

การพัฒนาเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ
Developing A Web Content Monitoring Agent

โดย

นางสาววิศรา กิรติศิริกุล

รหัส 44067021

วัน เดือน ปี..... 23 ส.ค. 2551

เลขทะเบียน..... 01940

เลขเรียกหนังสือ..... อ.พ. 0324 ก. 2545

"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

อาจารย์ที่ปรึกษา

ดร.ภัทธชัย ลลิตโรจน์วงศ์



H001940

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2545
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ
นักศึกษา	นางสาววิศรา กิรติศิริกุล
อาจารย์ที่ปรึกษา	ดร. ภัทรชัย ลลิตโรจน์วงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

เอเจนต์เฝ้าสังเกตเป็นเอเจนต์ที่ทำหน้าที่เฝ้าสังเกตกิจกรรมและเหตุการณ์ต่าง ๆ ที่จะเกิดขึ้น หรือจากการทำงานตามที่คุณใช้กำหนดขึ้นเอง โดยจะมีการส่งผลนั้นแสดงให้คุณใช้ทราบ หรือตัดสินใจเพื่อจัดการเองอย่างอัตโนมัติ โดยปกติโปรแกรมที่เป็นประโยชน์สำหรับผู้ใช้จะต้องมีความสามารถในการรองรับการทำงานของผู้ใช้ได้เป็นอย่างดี สำหรับอินเทลลิเจนต์เอเจนต์ก็เช่นเดียวกัน กล่าวคือ เอเจนต์ที่สามารถรองรับการทำงานของผู้ใช้นั้น จะต้องมีความสามารถในการรับรู้บางสิ่งบางอย่างที่เกี่ยวข้องกับผู้ใช้และสิ่งที่คุณใช้ชื่นชอบรวมไปถึงความสามารถในด้านการตัดสินใจด้วย โดยเทคนิคที่ในโครงการพัฒนาเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจนี้ใช้เทคนิคการเปรียบเทียบแบบ string ต่อ string เพื่อตรวจสอบการเปลี่ยนแปลงของเว็บเพจนั้น

Title	Developing A Web Content Monitoring Agent
Student	Miss Waritsara Keeratisirikun
Advisor	Dr. Pattarachai Lalitrojwong
Level Of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2002

ABSTRACT

A monitoring agent is an agent monitoring activities and events that will happen or acting what a user requires. Then, the result will be presented to the user or it automatically decides what to do by itself. For a program to be useful to user, it must be able to serve you well. Therefore, a good agent should be able to react or respond to the events. To do this, it is necessary for the monitoring agent to know something about users and their preferences. The technique in developing a Web Content Monitoring Agent project is compare string by string between old source and new source of HTML. It is the one in a many technique to monitor about web page changed.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานนี้สำเร็จได้ด้วยความช่วยเหลือและการสนับสนุนจากหลายฝ่าย จึงใคร่ขอกราบขอบพระคุณ ดังต่อไปนี้

1. ดร. ภัทรชัย ลลิตโรจน์วงศ์ ซึ่งเป็นที่ปรึกษาโครงการและอาจารย์ที่ปรึกษาวิชาสัมมนา ที่กรุณาให้คำปรึกษาในหัวข้อที่ติดขัดต่าง ๆ ที่เกิดขึ้นในการพัฒนาระบบงานนี้
2. คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เปิดโอกาสให้ใช้สื่อการเรียน และสถานที่สำหรับการพัฒนาระบบงานนี้ตลอดการศึกษาในช่วงเวลาที่พัฒนาระบบงานอย่างเต็มที่
3. เพื่อนร่วมรุ่นทุกท่านที่คอยให้ความช่วยเหลือ คอยให้กำลังใจและคำแนะนำมาโดยตลอด



สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่	
1. บทนำ	1
1.1 ความเป็นมาของปัญหา	1
1.2 ประโยชน์ที่คาดว่าจะได้รับ	2
2. ทฤษฎีที่เกี่ยวข้อง	3
2.1 อินเทลลิเจนซ์เอเจนต์	3
2.2 ภาษาสำหรับอินเทลลิเจนซ์เอเจนต์	8
2.3 ตัวอย่างของ Web Content Monitoring Agent	12
2.4 การประยุกต์ใช้งานอินเทลลิเจนซ์เอเจนต์บนอินเทอร์เน็ตและเว็บ	14
2.5 เอเจนต์เฝ้าสังเกต	18
3. การออกแบบและพัฒนาระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ	27
3.1 ขอบเขตของการพัฒนาระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ	27
3.2 ขั้นตอนการทำงานของระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ	27
3.3 การเข้าถึงอินเทอร์เน็ตโดยใช้โพรโทคอล HTTP	28
3.4 การติดต่อกับอีเมลล์โดยใช้โพรโทคอล MAPI	33
4. การทำงานของโปรแกรม	40
5. บทสรุป	48
5.1 ผลการพัฒนาเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ	48
5.2 ข้อจำกัดและข้อเสนอแนะ	49

สารบัญ (ต่อ)

บรรณานุกรม
ประวัติผู้เขียน

หน้า
50
52



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่

หน้า

2.1 สรุปลักษณะเฉพาะของการเตือนแต่ละประเภท

25



สารบัญรูป

รูปที่	หน้า	
2.1	สถาปัตยกรรมโดยรวมของอินเทลลิเจนต์เอเจนต์	4
2.2	สถาปัตยกรรมโดยรวมของเอเจนต์ที่มีการเรียนรู้	7
2.3	โครงสร้างภาษาที่เป็นลำดับชั้นของ KQML	10
2.4	ตัวอย่างข้อความ KQML (a) ฟังผู้รับ และ (b) ฟังผู้ส่ง	11
2.5	สถาปัตยกรรมของ WebCrawler	13
2.6	วงรอบในการตรวจพบเหตุการณ์	19
2.7	แผนภาพแสดงสถานะ	20
2.8	ข้อความและสถานะระบบ	22
3.1	แผนภาพการทำงานของเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ	28
3.2	การทำงานของ HTTP GET function	29
3.3	คำสั่งในการสร้าง CInternetSession object	29
3.4	คำสั่งในการสร้าง CHttpFile object	30
3.5	คำสั่งในการใช้งาน Read member function	31
3.6	คำสั่งในการลบ object	31
3.7	ตัวอย่างคำสั่งในการอ่านเว็บเพจ	32
3.8	คำสั่งในการโหลด Mapi32.dll	33
3.9	คำสั่งในการประกาศ Function pointer	34
3.10	คำสั่งในการค้นหา address ของฟังก์ชันใน Mapi32.dll	35
3.11	คำสั่งในการ log on	36
3.12	การกำหนดโครงสร้างของ MapiMessage	37
3.13	คำสั่งในการตรวจสอบชื่อ	37
3.14	คำสั่งในการส่งเมล	37
3.15	คำสั่งในการลบข้อมูลออกจากบัฟเฟอร์	38
4.1	หน้าจอของโปรแกรมเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ	41

สารบัญรูป (ต่อ)

	หน้า
4.2 หน้าจอเมื่อทำการกดปุ่ม Add	41
4.3 หน้าจอกรอกตัวอย่างข้อมูลของเว็บเพจ	42
4.4 หน้าจอการกำหนดช่วงเวลาในการตรวจสอบ	42
4.5 หน้าจอเมื่อทำการเพิ่มข้อมูลเรียบร้อยแล้ว	43
4.6 หน้าจอแสดงผลให้ผู้ใช้ทราบเมื่อตรวจสอบแล้วพบการเปลี่ยนแปลง	43
4.7 หน้าจอเมื่อต้องการลบ url	44
4.8 หน้าจอเมื่อต้องการแก้ไขข้อมูลของ url	44
4.9 หน้าจอเมื่อต้องการบันทึกเอกสาร HTML นั้น	45
4.10 หน้าจอเมื่อต้องการส่งเมล	45
4.11 หน้าจอเมื่อต้องการเปิดหน้า Internet Explorer	46
4.12 หน้าจอเมื่อต้องการทำการตรวจสอบเว็บเพจนั้นด้วยตัวเอง	46
4.13 หน้าจอข้อมูลเกี่ยวกับโปรแกรม	47

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

เศรษฐกิจของโลกมีเรื่องของดิจิทัลเข้ามาครอบงำอยู่ตลอดเวลา ผู้บริโภคมีความมุ่งหวังอย่างยิ่งที่จะได้รับสินค้าและบริการที่จัดส่งให้กับเขาอย่างรวดเร็ว ดังคำนิยามที่ว่า “เพียงคลิกเดียวก็ถึงแล้ว” ขณะที่อินเทอร์เน็ตที่มีความเร็วสูงก็เป็นอีกปัจจัยหนึ่งที่ทำให้มีการติดต่ออย่างคล่องตัวมากขึ้นมาอีก ซึ่งเป็นตัวกระตุ้นสำคัญในการเปลี่ยนแปลงกระบวนการต่าง ๆ ให้ก้าวหน้าไปได้อย่างรวดเร็วยิ่งขึ้นอีก และทำให้กระบวนการเป็นไปตามเวลาที่เกิดขึ้นจริงและสอดคล้องประสานได้อย่างดีด้วย

จุดเริ่มต้นแห่งอินเทลลิเจนต์เอเจนต์ (Intelligent Agents) ถือเป็นหัวข้อใหม่ที่มีการพูดถึงกันมากในปัจจุบัน แนวทางในการพัฒนานี้จะเป็นแนวทางที่ธุรกิจได้กำหนดไว้ เพื่อวางรากฐานไว้กับการควบคุม โลจิสติกส์ ณ เวลาที่เกิดขึ้นจริงในอนาคตอันใกล้นี้ และสอดคล้องกับการจัดการเป็นอย่างดี และจะเป็นรากฐานสำคัญต่อการพัฒนาในอนาคต และแน่นอนว่าผู้ที่สามารถเปลี่ยนแปลงก่อนย่อมจะได้เปรียบ และในช่วงต่อไปองค์กรต่าง ๆ จะเริ่มต้นตัวในการนำอินเทลลิเจนต์เอเจนต์เข้ามาปรับใช้ในองค์กรของตนอย่างแน่นอน

ด้วยความก้าวหน้าทางเทคโนโลยีดิจิทัลที่ได้ยกระดับการใช้งานทางด้านอินเทอร์เน็ตและได้มีการปรับองค์กรด้านธุรกิจด้านใหม่ขึ้น รวมถึงโครงสร้างด้านอุตสาหกรรมด้วย การเกิดขึ้นของการสร้างอินเทลลิเจนต์เอเจนต์ก็เป็นการสร้างความก้าวหน้าให้มีมากขึ้นด้วย เพื่อให้เกิดการบริการอย่างมีประสิทธิภาพอย่างถ้วนหน้า และมีการใช้งานอย่างเต็มที่ด้วย

โดยปกติอินเทลลิเจนต์เอเจนต์ส่วนใหญ่จะใช้ภาษา C++ หรือภาษา Java ในการสื่อความหมาย เพื่อก่อให้เกิดการเปลี่ยนแปลงทางด้านเงื่อนไข และปรับเปลี่ยนทางด้านสิ่งแวดล้อมเพื่อแก้ไขปัญหาทางด้านธุรกิจ จากกรณีนี้เอง ทำให้อินเทลลิเจนต์เอเจนต์ได้กลายเป็นสัญลักษณ์แห่งการเปลี่ยนแปลงในช่วงเวลาปรับเปลี่ยนที่ผ่านมา ซึ่งเอเจนต์จะต้องมีการปรับปรุงด้านการบริการอยู่ตลอดเวลาด้วย

จากความสามารถในการพัฒนาเอเจนต์บนเว็บที่มีความก้าวหน้าเร็วขึ้น ดังนั้นการเปลี่ยนแปลงข้อมูลภายในเว็บเพจก็สามารถเกิดขึ้นได้ตลอดเวลา ดังนั้นเอเจนต์จะต้องมีคำสั่งในการ

แฉ่งให้ผู้ใช้ทราบเมื่อเว็บเพจเกิดการเปลี่ยนแปลงเสมอ โดยปกติเอเจนต์เฝ้าสังเกตเว็บเพจ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Monitoring Web Page Agent) จะเป็นเอเจนต์ที่ทำหน้าที่เฝ้าสังเกตข้อมูลต่าง ๆ ที่จะเกิดขึ้นหรือทำงานตามที่ผู้ใช้กำหนด โดยจะมีการส่งผลให้ผู้ใช้ทราบหรือตัดสินใจเพื่อจัดการเองอย่างอัตโนมัติ

1.2 ประโยชน์ที่คาดว่าจะได้รับ

จากการพัฒนาระบบตามโครงการที่เสนอมานี้ คาดว่าจะได้รับประโยชน์ ดังนี้

1. เข้าใจหลักการและขั้นตอนของการเฝ้าสังเกตเว็บเพจ
2. เข้าใจในเรื่องคำสั่งที่ใช้ในการติดต่อกับ Internet และในการรับและส่งเมล
3. เป็นแนวทางการพัฒนาเอเจนต์เพื่อประยุกต์ใช้กับปัญหาทางอินเทอร์เน็ต ซึ่งในปัจจุบันมีจำนวนของข้อมูลที่เพิ่มขึ้นอย่างรวดเร็ว



บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงรายละเอียดของทฤษฎีต่าง ๆ ที่ใช้ในการพัฒนาระบบตามโครงการที่เสนอ โดยมีหัวข้อที่นำเสนอต่อไปนี้

1. อินเทลลิเจนต์เอเจนต์
2. ภาษาสำหรับอินเทลลิเจนต์เอเจนต์
3. ตัวอย่างของ Web Content Monitoring Agent
4. การประยุกต์ใช้งานอินเทลลิเจนต์เอเจนต์บนอินเทอร์เน็ตและเว็บ
5. เอเจนต์เฝ้าสังเกต

2.1 อินเทลลิเจนต์เอเจนต์

เอเจนต์เป็นเสมือนตัวแทน ที่มีการทำงานอย่างเป็นอิสระในการจัดการกับงานที่เป็นประโยชน์ต่อผู้ใช้ อีกทั้งยังสามารถติดต่อกับอินเทลลิเจนต์เอเจนต์อื่น และ/หรือคนในการปฏิบัติงานเพื่อให้ทำงานสำเร็จตรงตามวัตถุประสงค์ ตัวเอเจนต์จะมีความสามารถในการรับรู้สภาพแวดล้อม และสามารถนำข้อมูลที่ได้ดังกล่าวมาใช้ในการตัดสินใจเพื่อปฏิบัติงานอย่างใดอย่างหนึ่ง โดยที่ให้มีการรบกวนผู้ใช้ให้น้อยที่สุด อีกทั้งยังสามารถเพิ่มฐานความรู้ได้ด้วยตัวเอง ซึ่งเป็นสิ่งหนึ่งที่แตกต่างระบบผู้เชี่ยวชาญที่ไม่มีความสามารถในการเพิ่มเติมฐานความรู้ และเอเจนต์ยังมีความริเริ่มในการปฏิบัติงาน (Chorafas, 1998)

ในปัจจุบันได้มีการนำอินเทลลิเจนต์เอเจนต์มาใช้หลากหลายมากขึ้น เช่น เป็นเลขานุการส่วนตัว การจัดการกับจดหมายอิเล็กทรอนิกส์ การวางแผนและควบคุมเส้นทาง การดึงข้อมูลจากอินเทอร์เน็ตและฐานข้อมูล การจัดการรายการนัดหมายการประชุม การปฏิบัติงานในสายการผลิตของโรงงานอุตสาหกรรม ธุรกิจด้านอิเล็กทรอนิกส์ การจัดการกับทรัพยากรที่มีอยู่อย่างมีประสิทธิภาพ การซื้อขายแบบออนไลน์ การตัดสินใจ การออกแบบ และงานทางด้านติดต่อสื่อสาร (Gheorghe, 1998)

2.1.1 นิยามของอินเทลลิเจนต์เอเจนต์

ในปัจจุบัน การแบ่งลักษณะของเอเจนต์ได้มีการนำ Umbrella term หรือ Meta term หรือ class เข้ามาใช้ในการแสดงความแตกต่างที่ใช้เป็นเกณฑ์ในการแบ่งลักษณะและความสามารถของเอเจนต์ โดยที่นิยามเอเจนต์ที่ใช้กันมากได้แก่ ระบบคอมพิวเตอร์ที่สามารถรับรู้สภาพแวดล้อมที่มีราคาไม่แพงเกินไป อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความซับซ้อน มีความฉลาดและทำงานได้อย่างอัตโนมัติ เพื่อบรรลุวัตถุประสงค์หรือเป้าหมายที่กำหนดไว้ (Wang *et.al.* 2000) โดยที่เอเจนต์จะต้องมีคุณสมบัติที่สำคัญดังนี้ (Murugesan. 1998)

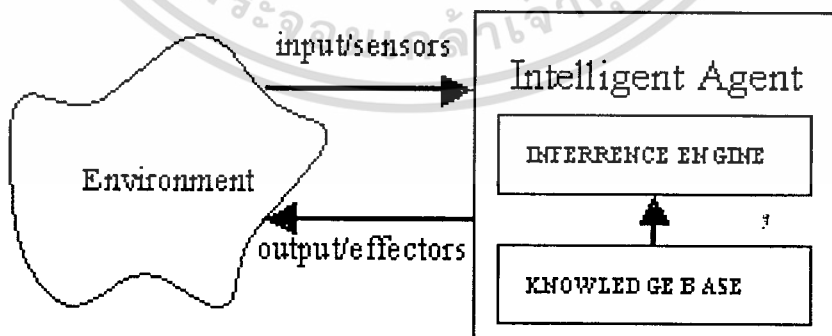
1. ภาวะอิสระ (Autonomy) เอเจนต์ต้องมีความสามารถในการปฏิบัติงานโดยที่รบกวนผู้ใช้หรือเอเจนต์อื่นให้น้อยที่สุด และยังสามารถควบคุมการกระทำและสภาวะของตนเองได้

2. ความสามารถในการเรื่องสังคม (Social Ability) เอเจนต์สามารถติดต่อกันระหว่างเอเจนต์หรือคนเมื่อเอเจนต์เห็นสมควร โดยที่เอเจนต์แต่ละตัวจะต้องสามารถสื่อสารกันได้ เพื่อแลกเปลี่ยนความรู้หรือข้อมูลซึ่งกันและกัน

3. ความสามารถในการโต้ตอบกับสภาพแวดล้อม (Responsiveness) กล่าวคือ เอเจนต์จะต้องมีความสามารถในการรับรู้สภาพแวดล้อมของตนเอง และมีความสามารถในการโต้ตอบได้ในระยะเวลาที่เหมาะสมเมื่อเกิดการเปลี่ยนแปลง

4. ความสามารถในการริเริ่ม (Proactiveness) เอเจนต์ต้องมีความสามารถในการเริ่มปฏิบัติงานเมื่อถึงเวลาที่เหมาะสมได้

พฤติกรรมของเอเจนต์จะขึ้นกับการติดต่อกันระหว่างขอบเขตโปรแกรมภายนอกของเอเจนต์และรูปแบบของขอบเขตภายใน ซึ่งประกอบด้วยฐานความรู้ (Knowledge Base) และเครื่องอนุมาน (Inference Engine) ที่มีอยู่ ดังรูปที่ 2.1 ฐานความรู้จะประกอบไปด้วยโครงสร้างข้อมูลที่แสดงด้วยเอนทิตีจากขอบเขตโปรแกรมของเอเจนต์ เช่น วัตถุ (objects) ความสัมพันธ์ระหว่างวัตถุ คลาสของวัตถุ กฎและการปฏิบัติ กระบวนการในการหาเหตุผลจากความรู้ประกอบด้วยโปรแกรมที่ใช้ในการจัดการ โครงสร้างข้อมูลในฐานความรู้ตามลำดับในการแก้ปัญหาสำหรับเอเจนต์ที่ถูกออกแบบมา (Gheorghe. 1998)



รูปที่ 2.1 สถาปัตยกรรมโดยรวมของอินเทลลิเจนต์เอเจนต์

ดังนั้นอินเทลลิเจนต์เอเจนต์จึงหมายถึงระบบคอมพิวเตอร์ที่มีความสามารถในการโต้ตอบกับสภาพแวดล้อม เช่น จากผู้ใช้ผ่านทางหน้าจอ การเรียนรู้จากเอเจนต์อื่นๆ อินเทอร์เน็ต หรือสิ่งแวดล้อมที่ซับซ้อนอื่น ให้สามารถปฏิบัติงานเพื่อบรรลุวัตถุประสงค์ที่กำหนดไว้โดยอัตโนมัติ และ

แม้ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแลกเปลี่ยนความรู้ระหว่างเอเจนต์ได้ รวมทั้งเสนอแนวคิดใหม่ ๆ ในการทำงานภายในช่วงเวลาที่เหมาะสมได้

เมื่ออินเทลลิเจนต์เอเจนต์มีการเรียนรู้จากสิ่งแวดล้อม จะส่งผลต่อการตีความสิ่งที่มองเห็น ระบุวิธีการแก้ปัญหา และการกำหนดขั้นตอน โดยจะกระทำบนสิ่งแวดล้อมนั้น เพื่อให้บรรลุเป้าหมายหรืองานสำหรับเอเจนต์ เอเจนต์สามารถติดต่อกับคนหรือเอเจนต์อื่นผ่านทางภาษาที่ใช้ในการติดต่อของเอเจนต์ แม้ว่าในบางครั้งเอเจนต์อาจจะไม่ตอบสนองคำสั่งที่ไม่รู้มาก่อน แต่จะมีความสามารถในการแปลความต้องการ โดยการสอบถามด้วยคำถามที่ชัดเจนขึ้นหรือว่าปฏิเสธความต้องการนั้นไปเลย

2.1.2 ประเภทของอินเทลลิเจนต์เอเจนต์

เอเจนต์สามารถแบ่งประเภทได้หลายแนวทาง อาจจะเป็นการแบ่งประเภทจากลักษณะเฉพาะของเอเจนต์ จากการทำงานอยู่บนพื้นฐานของเทคโนโลยี จากหน้าที่ที่เอเจนต์รับผิดชอบหรือสิ่งที่เอเจนต์ทำนั่นเอง หรือจากการรวมกันของเอเจนต์ การแบ่งประเภทของเอเจนต์ไม่ได้มีลักษณะเฉพาะ บางรูปแบบของเอเจนต์จะใช้การแบ่งโดยที่ไม่มีความสัมพันธ์กันของลักษณะเฉพาะก็ได้ (Murugesan, 1998)

(1) Static และ Mobile Agents

การแบ่งประเภทว่าเอเจนต์นี้เป็นแบบใด จะขึ้นอยู่กับความสามารถในการเปลี่ยนแปลงของเอเจนต์ เอเจนต์สามารถแบ่งประเภทตามเอเจนต์ที่ไม่เคลื่อนที่ (static หรือ stationary agents) หรือเอเจนต์ที่เคลื่อนที่ (mobile agents) เอเจนต์ที่ไม่เคลื่อนที่จะอยู่กับระบบคอมพิวเตอร์เพียงระบบเดียว ประมวลผลตามหน้าที่หรืองาน และติดต่อกับโลกภายนอกด้วยการแลกเปลี่ยนข้อความ เอเจนต์ประเภทนี้จะเหมือนกับซอฟต์แวร์แบบ conventional ส่วนเอเจนต์ที่เคลื่อนที่ได้ สามารถทำงานผ่านเครือข่ายไปยังระบบคอมพิวเตอร์อื่น ๆ โดยทำงานที่กำหนดไว้และประมวลผลข้อมูลที่ต้องการที่คอมพิวเตอร์เครื่องอื่น และส่งผลกลับมายังเครื่องต้นทางพร้อมกับข้อมูลที่ได้มาหรือข้อมูลที่ถูกระมวลผลแล้ว

(2) Interface Agents

Interface Agents จะมีส่วนช่วยเหลือผู้ใช้ที่สำคัญดังนี้

- ช่วยลดความซับซ้อนของส่วนต่อประสานและช่วยเพิ่มความสะดวกในการใช้งานตามหน้าที่ที่มีอยู่

- เตรียมส่วนต่อประสานผู้ใช้ที่ฉลาด ซึ่งสามารถรู้ได้ว่าเมื่อผู้ใช้มีปัญหาที่ยากมาก ๆ และจะคอยให้คำแนะนำหรือเสนอแนะให้กับผู้ใช้ในการแก้ปัญหา

- สังเกตและเรียนรู้ความชอบและนิสัยของผู้ใช้และพยายามดำเนินสิ่งที่ผู้ใช้ทำเป็นประจำให้สำเร็จโดยอัตโนมัติ

Interface Agents สามารถสังเกตขั้นตอนที่ผู้ใช้กระทำ เรียนรู้ความสามารถใหม่ ๆ ได้ และนำสิ่งที่ปฏิบัติให้กับผู้ใช้ได้ทราบ เตรียมความช่วยเหลือได้อย่างรวดเร็ว ปรับตัวส่วนต่อประสานให้ตรงตามความต้องการของผู้ใช้โดยอัตโนมัติ

(3) Collaborative Agents และ Multi-Agent System (MAS)

Collaborative Agents จะติดต่อและทำงานร่วมกันกับเอเจนต์อื่น และ/หรือคน ในการปฏิบัติงานทำหน้าที่เป็นตัวแทนของผู้ใช้

ในขณะที่ MAS ประกอบด้วยเอเจนต์ตั้งแต่สองตัวขึ้นไปมาติดต่อกัน ซึ่งงานบางงานนั้นเอเจนต์เพียงตัวเดียวไม่สามารถทำงานได้สำเร็จ หรือถ้าทำได้ก็มีประสิทธิภาพต่ำ ดังนั้น จึงมีการใช้กลุ่มของเอเจนต์มาแก้ปัญหาดังกล่าว โดยใช้ความร่วมมือและการทำงานร่วมกันในการดำเนินตามชุดของงานหรือตามวัตถุประสงค์ กลุ่มของเอเจนต์จะมีการแก้ปัญหา การติดต่อ และความสามารถในการทำงานร่วมกันระหว่างเอเจนต์ โดย Wooldrige (1996) ได้ให้นิยามของ MAS ไว้ว่าเป็นเครือข่ายหลวม ๆ ของผู้แก้ปัญหาที่สามารถทำงานได้เอง มาทำงานร่วมกันเพื่อแก้ปัญหาที่ไม่สามารถแก้ไขได้ด้วยตัวเอง และมีคุณสมบัติดังนี้

- เอเจนต์แต่ละตัวมีข้อมูลและความสามารถที่จำกัด
- ไม่มีส่วนควบคุมกลาง
- ข้อมูลจะถูกกระจายไปอยู่ตามส่วนต่าง ๆ
- การทำงานของเอเจนต์แต่ละตัวมีความเป็นอิสระต่อกัน

เอเจนต์ใน MAS จะมีความสามารถในการแก้ปัญหาและการติดต่อการประสานกัน และการจัดการกับเอเจนต์ตัวอื่น ปัญหาที่สำคัญ ๆ ได้แก่ ปัญหาในการสื่อสารระหว่างเอเจนต์ การทำงานร่วมกันระหว่างเอเจนต์ เป็นต้น

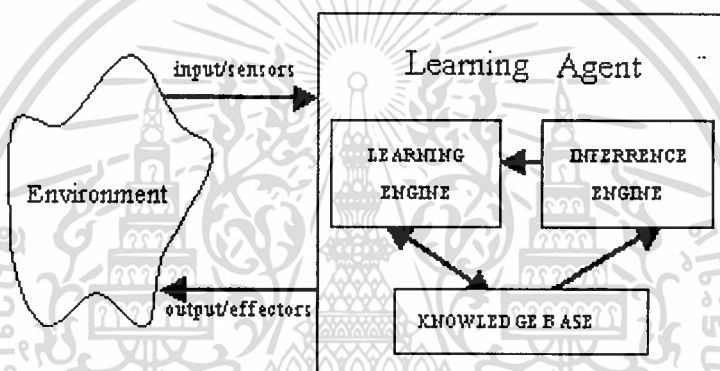
(4) Information Agents และ Internet Agents

ในโลกของเครือข่าย ข้อมูลจำนวนมากสามารถนำมาใช้ประโยชน์ได้จากแหล่งข้อมูลหลากหลายแห่งที่มาจากทั่วโลก Information Agents และ Internet Agents สามารถกรองจำนวนข้อมูลที่ที่เป็นประโยชน์บนอินเทอร์เน็ตได้ ส่งไปให้ผู้ใ้แค่เพียงข้อมูลที่ผู้ใช้สนใจเท่านั้น เอเจนต์เหล่านี้สามารถค้นหาและดึงข้อมูลตามต้องการและทำการประมวลผลข้อมูลที่มีอยู่ อีกทั้งยังช่วยเหลือในงานที่น่าเบื่อในการดึงข้อมูลที่สัมพันธ์กันจากแหล่งที่มาในเครือข่าย

(5) Learning Agents

ปัญหาต่างๆ จำนวนมากที่เกิดขึ้นในช่วงการพัฒนาและใช้อินเทลลิเจนต์เอเจนต์ในการเปลี่ยนความรู้ในฐานความรู้ หรือเรียกว่า การดึงความรู้ และการแปลงความรู้ที่ตอบกลับมาไปเปลี่ยนขอบเขตของแอฟพลิเคชัน หรือความต้องการของเอเจนต์ ซึ่งก็คือ การบำรุงรักษาความรู้ เอเจนต์ที่มีความสามารถในการดึงและบำรุงรักษาความรู้ได้ด้วยตัวเอง จะเรียกว่า เอเจนต์เรียนรู้ (Learning Agent)

นอกจากฐานความรู้และเครื่องอนุมานแล้ว เอเจนต์เรียนรู้ยังประกอบด้วยส่วนของการเรียนรู้ ซึ่งประกอบด้วย โปรแกรมที่สร้างและแก้ไข โครงสร้างข้อมูลในฐานความรู้ ดังรูปที่ 2.2



รูปที่ 2.2 สถาปัตยกรรมโดยรวมของเอเจนต์ที่มีการเรียนรู้

เอเจนต์ที่มีการเรียนรู้ สามารถเรียนรู้ได้จากแหล่งที่มาของข้อมูลข่าวสารหลาย ๆ ที่ในสิ่งแวดล้อม เอเจนต์สามารถเรียนรู้ได้จากผู้ใช้เอง หรือจากตัวเอเจนต์อื่น ซึ่งมาจากการออกคำสั่งโดยตรงของตัวเอเจนต์เอง หรือ ได้มาจากการสังเกตและเลียนแบบพฤติกรรมของเอเจนต์ เอเจนต์สามารถเรียนรู้จากที่เก็บข้อมูลข่าวสาร เช่น ฐานข้อมูล หรือสามารถเรียนรู้ได้จากประสบการณ์ของตัวเอเจนต์เอง

เอเจนต์ที่มีการเรียนรู้จะใช้เทคนิคของการเรียนรู้ในการเรียนรู้ว่าเอเจนต์จะตอบหรือไม่ได้ตอบกับสิ่งแวดล้อมภายนอกและผู้ใช้ เอเจนต์จะพยายามปรับปรุงประสิทธิภาพตามกาลเวลา และพยายามเรียนรู้สิ่งที่ผู้ใช้ชอบและสนใจ เอเจนต์สามารถเรียนรู้ได้ 4 วิธี คือ (Lange, 1998)

1. การสังเกตและเลียนแบบจากผู้ใช้ โดยสังเกตว่าผู้ใช้ทำอะไร
2. การประเมินผลจากผู้ใช้ โดยเอเจนต์สามารถเสนอข้อคิดเห็นหรือนำเสนอแนะการทำงานของผู้ใช้ แล้วเรียนรู้โดยการรับผลที่คืนกลับมาจากผู้ใช้
3. ความชัดเจนของคำสั่งจากผู้ใช้ โดยที่เอเจนต์จะรับคำสั่งที่ชัดเจนว่าเมื่อเกิดเหตุการณ์นี้

ขึ้น ต้องทำอย่างไร

เอกสารนี้เป็นทรัพย์สินทางปัญญาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การสอบถามจากเอเจนต์อื่นสำหรับข้อคิดเห็นหรือคำแนะนำต่าง ๆ

(6) Emotional Agents

เอเจนต์เหล่านี้สามารถอธิบายความรู้สึกและถ่ายทอดสถานะเหล่านี้ไปให้ผู้ใช้อื่น เอเจนต์จะมีการเตรียมสิ่งที่เรียกว่า illusion of life เพื่อสร้างความมั่นใจในการทำให้ผู้ใช้เชื่อว่าเอเจนต์เป็นโลกแห่งความจริง Emotional Agents ในบางครั้งจะเรียกว่า believable agents หรือ synthetic characters และ anthropoids ตัวอย่างของ Emotional Agents เช่น เครื่องมือ การทำงานของระบบทำงานแบบทันที (real-time) และการถ่ายข้อมูล เป็นต้น

2.2 ภาษาสำหรับอินเทลลิเจนต์เอเจนต์

ประเภทของภาษาสำหรับอินเทลลิเจนต์เอเจนต์ที่มีการกำหนดวิธีการในอินเทอร์เน็ตด้วยโครงสร้างของภาษาในยุคที่ 5 (fifth-generation languages) ซึ่งบนเว็บจะมีโพรโทคอล คือ HTTP และ HTML ที่จำเป็นต้องมีเป็นพื้นฐานอยู่แล้ว และมี Virtual Reality Modeling Language (VRML), Knowledge Query and Manipulation Language (KQML), Telescript, ActiveX และ Java ซึ่งมีการทำงานในระดับสูงขึ้นไป

ในการพัฒนาระบบเอเจนต์ จะมีภาษาอยู่ 2 ประเภท คือ Agent Language และ ACL (Agent Communication Language) (Wooldridge and Jennings. 1995)

2.2.1 ภาษาสำหรับเอเจนต์

ภาษาสำหรับเอเจนต์ (Agent Language) คือระบบที่อนุญาตให้สามารถโปรแกรมคอมพิวเตอร์ฮาร์ดแวร์หรือคอมพิวเตอร์ซอฟต์แวร์ในรูปแบบของโครงสร้างและทฤษฎีทางเอเจนต์ได้

ในอนาคตของเทคโนโลยีเอเจนต์บนเว็บได้รับการพัฒนาไปอย่างรวดเร็ว ดังนั้นในหัวข้อนี้จะขอยกตัวอย่างภาษาที่สนับสนุนการเขียนแอปพลิเคชันของเอเจนต์บนอินเทอร์เน็ตและเว็บ ดังนี้

(1) AGENT0 (Finin. 1996)

สถาบันเทคโนโลยีแห่งเมสซาชูเซตส์ ได้ออกแบบระบบ AOP (Agent-Oriented Programming) และได้สร้างตัวแปลภาษาสำหรับการเขียนโปรแกรมเรียกว่า ภาษา AGENT0 (A0) ซึ่งภาษา AGENT0 เป็น stab รุ่นแรกของภาษาโปรแกรมสำหรับใช้เป็นต้นแบบของ AOP ซึ่งใช้กันอยู่ในปัจจุบัน โดยที่ได้รับการพัฒนาอยู่ที่สแตนฟอร์ดภายใต้การควบคุมดูแลของ Yoav Shoham

(2) Telescript ของ General Magic (Watson. 1997)

Agent Language อีกประเภทหนึ่งที่ได้รับการนิยมนำมาใช้ในทางการค้าเป็นภาษา

แรก ได้แก่ ภาษา Telescript พัฒนาโดยบริษัท General Magic ภาษานี้จะเป็นแบบภาษาเชิงวัตถุ และ
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานหรือการเผยแพร่โดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นภาษาโปรแกรมแบบกระจายจะช่วยในการกระจายแอฟพลิเคชันของเครือข่ายไปยังส่วนต่าง ๆ โดยจะแบ่งลักษณะของ Telescript ออกเป็น 4 ส่วนหลัก ได้แก่

- การแสดงความหมาย (Abstraction)
- แนวคิดของเอเจนต์ (Concept of Agents)
- สถานที่ (Places)
- คำสั่งที่ต้องปฏิบัติ (Go Command)

เอเจนต์จะเคลื่อนที่ไปยังสถานที่อื่นที่จะทำการติดต่อกับเอเจนต์อื่น ๆ หรือเพื่อไปทำงานที่เป็นประโยชน์ต่อผู้ใช้โดยที่เอเจนต์เหล่านี้จะเคลื่อนที่ไปมาบนเครือข่าย ดังนั้น ภาษา Telescript จะเป็น Mobile Agent หรือเอเจนต์ที่เคลื่อนที่ได้นั่นเอง

(3) Agent Tcl (CADIP Symposium. 1999)

Agent Tcl เป็นระบบเอเจนต์ที่สามารถเคลื่อนย้ายได้ ใช้สำหรับการพัฒนาระบบเอเจนต์ที่มีความสามารถย้ายจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่งภายในระบบเครือข่ายที่ต่างชนิดกันหรือไม่เหมือนกัน วัตถุประสงค์ของภาษานี้ คือ

- จัดการเตรียมการสื่อสารระหว่างเอเจนต์
- สนับสนุนภาษาที่มีหลากหลายและกลไกในการเคลื่อนย้าย
- จัดเตรียมความพร้อมในการรักษาความปลอดภัย

อย่างไรก็ตาม การพัฒนาแอฟพลิเคชันในปัจจุบันมักไม่ครอบคลุมรูปแบบเหล่านี้ ซึ่งทำให้ประสิทธิภาพของแอฟพลิเคชันลดลงเนื่องมาจากการเขียนโปรแกรมจะยากและการตรวจหาข้อผิดพลาดที่เกิดขึ้นทำได้ยากเนื่องจากโอกาสเกิดข้อผิดพลาดมีสูง

(4) Java (Watson. 1997)

ในปัจจุบันการพัฒนาเอเจนต์บนเว็บนิยมใช้ภาษา Java เป็นส่วนใหญ่ เหตุผลส่วนหนึ่งที่มีการเลือกภาษา Java มาใช้ในการพัฒนาเอเจนต์ คือ

1. Java สามารถนำไปใช้งานร่วมกับระบบคอมพิวเตอร์ที่คนทั่ว ๆ ไปนิยมใช้ได้แทบจะทุกแพลตฟอร์ม เช่น พัฒนาระบบข้ามไมโครซิสเต็ม
2. Java เป็นภาษาที่สนับสนุนการทำงานในการออกแบบและโปรแกรมเชิงวัตถุ
3. Java มีความสามารถในการจัดการตัวเอง กล่าวคือ ผู้พัฒนาไม่ต้องกังวลเกี่ยวกับเรื่องการจัดการหน่วยความจำของโครงสร้างข้อมูลที่มีความซับซ้อน
4. Java มีความสามารถในการสนับสนุนการทำงานแบบอาร์เรย์หลายมิติ (multiple-dimensional arrays) และตารางแฮช (hash table) เป็นต้น

5. Java สนับสนุนการเขียนโปรแกรมบนอินเทอร์เน็ตและเว็บ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น และอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก Java เป็นภาษาโปรแกรมที่สามารถใช้งานกับสถาปัตยกรรมทั่วไปได้ คือ ไม่ขึ้นกับรูปแบบของระบบปฏิบัติการ รวมทั้งยังมีความสามารถในการถ่ายทอดคุณสมบัติ (Inheritance) อีกด้วย จึงเป็นภาษาหนึ่งนิยมนำมาใช้ในการพัฒนาเอเจนต์

ดังนั้นสรุปได้ว่า Agent Language คือภาษาที่ใช้ในการโปรแกรมตัวเอเจนต์ นั่นเอง

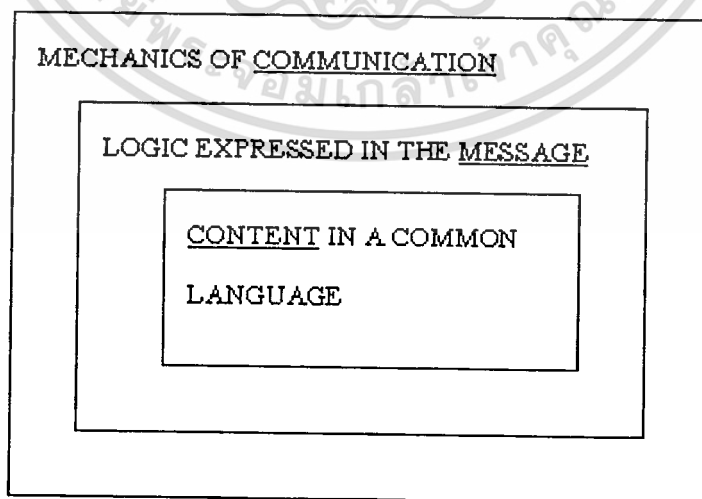
2.2.2 ACL (Agent Communication Language)

เอเจนต์แต่ละรายต้องมีการติดต่อสื่อสารระหว่างกัน ไม่ว่าจะเป็นการติดต่อกันระหว่างเอเจนต์ภายในกลุ่มที่ทำงานร่วมกัน หรือการแลกเปลี่ยนความรู้ระหว่างเอเจนต์ที่ทำงานอยู่คนละกลุ่มกัน ในปัจจุบัน ACL ที่มีการใช้งานอย่างแพร่หลายมี 2 ประเภท ได้แก่

(1) KQML (Knowledge Query and Manipulation Language) (Finin. 1996)

KQML เป็นภาษาและโพรโทคอลในระดับสูงเพื่อใช้ในการแลกเปลี่ยนข้อมูลและความรู้ โดยไม่ขึ้นกับไวยากรณ์ (Syntax) และไม่ขึ้นกับหลักการหรือความรู้เกี่ยวกับวัตถุนั้น และความสัมพันธ์ระหว่างความรู้ (Ontology) ของแต่ละเอเจนต์ รวมไปถึงไม่ขึ้นกับโพรโทคอลบนเครือข่ายที่ใช้ในการส่งข้อมูลที่อยู่ระดับต่ำกว่า เช่น TCP/IP, FTP หรือ SNMP เป็นต้น และไม่ขึ้นกับภาษาที่ใช้เขียนเอเจนต์ เช่น ภาษาปรัลล็อก เป็นต้น

KQML ใช้เป็นภาษาสำหรับแอปพลิเคชันของโปรแกรมที่มีการทำงานร่วมกันหรือติดต่อกับเอเจนต์รายอื่น โดยที่ข้อความแบบ KQML แบ่งเป็น 3 ส่วนหลัก ดังรูปที่ 2.3 ได้แก่ ส่วนของเนื้อหา (Content) ส่วนของข้อความ (Message) และส่วนของการสื่อสาร (Communication) โดยมีรายละเอียดคือ



รูปที่ 2.3 โครงสร้างภาษาที่เป็นลำดับชั้นของ KQML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนของเนื้อหา จะเป็นส่วนของข้อมูลที่เอเจนต์ต้องการจะส่ง โดยอยู่ในรูปแบบของ ภาษาและการแสดงความรู้ที่เอเจนต์นั้น ๆ เลือกใช้

2. ส่วนของข้อความ จะมีการระบุข้อความเช่น คำสั่ง คำตอบรับ คำยืนยัน การบอก ความเห็น หรือคำถาม เป็นต้น โดยระบุใน Performatives ซึ่งมีจุดประสงค์เพื่อทำการแชร์ความรู้และให้ความร่วมมือในการแก้ปัญหา โดยที่ชุดของ Performatives จะมี 3 กลุ่ม คือ

(1) Core Performatives เป็นแกนกลางที่บรรจุชุดของข้อความที่ใช้แสดงแบบง่าย ๆ ให้คนอื่นได้ทราบ

(2) Standard Performatives จะมีหน้าที่ในการสำรองชื่อไว้ให้กับเอเจนต์รายอื่นบ้าง รวมทั้งชื่อเฉพาะที่เอเจนต์ส่วนใหญ่เลือกมาใช้

(3) Extended Performatives จะบรรจุกลุ่มของเอเจนต์ที่สามารถแสดงได้และใช้เป็นส่วนขยายหรือเพิ่มเติม

โดยรวมแล้ว Performatives ใช้อธิบายการปฏิบัติงานที่เกิดขึ้น ซึ่งเอเจนต์จะมีความพยายามในการเข้าถึงข้อมูลของเอเจนต์รายอื่น และมีความพยายามในการปฏิบัติงานเพื่อให้บรรลุเป้าหมายที่วางไว้

3. ส่วนของการสื่อสาร เป็นความสามารถในการพัฒนาสถาปัตยกรรมพื้นฐานให้สามารถแชร์ความรู้ผ่านทางคลาสพิเศษของเอเจนต์ ซึ่งจะประกอบด้วยพารามิเตอร์ที่จำเป็นสำหรับการสื่อสาร เช่น Identifier ของผู้ส่ง ผู้รับ และของการสื่อสารในแต่ละครั้ง เป็นต้น

รูปที่ 2.4 เป็นตัวอย่างของข้อความแบบ KQML โดยแบ่งเป็นข้อความของผู้ส่ง และข้อความที่ตอบกลับมา

```
(ask-one
  :name-sender      mark
  :content          (QUANTITY MOBILE PHONE ?Quantity)
  :name-receiver    stock-server
  :reply-with       Mobile-stock
  :language         LPROLOG
  :ontology         SERV_TICKS)
```

(a)

```
(tell
  :name-sender      stock-server
  :content          (QUANTITY MOBILE PHONE 122)
  :name-receiver    mark
  :in-reply-to      Mobile-stock
  :language         LPROLOG
  :ontology         SERV_TICKS)
```

(b)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับข้อความของผู้ส่งแบ่งเป็นส่วนต่าง ๆ ดังนี้

- ส่วนของเนื้อหา ได้แก่ :content (QUANTITY MOBILE PHONE ?Quantity)
- ส่วนของข้อความ ได้แก่ :language LPROLOG และ :ontology SERV_TICKS และรวมไปถึง Performative ที่มีการระบุชื่อ ask-one ซึ่งเป็นการบอกว่าข้อความนี้เป็นข้อความประเภทสอบถามด้วย

- ส่วนของการสื่อสาร ได้แก่ :name-sender mark ในฝั่งผู้ส่ง และ :name-receiver stock-server และ :reply-with Mobile-stock

KQML จัดเป็น ACL ที่ได้รับความนิยมอย่างสูงในปัจจุบัน เนื่องจากมีข้อดีดังที่ได้กล่าวไปแล้วข้างต้น จึงมีการนำ KQML มาประยุกต์ใช้ในระบบเอเจนต์ต่าง ๆ เป็นจำนวนมาก

(2) AGENTBUILDER (AgentBuilding, 2001)

AGENTBUILDER เป็นเครื่องมือที่รวบรวมโปรแกรมที่ออกแบบมาเพื่อให้นักพัฒนาสามารถพัฒนาอินเทลลิเจนต์เอเจนต์ได้รวดเร็วและง่าย อีกทั้งยังช่วยลดเวลาและค่าใช้จ่ายในการพัฒนา และสามารถใช้งานอินเทลลิเจนต์เอเจนต์ได้อย่างมีประสิทธิภาพมากขึ้น AGENTBUILDER ทำให้การทำงานง่ายขึ้น ให้ผู้ใช้ติดต่อกับเอเจนต์ หรือตัวโปรแกรมที่ใช้สร้างเอเจนต์เป็นแบบกราฟฟิก โดยรายละเอียดสามารถหาอ่านจากเว็บไซต์ www.agentbuilder.co.th ได้

2.3 ตัวอย่างของ Web Content Monitoring Agent

ในปัจจุบัน Web Content Monitoring Agent มีการพัฒนาให้มีความสามารถหลากหลายมากขึ้น จึงขอยกตัวอย่างดังนี้

(1) DiffAgent (Murugesan, 1998)

DiffAgent เป็นอินเทลลิเจนต์เอเจนต์ที่พัฒนาโดยบริษัท IndustryNet ในความร่วมมือของมหาวิทยาลัย Carnegie Mellon ใช้สำหรับให้บริการด้านข่าวสาร DiffAgent จะตรวจตรา site หลาย ๆ แห่งบนอินเทอร์เน็ตและแจ้งให้กับผู้ใช้ที่เป็นของบริษัท IndustryNet ถ้ามีการตรวจสอบพบว่าการเปลี่ยนแปลงหรือแก้ไขเว็บเพจที่สัมพันธ์กัน นอกจากการแจ้งอย่างอัตโนมัติแล้ว เอเจนต์ยังเรียนรู้เกี่ยวกับสิ่งที่ผู้ใช้สนใจโดยการศึกษาว่าผู้ใช้คนนั้นอ่านเกี่ยวกับอะไร เอเจนต์จะทำการตรวจสอบในแต่ละรายการที่ผู้ใช้อ่านว่ามีคำสำคัญหรือวลีอะไรบ้าง และเอเจนต์จะส่งสิ่งเหล่านี้จากผู้ใช้กลับไปยังเอกสารที่สัมพันธ์กัน โดยจะอยู่ในรูปของคะแนน (score) โดยเอเจนต์จะใช้คะแนนและคำสำคัญในการสร้างแบบจำลองซึ่งเป็นสิ่งที่ผู้ใช้สนใจ แต่การใช้งาน DiffAgent ในปัจจุบันจะมีข้อจำกัดคือ ในการตรวจตราจะดูเฉพาะชุดของ site ที่ได้กำหนดไว้ก่อนเท่านั้น จะไม่มีความสามารถในการสำรวจแหล่งที่มาของข้อมูลใหม่ ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

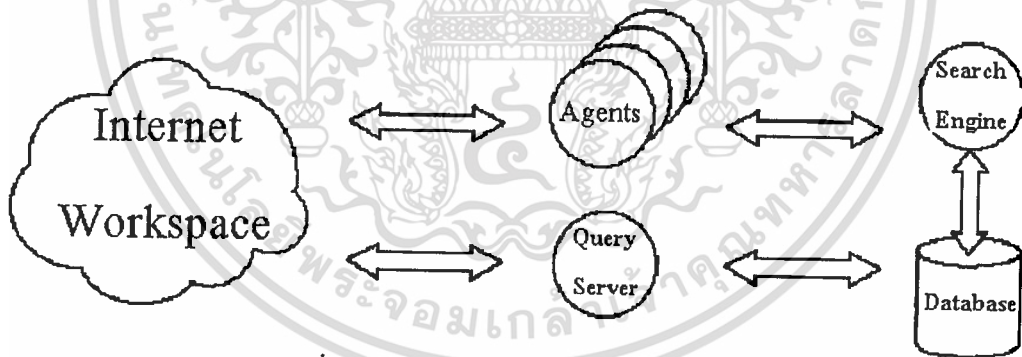
(2) WebCrawler (Chorafas. 1998)

WebCrawler ได้รับการพัฒนาโดยมหาวิทยาลัย Brian Pinkerton ในซีแอตเทิล ประเทศสหรัฐอเมริกา โดยที่ต้องการให้ WebCrawler เป็นเครื่องมือที่ใช้ค้นหาทรัพยากรที่ต้องการได้ โดยที่ WebCrawler จะจัดเตรียมเส้นทางที่เร็วเพื่อใช้ค้นหาทรัพยากร โดยมีการบำรุงรักษาครรรชนี (index) ของเว็บที่ใช้ตอบคำถามเกี่ยวกับเอกสารที่มีหัวข้อเฉพาะหรือพิเศษไว้

WebCrawler เริ่มประกาศใช้งานและเป็นที่รู้จักในปี 1994 โดยสร้างฐานข้อมูลที่บรรจุข้อมูลข่าวสารบนเว็บที่เป็นเอกสารจาก 6,000 รายของผู้ให้บริการ และ WebCrawler สามารถตอบคำถามได้มากกว่า 6,000 คำถามต่อวันและแต่ละสัปดาห์จะทำการเปลี่ยนแปลงแก้ไขข้อมูลด้วย ในปี 1995 WebCrawler ได้เข้ามาเป็นส่วนหนึ่งของ America online และทำเป็นลักษณะที่ให้ค้นทั่วไปสามารถเข้ามาใช้งานได้ฟรีผ่านทางเว็บ WebCrawler จะมีครรรชนีที่ใช้ระบุเว็บอยู่ประมาณ 100 MB ซึ่งบรรจุข้อมูลข่าวสารที่พบ โดยจะมีเอกสารที่แตกต่างกันมากกว่า 150,000 ฉบับ

WebCrawler จะทำงานตามหน้าที่ดังต่อไปนี้ คือ

- สร้างครรรชนีของเว็บ
- ทำการค้นหาตามคำสั่งที่ได้รับอย่างอัตโนมัติ



รูปที่ 2.5 สถาปัตยกรรมของ WebCrawler

สถาปัตยกรรมของ WebCrawler ประกอบไปด้วย 4 ส่วนหลัก ดังในรูปที่ 2.5 คือ

1. โปรแกรมที่ใช้ค้นหา (Search Engine) เป็นกิจกรรมโดยตรงของ WebCrawler และเป็นส่วนที่รับผิดชอบในการตัดสินใจว่าเอกสารใดที่ต้องค้นหาและเริ่มดึงข้อมูลเหล่านั้น
2. เอเจนต์ (agents) เป็นส่วนที่รับผิดชอบในการดึงเอกสารเหล่านี้มาจากเครือข่ายตามการควบคุมของ Search Engine

โปรแกรมเอเจนต์จะใช้โปรแกรมมาตรฐานคือ CERN WWW หรือเรียกว่า libWWW ในการสนับสนุนการเข้าถึงผ่านทางโพรโทคอลที่มีหลากหลายชนิด เช่น http, ftp และ gopher เป็นต้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ฐานข้อมูล (database) เป็นที่จัดเก็บเอกสารแบบถาวร และเชื่อมระหว่างเอกสารและบรรทัดที่มีทั้งหมดในรูปของ text ฐานข้อมูลจะถูกจัดเก็บลงบนดิสก์และมีการแก้ไขเปลี่ยนแปลงทุกครั้งเมื่อมีเอกสารเพิ่มเข้ามา

ในการเก็บข้อมูลในฐานข้อมูลจะเป็นข้อมูลเกี่ยวกับผู้ให้บริการ เอกสาร และการเชื่อมโยง ซึ่งเป็นตัวชี้ของเอกสาร ไปยังเอกสารอื่น แต่ Urls จะไม่นำมาถูกจัดเก็บไว้ในนี้ โดยที่จะมีแบ่งออกเป็น ส่วน ๆ ให้อยู่ในรูปของ object ซึ่งใช้อธิบายผู้ให้บริการและเอกสาร แต่ละ object จะถูกจัดเก็บในรูปของ Btree บนดิสก์ เอกสารและผู้ให้บริการจะถูกจัดเก็บบน Btree คนละต้นกัน การแบ่งข้อมูลในแบบนี้จะทำให้ WebCrawler ทำการสแกนชุดของผู้ให้บริการ ได้อย่างรวดเร็ว

4. ผู้ให้บริการตอบคำถาม (query server) เป็นการสร้างบริการด้านตอบคำถามให้กับอินเทอร์เน็ต ผู้ให้บริการตอบคำถามจะสร้างบริการค้นหาของ WebCrawler ให้สามารถใช้งานได้ผ่านทางรูปแบบการค้นหาของ HTML บนเว็บ

WebCrawler เป็นเครื่องมือที่เป็นประโยชน์สำหรับการค้นหาบนเว็บ WebCrawler จะมีการแสดงตัวเองว่าเป็น WebCrawler ในส่วนหัวของฟิลด์ใน HTTP สำหรับเอเจนต์ที่ร้องขอจากผู้ใช้เมื่อท่องเที่ยวไปในเว็บ

(3) WebWatcher (Murugesan, 1998)

ถูกพัฒนาโดยมหาวิทยาลัย Carnegie Mellon ซึ่ง WebWatcher เป็นเครื่องมือที่ช่วยเหลือผู้ใช้โดยมีการติดต่อให้คำแนะนำที่เป็นแนวทาง ผู้ใช้จะเริ่มการใช้งาน WebWatcher โดยการกำหนดรายละเอียดว่าข้อมูลอะไรที่จะค้นหา เช่น สำนักพิมพ์ โสมเพจของบุคคล หรือซอฟต์แวร์ เป็นต้น ขณะที่ผู้ใช้กำลังเรียกข้อมูลเฉพาะส่วนขึ้นมาผ่านทางเว็บเพจ WebWatcher จะกำหนด hypertext link บรรจุอยู่ด้วยในเพจนั้นด้วย

2.4 การประยุกต์ใช้งานอินเทอร์เน็ตเอเจนต์บนอินเทอร์เน็ตและเว็บ

สิ่งที่น่าสนใจในการจัดการอินเทอร์เน็ตเอเจนต์บนอินเทอร์เน็ต อินทราเน็ต และเว็บที่มีแอปพลิเคชันอยู่เป็นจำนวนมาก เช่น เลขส่วนตัว การจัดการกับจดหมายอิเล็กทรอนิกส์ การดึงและแสดงข้อมูล การวางแผนและจัดตารางการทำงาน การทำธุรกิจบนอินเทอร์เน็ต และการซื้อขายแบบออนไลน์ หรือช่วยเหลือในการตัดสินใจ

ตัวอย่างแอปพลิเคชันของอินเทอร์เน็ตเอเจนต์ที่มีการใช้งานบนอินเทอร์เน็ตและเว็บ มีดังต่อไปนี้ (Murugesan, 1998)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การสืบค้นข้อมูลโดยใช้เอเจนต์ (Agent-Based Information Retrieval)

เราสามารถใช้อเอเจนต์ช่วยในการสืบค้นข้อมูลในอินเทอร์เน็ต โดยเอเจนต์จะมีความสามารถในการทำงานต่าง ๆ ดังนี้

(1) การจัดการกับข้อมูลปริมาณมาก ๆ

สืบเนื่องมาจากจำนวนข้อมูลที่มีจำนวนมากมายมหาศาลที่มีความแตกต่างกันในเรื่องของการนำไปใช้ประโยชน์มีเพิ่มมากขึ้น ข้อมูลบนอินเทอร์เน็ตสามารถนำไปใช้เกิดประโยชน์ได้ง่ายและเสียค่าใช้จ่ายที่ต่ำ รวมไปถึงมีการใช้พลังงานน้อยที่สุด ซึ่งข้อมูลจำนวนมากจะเป็นแบบที่ต้องมีการเปลี่ยนแปลงหรือแก้ไขอยู่ตลอดเวลา

ปัญหาของการใช้อินเทอร์เน็ตและเว็บในการดึงข้อมูลที่น่าสนใจให้กับผู้ใช้ รวมไปถึง

- การรวมข้อมูลที่มีมากมายมหาศาล โดยที่มุ่งเน้นปัญหาในด้านความยากในการดึงข้อมูลให้ตรงกับปัญหาหรือความต้องการ ระยะเวลาที่ใช้ในการค้นหาข้อมูลและการแสดงข้อมูลที่มากเกินไปเกินความต้องการ

- ช่วงเวลาที่ข้อมูลบนอินเทอร์เน็ตมีการเปลี่ยนแปลง ข้อมูลที่ต้องการอาจจะพบหรือว่าสูญหายไปแล้ว

- ความแตกต่างกันของข้อมูลและบริการที่ให้กับข้อมูลแต่ละประเภท

การจัดการกับปัญหาเหล่านี้ จำเป็นที่จะต้องมีเครื่องมือที่ช่วยในการค้นหา ดึง กรอง และเลือกข้อมูล ณ ปัจจุบัน Intelligent Software Agent จะมีแนวทางในการแก้ไขกับข้อมูลที่แสดงมากเกินไปขอบเขตหรือความต้องการ และปัญหาในการดึงข้อมูล (Information Retrieval หรือ IR) โดยที่สามารถลดระดับความซับซ้อนของการดึงข้อมูลและการจัดการได้โดยใช้ความฉลาดในการค้นหาหนทางที่เป็นไปได้

(2) อินเทอร์เน็ตเฟซ

อินเทอร์เน็ตเอเจนต์จะติดต่อกับระบบการดึงข้อมูลเพื่อจัดหาหนทางให้สามารถใช้งานได้ง่าย สร้างความเป็นส่วนตัว หรือคัดแปลงหรือปรับอินเทอร์เน็ตเฟซได้

(3) การค้นหาข้อมูลผ่านตัวกลาง (Mediated Searches and Information Brokers)

ปัจจุบันเครื่องมือจำนวนมากที่ใช้ในการค้นหามีความสามารถในการค้นหาข้อมูลที่เป็นประโยชน์บนอินเทอร์เน็ตและเว็บได้ อย่างไรก็ตาม เครื่องมือแต่ละแบบก็มีโครงสร้างในการค้นหา การติดต่อ และวิธีการใช้งานที่แตกต่างกัน

การเลือกข้อมูลจากแหล่งข้อมูลบนอินเทอร์เน็ตและเว็บ ทำให้เกิดปัญหามากมาย ซึ่งรวมไปถึง

- การค้นหาแหล่งข้อมูลที่เหมาะสมสัมพันธ์กับคำถามและเหมาะสมกับเครื่องมือที่ใช้ค้นหา

- กำหนดหรือวางคำถามให้อยู่ในรูปแบบที่แหล่งข้อมูลเข้าใจ
- การดึงข้อมูล
- การเปรียบเทียบข้อมูลที่ได้รับมาจากหลาย ๆ แหล่งและทำให้ปะติดปะต่อกันและรวบรัด

(4) เอเจนต์กรองสารสนเทศ (Information Filtering Agents)

เอเจนต์จะทำการค้นหาข้อมูลในส่วนที่น่าสนใจจากแหล่งข้อมูลที่แตกต่างกันให้กับผู้ใช้ เอเจนต์ที่กรองข้อมูลมีความสามารถดังนี้

- รวบรวมบทความใหม่ ๆ เกี่ยวกับหัวข้อที่เลือกไว้ จากแหล่งข้อมูลต่าง ๆ เช่น จากเว็บเพจ หรือจากแหล่งข่าว เป็นต้น

- กรองข้อมูลโดยขึ้นอยู่กับความชอบของผู้ใช้
- กรองข้อมูลที่เป็นปัจจุบันให้กับผู้ใช้

ตัวอย่างของเอเจนต์ที่กรองข้อมูล เช่น NewsHound หรือ ZDNet Personal Views

(5) การจัดประเภทและการนำเสนอ

เอเจนต์ที่ดีควรมีความสามารถในการวิเคราะห์ จัดกลุ่มและนำเสนอข้อมูลที่ได้รับมา เพื่อเพิ่มความสะดวกให้กับผู้ใช้

2.4.2 การใช้เอเจนต์กับอีคอมเมิร์ซ (Agents in Electronic Commerce)

วัตถุประสงค์หลักของการใช้เอเจนต์กับอีคอมเมิร์ซ คือช่วยในเรื่องซื้อขายผ่านเอเจนต์ โดยเอเจนต์จะเป็นช่วยผู้ใช้ในเรื่องต่าง ๆ ดังนี้

- ช่วยผู้ใช้ตัดสินใจว่าจะซื้อผลิตภัณฑ์ใด เช่น การแสดงรายการของผลิตภัณฑ์ที่เป็นไปได้ทั้งหมด

- หาลักษณะ รายละเอียดและพิจารณาผลิตภัณฑ์ให้
- เพิ่มเติมข้อเสนอแนะ หรือคำแนะนำ
- เปรียบเทียบร้านค้าเพื่อที่จะหาสินค้าราคาที่ถูกที่สุดสำหรับผลิตภัณฑ์ที่ต้องการ
- ตรวจสอบ ดูแล และแจ้งให้ผู้ใช้ทราบเกี่ยวกับสินค้า ราคาพิเศษ และส่วนลด

เอเจนต์จะตอบสนองเรื่องสินค้า เสนอราคาสินค้าที่ต้องการ ติราคาจากราคาสินค้าที่เสนอมา ส่งมอบใบสั่งซื้อ ได้ใบรับสินค้า และชำระเงินแทนผู้ใช้

2.4.3 การใช้เอเจนต์กับงานบริการต่างๆ (Agent Service)

ตัวอย่างของ Agent Service มีดังนี้

- Announcement Agents เตือนให้ผู้ใช้ถึงโอกาสสำคัญ ๆ เช่น วันเกิด วันครบรอบ วันวาเลนไทน์ และสิ่งที่ชอบ ซึ่งเป็นความต้องการส่วนตัวเฉพาะบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Book Agents แจกให้ผู้ใช้ทราบเกี่ยวกับหนังสือใหม่ที่น่าออกมาจำหน่าย โดยที่ต้องเข้าหรือเหมาะกับสิ่งที่ผู้ใช้แต่ละคนสนใจอ่าน

- Business Information Monitoring Agents ดูแลการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตซึ่งต้องสัมพันธ์กับบริการ สินค้า โรงงานอุตสาหกรรม บริษัท และรายงานที่พบ

- Classified Agents ค้นหาฐานข้อมูลเกี่ยวกับสิ่งของที่ผู้ใช้ต้องการ และแจ้งให้ผู้ใช้ทราบผ่านทางจดหมายอิเล็กทรอนิกส์ ตัวอย่างของ Classified Agents เช่น AdHound ซึ่งให้ผู้ใช้ระบุประเภทของสิ่งที่ผู้ใช้ต้องการและทำการค้นหาจากฐานข้อมูลอย่างเหมาะสม และส่งจดหมายอิเล็กทรอนิกส์มาให้โดยระบุตำแหน่งที่จำแนกไว้แล้ว

- Job Agents ให้บริการในด้านการค้นหาลูกจ้าง ซึ่งเหมาะสมกับตำแหน่งงานที่นายจ้างต้องการ

- Direct Mail Agents จดหมายอิเล็กทรอนิกส์ที่น่าสนใจและเข้ากับกิจกรรม ประวัติด่วนตัว การดำเนินชีวิต และสิ่งที่ผู้ใช้ชอบ

- Financial Service Agents ส่งจดหมายอิเล็กทรอนิกส์เกี่ยวกับราคา ข่าวทางการเงิน และเงินทุนหมุนเวียน สำหรับผู้ใช้ที่ต้องการความปลอดภัยในการลงทุน

- Food and Wine Agents จำสถานะของผู้ใช้แต่ละคนว่าการซื้อขายที่ผ่านมาเป็นอย่างไร และสังเกตนิสัยของผู้ใช้ รวมทั้งทำบัญชีรายการและสินค้าที่มีด้วย

- Entertainment Agents ค้นหาสิ่งที่ผู้ใช้สนใจ เช่น อัลบั้มเพลง หรือภาพยนตร์ที่ชอบ เป็นต้น โดยขึ้นอยู่กับสถานการณ์ด้วย

2.4.4 การใช้เอเจนต์กับการทำธุรกรรมต่าง ๆ (Agent for Business Operations)

อินเทอร์เน็ตเอเจนต์จะค้นหาแอปพลิเคชันในอินเทอร์เน็ตภายในองค์กรและรูปแบบทางธุรกิจ โดยพิจารณาถึง

- Collaborative customization agents อินเทอร์เน็ตเอเจนต์จะต้องมีการทำงานอัตโนมัติในการทำธุรกรรม

- Process automation agents อินเทอร์เน็ตเอเจนต์จะต้องปฏิบัติงานทางอุตสาหกรรมอย่างอัตโนมัติ

- Database agents อินเทอร์เน็ตเอเจนต์จะจัดหาและเตรียมบริการต่าง ๆ เพื่อสนับสนุนการทำงานของผู้ใช้สำหรับฐานข้อมูลของธุรกิจ

- Resource brooking agents เอเจนต์จะแสดงการจัดสรรทรัพยากรที่มีอยู่ในสถาปัตยกรรมแบบไคลเอนต์เซิร์ฟเวอร์

2.4.5 การใช้เอเจนต์เพื่อความบันเทิง

ซอฟต์แวร์เอเจนต์ที่สามารถทำงานได้อย่างอิสระ จะมีความสามารถที่แฝงไว้ เช่น ส่วนของการให้ความเพลิดเพลินและการคุยกันแบบออนไลน์ เอเจนต์เพื่อความบันเทิง สามารถอธิบายให้ทราบถึงความรู้สึกและความคิดของสถานะภายในของเอเจนต์ให้ผู้ใช้ทราบได้

จะเห็นได้ว่า การนำเทคโนโลยีของเอเจนต์เข้ามาใช้ในการพัฒนาแอปพลิเคชัน สิ่งที่ต้องพิจารณาถึงหรือสิ่งที่เอเจนต์จะต้องมี คือความถูกต้อง ได้ข้อมูลที่น่าเชื่อถือ ประสิทธิภาพการทำงานแบบตอบสนองทันทีที่ต้องการ มีระบบป้องกันความปลอดภัย และมีความรับผิดชอบซึ่งเป็นความสามารถอีกข้อหนึ่งที่เอเจนต์ต้องมีด้วย

2.5 เอเจนต์เฝ้าสังเกต

อินเทลลิเจนต์เอเจนต์ส่วนใหญ่จะมีความสามารถในการบอกความรู้สึกของตัวเองให้กับผู้ใช้หรือเอเจนต์อื่นทราบได้ จากการที่มนุษย์อาศัยอยู่บนโลก เมื่อเกิดเหตุการณ์ใด ๆ เกิดขึ้นมนุษย์จะมีปฏิกิริยาตอบสนองต่อเหตุการณ์เหล่านั้นได้ โดยเอเจนต์ที่สามารถนำมาใช้ให้เกิดประโยชน์ได้ควรมีความสามารถในการตอบสนองหรือโต้ตอบกับเหตุการณ์ต่าง ๆ ที่เกิดขึ้นได้

โดยปกติโปรแกรมที่เป็นประโยชน์สำหรับผู้ใช้จะต้องมีความสามารถในการรองรับการทำงานของผู้ใช้ได้เป็นอย่างดี สำหรับอินเทลลิเจนต์เอเจนต์ก็เช่นเดียวกัน กล่าวคือ เอเจนต์ที่สามารถรองรับการทำงานของผู้ใช้นั้น จะต้องมีความสามารถในการรับรู้บางสิ่งบางอย่างที่เกี่ยวข้องกับผู้ใช้และสิ่งที่ผู้ใช้ชื่นชอบรวมไปถึงความสามารถในด้านการตัดสินใจด้วย

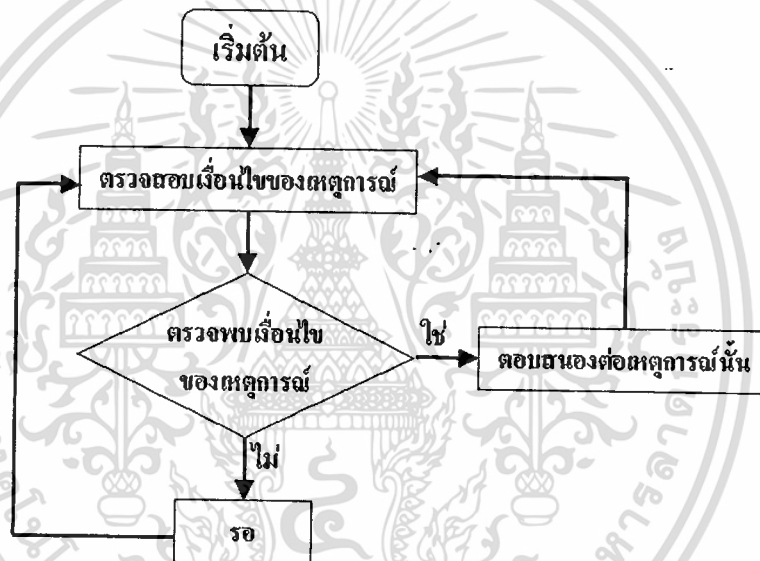
เอเจนต์เฝ้าสังเกตเป็นเอเจนต์ที่ทำหน้าที่เฝ้าสังเกตกิจกรรม (Activity) และเหตุการณ์ (Events) ต่าง ๆ ที่จะเกิดขึ้น หรือจากการทำงานตามที่ใช้กำหนดขึ้นเอง โดยจะมีการส่งผลนั้นแสดงให้ผู้ใช้ทราบ หรือตัดสินใจเพื่อจัดการเองอย่างอัตโนมัติ ลักษณะการทำงานของเอเจนต์เฝ้าสังเกตอาจจะเป็นแบบเอเจนต์ที่เคลื่อนที่ได้ เอเจนต์สามารถทำงานผ่านเครือข่ายไปยังระบบคอมพิวเตอร์อื่นได้

โดยในหัวข้อนี้จะกล่าวถึง เหตุการณ์ซึ่งถือเป็นส่วนที่ทำให้เกิดการแจ้งให้ทราบของเอเจนต์ เมื่อเหตุการณ์นั้นถูกระตุ้น คำสั่งที่แสดงอาการตอบจะสามารถตรวจสอบจากเงื่อนไขของการเตือน เมื่อเงื่อนไขของการเตือนเกิดขึ้น เอเจนต์จำเป็นจะต้องมีการเตรียมการสื่อสารหรือปฏิบัติงานไว้

2.6.1 เหตุการณ์

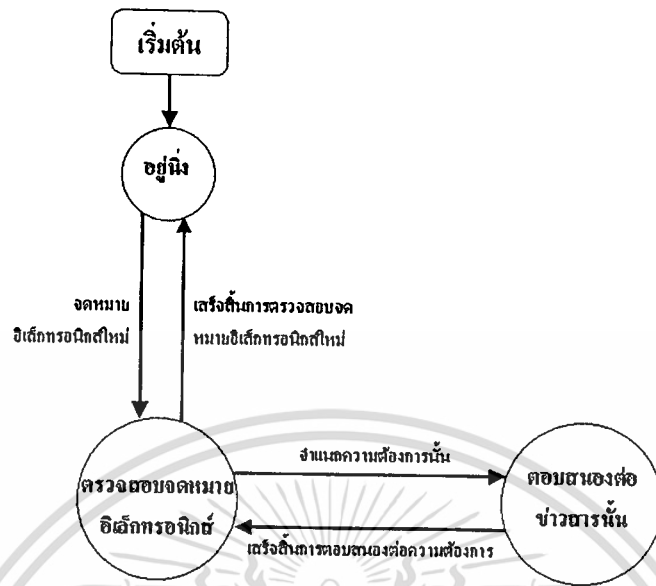
เอเจนต์จะใช้เหตุการณ์ (Event) เป็นตัวกระตุ้นให้เกิดการกระทำ (action) การกระทำของเอเจนต์จะทำในรูปของการโต้ตอบต่อเหตุการณ์ที่เกิดขึ้นได้อย่างชาญฉลาด เช่น การส่งเสียงเตือนให้ผู้ใช้ทราบ เมื่อมีการเปิดเครื่องคอมพิวเตอร์

ตัวอย่างของเหตุการณ์ เช่น เมื่อเกิดการเปลี่ยนแปลงข้อมูลบนเว็บเพจ เอเจนต์สามารถตรวจพบโดยการเปรียบเทียบได้ว่า ขณะนี้หน้าของเว็บเพจที่ผู้ใช้ให้ความสนใจได้มีการเปลี่ยนแปลงไปจากเดิมแล้ว ทำให้เกิดการตอบสนองต่อเหตุการณ์ที่เกิดขึ้น โดยลักษณะของการเฝ้าสังเกตจะเป็นวงรอบคือ check-sleep-check-sleep ไปเรื่อย ๆ ดังรูปที่ 2.6 (Pallmann, 1999)



รูปที่ 2.6 วงรอบในการตรวจพบเหตุการณ์

ในบางครั้งเราอาจจะมองภาพการทำงานของเอเจนต์เป็นการทำงานในลักษณะของ State Machine เมื่อเหตุการณ์นั้นเกิดขึ้นจะกระตุ้นให้เกิดการเปลี่ยนแปลงสถานะ รูปที่ 2.7 แสดงแผนภาพแบบง่าย ๆ สำหรับเอเจนต์เฝ้าสังเกตในการติดตามเฝ้าดูเมื่อมีจดหมายอิเล็กทรอนิกส์ใหม่เข้ามา และสามารถโต้ตอบได้อย่างอัตโนมัติ ดังนั้นเมื่อเอเจนต์ตรวจสอบและพบว่าเป็นข้อความที่ต้องการตอบกลับ เอเจนต์ก็จะทำการแสดงอาการโต้ตอบกลับไปยังคนที่ส่งข้อความนั้น



รูปที่ 2.7 แผนภาพแสดงสถานะ

จะเห็นได้ว่าเหตุการณ์ถือเป็นส่วนสำคัญสำหรับการออกแบบตัวเอเจนต์ โดยเหตุการณ์ของเอเจนต์มีหลายประเภท อาทิเช่น (Pallmann. 1999)

(1) การกระทำของผู้ใช้ (User Action)

การกระทำของผู้ใช้รวมถึงการใช้คีย์บอร์ดหรือเมาส์ด้วย ถือเป็นเหตุการณ์ง่ายๆ ที่เกิดขึ้นเนื่องจากว่าเหตุการณ์เหล่านี้สามารถโต้ตอบได้ เพราะเป็นกลไกมาตรฐานที่โปรแกรมวินโดวส์เข้ามาจัดการกับสิ่งเหล่านี้อยู่แล้ว

(2) Activity and Inactivity

โดยปกติผู้ใช้จะทำงานบนคอมพิวเตอร์ผ่านทาง การใช้คีย์บอร์ดหรือเมาส์ โดยเหตุการณ์ที่เกิดขึ้นสามารถแบ่งเป็น เหตุการณ์ที่เกิดจากการกระทำของผู้ใช้ (User action events) กับเหตุการณ์ที่เกิดจากกิจกรรม (Activity events) โดยจะแตกต่างกัน กล่าวคือ เหตุการณ์ที่เกิดจากการกระทำของผู้ใช้ หมายถึง การกระทำที่เกี่ยวข้องกับการป้อนข้อมูลทางคีย์บอร์ดหรือเมาส์ เช่น การคลิกปุ่ม ส่วนเหตุการณ์ที่เกิดจากกิจกรรม จะหมายถึง การบันทึกว่าคีย์บอร์ดหรือเมาส์กำลังถูกใช้งานเท่านั้น โดยจะไม่สนใจว่าเป็นอักษรใดบนคีย์บอร์ดที่ถูกกดหรือไม่สนใจว่าขณะนี้เมาส์กำลังทำอะไรอยู่ โดยที่เหตุการณ์ที่เกิดจากกิจกรรมจะเป็นการบันทึกการใช้งานของผู้ใช้ ณ ปัจจุบันเท่านั้น

สิ่งที่ทำหน้าที่ตรงกันข้ามกับเหตุการณ์ที่เกิดจากกิจกรรม คือ เหตุการณ์ที่อยู่นิ่ง (Inactivity events) เมื่อเวลาล่วงเลยผ่านไปโดยปราศจากกิจกรรมใด ๆ จากผู้ใช้ เราสามารถจับเวลา (timer) เพื่อใช้วัด Inactivity ได้ การจับเวลาจะเริ่มด้วยค่าระยะเวลาตามที่กำหนดไว้ เช่น 15 นาที ถ้ามีกิจกรรมการนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรรมการใช้เมาส์หรือคีย์บอร์ดเกิดขึ้นก่อนที่เวลาจะผ่านไปจนครบระยะเวลา การจับเวลาจะเริ่มต้นใหม่อีกครั้งด้วยค่าระยะเวลาเดิม ถ้าเมื่อใดก็ตามที่การจับเวลาวนครบกลับมาที่เดิมซึ่งในที่นี้คือ ครอบรอบ 1 ระยะเวลา เนื่องจากไม่มีการกระทำกับเมาส์หรือคีย์บอร์ดนานเกินไป ซึ่งหมายความว่าเหตุการณ์ที่อยู่เบื้องหลังได้เกิดขึ้นแล้ว กล่าวโดยสรุป เหตุการณ์ที่เกิดจากกิจกรรมจะสามารถนำมาใช้ในการปฏิบัติงานที่จะต้องมีการถูกระงับจากผู้ใช้งานเพียงอย่างเดียว

ในขณะที่เหตุการณ์ที่อยู่เบื้องหลังจะใช้สำหรับงานที่มีการควบคุมในช่วงที่ผู้ใช้ไม่ได้ใช้งาน โดยอาจจะหมายถึงรวมถึงเงื่อนไขที่ปฏิบัติงานเกี่ยวกับการบำรุงรักษาดีสก์ หรือการกู้คืน เป็นต้น

(3) วันที่และเวลา (Date and Time)

เป็นการรอกอยช่วงเวลาไม่ว่าจะเป็นวันและเวลาหรือเวลาตามช่วงที่กำหนดไว้เป็นส่วนที่ต้องทำงานร่วมกับของเงื่อนไข สามารถแบ่งได้เป็น 2 แบบ คือ

- Midnight Rollover เป็น โปรแกรมกำหนดค่าเริ่มต้นเมื่อวันที่ของระบบขึ้นวันใหม่ เช่น ถ้าต้องการปรับปรุงข้อมูลข่าวสารให้ทันสมัยหลังจากเที่ยงคืนไปแล้ว โดยตัวแปรที่เก็บจะเก็บเป็นแบบรายวัน เช่น ตัวแปรที่แสดงรายงานสรุปของวันนี้เมื่อถูกพิมพ์ออกมาแล้ว หลังจากเที่ยงคืนจะต้องมีการเซตกลับไปยังค่าตั้งต้นที่ได้กำหนดไว้

- End of Date, Week, Month or Year เมื่อถึงจุดสิ้นสุดในปฏิทินก็สามารถเริ่มต้นการปฏิบัติงานตามที่กำหนด เช่น เงื่อนไขที่มีความสามารถในการรักษาความปลอดภัยอาจจะต้องการยกเลิกประตูสำหรับสื่อสารเมื่อหมดเวลาทำการในแต่ละวัน เงื่อนไขที่บำรุงรักษาอาจจะต้องการส่งจดหมายอิเล็กทรอนิกส์ไปยังผู้จัดการหน่วยเพื่อเก็บข้อมูลทุกวันศุกร์ และเงื่อนไขที่จัดการทางการเงินจำเป็นต้องแสดงข้อมูลข่าวสารในช่วงสิ้นเดือนหรือมีการเก็บข้อมูลข่าวสารที่มีในทุก ๆ สิ้นปี เป็นต้น

(4) อีเมลใหม่ (New e-mail)

เหตุการณ์ คือเมื่อมีจดหมายอิเล็กทรอนิกส์ใหม่เข้ามา ผู้ใช้สามารถรับรู้ได้จากการแจ้งของระบบจดหมายอิเล็กทรอนิกส์ของผู้ใช้เองที่ตอบสนองต่อเหตุการณ์ประเภทนั้น แอปพลิเคชันที่เกี่ยวข้องกับจดหมายอิเล็กทรอนิกส์จะรวมถึงเงื่อนไขที่สามารถเริ่มทำงานเมื่อมีข้อความใหม่เข้ามาซึ่งตรงกับเงื่อนไขที่มี เช่น เงื่อนไขที่จะต้องเริ่มทำงานได้ตอบเมื่อมีจดหมายอิเล็กทรอนิกส์ทั่วไปเข้ามาใหม่ จดหมายอิเล็กทรอนิกส์จากเพื่อนร่วมงาน หรือจดหมายอิเล็กทรอนิกส์ที่มีข้อความหรือมีไฟล์ข้อมูลแนบมาด้วย เป็นต้น

(5) การเปลี่ยนแปลงของสารบบแฟ้ม (File directory changes)

เงื่อนไขที่กู้คืนอาจจะสำเนาไฟล์ทันทีที่ไฟล์เหล่านั้นมีสารบบที่แน่นอนแล้ว และการทำงานต้องแน่ใจว่าไฟล์นั้น ไม่มีใครใช้อยู่ ก่อนที่จะมีการทำงานกับไฟล์เหล่านั้น

เอกล่าวนั้นเป็นอีกสำเนาหนึ่งซึ่งไม่มีใครใช้อยู่ ในกรณีที่ผู้ใช้กดให้หน้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(6) การสื่อสารที่ซีพี/ไอพี (TCP/IP communication)

ถ้าเอเจนต์มีการใช้โพรโทคอล TCP/IP ในการสื่อสารกับระบบเอเจนต์อื่น คำสั่งที่ได้รับจากเครื่องอื่นจะถือว่าเป็นเหตุการณ์ โดยมีหลักการการทำงานเหมือนกับเว็บเซิร์ฟเวอร์ดังนี้ เมื่อมีความต้องการเข้ามาเว็บเซิร์ฟเวอร์ซึ่งเปรียบเสมือนกับการรอฟังการร้องขออยู่ ก็จะมีการโต้ตอบโดยการส่งเว็บเพจตอบกลับมา

(7) สภาวะของระบบ (System conditions)

เอเจนต์สามารถตรวจสอบสภาวะของระบบที่แตกต่างกัน โดยที่ข้อความของ Windows จะถูกส่งโต้ตอบกับสภาวะของระบบที่เกิดขึ้น ตัวอย่างของข้อความเป็นไปดังรูปที่ 2.8

Message ID	System Condition
WM_ENDSESSION	The windows session is terminating.
WM_POWERBROADCAST	Power management conditions, such as low battery power or suspended system, are occurring.
WM_QUERYENDSESSION	Windows plans to terminate the Windows sessions; Applications can acknowledge or vote the idea.
WM_TIMECHANGE	The system time has changed.
WM_USERCHANGED	The user has logged on or off.

รูปที่ 2.8 ข้อความและสภาวะระบบ

(8) การเปลี่ยนแปลงของข้อมูลในเว็บเพจ (Changes to data in Web page)

การเปลี่ยนแปลงข้อมูลภายในเว็บเพจก็สามารถเป็นเหตุการณ์ได้ เอเจนต์จะต้องมีคำสั่งในการแจ้งให้ผู้ใช้ทราบเมื่อเว็บเพจเกิดการเปลี่ยนแปลงหรือเฝ้าสังเกตข้อมูลบนเว็บเพจเมื่อเกิดการเปลี่ยนแปลง

การเฝ้าสังเกตการเปลี่ยนแปลงข้อมูลบนเว็บเพจ เอเจนต์จะต้องสำรวจเว็บเพจและทำความเข้าใจถึงการเปลี่ยนแปลงบนเว็บเพจ โดยจะมี 2 วิธี คือ

- ในส่วนของ HTTP Header จะมีฟิลด์ที่เก็บวันที่ไว้ ถ้าส่วนของวันที่เปลี่ยนแปลงไป จะหมายความว่า เหตุการณ์การเปลี่ยนแปลงของเว็บเพจนั้นได้เกิดขึ้นแล้ว

- เอเจนต์สามารถดึงเพจที่สมบูรณ์มาและเปรียบเทียบกับเพจก่อนหน้าที่เก็บไว้ เมื่อข้อมูลภายในเปลี่ยนแปลงไป ก็หมายถึงเพจนั้นมีการเปลี่ยนแปลงไปแล้ว ซึ่งหมายความว่าเหตุการณ์ที่เว็บเพจเปลี่ยนแปลงไปนั้นเกิดขึ้นแล้ว

2.6.2 การเตือน

ผู้ใช้สามารถมอบหมายให้เอเจนต์เริ่มทำการเตือนล่วงหน้าหรือเริ่มปฏิบัติงาน โดยกำหนดตัวแปรเพื่อใช้ควบคุมพฤติกรรมของเอเจนต์ เช่น การเตือนเนื่องในโอกาสสำคัญ หรือการสอบถามผู้ใช้แล้วมีการเริ่มการปฏิบัติงานบางอย่าง

โปรแกรมที่เป็นประโยชน์กับผู้ใช้จะต้องมีการรองรับการทำงานของผู้ใช้เป็นอย่างดี สำหรับอินเทลลิเจนต์เอเจนต์ก็เช่นเดียวกัน กล่าวคือ เอเจนต์ที่รองรับการทำงานของผู้ใช้นั้นจะต้องมีความสามารถในการรับรู้บางสิ่งบางอย่างเกี่ยวกับผู้ใช้และสิ่งที่ผู้ใช้ชื่นชอบ รวมไปถึงความสามารถในด้านการตัดสินใจด้วย เช่น เอเจนต์จะมีการเตือนให้ผู้ใช้ทราบเมื่อมีการเปลี่ยนแปลงของราคาสินค้าคงคลัง ซึ่งเอเจนต์จำเป็นต้องรู้เมื่อผู้ใช้ต้องการจะดูรายละเอียด หรือจะต้องทำอะไรต่อไปกับข้อมูลที่ได้มา

การแสดงการเตือนของเอเจนต์มักแสดงในรูปของความคิดเห็นหรือความรู้สึกในรูปของประโยค เช่น “Alert me when it’s 5:30 P.M.” หรือ “Find all Internet chat groups that discuss volleyball” เป็นต้น

เมื่อมีการแสดงการเตือนให้กับผู้ใช้ มันจะเป็นการดีที่สุดถ้ามีการใช้วิธีการตามากที่สุดกับผู้ใช้และกับกิจกรรมที่เอเจนต์จะต้องปฏิบัติ โดยการออกแบบแล้วถ้าเป็นไปได้ผู้ออกแบบควรจะทำออกแบบโดยหลีกเลี่ยงการใช้ถ้อยคำที่กำกวม ดังนั้นจะเห็นได้ว่าการเตือนถือเป็นส่วนสำคัญสำหรับการออกแบบตัวเอเจนต์เช่นเดียวกับเหตุการณ์ เพราะว่าการเตือนจะเป็นส่วนตอบรับจากผู้ใช้ที่มีพฤติกรรมของตัวเอเจนต์และความสามารถในการพิจารณาของตัวเอเจนต์เอง โดยการเตือนของเอเจนต์มีหลายประเภท ดังนี้ (Pallmann, 1999)

(1) Threshold Alert

เป็นการกำหนดค่าเริ่มต้นในการติดตามการเปลี่ยนแปลงของข้อมูล ถ้ามีการเคลื่อนไหวเกินกว่าหรือลดลงจนถึงค่าเริ่มต้น เอเจนต์จะต้องมีการเตรียมการปฏิบัติงานตามที่ได้กำหนดไว้ล่วงหน้า โดยปกติ Threshold Alert มักจะใช้ข้อมูล 2 แบบคือ

- numerical data เช่น ราคาหรือการนับจำนวน
- enumerated data เช่น การแบ่งชั้นหรือระดับ

Threshold Alert มักจะไม่ใช้กับข้อมูลพวกชื่อหรือสี จนกว่าจะมีการแปลงข้อมูลเหล่านี้ไปอยู่ในรูปแบบของ enumerated data เสียก่อน ตัวอย่างเช่น 1 = สีดำ 2 = สีขาว 3 = สีน้ำเงิน เป็นต้น

(2) Value Alert

จากที่ Threshold Alert จะใช้งานกับค่าที่เป็นตัวเลข แต่ในบางครั้งเราอาจจะมีความจำเป็นที่จะต้องใช้งานกับข้อมูลที่เป็นตัวอักษร อย่างเช่น ชื่อ ซึ่ง Value Alert จะแสดงฟิลด์ที่มีการกำหนดค่า

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนสิทธิ์หรือการสงวนสิทธิ์ในเนื้อหา โดยผู้จัดทำเอกสารขอสงวนสิทธิ์ในเนื้อหา
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น “Alert me when the country name is Thailand” โดยการกำหนดค่าในฟิลด์ของ Value Alert สามารถทำได้ง่าย โดยที่เมื่อเอเจนต์มีการกำหนดค่าของข้อมูลแล้ว มันจะมีการเปรียบเทียบกับค่าของข้อมูลเพื่อเป็นค่าที่จะเตือนถ้าค่าทั้งสองเท่ากัน เอเจนต์จะทราบได้ว่าเงื่อนไขของ Value Alert นั้นถูกพบแล้ว

ในบางครั้งเป็นไปได้ที่อาจจะพบค่าได้หลายค่า กล่าวคือ อาจจะมีค่าได้มากกว่า 1 ค่าในฟิลด์เดียว เช่น “Alert me when the state is NY or NJ” หรือบางครั้งค่าที่ระบุไว้ในฟิลด์อาจจะเป็น NOT เช่น “Alert me when the state is not CA” เป็นต้น ก็ได้ แต่ลักษณะของการค้นหาและเปรียบเทียบเงื่อนไขก็จะทำตามแบบเดียวกัน

(3) Reference Alert

จากข้อมูลที่มีมากมาย ไม่ว่าจะเป็นในรูปแบบของเอกสาร หรือในรูปแบบของเว็บเพจ มักจะมีส่วนของการอ้างอิง โดยแสดงในรูปชุดของคำหรือคำพูด ถ้าเรื่องที่กำลังสนใจมีส่วนใดส่วนหนึ่งของคำที่เข้ากับชุดของคำที่มีการอ้างอิง Alert ก็จะเกิดขึ้น Reference Alert มักจะนำมาใช้ในการดึงเอกสารที่มีอยู่จำนวนมากให้เหลือจำนวนน้อยลง

(4) New Information Alert และ Changed Information Alert

บางเวลาเอเจนต์มีความต้องการในการเสนอรายงานเมื่อส่วนใดส่วนหนึ่งของข้อมูลที่กำลังเฝ้าสังเกตอยู่มีการเปลี่ยนแปลง Changed Information Alert จะเก็บสำเนาข้อมูลที่เอเจนต์เฝ้าสังเกตไว้ และเมื่อข้อมูลล่าสุดแตกต่างกับสำเนาฉบับก่อนหน้า เงื่อนไขของการเตือนก็จะเริ่มขึ้น

ผู้ใช้จะนำ Changed Information Alert มาใช้ในเอเจนต์เพื่อแจ้งให้ผู้ใช้ทราบเกี่ยวกับเว็บเพจที่เขาหรือเธอชอบเมื่อเกิดการเปลี่ยนแปลง

ในขณะที่ New Information Alert จะมีการเก็บชุดของข้อมูลหรือหัวเรื่องไว้มากกว่าหลายเท่า โดยการเริ่มเตือนจะเกิดขึ้นเมื่อชุดของข้อมูลชุดปัจจุบันมีหัวเรื่องใหม่ที่ไม่อยู่ในสำเนาฉบับก่อนหน้า

(5) Task Completion Alert และ Task Failure Alert

Task Completion Alert จะแสดงสัญญาณเมื่อการปฏิบัติสัมฤทธิ์ผล โดยจะใช้เมื่องานของเอเจนต์นั้นทำสำเร็จ เช่น “All Web sites have been visited” หรือเป็นข้อมูลที่เอเจนต์บันทึกไว้โดยขึ้นกับข้อมูลที่ตรวจสอบ เช่น “Database server reports that it has finished defragmenting the customer database” เป็นต้น

Task Failure Alert คล้ายกับ Task Completion Alert แต่จะเป็นรายงานที่ไม่ประสบความสำเร็จ โดยที่ Task Failure Alert จะเป็นรายงานความผิดพลาดที่เอเจนต์ประสบแล้วสะท้อนกลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น “Could not open log file” หรือเป็นรายงานความผิดพลาดของข้อมูลที่เอเจนต์เฝ้าสังเกตอยู่ เช่น “Data source reports errors accessing request information” เป็นต้น

มันเป็นไปได้ไม่ได้ในการแยกแยะความแตกต่างของการเตือนในแต่ละประเภทได้หมด เพราะอินเทลลิเจนต์เอเจนต์สามารถออกแบบได้หลากหลายวัตถุประสงค์ ดังนั้นในตารางที่ 2.1 จะเป็นสรุปลักษณะเฉพาะของการเตือนแต่ละประเภทที่จะนำการเตือนในประเภทนั้นไปใช้หรือเหมาะสมกับเหตุการณ์ประเภทใดบ้าง

ตารางที่ 2.1 สรุปลักษณะเฉพาะของการเตือนแต่ละประเภท (Pallmann, 1999)

Alert Type	Characteristics	Best Suited for	Example
Threshold	A number increases or decreases by an amount that exceeds a value.	Numeric or enumerated data	“Alert me if company ABC’s stock price exceeds \$135/share.”
Value	A data field exactly matches a specified value.	Nonnumeric data fields	“Alert me if the furnace status Indicator reads ‘Over-Heating’”
Reference	A text stream or other data contains one or more references to a dictionary.	Documents	“Alert me if the document contains the words ‘energy’, ‘power’.”
Changed Information	A value changes from the last time it was inspected.	Fluctuations in prices, percentages, or rates in which any change is worth knowing about	“Alert me if product WIDGET’s price changes in its online catalog.”
New Information	A new item has been added to a list since the last time the list was obtained.	Advertising users about new material on a topic of interest to them	“Alert me whenever news headlines appear about company ABC.”
Task Completion	A function has completed, or acquired data indicates the completion of a task.	Confirmation	“Alert me when the backup completes.”
Task Failure	An error in performing a function causes notification	Notification of failures to user	“Alert me if the database posting fails”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวไปแล้วข้างต้นว่า เหตุการณ์เป็นส่วนที่ทำให้เกิดการแจ้งให้ทราบของเอเจนต์ เมื่อเหตุการณ์นั้นถูกกระตุ้น คำสั่งที่แสดงอาการตอบจะสามารถตรวจสอบจากเงื่อนไขของการเตือน เมื่อเงื่อนไขของการเตือนเกิดขึ้น เอเจนต์จำเป็นจะต้องมีการเตรียมการสื่อสารหรือปฏิบัติงานไว้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและพัฒนาระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ

3.1 ขอบเขตของออกแบบและการพัฒนาระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ

ทำการพัฒนาระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจที่ผู้ใช้สนใจ และค้นหาคำที่ผู้ใช้สนใจในเว็บเพจนั้น ๆ รวมไปถึงการแจ้งให้ผู้ใช้ทราบเมื่อเว็บเพจนั้นเกิดการเปลี่ยนแปลง โดยที่ระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ มีหน้าที่ เฝ้าสังเกตเว็บเพจ และติดตามเฝ้าดูการเปลี่ยนแปลงที่เกิดขึ้นอย่างต่อเนื่องตลอดเวลา โดยเมื่อมีการเปลี่ยนแปลงเอเจนต์จะทำการปฏิบัติงานที่เรียกว่า Trigger และมีการแสดงให้ผู้ใช้ทราบเมื่อมีการเปลี่ยนแปลง

3.2 การออกแบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ

จากการออกแบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ ได้ทำการออกแบบโดยแบ่งการเฝ้าสังเกตออกเป็น 4 ช่วงเวลา คือ

- ชั่วโมง ในทุก ๆ ชั่วโมง เอเจนต์จะทำการตรวจสอบข้อมูลปัจจุบันของเว็บเพจนั้นเปรียบเทียบกับข้อมูลที่เก็บไว้ก่อนหน้านี้ ถ้าข้อมูลของเว็บเพจนั้นเปลี่ยนแปลง ก็ทำมีกล่องข้อความเตือนขึ้นมาว่าเว็บเพจนั้นเกิดการเปลี่ยนแปลง และตรวจพบคำที่ค้นหาหรือไม่ อย่างไร

- วัน เมื่อเริ่มถึงเวลาที่กำหนดในวันถัดไป เอเจนต์จะทำการตรวจสอบข้อมูลปัจจุบันของเว็บเพจนั้นเปรียบเทียบกับข้อมูลที่เก็บไว้ก่อนหน้านี้ ถ้าข้อมูลของเว็บเพจนั้นเปลี่ยนแปลง ก็ทำมีกล่องข้อความเตือนขึ้นมาว่าเว็บเพจนั้นเกิดการเปลี่ยนแปลง และตรวจพบคำที่ค้นหาหรือไม่ อย่างไร

- สัปดาห์ เมื่อถึงวันนั้นของทุก ๆ สัปดาห์ เช่น กำหนดให้เอเจนต์ทำการตรวจสอบทุกวันจันทร์ เอเจนต์จะทำการตรวจสอบข้อมูลปัจจุบันของเว็บเพจนั้นเปรียบเทียบกับข้อมูลที่เก็บไว้ก่อนหน้านี้ ถ้าข้อมูลของเว็บเพจนั้นเปลี่ยนแปลง ก็ทำมีกล่องข้อความเตือนขึ้นมาว่าเว็บเพจนั้นเกิดการเปลี่ยนแปลง และตรวจพบคำที่ค้นหาหรือไม่ อย่างไร

- เดือน เมื่อถึงวันที่กำหนดของทุก ๆ เดือน เอเจนต์จะทำการตรวจสอบข้อมูลปัจจุบันของเว็บเพจนั้นเปรียบเทียบกับข้อมูลที่เก็บไว้ก่อนหน้านี้ ถ้าข้อมูลของเว็บเพจนั้นเปลี่ยนแปลง ก็ทำมีกล่องข้อความเตือนขึ้นมาว่าเว็บเพจนั้นเกิดการเปลี่ยนแปลง และตรวจพบคำที่ค้นหาหรือไม่ อย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในส่วนของ การนำเสนอ เมื่อเอเจนต์ตรวจสอบพบว่า ข้อมูลนั้นมีการเปลี่ยนแปลง จะทำการเปิดหน้าจอ Internet Explorer ของเว็บเพจนั้น ๆ ให้ผู้ใช้อย่างอัตโนมัติ และผู้ใช้สามารถทำการตรวจสอบได้ด้วยตัวเอง โดยที่ไม่ต้องรอให้โปรแกรมเอเจนต์ทำการตรวจสอบได้ อีกทั้งยังมีความสามารถในการส่งจดหมายอิเล็กทรอนิกส์ไปให้กับผู้อื่นได้อีกด้วย

3.3 ขั้นตอนการทำงานของระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ

โดยเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจจะมีการออกแบบตามลำดับขั้นตอนดังรูปที่ 3.1 ซึ่งขั้นตอนวิธีทั้งหมดของระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ มีดังต่อไปนี้

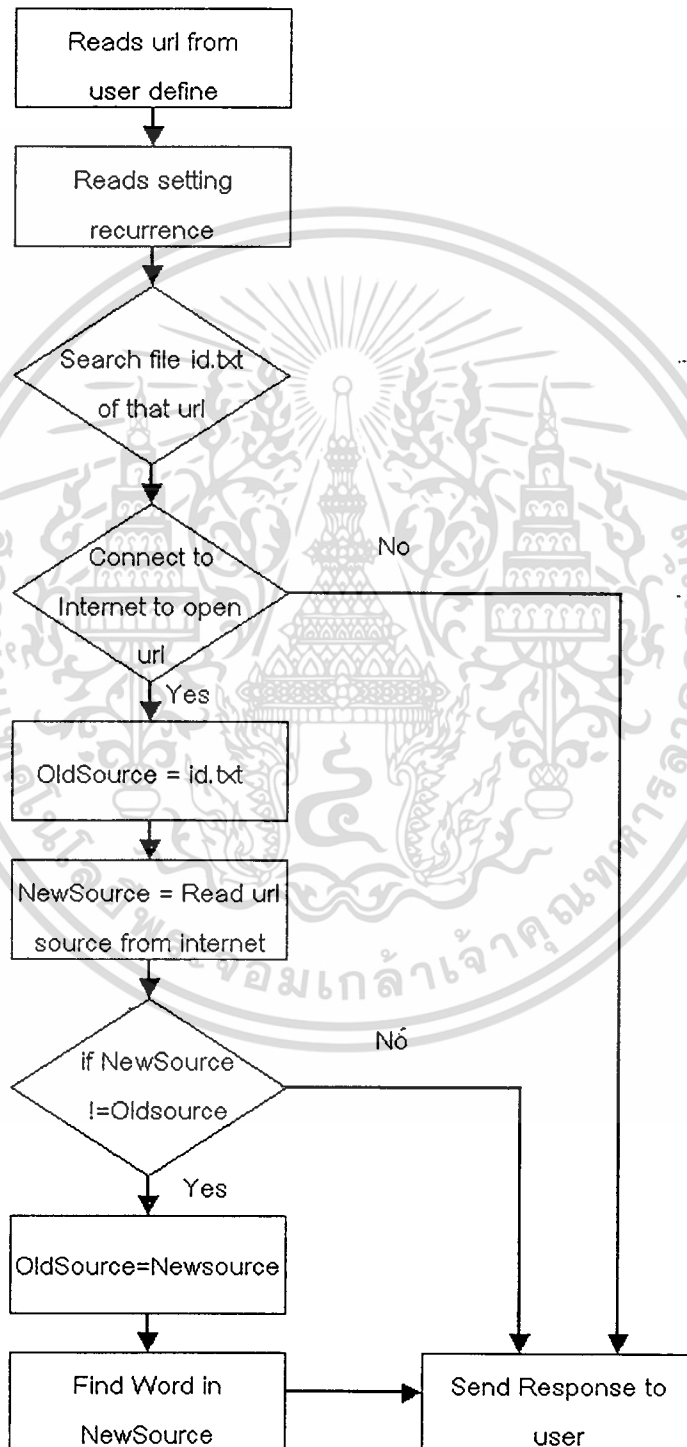
1. อ่านค่าของ Url ตามที่ผู้ใช้ได้กำหนดไว้
2. อ่านค่าเวลาที่ผู้ใช้ได้กำหนดไว้
3. ทำการค้นหา text file ที่จัดเก็บตามหมายเลข (id) ของ Url นั้น
4. สร้างการเชื่อมต่อกับอินเทอร์เน็ตเพื่ออ่านเอกสาร HTML ตาม Url นั้น
5. เมื่อทำการเชื่อมต่อเรียบร้อยแล้ว จะเก็บค่าของ text file ลงใน OldSource
6. อ่านเอกสาร HTML ตาม Url (โดยจะทำการอ่านเว็บเพจจากเว็บเซิร์ฟเวอร์ซึ่งจะอธิบายอย่างละเอียดในหัวข้อที่ 3.3.1) ลงใน NewSource
7. ทำการเปรียบเทียบกับเอกสาร HTML ฉบับก่อนหน้านี้ คือ OldSource โดยเทคนิคที่ใช้คือการเปรียบเทียบเอกสาร HTML นั้นแบบ string ต่อ string คือ ทำการเปรียบเทียบความยาวของ string ของเอกสาร HTML ใหม่ (NewSource) กับเอกสาร HTML ฉบับก่อนหน้า (OldSource) ที่ได้จัดเก็บเป็น text file เอาไว้
8. ถ้าหากว่าเอกสารที่เปรียบเทียบแตกต่างกัน นั้นแสดงว่า เอกสาร HTML ของ Url นั้นมีการเปลี่ยนแปลง ดังนั้นจะทำการเก็บเอกสาร NewSource แทนที่เอกสาร OldSource
9. ทำการค้นหาคำที่ผู้ใช้กำหนดในเอกสาร HTML นั้น โดยใช้หลักการการค้นหาคำ ๆ นั้นในเอกสาร HTML และนับจำนวนของคำนั้นเพิ่มขึ้นเรื่อย ๆ (count)
10. ส่งผลการตอบสนอง (response) ว่า เว็บเพจนั้นมีการเปลี่ยนแปลงหรือไม่ และพบจำนวนของคำนั้น ๆ ที่ผู้ใช้สนใจในเอกสารนั้นเป็นจำนวนเท่าไรกลับมายังผู้ใช้ โดยผลการตอบสนองต่อผู้ใช้นี้

- เมื่อไม่สามารถทำการเชื่อมต่อกับ Internet จะมีผลตอบสนองต่อผู้ใช้ในรูปกล่องข้อความว่า "Could not establish a connection with the server..."

- เมื่อสามารถทำการเชื่อมต่อกับ Internet ได้แต่เว็บเพจนั้นไม่เกิดการเปลี่ยนแปลง จะมีผลตอบสนองต่อผู้ใช้ในรูปกล่องข้อความว่า "not update ..."

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อสามารถทำการเชื่อมต่อกับ Internet ได้และเว็บเพจนั้นเกิดการเปลี่ยนแปลง จะมีผลตอบสนองของผู้ใช้ในรูปกล่องข้อความว่า “The Url has updated ...”



รูปที่ 3.1 แผนภาพการทำงานของเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ

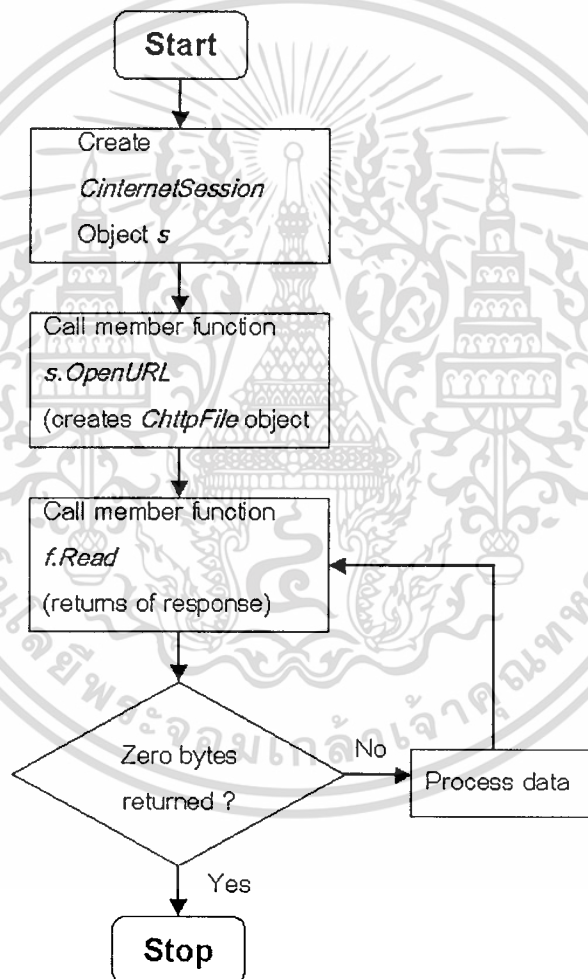
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเข้าถึงอินเทอร์เน็ตโดยใช้โปรโตคอล HTTP

Microsoft Windows ได้จัดเตรียมวิธีในการเข้าถึงอินเทอร์เน็ตไว้หลายวิธี หนึ่งในแนวทางที่ง่าย มีความน่าเชื่อถือได้ และค่อนข้างที่จะเป็นสากล นั่นก็คือ การใช้งาน WinInet Class ซึ่งถือเป็น Internet Explorer Component สำหรับการเข้าถึงอินเทอร์เน็ต

3.3.1 ขั้นตอนในการอ่านเว็บเพจ

ในการอ่านเว็บเพจหรือ ไฟล์ข้อมูล จะใช้งาน HTTP GET function จากรูปที่ 3.2 แสดงการอ่านข้อมูลจากเว็บเซิร์ฟเวอร์ซึ่งเป็นไปตามลำดับขั้นตอน ดังนี้



รูปที่ 3.2 การทำงานของ HTTP GET function

1. ทำการสร้าง CInternetSession object โดยรูปที่ 3.3 แสดงลักษณะของคำสั่งจะประกอบด้วย ชื่อของเอเจนต์ context ID และลักษณะของการเข้าถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CInternetSession Session ("Agent Program",

1,

INTERNET_OPEN_TYPE_PRECONFIG);

รูปที่ 3.3 คำสั่งในการสร้าง CInternetSession object

2. ทำการสร้าง CHttpFile object โดยจะเรียกใช้งาน OpenUrl member function ผ่านทาง CInternetSession object ในการเรียก OpenUrl function ของ CInternetSession เพื่อเข้าถึงข้อมูลบนเว็บเซิร์ฟเวอร์จะต้องระบุ Url, context ID, access flags และส่วนของ header ซึ่งรูปแบบการใช้งานเป็นดังรูปที่ 3.4

```
DWORD dwFlags = INTERNET_FLAG_DONT_CACHE
                | INTERNET_FLAG_TRANSFER_BINARY
                | INTERNET_FLAG_EXISTING_CONNECT
                | INTERNET_FLAG_RELOAD;
CString sHeader = "Accept : /*\n";
CHttpFile* pHttpFile = (CHttpFile*)Session.OpenURL(sUrl,
                                                    1,
                                                    dwFlags,
                                                    sHeader,
                                                    -1L);
```

รูปที่ 3.4 คำสั่งในการสร้าง CHttpFile object

ถ้า CHttpFile เป็น NULL การ OpenUrl นั้นจะถือว่าล้มเหลว คือ ไม่สามารถหา Url ได้

3. เรียก Read member function ของ CHttpFile object ซึ่งจะคืนค่าเป็นส่วนหนึ่งของข้อมูลตอบสนองต่อโปรแกรม ทำซ้ำไปจนกระทั่งการ Read คืนค่ากลับมาเป็น 0 โดยทำเป็นลักษณะของ do while loop คือ ถ้าไม่มีข้อมูลก็จะออกจาก loop ดังรูปที่ 3.5

```

Char buffer[1024];
Int nRead;
Do
{
    nRead = pHttpFile->Read(buffer, 1023);
    if (nRead != 0)
    {
        buffer[nRead] = 0;
        sResult += buffer;
    }
} while (nRead != 0);

```

รูปที่ 3.5 คำสั่งในการใช้งาน Read member function

4. ทำการลบ object ที่สร้างมาทั้งหมด โดยใช้คำสั่งดังรูปที่ 3.6

```
delete pHttpFile;
```

รูปที่ 3.6 คำสั่งในการลบ object

เมื่อนำคำสั่งในแต่ละส่วนมารวมเข้าด้วยกัน จะเป็นคำสั่งที่ใช้ในการอ่านเว็บเพจหรือไฟล์ข้อมูล จากรูปที่ 3.7 แสดงตัวอย่างคำสั่ง โดยจะนำคำสั่งต่าง ๆ ในรูปที่ 3.3 –3.6 มารวมกันเพื่อเป็นคำสั่งที่ใช้ในการอ่านเว็บเพจ ซึ่งจะมีการกำหนดค่าของตัวแปร sUrl เป็น Url ที่ต้องการจะเข้าถึงเมื่อโปรแกรมเริ่มทำงาน เช่น “http://www/it/kmitl/ac/th” ดังรูป และมีตัวแปร sResult เป็นตัวแปรที่เก็บผลการตอบสนองของ server เมื่อการทำงานเสร็จสิ้น โดยตัวแปร sUrl จะระบุเป็นค่าคงที่ และผลที่ตอบสนองกลับมาจะอยู่ในรูปของกล่องข้อความ

```

CString sURL = "http://www.it.kmitl.ac.th";
CString sResult = "";
Char buffer[1024];
Int nRead;
Try {
DWORD dwFlags = INTERNET_FLAG_DONT_CACHE
                | INTERNET_FLAG_TRANSFER_BINARY
                | INTERNET_FLAG_EXISTING_CONNECT
                | INTERNET_FLAG_RELOAD;
CString sHeader = "Accept : /* \r\n";
ChttpFile* pHttpFile = (ChttpFile*)Session.OpenURL(sUrl, 1, dwFlags, sHeader, -1L);
if(pHttpFile){
do {
nRead = pHttpFile->Read(buffer, 1023);
if (nRead != 0)
{
buffer[nRead] = 0;
sResult += buffer;
}
} while (nRead != 0);
delete pHttpFile;
AfxMessageBox(sResult);
}
else { AfxMessageBox("An HTTP Connection Could no be open to " + sURL); }
catch(CInternetException* pEx)
{
AfxMessageBox("An exception occurred attempting to access " + sURL);
pEx->Delete();
}
}

```

รูปที่ 3.7 ตัวอย่างคำสั่งในการอ่านเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การติดต่อกับอีเมลโดยใช้โพรโทคอล MAPI

วิธีการในการเข้าถึงอีเมลใน Windows มี 2 วิธี คือ การใช้ MAPI (Microsoft's Mail API) และการใช้ SMTP/POP3 (Simple Mail Transfer Protocol and Post Office Protocol, Version 3) ในอดีตการติดต่อกับผู้อื่นที่อยู่บนเครือข่ายของผู้ใช้เองจะต้องใช้ MAPI และการติดต่อกับผู้อื่นที่อยู่บนเครือข่ายอินเทอร์เน็ตจะต้องใช้โพรโทคอล TCP/IP SMTP เพื่อใช้สำหรับการส่งเมล และจะต้องใช้โพรโทคอล POP3 สำหรับอ่านเมล แต่ในปัจจุบันการติดต่อสื่อสารแบบ MAPI ได้รวบรวมความสามารถของ SMTP/POP3 เข้าไปด้วย ทำให้มีความสามารถในการติดต่อกับบุคคลอื่นทั้งภายในและภายนอกบริษัทได้ และเทคโนโลยีของโพรโทคอล TCP/IP ที่สนับสนุนอินเทอร์เน็ตก็ยังคงเป็นส่วนหนึ่งของอินเทอร์เน็ตของแต่ละบริษัทด้วย

3.4.1 ชนิดของ MAPI

MAPI สามารถแบ่งได้เป็น 3 แบบ คือ

1. Simple MAPI หรือในบางครั้งเรียกว่า SMAPI
2. Common messaging calls (CMC)
3. Full MAPI

Simple MAPI และ CMC เป็นส่วนหนึ่งของ Full MAPI โดยปกติการใช้งาน Full MAPI จะใช้งานเมื่อต้องการการทำงานในระดับสูง เช่น การทำ mail server เป็นต้น

3.4.2 การติดต่อกับ MAPI

ในการติดต่อกับ MAPI โดยปกติจะเรียกใช้งานฟังก์ชันต่างๆ เช่น MAPISession และ MAPIMessage โดยที่ในการใช้งาน MAPI จะต้องทำขั้นตอนดังนี้

1. ทำการโหลด Mapi32.dll จากรูปที่ 3.8 แสดงลักษณะของคำสั่งเพื่อโหลด Mapi32.dll ถ้าฟังก์ชัน LoadLibrary แสดงค่าคืนกลับมาเป็น NULL แสดงว่าไม่สามารถทำการโหลด Mapi32.dll ได้

```
m_hInstMail = ::LoadLibrary("Mapi32.dll");
if(m_hInstMail == NULL)
{
    ... handle error ...
}
```

รูปที่ 3.8 คำสั่งในการ โหลด Mapi32.dll

2. หา address ของฟังก์ชัน MAPI ที่ต้องการใช้งาน จากรูปที่ 3.9 แสดงคำสั่งในการระบุ address ของฟังก์ชัน เมื่อทำการโหลด Mapi32.dll เรียบร้อยแล้ว จะทำการเริ่มการเรียกใช้งานฟังก์ชันเพื่อที่จะทำการระบุ address ที่แน่นอนของฟังก์ชันนั้น ๆ

```

ULONG (PASCAL *lpfnMAPISendMail)(ULONG, ULONG, MapiMessage*,
                                FLAGS, ULONG);
ULONG (PASCAL *lpfnMAPIResolveName)(LHANDLE, ULONG, LPTSTR,
                                    FLAGS, ULONG, MapiRecipDesc **);
ULONG (FAR PASCAL *lpfnMAPILogon)(ULONG, LPSTR, LPSTR,
                                  FLAGS, ULONG, LPLHANDLE);
ULONG (FAR PASCAL *lpfnMAPILogoff)(LHANDLE, ULONG, FLAGS, ULONG);
ULONG (FAR PASCAL *lpfnMAPIFreeBuffer)(LPVOID);
ULONG (FAR PASCAL *lpfnMAPIAddress)(LHANDLE, ULONG, LPSTR, ULONG,
                                    LPSTR, ULONG, MapiRecipDesc *, FLAGS,
                                    ULONG, LPULONG, MapiRecipDesc **);
ULONG (FAR PASCAL *lpfnMAPIFindNext)(LHANDLE, ULONG, LPSTR,
                                      LPSTR, FLAGS, ULONG, LPSTR);
ULONG (FAR PASCAL *lpfnMAPIReadMail)(LHANDLE, ULONG, LPSTR, FLAGS,
                                     ULONG, lpMapiMessage FAR *lpMessage);

```

รูปที่ 3.9 คำสั่งในการประกาศ Function pointer

เมื่อทำการประกาศ Function pointer แล้ว จะทำการค้นหา address ฟังก์ชันของ Mapi32.dll เหล่านั้น แล้วเก็บลงในตัวแปร Function pointer ตามที่ได้ประกาศไว้ ดังคำสั่งในรูปที่ 3.10

```

(FARPROC&)lpfnMAPILogon = GetProcAddress(m_hInstMail, "MAPILogon");
    if (lpfnMAPILogon == NULL)
    {
        ... handle error ...
    }

(FARPROC&)lpfnMAPIFindNext = GetProcAddress(m_hInstMail, "MAPIFindNext");
    if (lpfnMAPIFindNext == NULL)
    {
        ... handle error ...
    }

(FARPROC&)lpfnMAPIReadMail = GetProcAddress(m_hInstMail, "MAPIReadMail");
    if (lpfnMAPIReadMail == NULL)
    {
        ... handle error ...
    }

(FARPROC&)lpfnMAPIFreeBuffer = GetProcAddress(m_hInstMail, "MAPIFreeBuffer");
    if (lpfnMAPIFreeBuffer == NULL)
    {
        ... handle error ...
    }

(FARPROC&)lpfnMAPIResolveName = GetProcAddress(m_hInstMail,
                                                MAPIResolveName");
    if (lpfnMAPIResolveName == NULL)
    {
        ... handle error ...
    }

```

รูปที่ 3.10 คำสั่งในการค้นหา address ของฟังก์ชันใน Mapi32.dll

3. ทำการ log on ไปยัง mail session โดยปกติใน MAPI มี 3 แนวทางในการ log on เข้าสู่ระบบอีเมลของผู้ใช้เอง คือ
 - log on เข้าสู่ session ที่สร้างไว้แล้ว
 - log on เข้าสู่ session ใหม่ โดยการระบุข้อมูลของ account
 - ใช้ dialog box สำหรับผู้ใช้เพื่อ log on

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติในการ log on เข้าสู่การใช้งาน มักใช้ MAPI session ที่มีอยู่แล้ว โดยจะมีการเรียกใช้งานฟังก์ชัน MAPILogon

```
ULONG IResult = IpfmMAPILogon(0, NULL, NULL, 0, 0, &m_lhSession);
if (IResult != SUCCESS_SUCCESS)
    ... handle error ...
```

รูปที่ 3.11 คำสั่งในการ log on

3.4.3 การส่งอีเมล

โดยปกติในการส่งอีเมลด้วย MAPI จะประกอบด้วยโครงสร้าง 4 ส่วน คือ

1. MapiMessage ทำการกำหนดโครงสร้างของข้อความในจดหมายที่จะส่ง โดยใช้คำสั่งดังรูปที่ 3.12
2. MapiResolveName ทำหน้าที่ในตรวจสอบชื่อ โดยใช้คำสั่งดังรูปที่ 3.13
3. MapiSendMail ทำหน้าที่ส่งอีเมล หลังจากที่มีการคืนค่าของ SUCCESS_SUCCESS จะเป็นไปดังรูปที่ 3.14
4. MapiFreeBuffer ทำหน้าที่ในการลบข้อมูลออกจากบัฟเฟอร์หลังจากที่ได้ส่งเมลไปแล้ว โดยจะใช้คำสั่ง MapiFreeBuffer ดังรูปที่ 3.15

```

MapiMMessage message;
memset(&m_message, 0, sizeof(m_message)); // Set initialize to byte of 0
    m_message. ulReserved = 0;           // Set ulReserved to 0
    m_message. lpszMessageType = NULL; // Set lpszMessageType = NULL
    char subject[512];
    strcpy(subject, sSubject);
    m_message. lpszSubject = subject;    // Set lpszSubject to the buffer address
    char text[5000];
    strcpy(text, sMessage);
    m_message. lpszNoteText = text;     // lpszNoteText is a pointer to message
    m_message. flFlags = MAPI_SENT;     // flFlags is an integer contain bit flags
    m_message. lpOriginator = NULL;     // lpOriginator is a pointer to sender
                                         message
    m_message. nRecipCount = 1;         // nRecipCount is a number of recipients
                                         should be 1 or higher

```

รูปที่ 3.12 การกำหนดโครงสร้างของ MapiMessage

```

char recipient[512];
strcpy(recipient, sTo);
ULONG IResult = lpfnMAPIResolveName(m_lhSession, 0, recipient,
                                     0, 0, &m_message.lpRecips);
if (IResult == SUCCESS_SUCCESS)
    ... handle error ...

```

รูปที่ 3.13 คำสั่งในการตรวจสอบชื่อ

```
IResult = lpfnMAPISendMessage(0, 0, &m_message, 0, 0);
```

รูปที่ 3.14 คำสั่งในการส่งเมล

```

IResult = lpfnMAPI SendMail(0, 0, &m_message, 0, 0);
lpfnMAPI FreeBuffer(m_message.lpRecips);
if (IResult == SUCCESS_SUCCESS)
    ... handle error ...

```

รูปที่ 3.15 คำสั่งในการลบข้อมูลออกจากบัฟเฟอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

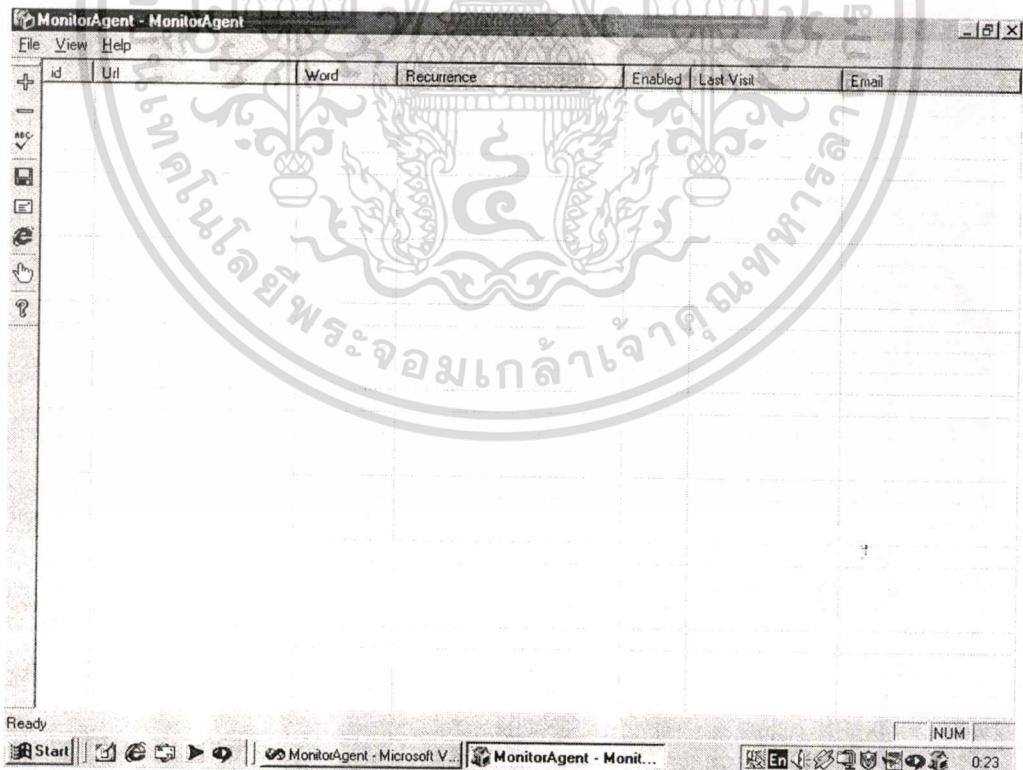
การทำงานของโปรแกรม

จากการพัฒนาระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ สามารถสรุปขั้นตอนในการทำงานของโปรแกรมได้ดังนี้

1. เมื่อเริ่มทำการรัน โปรแกรมจะเป็นดังรูปที่ 4.1 โปรแกรมจะมีการติดต่อกับฐานข้อมูลคือ MonitorAgent.mdb เพื่อเก็บข้อมูล และติดต่อกับ Microsoft Outlook เพื่อใช้ในการส่งเมล
2. ทำการใส่ข้อมูลของ Url ที่ผู้ใช้สนใจหรือต้องการให้โปรแกรมทำการเฝ้าสังเกตโดยการคลิกที่ปุ่ม Add บน Toolbar ดังรูปที่ 4.2 ข้อมูลที่ผู้ใช้ต้องใส่ คือ ชื่อของ Url นั้น คำที่สนใจหรือต้องการค้นหา (ถ้าหากคำที่ผู้ใช้สนใจมีมากกว่า 1 คำให้ทำการค้นคำเหล่านั้นด้วย “;”) และอีเมล (ผู้ใช้จะใส่หรือไม่ใส่ก็ได้)
3. ถ้าต้องการให้โปรแกรมทำการตรวจสอบให้อย่างอัตโนมัติให้คลิกเครื่องหมายหน้า Enabled Recurrence ดังรูปที่ 4.3 และถ้าหากต้องการยกเลิกการตรวจสอบของโปรแกรมสามารถทำได้โดยการคลิกเครื่องหมายหน้า Enabled Recurrence ออก
4. ทำการกำหนดช่วงเวลาในให้โปรแกรมทำการตรวจสอบโดยจะแบ่งการตรวจสอบออกเป็นทุกชั่วโมง ทุกวัน ทุกสัปดาห์ และทุกเดือน ดังรูปที่ 4.4
5. เมื่อทำการกรอกข้อมูลต่าง ๆ เรียบร้อยแล้ว จะเป็นไปดังรูปที่ 4.5
6. หลังจากนั้นเมื่อเวลาผ่านไปจนกระทั่งครบช่วงเวลาที่ใช้ได้ทำการกำหนด โปรแกรมจะทำการตรวจสอบเว็บเพจนั้นว่ามีการเปลี่ยนแปลงหรือไม่ และตรวจสอบในเอกสาร HTML แล้วพบคำที่ผู้ใช้สนใจเป็นจำนวนกี่คำ โปรแกรมจะแสดงผลให้ผู้ใช้ทราบผลการทำงานผ่านทาง Message Box ดังรูปที่ 4.6
7. ถ้าหากผู้ใช้ต้องการลบข้อมูลที่ทำการกรอกไปแล้ว สามารถลบได้โดยการกดปุ่ม Delete บน Toolbar โปรแกรมจะทำการลบข้อมูลนั้นออกจากระบบ ดังรูปที่ 4.7
8. ถ้าหากผู้ใช้ต้องการแก้ไขข้อมูลที่ทำการกรอกไปแล้ว สามารถแก้ไขได้โดยการกดปุ่ม Edit บน Toolbar ดังรูปที่ 4.8

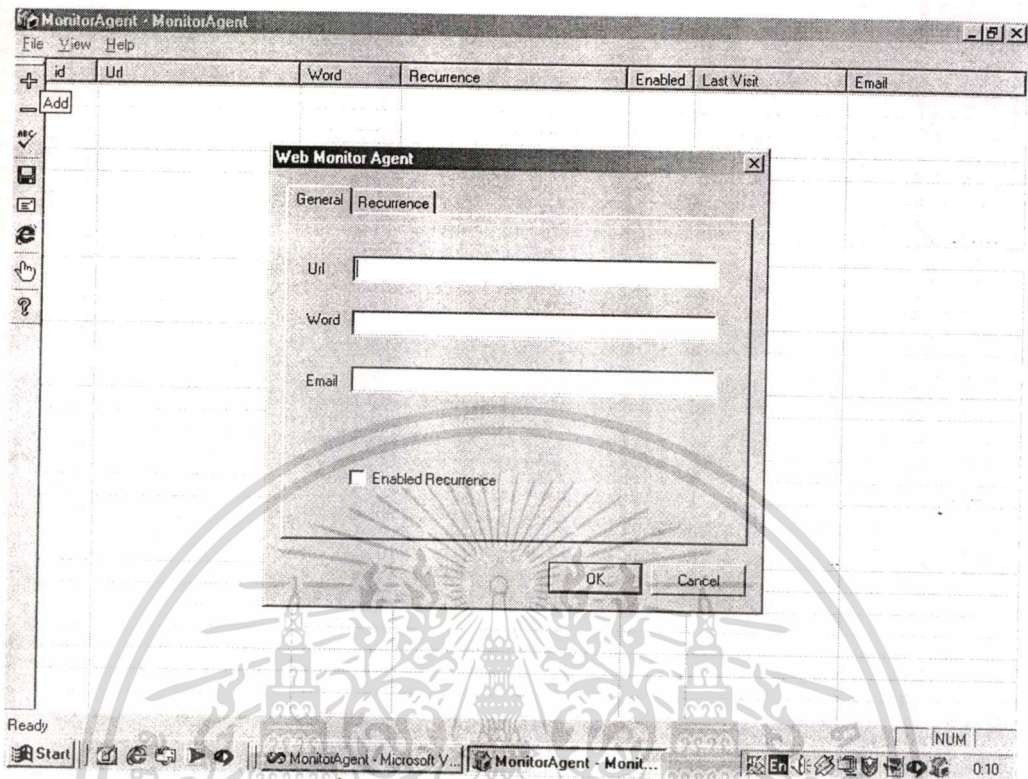
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ถ้าหากผู้ใช้ต้องการบันทึกเอกสาร HTML สามารถบันทึกได้โดยการกดปุ่ม Save บน Toolbar โปรแกรมจะขึ้นหน้าจอเพื่อให้ผู้ใช้ระบุว่าจะทำการบันทึก HTML นั้นไว้ที่ใด ดังรูปที่ 4.9
10. ถ้าหากผู้ใช้ต้องการส่งอีเมล สามารถส่งอีเมลได้โดยการกดปุ่ม Mail บน Toolbar โดยการส่งเมลจะส่งผ่านโปรแกรม Microsoft Outlook ดังรูปที่ 4.10
11. ถ้าหากผู้ใช้ต้องการเปิดหน้าจอ Internet Explorer ของเว็บเพจ สามารถเปิดเว็บเพจนั้นได้โดยการกดปุ่ม Browse บน Toolbar โปรแกรมจะทำการเปิดหน้าจอ Internet Explorer ของเว็บเพจนั้นทันที ดังรูปที่ 4.11
12. ถ้าหากผู้ใช้ต้องการตรวจสอบเว็บเพจนั้นด้วยตัวเอง สามารถตรวจสอบเว็บเพจนั้นๆ ได้โดยการกดปุ่ม Manual Monitor บน Toolbar โปรแกรมจะทำการตรวจสอบว่าเว็บเพจนั้นมีการเปลี่ยนแปลงและพบค่าที่ผู้ใช้ค้นหาหรือไม่และแจ้งให้ผู้ใช้ทราบทันทีเมื่อตรวจสอบเสร็จ ดังรูปที่ 4.12
13. ข้อมูลเกี่ยวกับโปรแกรม สามารถดูได้ โดยการกดปุ่ม About บน Toolbar ดังรูปที่ 4.13

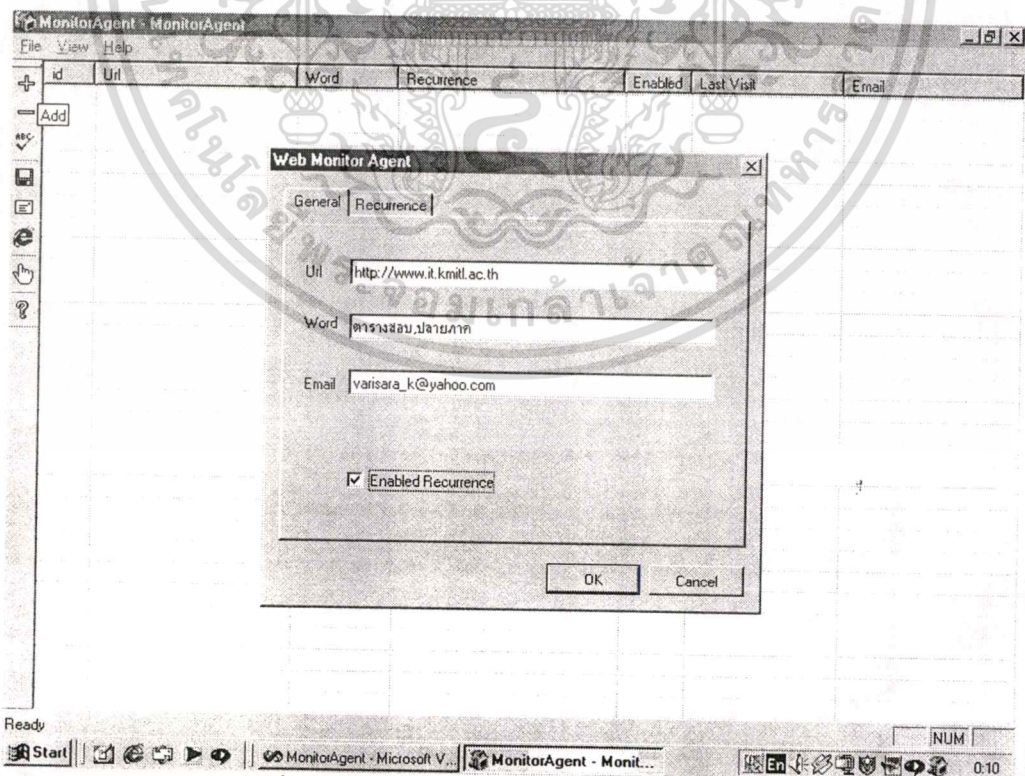


รูปที่ 4.1 หน้าจอของโปรแกรมเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

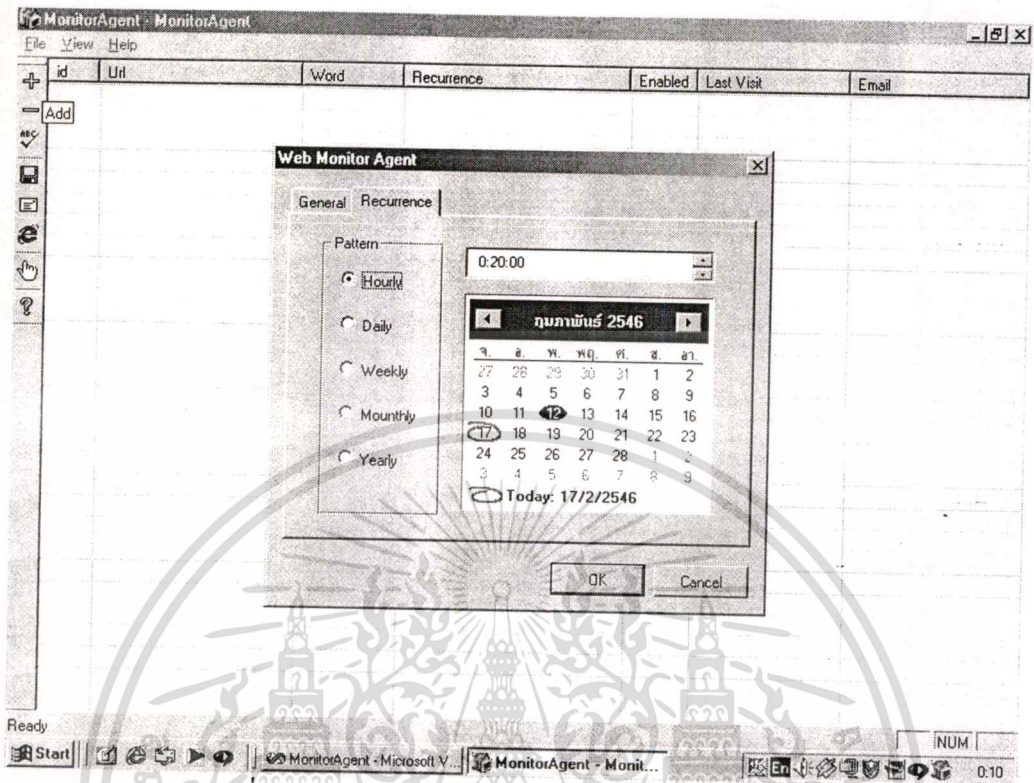


รูปที่ 4.2 หน้าจอเมื่อทำการกดปุ่ม Add

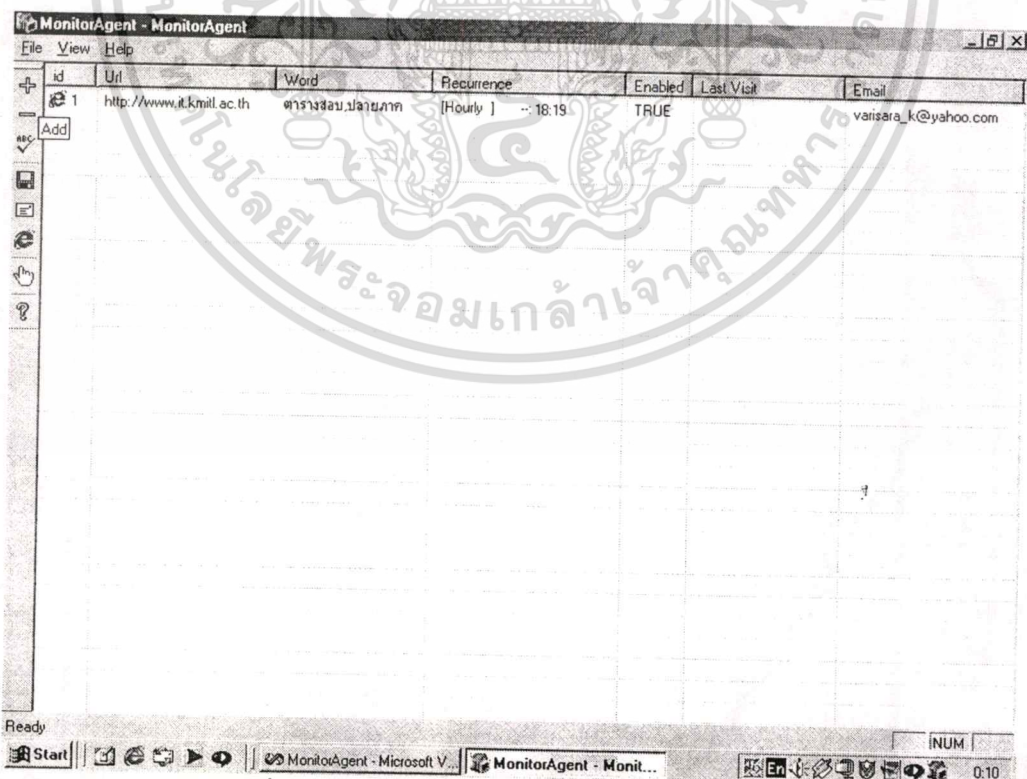


รูปที่ 4.3 หน้าจอกรอกตัวอย่างข้อมูลของเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

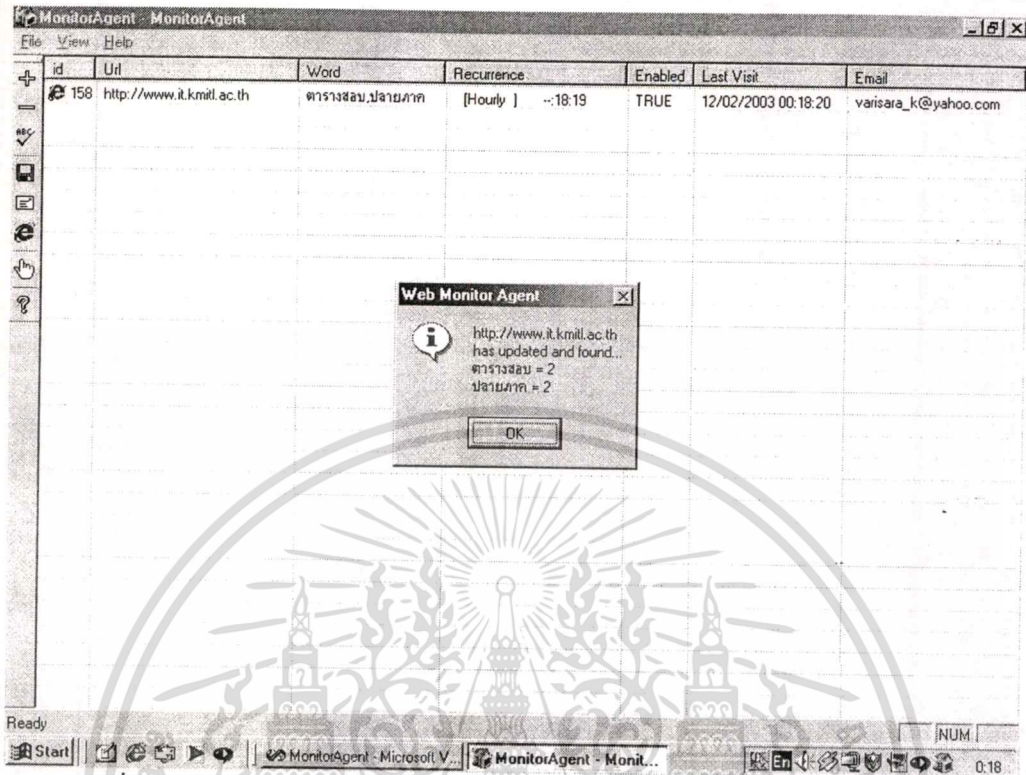


รูปที่ 4.4 หน้าจอการกำหนดช่วงเวลาในการตรวจสอบ

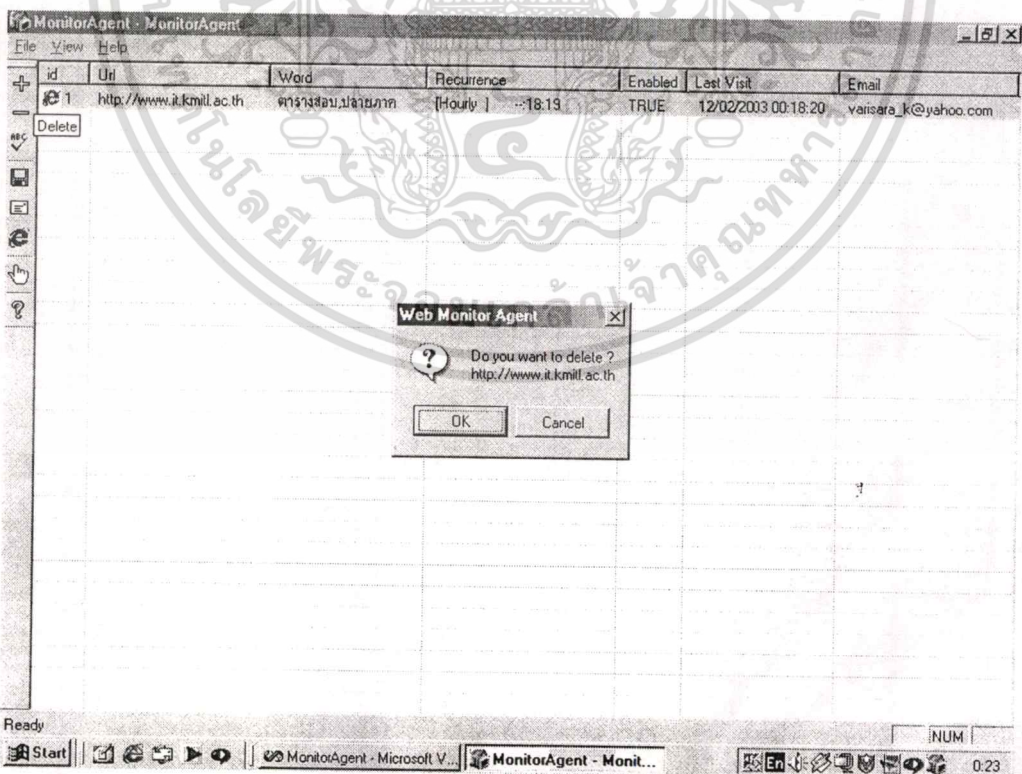


รูปที่ 4.5 หน้าจอเมื่อทำการเพิ่มข้อมูลเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

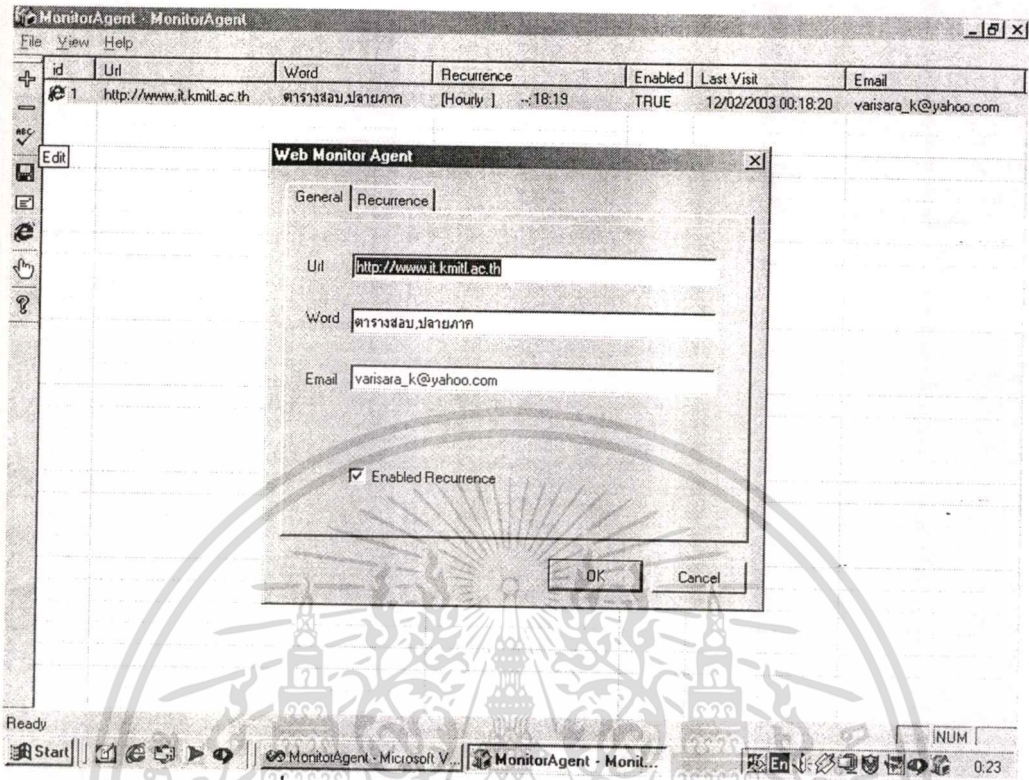


รูปที่ 4.6 หน้าจอแสดงผลให้ผู้ใช้งานทราบเมื่อตรวจสอบแล้วพบการเปลี่ยนแปลง

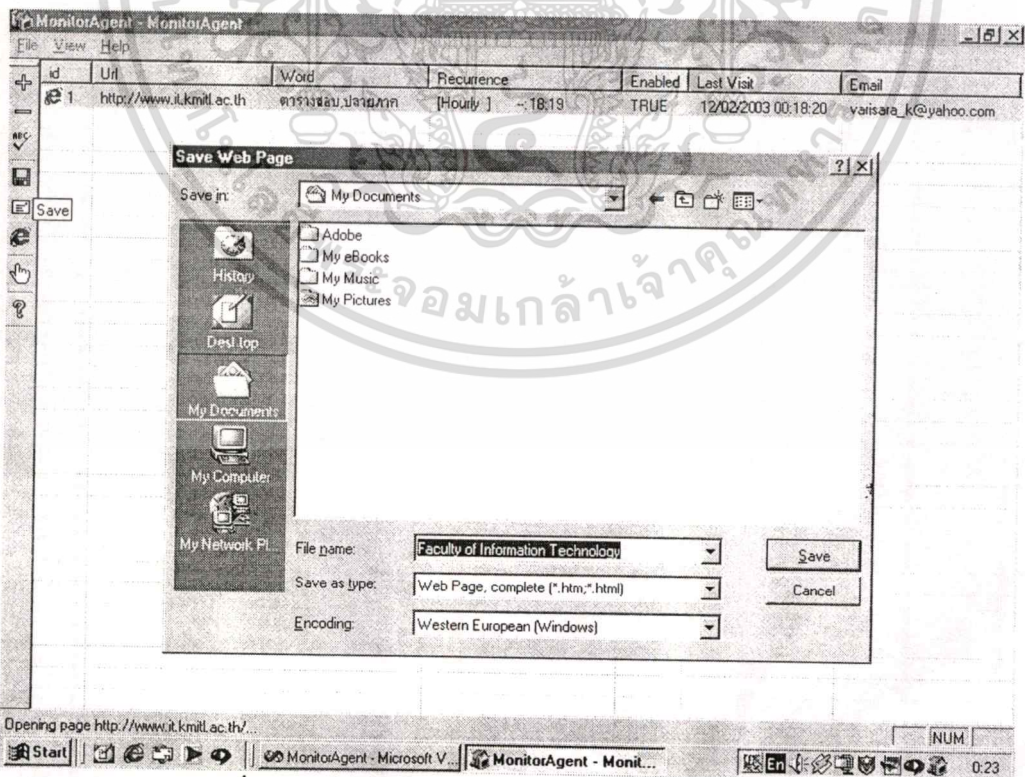


รูปที่ 4.7 หน้าจอเมื่อต้องการลบ Uri

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

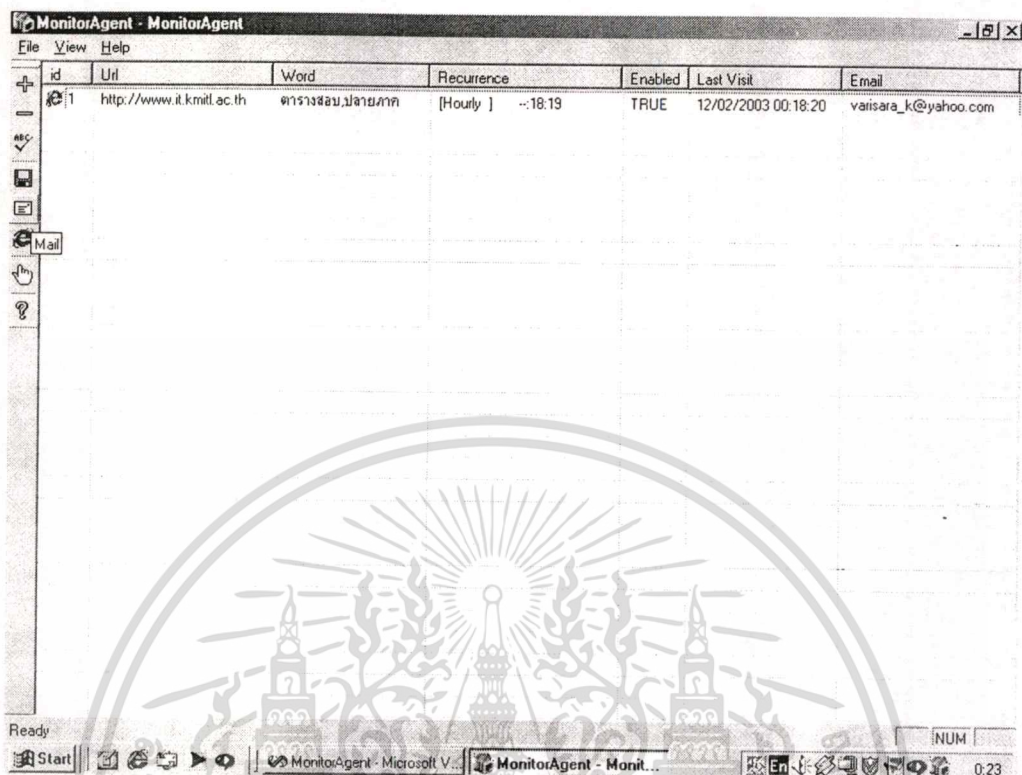


รูปที่ 4.8 หน้าจอเมื่อต้องการแก้ไขข้อมูลของ Url

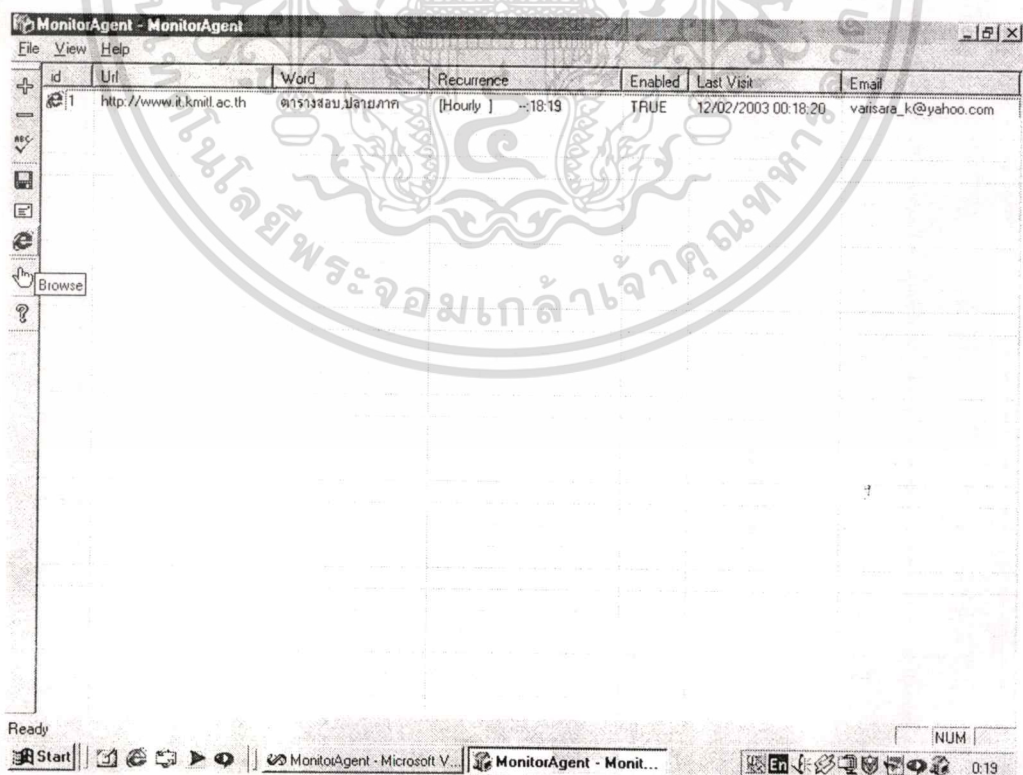


รูปที่ 4.9 หน้าจอเมื่อต้องการบันทึกเอกสาร HTML นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

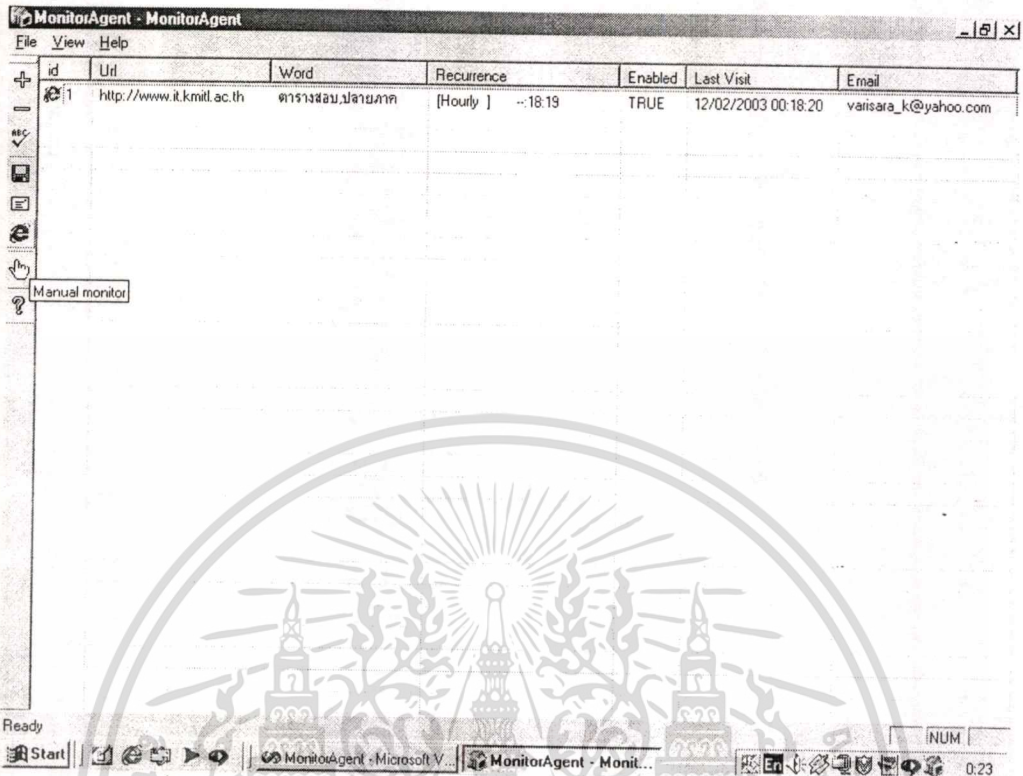


รูปที่ 4.10 หน้าจอเมื่อต้องการส่งเมล

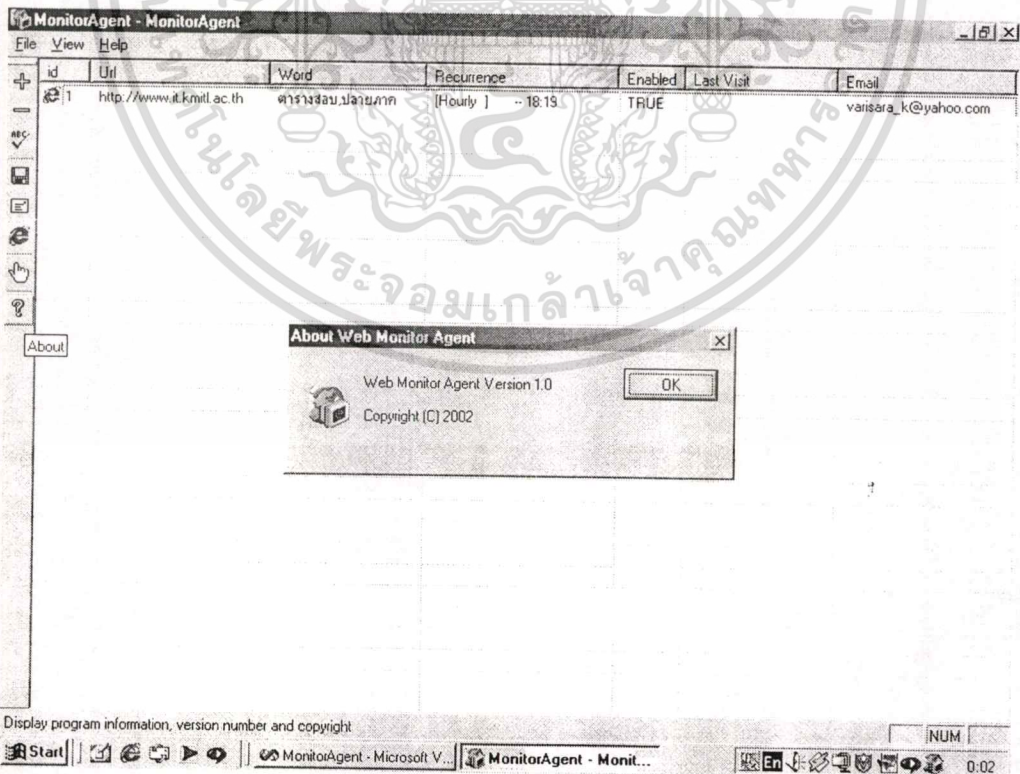


รูปที่ 4.11 หน้าจอเมื่อต้องการเปิดหน้า Internet Explorer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 หน้าจอเมื่อต้องการทำการตรวจสอบเว็บเพจนั้นด้วยตัวเอง



รูปที่ 4.13 หน้าจอข้อมูลเกี่ยวกับโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป

จากการศึกษาการวิเคราะห์และออกแบบระบบที่ติดต่อกับโลกอินเทอร์เน็ตผ่านทาง WinInet Class เพื่อดึงข้อมูลเว็บเพจและใช้ Mapi32.dll สำหรับการส่งเมล โดยนำทั้ง 2 ส่วนมาใช้ในการพัฒนาเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ ในส่วนของการพัฒนาได้ใช้ภาษา Microsoft Visual C++ 6.0 ภายใต้ Environment ของ Windows 2000

5.1 ผลการพัฒนาเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ

จากการศึกษาและพัฒนาระบบเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจสามารถสรุปผลการศึกษาดังนี้

1. ในการพัฒนาอินเทอร์เน็ตเอเจนต์ส่วนที่ยากที่สุด คือการสร้างฐานความรู้ และยังคงต้องคำนึงถึงการแสดงความรู้ สถาปัตยกรรม การทำงานร่วมกันระหว่างเอเจนต์ และภาษาที่ใช้ด้วยเสมอ โดยจะต้องเลือกให้เหมาะสมกับระบบที่พัฒนามากที่สุด
2. เอเจนต์เฝ้าสังเกตจะทำการตอบสนองต่อเหตุการณ์ด้วยการกระทำ บางการกระทำอาจจะแบ่งแยกอย่างละเอียดหรือบางการกระทำอาจจะมองไม่เห็น ในขณะที่บางครั้งอาจจะส่งเสียงดังหรือทำเพียงแค่การแจ้งให้ผู้ใช้ทราบด้วยข้อความก็ได้ โดยจะมีการเตือนเป็นคำสั่งหรือเงื่อนไขเพื่อให้เกิดการกระทำซึ่งสิ่งเหล่านี้ผู้ใช้จะเป็นคนกำหนด
3. เหตุการณ์เป็นส่วนที่ทำให้เกิดการแจ้งให้ทราบของเอเจนต์ เมื่อเหตุการณ์นั้นถูกกระตุ้น คำสั่งที่แสดงอาการตอบจะสามารถตรวจสอบจากเงื่อนไขของการเตือน ดังนั้นในการพัฒนาผู้พัฒนาอินเทอร์เน็ตเอเจนต์ควรจะคิดไว้เสมอว่า ในการออกแบบเอเจนต์เฝ้าสังเกตส่วนที่ยากที่สุด คือการออกแบบเหตุการณ์และส่วนของการเตือน
4. ในการพัฒนาใช้ความสามารถของ WinInet Class ของ Window เพื่อเชื่อมต่อเครื่องของผู้ใช้งานเข้าสู่โลกอินเทอร์เน็ตเพื่อทำการดึงข้อมูลเว็บเพจจากเว็บเซิร์ฟเวอร์แล้วนำมาเก็บไว้ที่เครื่องของผู้ใช้ และใช้โปรโตคอล MAPI สำหรับการส่งเมลโดยใช้งานผ่านทาง Microsoft Outlook และทำการพัฒนาโดยใช้ภาษา Visual C++ 6.0 เพื่อทำการพัฒนาโปรแกรมเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจ เนื่องจากมีความยืดหยุ่นสูงและสามารถถ่ายทอดคลาสที่นำมาใช้งานได้ และใช้คลาส afxdao เพื่อเชื่อมต่อกับฐานข้อมูล

MonitorAgent.mdb ที่ผู้พัฒนาสร้างขึ้นมาเพื่อใช้ในการจัดเก็บข้อมูลของ Url ต่าง ๆ ที่ผู้ใช้งานต้องการเฝ้าสังเกต

5. จากการทดสอบการทำงานของโปรแกรมเอเจนต์เพื่อเฝ้าสังเกตเว็บเพจสามารถทำงานได้ตามขอบเขตและวัตถุประสงค์ที่วางไว้

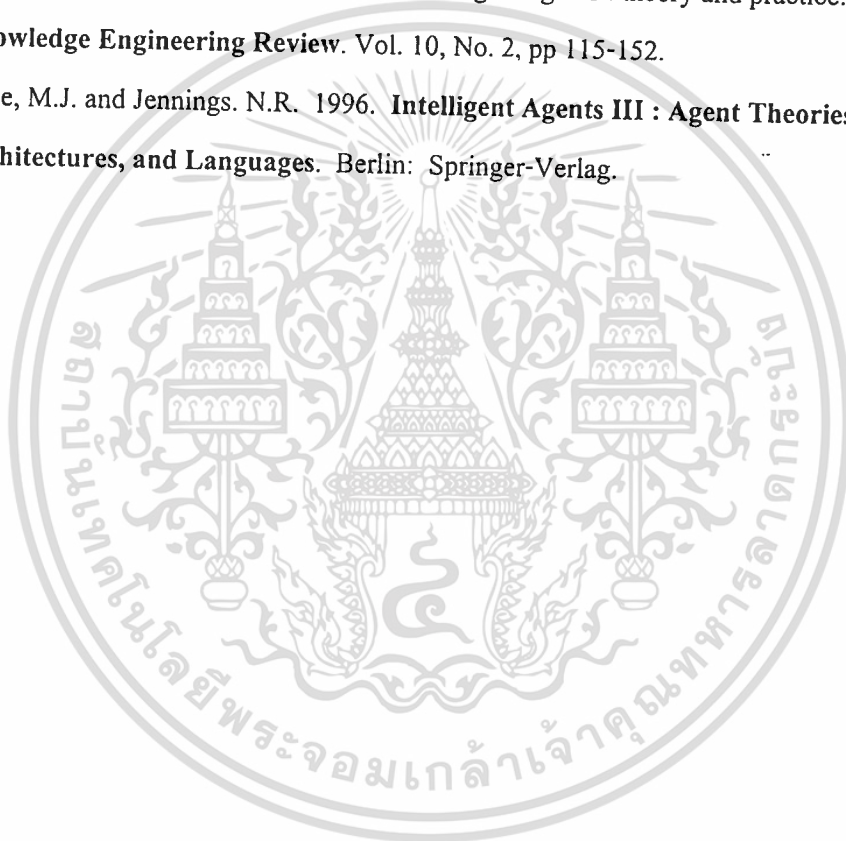
5.2 ข้อจำกัดและข้อเสนอแนะ

จากที่กล่าวไปแล้วว่า เอเจนต์เป็นเสมือนตัวแทน ที่มีการทำงานอย่างเป็นอิสระในการจัดการกับงานที่เป็นประโยชน์ต่อผู้ใช้ อีกทั้งยังสามารถปฏิบัติงานเพื่อให้ทำงานสำเร็จตรงตามวัตถุประสงค์ ดังนั้นในการพัฒนาเอเจนต์จึงค่อนข้างมีความยุ่งยากในการที่จะทำความเข้าใจทั้งหมด ดังนั้นจึงควรศึกษาข้อมูลของอินเทลลิเจนต์และโค้ด program ต่าง ๆ ที่มีให้บริการตามอินเทอร์เน็ต แต่โดยลักษณะแล้วมักเป็นแบบ Low Level Module จากนั้นจึงต้องนำโค้ดมาพัฒนาให้เป็น High Level Module มากขึ้นเพื่อให้แอปพลิเคชันมีความยืดหยุ่นมากขึ้น ซึ่งในส่วนของอัลกอริทึมนั้น เราสามารถที่จะนำเอาหลักการการทำงานในลักษณะของปัญญาประดิษฐ์ (Artificial Intelligent) มาช่วยในการพัฒนาโดยทำให้เอเจนต์มีความสามารถในการแก้ไขฐานความรู้และเรียนรู้ได้ด้วยตัวเอง ซึ่งเป็นส่วนช่วยให้เอเจนต์ที่พัฒนามีความฉลาดเพิ่มมากขึ้น ตัวอย่างของหลักการที่สามารถนำมาประยุกต์ใช้ในการพัฒนา เช่น Fuzzy Logic , Neural Network เป็นต้น

บรรณานุกรม

- AgentBuilding. 2001. **Agentbuilding**. [online]. Available :
<http://www.agentbuilder.com/Documentation/brochures/ABFamilyBrochure.pdf>.
- Amit. et al. 1997. **Application Maintenance Using Software Agents**. [online]. Available :
<http://fp.ieeeexplore.ieee.org/iel5/7667/20963/00972666.pdf?isNumber=20963&prod=CNF&arnumber=972666&arSt=55&ared=64arAuthor=Sharma%3B+Capretz%2C+M.M.M.>
- CADIP Symposium. 1999. **Agent Monitoring Agent**. [online]. Available :
<http://www.csee.umbc.edu/cadip/1999Symposium/AMACADIP.doc>.
- Cheong, Fah-Chun. 1996. **Internet Agents : Spiders, Wanderers, Brokers, and Bots**. Indianapolis, IN: New Riders.
- Chorafas, Dimitris N. 1998. **Agent Technology Handbook**. New York, NY: McGraw-Hill.
- Cisco Systems. 2000. **Using Alert Monitor**. [online]. Available :
http://www.cisco.com/univercd/cc/doc/product/rtrmgmt/sw_ntman/cwsi2/cwsiug2/ustd/30td.htm.
- Finin, Tim. 1996. **Agents Communication Languages**. [online]. Available :
<http://agents.umbc.edu/technology/acl.shtml>.
- Gheorghe, Tecuci. 1998. **Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies**. San Diego, CA: Academic Press.
- Labrou. et al. 1997. **What is KQML?**. [online]. Available :
<http://www.cs.umbc.edu/kqml/whats-kqml.html>.
- Lange. et al. 1998. **Programming and Deploying Java Mobile Agents with Aglets**. Reading, MA: Addison-Wesley.
- Lee, K.W. Joseph. et al. 1997. "Intelligent Agents for Matching Information Providers and Consumers on the World-Wide-Web" **IEEE Intelligent Agents**. Vol.1, No.1. pp.189-199.
- Murugesan, San. 1998. "Intelligent agents on the Internet and Web." **IEEE Intelligent Agents**. Vol.1, No.1, pp.97-102.

- Pallmann, David. 1999. **Programming Bots, Spiders, and Intelligent Agents in Microsoft Visual C++**. Washington, DC: Microsoft Press.
- Wang, Dan. et al. 2000. **Illustrates Computer Dictionary for Dummies**. Foster City, CA: IDG Books Worldwide.
- Watson, Mark. 1997. **Intelligent Java Applications for the Internet and Intranets**. San Francisco, CA: Morgan Kaufmann.
- Wooldridge, M.J. and Jennings. N.R. 1995. "Intelligent agents : theory and practice." **Knowledge Engineering Review**. Vol. 10, No. 2, pp 115-152.
- Wooldridge, M.J. and Jennings. N.R. 1996. **Intelligent Agents III : Agent Theories, Architectures, and Languages**. Berlin: Springer-Verlag.



ประวัติผู้เขียน

ชื่อ นางสาว วริศรา กิรตศิริกุล
 ภูมิลำเนา เลขที่ 5/3 หมู่ 4 ถนนรังสิต-นครนายก ต.บึงนารางค์
 อ.ธัญบุรี จ.ปทุมธานี 12110
 วุฒิกการศึกษา วท.บ. (วิทยาการคอมพิวเตอร์) คณะวิศวกรรมและเทคโนโลยีการเกษตร
 สถาบันเทคโนโลยีราชมงคล
 อีเมล waritsara_k@hotmail.com , varisara_k@yahoo.com



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้