที่จะติดต่อไปยังป๊อป 3 เซิร์ฟเวอร์ได้โดยตรง ซึ่งป๊อปออบเจกต์มีเมธอดที่สำคัญดังนี้

**getwelcome()** ส่งค่ากลับมาเป็นสตริงข้อความต้อนรับจาก ป๊อป 3 เซิร์ฟเวอร์

**user(username)** ส่งชื่อผู้ใช้ไปยังป๊อป 3 เซิร์ฟเวอร์ และค่าที่ตอบกลับมาคือสตริงรหัสผ่านจากผู้ใช้

**pass(password)** ส่งรหัสผ่านไปยังป๊อป 3 เซิร์ฟเวอร์ สิ่งที่ตอบกลับมา คือจำนวนเมลล์ และขนาดของเมลล์บ็อกซ์(เมลล์บ็อกซ์ที่เซิร์ฟเวอร์จะถูกถือจนกว่าจะใช้เมธอด quit() )

**apop(user, secret)** ใช้ APOP ซึ่งมีความปลอดภัยสูงกว่าในการพิสูจน์ตนเพื่อล็อกอินเข้าป๊อป 3 เซิร์ฟเวอร์

**rpop(user)** ใช้ RPOP ในการพิสูจน์ตน (เหมือนกับ UNIX r-command)เพื่อล็อกอินเข้าป๊อป 3 เซิร์ฟเวอร์

**stat()** ส่งค่ากลับเป็นทUPLE ของอินทิเจอร์ 2 ตัว คือ จำนวนเมลล์ และขนาดของเมลล์บ็อกซ์

**list([which])** ใช้สำหรับร้องขอรายการของเมลล์ที่มีอยู่ทั้งหมดในรูปแบบ (response, ['msg\_num octets', ...])

**retr(which)** ดึงข้อมูลเมลล์ ฉบับที่ which ผลลัพธ์อยู่ในรูปแบบ (response, ['line', ...], octets)

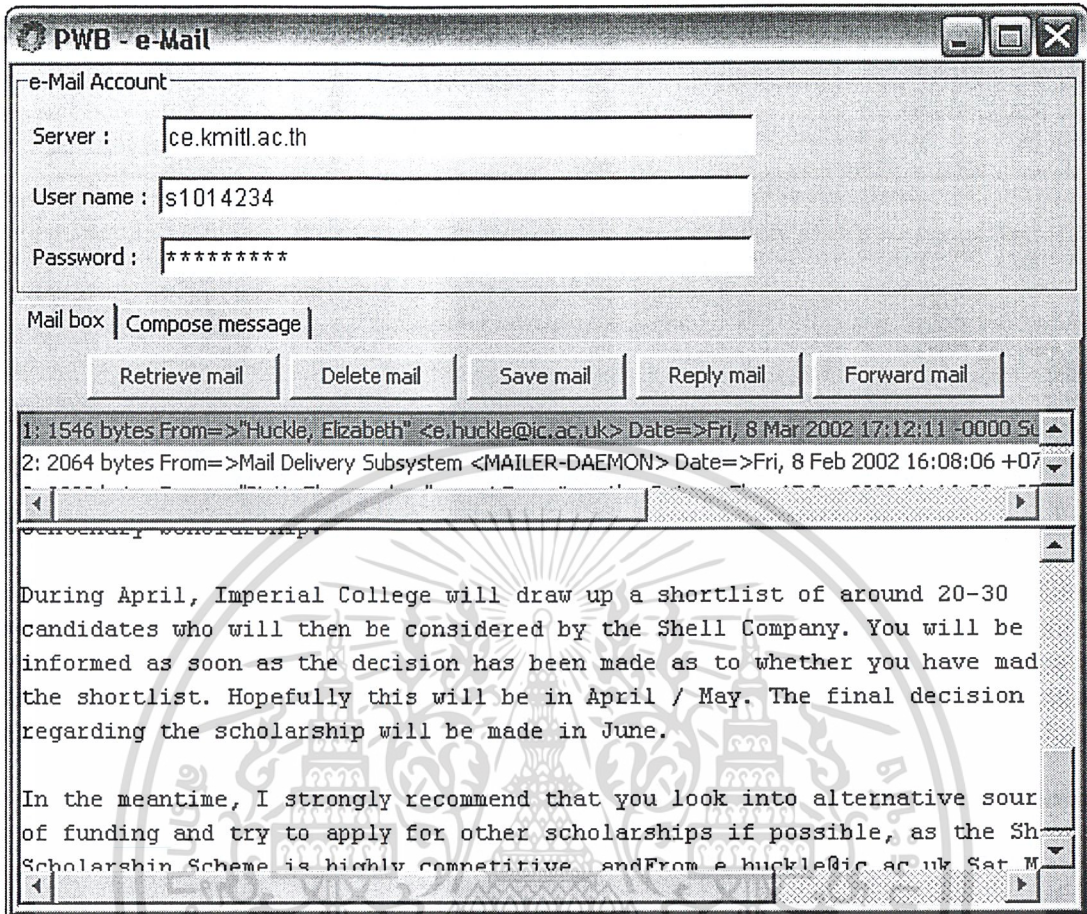
**delete(which)** ลบเมลล์ฉบับที่ which

**noop()** ใช้สำหรับการคงสถานะการเชื่อมต่อเอาไว้

**quit()** ล็อกออฟ และจะอันล็อกเมลล์บ็อกซ์ พร้อมทั้งยุติการเชื่อมต่อ

นอกจากนี้ก็ยังมียังมีโมดูล rfc822 ซึ่งในโมดูลนี้มีคลาส Message ซึ่งมีประโยชน์ในการดึงข้อมูลต่างๆ จากส่วนหัวของเมลล์

รูปที่ 5.13 แสดงรูปของโปรแกรมเมลล์ไคลเอ็นต์ โดยที่ผู้ใช้จะต้องกรอกข้อมูลที่จะใช้ในการดาวน์โหลดเมลล์คือ ชื่อเครื่องเซิร์ฟเวอร์ที่เมลล์มีเมลล์บ็อกซ์ของเรา ชื่อผู้ใช้ และรหัสผ่านแล้วจึงกดปุ่ม retrieve mail ซึ่งจะทำการเชื่อมต่อไปยังเมลล์เซิร์ฟเวอร์ และดึงข้อมูลเมลล์ทั้งหมดมาแสดงผลที่โปรแกรมเมลล์ไคลเอ็นต์ และยุติการเชื่อมต่อกับเมลล์เซิร์ฟเวอร์ และหลังจากที่ดาวน์โหลดเมลล์มาที่ไคลเอ็นต์แล้วก็จะสามารถ Delete mail, Save mail, Reply mail และ Forward mail ได้



รูปที่ 5.13 รูปแสดงหน้าต่างโปรแกรมเมล (รับเมล)

## ส่งเมล

โมดูลที่สำคัญที่ใช้ในการส่งเมล คือ `smtplib` ซึ่งจะมีคลาส `SMTP` ซึ่งสามารถใช้ติดต่อกับเซิร์ฟเวอร์ใดๆที่ให้บริการ SMTP ได้ เมธอดที่สำคัญของ SMTP ออบเจ็กต์คือ

`sendmail (from_addr, to_addrs, msg)` ใช้ในการส่งอีเมล โดย `from_addr` หมายถึง email address ของผู้ส่ง `to_addrs` คือ list ของ email address ของผู้รับซึ่งมีได้มากกว่า 1 ชื่อ `msg` คือข้อความที่ส่ง

`quit()` ใช้ยุติการเชื่อมต่อกับเอสเอ็มทีพีเซิร์ฟเวอร์

รูปที่ 5.14 แสดงรูปโปรแกรมเมลโคลเอ็นด์ในส่วนของกรการส่งเมลซึ่งข้อมูลที่จำเป็นคือ ชื่อเซิร์ฟเวอร์, email address ของผู้รับซึ่งถ้าต้องการส่งมากกว่า 1 คนให้ขึ้นด้วยเครื่องหมายเซมิโคลอน(;), email address ของผู้ส่ง, หัวข้อเมล และข้อความจดหมาย

เมื่อกดปุ่ม `send` แล้วโปรแกรมก็จะติดต่อไปยังเอสเอ็มทีพีเซิร์ฟเวอร์ และทำการส่งเมลไปที่เอสเอ็มทีพีเซิร์ฟเวอร์ซึ่งเอสเอ็มทีพีเซิร์ฟเวอร์ก็จะใช้ MTA (Mail Transfer Agent) ในการนำส่งเมลต่อไปจนถึงปลายทาง

**PWB - e-Mail**

e-Mail Account

Server : ce.kmitl.ac.th

User name : s1014234

Password : \*\*\*\*\*

Mail box **Compose message**

To : emme#i-kool.com; visit@ce.kmitl.ac.th

From : meng234@hotmail.com

Subject : Test Subject

This is test .....

Thank you for using PWB Mail

Send Clear

รูปที่ 5.14 รูปแสดงหน้าต่างโปรแกรมเมล (ส่งเมล)

#### 5.6.4 เอสเอ็มเอสไคลเอนต์ (SMS Client)

ใช้ในการส่งข้อความสั้นไปยังโทรศัพท์มือถือโดยผ่านเครือข่ายของเอไอเอส โดยหลักการทำคือ จะทำการร้องขอซิงโครไนซ์ที่ส่งข้อความสั้นที่เว็บไซต์ <http://www.gsm.ais900.com/gsmweb/> โดยจะต้องสมัครสมาชิกก่อนจึงจะสามารถส่งข้อความสั้นได้

Welcome to GSM Web Messaging 2K Edition - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail Stop

Address <http://www.gsm.ais900.com/gsmweb/> Go Norton AntiVirus

## GSM Web Messaging

communication connection

GSM Mobile No. :+66

GSM Password:

Type your message here or select from Poem or Cool message

Poem or Cool message

[ up to 140 characters in English/ 70 characters in Thai ]

**GSM** advance Welcome to GSM Web Messaging 2K Edition

Please fill in the GSM number you wish to contact. Internet

รูปที่ 5.15 รูปแสดงเว็บเพจที่ใช้ส่งข้อความสั้น

ซึ่งในการร้องขอซึ่จึ่ไอน้ันจะต้งหาชื่อของโปรแกรมซึ่จึ่ไอ่ที่ใช้ในการส่งเอสเอ็มเอส และชื่อตัวแปรต่างๆที่อยู่ในฟอร์มจากซอร์สโค้ดเอชทีเอ็มแอลดังนี้

```
<FORM name=fmGsm action=./web2sms.php method=post>
```

```
<input onBlur=ProcNumberValid();
```

```
onFocus="ProcHelp('Please fill in the GSM number you wish to contact.');"

```

```
maxlength=8 size=15 name=szGsmNo minlength="7">
```

```
<input +
```

```
onFocus="ProcHelp('Some GSM subscriber may require you to know a password to GSM
mobile them.');"

```

```
type=password maxlength=8 size=21 name=szGsmPwd>
```

```
<textarea onKeyPress="RefreshLength();" onFocus="ProcHelp('Please type your message,this message
send to GSM mobile.');" name=szMesg rows=4 cols=40></textarea>
```

จากซอร์สโค้ดของเอ็มแอลซีจีไอที่จะเรียกชื่อ ./web2sms.php ซึ่งชื่อเต็มคือ  
<http://www.gsm.ais900.com/gsmweb/web2sms.php> และวิธีที่ใช้ส่งข้อมูลคือแบบ post และชื่อตัวแปร 3  
 ตัวคือ

szGsmNo เป็นข้อมูลหมายเลขโทรศัพท์

szGsmPwd เป็นข้อมูลรหัสผ่านของการส่งข้อความสั้น

szMesg เป็นข้อมูลข้อความที่ต้องการส่ง

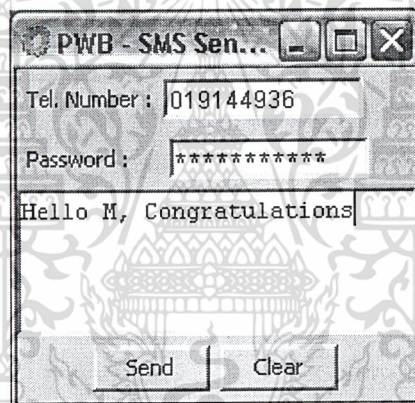
ดังนั้นจึงสามารถเรียกซีจีไอเพื่อส่งข้อความสั้นโดยใช้ urllib ของไพธอนได้ดังนี้

```
postDataDict = {'szGsmNo':telNo, 'szGsmPwd':password, 'szMesg':message}
```

```
encodedData = urlencode(postDataDict)
```

```
res = urlopen('http://www.gsm.ais900.com/gsmweb/web2sms.php', encodedData)
```

ถ้าผลตอบกลับมาไม่มีข้อผิดพลาดใดๆ ก็แสดงว่าข้อความสั้นได้ถูกส่งเรียบร้อยแล้ว



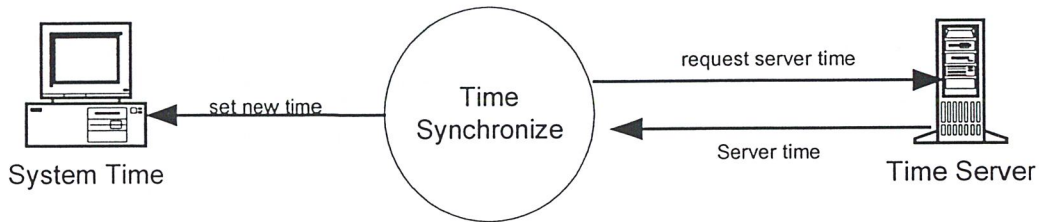
รูปที่ 5.16 รูปแสดงหน้าต่างของโปรแกรมส่งข้อความมือถือ

## 5.7 โมดูลเวลา

เวลาที่เอเจนต์ใช้อ้างอิงในการทำงาน โดยผู้ใช้สามารถเลือกได้ว่าจะให้เวลานี้ถูกตั้งค่าใหม่ทุกครั้ง  
 ที่เริ่มใช้งานโปรแกรมเอเจนต์ (โดยอ้างอิงจากเครื่องเซิร์ฟเวอร์) หรือไม่ก็ได้ การตั้งค่าเวลาสามารถทำได้  
 ผ่านตัวตั้งเวลา

### ตัวตั้งเวลา (Time Synchronizer)

คือส่วนที่ใช้ในการเซตเวลาให้กับเวลาระบบโดยที่ส่วนนี้จะทำหน้าที่ในการร้องขอข้อมูลเวลา  
 จากเซิร์ฟเวอร์ แล้วนำมาเซตให้กับเวลาระบบดังรูปที่ 5.17



รูปที่ 5.17 รูปแสดงการทำงานของตัวตั้งเวลา

โดยหลักการทำงานคือ จะให้ตัวเซตเวลาทำการร้องขอซีจีไอที่เก็บอยู่ที่ไทม์เซิร์ฟเวอร์โดยซีจีไอตัวนี้จะทำหน้าที่ส่งค่าเวลากลับมาในรูปแบบมิลลิวินาที (millisecond) และเพื่อให้เวลาที่เอเจนต์ใกล้เคียงกับเวลาของระบบมากที่สุดจึงมีการจับเวลาที่ใช้ในการขอข้อมูลเวลาจากเซิร์ฟเวอร์แล้วบวกเพิ่มเข้าไปก่อนเซตค่าให้กับเวลาระบบ ขั้นตอนการทำงานทั้งหมดสรุปได้ดังนี้

1. เก็บค่าเวลาปัจจุบันที่ไคลเอ็นต์ (เป็นข้อมูลชนิดลองก์) ซึ่งเป็นเวลาเริ่มต้นในการเริ่มร้องขอเวลา
2. ร้องขอเวลาจากเซิร์ฟเวอร์ และแปลงข้อมูลให้เป็นตัวเลขชนิดลองก์ (long) จะได้เวลาเซิร์ฟเวอร์
3. เก็บค่าเวลาปัจจุบันที่ไคลเอ็นต์อีกครั้ง (เป็นข้อมูลชนิดลองก์) ซึ่งเป็นเวลาสิ้นสุดการร้องขอ
4. ค่าเวลาของระบบคือ  $2 + (3 - 1)$

ซีจีไอที่ใช้ในการส่งค่าเวลากลับมาอาจเขียนด้วยภาษาใดก็ได้ซึ่งทางผู้พัฒนาได้เลือกจาว่าเซิร์ฟเล็ต (Java Servlet)

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.Date;

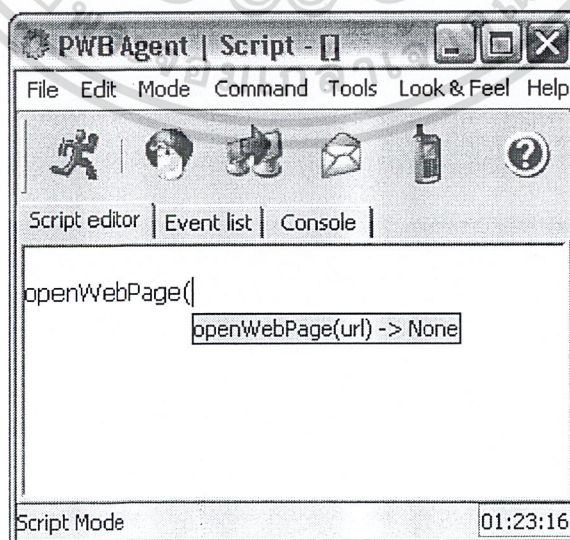
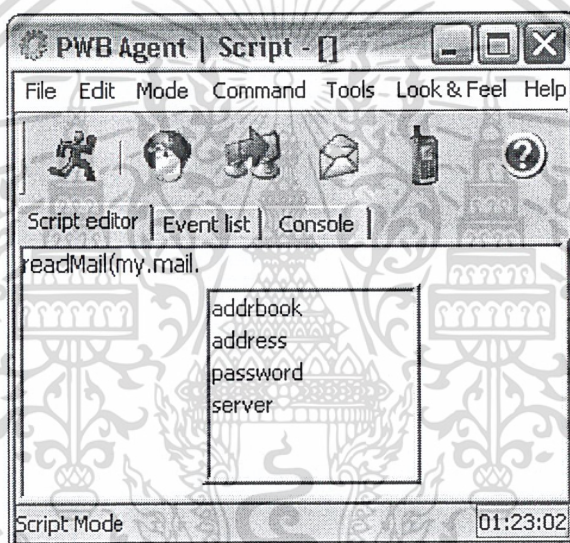
public class timecgi extends HttpServlet {

    public void doGet( HttpServletRequest request,
                      HttpServletResponse response )
        throws ServletException, IOException
    {
        PrintWriter output;
        response.setContentType( "text/html" ); // content type
        output = response.getWriter(); // get writer
        Date curr = new Date();
        output.println( curr.getTime() );
        output.close(); // close PrintWriter stream
    }
}
```

### 5.8 ระบบช่วยเหลือผู้ใช้งาน (Intelligent Script Editor)

เนื่องจากการที่มีฟังก์ชันช่วยเหลือมากมาย ซึ่งแต่ละตัวมีค่า parameter ที่ต้องการแตกต่างกันไปทำให้เป็นเรื่องยากในการจดจำ ผู้พัฒนาจึงพัฒนาระบบช่วยเหลือขึ้นมา 3 ส่วนดังนี้

1. ระบบช่วยเหลือฟังก์ชัน เป็นการช่วยบอก function signature ให้แก่ผู้ใช้ทำให้สามารถใช้งานฟังก์ชันได้อย่างถูกต้อง
2. ระบบช่วยเหลือออบเจกต์และโมดูล เป็นการช่วยบอก object, class, module members เป็นรายการออกมาและสามารถเรียกใช้ได้
3. ระบบช่วยเหลือทดลองภาษาสคริปต์ เป็นหน้าต่างคอนโซลที่รับคำสั่งจากผู้ใช้งานแบบโต้ตอบ หรือ Interactive PWB-script language console

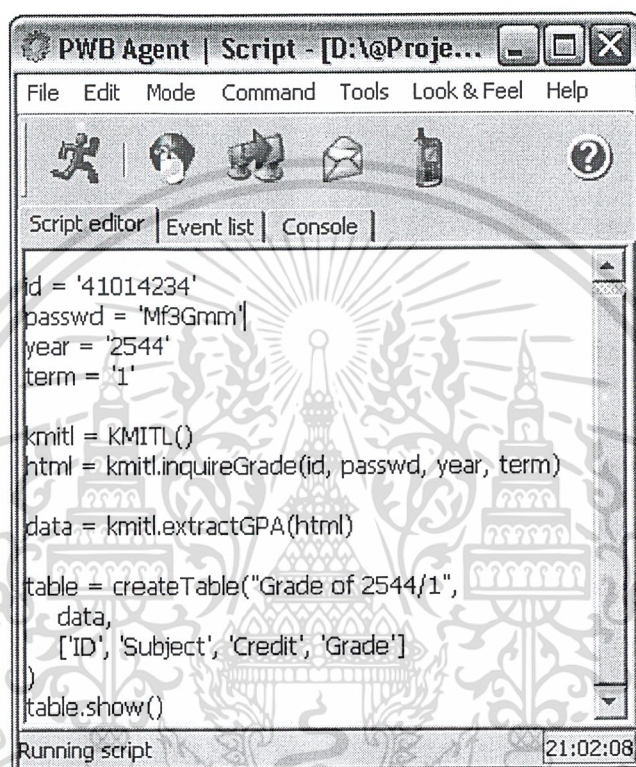


รูปที่ 5.18 รูปแสดงระบบช่วยเหลือผู้ใช้งาน

## บทที่ 6

## ผลการทดลองใช้โปรแกรม

ตัวอย่างที่ 1 สคริปต์สั่งการดึงข้อมูลเกรดจากเว็บสถาบันมาแสดงผลในรูปแบบตาราง



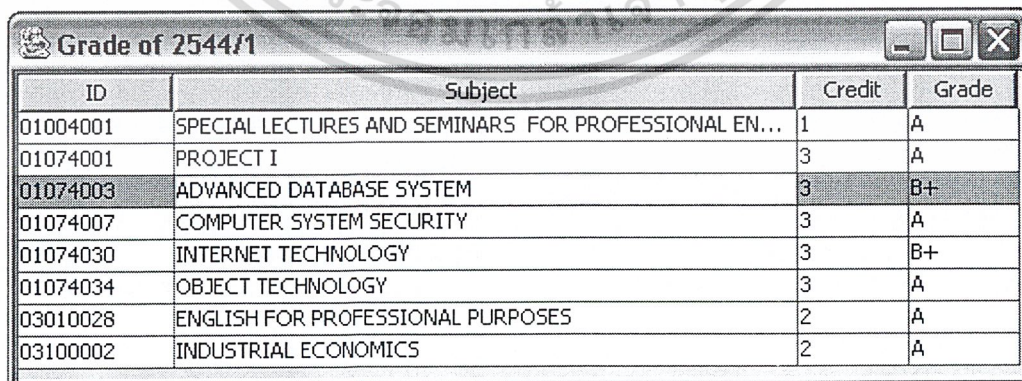
```

id = '41014234'
passwd = 'Mf3Gmm'
year = '2544'
term = '1'

kmitl = KMITL()
html = kmitl.inquireGrade(id, passwd, year, term)
data = kmitl.extractGPA(html)
table = createTable("Grade of 2544/1",
  data,
  ['ID', 'Subject', 'Credit', 'Grade']
)
table.show()
Running script
21:02:08

```

รูปที่ 6.1 ตัวอย่างสคริปต์ที่ 1



ID	Subject	Credit	Grade
01004001	SPECIAL LECTURES AND SEMINARS FOR PROFESSIONAL EN...	1	A
01074001	PROJECT I	3	A
01074003	ADVANCED DATABASE SYSTEM	3	B+
01074007	COMPUTER SYSTEM SECURITY	3	A
01074030	INTERNET TECHNOLOGY	3	B+
01074034	OBJECT TECHNOLOGY	3	A
03010028	ENGLISH FOR PROFESSIONAL PURPOSES	2	A
03100002	INDUSTRIAL ECONOMICS	2	A

รูปที่ 6.2 ผลลัพธ์จากตัวอย่างที่ 1

ตัวอย่างที่ 2 สคริปต์สั่งการดึงข้อมูลสภาพอากาศ (อุณหภูมิ) ของประเทศไทยตอนเหนือมาแสดงผล

```

w = Weather()
body = w.temperature('NORTH')
header = w.temperatureHeader
table = createTable("Weather in North Thailand",
    body,
    header)
table.show()

```

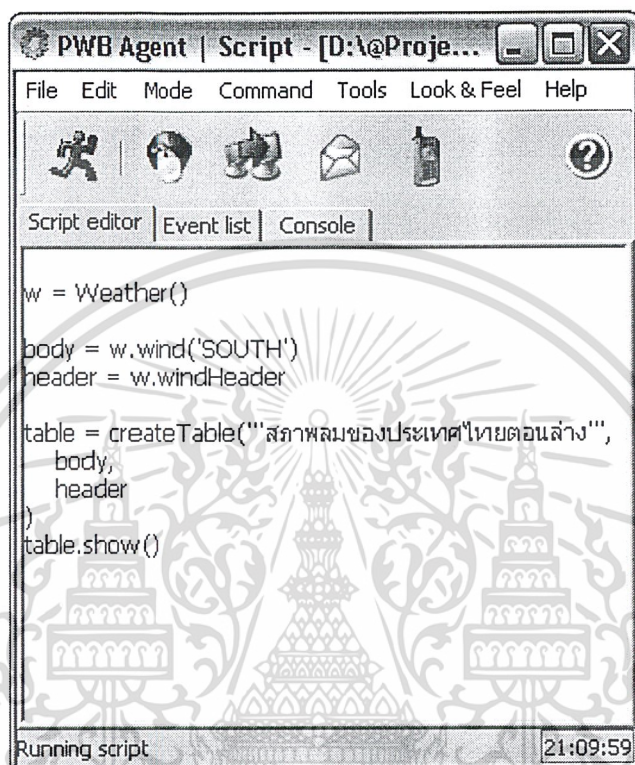
Running script 20:39:09

รูปที่ 6.3 ตัวอย่างสคริปต์ที่ 2

Station	C	F	dew
Chiang Mai	28.7	83.66	15.1
Chiang Rai	26.3	79.34	16.8
Kamphaeng Phet	31.2	88.16	24.0
Lampang	31.3	88.34	18.5
Lamphun	31.0	87.8	16.7
M.H.S.(Mae Sariang)	30.9	87.62	16.6
Mae Hong Son	30.1	86.18	15.5
Mae Jo(Agromet.)	29.6	85.28	20.4
Nan	30.5	86.9	14.0
Nan (Thung Chang Agromet.)	29.0	84.2	20.7
Nan(Agromet.)	28.0	82.4	20.4
Nan(Tha Wang Pha )	29.5	85.1	22.1
Phayao	28.5	83.3	20.5

รูปที่ 6.4 ผลลัพธ์จากตัวอย่างที่ 2

ตัวอย่างที่ 3 สคริปต์สั่งการดึงข้อมูลสภาพอากาศ (ลม) ของประเทศไทยตอนเหนือมาแสดงผล



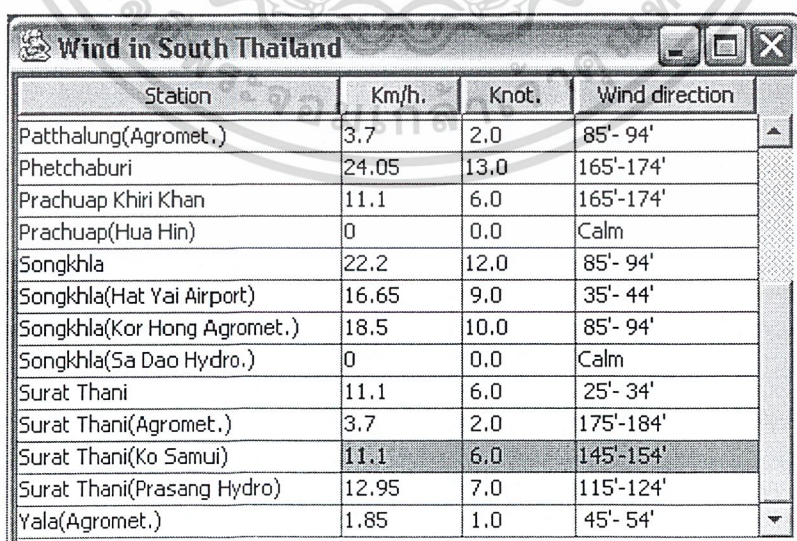
```

w = Weather()
body = w.wind('SOUTH')
header = w.windHeader
table = createTable("สภาพลมของประเทศไทยตอนล่าง",
body,
header
)
table.show()

```

Running script 21:09:59

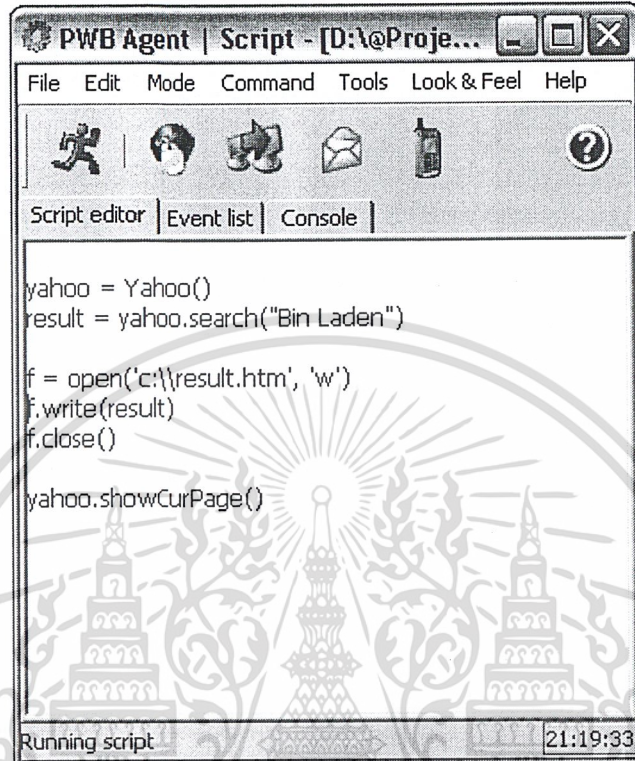
รูปที่ 6.5 ตัวอย่างสคริปต์ที่ 3



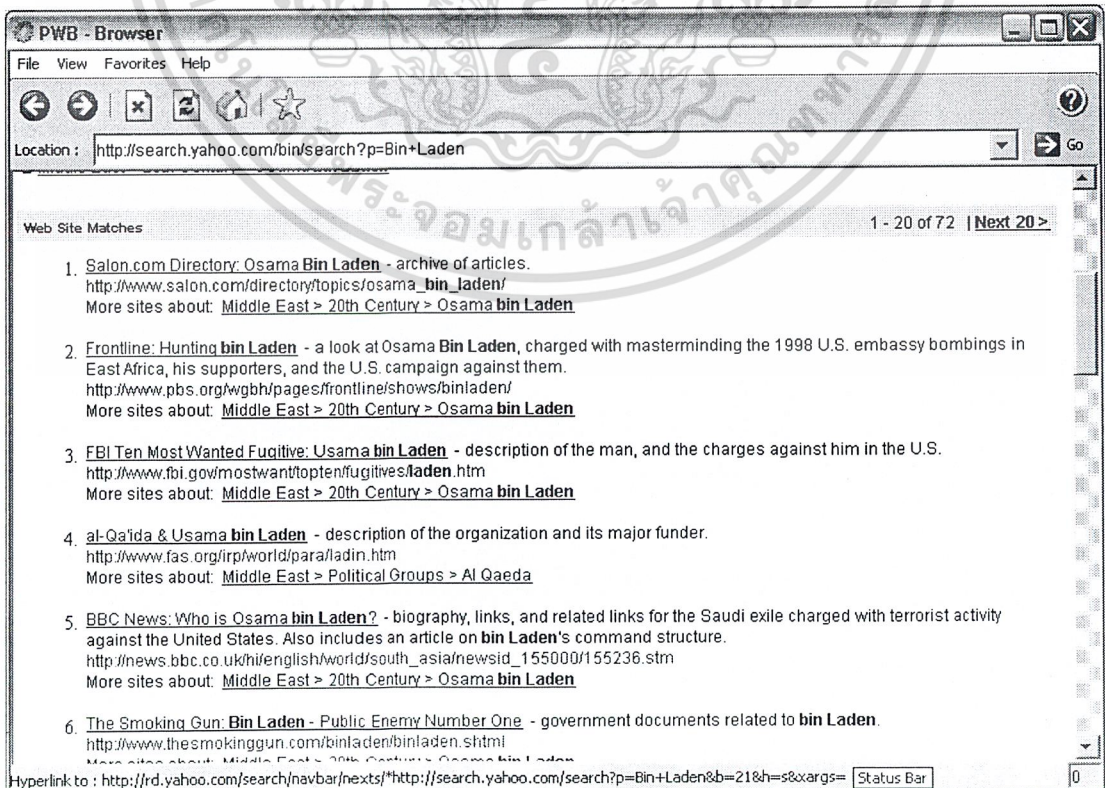
Station	Km/h.	Knot.	Wind direction
Patthalung(Agromet.)	3.7	2.0	85'- 94'
Phetchaburi	24.05	13.0	165'-174'
Prachuap Khiri Khan	11.1	6.0	165'-174'
Prachuap(Hua Hin)	0	0.0	Calm
Songkhla	22.2	12.0	85'- 94'
Songkhla(Hat Yai Airport)	16.65	9.0	35'- 44'
Songkhla(Kor Hong Agromet.)	18.5	10.0	85'- 94'
Songkhla(Sa Dao Hydro.)	0	0.0	Calm
Surat Thani	11.1	6.0	25'- 34'
Surat Thani(Agromet.)	3.7	2.0	175'-184'
Surat Thani(Ko Samui)	11.1	6.0	145'-154'
Surat Thani(Prasang Hydro)	12.95	7.0	115'-124'
Yala(Agromet.)	1.85	1.0	45'- 54'

รูปที่ 6.6 ผลลัพธ์จากตัวอย่างที่ 3

ตัวอย่างที่ 4 สคริปต์สั่งงานบราวเซอร์ให้ค้นหาข้อมูลเกี่ยวกับ “Bin Laden” เมื่อค้นหาเสร็จให้เขียนผลลัพธ์ลงเป็นไฟล์และแสดงผลออกทางจอภาพ

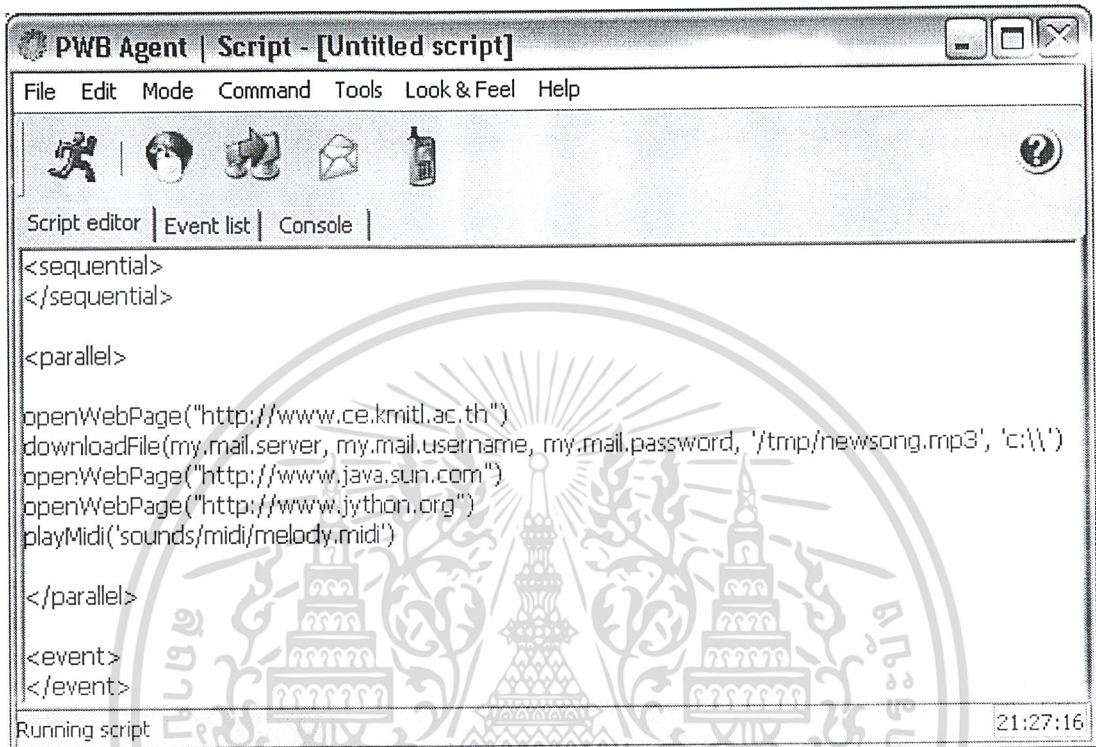


รูปที่ 6.7 ตัวอย่างสคริปต์ที่ 4



รูปที่ 6.8 ผลลัพธ์จากตัวอย่างที่ 4

ตัวอย่างที่ 5 สคริปต์แบบขนานสั่งงานหลายอย่างพร้อมๆกัน โดยจากตัวอย่างเป็นการเปิดหน้าต่างเว็บ บราวเซอร์ขึ้นมาใหม่ 3 หน้าต่าง, ถ่ายโอนเพิ่มข้อมูล, และเล่นเพลงมิดิ ไปพร้อมๆกันอย่างขนาน

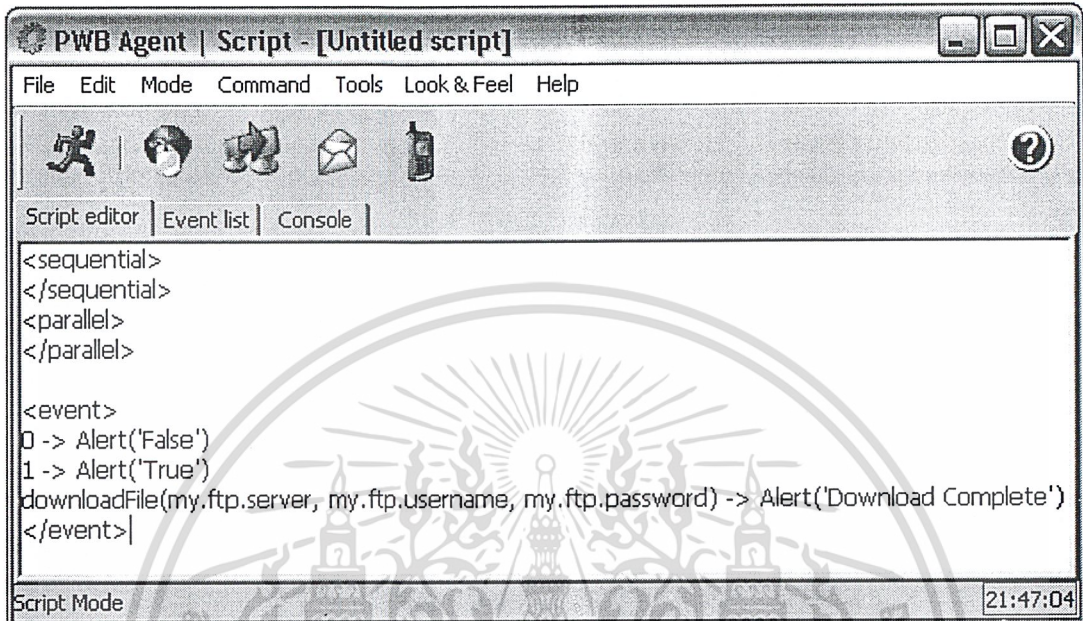


รูปที่ 6.9 ตัวอย่างสคริปต์ที่ 5

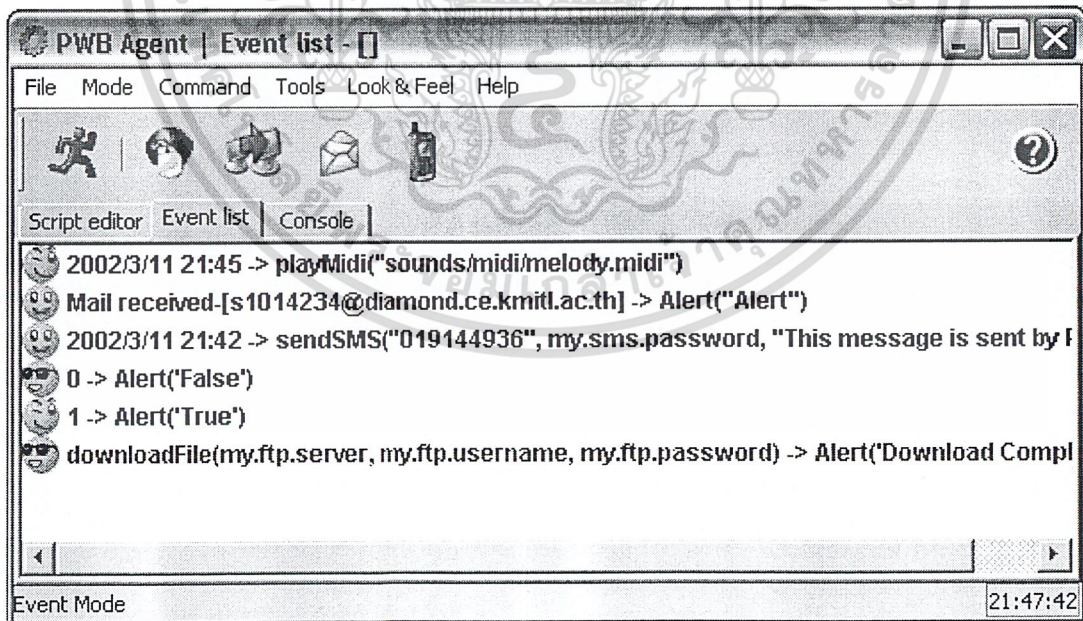


รูปที่ 6.10 ผลลัพธ์จากตัวอย่างที่ 5

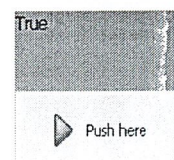
ตัวอย่างที่ 6 สคริปต์ในส่วนอีเวนต์ สั่งงานให้ทำงานเมื่อเงื่อนไขเป็นจริงขึ้นมาเท่านั้น ซึ่งเอเจนต์สามารถทำงานได้ อย่างถูกต้อง



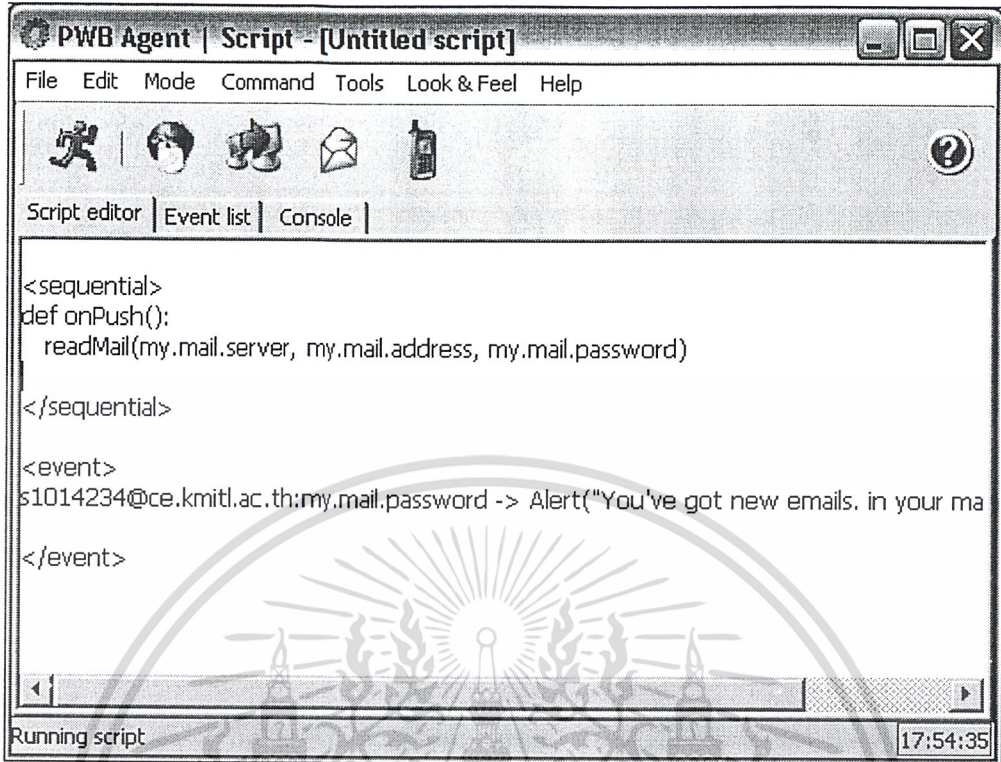
รูปที่ 6.11 ตัวอย่างสคริปต์ที่ 6



อีเวนต์ อันที่ 1 กับ 3 นั้นจะทำงานเมื่อถึงเวลาที่กำหนด ส่วนอันที่ 2 นั้นจะทำงานเมื่อ ได้รับเมลล์มาใหม่เท่านั้น

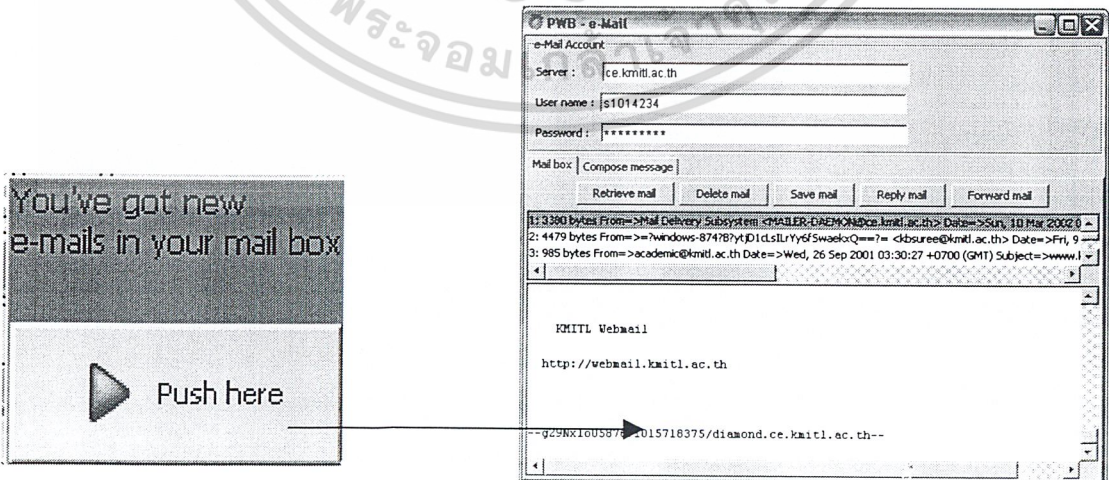


รูปที่ 6.12 ผลลัพธ์จากตัวอย่างที่ 6



รูปที่ 6.13 ตัวอย่างสคริปต์ที่ 7

เมื่อ ที่ e-mail address s1014234@ce.kmitl.ac.th ได้รับเมลี่ใหม่ จะเกิด action ขึ้น โดยแดงตัวรูปข้าง ๆ และเมื่อเรากดปุ่ม Push here ฟังก์ชัน onPush ก็จะถูกเรียกทำงานแล้วก็จะปรากฏ รายการเมลี่ออกมา ดังรูปที่ 6.14



รูปที่ 6.14 ผลลัพธ์จากตัวอย่างที่ 7

## บทที่ 7

### บทสรุป

#### 7.1 บทสรุป

ปริญญาโทฉบับนี้แสดงให้เห็นถึงการพัฒนาโปรแกรมเว็บเบราว์เซอร์ที่สามารถโปรแกรมได้ โดยเริ่มตั้งแต่การสำรวจเว็บเบราว์เซอร์ที่มีอยู่ในปัจจุบัน และเบราว์เซอร์ที่สามารถโปรแกรมได้โครงการแรกจากนั้นจึงเริ่มศึกษาความรู้ต่างๆที่นำมาใช้ในการพัฒนาโปรแกรม โดยเลือกภาษาไพธอนมาใช้ในการพัฒนาโปรแกรม เนื่องจากมีความเหมาะสมในการพัฒนาเอเจนต์ดังที่ได้กล่าวมาแล้ว จากนั้นจึงวางโครงสร้างของโปรแกรม และกำหนดความสามารถต่างๆที่เอเจนต์สามารถทำได้ แล้วจึงพัฒนาโปรแกรมเบราว์เซอร์ที่สามารถโปรแกรมได้มีความสามารถดังนี้

- ❖ เว็บเบราว์เซอร์สามารถอ่านไฟล์เอชทีเอ็มแอลต่างๆบนเครือข่ายอินเทอร์เน็ตได้
- ❖ สามารถสั่งงานในลักษณะของการทำงานตามเหตุการณ์ได้
- ❖ สามารถสั่งงานด้วยภาษาสคริปต์ได้โดยภาษาสคริปต์มีความสามารถของภาษาไพธอนทั้งหมด และมีซินแทกซ์เพิ่มเติมเพื่อเพิ่มความสามารถในการใช้งาน
- ❖ มีโมดูลที่ใช้ในการติดต่อผ่านอินเทอร์เน็ตที่สำคัญ 4 คือ เว็บเบราว์เซอร์ เอฟทีพี โคลเอ็นต์ แมล์ โคลเอ็นต์ และเอสเอ็มเอส โคลเอ็นต์ให้ผู้ใช้สั่งงานได้โดยตรง
- ❖ มีโมดูลที่เก็บฟังก์ชันช่วยเหลือซึ่งจะเป็นประโยชน์ในการใช้เป็นแอ็กชันเพื่อตอบสนองต่อเหตุการณ์ที่เกิดขึ้น
- ❖ เอเจนต์สามารถสั่งงานโมดูลต่างๆและฟังก์ชันที่ผู้ใช้กำหนดขึ้นได้โดยอัตโนมัติ
- ❖ สามารถดึงข้อมูลจากเว็บเพจ และสกัดเอาข้อมูลที่สำคัญมาใช้งานได้ตามต้องการ
- ❖ สามารถทำงานได้บนทุกแพลตฟอร์ม
- ❖ สามารถทำงานได้ตามที่ผู้ใช้กำหนด
- ❖ สามารถทำงานได้โดยอัตโนมัติตามข้อมูลในฐานความรู้
- ❖ สามารถเพิ่มเติมความสามารถได้อย่างไม่จำกัดโดยขึ้นอยู่กับสคริปต์ที่ผู้ใช้สร้างให้เอเจนต์
- ❖ มีความยืดหยุ่นในการใช้งาน และในการนำข้อมูลจากอินเทอร์เน็ตมาใช้งานให้มีประสิทธิภาพ

สิ่งที่ปรับปรุงจากโครงการเก่ามีดังนี้

- ❖ เว็บเบราว์เซอร์ต้องมีความสามารถใกล้เคียงถึงเทียบเท่าโปรแกรมเว็บเบราว์เซอร์ตามทั่วไปในปัจจุบัน คือต้องมีความสามารถจัดการประวัติการใช้งาน สามารถจัดการรายการยูอาร์แอลโปรด และต้องสามารถรายงานสถานการณ์ทำงานได้อย่างถูกต้อง
- ❖ เพิ่มความสามารถในเรื่อง ไฟล์ทรานสเฟอร์ โพรโทคอล และอีเมลล์
- ❖ ออกแบบโปรแกรมหลักเอเจนต์ให้มีการใช้งานได้ง่ายมากยิ่งขึ้น และมีระบบช่วยเหลือผู้ใช่มากขึ้น เช่น สคริปต์เอดิเตอร์ที่ชาญฉลาด, ระบบโพรไฟล์ผู้ใช้
- ❖ ภาษาสคริปต์ที่ใช้ต้องมีความสมบูรณ์, มีความสามารถที่หลากหลาย และใช้งานได้ง่าย

❖ จะต้องมีฟังก์ชันไลบรารี (Built-in Script Utilities) มากๆเพื่อการทำงานที่ครอบคลุม

## 7.2 วิจัยรณั

ปัญหา และข้อจำกัดของบราวเซอร์ที่สามารถโปรแกรมได้

- เริ่มการทำงานช้าเนื่องจากทำงานอยู่บนแพลตฟอร์มของจาวา
- ยังไม่สามารถแสดงผลเท็กเอชทีเอ็มแอลได้ครบถ้วน (สามารถแสดงได้ตามเอชทีเอ็มแอล 3.2) และยังไม่สามารถแสดงผลข้อมูลที่เป็นไมม์(Mime) และไคลเอ็นต์สคริปต์ เช่น จาวาสคริปต์(Java script) และวีบีสคริปต์(VB Script) ได้
- สนับสนุนโปรโตคอลเมล์ยังไม่ครบทุกรูปแบบ คือ สนับสนุนแค่โปรโตคอลป๊อป 3 เท่านั้น
- ไวยากรณ์ของคำสั่งสคริปต์ยึดติดกับภาษาไจซอนทำให้การเพิ่มไวยากรณ์ใหม่ๆเข้าไปบางครั้งอาจทำไม่ได้

การแก้ไข

- หาโมดูลที่ใช้ในการแสดงผลเอชทีเอ็มแอลที่มีประสิทธิภาพมากกว่านี้
- เขียนโปรแกรมให้ครอบคลุมทุกโปรโตคอลเมล์ เช่น IMAP และควรจะสามารถจัดการกับ MIME type ได้ทั้งหมดด้วย
- เขียนพาร์เซอร์ภาษาสคริปต์เอง

แนวทางในการพัฒนาต่อ

- ควรพัฒนาต่อให้เอเจนต์มีความชาญฉลาด และสามารถโปรแกรมตัวเองได้
- ควรมีความสามารถในการแกะข้อมูลจากเอกสารเอ็กซ์เอ็มแอลซึ่งเป็นมาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ต

## ภาคผนวก

# คู่มือการใช้งานโปรแกรมบราวเซอร์ที่สามารถโปรแกรมได้ เวอร์ชัน 2.1 (Manual Program PWB (Programmable Web Browser) version 2.1)

### 1. ความต้องการของระบบ (System requirement)

- ระบบปฏิบัติการ Windows 98/Me/2000/XP, Linux, Mac แต่ทำงานได้อย่างสมบูรณ์ที่สุดบน Windows XP Professional
- Java Runtime Environment (JRE) หรือ Java Develop Kit (JDK) 1.4
- สำหรับการทำงานของโปรแกรมโดยใช้ Jython นั้นระบบจะต้องลง Jython 2.1 ด้วย แต่ทางผู้พัฒนาได้สร้างเป็น Jar file เอาไว้ด้วยซึ่งสามารถทำงานผ่าน Java ได้โดยตรง
- เครื่องที่ลงโปรแกรมต้องสามารถเชื่อมต่อเข้าสู่อินเทอร์เน็ตได้ ถึงแม้โปรแกรมจะสามารถทำงานแบบ offline ได้ แต่การทำงานส่วนใหญ่ต้องใช้อินเทอร์เน็ต

### 2. การติดตั้งโปรแกรม (Program Installation)

- ❖ การลงโปรแกรม Java-Python หรือ Jython
  1. ลง JRE หรือ JDK 1.4
  2. เรียกโปรแกรม Jython Installer โดยรัน java jython2-1 และทำตามขั้นตอนของโปรแกรม
- ❖ การเรียกโปรแกรม PWB โดย Jython
 

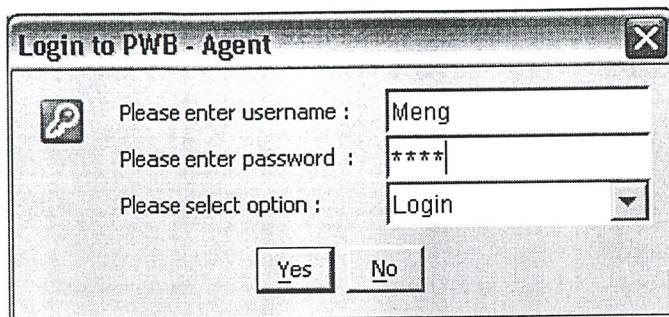
```
$ jython WebAgent.py
```
- ❖ การแปลง Jython file ให้เป็น Java class file
 

```
$ jythonc -d WebAgent.py
```
- ❖ การเรียกโปรแกรม PWB โดย Java
 

```
$ java -jar WebAgent.jar หรือ double-click ที่ไฟล์ WebAgent.jar
```

### 3. การใช้งานโปรแกรม PWB – Agent

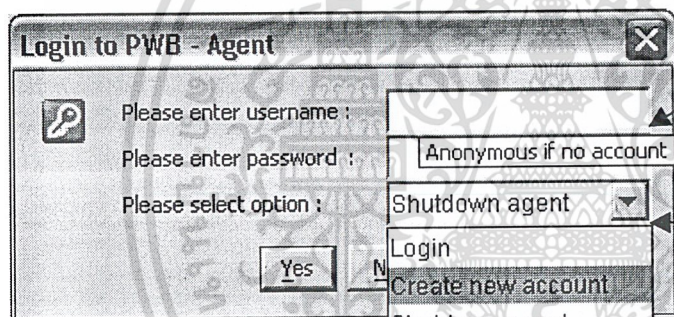
เมื่อเปิดโปรแกรมขึ้นมาจะพบกับ Login Dialog ดังรูปที่ 1



รูปที่ 1 Login Dialog

ที่ Login Dialog จะมี option ให้เลือกดังนี้

- Login : ให้กรอกชื่อผู้ใช้งานและรหัสผ่านที่เคยใช้งานมาก่อนแล้ว สำหรับผู้ใช้งานใหม่ให้เลือก option 2
- Create new account : ทำตามขั้นตอนเดียวกับ Login
- Shutdown agent : ปิดโปรแกรม

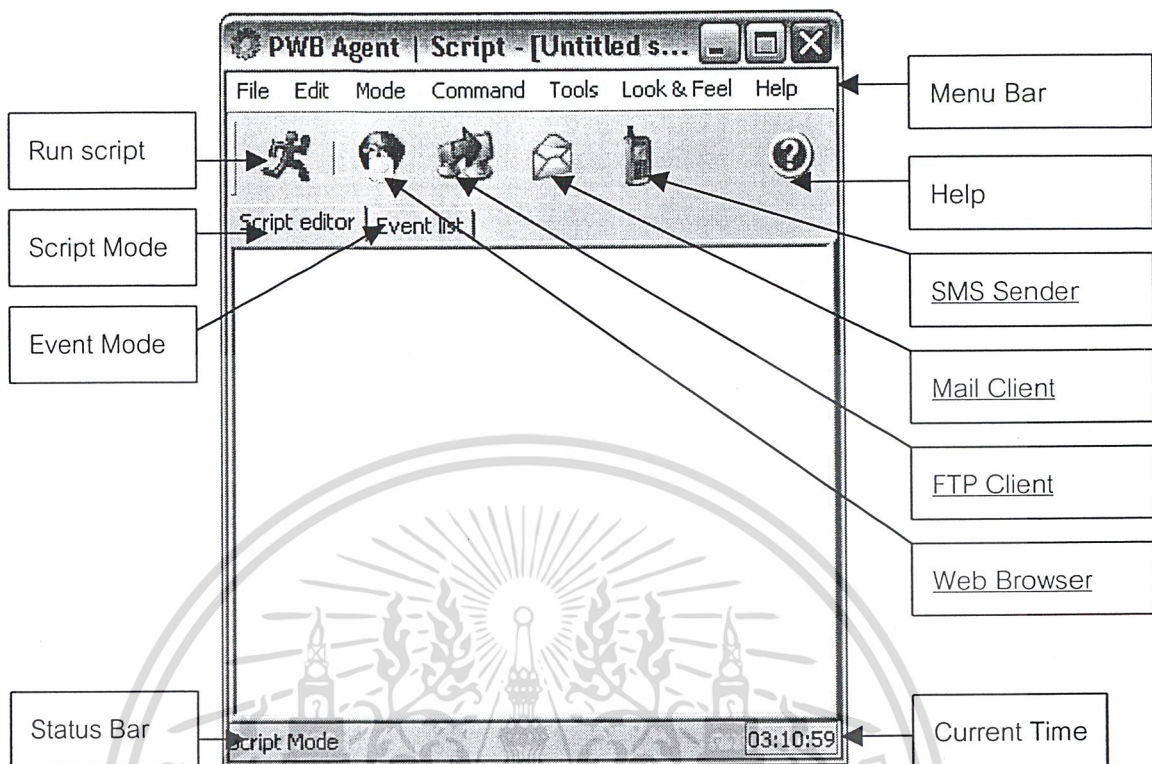


Input your username and password respectively, or type Anonymous for instant access

You may select an option here

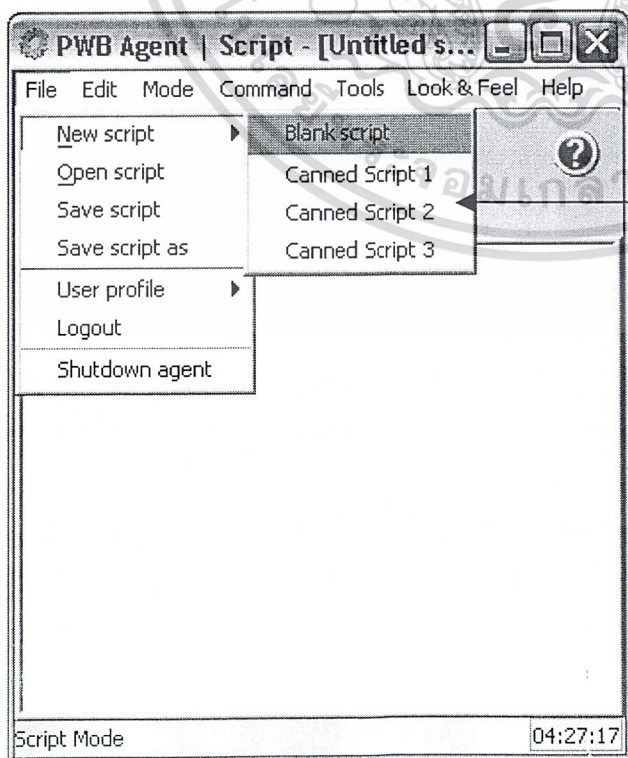
รูปที่ 2 Option in Login Dialog

**PWB - Agent** หลังจาก login เข้าระบบแล้วโปรแกรมจะ อ่านข้อมูลส่วนตัวผู้ใช้งาน (profile) และค่ากำหนดต่างๆ (setting) โดยค่าเหล่านี้สามารถแก้ไขได้โดย Edit profile หน้าตาและรายละเอียดของโปรแกรมแสดงดังรูปที่ 3



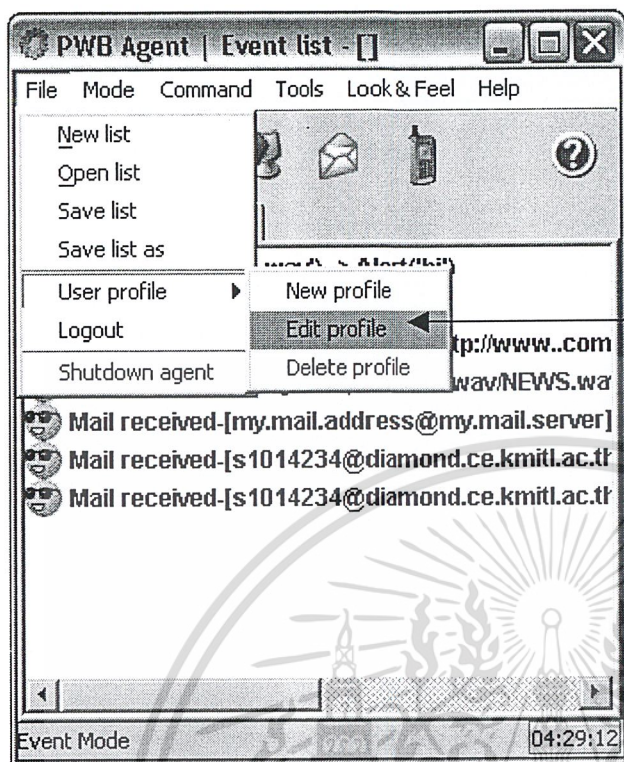
รูปที่ 3 หน้าต่างหลัก PWB - Agent

**File Menu** สร้าง, เปิด, บันทึก ไฟล์สคริปต์และอีเวนต์ โดยการใช้งานหลักๆ จะเหมือนกับ โปรแกรมทั่วไป การเปลี่ยนโหมดการทำงานมีผลทำให้ เมนูมีการเปลี่ยนแปลง



การใช้งานของ script และ event mode เหมือนกัน ส่วนแบบ script จะมี canned script เป็น template ให้เลือกใช้

รูปที่ 4 การใช้งาน File Menu



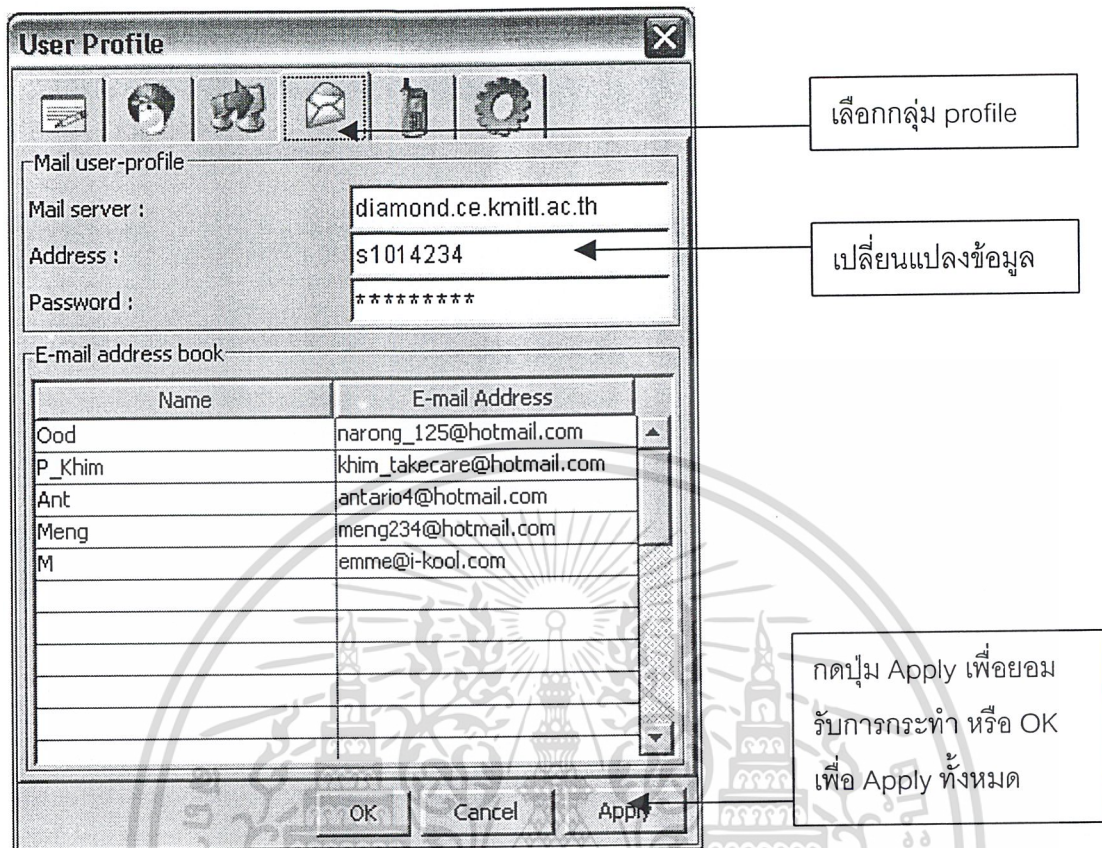
เมื่อเลือกที่ File > User profile > Edit profile จะปรากฏ User profile dialog ดังรูปที่ 6

### รูปที่ 5 การแก้ไขข้อมูลผู้ใช้ (profiles and settings)

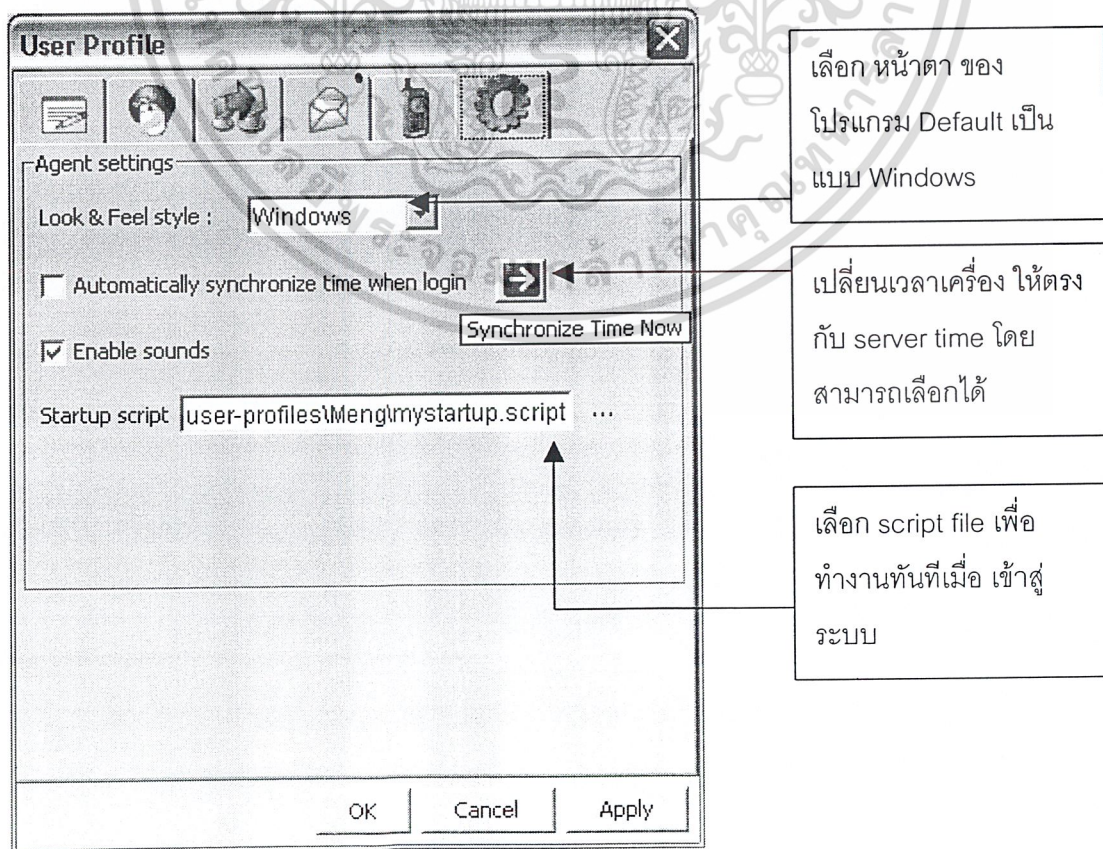
**User Profile Dialog** เมื่อเลือก File > User profile > Edit profile จะพบกับรูปที่ 6 ซึ่งเราสามารถกำหนดค่าต่างๆตามต้องการ โดยแบ่งเป็นหมวดหมู่ 6 กลุ่ม

1. Profile : รายละเอียดและประวัติของผู้ใช้งาน
2. Browser : รายละเอียดของโปรแกรมบราวเซอร์มี favorites และ cache URLs
3. FTP Client : รายละเอียดของโปรแกรมไฟล์ทรานซ์เฟอร์ มี FTP server, username, password, FTP favorites
4. Mail Client : รายละเอียดของโปรแกรมเมล มี Mail server, mail address, password, E-mail address book
5. SMS Sender : รายละเอียดของโปรแกรม SMS มี username, password, phonebook
6. Settings : รายละเอียดการกำหนดค่าของโปรแกรม PWB-Agent มี look&feel, server time, sound, and startup script

\* ข้อ 2-5 มีผลต่อการเปลี่ยนแปลงค่าตัวแปรใน โปรแกรมนั้นๆด้วย นอกจากนี้ค่าต่างๆ เหล่านี้สามารถ เรียกใช้ได้ทาง script editor ในนามของ my object รูปแบบการเรียกใช้คือ my.mail.address, my.browser.favorites['ImperialCollege'] เป็นต้น รายละเอียดกล่าวไว้ในเรื่อง script language



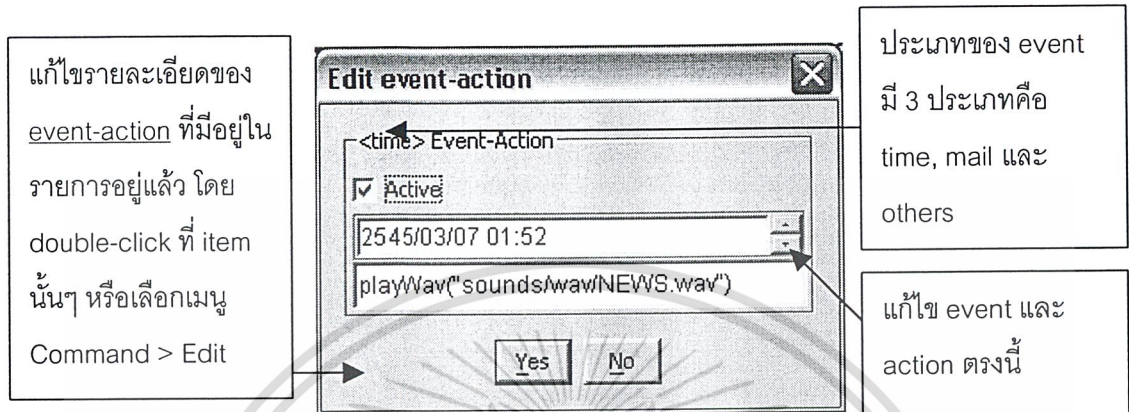
รูปที่ 6 User Profile > Mail



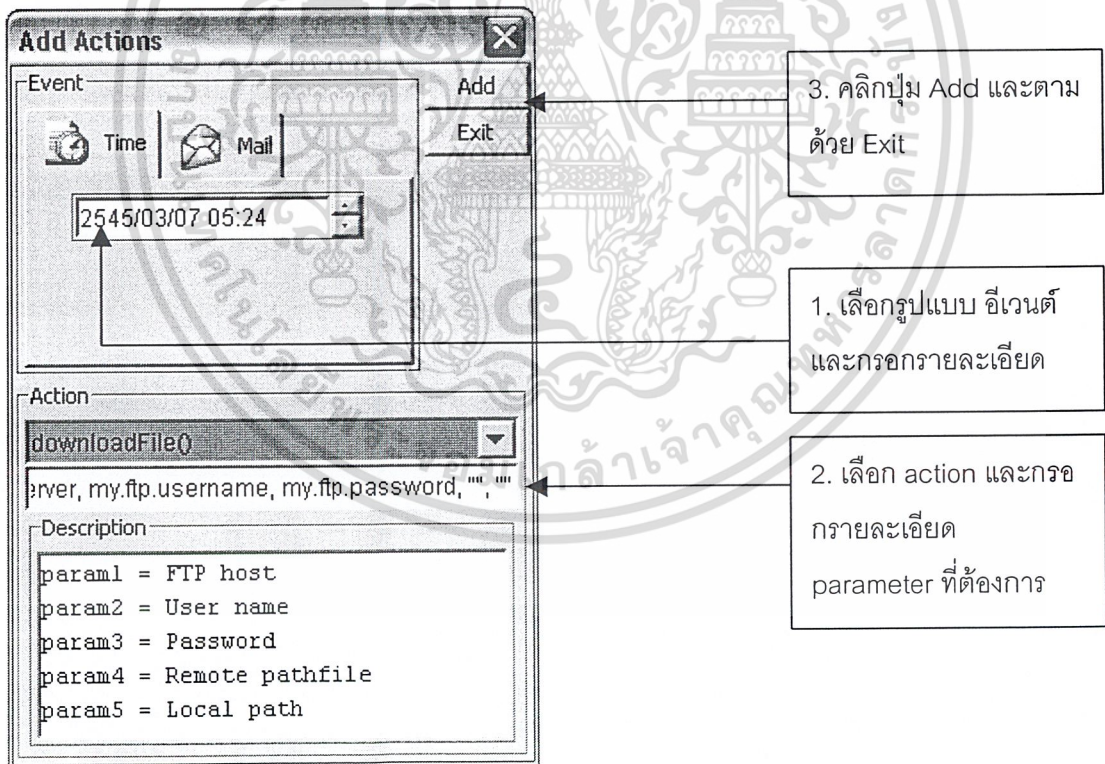
รูปที่ 7 User Profile > Settings

**Command Menu** มี 2 โหมดการทำงานคือ

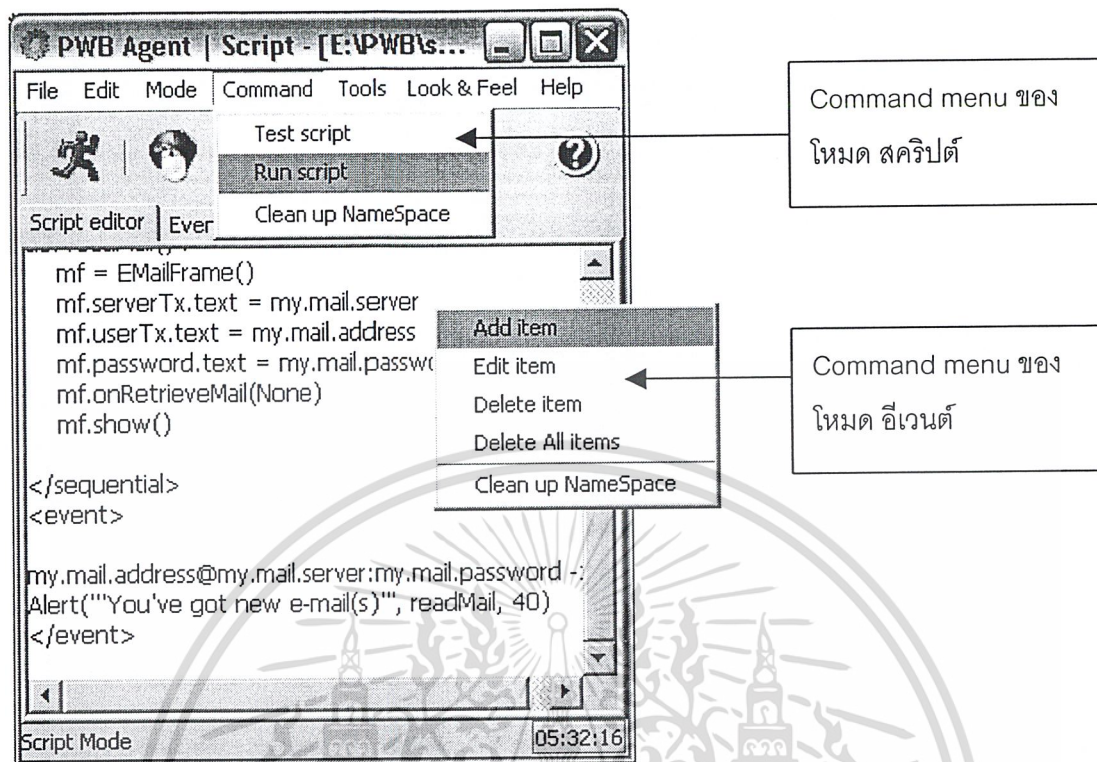
1. script mode : ทดสอบความถูกต้องของสคริปต์, run script, clear name space
2. event mode : เพิ่มอีเวนต์, แก้ไขอีเวนต์, ลบและลบทั้งหมด ค้างรูปข้างล่าง



รูปที่ 8 Edit event-action

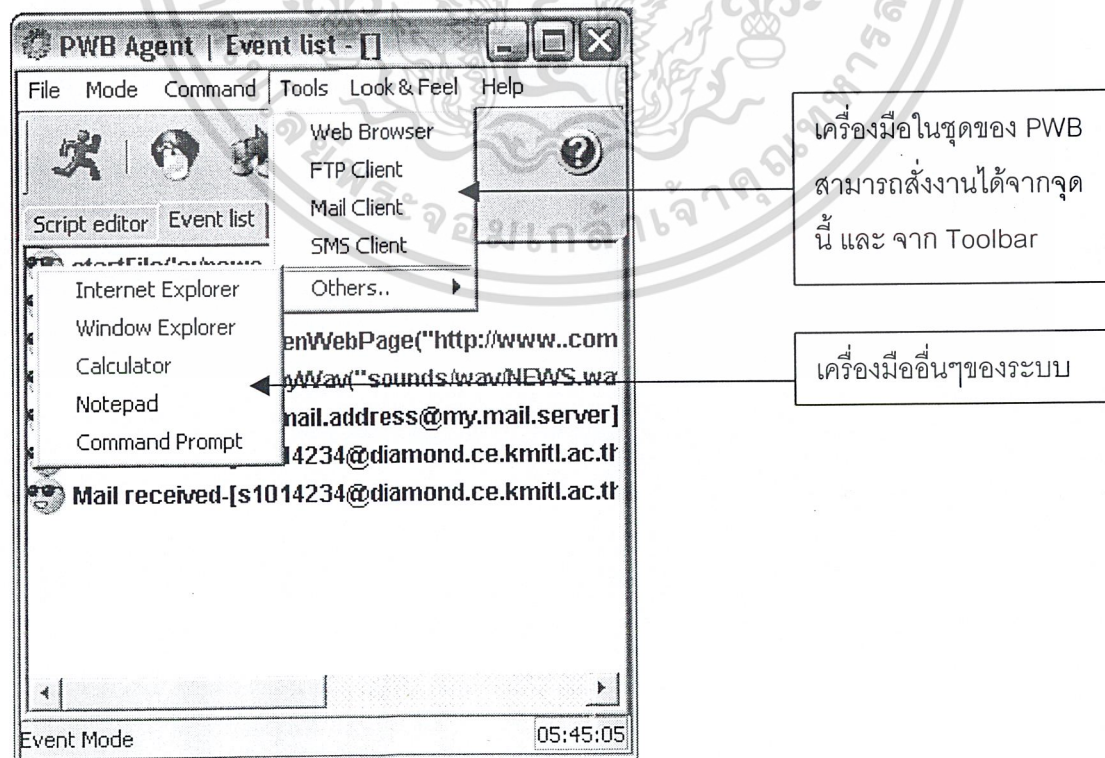


รูปที่ 9 Add Acitons



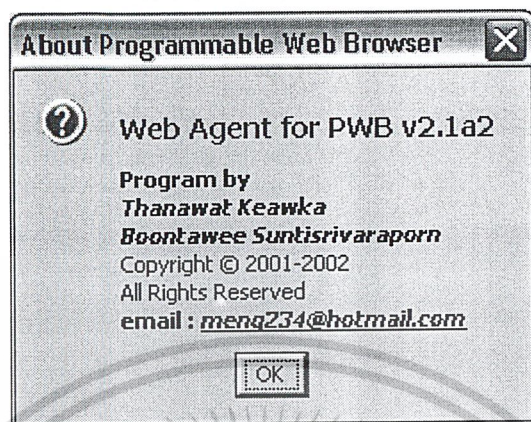
รูปที่ 10 Command Menu

**Tools Menu**



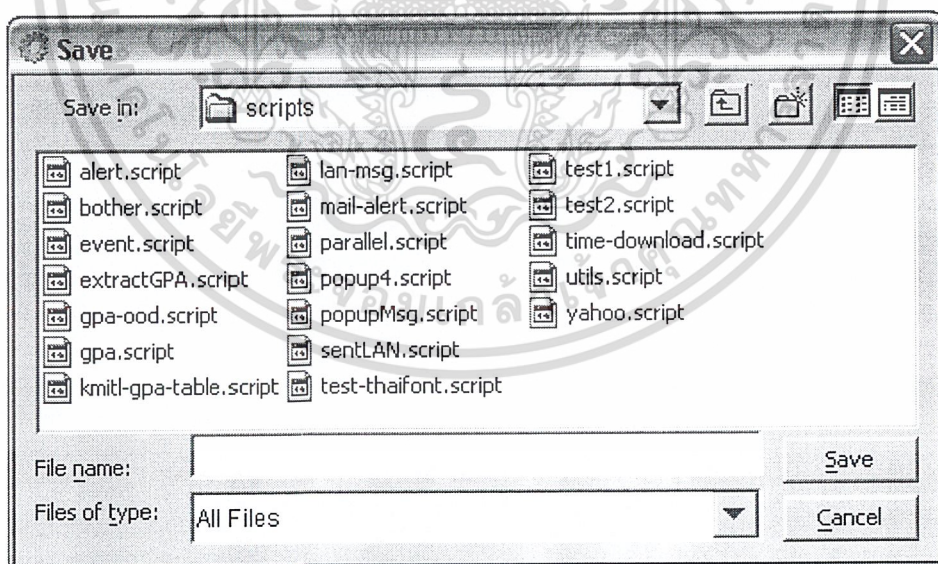
รูปที่ 11 Tools Menu

### About Programmable Web Browser Dialog



รูปที่ 12 About Programmable Web Browser Dialog

**(Intelligent) Script Editor** เป็นหน้าต่างเขียน และอ่าน script language ที่ใช้สั่งงาน Agent ข้อเด่นของ script editor นี้คือมีความสามารถในการ show tool tip เพื่อความสะดวกต่อผู้ใช้งาน โดยสามารถบันทึก และเปิดไฟล์ได้ดังรูป ส่วน syntax และ รายละเอียดของภาษาสคริปต์ จะกล่าวในเรื่อง Script language



รูปที่ 12 Save script Dialog

The screenshot shows the PWB Agent Script editor interface. The title bar reads "PWB Agent | Script - [Untitled]". The menu bar includes "File", "Edit", "Mode", "Command", "Tools", "Look & Feel", and "Help". The toolbar contains icons for running, undo, redo, save, and help. The main area is split into "Script editor" and "Event list". The script editor contains the following code:

```

newmails = checkMail(my.mail.address,
                    my.mail.browser,
                    my.mail.ftp,
                    my.mail.mail,
                    my.mail.sms);

if (newmails > 0) :
    Alert("You've got new e-mail(s)", readMail, 40)
    Alert(msg [,func [,delay]] -> popup obj)
    
```

Annotations on the right side of the image explain the components:

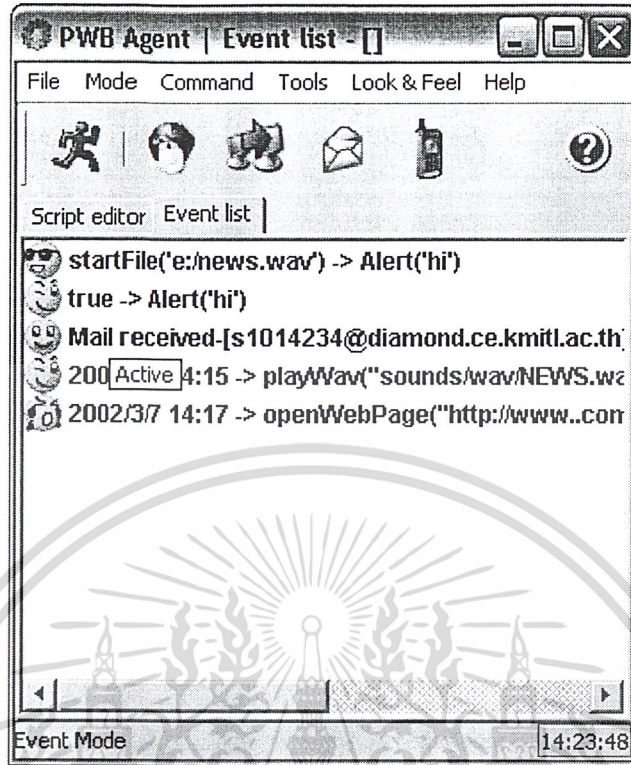
- Title บอกรหัสชื่อไฟล์ของ สคริปต์**  
ปัจจุบัน Untitled หรือ ว่างเปล่า คือ สคริปต์ยังไม่มีชื่อ
- พื้นที่ใช้เขียนสคริปต์** (points to the script editor area)
- เมื่อ script editor ตรวจสอบภาษา แล้วจะ แสดงข้อความช่วยเหลือแก่ผู้ใช้งาน** ซึ่งมีอยู่ 2 รูปแบบ คือ module member list และ definition prototype (points to the dropdown menu showing browser, ftp, mail, sms)
- เราสามารถตรวจสอบความถูกต้องของภาษา โดยยังไม่ run จริงๆ ได้** (points to the "Script correct" status bar)

รูปที่ 13 Intelligent Script Editor

**Event List Monitor** เป็นรายการของ event->action ในขณะใดๆ เราสามารถเพิ่ม แก้ไข ลบ รายการหนึ่งรายการใดในนี้ได้ดังกล่าวแล้วข้างต้น ในเรื่อง Command in event mode ตัวอย่างรายการใน event-action list

The diagram shows an entry in the Event List Monitor: "Mail received-[s1014234@diamond.ce.kmitd.ac.th] -> Alert("you've got new mails", readMail)".

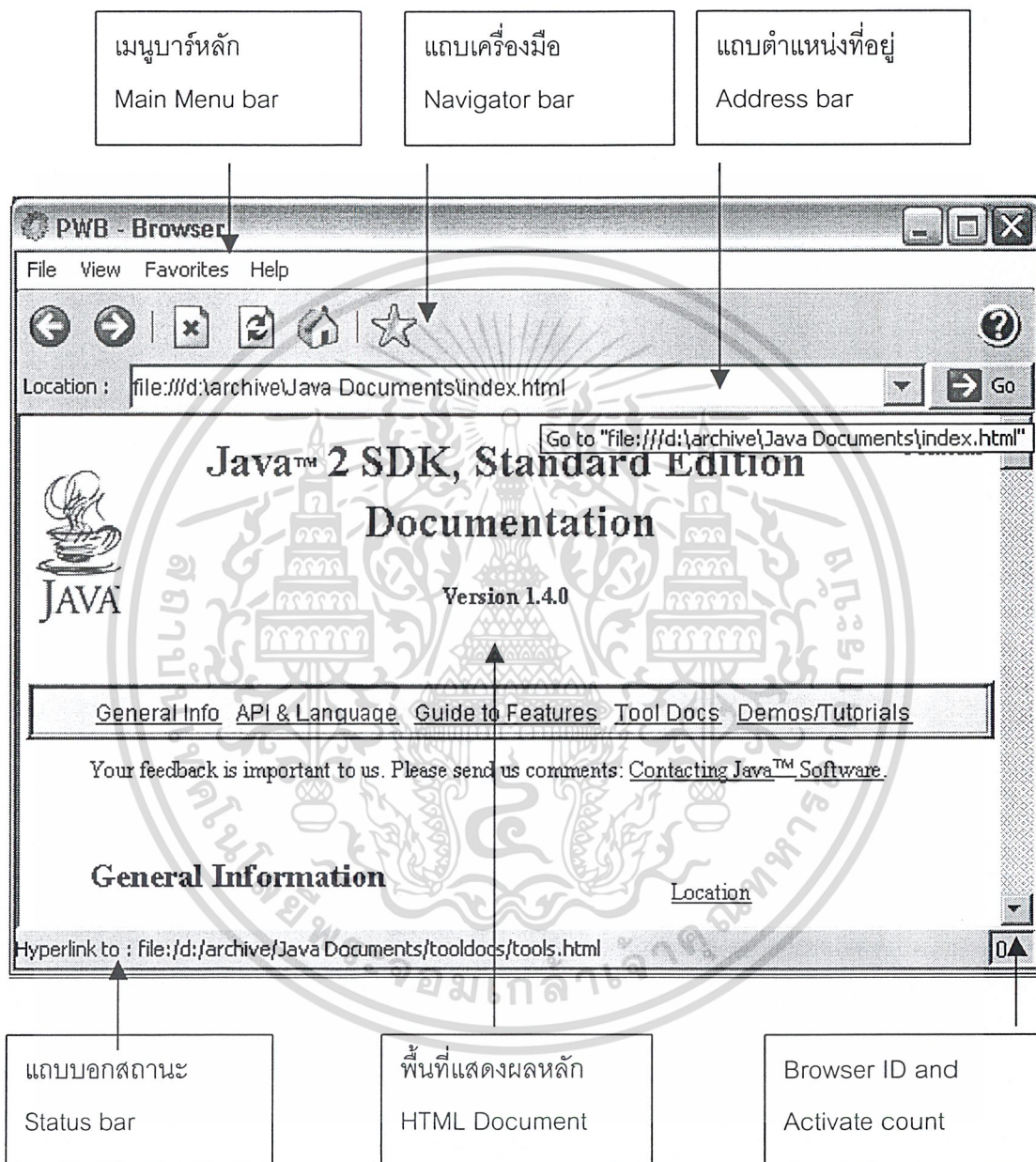
- Icon บอกรหัสสถานะของ event-action** นี้ซึ่งมีอยู่ 4 สถานะคือ active, halt, done, และ fail (points to the icon on the left)
- Event มี 3 ประเภทคือ time, mail received, และ others** (points to the event name)
- Action สามารถเป็นได้ทั้ง PWB Built-in function หรือ User-defined function** (points to the action name)



รูปที่ 14 Event list monitor

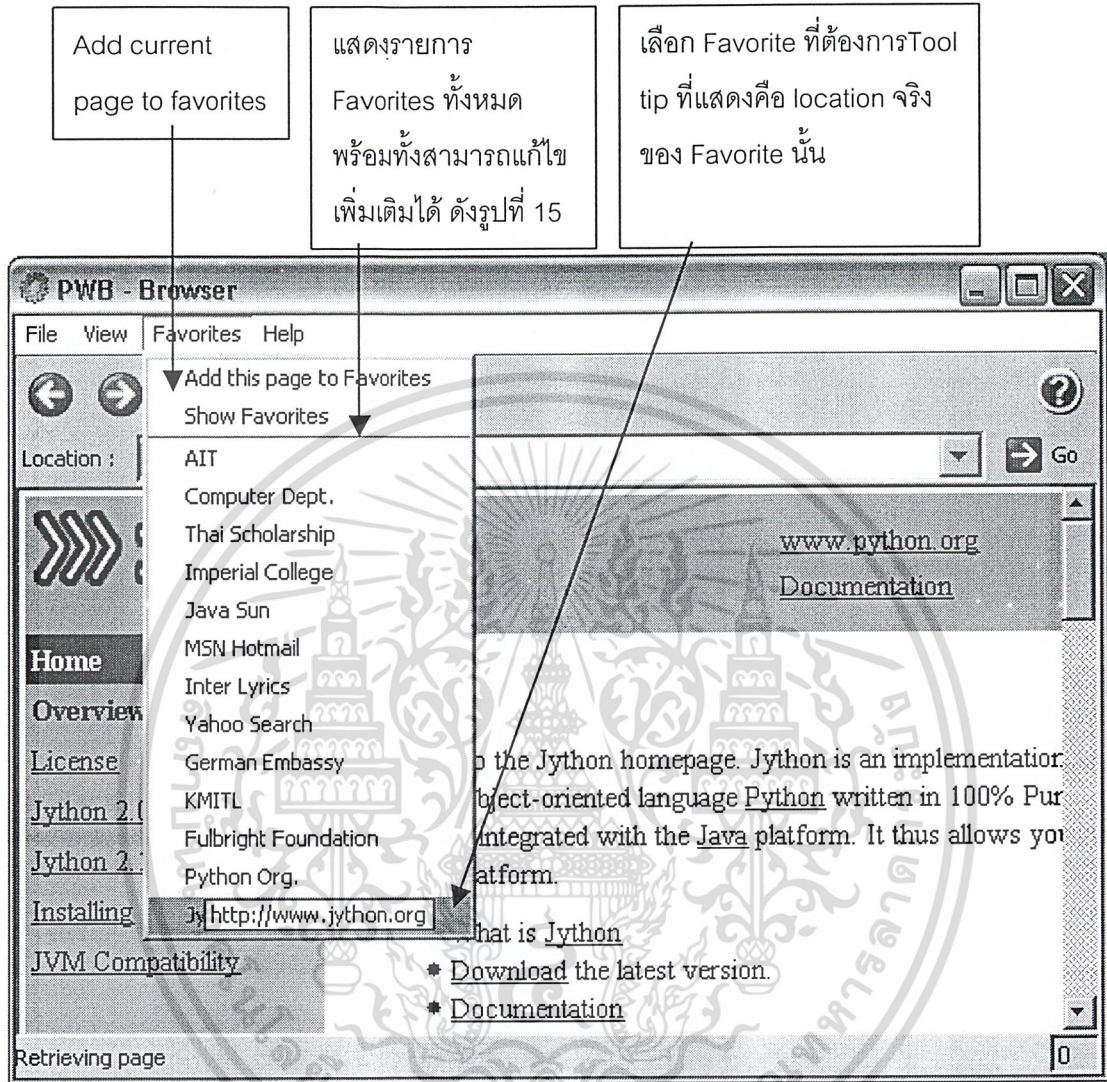
#### 4. การใช้งานโปรแกรม PWB – Browser

การใช้งานโดยรวมจะเหมือนกับ Web Browser ทั่วไป ดังรูปที่ 13

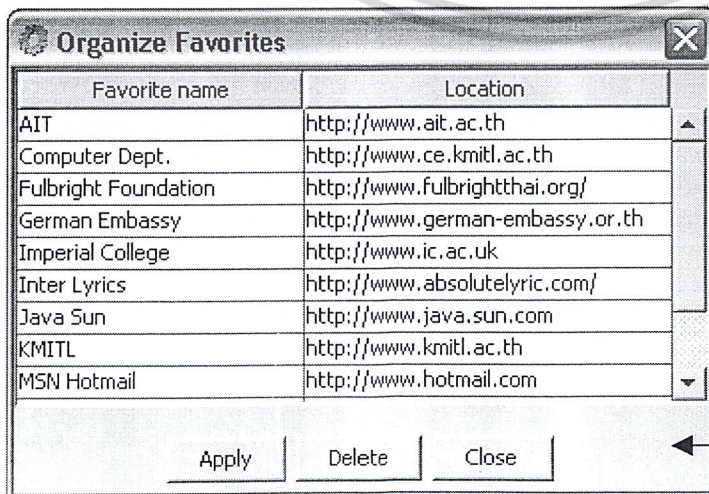


รูปที่ 13 PWB – Browser

**Favorites Menu** ใช้เลือกเว็บเพจที่เราชอบและเข้าเป็นประจำ



รูปที่ 14 การใช้งาน Favorites

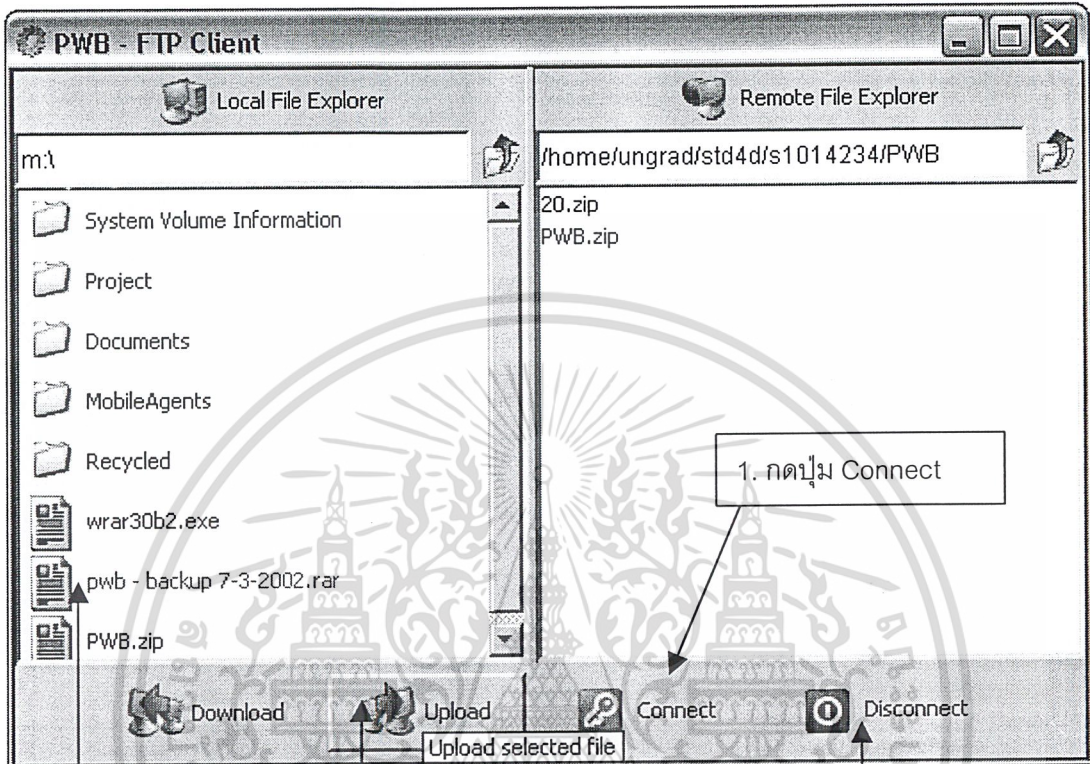


Organize Favorites Dialog ใช้สำหรับ แก้ไข และ ลบ Favorites นอก จากนั้น Favorites ยังสามารถ แก้ไขได้ผ่านทาง File > User profile > Edit profile > Browser ของ PWB-Agent อีกด้วย

รูปที่ 15 Organize Favorites Dialog

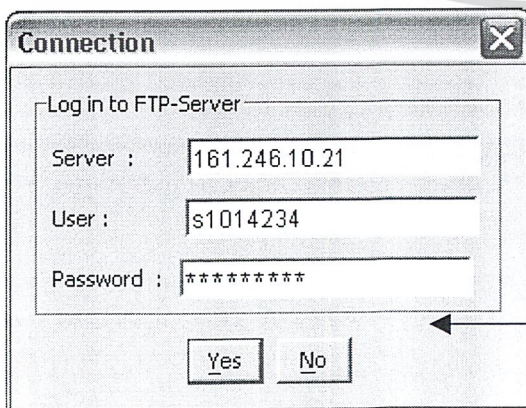
## 5. การใช้งานโปรแกรม PWB – FTP Client

เมื่อเปิดโปรแกรมขึ้นมา และกดปุ่ม Connect จะพบกับ dialog ดังรูปข้างล่าง ให้เรากรอกรายละเอียดให้ครบ และกดปุ่ม Yes เมื่อเชื่อมต่อสำเร็จก็จะปรากฏ dialog ดังรูปที่ 18



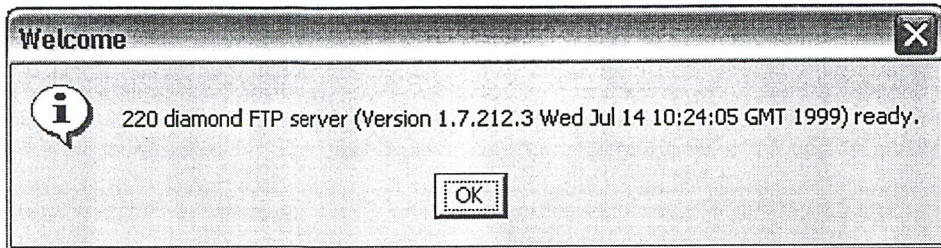
รูปที่ 16 PWB – FTP Client

- |   |  |  |
|---|--|--|
| <p>3. เลือกไฟล์ที่ต้องการ upload ในช่อง Local File Explorer ในขั้นตอนนี้เราสามารถเลือกหลายไฟล์ได้</p> | <p>4. กดปุ่ม Upload เพื่อทำการถ่ายโอนเพิ่มข้อมูลที่เลือกไปยัง current remote directory</p> | <p>5. กดปุ่ม Disconnect เพื่อปิดการเชื่อมต่อ</p> |
|---|--|--|



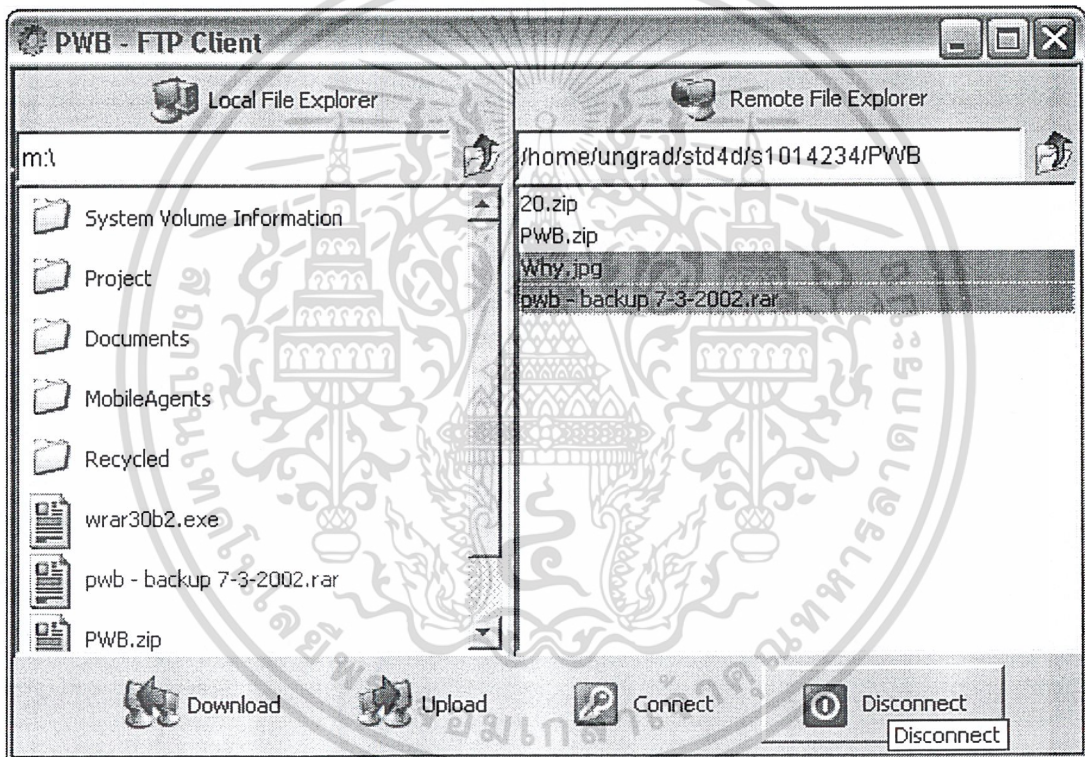
2. กรอกรายละเอียดให้ครบ แล้วกดปุ่ม Yes เพื่อเชื่อมต่อเข้าสู่ FTP server

รูปที่ 17 Connection Dialog



รูปที่ 18 Welcome Dialog

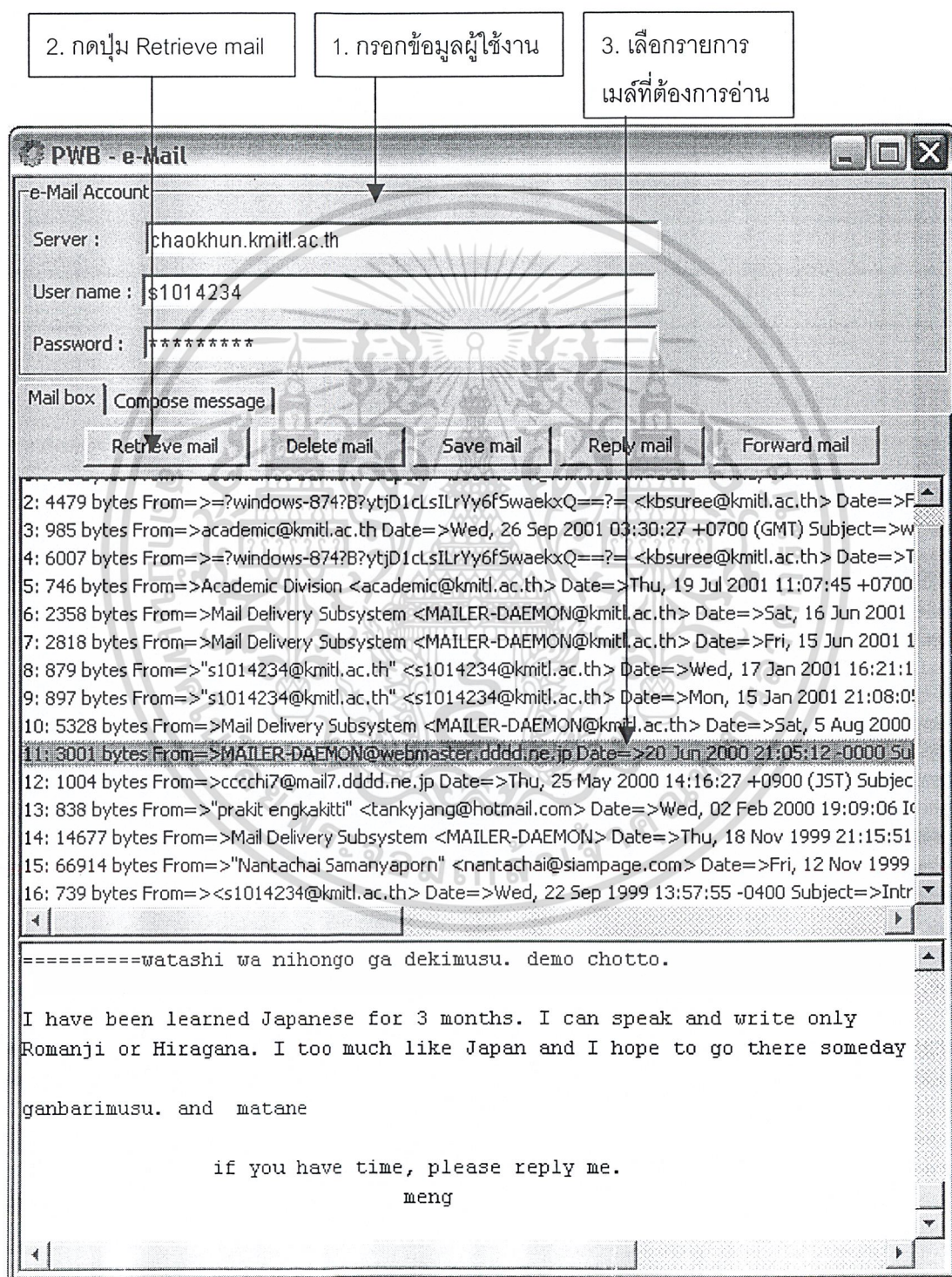
สำหรับการ Download ก็มีขั้นตอนใกล้เคียงกับการ Upload โดยเลือกไฟล์ที่ Remote file explorer และเลือก local directory ที่ต้องการ จากนั้นกดปุ่ม Download เป็นอันเรียบร้อย



รูปที่ 16 PWB - FTP Client

## 6. การใช้งานโปรแกรม PWB – Mail Client

โปรแกรมนี้ใช้ได้กับ Mail Server ที่รองรับ protocol POP3 เท่านั้น ไม่สามารถนำมาใช้กับ Web Mail เช่น Hotmail.com, Thaimail.co.th, หรือ webmail1.kmitl.ac.th ได้ ขั้นตอนการใช้งานดูได้ดังรูปที่ 17 และ 18



รูปที่ 17 PWB – Mail Client (Mailbox)

จากรูปที่ 17 เราสามารถกดปุ่ม Reply mail เพื่อตอบกลับได้ จะปรากฏหน้าต่าง Compose message ดังรูปที่ 18

ขั้นตอนการ Compose message

1. กรอกข้อมูล e-mail ของผู้รับปลายทาง
2. กรอกหัวเรื่อง (subject) และเนื้อเรื่อง (body)
3. กดปุ่ม Send

PWB - e-Mail

e-Mail Account

Server : chaokhun.kmitl.ac.th

User name : s1014234

Password : \*\*\*\*\*

Mail box Compose message

To : MAILER-DAEMON@webmaster.dddd.ne.jp

From : s1014234@chaokhun.kmitl.ac.th

Subject : Re:failure notice

I'm looking forward to your reply.

>>

>> matane. chigu

>>

>>

>>

>>

>Ohayo gozaimusu, hajimemashite

>

>watashi wa meng desu. And now I'm 20 years old.

>I'm studing at KMITL at the 3rd year of computer engineering.

>My hometown is in ChiangRai, the northeast province in Thailand, however

>now a day, I live in Bangkok to study

>

>=====watashi wa nihongo ga dekimusu. demo chotto.

>

>I have been learned Japanese for 3 months. I can speak and write only

>Romanji or Hiragana. I top much like Japan and I hope to go there someday

>

>ganbarimusu. and matane

>

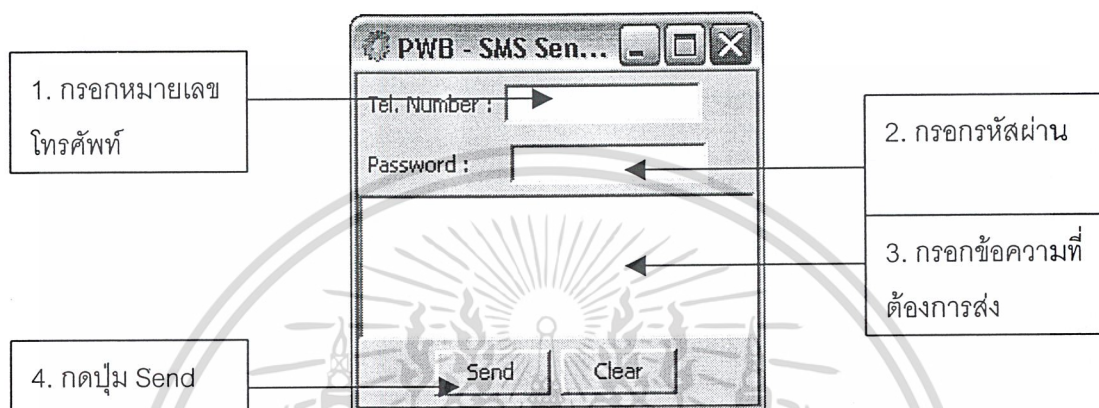
Send Clear

รูปที่ 18 PWB – Mail Client (Compose Message)

## 7. การใช้งานโปรแกรม PWB – SMS Sender

เป็นโปรแกรมย่อยตัวเล็กๆ ที่อำนวยความสะดวกแก่ผู้ใช้งาน อีกทั้งยังสามารถใช้ร่วมกับการทำงานของภาษาสคริปต์ได้อีกด้วย

เนื่องจากปัจจุบันยังไม่มีผู้ให้บริการเว็บรายใดที่ให้บริการส่งข้อความ (Short Message) ผ่านทุกระบบโดยไม่เก็บค่าบริการ ทางผู้พัฒนาเลยทดลองระบบกับบริการของ AIS 900 เท่านั้น



รูปที่ 19 PWB – SMS Sender

## 8. การเขียน Script language

โครงสร้างและไวยากรณ์ภาษาสคริปต์ที่ใช้ คือ ภาษา Jython หรือ Python ซึ่งมีข้อดีของภาษานี้คือ ความง่าย (simplicity) และ มีความสามารถมากมาย (powerful) ผู้ใช้งานเพียงรู้ syntax พื้นฐาน ประกอบกับ สคริปต์อำนวยความสะดวก (PWB Built-in script) ก็สามารถทำงานได้อย่างง่ายดาย เนื่องจากมีการช่วยเหลือจาก Intelligent Script Editor

## บรรณานุกรม

- [1] วชิระ ศิริพจนาวรรณ, สุพัฒนดา โชติพันธ์ : บราวเซอร์ที่สามารถโปรแกรมได้ : ปรินูญานิพนธ์  
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ  
ทหารลาดกระบัง ปีการศึกษา 2543
- [2] Stephen A. Thomas : HTTP Essentials “ Protocols for Secure, Scaleable Web Sites” : Wiley  
Computer Publishing, John Wiley & Sons, Inc. 2001.
- [3] Martin C. Brown : Python Annotated Archives : Osborne McGraw-Hill 2000.
- [4] Wesley J. Chun : Core Python Programming : Prentice Hall 2001.
- [5] Andre S. Lessa : Python : Developer's Handbook :Sams 2001.
- [6] Mark Lutz : Programming Python (2<sup>nd</sup> Edition) : O'Reilly & Associates 2001.
- [7] Mark Hammond & Andy Robinson : Python “Programming on Win32” : O'Reilly &  
Associates 2000.
- [8] H.M. Deitel, P.J. Deitel, K. Steinbuhler : e-Business and e-Commerce for Manager : Deitel,  
Prentice-Hall, Inc. 2001.
- [9] Andrew S. Tanenbaum : Computer Networks : Prentice-Hall, Inc. 1996.
- [10] H.M.Deitel, P.J.Deitel : Java How to program Fourth edition : Prentice Hall 2001.
- [11] Robert W. Bill : Jython for Java programmer :New Riders
- [12] สุรศักดิ์ สงวนพงษ์ : สถาปัตยกรรมและโปรโตคอลที่ซีพี/ไอพี :บริษัท ซีเอ็ดดูเคชั่น จำกัด  
(มหาชน)
- [13] “Intelligent Sostware Agents on the Internet,” <http://www.hermans.org/agents/index.html>
- [14] “Python Official Homepage,” <http://www.python.org>
- [15] “Jython Official Homepage,” <http://www.jython.org>
- [16] “The Source for Java Technology,” <http://java.sun.com>