

การพัฒนาโปรแกรมประยุกต์เชิงวัตถุโดยใช้ภาษาจาวา
Object Oriented Application using Java



โดย
นายภูมินทร์ หลิมอมรรัตน์
นายวงวิห อัครบัณฑิตสกุล

อาจารย์ที่ปรึกษา
ดร. วรวัฒน์ ตีมโกคา

ปฏิญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาโปรแกรมประยุกต์เชิงวัตถุโดยใช้ภาษาจาวา

Object Oriented Application using Java

ผู้จัดทำ

1. นาย ภูมิินทร์ หลิมอมรรัตน์ รหัสประจำตัว 39014394
2. นาย วงวิช อัครบัณฑิตสกุล รหัสประจำตัว 39014442



Sh.

อาจารย์ที่ปรึกษา

(ดร. วรวัฒน์ ลิ้มโกศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมประยุกต์เชิงวัตถุโดยใช้ภาษาจาวา

นาย ภูมินทร์ หลิมอมรรรัตน์ 39014394

นาย วงวิช อัครบัณฑิตสกุล 39014442

ดร. วรวัฒน์ ลิ้มโกภา อาจารย์ที่ปรึกษา

ปีการศึกษา 2542

บทคัดย่อ

ภาษาจาวา (Java) เป็นภาษาหนึ่งที่มีลักษณะการ โปรแกรมเป็นแบบเชิงวัตถุ (Object Oriented Programming, OOP) ซึ่งจาวามีจุดเด่นมากตรงที่เราทำการเขียนโปรแกรมขึ้นมาเพียงครั้งเดียว แต่สามารถนำไปใช้งานได้ในทุกระบบปฏิบัติการดังคำกล่าวที่ว่า “Write once run anywhere” และ จาวายังประกอบด้วยส่วนประกอบที่ใช้ทำงานในระบบคอมพิวเตอร์สมัยใหม่ เช่น ส่วนของกราฟฟิกและระบบการติดต่อผู้ใช้ด้วยกราฟฟิก(GUI) ส่วนประกอบของระบบเครือข่าย และ ระบบติดต่อกับฐานข้อมูล ด้วยเหตุนี้เองในปัจจุบันจึงมีคนนิยมใช้ภาษาจาวามากขึ้นเรื่อยๆ ทั่วโลก

ภาษาจาวาเป็นเครื่องมือที่ยอดเยี่ยมมาก สำหรับใช้งานในอินเทอร์เน็ต จาวามีฟังก์ชันการทำงานเกี่ยวกับระบบเครือข่ายที่ทำงานได้ดี การนำมาประยุกต์ใช้งานในระบบเครือข่ายจึงเป็นเรื่องง่าย เช่น การเขียนแอปพลิเคชันเพื่อใช้ในการบริหารเครือข่ายโดยผ่านทาง SNMP (Simple Network Management Protocol)

SNMP เป็นมาตรฐานที่นิยมใช้กันในระบบบริหารเครือข่าย ซึ่งเมื่อใช้ร่วมกับ Java Servlet จะช่วยให้สามารถใช้งาน SNMP ผ่าน Web Browser ได้ ส่งผลให้ใช้งานได้สะดวกและเพิ่มประสิทธิภาพในการทำงาน

Object Oriented Programming by Java

Pumin Limamonrat

Wongwich Akarabunditskul

Dr. Worawat Limpoka Advisor

Abstract

Java language is an Object Oriented Programming language. Its main advantage are portability and hardware independency as the saying “Write once run anywhere”. Java has many components that use in the new computer programming such as Graphic and Graphic User Interface (GUI) component, Network component and Database component. Many programmers in the world use Java.

Java is a perfect tool for use in Internet. Java has many APIs which are used for Networking. Java is easy for implementing Network Management. The Network Management uses the popular protocol - “SNMP”.

When SNMP(Simple Network Management Protocol) is used in Network Management together with Java Servlet , it enables to implement SNMP through web browser , increasing its facilities and efficiency in the operation.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้ คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและความร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึง เพราะเป็นส่วนสำคัญที่ทำให้ปริญญาบัตรนี้เสร็จลงได้ ก็คือ ท่านอาจารย์ ดร. วรวัฒน์ ลิ้มโกทา อาจารย์ที่ปรึกษาปริญญาบัตร ที่ให้ความเอาใจใส่ แนะนำและช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก ขอขอบพระคุณอาจารย์ทุกๆ ท่านที่ได้ช่วยประสิทธิประสาทวิชาความรู้ทั้งหลายให้กับข้าพเจ้าได้นำมาใช้ในปัจจุบัน รวมถึงภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้แหล่งข้อมูลและบริการเครื่องมือเครื่องใช้ จนสามารถทำงานสำเร็จลงได้ด้วยความช่วยเหลือจากหลายๆ ฝ่ายที่กล่าวมา จนปริญญาบัตรเล่มนี้สำเร็จลงได้ ข้าพเจ้าจะรำลึกบุญคุณครั้งนี้ไว้ตลอดไป

ภูมินทร์ หลิมอมรรัตน์
วงวิษ อัครบัณฑิตสกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้าที่

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพและตาราง	VI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินการ	2
บทที่ 2 ระบบเครือข่าย TCP/IP	4
2.1 เน็ตเวิร์คแอ็กเซสเลเยอร์ (Network Access Layer)	4
2.2 อินเทอร์เน็ตเลเยอร์ (Internet Layer)	5
2.3 ทรานสปอร์ตเลเยอร์ (Transport Layer)	6
2.4 แอปพลิเคชันเลเยอร์ (Application Layer)	8
บทที่ 3 การบริหารเครือข่าย (Network Management)	10
3.1 ฟังก์ชันการบริหารเครือข่ายของ OSI (OSI Management Functional Areas)	10
3.2 สถาปัตยกรรมการบริหารเครือข่าย (Network Management Architecture)	11
3.3 เมเนจเมนต์อินฟอร์เมชันเบส (Management Information Base , MIB)	13
3.4 SNMP (Simple Network Management Protocol)	16
3.5 Remote Network Monitoring (RMON)	22
บทที่ 4 ภาษาจาวา (Java)	24
4.1 การเขียนโปรแกรมแบบอ้างอิงเชิงวัตถุ (Object Oriented Programming)	24
4.2 จาวาแอปพลิเคชัน (Java Application)	30
4.3 จาวาแอปเพล็ต (Java Applet)	33
4.4 จาวาเน็ตเวิร์ค (Java Network)	39
4.5 จาวาเซิร์ฟเล็ต (Java Servlet)	42
4.6 การเชื่อมต่อกับฐานข้อมูล (Database) ด้วยภาษาจาวา	48
บทที่ 5 การออกแบบและการสร้างโปรแกรมในโครงการส่วนแรก	54
5.1 UML (Unified Modeling Language)	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2	การออกแบบฐานข้อมูล (Database Design)	58
5.3	การสร้างโปรแกรมและการทำงาน	60
5.4	ตัวอย่างผลการทำงาน	66
5.5	สรุปมิบของอุปกรณ์	73
บทที่ 6	เอสเอ็นเอ็มพีเอเจนต์ (SNMP Agent)	75
6.1	ส่วนประกอบของ เอสเอ็นเอ็มพีเอเจนต์	75
6.2	ขั้นตอนการทำงานของ เอสเอ็นเอ็มพีเอเจนต์	76
6.3	การพัฒนาเอสเอ็นเอ็มพีเอเจนต์โดยใช้ภาษาจาวา	78
6.4	ตัวอย่าง เอสเอ็นเอ็มพีเอเจนต์	79
บทที่ 7	บทวิจารณ์และสรุปผล	82
7.1	สรุปผลการดำเนินงาน	82
7.2	แนวทางการพัฒนาต่อ	83
บทที่ 8	การติดตั้งองค์ประกอบของโครงการ	84
8.1	ส่วนของ การบริหารเครือข่าย (Network Management)	84
8.2	ส่วนของ เอสเอ็นเอ็มพีเอเจนต์ (SNMP Agent)	92
8.3	ความต้องการของระบบ (System Requirement)	93
ภาคผนวก ก		94
ภาคผนวก ข		108
หนังสืออ้างอิง		111

สารบัญญภาพและตาราง

รูปที่	หน้าที่
2-1 รูปแบบของยูดีพีเมสเสจ (UDP Message Format)	7
2-2 รูปแบบของทีซีพีเซกเมนต์ (TCP Segment Format)	7
2-3 ภาพรวมการทำงานของ TCP/IP ทั้ง 4 เลเยอร์	9
3-1 ลักษณะของ SNMP	13
3-2 บางส่วนของมิบ	14
3-3 เอสเอ็นเอ็มพีเมสเสจ	16
3-4 เก็ตรีควีสเมสเสจ (GetRequest Message)	17
3-5 เก็ตเนกซ์ทรีควีสเมสเสจ (GetNextRequest Message)	17
3-6 เซ็ตรีควีสเมสเสจ (SetRequest Message)	18
3-7 แทรปเมสเสจ (Trap Message)	19
3-8 รูปแบบพีดียูทั่วไป (Common PDU Format)	19
3-9 รูปแบบของ Variable Bindings List	21
3-10 รูปแบบของ แทรปพีดียู	21
3-11 ตัวอย่างการใช้ RMON	22
3-12 RMON MIB	23
4-1 แสดงการส่งข้อความ (message) จากวัตถุ A ให้กับวัตถุ B	26
4-2 แสดงการส่งข้อความ (message) จากวัตถุ A ให้กับวัตถุ B โดยมีพารามิเตอร์(parameter) ด้วย	27
4-3 แสดงการเรียงลำดับของคลาส โดยมี คลาสย่อย (subclass) ของ คลาสหลัก (superclass)	28
4-4 แสดงตำแหน่งของตัวแปรกับเมธอดในลักษณะของ cell	29
4-5 แสดงการเรียกใช้โปรแกรมผ่านหน้าต่าง DOS	31
4-6 ผลลัพธ์จากรูปที่ 4-5	31
4-7 แสดงการทำงานของแอปเพล็ต	33
4-8 แสดงวงจรชีวิตของแอปเพล็ต	36
4-9 แสดงตัวอย่างของแอปเพล็ต	37
4-10 แสดงของห่อข้อมูลแบบ UDP	42
4-11 แสดงการทำงานของเซิร์ฟเล็ต	43
4-12 แสดงโมดูลการเชื่อมต่อกับ JDBC อย่างง่าย	48
4-13 แสดงการเชื่อมต่อแบบ Thin-client	49
4-14 แสดงการเชื่อมต่อผ่าน JDBC-ODBC Bridge	50
4-15 แสดงการเชื่อมต่อผ่าน DBMS Driver	50
5-1 สัญลักษณ์ของเอสไอซีไอเอ็น	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5-2	สัญลักษณ์ของการถ่ายทอด	56
5-3	สัญลักษณ์ของอะกรีเกชัน	56
5-4	คลาสไดอะแกรมของส่วนที่เป็น จาวาแอปพลิเคชัน	57
5-5	Network Management Console ซึ่งเกิดจากการรัน ส่วนจาวาแอปพลิเคชัน	66
5-6	สถานะการทำงานของ แอปพลิเคชัน ในขณะที่ทำการเก็บข้อมูลจากมิบ ของอุปกรณ์	66
5-7	สถานะของอุปกรณ์โดยรวม	67
5-8	หน้าจอ Direct Query	68
5-9	ผลลัพธ์ที่ได้จากการเรียกค้น (Query)	69
5-10	ข้อมูลของ 161.246.6.70 ส่วนบน	70
5-11	ข้อมูลของ 161.246.6.70 ส่วนล่าง	70
5-12	สถิติการเปิดปิดอุปกรณ์ ไอพี 161.246.6.70	71
5-13	สถิติของจำนวนข้อมูลที่เข้า และ จำนวนข้อมูลที่ออก ในแต่ละอินเทอร์เน็ต	72
6-1	ขั้นตอนการทำงานของเอสเอ็นเอ็มพีเอเจนท์	77
6-2	การทดสอบเอเจนท์ด้วย HP Openview	81
6-3	การทดสอบเอเจนท์ผ่านโปรแกรม snmpServlet	81
8-1	การใช้ Sysedit ในการแก้ไขแฟ้มระบบ	85
8-2	การแก้ไขตัวแปรระบบใน Windows NT	86
8-3	การทำงานของโปรแกรมก่อนและหลังปรับแต่งฟอนต์	88
8-4	ส่วนที่ต้องแก้ไขใน font.properties.th	88
8-5	หน้าจอเริ่มต้น	89
8-6	หน้าจอ Administrator	90
8-7	หน้าจอการเพิ่ม Servlets Directory	90
8-8	หมายเลขพอร์ตที่ใช้	91
8-9	การปรับแต่ง Jrun Web Server	91
8-10	การปรับแต่งค่าพอร์ต	92

ตารางที่

3-1	คำเงื่อนไขความผิดพลาด (error condition) แบบต่างๆ	20
-----	--	----

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันระบบเครือข่ายคอมพิวเตอร์ ได้ถูกนำมาใช้กันอย่างกว้างขวาง และนับวันแนวโน้มของเครือข่ายขนาดใหญ่ ก็มีมากขึ้น การบริหารเครือข่ายที่ซับซ้อน ใหญ่โต ซึ่งประกอบไปด้วยอุปกรณ์หลากหลาย เป็นไปด้วยความยากลำบากและมีต้นทุนสูง จึงทำให้มีการพัฒนาโปรโตคอลการบริหารระบบเครือข่าย (Network Management Protocol) ขึ้น

SNMP (Simple Network Management Protocol) เป็นโปรโตคอลการบริหารระบบเครือข่ายหนึ่งที่ถูกพัฒนาขึ้น เพื่อให้การบริหารเครือข่ายคอมพิวเตอร์เป็นไปได้อย่างง่ายขึ้น โดย SNMP นั้น จะเป็นมาตรฐานสำหรับ TCP/IP ซึ่งใช้งานกันอย่างแพร่หลาย และนอกจากนั้นแล้ว ใน SNMP ก็ยังมีส่วนเพิ่มเติมที่สำคัญ นั่นก็คือ การมอนิเตอร์เครือข่ายระยะไกล (Remote Network Monitoring ,RMON) ซึ่งช่วยขยายขีดความสามารถของ SNMP ในการบริหารเครือข่ายส่วนท้องถิ่น (Local-Area Network ,LAN) อีกด้วย

สำหรับโมเดลของการบริหารเครือข่าย ซึ่งถูกใช้ใน การบริหารเครือข่าย TCP/IP (TCP/IP network management) นั้น ประกอบไปด้วยส่วนหลักๆ ดังต่อไปนี้

- เมเนจเมนต์สเตชัน (Management Station)
เป็นเครื่องที่มีแอปพลิเคชัน ซึ่งช่วยให้ผู้บริหารเครือข่าย สามารถทำการบริหารระบบเครือข่ายได้
- เมเนจเมนต์เอเจนต์ (Management Agent)
อุปกรณ์ต่างๆ เช่น โฮสต์ (host) เราเตอร์ (router) ฮับ (hub) ภายในระบบเครือข่าย ที่ต้องการจะบริหาร จะต้องมีารติดตั้งซอฟต์แวร์เอเจนต์ (Agent software) ลงไปก่อน
- เมเนจเมนต์อินฟอร์เมชันเบส (Management Information Base, MIB)
ข้อมูลต่างๆ ที่เกี่ยวข้องกับอุปกรณ์ จะถูกเก็บไว้ใน เมเนจเมนต์อินฟอร์เมชันเบส (MIB)
- โปรโตคอลการบริหารเครือข่าย (Network Management Protocol)

เป็นตัวเชื่อมระหว่างเมเนจเมนต์สเตชัน และ เมเนจเมนต์เอเจนต์ เช่น SNMP

หลักการทำงานโดยทั่วไป ที่เกิดขึ้นส่วนใหญ่นั้น ก็คือ เมเนจเมนต์สเตชัน จะส่งสัญญาณร้องขอไปยัง เมเนจเมนต์เอเจนต์ เพื่อกระทำการกับ เมเนจเมนต์อินฟอร์เมชันเบสที่ ภายในตัวอุปกรณ์ เช่น การอ่านค่าข้อมูล เป็นต้น

สำหรับโครงการที่ได้จัดทำขึ้นนี้ ในส่วนแรก จะช่วยให้ผู้ใช้สามารถที่จะดูข้อมูลต่างๆ ที่เกี่ยวข้องกับการบริหารเครือข่ายได้ โดยผ่านทางเว็บเพจ บนเครือข่ายอินเทอร์เน็ต ซึ่งในปัจจุบันเครือข่ายอินเทอร์เน็ตได้เติบโตอย่างรวดเร็ว กว้างขวาง ไปทั่วโลก ประโยชน์จากตรงนี้จะทำให้ผู้บริหารสามารถที่จะดูเครือข่ายของตน จากต่างที่ เช่น ต่างประเทศ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโครงการในส่วนที่ 2 นั้น จะเน้นพัฒนาไปที่ เอสเอ็นเอ็มพีเอเจนต์ (SNMP Agent) โดยจะทำการเขียนซอฟต์แวร์ ที่เป็นเอสเอ็นเอ็มพีเอเจนต์ ขึ้นใช้เอง

โดยทั้งสองส่วนของโครงการที่กล่าวไปนั้น จะใช้ภาษาจาวา (Java) ในการอิมพลีเมนต์ (implement) เนื่องจาก ภาษาจาวา เป็นหนึ่งในไม่กี่ภาษาที่มีการกำหนดให้มีชุดคำสั่งการเชื่อมต่อเครือข่าย ผ่านซ็อกเก็ต (Socket) ให้เป็นมาตรฐานแทนการใช้คอมโพเนนต์ (component) ภายนอก จึงทำให้การเขียนโปรแกรมเป็นไปอย่างสะดวกรวดเร็วและมีข้อผิดพลาดน้อยกว่า เมื่อเทียบกับภาษาอื่น นอกจากนี้แล้ว ภาษาจาวายังมีลักษณะที่เป็นอ็อบเจกต์โอเรียนเทด (Object-Oriented) อย่างแท้จริง และช่วยให้สามารถทำงานได้ในหลายๆ ระบบที่ต่างกันอีกด้วย

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อให้สามารถวิเคราะห์ ออกแบบ และพัฒนาระบบงาน ตามแนวคิดของอ็อบเจกต์ (Object)
- 1.2.2 เพื่อให้สามารถตรวจสอบข้อมูลที่เกี่ยวข้องกับการบริหารเครือข่าย ผ่านทางเว็บเพจ บนเครือข่ายอินเทอร์เน็ต ได้
- 1.2.3 เพื่อให้สามารถพัฒนาเอสเอ็นเอ็มพีเอเจนต์ ใช้เอง ได้ในระดับหนึ่ง

1.3 ขอบเขตของโครงการ

โครงการนี้ แบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ ในส่วนแรก จะเป็นโครงการบริหารเครือข่าย โดยใช้โปรโตคอล SNMP (Simple Network Management Protocol) เพื่อดึงข้อมูลจากเมเนจเมนต์อินฟอร์เมชันเบส 2 (Management Information Base 2, MIB-2) บนอุปกรณ์ต่างๆ ที่ต้องการบริหาร เป็นหลัก สำหรับการดึงข้อมูลที่ว่านี้ จะใช้ภาษาจาวา ซึ่งเขียนในลักษณะของอ็อบเจกต์ (Object) ดึงข้อมูลมาเก็บในฐานข้อมูลบนเครื่องเซิร์ฟเวอร์ (Server) จากนั้น จะทำการเขียนโปรแกรมติดต่อกับฐานข้อมูล เพื่อทำการแสดงผลในลักษณะต่างๆ บนเว็บเพจ โดยโปรแกรมที่ว่านี้ จะใช้จาวาเซิร์ฟเล็ต (Java Servlet) เป็นหลักในการเขียน สำหรับโครงการส่วนที่ 2 นั้น จะเป็นการพัฒนาเอสเอ็นเอ็มพีเอเจนต์ บนเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer, PC) โดยในส่วนนี้ ก็จะใช้ภาษาจาวาในการเขียน เช่นกัน ซึ่งในการพัฒนาเอสเอ็นเอ็มพีเอเจนต์นี้ จะทำการพัฒนาเพียงบางกลุ่มของเมเนจเมนต์อินฟอร์เมชันเบส 2 (MIB-2)

1.4 วิธีการดำเนินการ

โครงการนี้ เริ่มต้นด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับโครงการ ซึ่งมีเรื่องต่างๆ ดังนี้ ระบบเครือข่าย TCP/IP ซึ่งมีรายละเอียดดังในบทที่ 2 ความรู้ที่เกี่ยวข้องกับการบริหารเครือข่ายต่างๆ ซึ่งประกอบไปด้วย ฟังก์ชันการบริหารเครือข่ายของ OSI (OSI Management Functional Areas) สถาปัตยกรรมการบริหารเครือข่าย (Network Management Architecture) เมเนจเมนต์อินฟอร์เมชันเบส 2 (MIB) SNMP (Simple Network Management Protocol) และ Remote Network Monitoring (RMON) ซึ่งรายละเอียดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมดจะอยู่ในบทที่ 3 ความรู้ที่เกี่ยวข้องกับภาษาจาวา ซึ่งประกอบไปด้วย การโปรแกรมแบบอ้างอิงเชิงวัตถุ จาวาแอปพลิเคชัน (Java Application) จาวาแอปเพล็ต (Java Applet) จาวาเน็ตเวิร์ค (Java Network) จาวาเซิร์ฟเล็ต (Java Servlet) และการเชื่อมต่อฐานข้อมูล (Database) ด้วยภาษาจาวา ซึ่งรายละเอียดทั้งหมดจะอยู่ในบทที่ 4 ความรู้ต่างๆ ที่เน้นเกี่ยวกับเอสเอ็นเอ็มพีเอเจนต์ (SNMP Agent) ซึ่งประกอบไปด้วยส่วนประกอบของเอสเอ็นเอ็มพีเอเจนต์ และขั้นตอนการทำงานของ เอสเอ็นเอ็มพีเอเจนต์ ซึ่งรายละเอียดทั้งหมดจะอยู่ในบทที่ 6 2 หัวข้อแรก

เมื่อได้ทำการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับโครงการ ไปเรียบร้อยแล้ว ก็มาทำการพัฒนาโปรแกรมของโครงการส่วนแรกโดยใช้ UML ต่อจากนั้นก็ทำการออกแบบฐานข้อมูลที่จะใช้ ตามด้วยการอิมพลีเมนต์โครงการ และสุดท้ายก็ติดตามผลการทำงานที่ได้ ซึ่งรายละเอียดทั้งหมดจะอยู่ในบทที่ 5 สำหรับโครงการส่วนที่ 2 นั้น ก็จะทำการอิมพลีเมนต์ โดยไม่ใช้ฐานข้อมูล ซึ่งรายละเอียดทั้งหมดจะอยู่ในบทที่ 6 2 หัวข้อหลัง



บทที่ 2

ระบบเครือข่าย TCP/IP

ในการศึกษาการทำงานของเครือข่ายใดๆ เราจะเริ่มต้นด้วยการมองการทำงานของมันเป็นชั้นๆ การทำงานทั้งหมดของโพรโทคอลจะประกอบไปด้วยเลเยอร์หลายๆเลเยอร์ที่ซ้อนกันได้ออกมาในรูปแบบที่เรียกว่า Protocol Stack แต่ละชั้นจะมีหน้าที่การทำงานอย่างชัดเจนและไม่เกี่ยวข้องกัน แต่ละชั้นจะรู้เพียงวิธีส่งข้อมูลไปยังชั้นอื่นๆ แต่จะไม่รู้ถึงการทำงานข้างใน แต่ละโพรโทคอลจะมีการแบ่งการทำงานออกเป็นจำนวนชั้นไม่เท่ากัน แต่ก็มีมาตรฐานที่เป็นที่ยอมรับกันทั่วไปเรียกว่า Open System Interconnect (OSI) Reference Model ซึ่งแบ่งการทำงานออกเป็น 7 ชั้น

- ฟิสิคัลเลเยอร์ (Physical Layer)
- ดาต้าลิงก์เลเยอร์ (Data Link Layer)
- เน็ตเวิร์กเลเยอร์ (Network Layer)
- ทรานสปอร์ตเลเยอร์ (Transport Layer)
- เซสชันเลเยอร์ (Session Layer)
- 프리เซนเทชันเลเยอร์ (Presentation Layer)
- แอปพลิเคชันเลเยอร์ (Application Layer)

ในแต่ละชั้นจะมีข้อกำหนดในการทำงานที่ต่างกัน แต่ในการศึกษา TCP/IP นั้น เราจะใช้โมเดลใหม่ โดยแบ่งออกได้เป็น 4 ชั้น หลักการทำงานของ TCP/IP นั้น จะแบ่งออกตามระดับชั้นของการทำงาน

- เน็ตเวิร์กแอ็กเซสเลเยอร์ (Network Access Layer)
- อินเทอร์เน็ตเลเยอร์ (Internet Layer)
- โฮสต์ทูโฮสต์ทรานสปอร์ตเลเยอร์ (Host-to-Host Transport Layer)
- แอปพลิเคชันเลเยอร์ (Application Layer)

2.1 เน็ตเวิร์กแอ็กเซสเลเยอร์ (Network Access Layer)

ส่วนนี้จะทำหน้าที่จัดส่งข้อมูลผ่านสายสัญญาณชนิดต่างๆ โดยการทำงานจะเป็นอย่างไรขึ้นอยู่กับสายสัญญาณที่ใช้อยู่ สายแต่ละชนิดจะมีโพรโทคอลที่ใช้ควบคุมการทำงาน แตกต่างกันไป เช่น อีเธอร์เน็ต (Ethernet) เอฟดีดีไอ (FDDI) เฟรมรีเลย์ (Frame Relay) เป็นต้น

งานที่เกิดขึ้นที่ชั้นนี้ คือ การเพิ่มเฮดเดอร์ (Header) เข้าไปในดาต้าแกรม (Datagram) เพื่อให้กลายเป็นเฟรม (Frame) แล้วส่งไปตามเครือข่าย การเปลี่ยนเลขไอพี (ip) ให้กลายเป็นฟิสิคัลแอดเดรส (Physical Address) ของระบบเครือข่าย เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 อินเทอร์เน็ตเลเยอร์ (Internet Layer)

อินเทอร์เน็ตโปรโตคอล (internet protocol) เป็นการทำงานแบบไม่มีการเชื่อมต่อ กล่าวคือจะไม่มีการตรวจสอบการเชื่อมต่อ (Handshaking) กับปลายทางก่อนส่งข้อมูล แต่จะทำการส่งข้อมูลโดยไม่คำนึงถึงว่าปลายทางจะได้รับข้อมูลหรือไม่ และยังไม่มีการตรวจสอบความถูกต้องของข้อมูลที่ส่งไปด้วย เป็นเพียงการส่งไปให้ถึงปลายทางเท่านั้น ในส่วนนี้มีการทำงานที่สำคัญ 2 ส่วน คือ

2.2.1 อินเทอร์เน็ตโปรโตคอล (Internet Protocol) มีหน้าที่สำคัญ คือ

2.2.1.1 การสร้างคำคำแกรม

คำคำแกรมเป็นส่วนประกอบที่เล็กที่สุดของข้อมูลที่อินเทอร์เน็ตโปรโตคอลจะใช้ในการส่ง ถูกออกแบบมาให้ใช้กับเครือข่ายแพ็คเกจสวิตช์ (Packet switch) ซึ่งข้อมูลมักแบ่งออกเป็นหลายๆ คำคำแกรม ที่เป็นอิสระจากกัน โดยมีเฮดเดอร์ (Header) เป็นตัวเก็บรายละเอียดของตัวเอง กำหนดเส้นทางเป็นของตัวเอง ลำดับการไปถึงของคำคำแกรมแต่ละตัวก็ไม่แน่นอน ซึ่งจะเป็นหน้าที่ของอินเทอร์เน็ตโปรโตคอล เครื่องปลายทางที่จะประกอบคำคำแกรมเหล่านั้น ออกมาเป็นข้อมูลที่สมบูรณ์อีกครั้ง

2.2.1.2 กำหนดเส้นทางคำคำแกรม (Routing Datagram)

แต่ละคำคำแกรม จะมีไอพี (ip) ปลายทางกำหนดไว้ ถ้าไอพีเหล่านั้นอยู่ในเครือข่ายเดียวกันจะทำการส่งได้ทันที แต่หากปลายทางอยู่ต่างเครือข่ายกันจะทำการส่งผ่านเกตเวย์ (Gateway) ของเครือข่ายนั้นๆ ซึ่ง เกตเวย์ ก็คือ อุปกรณ์ที่ใช้เชื่อมโยงเครือข่ายเข้าด้วยกัน

2.2.1.3 การแบ่งเป็นส่วนๆ (Fragmenting)

ในการเดินทางนั้น คำคำแกรม อาจต้องผ่านเครือข่ายหลายๆ ชนิด ซึ่งแต่ละช่วงจะใช้ Network Access Protocol ที่แตกต่างกัน ในแต่ละโปรโตคอลมักจะมี unit data หรือที่เรียกว่า Maximum Transmission Unit (MTU) หรือ คือ ขนาดที่มากที่สุดของข้อมูลที่จะถูกส่งออกไปได้ หากขนาดของข้อมูลใหญ่กว่าขนาดที่กำหนด จะต้องถูกแบ่งข้อมูลเป็นชิ้นเล็กๆ เพื่อให้ส่งออกไปได้

2.2.2 ไอซีเอ็มพี (Internet Control Message Protocol ,ICMP)

เป็นโปรโตคอลที่ทำงานอยู่ในอินเทอร์เน็ตเลเยอร์เช่นกัน ซึ่งทำหน้าที่คอยส่งสัญญาณควบคุมต่างๆ ไปยังเครื่องปลายทาง โดยโปรโตคอลนี้จะอาศัย IP Datagram ในการส่งสัญญาณควบคุมเหล่านั้น คำสั่งควบคุมที่สำคัญของ ICMP คือ

2.2.2.1 Flow control

จะถูกส่งไปยังเครื่องปลายทางหรือเครื่องเกตเวย์ที่อยู่ในเส้นทางเดินของดาต้าแกรม โดยจะถูกส่งไปยังเครื่องต้นทาง เมื่อมีดาต้าแกรมถูกส่งมาเร็วเกินไปจนประมวลผลไม่ทัน ซึ่งจะส่งผลให้เครื่องต้นทางหยุดส่งดาต้าแกรมมาชั่วคราว

2.2.2.2 Detecting unreachable destination

สัญญาณนี้จะถูกส่งกลับไปยังเครื่องต้นทาง เพื่อบอกว่าหาปลายทางไม่พบ

2.2.2.3 Redirecting routes

สัญญาณนี้จะเกิดขึ้นเฉพาะในเครือข่ายที่มีเกตเวย์อยู่มากกว่าหนึ่งตัว โดยเกตเวย์ตัวใดตัวหนึ่งอาจส่งสัญญาณนี้ไปยังเครื่องต้นทาง เพื่อบอกให้มันเปลี่ยนเส้นทางการส่งดาต้าแกรมไปยังเกตเวย์ตัวอื่น โดยอาจมีสาเหตุมาจาก เส้นทางขัดข้อง หรือมีการจราจรหนาแน่นมาก

2.2.2.4 Checking remote hosts

เครื่องๆ หนึ่ง สามารถทำการส่งเอกโคเมสเสจ (Echo Message) ไปยังเครื่องใดๆ เพื่อทดสอบว่าอินเทอร์เน็ตโพรโตคอล ของเครื่องนั้นทำงานอยู่หรือไม่ เมื่อเครื่องปลายทางได้รับสัญญาณนี้ ก็จะส่งสัญญาณเดิมนั้นกลับมา ตัวอย่างของการใช้เมสเสจนี้ คือ คำสั่ง ping ของยูนิกซ์ ซึ่งเรามักใช้ตรวจสอบเครื่องปลายทางว่าทำงานอยู่หรือไม่

2.3 ทรานสปอร์ตเลเยอร์ (Transport Layer)

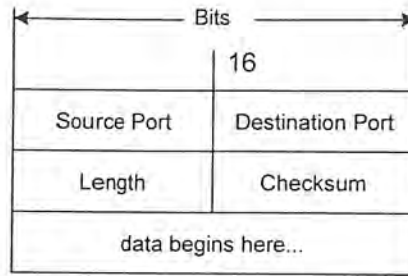
ในเลเยอร์นี้ มีโพรโตคอลที่สำคัญอยู่ 2 โพรโตคอลด้วยกัน คือ ทีซีพี (Transmission Control Protocol ,TCP) และ ยูดีพี (User Datagram Protocol ,UDP) ทั้งสองโพรโตคอลทำหน้าที่เหมือนกับ เป็นตัวเชื่อมต่อการส่งข้อมูลระหว่างแอปพลิเคชันเลเยอร์ กับ อินเทอร์เน็ตเลเยอร์ ถึงแม้จะทำหน้าที่เดียวกัน แต่ก็มีรายละเอียดการทำงานบางอย่างที่แตกต่างกัน คือ ทีซีพี จะมีการตรวจสอบและแก้ไขความผิดพลาดของข้อมูลที่รับเข้ามา ในขณะที่ยูดีพีไม่มี ทำให้ทีซีพีมีความน่าเชื่อถือมากกว่ายูดีพี แต่ในขณะที่เดียวกันยูดีพี เป็นการส่งข้อมูลที่มีโอเวอร์เฮด (Overhead) น้อยกว่าทีซีพี

2.3.1 ยูดีพี (User Datagram Protocol ,UDP)

เป็นโพรโตคอลแบบคอนเนกชันเลส (Connectionless) คือ ไม่มีการทำแฮนด์เชค (handshake) กับเครื่องปลายทาง และไม่มีการตรวจสอบความถูกต้องของข้อมูลที่รับมา ทำให้มีความเร็วในการทำงาน (ในเลเยอร์นี้) สูง เหมาะที่จะใช้กับงานที่ส่งข้อมูลขนาดเล็กและส่งบ่อย ๆ

เฮดเดอร์ (header) ของยูดีพี ได้แสดงไว้ในรูปที่ 2-1 ซึ่งจะเห็นได้ว่า มีการส่งหมายเลขพอร์ตต้นทางและปลายทางไปด้วย

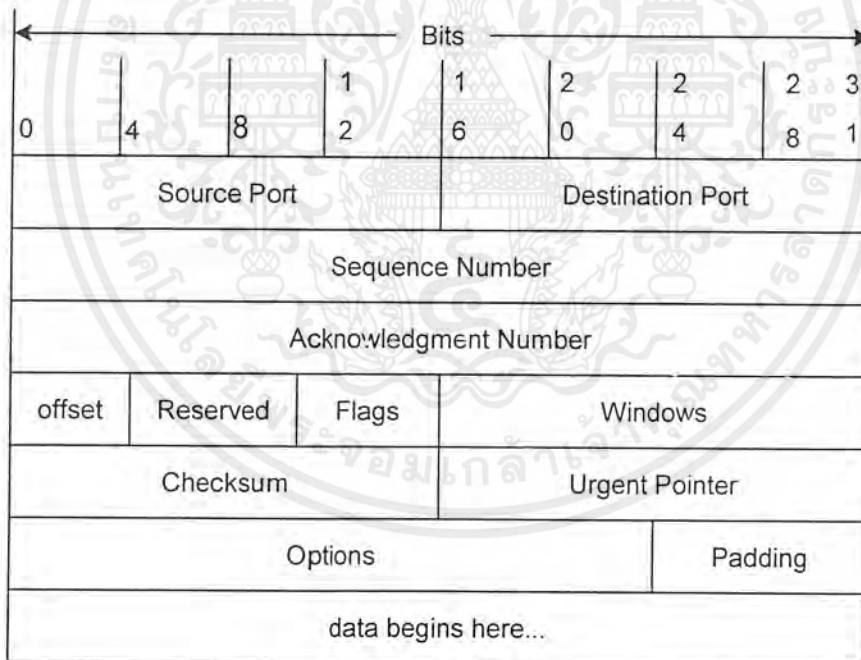
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 รูปแบบของยูดีพีเมสเสจ (UDP Message Format)

2.3.2 ทีซีพี (Transmission Control Protocol ,TCP)

เป็นโปรโตคอลแบบ คอนเนกชัน โอเรียนเทด (Connection-Oriented) และมีการตรวจสอบความถูกต้องของข้อมูลที่ได้รับด้วย ทำให้เป็นการส่งที่มีความน่าเชื่อถือ เขตเตอร์ของทีซีพีได้แสดงไว้ในรูปที่ 2-2



รูปที่ 2-2 รูปแบบของทีซีพีเซกเมนต์ (TCP Segment Format)

การส่งข้อมูลของทีซีพีนั้น จะส่งทีละเซกเมนต์ (segment) (อ้างจากรูปที่ 2-2 เราเรียก unit data ในขั้นนี้ว่า เซกเมนต์) โดยเครื่องปลายทางจะส่งสัญญาณตอบรับกลับมายังเครื่องต้นทาง สำหรับทุกๆ เซกเมนต์ที่มันได้รับและตรวจสอบแล้วไม่พบข้อผิดพลาด ถ้าหากว่า เครื่องต้นทางไม่ได้รับสัญญาณเอกสารนี้เป็นเอกสารที่ส่งวนไวดำเนินการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบรับกลับมาสำหรับเซกเมนต์ มันก็จะสันนิษฐานว่าเซกเมนต์นั้นมีปัญหาและทำการส่งเซกเมนต์นั้นไปอีกครั้ง วิธีการเราเรียกว่า Positive Acknowledment with Re-transmission (PAR)

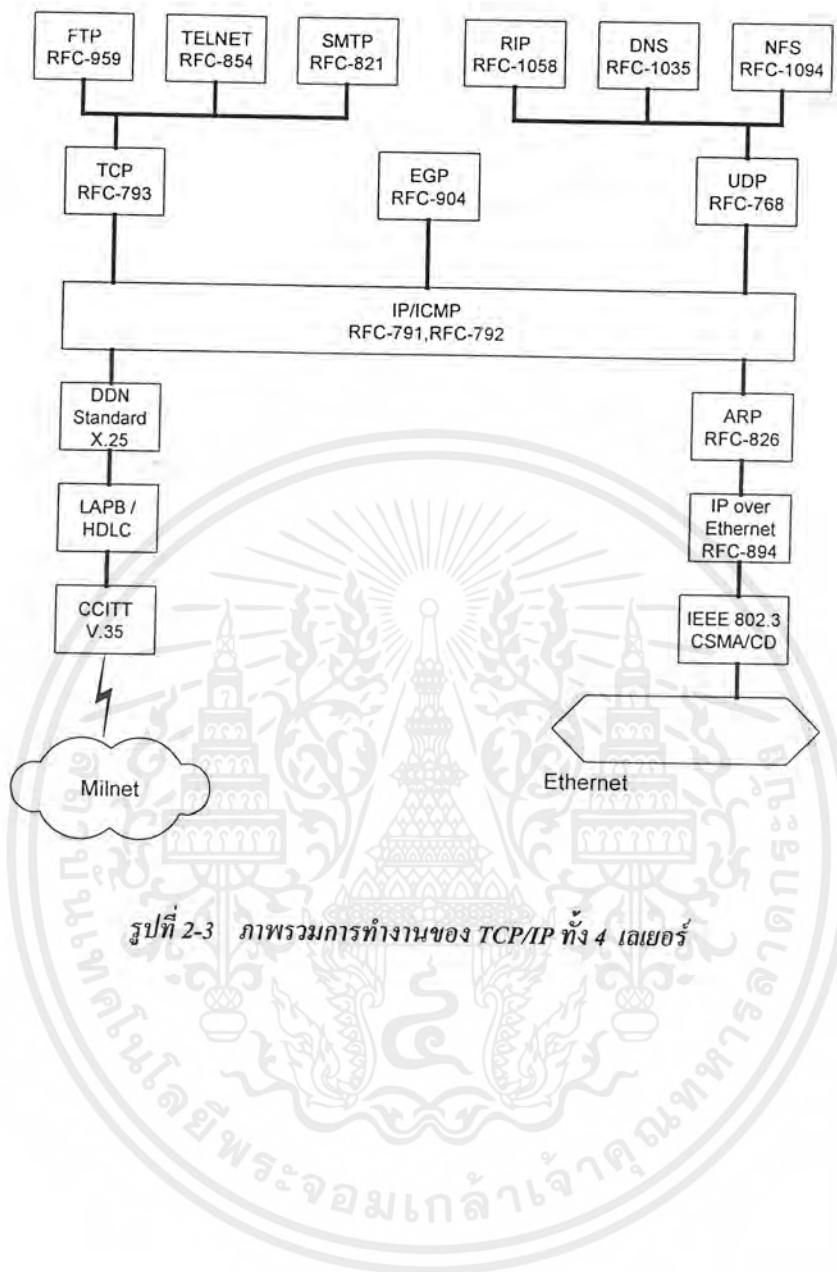
การที่บอกว่า ทีซีพีเป็นการส่งข้อมูลแบบคอนเนกชันโอเรียนเทคชั่น แสดงว่ามันจะต้องมีการส่งสัญญาณแฮนด์เชก (handshake) ระหว่างเครื่องต้นทางและเครื่องปลายทาง เพื่อให้แน่ใจว่าสามารถติดต่อถึงกันได้ แล้วเริ่มส่งข้อมูล วิธีการแฮนด์เชกที่ทีซีพีใช้นั้น เรียกว่า ทรีเวย์แฮนด์เชก (three-way handshake) โดยเมื่อเครื่องต้นทางต้องการส่งข้อมูล มันจะส่งสัญญาณซิงโครไนซ์ (Synchronize ,SYN) ไปยังเครื่องปลายทาง (สัญญาณนี้จะถูกส่งในรูปแบบของเซกเมนต์ธรรมดา) เมื่อเครื่องปลายทางได้รับสัญญาณ SYN แล้ว มันก็จะส่งสัญญาณแอกโนเลดเมนต์ (Acknowledment ,ACK) กลับมา เมื่อ ACK กลับไปถึงเครื่องต้นทาง ก็ถือว่าเส้นทางเชื่อมต่อได้เกิดขึ้นแล้ว เครื่องต้นทางจะส่ง ACK กลับไปให้เครื่องปลายทางพร้อมกับเริ่มต้นการส่งข้อมูล

2.4 แอปพลิเคชันเลเยอร์ (Application Layer)

เลเยอร์นี้ เป็นเลเยอร์บนสุด ซึ่งประกอบไปด้วยหลายโปรโตคอล (เรามักเรียกว่า โปรแกรมมากกว่า) ซึ่งนับวันก็จะยังมีเพิ่มมากขึ้น ตัวอย่างของโปรโตคอลที่ใช้กันบ่อยในเลเยอร์นี้ ได้แก่

Telnet	เรียกอีกอย่างหนึ่งว่า เน็ตเวิร์คเทอร์มินัล โปรโตคอล (Network Terminal Protocol) ใช้ในการจำลองหน้าจอผ่านเครือข่าย
FTP	File Transfer Protocol ใช้ในการโอนถ่ายเพิ่มข้อมูล
SMTP	Simple Mail Transfer Protocol ใช้ในการรับ-ส่งอีเมล
โปรโตคอล เหล่านี้ทำงานโดยใช้ ทีซีพี ในทรานสปอร์ตเลเยอร์ ส่วน โปรโตคอลที่ใช้ยูดีพีนั้น	
ได้แก่	
Domain Name Service (DNS)	ใช้ในการแปลงชื่อเครื่องให้เป็นเลขไอพี หรือกลับกัน
Routing Information Protocol (RIP)	ใช้แลกเปลี่ยน routing information ระหว่างอุปกรณ์ต่าง ๆ
Network File System (NFS)	ใช้เพื่อแชร์ข้อมูลในดิสก์ ให้กับหลายๆ เครื่อง

รูปที่ 2-3 แสดงให้เห็นถึงภาพรวมการทำงานของ TCP/IP ทั้ง 4 เลเยอร์ โดยเครื่องๆ นี้ มีการเชื่อมโยงกับเครือข่ายอยู่สองทางคือ อีเทอร์เน็ต (Ethernet) และ มิลเน็ต (Milnet)



รูปที่ 2-3 ภาพรวมการทำงานของ TCP/IP ทั้ง 4 เลเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การบริหารเครือข่าย (Network Management)

3.1 ฟังก์ชันการบริหารเครือข่ายของ OSI (OSI Management Functional Areas)

แบ่งได้ 5 ด้าน ได้แก่

3.1.1 การบริหารทางด้านคอนฟิก (Configuration Management)

การบริหารทางด้านคอนฟิก จะประกอบด้วยหน้าที่ต่างๆ ดังต่อไปนี้

- กำหนดข้อมูลทางด้านคอนฟิก (define configuration information)
- เชื่อม และ ปรับค่าแอททริบิวต์ (attribute)
- กำหนด (define) และปรับปรุงความสัมพันธ์ (relationships) ต่างๆ
- เริ่มต้น (initialize) และ หยุด (terminate) โอเปอเรชัน (Operation) ของเครือข่าย
- กระจาย (distribute) ซอฟต์แวร์
- ตรวจสอบค่าและ ความสัมพันธ์ (relationships) ต่างๆ
- รายงานสถานะทางด้านคอนฟิก (configuration status)

3.1.2 การบริหารทางด้านความผิดพลาด (Fault Management)

จุดประสงค์ของการบริหารทางด้านความผิดพลาด นั้น มีดังต่อไปนี้

- ระบุจุดที่ผิดพลาด ภายในระบบเครือข่าย
- แยกจุดที่ผิดพลาดนั้นออก หากเป็นไปได้
- ทำการปรับคอนฟิกใหม่ (Reconfigure) หรือ การปรับปรุงระบบเครือข่าย ในส่วนที่ยังเหลืออยู่
- แก้ไขจุดที่ผิดพลาด เพื่อให้สามารถใช้งานระบบได้ดังเดิม

นอกจากนั้นแล้ว ในระบบการบริหารทางด้านความผิดพลาดที่ดัดนั้น จะสามารถคาดการณ์ความผิดพลาดที่อาจจะเกิดขึ้นในอนาคตได้ โดยจะใช้ข้อมูลซึ่งพิจารณาจากสถานะต่างๆ ที่มีการเปลี่ยนแปลงไป

3.1.3 การบริหารทางด้านประสิทธิภาพ (Performance Management)

การบริหารทางด้านประสิทธิภาพ เป็นการบริหารเพื่อให้ได้มาซึ่งประสิทธิภาพของระบบเครือข่าย โดยที่ตัวบ่งชี้ประสิทธิภาพ (Performance Indicator) นั้น มีหลายแบบ เช่น

- ในเชิงความพอใจของผู้ใช้ เช่น ความถูกต้อง เวลาตอบสนอง (response time)
 - ในเชิงประสิทธิภาพที่ได้ เมื่อเทียบกับต้นทุน (cost)
- ได้แก่ ทราฟฟิค (throughput) การใช้ประโยชน์ (utilization)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การบริหารทางการเก็บบันทึก (Accounting Management)

การบริหารทางการเก็บบันทึก จะใช้การติดตามร่องรอย (keep track) การใช้ทรัพยากรในระบบเครือข่ายของผู้ใช้ โดยตัวอย่างของทรัพยากร ดังกล่าวแบ่งได้ดังต่อไปนี้

- Communications facilities เช่น สายโทรศัพท์ (leased line)
 - ฮาร์ดแวร์ต่างๆ เช่น เซิร์ฟเวอร์
 - ซอฟต์แวร์และระบบ เช่น แอปพลิเคชันต่างๆ ในเซิร์ฟเวอร์
 - บริการต่างๆ เช่น บริการข้อมูลต่างๆ ที่มีให้แก่ผู้ใช้ ในระบบเครือข่าย
- โดยข้อมูลที่ได้จากการติดตามร่องรอย นี้ จะถูกบันทึกเก็บไว้ เพื่อใช้ประโยชน์ต่างๆ ได้

3.1.5 การบริหารทางด้านความปลอดภัย (Security Management)

หน้าที่ของการบริหารทางด้านความปลอดภัย สามารถแบ่งได้เป็น 3 ประเภท ดังนี้

- ดูแลรักษาข้อมูลทางด้านความปลอดภัย (maintain security information)
 - ตัวอย่างฟังก์ชัน (function) ที่มีความสัมพันธ์กับการดูแลรักษาข้อมูลทางด้านความปลอดภัย เช่น การเก็บบันทึกเหตุการณ์ (event logging) รายงานการบุกรุก (reporting security violation) เป็นต้น
- ควบคุมการเข้าถึงทรัพยากร (control resource-access service)
 - ผู้ใช้อาจมีสิทธิ์ในการเข้าถึงทรัพยากรที่แตกต่างกันไป ดังนั้นจะต้องมีการควบคุมในด้านนี้อย่างชัดเจน
- ควบคุมกระบวนการเข้ารหัส (control the encryption process)
 - ตัวอย่างในฟังก์ชัน นี้ เช่น ระบบจะต้องมีการเข้ารหัส (encrypt) การแลกเปลี่ยนใดๆ ระหว่างเมนเนจเม้นท์สเตชัน (Management station) และ เมนเนจเม้นท์เอเจนต์ (Management Agent) ได้

3.2 สถาปัตยกรรมการบริหารเครือข่าย (Network Management Architecture)

โมเดลของการบริหารเครือข่าย ซึ่งถูกใช้ใน การบริหารเครือข่าย TCP/IP (TCP/IP network management) นั้น ประกอบไปด้วยส่วนหลักๆ ดังต่อไปนี้

3.2.1 เมนเนจเม้นท์สเตชัน (Management Station)

เมนเนจเม้นท์สเตชัน จะทำหน้าที่ในการเป็นส่วนติดต่อระหว่าง ผู้บริหารเครือข่าย (network manager) กับ ระบบการบริหารเครือข่าย (Network Management System) โดยเครื่องที่เป็น เมนเนจเม้นท์สเตชัน นี้ จะมีแอปพลิเคชัน ซึ่งช่วยให้ผู้บริหารเครือข่าย สามารถทำการบริหารระบบเครือข่ายได้

3.2.2 เมเนจเม้นท์เอเจนท์ (Management Agent)

อุปกรณ์ต่างๆ เช่น โฮสต์ (host) บริดจ์ (bridge) เราเตอร์ (router) ฮับ (hub) ภายในระบบเครือข่าย ที่ต้องการจะบริหาร จะต้องมี การติดตั้งซอฟต์แวร์เอเจนท์ (Agent software) ลงไปก่อน โดยเอเจนท์ จะทำหน้าที่รับเมสเสจ (message) ที่ส่งมาจากเมเนจเม้นท์สแตชัน จากนั้น เอเจนท์ ก็จะทำการตอบสนองตามเมสเสจ ที่เมเนจเม้นท์สแตชัน ส่งมา (มีการตรวจสอบความถูกต้อง เช่น คอมมิวนิตีเนม (community name) ก่อน)

นอกจากนั้นแล้ว ในบางกรณีซึ่งเกิดเหตุการณ์สำคัญๆ เช่น ลิงก์ดาวน์ (link down) เอเจนท์ก็สามารถที่จะส่ง แทรปเมสเสจ (trap message) ไปบอกเมเนจเม้นท์สแตชัน โดยที่เมเนจเม้นท์สแตชัน ไม่ต้องร้องขอได้

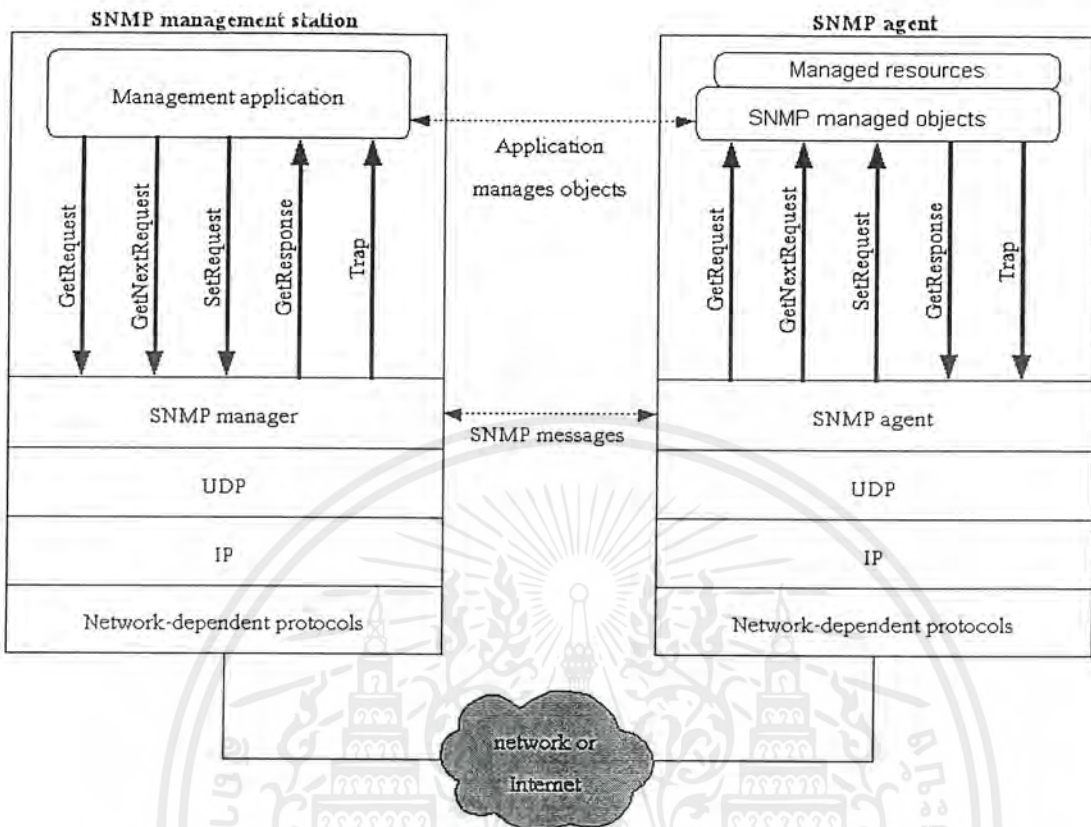
3.2.3 เมเนจเม้นท์อินฟอร์เมชันเบส (Management Information Base, MIB)

ข้อมูลต่างๆ ที่เกี่ยวข้องกับอุปกรณ์ จะถูกเก็บไว้ในเมเนจเม้นท์อินฟอร์เมชันเบส ซึ่งมีลักษณะคล้ายฐานข้อมูล (ต้องมีการติดตั้ง เอเจนท์ ก่อน)

3.2.4 โพรโตคอลการบริหารเครือข่าย (Network Management Protocol)

โพรโตคอลการบริหารเครือข่าย จะเป็นตัวเชื่อมระหว่างเมเนจเม้นท์สแตชัน และเมเนจเม้นท์เอเจนท์ ตัวอย่างของโพรโตคอลการบริหารเครือข่าย เช่น SNMP เป็นต้น

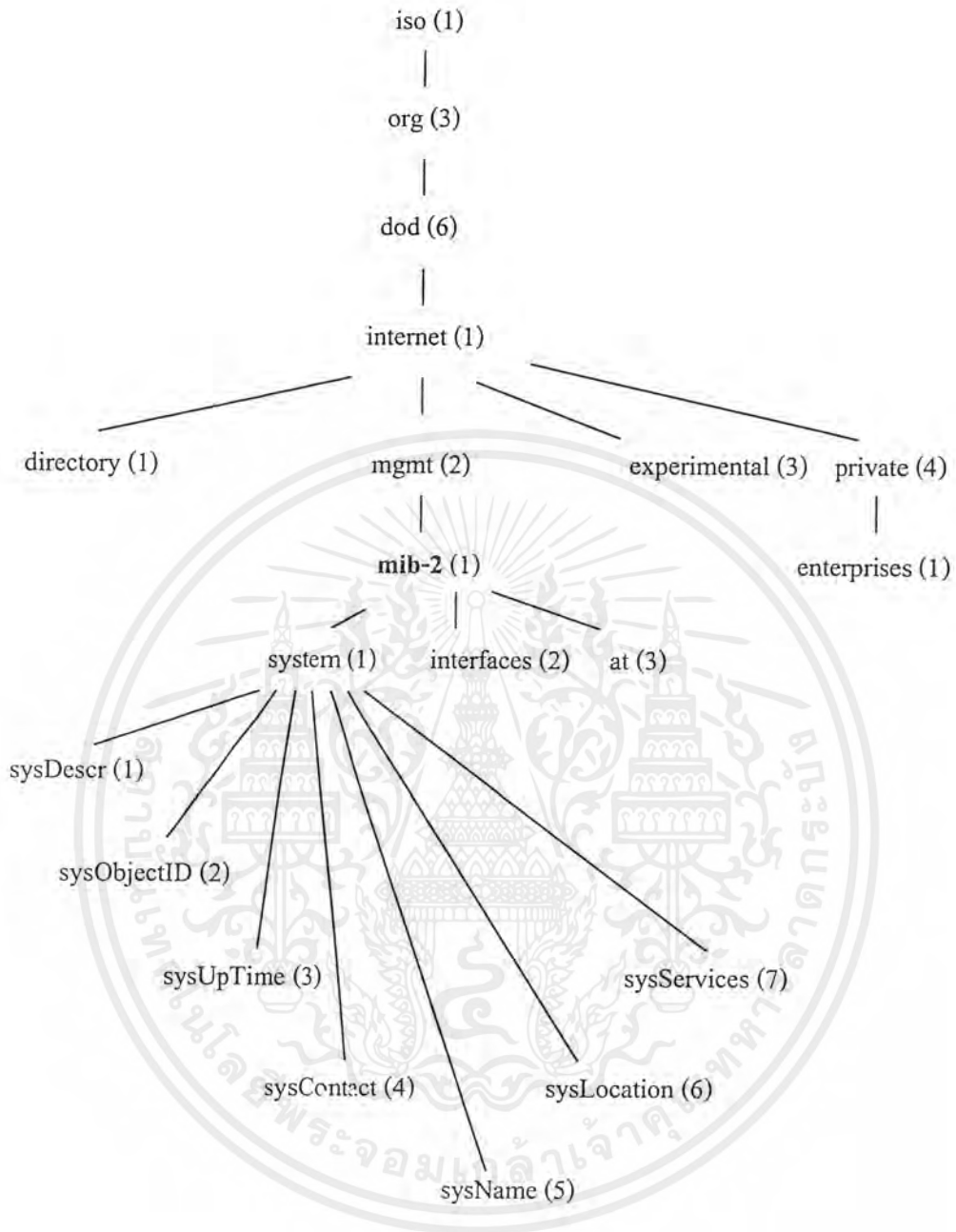
SNMP ถูกออกแบบให้เป็นโพรโตคอลในระดับแอปพลิเคชัน (application-level protocol) โดยมันจะทำงานบนยูดีพี (User Datagram Protocol, UDP) ซึ่งเป็น โพรโตคอลแบบคอนเนกชันเลส (connectionless) และ ไอพี (Internet Protocol, IP) ดังรูป



รูปที่ 3-1 ลักษณะของ SNMP

3.3 เมเนจเม้นท์อินฟอร์เมชันเบส

ที่อุปกรณ์ต่างๆ ซึ่งถูกบริหารนั้น จะมีการเก็บข้อมูลที่เกี่ยวข้องกับตัวของอุปกรณ์นั้นๆ ไว้ในเมเนจเม้นท์อินฟอร์เมชันเบส (Management Information Base หรือ มิบ) โดยโครงสร้างของมิบนั้นจะมีลักษณะเหมือนต้นไม้ (tree) ดังรูป



รูปที่ 3-2 บางส่วนของมิบ

3.3.1 เมเนจอ็อบเจกต์ (Managed Object)

เมเนจอ็อบเจกต์ เป็น ชนิดหรือคลาส ของข้อมูลที่เก็บในอุปกรณ์ เช่น sysDescr จะแสดงถึง คำบรรยาย (Description) ของอุปกรณ์

สำหรับการอ้างถึงเมเนจอ็อบเจกต์นั้น จะใช้ตัวบ่งชี้อ็อบเจกต์ (Object Identifier) โดยในแต่ละเมเนจอ็อบเจกต์นั้น จะมีตัวบ่งชี้อ็อบเจกต์เฉพาะตัว เช่น สำหรับ sysDescr จะเป็น 1.3.6.1.2.1.1.1 หรือ iso.org.dod.internet.mgmt.mib-2.system.sysDescr เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 อินสแตนซ์ (Instance) ของ เมเนจ็อบเจกต์

สำหรับการเข้าถึงค่าของข้อมูลนั้น จะต้องระบุอินสแตนซ์ ของเมเนจ็อบเจกต์ด้วย ซึ่งหากเมเนจ็อบเจกต์ใด มีอินสแตนซ์เดียว ก็สามารถใส่เลข 0 ต่อท้ายตัวบ่งชี้็อบเจกต์ (Object Identifier) ของเมเนจ็อบเจกต์นั้นไปได้เลย เช่น sysDescr ก็จะเป็น 1.3.6.1.2.1.1.0 แต่หากเมเนจ็อบเจกต์ใด มีหลายอินสแตนซ์ ก็จะต้องระบุเป็นหมายเลขกำกับด้วย เช่น ifType ภายใต้ interfaces group จะเป็น 1.3.6.1.2.1.2.1.3.1 สำหรับอินเทอร์เฟซ (interface) อันแรก และจะเป็น 1.3.6.1.2.1.2.1.3.4 สำหรับอินเทอร์เฟซอันที่ 4 (ใน 1 อุปกรณ์ อาจมีหลายอินเทอร์เฟซได้)

3.3.3 กลุ่มของมิบ

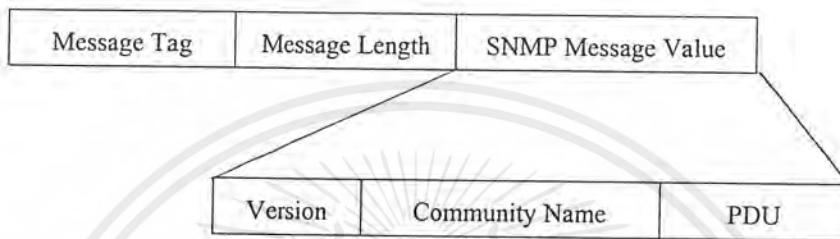
- **มิบ-2 (Mib-2 sub-tree)** เป็นมาตรฐาน มี 10 กลุ่มหลัก ได้แก่
 1. System group (1.3.6.1.2.1.1)
 2. Interfaces group (1.3.6.1.2.1.2)
 3. Address Translation group (1.3.6.1.2.1.3)
 4. Internet Protocol group (1.3.6.1.2.1.4)
 5. Internet Control Message Protocol group (1.3.6.1.2.1.5)
 6. Transmission Control Protocol group (1.3.6.1.2.1.6)
 7. User Datagram Protocol group (1.3.6.1.2.1.7)
 8. Exterior Gateway Protocol group (1.3.6.1.2.1.8)
 9. Transmission group (1.3.6.1.2.1.10)
 10. Simple Network Management Protocol group (1.3.6.1.2.1.11)
- **ส่วนขยายจากมิบ-2 (Extensions to the mib-2 sub-tree)**
เป็นส่วนเพิ่มเติมจาก มิบ-2 เช่น Remote Network Monitoring Objects (1.3.6.1.2.1.16) Host Resources Objects (1.3.6.1.2.1.25) เป็นต้น
- **มิบสำหรับการทดลองวิจัย (Experimental sub-tree)**
จะบรรจุ กลุ่มของมิบ (MIB group) ซึ่งถูกสร้างเพื่อการทดสอบ งานวิจัย ตลอดจนเพื่อจุดประสงค์ในการทดลองอื่นๆ เช่น AppleTalk (1.3.6.1.3.17) เป็นต้น
- **มิบสำหรับแต่ละผู้ผลิต (Private sub-tree)**
จะบรรจุ กลุ่มของมิบสำหรับแต่ละผู้ผลิต เช่น Hewlett Packard (1.3.6.1.4.1.11) 3Com (1.3.6.1.4.1.43) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 SNMP

3.4.1 เอสเอ็นเอ็มพีเมสเสจ (SNMP Message)

เอสเอ็นเอ็มพีเมสเสจ ที่ใช้ในการสื่อสารกันระหว่างเมเนจเมนต์สเตชัน และ เมเนจเมนต์เอเจนต์ (ทั้ง คอมมูนิตีเมสเสจ และ เรสปอนส์เมสเสจ) นั้น มีรูปแบบ ดังรูป



รูปที่ 3-3 เอสเอ็นเอ็มพีเมสเสจ

โดยทั้ง 3 ส่วนสำคัญนั้น สามารถอธิบายได้ดังนี้

ฟิลด์ระบุเวอร์ชัน (Version field)

เป็นส่วนที่ระบุเวอร์ชันของ SNMP โดยที่ถ้าเป็นเวอร์ชัน 1 ค่าในฟิลด์นี้จะเป็น 0 (เวอร์ชัน ลบด้วย 1)

ฟิลด์ระบุคอมมูนิตีเนม (Community Name field)

จะบรรจุ คอมมูนิตีเนม ซึ่งจะถูกใช้ในกระบวนการพิสูจน์ตน (authenticate) โดยเอเจนต์จะมีรายการของคอมมูนิตีเนม ที่ถูกต้องใช้ได้ เอาไว้เข้าคู่ (match) กับคอมมูนิตีเนม ในเอสเอ็นเอ็มพีเมสเสจ ที่รับมา ซึ่งถ้าเข้าคู่กับในรายการ เมสเสจก็สามารถประมวลผลต่อได้ แต่ถ้าไม่เข้าคู่ เมสเสจก็จะถูกทิ้งไป (ไม่ประมวลผลต่อ นั่นเอง)

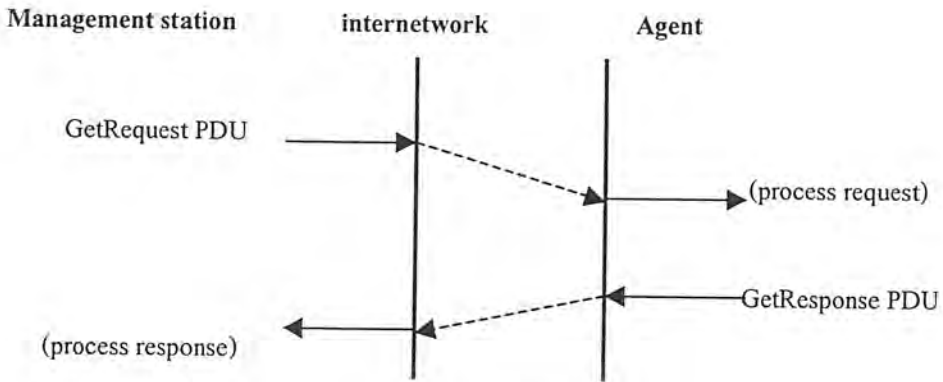
พีดียู (Protocol Data Unit ,PDU)

สำหรับ พีดียู จะต้องเป็น พีดียูชนิดใดชนิดหนึ่งใน 5 ชนิด ดังต่อไปนี้

- เกิดรีเควสพีดียู (GetRequest-PDU)

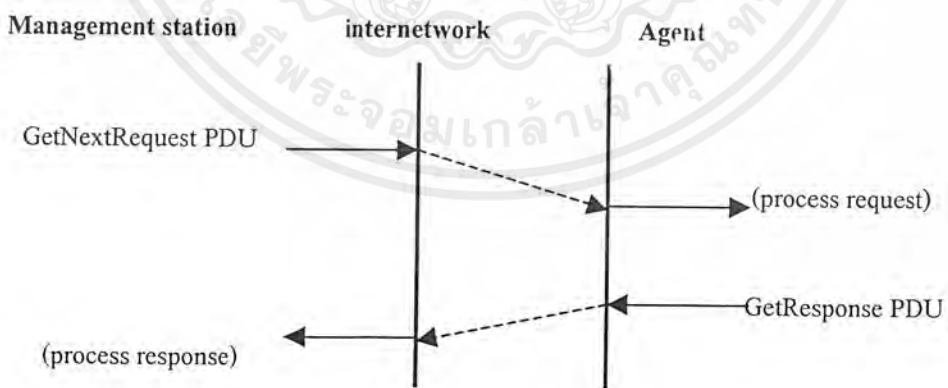
เมเนจเมนต์สเตชัน จะใช้เกิดรีเควสพีดียู ในการร้องขอค่าข้อมูลของอินสแตนซ์ของเมเนจอ็อบเจกต์ จากมิบ ของเมเนจเมนต์เอเจนต์ โดยที่ถ้าเงื่อนไขต่างๆ เช่น คอมมูนิตีเนม ถูกต้องครบถ้วน เมเนจเมนต์เอเจนต์ก็จะตอบกลับ

โดยใช้เกิดเรสปอนส์พีดียู (GetResponse-PDU) ซึ่งมีค่าข้อมูลตามต้องการ ดังรูป



รูปที่ 3-4 เกตรีเควสเมจเสจ (GetRequest Message)

- **เกตเนกทรีเควสพีดียู (GetNextRequest-PDU)**
 เกตเนกทรีเควสพีดียู จะมีรูปแบบ (format) เหมือนกันกับรูปแบบของ
 เกตรีเควสพีดียู ทุกประการ อีกทั้งการแลกเปลี่ยนเมสเสจ ก็มีลักษณะเช่นเดียวกัน
 ในการแลกเปลี่ยนของ เกตรีเควสพีดียู ด้วย ดังรูป

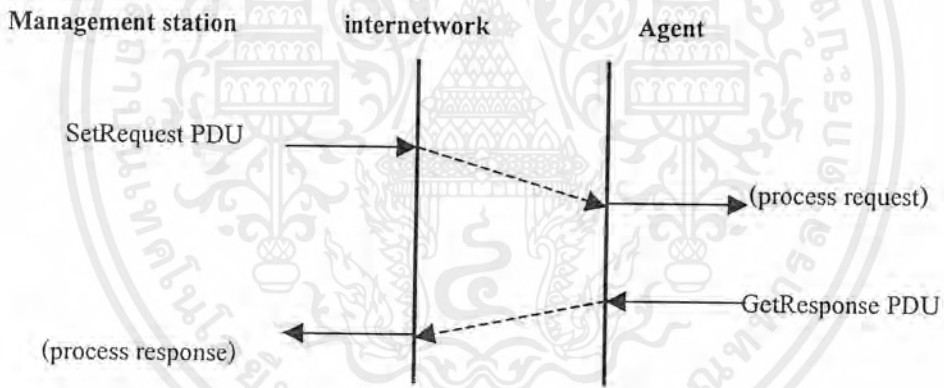


รูปที่ 3-5 เกตเนกทรีเควสเมจเสจ (GetNextRequest Message)

ถ้าสำหรับข้อแตกต่างของ เกิดเนกทรีเคสพีดียู กับ เกิดรีเคสพีดียู นั้น ก็คือ ในการร้องขอด้วยเกิดเนกทรีเคสพีดียู ค่าที่ได้จะเป็นค่าของอ็อบเจกต์อินสแตนซ์ (object instance) ซึ่งอยู่ถัดไป โดยที่การร้องขอด้วยเกิดเนกทรีเคสพีดียูนั้น จะมีประสิทธิภาพมากกว่าเกิดรีเคสพีดียู ในกรณีที่เจดิงค่าของอ็อบเจกต์เป็นชุดๆ เพราะว่า หากอันใดอันหนึ่งเกิดไม่สนับสนุน จะทำให้ไม่เกิด กรณีที่ไม่ส่งกลับ (return) ค่าเลย ดังเช่นในการใช้เกิดรีเคสพีดียู (ในการใช้เกิดรีเคสพีดียู ร้องขอค่าของอ็อบเจกต์ เป็นชุดๆ นั้น หากอันใดอันหนึ่งเกิดไม่สนับสนุน จะส่งผลให้ไม่ส่งกลับค่าข้อมูลเลย)

- เซ็ตรีเคสพีดียู (SetRequest-PDU)

เมเนจเม้นท์สเตชัน จะใช้เซ็ตรีเคสพีดียู ในการปรับเปลี่ยนค่าข้อมูลของอ็อบเจกต์อินสแตนซ์ ภายในมิบของเมเนจเม้นท์เอเจนท์ โดยจะมีลักษณะการแลกเปลี่ยนของเมสเสจ ดังรูป



รูปที่ 3-6 เซ็ตรีเคสเมสเสจ (SetRequest Message)

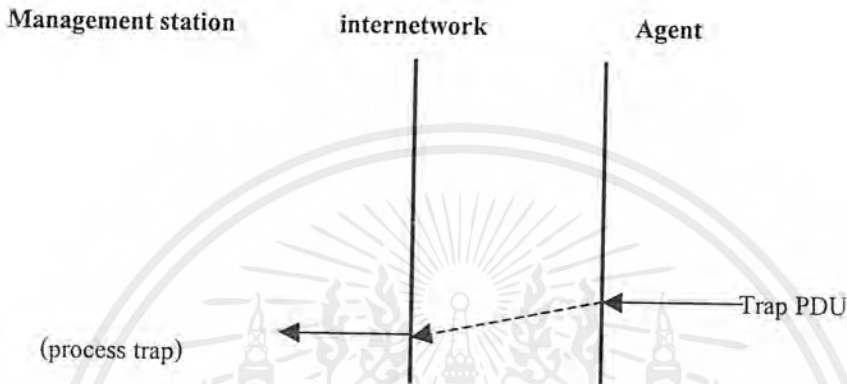
- เกิดเรสปอนส์พีดียู (GetResponse-PDU)

เมื่อเมเนจเม้นท์เอเจนท์ ได้ประมวลผล เกิดรีเคสพีดียู เกิดเนกทรีเคสพีดียู หรือ เซ็ตรีเคสพีดียู เป็นที่เรียบร้อยแล้ว มันก็จะส่ง เกิดเรสปอนส์พีดียู ซึ่งมีค่าข้อมูลที่เมเนจเม้นท์สเตชันร้องขอ ตลอดจนการแสดงความผิดพลาด (error) ที่อาจเกิดขึ้นให้แก่เมเนจเม้นท์สเตชัน (ในการส่งเกิดเรสปอนส์พีดียูกลับไปในั้น จะต้องมีการระบุรีเคสไอดี (request ID) ให้สัมพันธ์กันกับพีดียู ซึ่งส่งมายังเมเนจเม้นท์เอเจนท์ ก่อนหน้าด้วย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **แตรปพีดียู (Trap-PDU)**

แตรปพีดียู เป็นพีดียู ที่ถูกส่งจากเมเนจเม้นท์เอเจนท์ ไปยังเมเนจเม้นท์สเตชัน โดยที่เมเนจเม้นท์สเตชัน ไม่ได้ร้องขอ การส่งพีดียูชนิดนี้ จะเกิดขึ้นในกรณีที่มีเหตุการณ์สำคัญๆ เช่น ลิงก์ดาวน (link down) ลิงก์อัพ (link up) เป็นต้น โดยจะมีลักษณะดังรูป

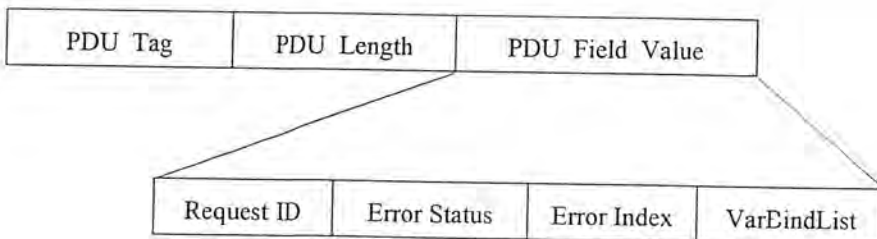


รูปที่ 3-7 แตรปเมสเสจ (Trap Message)

เอสเอ็นเอ็มพีเมสเสจ ดังกล่าวนี จะถูกกำหนด (define) ในรูปแบบของภาษา ASN.1 และภาษา ASN.1 จะถูกเข้ารหัสเป็นอ็อกเตต (octet) ส่งไปยังปลายทางโดยใช้ Basic Encoding Rules:BER อีกที

ส่วนของพีดียู ในเอสเอ็นเอ็มพีเมสเสจ นั้น สามารถแบ่งย่อยได้ดังนี้

- สำหรับ เกิดรีเวสพีดียู เกิดเนกทีวีเวสพีดียู เซอร์เวสพีดียู และ เกิดเรสปอนส์พีดียู จะมีลักษณะดังนี้



รูปที่ 3-8 รูปแบบพีดียูทั่วไป (Common PDU Format)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบต่างๆ สามารถอธิบายได้ดังนี้

PDU Tag

PDU Tag จะ เป็นส่วนที่ระบุถึงชนิดของพีดียู

PDU Length

PDU Length จะเป็นส่วนที่ระบุถึงขนาดความยาวของพีดียู

PDU Field Value ประกอบไปด้วย

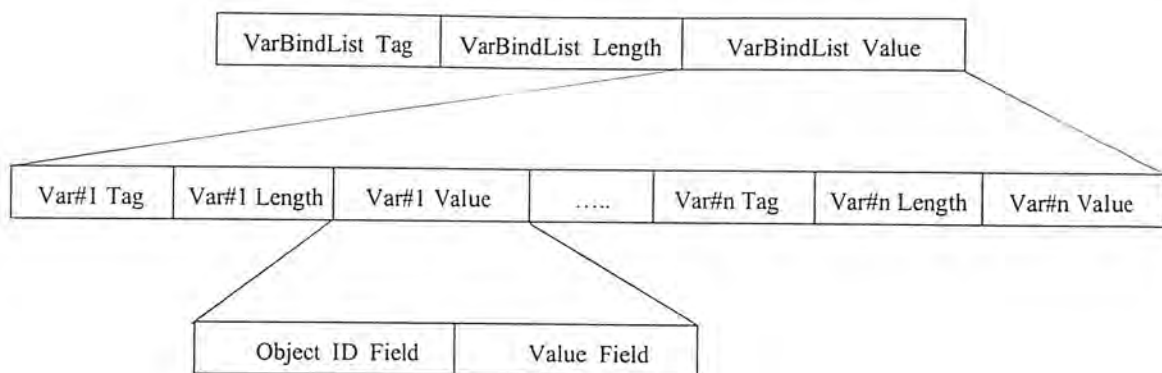
- **Request ID** : จะเป็นส่วนที่ระบุหมายเลขของการร้องขอ ซึ่งส่งจากเมนเฟรมที่สแตชันไปยังเมนเฟรมเอเจนต์ โดยที่ฟิลด์นี้ จำเป็นที่จะต้องเข้าคู่ (match) กับ เกิดเรสปอนส์พีดียู ที่ได้รับมาจากเมนเฟรมเอเจนต์
- **Error Status** : จะเป็นส่วนที่ถูกใช้โดยเมนเฟรมเอเจนต์ ในการส่งเกิดเรสปอนส์พีดียู เท่านั้น (สำหรับ เกิดรีเคสพีดียู เกิดเนกทีฟรีเคสพีดียู และ เซอร์เคสพีดียู จะมีค่าในฟิลด์นี้ เป็น 0) โดยที่ค่าในฟิลด์นี้ จะบ่งบอกถึงสถานะของคำสั่ง (พีดียู) ก่อนหน้า ที่ส่งเข้ามายังเมนเฟรมเอเจนต์ ซึ่งหากค่าในฟิลด์นี้ ไม่เท่ากับ 0 แสดงว่าความผิดพลาดได้เกิดขึ้นแล้ว (ค่าในฟิลด์นี้ จะขึ้นกับเงื่อนไขความผิดพลาดแบบต่างๆ ที่เกิดขึ้น ตามตาราง)

Status Name	Status Value
NoError	0
TooBig	1
NoSuchName	2
BadValue	3
ReadOnly	4
GenError	5

ตารางที่ 3-1 แสดง ค่าเงื่อนไขความผิดพลาด (error condition) แบบต่างๆ

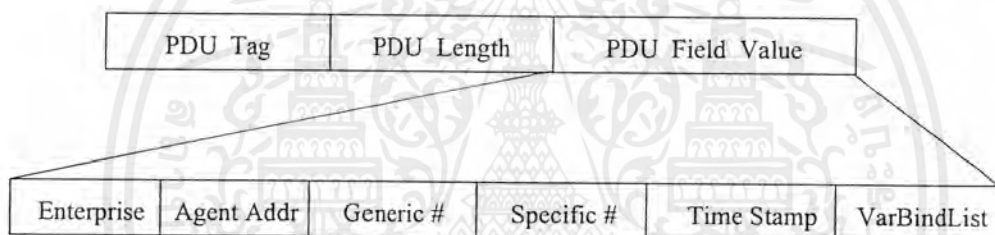
- **Error Index** : จะเป็นส่วนที่ให้ข้อมูลเพิ่มเติมเกี่ยวกับ เงื่อนไขความผิดพลาด โดย Error Index จะชี้ไปยัง variable ตัวแรก ใน Variable Bindings List ซึ่งทำให้เกิดเงื่อนไขความผิดพลาด
- **VarBindList** : จะเป็นรายการอินสแตนซ์ ของเมนเฟรมอ็อบเจกต์ โดยในส่วน VarBindList นี้ จะแบ่งย่อยได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-9 รูปแบบของ Variable Bindings List

➤ สำหรับ แทรปพีเดีย จะมีลักษณะดังนี้



รูปที่ 3-10 รูปแบบของ แทรปพีเดีย

- **Enterprise** : จะบรรจุตัวบ่งชี้ชื่อออบเจกต์ (Object Identifier) สำหรับอุปกรณ์เครือข่ายที่ผลิต แทรป (Trap)
- **Agent Address** : จะบรรจุไอพีแอดเดรส (IP Address) ของเมเนจเม้นเอเจนท์
- **Generic Trap** : จะบรรจุ ค่าจำนวนเต็มซึ่งแทน แทรปซึ่งกำหนดไว้แล้ว (predefined trap) มาตรฐาน 1 ใน 7 อย่าง ดังต่อไปนี้
 1. ColdStart
 2. WarmStart
 3. LinkDown
 4. LinkUp
 5. authenticationFailure
 6. egpNeighborLoss
 7. enterpriseSpecific
- **Specific Trap** : จะบรรจุ ค่าของแทรปซึ่งถูกกำหนด สำหรับแอนเตอร์ไพรส์ (enterprise) เฉพาะ นั้นๆ
- **Agent Address** : จะบรรจุ “เวลา” ที่ แทรป ถูกผลิตขึ้น

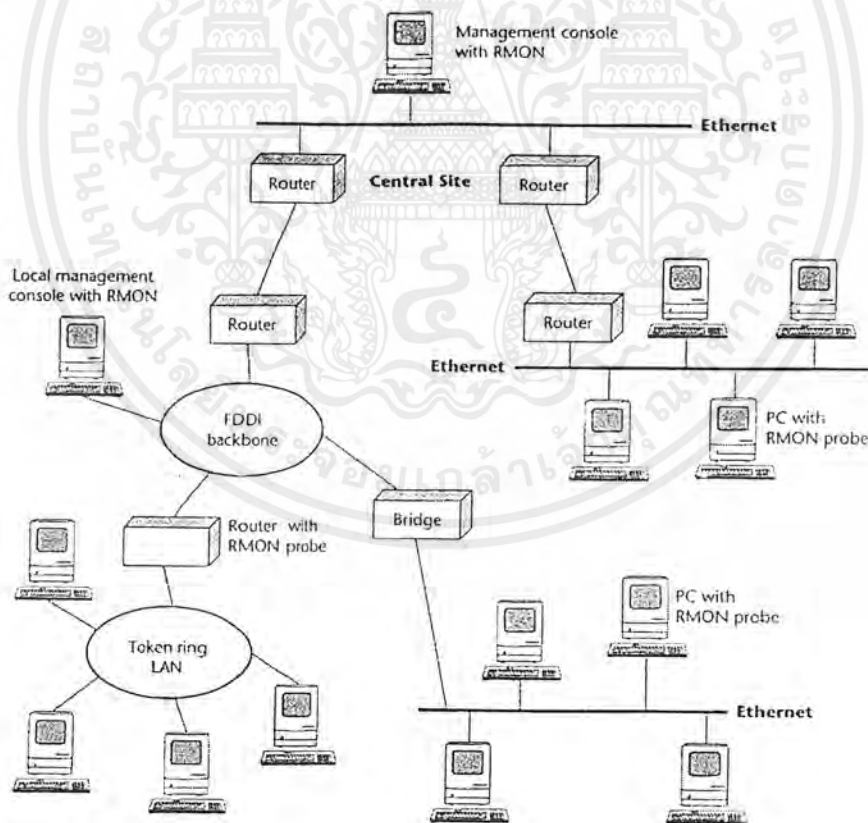
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การมอนิเตอร์เครือข่ายระยะไกล (Remote Network Monitoring ,RMON)

สำหรับเครือข่ายที่มีขนาดใหญ่ มีเครือข่ายย่อยมาก การใช้ SNMP เพียงอย่างเดียว อาจทำให้ไม่สามารถบริหารเครือข่ายได้อย่างมีประสิทธิภาพ เนื่องจากในการทำงานนั้น จะมีลักษณะที่เมเนจเมนต์สแตชัน จะคอยโพล (poll) ร้องขอข้อมูลจากเมเนจเมนต์เอเจนท์ที่อยู่ในอุปกรณ์อยู่เรื่อยๆ ซึ่งหากต้องการข้อมูลจากอุปกรณ์ในความถี่สูงๆ (เช่น ทุกๆ 10 วินาที) เมสเสจที่ต้องสื่อสารกันระหว่างเมเนจเมนต์สแตชัน และ เมเนจเมนต์เอเจนท์ ก็จะมีปริมาณมาก ส่งผลให้การจราจร (traffic) ในเครือข่ายมากขึ้น ประสิทธิภาพของเครือข่ายลดลงได้ อีกทั้งในกรณีเช่นนี้ จำเป็นต้องอาศัย เมเนจเมนต์สแตชัน ที่มีความสามารถสูง เนื่องจากต้องทำการประมวลผลและแสดงผลข้อมูลมากมายที่มาจากหลายเครือข่ายย่อยๆ

และนอกจากนั้นแล้ว การใช้ SNMP เพียงอย่างเดียว ทำให้ เมเนจเมนต์สแตชัน ยากที่จะทราบถึงสภาพการจราจร (traffic) โดยรวม ภายในระบบเครือข่ายย่อยๆ อีกด้วย

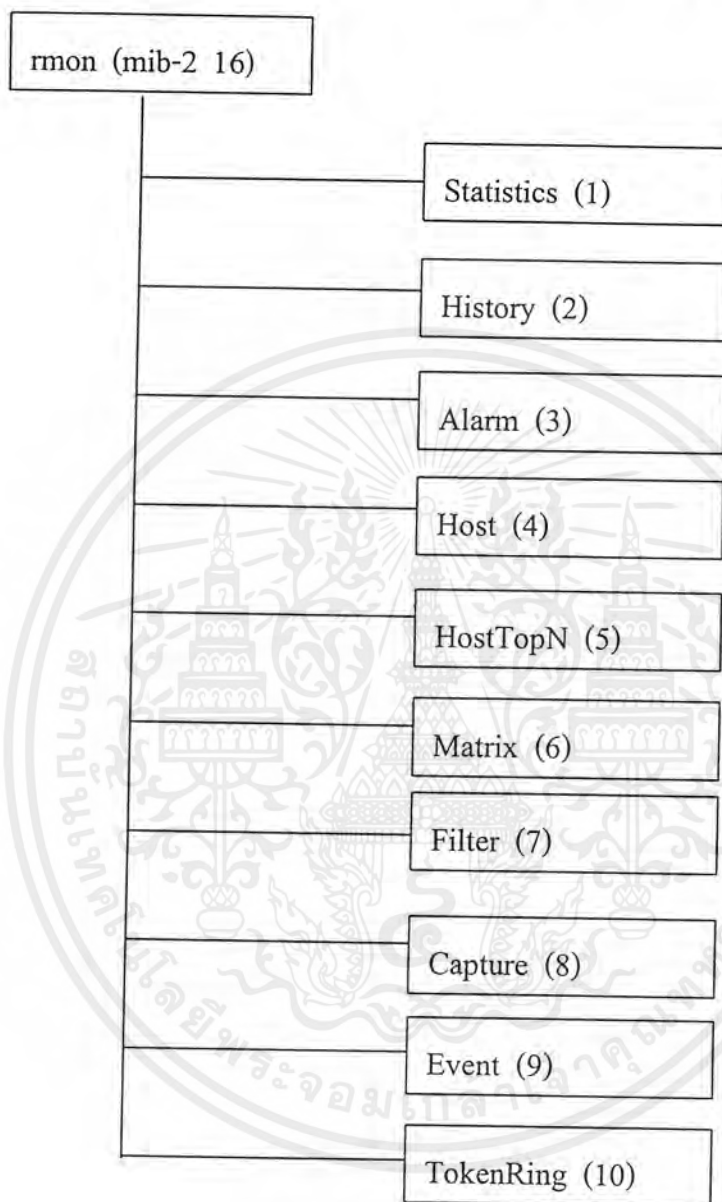
สำหรับ concept ของ RMON นั้น จะมีสิ่งที่เรียกว่า “รีโมทมอนิเตอร์ (remote monitor)” อยู่ โดยรีโมทมอนิเตอร์ที่ว่ามี สามารถเป็นได้ทั้งอุปกรณ์ซึ่งเป็นฮาร์ดแวร์ หรืออาจเป็นซอฟต์แวร์ ก็ได้ ตัวอย่างเช่น พีซี (PC) ที่มี RMON probe ดังรูป



รูปที่ 3-11 ตัวอย่างการใช้ RMON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรีโมทมอนิเตอร์ จะมีความสามารถหลายอย่างในการทำงานที่สูง มันสามารถเก็บรวบรวมข้อมูล ทำสถิติ เช่น จำนวนการชน (collision) ภายในเครือข่ายย่อยที่มันอยู่ได้ โดยข้อมูลต่างๆ นั้น จะอยู่ในรูปของ RMON MIB ซึ่งอยู่ภายใต้มิม-2 กลุ่มที่ 16 (mib-2 16) ดังรูป



รูปที่ 3-12 RMON MIB

เมเนจเมนต์สเตชัน สามารถที่จะร้องขอข้อมูลต่างๆ ที่อยู่ภายใน RMON MIB ได้ เช่นเดียวกับมิมอื่นๆ ไป ซึ่งจาก concept ของ RMON ดังกล่าว จะเป็นการช่วยแบ่งเบาภาระของเมเนจเมนต์สเตชัน ให้น้อยลงได้

บทที่ 4

ภาษาจาวา (Java)

4.1 การเขียนโปรแกรมแบบอ็อบเจกต์เชิงวัตถุ (Object Oriented Programming)

การเขียนโปรแกรมแบบอ็อบเจกต์เชิงวัตถุ เป็นการสร้างส่วนย่อย ๆ ของโปรแกรมที่เรียกว่า วัตถุ (Object) ซึ่งมีคุณสมบัติเฉพาะที่สามารถกระทำงานบางอย่างให้สำเร็จไปได้ขึ้นมา และนำส่วนประกอบย่อย ๆ เหล่านั้นมาประกอบกันขึ้นเป็นโปรแกรมใหญ่ เพื่อให้สามารถทำงานได้บรรลุวัตถุประสงค์ที่ตั้งไว้ โดยในความเป็นจริงแล้ว เราอาจจะไม่ได้สร้างส่วนย่อย ๆ เหล่านั้นขึ้นมาเอง แต่เราได้นำส่วนย่อยที่ผู้อื่นสร้างไว้แล้วมาใช้งานเท่านั้น ซึ่งเราอาจจะไม่รู้ด้วยซ้ำว่าใครเป็นคนสร้าง และสร้างขึ้นมาได้อย่างไร แต่เราต้องรู้ว่า โปรแกรมส่วนย่อยเหล่านั้นสามารถทำอะไรได้บ้าง และจะสั่งให้ทำงานได้อย่างไร

ขออธิบายให้เข้าใจง่ายขึ้นด้วยการยกตัวอย่างเปรียบเทียบ โดยเป็นตัวอย่างการประกอบรถยนต์ ซึ่งในการประกอบรถยนต์ขึ้นมาคันหนึ่งนั้นต้องใช้ชิ้นส่วนมากมายหลายชนิดได้แก่

- เครื่องยนต์
- ตัวถัง
- ล้อรถยนต์
- อุปกรณ์ตกแต่ง

โดยอุปกรณ์ที่จะนำมาประกอบเข้าด้วยกันนี้เราไม่จำเป็นต้องสร้างขึ้นเองทั้งหมด เราอาจจะไปซื้อมาหรือใช้อะไหล่เก่าที่มีอยู่ก็ได้ และหลังจากประกอบจนได้รถที่สามารถวิ่งได้มาคันหนึ่งแล้ว เราอาจต้องการสร้างรถคันใหม่เนื่องจากได้อุปกรณ์ชิ้นใหม่มาและสามารถถอดชิ้นส่วนที่ประกอบให้กับรถคันเดิม นำไปประกอบให้กับรถคันใหม่ของเราได้

4.1.1 คลาส (Class)

เมื่อมองรอบ ๆ ตัวเราจะเห็นว่ามิวัตถุ (object) หลายอย่าง ที่ถือว่าเป็นชนิดเดียวกัน อย่างเช่น จักรยานของเราเอง ซึ่งถือได้ว่าเป็นเพียงแค่จักรยานคันหนึ่ง ของจักรยานที่มีอยู่ในโลกนี้ เมื่อเรานำลักษณะดังกล่าวมาเชื่อมโยงกับการเชื่อมโยงเชิงวัตถุ แล้วสามารถกล่าวได้ว่า จักรยานของเรานั้นเป็น ตัวอย่างหนึ่ง (instance) ของวัตถุของคลาสที่เรียกว่าจักรยาน นั่นคือ ในการสร้างวัตถุขึ้นมาหนึ่งชิ้นเพื่อใช้งานนั้น เราจะต้องเริ่มจากการสร้างต้นแบบของวัตถุนั้น ๆ เสียก่อน แล้วจึงทำการสร้างวัตถุที่จะใช้งานจริง ๆ ที่มาจากต้นแบบที่สร้างขึ้นมา ซึ่งต้นแบบของวัตถุที่จะต้องทำการสร้างขึ้นมาเรียกว่า คลาส (class) และวัตถุที่จะสร้างขึ้นมาเรียกว่า ออบเจกต์ (object)

คลาส (Class) เป็นต้นแบบ (blueprint) ของวัตถุหลาย ๆ ชนิดที่มีลักษณะเหมือนกันบางอย่าง ทำให้สามารถจัดให้อยู่ในจำพวกเดียวกันได้ ตัวอย่างเช่น คลาสของสุนัข คือสิ่งมีชีวิตที่มีขาสี่ขา มีขนเป็นเส้นทั่วร่างกาย สามารถเห่าได้ มีเขี้ยวยาว เป็นต้น สิ่งเหล่านี้เป็นคุณลักษณะ (attributes) และพฤติกรรม (behavior) พื้นฐานที่สุนัขควรจะมี ดังนั้นในการอ้างถึงสุนัข เป็นการอ้างถึงชื่อโดยรวมของสัตว์ประเภทหนึ่งที่มีลักษณะดังกล่าวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการสร้างคลาสขึ้นมาหนึ่งคลาสนั้น เราจะต้องทำการสร้างองค์ประกอบให้กับคลาสนั้น ๆ ด้วย ซึ่งองค์ประกอบพื้นฐานที่สำคัญของคลาสนั้นแบ่งออกได้เป็นสองชนิดใหญ่ ๆ ดังนี้คือ คุณลักษณะ (Attributes) และ พฤติกรรม (Behavior) โดยมีรายละเอียดดังนี้

คุณลักษณะของคลาส (Attributes)

คุณลักษณะของคลาสคือสิ่งที่เป็นลักษณะเฉพาะตัวที่จะแสดงให้เห็นถึงความแตกต่างกันของวัตถุแต่ละชนิด เช่น สี ขนาด ความสูง น้ำหนัก เป็นต้น ตัวอย่างเช่น หากเราพิจารณาถึงคลาสของสุนัขแล้ว คลาสของสุนัขจะมีคุณลักษณะพื้นฐานต่าง ๆ ที่สามารถเห็นได้ทั่วไป ได้แก่

- สุนัขมีขาสี่ขา
- สุนัขมีขนสั้นทั่วตัว
- สุนัขมีฟันแหลมคม
- สุนัขมีหาง

สำหรับในทางการเขียนโปรแกรมแล้วค่าคุณลักษณะทั้งหลายจะทำการระบุโดยใช้ตัวแปรต่าง ๆ เป็นตัวระบุ

พฤติกรรมของคลาส (Behavior)

พฤติกรรมของคลาสเป็นส่วนที่แสดงให้เห็นว่าคลาสนั้น ๆ สามารถทำอะไรได้บ้าง สามารถทำงานร่วมกับคลาสนั้น ๆ ได้อย่างไร หรือคลาสนั้น ๆ สามารถสั่งให้คลาสนั้น ๆ ทำงานอะไรให้ได้ อย่างเช่น สำหรับคลาสของสุนัข

- สุนัขสามารถเห่าได้
- เมื่อสุนัขหิวมันก็จะกินอาหาร
- เมื่อง่วงมันก็จะนอน
- สุนัขสามารถวิ่งได้

เช่นเดียวกันสำหรับในการเขียนโปรแกรมแล้ว พฤติกรรมต่าง ๆ เหล่านี้จะถูกเขียนให้เป็น เมธอด (Method) เพื่อจะสั่งให้มันแสดงพฤติกรรม โดยในแต่ละพฤติกรรมก็จะแทนแต่ละเมธอดซึ่งถ้าเทียบเมธอดดังกล่าวกับโปรแกรมภาษาอื่น ๆ แล้ว จะเทียบได้กับฟังก์ชัน (Function) หรือ โพรซีเจอร์ (Procedure) นั่นเอง

4.1.2 ออบเจกต์ (Object)

เมื่อเรามองไปรอบ ๆ ก็จะพบว่าสิ่งต่าง ๆ หลายอย่างที่เป็นวัตถุอย่างเช่น สุนัขของคุณ, โต๊ะ, โทรทัศน์ จักรยาน และอื่น ๆ อีกมากมาย และดังที่ได้กล่าวไว้ในข้างต้นว่าวัตถุเป็นสิ่งที่มีความจริง ๆ สร้างมาจากคลาส และได้มีการสืบทอดคุณลักษณะ (attributes) และ พฤติกรรม (behavior) มาจากคลาสของมัน

นอกจากนี้วัตถุก็ยังมีคุณลักษณะ (Attribute) เฉพาะตัวของแต่ละวัตถุอีกต่างหาก ซึ่งในการเขียนโปรแกรมค่าเหล่านี้ก็จะถูกเก็บในตัวแปรอีกเช่นกัน แต่จะเป็นตัวแปรของแต่ละวัตถุดังนั้นทำให้ค่าตัวแปรของวัตถุแต่ละตัว

นั่นจึงมีค่าที่แตกต่างกันไปตามลักษณะของวัตถุแต่ละตัว แต่ค่าในตัวแปรที่ได้มาจากคลาสเดียวกันนั้นจะมีค่าเท่ากันเสมอ

ดังนั้นความแตกต่างระหว่างวัตถุกับคลาสก็คือเมื่อเปรียบเทียบกับโลกความจริง เมื่อก้าวถึงจักรยาน วัตถุ ก็จะเป็นจักรยานที่เรากำลังปั่นอยู่ หรือ จักรยานคันอื่นๆในโลก แต่ว่าคลาสจะหมายถึงความถึงกลุ่มของวัตถุที่เราเรียกกันว่าจักรยาน หรืออาจพูดว่ามันคือ blueprint ของจักรยานที่ใช้สำหรับผลิตจักรยานขึ้นมา

4.1.3 เมธอด (Method)

นอกจากตัวแปรต่าง ๆ แล้ว เมธอดเป็นส่วนประกอบที่สำคัญอีกส่วนหนึ่งของคลาสซึ่งทำหน้าที่เหมือนกับ ฟังก์ชัน หรือ โปรซีเจอร์ ในภาษาปาสคาล เมธอดเป็นตัวแทนคุณสมบัติของออบเจกต์หนึ่ง ๆ ซึ่งทำให้ออบเจกต์ แต่ละออบเจกต์ที่มาจากคนละคลาสเกิดความสามารถที่แตกต่างกันออกไป

4.1.4 โพลิมอร์ฟิซึม (Polymorphism)

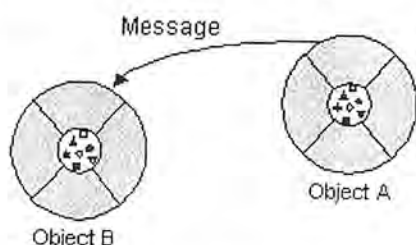
โพลิมอร์ฟิซึม คือการที่โปรแกรมสามารถมีเมธอดที่ชื่อเดียวกันได้ ซึ่งในการแยกความแตกต่างระหว่าง เมธอดนั้นจะไม่ใช้แค่ชื่อของเมธอดเท่านั้น แต่จะใช้ชื่อเต็มของเมธอด (fully qualified name) ซึ่งชื่อเต็มที่ว่านี้ ประกอบด้วย ชื่อเมธอด, ชื่อคลาส, และ อาร์กิวเมนต์ (Argument) ของเมธอดนั้น ดังนั้นเมธอดต่าง ๆ ที่มีการใช้ใน โปรแกรมอาจจะมีชื่อซ้ำกันก็ได้ แต่ชื่อเต็มจะต้องไม่เหมือนกัน ซึ่งอาจจะอยู่ในลักษณะที่ว่า

- เมธอดที่มีชื่อเหมือนกันแต่อยู่คนละคลาส
- เมธอดที่มีชื่อเดียวกันในคลาสเดียวกัน (function overloading)

4.1.5 ข้อความ (Message)

วัตถุเดี่ยว ๆ ที่มีเพียงวัตถุเดียว โดยทั่วไปแล้วจะไม่ค่อยมีประโยชน์มากนัก แต่มักจะไปปรากฏเป็นส่วนประกอบของโปรแกรมที่ใหญ่กว่า หรืออยู่ในแอปพลิเคชันที่มีวัตถุอื่น ๆ รวมอยู่ด้วย เมื่อวัตถุต่าง ๆ ถูกกระทำซึ่งกันและกันแล้วผู้เขียนโปรแกรมก็จะได้รับผลประโยชน์ที่สูงกว่าและจะได้พฤติกรรม (Behavior) ที่ซับซ้อนกว่า อย่างเช่น จักรยาน ซึ่งประกอบขึ้นด้วย ตะขอ โครมรถที่รวมกันของไทเทเนียม ยาง และอื่นโดยส่วนต่าง ๆ เหล่านี้ จะไม่สามารถทำอะไรได้มากนัก แต่เมื่อรวมกันเป็นจักรยาน มันจะให้ประโยชน์ที่มากมาย

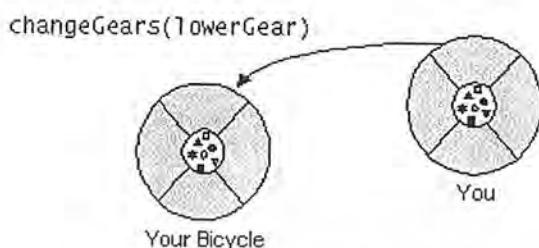
การกระทำซึ่งกันและกัน (interact) และการติดต่อสื่อสาร (communicate) กันของซอฟต์แวร์ วัตถุกับตัวอื่น ๆ จะใช้วิธีการส่ง ข้อความ (messages) ถึงอีกตัวหนึ่ง เช่น เมื่อวัตถุ A ต้องการให้วัตถุ B ปฏิบัติการของเมธอดอันใดอันหนึ่งของวัตถุ B แล้ววัตถุ A ก็จะส่งข้อความ (messages) ให้กับวัตถุ B



รูปที่ 4-1 แสดงการส่งข้อความ (message) จาก วัตถุ A ให้กับวัตถุ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บางครั้งวัตถุที่เป็นตัวรับข้อความ (message) จะต้องการข้อมูล (information) มากกว่าข้อความ (message) เพียงอย่างเดียว ซึ่งมันจะรู้ได้ทันทีว่าจะทำอะไร เช่น เมื่อเราต้องการเปลี่ยนเกียร์จักรยาน เราต้องบอกถึงระดับเกียร์ที่เราต้องการด้วย ซึ่งข้อมูลที่ได้ส่งมากับ ข้อความ (message) นั่นคือ พารามิเตอร์ (parameters)



รูปที่ 4-2 แสดงการส่งข้อความ (message) จาก วัตถุ A ให้กับ วัตถุ B โดยมีพารามิเตอร์ (parameter) ด้วย

ส่วนประกอบ 3 อย่างต่อไปนี้เพียงพอสำหรับข้อมูลที่จะส่งให้กับวัตถุเพื่อให้วัตถุนั้นดำเนินการ (perform) ไปตามเมธอดที่เราต้องการ ซึ่งได้แก่

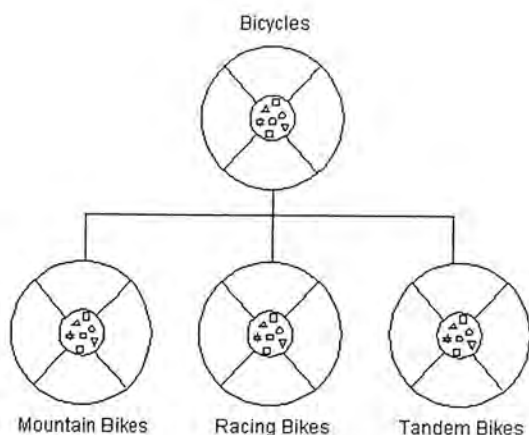
1. ที่อยู่ (Address) ของวัตถุที่ถูกส่งไป (จักรยาน)
2. ชื่อของเมธอดที่ต้องการให้กระทำ (เปลี่ยนเกียร์)
3. พารามิเตอร์ที่เมธอดนั้นต้องการ (ระดับเกียร์ที่จะเปลี่ยน)

ข้อดีของการใช้การส่งข้อความ (Message)

พฤติกรรม (behavior) ของวัตถุถูกแสดงผ่านเมธอดของมัน ดังนั้น การส่งผ่านข้อความ (message) จะสนับสนุนทุกการกระทำซึ่งกันและกัน (interaction) ที่เป็นไปได้ระหว่างวัตถุโดยที่วัตถุไม่จำเป็นต้องอยู่ใน Process หรืออยู่ใน machine เดียวกันเพื่อส่งและรับข้อความ (message) กลับ และส่งต่อให้สิ่งอื่น

4.1.6 การสืบทอด (Inheritance)

การสืบทอด (Inheritance) ในภาษาไทยมีความหมายตรงๆ ว่า การสืบทอด หรือ มรดกตกทอด แต่ถ้าเราจะมากล่าวถึง ในเชิงของการเชื่อมโยงเชิงวัตถุ (Object Oriented) โดยพิจารณาเกี่ยวกับวัตถุในโลกปัจจุบันแล้ว เมื่อยกตัวอย่างกับ จักรยานก็จะได้ว่า เมื่อพิจารณาถึงจักรยานชนิดต่างๆ เช่น mountain bike, race bike หรือ tandems ชนิดของจักรยานเหล่านี้ ก็จะถูกเรียกว่า คลาสย่อย (subclass) ของ คลาสหลัก (superclass) ซึ่งมีชื่อว่า จักรยาน



รูปที่ 4-3 แสดงการเรียงลำดับของคลาส โดยมี คลาสย่อย (subclass) ของ คลาสหลัก (superclass)

โดยแต่ละคลาสย่อยนั้นจะมีการสืบทอด คุณลักษณะ (attribute), พฤติกรรม (behavior) มาจากคลาสหลัก (superclass) ทุกอย่างเช่น mountain bike race bike ซึ่งเป็นคนละคลาสย่อย (subclass) ก็จะมีสถานะที่สืบทอดมาเหมือนกันคือ เช่น มี attribute ความเร็ว และ จังหวะในการปั่น เป็นต้น หรือว่าจะมี behaviors ที่สืบทอดมาจากคลาสหลัก (superclass) เดียวกันเช่น การเบรก หรือการเปลี่ยน เกียร์ เป็นต้น

แต่อย่างไรก็ตาม คลาสหลัก (superclass) ไม่ได้ได้รับการจำกัดว่าจะต้องมีเฉพาะ คุณลักษณะ (attribute) และ พฤติกรรม (behavior) ที่ได้รับการสืบทอดมาจากคลาสหลัก (superclass) ของมันเท่านั้น แต่ละคลาสย่อย (subclass) สามารถ ทำการเพิ่มตัวแปร และเมธอดให้กับตัวเองได้ ยกตัวอย่างเช่น tandem จะเป็นจักรยานซึ่งมีสองที่นั่งและมี 2 handle bars หรือ mountain bike ก็จะมีชุดเกียร์พิเศษ ซึ่งมีอัตราทดที่ดีๆ เพิ่มเติมเข้ามา

นอกจากนี้ คลาสย่อย (subclass) ก็ยังสามารถทำการแก้ไขหรือปรับปรุงเมธอดที่ได้รับมาจากคลาสหลัก (superclass) ให้มีประสิทธิภาพ หรือมีความสามารถมากขึ้นเพื่อนำมาใช้ประโยชน์เฉพาะทางได้อีกด้วย เช่น ถ้าเรามี mountain bike ที่มีชุดเกียร์พิเศษ เราก็จะทำการแก้ไขเมธอด “การเปลี่ยนเกียร์” เสียใหม่เพื่อให้ผู้ใช้สามารถใช้ ชุดเกียร์ใหม่ได้

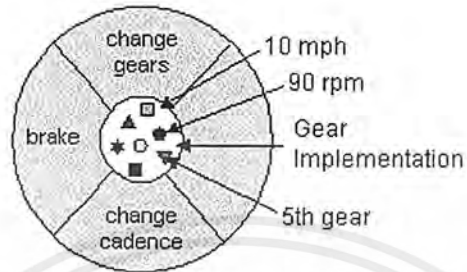
ในการสืบทอดคุณสมบัติของคลาสหลัก (superclass) นั้น คลาสย่อย (subclass) ไม่ได้ถูกจำกัดว่าจำเป็นต้องมีการสืบทอด เฉพาะ คลาสหลัก (superclass) ที่อยู่ก่อนหน้ามันเพียงชั้นเดียวเท่านั้น การสืบทอดคุณสมบัติของคลาสหลัก (superclass) นั้นสามารถทำได้ตั้งแต่ ต้นตระกูลของคลาสจนถึงปลายสุดของ subclass (class hierarchy)

ข้อดีของการใช้การสืบทอด (Inheritance)

ประโยชน์ของการสืบทอด (Inheritance) นั่นก็คือคลาสย่อย (subclass) จะมีคุณสมบัติพื้นฐานที่สืบทอดมาจาก คลาสหลัก (superclass) ของมัน และการใช้คุณสมบัติของ Inheritance จะทำให้ สามารถใช้โปรแกรมที่อยู่ใน คลาสหลัก (superclass) ได้ตลอดเวลา

4.1.7 เอนแคปซูเลชัน (Encapsulation)

สำหรับในสร้างเมธอดนั้น เราจะต้องมีการกำหนดตัวควบคุมการเข้าถึงเมธอดให้มันด้วย เพื่อเป็นการป้องกันค่าต่าง ๆ ที่เก็บในตัวแปร โดยเมื่อเราได้กำหนดตัวควบคุมดังกล่าวแล้ว เมธอดนั้นจะยอมให้วัตถุที่มีลักษณะตามที่ได้กำหนดไว้เท่านั้นที่จะสามารถขอใช้เมธอดนั้นได้



รูปที่ 4-4 แสดงตำแหน่งของ ตัวแปร กับเมธอดในลักษณะของ cell

จากรูปจะเห็นว่าค่าตัวแปรของวัตถุถูกจัดไว้ภายในส่วนกลางของวัตถุ ส่วนเมธอดต่างก็อยู่รอบนอก ตัวแปร และทำการซ่อนส่วนกลาง (ส่วนที่ตัวแปรอยู่) ของวัตถุเก็บไว้แยกออกจากวัตถุอื่น ๆ ในโปรแกรมนั้น ลักษณะเช่นนี้ เรียกว่า “encapsulation” ซึ่งเป็นการกำหนดให้ค่า ตัวแปร ต่าง ๆ ในวัตถุสามารถถูกอ้างอิง (reference) ได้เฉพาะเมธอดของวัตถุนั้น เท่านั้น แต่ก็มีบ่อยครั้งที่การดำเนินการหรือประสิทธิภาพเป็นเหตุผลทำให้วัตถุนั้นอาจจะต้องการแสดงค่า ตัวแปร บางค่าหรือบิตบ่งบางเมธอดเพื่อห้ามไม่ให้วัตถุอื่น ๆ ทำการขอใช้เมธอดดังกล่าว

ข้อดีของเอนแคปซูเลชัน (Encapsulation)

เอนแคปซูเลชัน (encapsulation) จะเกี่ยวข้องกับตัวแปรและเมธอดซึ่งเป็นตัวช่วยในการพัฒนา โปรแกรมให้เป็นที่ไปในลักษณะที่ง่ายขึ้น คือ

- **Modularity**

สำหรับรหัสโปรแกรม (source code) ของวัตถุหนึ่ง ๆ สามารถที่จะเขียนและยังคงไว้ได้โดยไม่ขึ้นกับรหัสโปรแกรม (source code) ของวัตถุอื่น ดังนั้นวัตถุจึงง่ายที่จะทำการส่งไปยังส่วนอื่น ๆ ภายในระบบ เช่น คุณสามารถทำการมอบจักรยานของคุณ ให้กับคนอื่นได้โดยที่จักรยานยังใช้งานได้อยู่

- **Information Hiding**

วัตถุที่มีการเชื่อมต่อกันในลักษณะที่เป็นสาธารณะ โดยที่วัตถุอื่น สามารถทำการติดต่อสื่อสารกับมันได้ แต่วัตถุนั้นสามารถยังคงรักษาลักษณะเฉพาะของรูปแบบ (information) และ เมธอดซึ่งสามารถทำการเปลี่ยนแปลงได้ตลอดเวลา โดยปราศจากผลกระทบต่อวัตถุที่ขึ้นกับมัน เช่น คุณไม่จำเป็นต้องเข้าใจระบบกลไกการทำงานของเกียร์ในจักรยานของคุณ เพื่อจะสั่งให้มันทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 JAVA Application

เป็นการนำ JAVA มาเขียนเป็นโปรแกรมที่สามารถใช้งานได้อย่างอิสระ เหมือนกับการเขียนโปรแกรมภาษาสูงอื่นๆสามารถนำไปใช้งานบนระบบต่างๆได้ การใช้งาน โปรแกรมที่เขียนจำเป็นต้องทำงานภายใต้สภาพแวดล้อมของ JAVA (JAVA Virtual Machine) ซึ่งถูกกำหนดให้ทำงานครอบงำอยู่บนระบบที่ต้องการใช้งาน โปรแกรม JAVA นั้นๆ การใช้งาน JAVA นั้นตัวโปรแกรมจะถูกคอมไพล์เป็นแฟ้มที่มีนามสกุล class ซึ่งภายในจะเป็นคำสั่งใน JAVA ที่ถูกย่อเป็นการทำงานอย่างง่ายเรียกว่า “Byte code” การเรียกใช้งานแฟ้มนี้จะสามารถทำได้ใน สภาพแวดล้อมของ JAVA ตามที่ได้กล่าวมา ซึ่งเพราะเหตุนี้จึงทำให้ JAVA สามารถทำงานบนระบบใดๆก็ได้ ที่สนับสนุน JAVA

JAVA Application คือโปรแกรมที่ถูกสร้างขึ้นเพื่อทำงาน โดยถูก java.exe ทำการอินเทอร์พรีท (Interpret) ถือว่าเป็นโปรแกรมที่สมบูรณ์เนื่องจากไม่จำเป็นต้องมีโปรแกรมอื่นช่วยในการทำงาน สามารถควบคุมการดำเนินไปของตัวเองได้ สำหรับตัว java.exe นั้นเป็น ตัวสร้างสภาพแวดล้อมของ JAVA ตัวหนึ่งของบริษัท Sun Microsystems inc. ที่เรียกรวมๆว่า JDK (JAVA Development Kit) ซึ่งใน JDK จะทั้งส่วนที่ใช้ในการสร้างและอินเทอร์พรีท (Interpret) JAVA รวมอยู่ด้วยกันซึ่งการติดตั้งและการใช้งานโปรแกรมใน JDK นั้นจะมีรวมไว้ในภาคผนวก

ลักษณะทั่วไปของโปรแกรม JAVA นั้นคือ

1. มีคลาสอย่างน้อย 1 คลาส อยู่ในโปรแกรม (ใน 1 โปรแกรมจะสามารถมีคลาสได้มากกว่า 1 คลาสแต่การเรียกใช้เมธอดในแต่ละคลาสจะแตกต่างกันไป) และจะต้องเป็น แอพพลิเคชันคลาส เท่านั้น ซึ่งหมายถึงคลาสที่ใช้ทำงาน
2. ภายใน 1 โปรแกรมจะมีคลาสที่เป็น Public เพียงคลาสเดียวเท่านั้น ทั้งนี้ ชื่อคลาสที่เป็น Public จะต้องเป็นชื่อเดียวกับชื่อโปรแกรมเสมอ
3. ภายในโปรแกรมจะต้องมีเมธอดที่ชื่อ “main” เสมอ โดยเมธอด “main” นี้จะเป็นเมธอดแรกที่โปรแกรมเรียกใช้ ซึ่งภายใน “main” จะสามารถไปเรียกเมธอดอื่นได้
4. ก่อนการเรียกใช้เมธอดใดๆ จะต้องสร้างวัตถุของคลาสนั้นขึ้นมาก่อน

การใช้งาน JAVA แอพพลิเคชัน

1. โปรแกรมภาษา JAVA จะต้องเก็บในแฟ้มที่มีนามสกุล .java เมื่อถูกแปลแล้วจะได้แฟ้มนามสกุล .class ที่มีชื่อเหมือนกับโปรแกรมที่กำหนดไว้โดยจะได้ 1 คลาสต่อโปรแกรมภาษา JAVA 1 โปรแกรม
2. เราสามารถแปล โปรแกรมหลายๆโปรแกรมได้พร้อมๆกันได้หากมีแฟ้มภาษา JAVA หลายๆโปรแกรมอยู่ในที่เก็บเดียวกันเช่น javac *.java
3. การใช้งานโปรแกรม JAVA นั้นใช้งานได้ทั้งบน DOS และ บนวินโดวส์สามารถใช้งานส่วนประกอบต่างๆบนวินโดวส์ได้และสามารถใช้งานโปรแกรมที่ออกแบบมาใช้งานบนอินเทอร์เน็ตได้อย่างสมบูรณ์แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การเรียกใช้โปรแกรมที่ได้ทำการแปลแล้วจะสามารถเรียกผ่านหน้าต่าง DOS โดยการใช้คำสั่ง `java.exe` โดยที่จะมีการแสดงผลออกทางหน้าต่างของ DOS และ เป็นโปรเซสของวินโดวส์โดยใช้คำสั่ง `javaw.exe` ในที่นี้จำเป็นต้องใช้การทำงานเป็นแบบหน้าต่างบนวินโดวส์เท่านั้น

```

JAVA
Auto
Usage: JAVA.EXE [-options] class [args...]
        (to execute a class)
or JAVA.EXE -jar [-options] jarfile [args...]
        (to execute a jar file)

where options include:
-cp <directories and zip/jar files separated by ;>
    set search path for application classes and resources
-D<name>=<value>
    set a system property
-verbose[:class|gc|jni]
    enable verbose output
-version
    print product version
-? -help
    print this help message
-X
    print help on non-standard options

C:\jdk1.2.2\demo\jfc\SimpleExample>edit readme.rxr

C:\jdk1.2.2\demo\jfc\SimpleExample>java -jar SimpleExample.jar
Font specified in font.properties not found [Cordia New]
Font specified in font.properties not found [Browallia New]
Font specified in font.properties not found [Browallia New]
Font specified in font.properties not found [Angsana New]
Font specified in font.properties not found [Courier New]

```

รูปที่ 4-5 แสดงการเรียกใช้โปรแกรมผ่านหน้าต่าง DOS



รูปที่ 4-6 ผลลัพธ์จากรูปที่ 4-5

ตัวอย่างโปรแกรมบางส่วน JAVA แอปพลิเคชันที่แสดงผลลัพธ์

```

/*
 * @(#)SimpleExample.java      1.23 99/04/23
 */
import java.awt.*;
import java.awt.event.*;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void updateState() {
    String lmfName = UIManager.getLookAndFeel().getClass().getName();
    if (lmfName.indexOf(metal) >= 0) {
        metalButton.setSelected(true);
    } else if (lmfName.indexOf(windows) >= 0) {
        windowsButton.setSelected(true);
    } else if (lmfName.indexOf(motif) >= 0) {
        motifButton.setSelected(true);
    } else {
        System.err.println("SimpleExample is using an unknown L&F: " + lmfName);
    }
}

public static void main(String s[]) {
    SimpleExample panel = new SimpleExample();
    frame = new JFrame("SimpleExample");
    frame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {System.exit(0);}
    });
    frame.getContentPane().add("Center", panel);
    frame.pack();
    frame.setVisible(true);

    panel.updateState();
}
}

```

โปรแกรมภาษา JAVA ที่ได้จะสามารถนำไปใช้งานที่ใดก็ได้ที่ JAVA สนับสนุนแต่เนื่องจาก JAVA มีการพัฒนาตลอดเวลาจึงทำให้เกิดมาตรฐานขึ้นมาหลายๆ แบบเช่น JDK 1.0, JDK1.1 และ JDK 1.2 และจากหลากหลายระบบปฏิบัติการเช่นในระบบของวินโดวส์ซึ่งได้พัฒนา JAVA ในแนวทางของตัวเองขึ้นมา จึงทำให้การนำไปใช้งานยังคงไม่ราบรื่นอยู่บาง รวมทั้งความล่าช้าในการเรียกระบบขึ้นทำงานและการทำงานที่ยังไม่สามารถจัดการทำงานของโปรแกรมแบบมาตรฐานนี้ได้จึงทำให้การใช้งาน JAVA แอปพลิเคชันยังไม่น่าสนใจในตอนนี้แต่ในอนาคตยังคงต้องพิสูจน์ตัวเองต่อไปว่ามีคุณค่าพอในการใช้งานหรือไม่

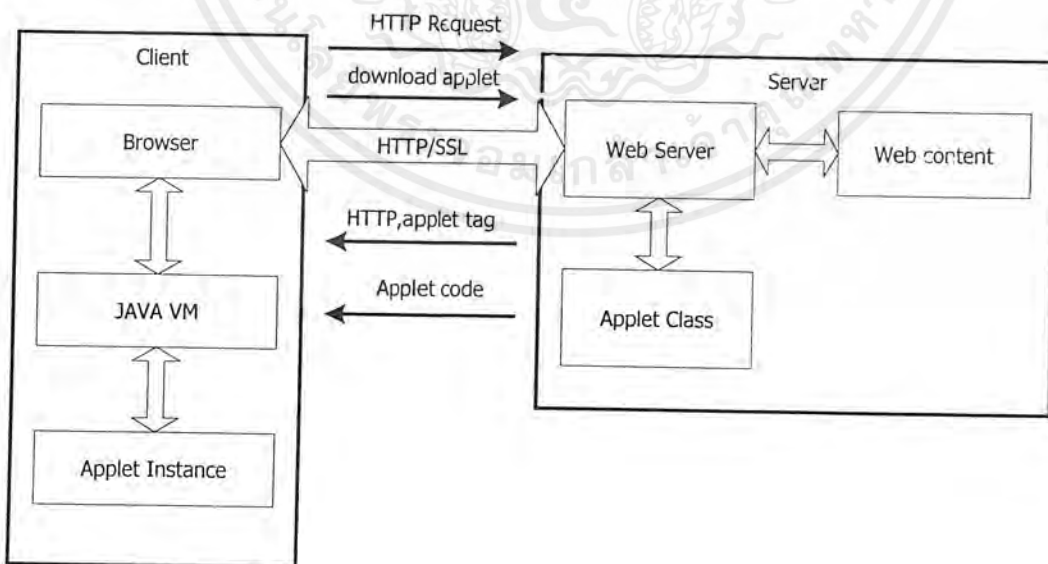
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 JAVA Applet

แอปเพล็ต (Applet) เป็นโปรแกรม JAVA อีกชนิดหนึ่งที่มีจุดประสงค์จะนำมาใช้ร่วมกับ เว็บเพจเพื่อให้ผลลัพธ์จาก เว็บเพจมีประสิทธิภาพดียิ่งขึ้น แต่แอปเพล็ตก็มีขีดความสามารถและกำหนดต่างๆที่จำกัดกว่า Application รวมทั้งด้านการป้องกันความปลอดภัยของข้อมูล

แอปเพล็ตเป็นโปรแกรมภาษา JAVA ที่ถูกสร้างขึ้นมาเพื่อทำงานภายใต้ เว็บเบราว์เซอร์ (Web Browser) เมื่อถูกเรียกจาก HTML ที่ถูกเรียกใช้แล้วก็จะถูกเว็บเบราว์เซอร์ดึงเข้ามาทำงานภายใต้สถานะแวดล้อมของเว็บเบราว์เซอร์ดังนั้นโปรแกรมภาษา JAVA ที่เป็น แอปเพล็ตจะไม่สามารถควบคุมการทำงานของตัวเองได้อย่างอิสระทั้งหมดแต่จะถูกควบคุมตามสถานะของเว็บเบราว์เซอร์หน้าที่มีแอปเพล็ตอยู่ การที่จะทำให้โปรแกรมเป็นแอปเพล็ตนั้นจะต้องมีการใส่แพ็คเกจที่ควบคุมการทำงานของแอปเพล็ตด้วยการใส่ประโยค "import java.applet.*" เพื่อให้คอมไพเลอร์มองเห็นคลาสแอปเพล็ตซึ่งจำเป็นต้องมีทุกๆแอปเพล็ตด้วย และประโยค "import java.awt.*; เพื่อให้คอมไพเลอร์มองเห็นคลาส Abstract Windows Toolkits ซึ่งเป็นคลาสที่สำคัญในการเขียนโปรแกรมให้ทำงานในสภาพแวดล้อมที่เป็นกราฟฟิก โดยจะสังเกตว่าแอปเพล็ตไม่มี main () ที่มีใน JAVA แอปเพล็ตขึ้นเป็นจุดเริ่มต้นในการทำงาน

แอปเพล็ตจะมีการสร้างไม่แตกต่างไปจากการสร้าง JAVA แอปเพล็ตขึ้นซึ่งเมื่อแปลแล้วจะได้เพิ่ม .class เช่นเดียวกันแต่การทำงานของแอปเพล็ตจะไม่สามารถถูกเรียกใช้จาก java.exe ได้แต่จะต้องเรียกผ่านแอปเพล็ตแท็ก (Applet tag) ในภาษา HTML ซึ่งจะมียูอยู่บนเว็บเบราว์เซอร์ที่สนับสนุน JAVA ทุกตัว หรือมิฉะนั้นจะสามารถเรียกผ่านโปรแกรม Appletviewer.exe ที่เป็น โปรแกรมที่รวมอยู่ในชุด JDK ซึ่งเป็น โปรแกรมที่ใช้สำหรับทดสอบแอปเพล็ตแต่การทำงานบน Appletviewer.exe นี้จะไม่สามารถทำการทดสอบการทำงานของแอปเพล็ตที่มีการติดต่อกับระบบเครือข่ายได้



รูปที่ 4-7 แสดงการทำงานของ Applet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 APPLET TAG

แอปเพล็ตแท็ก (Applet tag) เป็นข้อความที่อยู่ระหว่าง <APPLET> กับ </APPLET> ใช้สำหรับเรียกแอปเพล็ตให้ทำงานผ่าน HTML รูปแบบของการใช้แอปเพล็ตแท็กมีดังนี้

```
<applet
    code=applet.class
    width=xxx
    height=xxx
    [codebase=URL]
    [name=applet_name]
    [vspace=xxx]
    [hspace=xxx]
    [alt=any_text]>
    [<param name=parameter_name value=parameter_value>*]
    [HTML text display in absent of JAVA ]
</applet>
```

หมายเหตุ [] หมายถึงอาจมีหรือไม่มีก็ได้

สำหรับความหมายของส่วนประกอบของแท็กแต่ละตัวคือ

1. Code ใช้ระบุแอปเพล็ตที่จะถูกเรียกใช้งานซึ่งเป็นแฟ้ม .class หากเป็นแอปเพล็ตที่อยู่ในเว็บไซต์อื่นจะต้องระบุที่อยู่ให้ครบด้วยแต่มีข้อจำกัดคือจะต้องเป็นไซต์ที่เป็นที่เดียวกับแฟ้ม HTML ที่เรียกใช้งานมิฉะนั้นจะไม่สามารถทำงานได้ อันเนื่องมาจากข้อจำกัดของตัวแอปเพล็ตเอง
2. Width ใช้ระบุความยาวของพื้นที่ในแนวนอนของที่ใช้แสดงแอปเพล็ตในเลขไซต์นั้นๆ
3. Height ใช้ระบุความยาวของพื้นที่ในแนวตั้งของที่ใช้แสดงแอปเพล็ตในเลขไซต์นั้นๆ
4. Codebase ใช้ระบุตำแหน่งหลักที่ใช้กับแอปเพล็ตนั้นๆ ซึ่งเป็นที่เก็บแอปเพล็ตที่ใช้งานในขณะนั้น หากไม่ระบุจะกำหนดเป็นตำแหน่งเดียวกับแฟ้ม HTML
5. Name ใช้กำหนดชื่อให้แอปเพล็ต ใช้ในกรณีที่มีการใช้งานแอปเพล็ตหลายๆ ตัวในในหน้าเว็บเพจเดียวกัน ซึ่งจะทำให้สามารถอ้างอิงถึงอีกแอปเพล็ตตัวอื่นๆอีก ในการจะได้อ้างอิงแอปเพล็ตอีกตัวจะต้องใช้ `getApplet()` เพื่อควบคุมการทำงาน
6. Align ใช้ระบุการวางตำแหน่งของแอปเพล็ตในพื้นที่ของแอปเพล็ต ซึ่งคล้ายกับ IMG แท็ก ของ HTML โดยมีค่าที่เป็นไปได้คือ LEFT, RIGHT, TOP, BOTTOM, MIDDLE, BASELINE, TEXTTOP, ABSMIDDLE และ ABSBOTTOM
7. Vspace และ hspace ใช้ระบุช่องว่างระหว่างแอปเพล็ตกับข้อความในเว็บเพจโดยระบุเป็นจำนวน จุด การใช้งานจะเป็นลักษณะเดียวกับ IMG แท็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Alt ใช้แสดงข้อความในกรณีที่เว็บเบราว์เซอร์นั้นๆไม่สามารถแสดงแอปเพล็ตได้ แต่เข้าใจการทำงานของแอปเพล็ตแท็กได้ โดยอาจเกิดจากปัญหาที่ตัวแปลภาษา JAVA ในระบบที่แอปเพล็ตทำงานอยู่ หรือ ปัญหาอื่นๆ

การส่งพารามิเตอร์จาก HTML ไปยังแอปเพล็ตสามารถใช้ PARAM แท็กซึ่งอยู่ในแอปเพล็ตแท็ก เป็นการส่งค่าที่เป็นสตริงให้กับแอปเพล็ตเพื่อใช้งานโดยมีรูปแบบดังนี้

```
<PARAM name=parameter_name value=parameter_value>
```

ในหนึ่งแท็กสามารถส่งตัวแปรได้หนึ่งตัวแต่ในหนึ่งแอปเพล็ตแท็กสามารถส่ง PARAM แท็กเท่าใดก็ได้ การนำข้อมูลจากแท็กมาใช้งานในแอปเพล็ตสามารถทำได้โดยใช้เมธอด

```
public String getParameter(String name);
```

ตัวอย่างการส่งพารามิเตอร์จาก HTML ไปยังแอปเพล็ต

```
//parameter test in JAVA
import java.awt.*;
import java.applet.*;
public class ParamTest extends Applet {
    String m;
    Public void init() {
        String f = getParameter("font");
        Int S = Integer.parseInt(getParameter("size");
        SetFont = (new Font (f, Font.BOLD, s));
        M = getParameter("msg");
    }
    public void paint(Graphics g)
}
/* <applet code="ParamTest.class" width=200 height=100>
<param name=font value="TimeRoman">
<param name=size value="26">
<param name=msg value="Hello">
</applet> */
```

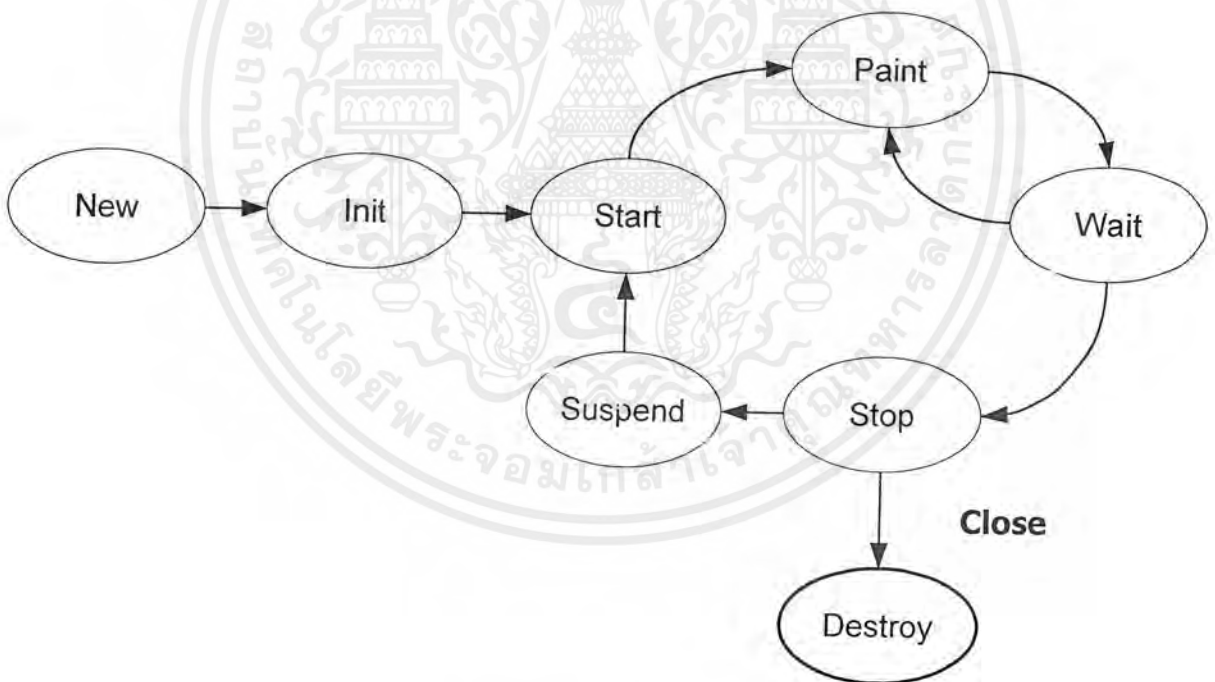
ประโยชน์ของการใช้ PARAM แท็กคือการใช้แอปเพล็ตตัวเดียวกันแสดงข้อมูลหลายๆครั้ง การแสดงข้อมูลที่แตกต่างกันหรือการส่งข้อมูลจาก Server size โปรแกรมเช่นเซิร์ฟฟเล็คเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 โครงสร้างของ Applet

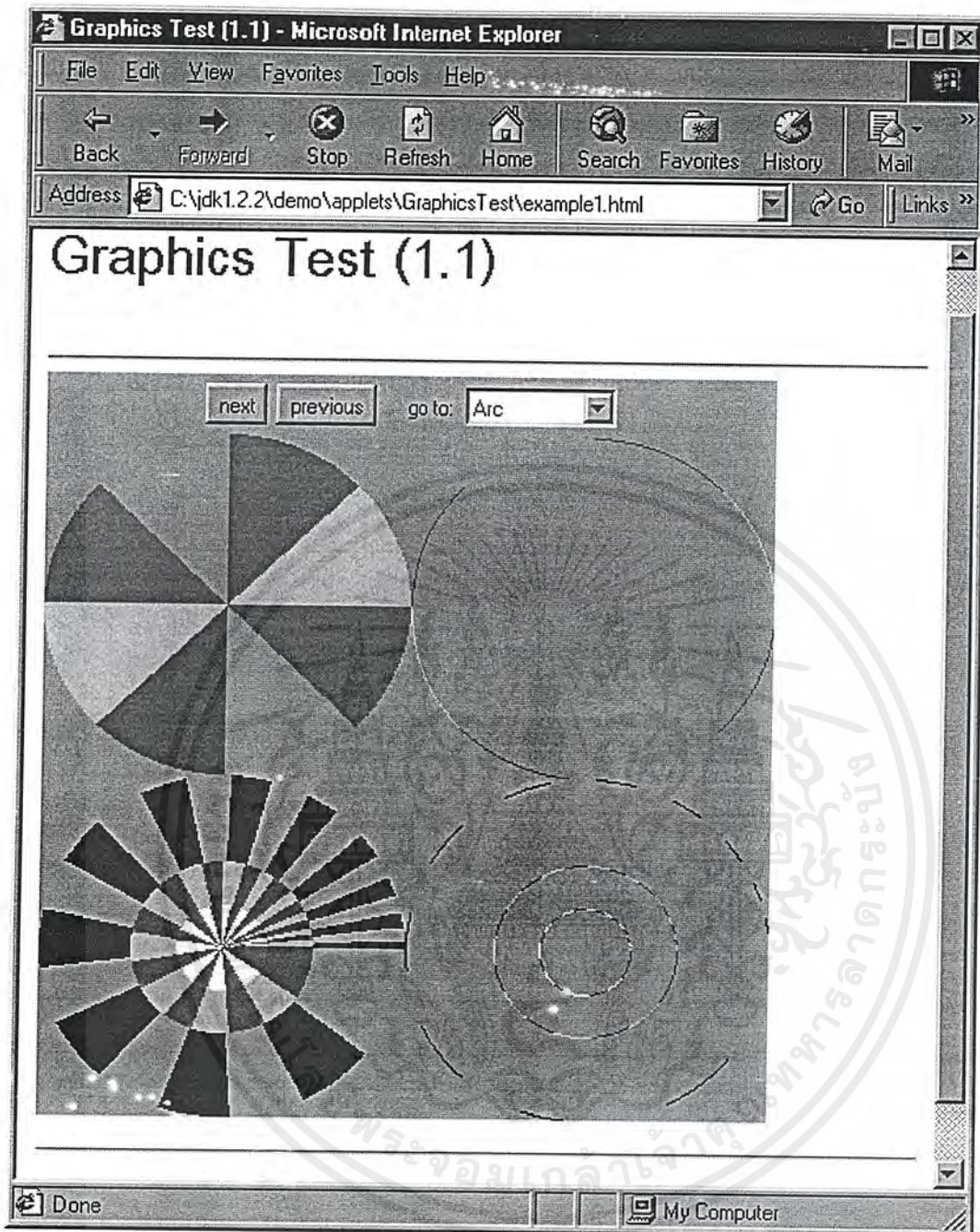
แอปเพล็ตแตกต่างจากแอปพลิเคชันที่ไม่ได้เริ่มต้นการทำงานโดยการเรียกใช้เมธอดชื่อ “main” แต่จะมีโครงสร้างเป็นของตัวเอง จะประกอบด้วยเมธอดดังต่อไปนี้

- `init()` เป็นเมธอดที่ถูกเรียกทันทีที่แอปเพล็ตนั้นเข้ามาทำงานเว็บเบราว์เซอร์ โดยจะถูกเรียกเพียงครั้งเดียวเท่านั้น มักใช้ในการกำหนดค่าเริ่มต้นให้กับโปรแกรม
- `start()` จะถูกเรียกก็ต่อเมื่อ `init()` เมธอดถูกเรียกใช้งานแล้วเท่านั้น โดยเมื่อเข้าสู่การทำงานของเว็บเพจ ใดๆ `start()` เมธอดจะถูกเรียกใช้งานด้วย และจะถูกเรียกซ้ำเมื่อกลับมายังหน้าเว็บเพจเดิม
- `stop()` ถูกเรียกใช้งานเมื่อมีการย้ายไปยังหน้าเว็บเพจอื่นๆ หรือเมื่อนั้นถูกปิดลงชั่วคราว โดยจะหยุดการแสดงผลติดต่อกับผู้ใช้ ซึ่งเทรคต่างๆที่ทำงานอยู่จะยังคงทำงานต่อไป
- `destroy()` เรียกใช้เมื่อต้องการออกจากเว็บเบราว์เซอร์ โดยจะทำการคืนทรัพยากรของระบบให้ด้วย แต่ต้องเรียก `stop()` เมธอดก่อนเสมอ
- `paint()` เป็นเมธอดที่ใช้กับการทำงาน แบบกราฟฟิกในแอปเพล็ต โดยจะเรียกใช้เมื่อต้องการเขียนหน้าเว็บเพจนั้นๆ ใหม่อีกครั้ง



รูปที่ 4-8 แสดงวงจรชีวิตของแอปเพล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-9 แสดงตัวอย่างของแอปเพล็ต

ตัวอย่างของโปรแกรมบางส่วนของแอปเพล็ตที่แสดงในรูปที่ 4-8

```

/*
 * @(#)AppletFrame.java 1.3 96/12/06
 * Copyright (c) 1997 Sun Microsystems, Inc. All Rights Reserved.
 */
import java.awt.Frame;
import java.awt.Event;
import java.awt.Dimension;
import java.applet.Applet;
import java.awt.AWTEvent;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Applet to Application Frame window
class AppletFrame extends Frame
{
    public static void startApplet(String className, String title, String args[])
    {
        // local variables
        Applet a;
        Dimension appletSize;
        try
        {
            // create an instance of your applet class
            a = (Applet) Class.forName(className).newInstance();
        }
        catch (ClassNotFoundException e) { return; }
        catch (InstantiationException e) { return; }
        catch (IllegalAccessException e) { return; }

        // initialize the applet
        a.init();
        a.start();

        // create new application frame window
        AppletFrame f = new AppletFrame(title);

        // add applet to frame window
        f.add("Center", a);

        // resize frame window to fit applet
        // assumes that the applet sets its own size
        // otherwise, you should set a specific size here.
        appletSize = a.getSize();
        f.pack();
        f.setSize(appletSize);

        // show the window
        f.show();
    } // end startApplet()

    // constructor needed to pass window title to class Frame
    public AppletFrame(String name)
    {
        // call java.awt.Frame(String) constructor
        super(name);
    }

    // needed to allow window close
    public void processEvent(AWTEvent e)
    {
        // Window Destroy event
        if (e.getID() == Event.WINDOW_DESTROY)
        {
            // exit the program
            System.exit(0);
        }
    } // end handleEvent()
} // end class AppletFrame

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ข้อจำกัดของ Applet

โดยทั่วไปแล้วแอปพลิเคชันจะถูกเรียกใช้งานจากเครือข่ายซึ่งไม่อนุญาตให้มีการเข้าถึงเครื่องคอมพิวเตอร์เครื่องอื่นได้ เพื่อความปลอดภัยจึงกำหนดให้จำกัดการทำงานบางประการของแอปพลิเคชันดังนี้

- ไม่สามารถออกคำสั่งเรียกใช้งานแฟ้ม ลบ เขียน ตรวจสอบรายชื่อแฟ้มข้อมูล และเรียกใช้งานโปรแกรมต่างๆของเครื่องลูกข่ายได้
- ไม่สามารถออกจากระบบ หรือปิดการทำงานได้โดยตัวเอง การปิดโปรแกรมจะขึ้นกับการทำงานของบราวเซอร์เท่านั้น
- ไม่สามารถทำการติดต่อกับเครือข่ายได้ ยกเว้นย้อนกลับไปยัง Host ที่ Applet เข้ามา แต่จำกัดมิให้เรียกใช้งานพอร์ตที่หมายเลขสูงกว่า 1024
- ไม่สามารถโหลดไลบรารีต่างๆ ขึ้นมาใช้งาน หรือการเรียกใช้โปรแกรมในระบบของเครื่องที่มีใช้ภาษา JAVA
- จำกัดการทำงานของ AWT (Abstract Windows Toolkits) บางตัวเช่น ห้ามเรียกใช้งานเครื่องพิมพ์ ห้ามใช้ Clipboard
- ไม่อนุญาตให้เข้าถึงข้อมูลของระบบ (System Properties) โดยสามารถอ่านได้บางตัวแต่ต้องกำหนดให้สามารถอ่านได้จากระบบบราวเซอร์เท่านั้น

แอปพลิเคชันถูกออกแบบมาเพื่อให้ใช้บนอินเทอร์เน็ตโดยใช้ร่วมกับระบบเครือข่าย WWW โดยตรง สามารถใส่ลงในเครื่องแม่ข่ายของเว็บได้โดยตรง ช่วยให้การพัฒนาระบบงานบนเครือข่าย WWW สามารถทำได้ง่ายและสะดวกมากยิ่งขึ้น

4.4 JAVA Network

เนื่องจาก JAVA ถูกออกแบบมาเพื่อใช้งานกับ Internet โดยเฉพาะ จึงมีชุดคำสั่งสำหรับการเชื่อมต่อเครือข่ายรวมอยู่ด้วยการใช้งานมีให้ใช้งานได้หลากหลายแบบตามความเหมาะสม โดยที่ภาษา JAVA ในปัจจุบันทำงานบนระบบเครือข่ายแบบ TCP/IP เท่านั้นยังไม่พบว่ามีการใช้งานในระบบเครือข่ายที่มีโพรโทคอลแบบอื่น

ระบบการเชื่อมโยงกันมิได้หลายลักษณะตามการใช้งานแบ่งเป็น

4.4.1 Socket

เป็นการเชื่อมต่อแบบพื้นฐาน โดยจะทำการแลกเปลี่ยนข้อมูลกันได้ตลอดราบเท่าที่ยังเชื่อมต่อกันอยู่ ตัวอย่างของการเชื่อมต่อแบบ Socket คือ การใช้ TELNET และ FTP

1. สร้างช่องทางการเชื่อมต่อ (Socket) ขึ้นมาและขอการเชื่อมต่อไปยังเครื่องแม่ข่าย
2. เครื่องแม่ข่ายตรวจสอบการเข้าเชื่อมต่อ หากได้รับอนุญาตก็จะคงการเชื่อมต่อนั้นไว้ มิฉะนั้นก็จะทำการยกเลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการรับส่งข้อมูลในที่นี้สามารถส่งข้อมูลได้ทั้งสองทางโดยข้อมูลจะถูกเก็บไว้ในที่เก็บชั่วคราวจนกว่าโปรแกรมจะรับไปประมวลผล
4. เมื่อต้องการยกเลิกการติดต่อก็ทำได้โดยการส่งคำสั่งขอยกเลิกการติดต่อ

ตัวอย่างโปรแกรมที่ใช้ Socket

```
import java.net.*;
import java.io.*;

public class Whois {

    public final static int port = 43;
    public final static String hostname = "whois.internic.net";

    public static void main (String args [ ]) {

        Socket theSocket;

        Try {
            Socket = new Socket (hostname, port , true);
            Ps = new printStram (theSocket.getOutputStream( ));
            for (int i = 0 ;i < length; I++)ps.print(args[I] +” “);
            ps.print(“\r\n”);
            theWhoisStream = new DataInputStream (theSocket.getInputStream( ));
            String s;
            while ((s = theWhoisStream.readLine( )) != null) {
                System.out.println(s);
            }
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

4.4.2 URLConnection

เป็นการเชื่อมต่อเครือข่ายแบบที่ใช้การแบบชั่วคราวที่ใช้การกำหนดตำแหน่งข้อมูล URL (Universal Resource Location) ลักษณะแบบนี้เทียบได้กับการเชื่อมต่อใน WWW ที่เครื่องลูกข่ายไม่จำเป็นต้องเชื่อมต่ออยู่ตลอดเวลาโดยจะทำการเชื่อมต่อเฉพาะเมื่อต้องการข้อมูลเท่านั้น มีขั้นตอนในการทำงานดังนี้

1. เครื่องลูกข่ายจะทำการร้องขอข้อมูล (Client Request) ซึ่งระบบเครือข่ายจะทำการกำหนดเส้นทางการเชื่อมต่อจากข้อมูล URL ที่ได้จากเครื่องลูกข่าย และตรวจสอบการเชื่อมต่อได้จากข้อมูลที่รับมาจากเครื่องแม่ข่าย ทำการเชื่อมต่อเพื่อการรับส่งข้อมูล โดย URL ที่ส่งไปอาจแนบข้อมูลบางส่วนไปด้วย (GET Method)
2. ทำการแลกเปลี่ยนข้อมูล โดยเครื่องลูกข่ายจะทำการส่งข้อมูลให้เครื่องแม่ข่ายตีความ ในกรณีที่ใช้ GET Method จะมีเฉพาะส่วนหัวของข้อมูล (Header) ในกรณีที่ใช้การแลกข้อมูล (POST Method) จะทำการส่งข้อมูลเป็นกระแสข้อมูล (DataStream)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เครื่องแม่ข่ายทำการส่งข้อมูลกลับเป็นกระแสข้อมูลเช่นกัน ให้เครื่องลูกข่ายรับไปประมวล โดยข้อมูลไม่จำเป็นต้องเป็นข้อมูลจากแหล่งเดียวอาจส่งมาจากหลายๆ แหล่งได้
4. ทำการยกเลิกการเชื่อมต่อเมื่อได้รับข้อมูลครบแล้ว

URLConnection เป็นการเชื่อมต่อที่ง่ายและสะดวกต่อการใช้งานเหมาะสำหรับการใช้งานที่มีการเรียกใช้ข้อมูลไม่ถี่มากเช่นการเรียกข้อมูลบนเว็บ

ตัวอย่างโปรแกรมที่ใช้ URLConnect

```
import java.net.*;
import java.io.*;

public static void main (String args [ ]) {
String thisLine
URL u;

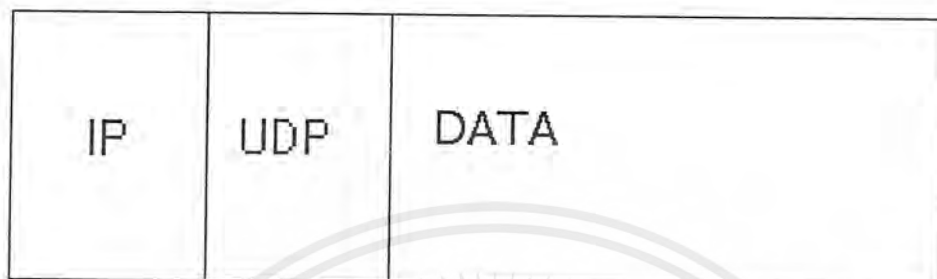
If (args.length > 0) {
Try {
u = new URL(args[0]);
try {
DataInputStream theHTML = new DataInputStream(u.openStream( ));
try {
while ((thisLine = theHTML.readLine( )) != null) {
System.out.println(thisLine);
}
}
} catch (Exception e) {
System.out.println(e);
}
} catch (Exception e) {
System.out.println(e);
}
} catch (MalformedURLException e){
}
}
}
```

4.4.3 DatagramSocket

เป็นการเชื่อมต่อแบบที่มีการกำหนดตัวห่อข้อมูล (Datagram) ที่อยู่ใน Internet Protocol ได้การเชื่อมต่อแบบนี้จะได้เปรียบการเชื่อมต่อแบบ TCP ตรงที่ขนาดข้อมูลจะไม่ถูกกำหนดตายตัวทำให้การส่งข้อมูลทำได้รวดเร็วกว่าการส่งแบบที่ขนาดของห่อข้อมูล (Datagram) คงที่ซึ่งโดยปกติแล้วขนาดของห่อข้อมูลแม้จะสามารถปรับขนาดได้ แต่ใน TCP จะต้องส่งข้อมูลในขนาดที่เท่ากันทุกครั้ง ในขณะที่ข้อมูลมีขนาดไม่ใหญ่มากแต่มีการแลกเปลี่ยนข้อมูลกันบ่อยครั้งการใช้ UDP จะทำให้การส่งข้อมูลทำได้เร็วเพราะมีขนาดเล็กส่งให้หมดได้โดยใช้เวลาน้อย และการที่มีขนาดของห่อข้อมูลที่เล็กทำให้โอกาสที่ข้อมูลจะผิดพลาดก็น้อยตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีอีกประการของ UDP ก็คือการส่งข้อมูลไม่จำเป็นต้องทำการเชื่อมต่อโดยตรง สามารถกำหนดจุดหมายของข้อมูลเพราะไม่จำเป็นต้องทำการตรวจรับข้อมูล (Handshaking) ตลอดเวลา หากจะเปรียบเทียบกับ การส่งพัสดุ ฉะนั้นการส่งแบบ UDP จะเหมาะกับการส่งข้อมูลในระบบที่ไม่เสถียร แต่ UDP จะไม่เหมาะกับการส่ง ข้อมูลขนาดใหญ่เพราะส่วนหัวที่ต้องมีทุกครั้งที่จะส่งข้อมูลจะเพิ่มภาระ (Overhead) ของระบบทำให้ประสิทธิภาพต่ำ ลง



รูปที่ 4-10 แสดงของห่อข้อมูลแบบ UDP

ตัวอย่างโปรแกรมแบบ UDP

```
import java.net.*;

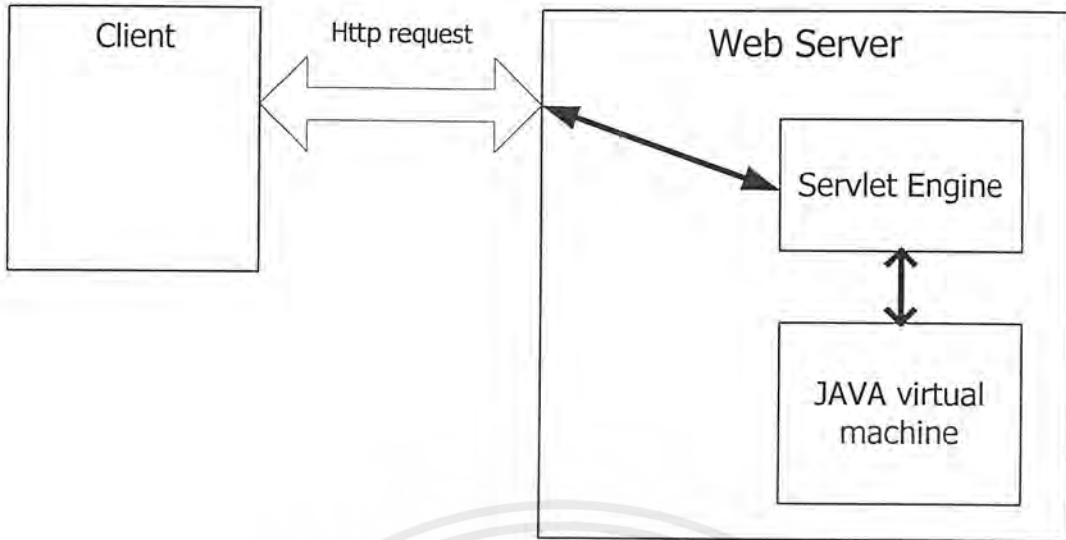
public class UDPSender {

    public void main (String args[ ]) {
        String s = "This is a test";
        byte [] data = new byte[s.length()];
        s.getBytes(0 , s.length() , data , 0);
        try {
            InetAddress ia = InetAddress.getByAddress("sunsite.unc.edu");
            Int port = 7;
            DatagramPacket dp = new DatagramPacket(data , data.length , ia , port);
        }
        catch (UnknownHostException e) {
        }
    }
}
```

4.5 JAVA Servlet

เซิร์ฟเว็ตคือระบบของ JAVA แอปพลิเคชันที่ถูกกำหนดให้ทำงานบนเครื่องฝั่งแม่ข่ายได้ ซึ่งเป็นการเพิ่มความสามารถในการทำงานของเครื่องแม่ข่ายเว็บ (Web Server) โดยสามารถใช้ความสามารถของ JAVA ได้ จึงมีคุณสมบัติของ JAVA ให้ใช้ไม่ว่าจะเป็นการที่โปรแกรมมีขนาดเล็ก สามารถใช้ในระบบที่หลากหลาย มีความปลอดภัยสูง และ ยังสามารถใช้คลาสภาษา JAVA ได้ทันที กอปรกับที่ปัจจุบันได้มีองค์กรหลายๆองค์กรนำ JAVA และ Servlet เข้าไปใช้ในองค์กรมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-11 แสดงการทำงานของเซิร์ฟเวสต์

4.5.1 ระบบที่ใช้ทำงานของเซิร์ฟเวสต์

การทำงานของเซิร์ฟเวสต์จะเป็นแบบเดียวกับโปรแกรม JAVA ทั่วไปที่ต้องทำงานอยู่บนสภาพแวดล้อมของ JAVA แต่จะแตกต่างกันที่เซิร์ฟเวสต์จะถูกบังคับให้ทำงานบนเครื่องแม่ข่ายเท่านั้น และการแสดงผลลัพธ์ จะออกมาทางเว็บเบราว์เซอร์ ซึ่งการควบคุมการทำงานของเซิร์ฟเวสต์จะอยู่บนระบบควบคุมเซิร์ฟเวสต์ (Servlet Engine) หากเปรียบเทียบก็จะเหมือนกับแอปพลิเคชัน แต่เนื่องจากเซิร์ฟเวสต์จะทำงานอยู่บนเครื่องแม่ข่ายซึ่งมีการรักษาความปลอดภัยอยู่แล้วดังนั้นข้อจำกัดต่างๆที่มีอยู่บนแอปพลิเคชันจึงไม่มีในเซิร์ฟเวสต์

การทำงานของเซิร์ฟเวสต์จะขึ้นอยู่กับ Servlet Engine เป็นสำคัญ ตัว Servlet Engine เองสามารถแบ่งได้เป็น

1. ระบบที่ทำงานได้เอง (Standalone Servlet Engine) จะเป็นระบบที่เป็นเครื่องแม่ข่ายแล้วทำการรวบรวม Servlet Engine เข้าไปด้วย การควบคุมการทำงานก็สามารถควบคุมได้จากจุดเดียวกันทั้งเครื่องแม่ข่ายและ Servlet Engine ทำให้สะดวกและการทำงานมีประสิทธิภาพมากขึ้น แต่ก็มีข้อเสียเปรียบคือ JAVA จะมีการพัฒนาอยู่เรื่อยๆ ซึ่งจะทำให้ระบบล้าสมัยลงอย่างรวดเร็วการปรับปรุงจะทำได้ยากกว่าเพราะจะต้องทำการแก้ไขทั้งระบบเครื่องแม่ข่าย ตัวอย่างของระบบนี้เช่น

- Sun 's JAVA Web Server www.sun.com/products/Jeeves
- The WWW Consortium 's Jigsaw Server www.w3.org/Jigsaw
- O'Reilly 's Website Professional (Version 2.1 ขึ้นไป) www.oreilly.com
- Netscape's Enterprise Server www.netscape.com/download
- Lotus 's Domino Go Web Server www.lotus.com/dominogowebserver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ระบบแบบเพิ่มเข้าไป (Add-on Servlet Engine) เป็นโปรแกรมที่ติดตั้งภายหลังจากที่ติดตั้งเครื่องแม่ข่ายเว็บแล้วระบบนี้จะเป็นตัวเลือกที่ช่วยให้ไม่จำเป็นต้องผูกติดกับระบบใดระบบหนึ่งการทำงานสามารถทำงานได้ในแบบเดียวกัน แต่การควบคุมจะแยกจากตัวเครื่องแม่ข่ายเว็บตัวอย่างเช่น

- Apache Jserv www.apache.org
- Live Software 's Jrun www.livesoftware.com/products/jrun
- IBM's WebSphere www.software.ibm.com/websphere
- New Atlanta 's ServletExec www.newatlanta.com

4.5.2 การเรียกใช้งานเซิร์ฟเล็ต

การสั่งให้เซิร์ฟเล็ตทำงานนั้นทำได้หลายวิธี

- การเรียกตัวเซิร์ฟเล็ตโดยตรง

วิธีนี้จะเป็นการเรียกโปรแกรมเซิร์ฟเล็ตที่อยู่ในเครื่องแม่ข่ายโดยเรียกผ่าน URL ของเว็บเบราว์เซอร์ การเรียกโดยวิธีนี้จะไม่มีการส่งข้อมูลให้เซิร์ฟเล็ตมักใช้ในการทดสอบเซิร์ฟเล็ตเช่น <http://webserver.com:8080/servlets/HelloWorld>

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

public class HelloWorld extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        out.println("<HTML>");
        out.println("<HEAD><TITLE>Hello World</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("<BIG>Hello World</BIG>");
        out.println("</BODY></HTML>");
    }
}
```

- การเรียกใช้ผ่านแบบฟอร์มบนเว็บเบราว์เซอร์

เป็นการเรียกใช้ผ่านแบบฟอร์มที่สร้างขึ้นด้วยภาษา HTML การเรียกใช้ในแบบนี้จะมีวิธีส่งผ่านข้อมูล 2 แบบ คือ GET Method วิธีนี้จะส่งข้อมูลผ่านทาง URL โดยตรงซึ่งสะดวกและรวดเร็วแต่ข้อมูลจะไม่ปลอดภัยและไม่เหมาะกับการส่งข้อมูลจำนวนมาก และ POST Method ที่จะส่งข้อมูลหลังจากที่ทำการเชื่อมต่อ ซึ่งมีข้อดีคือการส่งข้อมูลจะปลอดภัยมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

form.html

```
<HTML>
<HEAD>
<TITLE>Introduction</TITLE>
</HEAD>
<BODY>
<FORM METHOD=GET ACTION="/dervlet/Hello">
If you don't mind me asking, what is your name?
<INPUT TYPE=TEXT NAME="name"><P>
<INPUT TYPE=SUBMIT>
</FORM>
</BODY>
</HTML>
```

Hello.java

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

public class HelloWorld extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        out.println("<HTML>");
        out.println("<HEAD><TITLE>Hello World</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("<BIG>Hello World</BIG>");
        out.println("</BODY></HTML>");
    }
}
```

- การเรียกผ่าน SERVER SIDE INCLUDE

ในระบบที่ใช้ HTML จะมีการเรียกเซิร์ฟเวอร์ที่ยุ่งยาก และ เนื่องจากเซิร์ฟเวอร์ มีข้อจำกัดน้อยกว่าแอปพลิเคชัน จึงควรเก็บข้อมูลของการเรียก เซิร์ฟเวอร์ นี้ไว้โดยใช้ Server Side Include (SSI) ซึ่งมีแท็ก <SERVLET> ที่ใช้เรียก เซิร์ฟเวอร์

Server Side Include เป็นแฟ้มที่มีลักษณะคล้าย HTML มีนามสกุลเป็น .SHTML แต่จะประมวลผลแท็กบางแท็กบนเครื่องแม่ข่าย ซึ่งจะช่วยให้ เซิร์ฟเวอร์ ทำงานได้ปลอดภัยขึ้น สามารถรับแท็ก <PARAM> ได้เช่นเดียวกับแอปพลิเคชัน

การที่เครื่องแม่ข่ายจะสามารถประมวลผลแท็กต่างๆ ในแบบ SSI ได้ นั่นตัวเครื่องแม่ข่ายจะต้องมีความสามารถในการเพิ่มโมดูลที่ช่วยเสริมการทำงานได้ ตัวโมดูลนี้จะช่วยให้เครื่องแม่ข่ายสามารถประมวลผลแท็กต่างๆ ได้เพิ่มขึ้น ซึ่งตัวโมดูลอาจได้จากผู้ผลิตเครื่องแม่ข่ายหรือจากผู้พัฒนาตัวโปรแกรมเสริมที่จะช่วยให้เครื่องแม่ข่ายทำงานร่วมกับโปรแกรมได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการเรียกใช้เซิร์ฟเล็ตโดยวิธี SSI

Time.shtml

```

<HTML>
<HEAD><TITLE>Times!</TITLE></HEAD>
<BODY>
The current time here is:
<SERVLET CODE=currentTime>
</SERVLET>
<P>
The current time in London is:
<SERVLET CODE=currentTime>
<PARAM NAME=zone VALUE=GMT>
</SERVLET>
<P>
And the current time in New York is:
<SERVLET CODE=currentTime>
<PARAM NAME=zone VALUE=EST>
</SERVLET>
<P>
</BODY>
</HTML>

```

currentTime.java

```

import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class CurrentTime extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        //PrintWriter out = res.getWriter();
        PrintWriter out = new PrintWriter(res.getOutputStream(), true);

        Date date = new Date();
        DateFormat df = DateFormat.getInstance();

        String zone = req.getParameter("zone");
        if (zone != null) {
            TimeZone tz = TimeZone.getTimeZone(zone);
            df.setTimeZone(tz);
        }

        out.println(df.format(date));
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.3 การพัฒนาโปรแกรมด้วย Servlets

การพัฒนาเซิร์ฟเล็ตก็ไม่ต่างไปจากการเขียนโปรแกรมภาษา JAVA และแอปเพล็ต ซึ่งจำเป็นต้องใช้ JDK ในการพัฒนาหรือใช้โปรแกรมสำเร็จรูปอื่นๆที่พัฒนาขึ้นมาเพื่อใช้กับ JAVA และเช่นเดียวกับแอปเพล็ตที่จำเป็นต้องใช้โปรแกรมทดสอบเช่น Appletviewer.exe แต่เนื่องจากเซิร์ฟเล็ตไม่ได้เป็นคลาสมาตรฐานสำหรับ JAVA จึงไม่มีระบบในการทำงานนั้นบน JDK จำเป็นต้องติดตั้งให้ใช้งานเฉพาะ ซึ่งสามารถหาได้จากเว็บของ SUN เรียกว่า JAVA Servlet Development Kit (JSDK) โดยใช้ตัวทดสอบเซิร์ฟเล็ตที่มากับโปรแกรมเรียกว่า Servletrunner.exe ซึ่งจะมีความสามารถในการตรวจสอบการทำงาน (Debug) ของเซิร์ฟเล็ตหรือในระบบบางตัวสามารถตั้งให้มีระบบช่วยตรวจสอบ (Debug Mode) มาช่วยในการพัฒนาได้

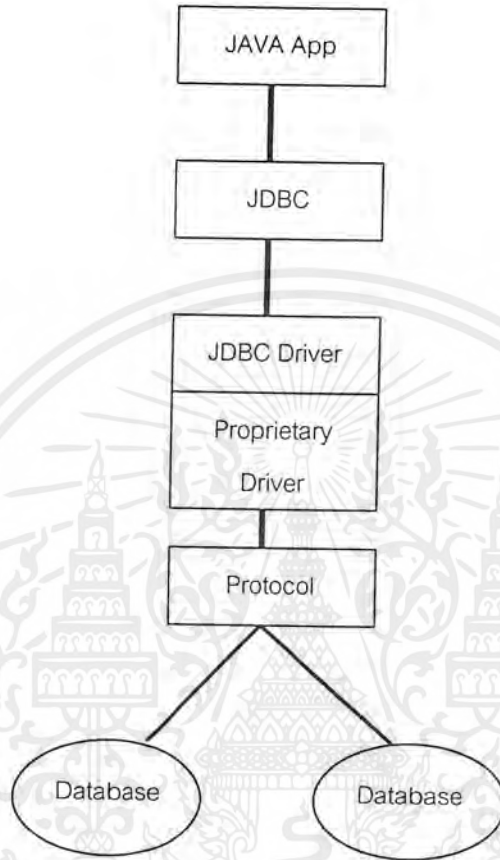
4.5.4 ข้อได้เปรียบของ Servlets

- 1 ระบบการรักษาความปลอดภัย
 - การเรียกใช้งานอยู่ในการควบคุมของ HTTP ซึ่งรวมการรักษาความปลอดภัยบนเว็บ
 - การควบคุมการเข้าถึง Back-tier ที่มีความสำคัญเช่น เครื่องแม่ข่ายฐานข้อมูล
 - ไม่ทำให้ระบบของเครื่องแม่ข่ายล้มเหลวได้ง่าย
- 2 การใช้งาน
 - สามารถใช้งาน API ของ JAVA ที่มีอยู่มากมายได้ทันที
 - สามารถใช้ RMI เพื่อการสื่อสารกันระหว่างโปรแกรมได้ง่ายและสะดวก
 - สามารถดูแลและปรับปรุงระบบได้ง่าย
- 3 ประสิทธิภาพ
 - การทำงานเป็นแบบ Dynamic Loading (ระบบใช้หน่วยความจำของ Server) และการใช้ instance แทนการแยกเป็น Process ช่วยให้ทำงานได้เร็วกว่า CGI
 - สนับสนุนการทำงานกับระบบ Multiprocessor
 - ใช้การเชื่อมต่อแบบ HTTP request ซึ่งเร็วกว่าการใช้ Socket ใน CGI
- 4 การใช้งานในหลายๆระบบ
 - สามารถใช้ JAVA เข้าไปแทนระบบต่างของเครื่องแม่ข่ายได้เช่น NSAPI, ISAPI และ ICAP
 - สามารถใช้งานในระบบที่แตกต่างกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การเชื่อมต่อกับฐานข้อมูล (Database) ด้วยภาษา JAVA

ในภาษาโปรแกรมสมัยใหม่จะมีช่องทางสำหรับเชื่อมต่อกับ Database อยู่ในภาษา JAVA ก็เช่นกัน โดยการใช้ JDBC (JAVA Database Connectivity)



รูปที่ 4-12 แสดงโมดูลการเชื่อมต่อกับ JDBC อย่างง่าย

การทำงานของ JDBC จะแบ่งเป็นขั้นตอนหลักๆ ได้ดังนี้คือ

4.6.1 การเชื่อมต่อกับฐานข้อมูล

การเชื่อมต่อกับฐานข้อมูลด้วย JDBC จะมีรูปแบบหลักในการทำงานดังนี้

`jdbc:<subprotocol>:<subname>`

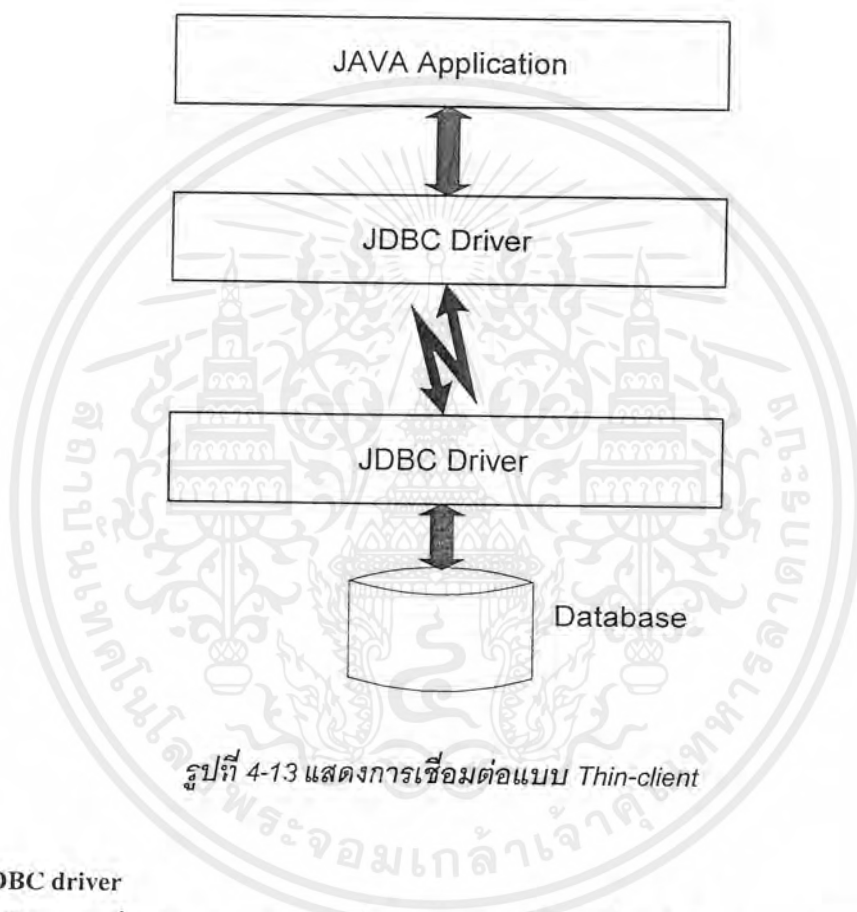
จะมี Protocol หลักในการทำงานเป็น JDBC ซึ่งมีการทำงานแบบเดียวในการเชื่อมต่อกับตัว JAVA แต่จะแตกต่างกันที่ Sub Protocol ซึ่งเป็นตัวบอกชนิดของการเชื่อมต่อกับตัวแม่ข่ายของฐานข้อมูล และ Sub Name จะเป็นตั้งบอกข้อมูลของการเข้าถึงฐานข้อมูลนั้นๆ เช่น เลขพอร์ท หรือ ชื่อของเครื่องแม่ข่าย สำหรับ Sub Protocol มีได้หลายแบบตามชนิดของการเชื่อมต่อคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.1.1 Thin driver

เป็นรูปแบบดั้งเดิมของการเชื่อมต่อด้วย JDBC เป็นการเชื่อมต่อที่ทั้งด้านเครื่องลูกข่ายและเครื่องแม่ข่ายทำการติดต่อกันผ่านทางโปรแกรมที่พัฒนาโดย JAVA ทั้งหมด (Pure JAVA) โดยฐานข้อมูลส่วนใหญ่ที่สนับสนุน JAVA ก็จะสามารถสนับสนุนการเชื่อมต่อแบบนี้ แต่ปัจจุบันไม่เป็นที่นิยมใช้งานแล้ว ตัวอย่างการขอเชื่อมต่อโดยใช้ Thin driver

```
DriverManager.getConnection ("jdbc:oracle:thin:@myhost:1521:orcl", "scott", "tiger");
```



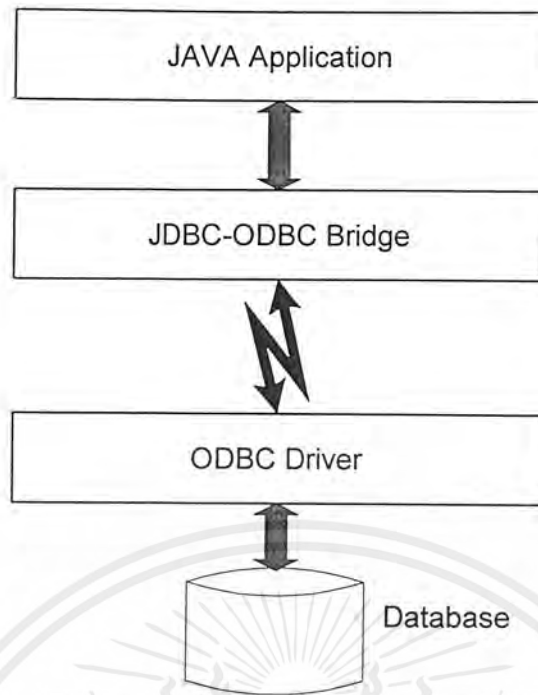
รูปที่ 4-13 แสดงการเชื่อมต่อแบบ Thin-client

4.6.1.2 ODBC driver

ใช้ในระบบที่ทำงานบน Windows ซึ่งตัว JDBC จะทำการเชื่อมต่อกับ ODBC (Open Database Connectivity) สามารถทำงานได้กับ Database ส่วนใหญ่เพราะโดยทั่วไปจะสนับสนุน ODBC อยู่แล้ว สำหรับ JDK จะมีโปรแกรม JDBC-ODBC มาให้เป็นมาตรฐานมีข้อดีคือ สามารถสร้างระบบได้อย่างรวดเร็ว มีราคาถูกรวมทั้งสามารถเชื่อมต่อกับฐานข้อมูลได้หลากหลายชนิดเพราะโดยทั่วไประบบฐานข้อมูลที่ใช้กับระบบวินโดวส์จะสามารถใช้งานกับ ODBC

```
DriverManager.getConnection ("jdbc:odbc:Fred", "Fernanda", "J8");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

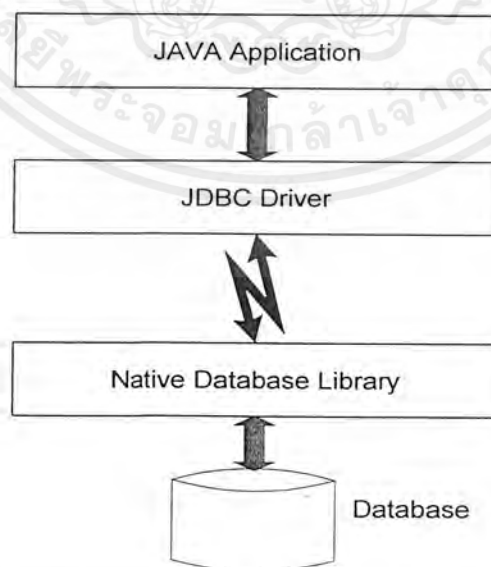


รูปที่ 4-14 แสดงการเชื่อมต่อผ่าน JDBC-ODBC Bridge

4.6.1.3 DBMS driver

เช่น OCI (Oracle Call Interface) ของ Oracle จะเป็นการปรับให้ใช้ JDBC ทำการเชื่อมต่อโดยตรงกับ DBMS ซึ่งจะทำงานได้มีประสิทธิภาพมากที่สุด แต่การใช้งานจะขึ้นอยู่กับ DBMS ที่ใช้ เช่น

```
Drive:Manager.getConnection ("jdbc:oracle:oci8:@MyHostString", "scott", "tiger");
```



รูปที่ 4-15 แสดงการเชื่อมต่อผ่าน DBMS Driver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.1 การจัดการภาษา SQL ใน JDBC

ภาษา SQL เป็นภาษาที่นิยมใช้ในงานฐานข้อมูลซึ่ง JDBC ก็ได้นำมาใช้ในการจัดการข้อมูลต่างๆ แต่การใช้งานจะแตกต่างกับการเขียนคำสั่ง SQL โดยตรงในโปรแกรมฐานข้อมูล

การส่งคำสั่ง SQL ให้ JDBC จะกระทำผ่านตัวแปรประเภท Statement ซึ่งจะกำหนดคำสั่ง SQL ลงไปแล้วใช้ เมธอด Execute ในการทำงาน โดยแบ่งออกเป็น 2 ประเภทคือการแก้ไขข้อมูล (Update) เช่น Create table หรือ Insert จะใช้คำสั่ง executeUpdate และการค้นหาข้อมูล (Query) เช่น Select ใช้คำสั่ง executeQuery ในการใช้งานตัวแปร statement จะต้องมีการเชื่อมต่อกับฐานข้อมูลก่อนแล้วกำหนดให้ตัวแปร statement ทำงานกับฐานข้อมูลใดตัวอย่างเช่น

```
Statement stmt = con.createStatement();
```

การเรียกใช้คำสั่ง SQL ผ่านตัวแปร statement ดังตัวอย่าง

```
stmt.executeUpdate("CREATE TABLE COFFEES " +
"(COF_NAME VARCHAR(32), SUP_ID INTEGER, PRICE FLOAT, " +
"SALES INTEGER, TOTAL INTEGER)");
```

หรืออาจกำหนดคำสั่ง SQL ลงในดั่งแปรแล้วจึงทำคำสั่ง ดังตัวอย่าง

```
stmt.executeUpdate(createTableCoffees);
```

และตัวอย่างการใช้คำสั่ง Insert

```
stmt.executeUpdate("INSERT INTO COFFEES VALUES ('Colombian', 101, 7.99, 0, 0);");
```

ในการใช้คำสั่ง executeQuery จะต้องสร้างตัวแปร ResultSet ขึ้นมาเก็บผลลัพธ์ที่ได้จากการประมวลผล

```
ResultSet rs = stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");
```

การประมวลผล ResultSet ที่ได้จะใช้เมธอด getString, getInt, getByte หรืออื่นๆ ในประเภทเดียวกัน โดยแบ่งตามชนิดของข้อมูลที่ถูกเก็บไว้ในฐานข้อมูล โดยจะดึงข้อมูลจากแถวปัจจุบัน (current row) และใช้เมธอด next ในการเข้าถึงแถวถัดไป ดังตัวอย่าง

```
String query = "SELECT COF_NAME, PRICE FROM COFFEES";
ResultSet rs = stmt.executeQuery(query);
while (rs.next()) {
String s = rs.getString("COF_NAME");
float n = rs.getFloat("PRICE");
System.out.println(s + " " + n);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างที่แสดงผ่านมาจะเป็นการทดสอบการทำงานกับคำสั่ง SQL ที่คงที่ แต่ในความเป็นจริงข้อความที่ต้องการจะเปลี่ยนแปลงได้ตามโปรแกรมผ่านตัวแปร ในกรณีนี้ให้ตัวแปร PreparedStatement แทนการใช้ statement ซึ่งไม่สามารถรับการเปลี่ยนแปลงหลังการประกาศได้ ดังตัวอย่าง

```
PreparedStatement updateSales = con.prepareStatement(
    "UPDATE COFFEES SET SALES = ? WHERE COF_NAME LIKE ? ");
updateSales.setInt(1, 75);
updateSales.setString(2, "Colombian");
updateSales.executeUpdate();
```

การกำหนดค่าจะใช้ลำดับของ ? แทนตำแหน่งของตัวแปรที่จะถูกนำไปแทน

4.6.2 การยกเลิกการติดต่อ

ขณะที่ทำการติดต่ออยู่จะสามารถทำคำสั่งที่ต้องการได้หลายครั้งโดยไม่จำเป็นต้องขอการเชื่อมต่อทุกครั้งที่จะทำคำสั่ง แต่หากต้องการยกเลิกการติดต่อสามารถทำได้โดยใช้เมธอด close

4.6.3 ตัวอย่างโปรแกรมที่ใช้ JDBC

```
import java.net.URL;
import java.sql.*;

class SimpleSelect {

public static void main (String args[]) {
    String url = "jdbc:odbc:my-dsn";
    String query = "SELECT * FROM emp";

    try {

        Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
        DriverManager.setLogStream(System.out);
        Connection con = DriverManager.getConnection (
            url, "my-user", "my-passwd");
        checkForWarning (con.getWarnings ());
        DatabaseMetaData dma = con.getMetaData ();

        System.out.println("\nConnected to " + dma.getURL());
        System.out.println("Driver      " +
            dma.getDriverName());
        System.out.println("Version    " +
            dma.getDriverVersion());
        System.out.println("");

        Statement stmt = con.createStatement ();
        ResultSet rs = stmt.executeQuery (query);
        dispResultSet (rs);
        rs.close();
        stmt.close();
        con.close();

    }
    catch (SQLException ex) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

System.out.println ("\n*** SQLException caught ***\n");
while (ex != null) {
    System.out.println ("SQLState: " +
        ex.getSQLState ());
    System.out.println ("Message: " + ex.getMessage ());
    System.out.println ("Vendor: " +
        ex.getErrorCode ());
    ex = ex.getNextException ();
    System.out.println ("");
}
}
catch (java.lang.Exception ex) {
    ex.printStackTrace ();
}
}

private static boolean checkForWarning (SQLWarning warn)
    throws SQLException {
    boolean rc = false;

    if (warn != null) {
        System.out.println ("\n *** Warning ***\n");
        rc = true;
        while (warn != null) {
            System.out.println ("SQLState: " +
                warn.getSQLState ());
            System.out.println ("Message: " +
                warn.getMessage ());
            System.out.println ("Vendor: " +
                warn.getErrorCode ());
            System.out.println ("");
            warn = warn.getNextWarning ();
        }
    }
    return rc;
}

private static void dispResultSet (ResultSet rs)
    throws SQLException
{
    int i;
    ResultSetMetaData rsmd = rs.getMetaData ();
    int numCols = rsmd.getColumnCount ();
    for (i=1; i<=numCols; i++) {
        if (i > 1) System.out.print(",");
        System.out.print(rsmd.getColumnLabel(i));
    }
    System.out.println("");
    boolean more = rs.next ();
    while (more) {
        for (i=1; i<=numCols; i++) {
            if (i > 1) System.out.print(",");
            System.out.print(rs.getString(i));
        }
        System.out.println("");
        more = rs.next ();
    }
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การออกแบบและการสร้างโปรแกรมในโครงการส่วนแรก

5.1 UML

UML หรือ Unified Modeling Language เป็นภาษาในการโมเดลมาตรฐาน UML เกิดจากการพัฒนา ร่วมกันของผู้นำเทคโนโลยีทางด้านอ็อบเจกต์ 3 คน คือ Grady Booch Ivar Jacobson และ Jim Rumbaugh โดยก่อนที่จะมาเป็น UML นั้น ผู้นำทั้ง 3 คนนี้ต่างก็มีโมเดลทางด้านอ็อบเจกต์เป็นของตัวเอง ต่อมาทางบริษัท Rational Software จึงได้ร่วมมือกับผู้นำทั้ง 3 คนนี้ ในการพัฒนาโมเดลร่วมกัน เพื่อให้เกิดความเป็นหนึ่งเดียว สำหรับการพัฒนาระบบด้วยอ็อบเจกต์โอเรียนเท็ด

UML นั้นได้รวมแนวคิดของวิธีการต่าง ๆ เข้าไว้ด้วยกัน จุดประสงค์ของ UML ก็คือ ต้องการสร้าง โมเดลในการพัฒนาที่เข้าใจและสร้างได้ง่าย แต่สามารถนำไปใช้ได้กับทุกระบบ ตัว UML เองนั้นได้รับการ สนับสนุนจากผู้นำในอุตสาหกรรมทางด้านคอมพิวเตอร์หลาย ๆ ราย เช่น HP, ICON Computing, IBM, I-Logix, Intellicorp, MCI Systemhouse, Microsoft, ObjecTime, Oracle, Platinum Technology, Ptech, Reich Technologies, Softeam, Sterling Software, Taskon และ Unisys

ใน UML มีโมเดลที่มองเห็นได้สำหรับระบบหลาย ๆ โมเดล โดยแต่ละโมเดลก็จะแสดงมุมมองต่อ ระบบที่ไม่เหมือนกัน ซึ่งประกอบไปด้วย

- ยูสเคสไดอะแกรม (Use-Case Diagram)
ใช้แสดงการติดต่อระหว่างระบบกับผู้ใช้
- คลาสไดอะแกรม
ใช้แสดงโครงสร้างทางลอจิกของระบบ
- อ็อบเจกต์ไดอะแกรม
ใช้แสดงอ็อบเจกต์และความสัมพันธ์
- สเตทไดอะแกรม (State Diagram)
ใช้แสดงพฤติกรรมของระบบ
- คอมโพเนนต์ไดอะแกรม
ใช้แสดงโครงสร้างทางกายภาพของซอฟต์แวร์
- ดีพลอยเม้นต์ไดอะแกรม (Deployment Diagram)
ใช้แสดงการติดต่อระหว่างซอฟต์แวร์กับฮาร์ดแวร์
- ซีควเอนซ์ไดอะแกรม (Sequence Diagram)
ใช้แสดงลำดับของเมสเสจระหว่างอ็อบเจกต์
- คอลลาบอเรชันไดอะแกรม (Collaboration Diagram)
ใช้แสดงข้อมูลพื้นฐานเช่นเดียวกับซีควเอนซ์ไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับความแตกต่างระหว่าง ซีควেনซ์ไดอะแกรม กับ คอลลาบอเลชันไดอะแกรม นั้น คือ ซีควেনซ์ไดอะแกรม จะแสดงลำดับของเมสเสจเป็นหลัก ส่วน คอลลาบอเลชันไดอะแกรม นั้นจะแสดง โครงสร้างของอ็อบเจกต์ที่ทำงานร่วมกันเป็นหลัก

5.1.1 คลาสไดอะแกรม (Class Diagram)

คลาสไดอะแกรมเป็นไดอะแกรมที่มีความสำคัญมากที่สุด ในการวิเคราะห์และออกแบบโดยใช้วิธีอ็อบเจกต์โอเรียนเท็ด เนื่องจากคลาสไดอะแกรมจะแสดงโครงสร้างของอ็อบเจกต์และคลาสที่มีในระบบ รวมทั้งแสดงความสัมพันธ์ด้วย คลาสไดอะแกรมจะเป็นโครงสร้างหลักของระบบและยังใช้ในการแยกย่อยรายละเอียด (Decomposite) ด้วยวิธีอ็อบเจกต์โอเรียนเท็ด อย่างไรก็ตามก็ตีคลาสดิอะแกรมนั้น อาจจะมีหน้าที่คล้ายกับอ็อบเจกต์ไดอะแกรม แต่ว่าคลาสดิอะแกรมจะแสดงเฉพาะคลาส โดยไม่ได้แสดงอินสแตนซ์ หรืออ็อบเจกต์ทั้งหมด

5.1.2 ความสัมพันธ์ระหว่างคลาส

สามารถแบ่งหลักๆ ได้ดังนี้

- แอสโซซิเอชัน (Association)

เมื่ออ็อบเจกต์หนึ่งใช้บริการของอีกอ็อบเจกต์ แต่ไม่ได้เป็นเจ้าของอ็อบเจกต์นั้น เราเรียกความสัมพันธ์นี้ว่า แอสโซซิเอชัน ซึ่งความสัมพันธ์แบบนี้ จะใช้ได้ อย่างเหมาะสมก็ต่อเมื่อ

- อ็อบเจกต์ใช้บริการของอีกอ็อบเจกต์หนึ่ง แต่ไม่ได้มีความสัมพันธ์แบบอะกริเกชัน
- วงจรชีวิตของคลาสที่ใช้ ไม่ได้เป็นหน้าที่ความรับผิดชอบของคลาสที่เรียกใช้ ทั้งการสร้างและทำลายอ็อบเจกต์
- ความสัมพันธ์ระหว่างอ็อบเจกต์นั้น น้อยกว่าอะกริเกชัน
- ความสัมพันธ์เป็นคุณลักษณะของ คลาสเอ็นดี-เซิร์ฟเวอร์
- อ็อบเจกต์ที่ถูกเรียก มีการแชร์และถูกใช้มากเท่าๆ กับที่อ็อบเจกต์อื่น ๆ เรียกใช้



รูปที่ 5-1 สัญลักษณ์ของแอสโซซิเอชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● การถ่ายทอด (Inheritance)

เมื่อคลาสหนึ่งเป็นความเฉพาะของอีกคลาสหนึ่งนั้น เราจะใช้ความสัมพันธ์ที่เรียกว่า เจเนอรัลไลเซชัน (Generalization) หรือ การถ่ายทอด (Inheritance) หมายความว่า คลาสลูกจะมีคุณสมบัติทุกอย่างที่คลาสพ่อแม่มี ถึงแม้ว่าคลาสลูกจะมีความเฉพาะมากกว่าก็ตาม คลาสลูกอาจจะเพิ่มคุณสมบัติของคลาสพ่อแม่ โดยการเพิ่มแอทริบิวต์และเมธอดเข้าไป เราสามารถเรียกความสัมพันธ์นี้ว่า "เป็นชนิดหนึ่งของ" เช่น สัตว์เลี้ยงลูกด้วยนมเป็นชนิดหนึ่งของสัตว์ อินฟราเรดเซนเซอร์เป็นชนิดหนึ่งของเซนเซอร์ เป็นต้น



รูปที่ 5-2 สัญลักษณ์ของการถ่ายทอด

● อะกรีเกชัน (Aggregation)

ความสัมพันธ์แบบอะกรีเกชันนี้ จะใช้ก็ต่อเมื่อมี อ็อบเจกต์หนึ่งบรรจุอีก อ็อบเจกต์หนึ่งไว้ ทั้งในทางความคิดและทางกายภาพ คลาสที่ใหญ่กว่า เราเรียกว่า เจ้าของหรือ Owner หรือ whole ซึ่งเป็นคลาสที่มีหัวลูกศรรูปสี่เหลี่ยมอยู่ ส่วนคลาสที่เล็กกว่านั้น เราเรียกว่า คลาสคอมโพเนนต์ (Component) หรือ owned หรือ part โดยทั่วไป คลาสเจ้าของมีหน้าที่สร้างและทำลายคลาสคอมโพเนนต์ ใน UML ขอมให้ คลาสคอมโพเนนต์สามารถแชร์กับคลาสเจ้าของหลายคลาสได้ และเมื่ออ็อบเจกต์มีการแชร์ความเป็นเจ้าของ เราก็จะต้องใช้กฎของ ac hoc ในการตัดสินใจว่าคลาสใดจะทำหน้าที่สร้างและทำลายคลาสคอมโพเนนต์นั้น

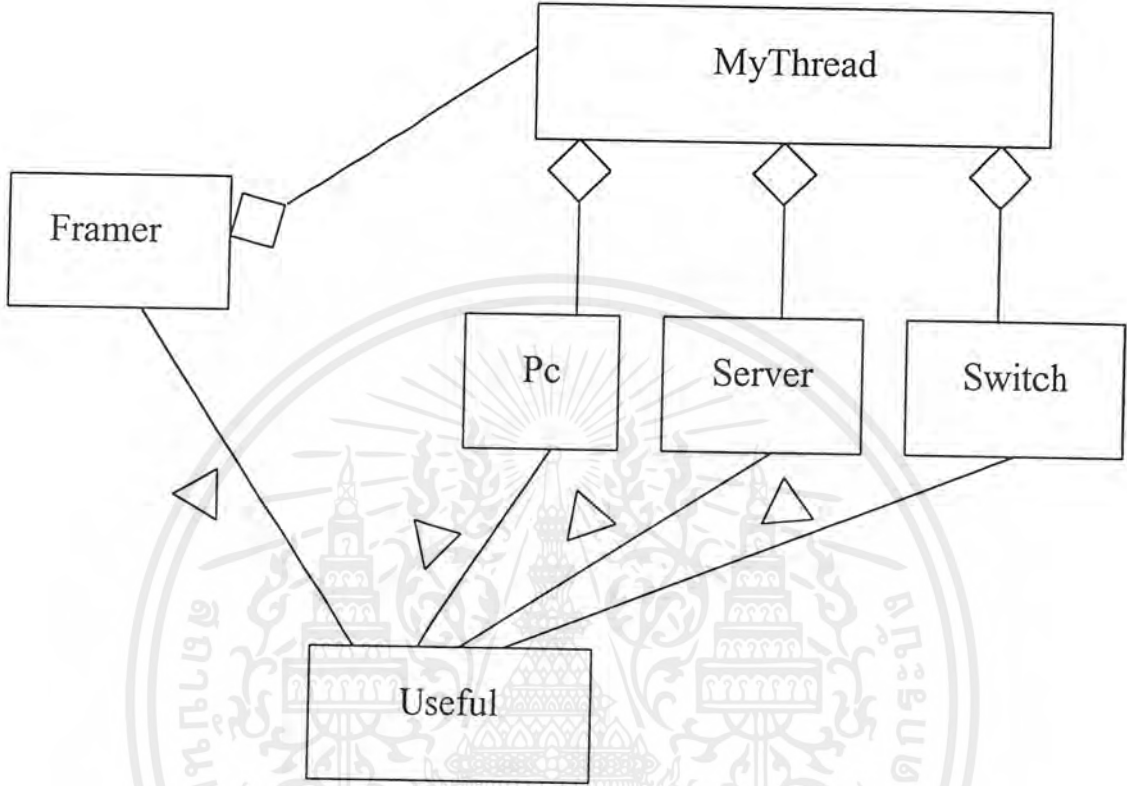


รูปที่ 5-3 สัญลักษณ์ของอะกรีเกชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 คลาสไดอะแกรม ภายในโครงการ

- ส่วนที่เป็น จาวาแอปพลิเคชัน (Java Application) ทำหน้าที่เก็บข้อมูลลงฐานข้อมูลตลอดเวลา



รูปที่ 5-4 คลาสไดอะแกรมของส่วนที่เป็น จาวาแอปพลิเคชัน

- ส่วนที่เป็น จาวาเซิร์ฟเล็ต (Java Servlet) ทำหน้าที่ประมวลผลข้อมูล แล้วส่งกลับไปยังเว็บเพจ
 ในส่วนนี้ หากต้องการจะประมวลผลข้อมูลเกี่ยวกับเรื่องใด ก็จะแยกเป็น
 คลาสต่างหาก เช่น คลาส inout คลาส pview เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การออกแบบฐานข้อมูล (Database Design)

ฐานข้อมูลภายในโครงการ ประกอบด้วย ตารางดังต่อไปนี้

- **Device**

ตาราง Device เป็นตารางที่มีหน้าที่ในการเก็บค่าข้อมูล ในส่วน System group ของอุปกรณ์

Device (IP , Description , ObjectID , Uptime , Contact , Name , Location , Services)

IP	Description	ObjectID	Uptime	Contact	Name	Location	Services

- **Interface**

ตาราง Interface เป็นตารางที่มีหน้าที่ในการเก็บค่าข้อมูลบางส่วนของ Interface group ในอุปกรณ์

Interface (Inter , Description , Type , Speed)

Inter	Description	Type	Speed

- **Datamid**

ตาราง Datamid เป็นตารางที่มีหน้าที่ในการเก็บค่าข้อมูลกลุ่มมิมบิที่สนใจ ณ เวลาเที่ยงคืนของทุกคืน

Datamid (Inter, Timemill, Dates, InOctet, OutOctet, SingleCol, MultiCol, DeferTran, Octet, CRC, Colli)

Inter	Timemill	Dates	InOctet	OutOctet	...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Keepalltime**

ตาราง Keepalltime เป็นตารางที่มีหน้าที่ในการเก็บค่าข้อมูลกลุ่มมิมที่สนใจ โดยค่าในตารางนี้จะถูก update อยู่ตลอดเวลา

Keepalltime (Inter , InOctet , OutOctet , SingleCol , MultiCol , DeferTran , Octet , CRC , Colli)

Inter	InOctet	OutOctet	SingleCol	MultiCol	...

- **Inoutdevice**

ตาราง Inoutdevice จะเป็นตารางที่เก็บค่าการเข้าออกของอุปกรณ์ พร้อมทั้งข้อมูลของกลุ่มมิมที่สนใจ

Inoutdevice (Inter,Date1,Timemill1,Backing,InOctet1,OutOctet1,SingleCol1,MultiCol1,DeferTran1,Octet1,CRC1,Colli1,Date2,Timemill2,InOctet2,OutOctet2,SingleCol2,MultiCol2,DeferTran2, Octet2,CRC2,Colli2)

Inter	Date1	Timemill1	Backing	InOctet1	OutOctet1	...

- **Forswitch**

ตาราง Forswitch จะเป็นตารางที่เก็บค่าข้อมูลกลุ่มมิมที่สนใจแบบทุกๆ ชั่วโมง ของสวิตช์

Forswitch (Inter , Timemill , Dates , Octet , CRC , Colli , Hoctet , HCRC , Hcolli)

Inter	Timemill	Dates	Octet	CRC	Colli	...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การสร้างโปรแกรมและการทำงาน

5.3.1 ส่วนที่เป็น จาวาแอปพลิเคชัน (Java Application) ทำหน้าที่เก็บข้อมูลลงฐานข้อมูลตลอดเวลา

5.3.1.1 คลาส Framer

สำหรับ คลาส Framer มีส่วนประกอบที่สำคัญ คือ

- **คอนสตรักเตอร์ Framer**

คอนสตรักเตอร์ Framer นั้น จะทำหน้าที่ส่วนใหญ่ เกี่ยวกับการเชื่อมต่อเซิร์ฟเวอร์อินเทอร์เน็ต (User Interface) ของเฟรม (Frame) ซึ่งติดต่อกับผู้ใช้ ซึ่งเฟรมนี้ จะปรากฏขึ้นเมื่อเริ่ม โปรแกรม

- **เมธอด handleEvent**

เมธอด handleEvent จะจัดการ ในเรื่องเกี่ยวกับการออกจากโปรแกรม

- **เมธอด Rightbutton_mouseClicked**

เมธอด Rightbutton_mouseClicked จะสร้างเธรด (Thread) ในการทำงานของโปรแกรม และสั่งให้เธรดที่สร้างขึ้นมา รัน (run)

- **เมธอด Leftbutton_mouseClicked**

เมธอด Leftbutton_mouseClicked จะหยุดการทำงานของเธรด และทำการลบข้อมูลของทุกตาราง ในฐานข้อมูล

5.3.1.2 คลาส MyThread

สำหรับ คลาส MyThread มีส่วนประกอบที่สำคัญ คือ

- **เมธอด run**

สำหรับ เมธอด run นั้น ในตอนเริ่มต้นจะทำการสร้างอินสแตนซ์ ของคลาสอุปกรณ์ (คลาส Pc คลาส Server คลาส Switch) ตามต้องการ จากนั้นก็จะเข้าสู่ลูป (loop) while(true) เพื่อวนตรวจสอบและเก็บค่าของแต่ละอุปกรณ์

```
while (true) {
    pc6157.conCheck();
    pc6127.conCheck();
    server43.conCheck();
    ....
    try{
        sleep(30000); //หยุดการทำงานไป 30 s
    } catch (InterruptedException e){ }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.1.3 คลาส Useful

สำหรับ คลาส Useful จะเป็นคลาสที่รวมเมธอดที่ใช้งานทั่วไป ทั้งการ select insert delete update ตารางในฐานข้อมูล ตลอดจนเมธอดที่ใช้ในการเรียกค้น (query) ข้อมูลจากอุปกรณ์

5.3.1.4 คลาส Pc คลาส Server คลาส Switch

ทั้ง 3 คลาส มีส่วนประกอบที่สำคัญ คือ

- คอนสตรัคเตอร์ ของแต่ละคลาส

สำหรับคอนสตรัคเตอร์ ของแต่ละคลาส จะทำหน้าที่ในการเซตค่าเริ่มต้นให้กับ อินสแตนซ์ เพื่อนำไปใช้งานต่อภายในเมธอด

- เมธอด ConCheck

เมธอด ConCheck จะประกอบไปด้วยอัลกอริทึมที่สำคัญ ในการจัดการค่าข้อมูล ต่างๆ จากอุปกรณ์ ซึ่งในที่สุด จะได้ค่าข้อมูลที่ถูกต้องในฐานข้อมูล พร้อมนำไป ใช้งานต่อในการแสดงผลบนเว็บเพจ

5.3.2 ส่วนที่เป็น จาวาเซิร์ฟฟเล็ท (Java Servlet) ทำหน้าที่ประมวลผลข้อมูล แล้วส่งกลับไปยังเว็บเพจ

5.3.2.1 คลาส viewall

สำหรับ คลาส viewall มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอดนี้ จะทำการ select ข้อมูลจากฐานข้อมูลขึ้นมา เพื่อทำการ ตรวจสอบว่าอุปกรณ์แต่ละตัว ในขณะนี้ มีสถานะเป็นอย่างไรบ้าง จากนั้นจะทำการแสดงผลสถานะของอุปกรณ์เหล่านั้น กลับไปยังเว็บเพจ

5.3.2.2 คลาส pview (ใช้สำหรับกลุ่มของ พีซี)

สำหรับ คลาส pview มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายในเมธอดนี้ จะทำการ select ข้อมูลจาก ตาราง Device และ ตาราง Interface ในฐานข้อมูล เพื่อนำขึ้นมาแสดงผลบนหน้าเว็บเพจ (ตารางทั้งสองนี้ จะบรรจุ ข้อมูลที่เกี่ยวข้องกับ System group และ Interface group ของอุปกรณ์ ตามลำดับ)

นอกจากนั้นแล้ว ยังทำการเรียกค้นข้อมูลปัจจุบัน ณ ขณะนั้น ในเรื่องเกี่ยวกับ จำนวนเวลารวมที่อุปกรณ์ทำงานขึ้นมาในครั้งล่าสุด และในเรื่องเกี่ยวกับผลรวม ค่าอ็อกเทต(octet) เข้าออกอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.2.3 คลาส inout

สำหรับ คลาส inout มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากตาราง Inoutdevice ในฐานข้อมูลขึ้นมา เพื่อทำการแสดงผลข้อมูลการเปิดและปิดอุปกรณ์ ที่เกิดขึ้น พร้อมทั้งระยะเวลาที่อุปกรณ์ทำงานในครั้งนั้นๆ ในรูปแบบของตารางบนหน้าเว็บเพจ

5.3.2.4 คลาส datainout

สำหรับ คลาส datainout มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากตาราง Inoutdevice และตาราง Keepalltime ในฐานข้อมูลขึ้นมา เพื่อทำการแสดงผลจำนวนข้อมูลที่เข้าและจำนวนข้อมูลที่ออก ในแต่ละครั้งของการเปิดและปิดอุปกรณ์ ซึ่งจะอยู่ในรูปแบบของตารางบนหน้าเว็บเพจ

5.3.2.5 คลาส snmpServlet4

สำหรับ คลาส snmpServlet4 มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายในเมธอด นี้ จะทำการเปิดช็อกเก็ตไปขอข้อมูล จากมิบ ในตัวอุปกรณ์ โดยตรง จากนั้นจะทำการแสดงผลข้อมูลที่ได้อ้างกล่าว ไปยังหน้าเว็บเพจ

5.3.2.6 คลาส inoutpday

สำหรับ คลาส inoutpday มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากตาราง Inoutdevice ในฐานข้อมูลขึ้นมา เพื่อประมวลผล ซึ่งจะได้สถิติจำนวนเวลาการใช้เครื่องต่อวัน แสดงอยู่ในรูปของกราฟแท่ง

5.3.2.7 คลาส data1pday คลาส datao1pday

ทั้ง 2 คลาส มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากตาราง Inoutdevice ตาราง Keepalltime และตาราง Datamid ในฐานข้อมูลขึ้นมา เพื่อทำการประมวลผล ซึ่งจะได้สถิติออกแทนที่ได้รับและส่งออก ในอินเทอร์เน็ตเฟสที่ 1 ต่อวัน ตามลำดับ แสดงอยู่ในรูปของกราฟแท่ง (data1pday แสดงออกแทนที่ได้รับ และ datao1pday แสดงออกแทนที่ส่งออก)

5.3.2.8 คลาส data2pday คลาส datao2pday

ทั้ง 2 คลาส มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากตาราง Inoutdevice ตาราง Keepalltime และตาราง Datamid ในฐานข้อมูลขึ้นมา เพื่อทำการประมวลผล ซึ่งจะได้สถิติการเปรียบเทียบออกแทนที่ได้รับและส่งออก ต่อวัน ในอินเทอร์เน็ตเฟสที่ 2 ตามลำดับ แสดงอยู่ในรูปของกราฟแท่ง (data2pday แสดงออกแทนที่ได้รับ และ datao2pday แสดงออกแทนที่ส่งออก)

5.3.2.9 คลาส com1pday คลาส com2pday

ทั้ง 2 คลาส มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากตาราง Inoutdevice ตาราง Keepalltime และตาราง Datamid ในฐานข้อมูลขึ้นมา เพื่อทำการประมวลผล ซึ่งจะได้สถิติการเปรียบเทียบออกแทนที่ได้รับและส่งออก ต่อวัน ในอินเทอร์เน็ตเฟสที่ 1 และ ในอินเทอร์เน็ตเฟสที่ 2 ตามลำดับ แสดงอยู่ในรูปของกราฟแท่ง (com1pday แสดงในอินเทอร์เน็ตเฟสที่ 1 และ com2pday แสดงในอินเทอร์เน็ตเฟสที่ 2)

5.3.2.10 คลาส serview (ใช้สำหรับกลุ่มของ เซิร์ฟเวอร์)

สำหรับ คลาส serview มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายในเมธอดนี้ จะทำการ select ข้อมูลจาก ตาราง Device และ ตาราง Interface ในฐานข้อมูล เพื่อนำขึ้นมาแสดงผลบนหน้าเว็บเพจ (ตารางทั้งสองนี้ จะบรรจุ ข้อมูลที่เกี่ยวข้องกับ System group และ Interface group ของอุปกรณ์ ตามลำดับ)

นอกจากนั้นแล้ว ยังทำการเรียกค้นข้อมูลปัจจุบัน ณ ขณะนั้น ในเรื่องเกี่ยวกับ จำนวนเวลารวมที่อุปกรณ์ทำงานขึ้นมาในครั้งล่าสุด ในเรื่องเกี่ยวกับผลรวมค่า อ็อกเทต(octet) เข้าออก และในเรื่องเกี่ยวกับผลรวมค่าใน Ethernet Interface MIB อีกด้วย

5.3.2.11 คลาส singlecolpdays คลาส multicolpdays คลาส defertranpdays

ทั้ง 3 คลาส มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากตาราง Inoutdevice ตาราง Keepalltime และตาราง Datamid ในฐานข้อมูลขึ้นมา เพื่อทำการประมวลผล ซึ่งจะได้สถิติค่าใน Ethernet Interface MIB ที่ได้เลือกมาแสดงผล ต่อวัน แสดง อยู่ในรูปของกราฟแท่ง

5.3.2.12 คลาส swview (ใช้สำหรับกลุ่มของ สวิตช์)

สำหรับ คลาส swview มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายในเมธอดนี้ จะทำการ select ข้อมูลจาก ตาราง Device และ ตาราง Interface ในฐานข้อมูล เพื่อนำขึ้นมาแสดงผลบนหน้าเว็บเพจ (ตารางทั้งสองนี้ จะบรรจุ ข้อมูลที่เกี่ยวข้องกับ System group และ Interface group ของอุปกรณ์ ตามลำดับ)

นอกจากนั้นแล้ว ยังทำการเรียกค้นข้อมูลปัจจุบัน ณ ขณะนั้น ในเรื่องเกี่ยวกับ จำนวนเวลารวมที่อุปกรณ์ทำงานขึ้นมาในครั้งล่าสุด และในเรื่องเกี่ยวกับผลรวม ค่าใน RMON MIB อีกด้วย

5.3.2.13 คลาส swperday

สำหรับ คลาส swperday มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากราย Forswitch ในฐานข้อมูล ขึ้นมา เพื่อทำการประมวลผล ซึ่งจะได้ค่าสถิติในแบบต่อวัน ของสวิทช์ แสดงอยู่ในรูปของกราฟแท่ง

5.3.2.14 คลาส swperhour

สำหรับ คลาส swperhour มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากราย Forswitch ในฐานข้อมูล ขึ้นมา เพื่อทำการประมวลผล ซึ่งจะได้ค่าสถิติในแบบต่อชั่วโมง ของสวิทช์ แสดงอยู่ในรูปของกราฟแท่ง

5.3.2.15 คลาส swperhour

สำหรับ คลาส swperhour มีส่วนประกอบที่สำคัญ คือ

- เมธอด doGet

ภายใน เมธอด นี้ จะทำการ select ข้อมูลจากราย Forswitch ในฐานข้อมูล ขึ้นมา เพื่อทำการประมวลผล ซึ่งจะได้ค่าสถิติในแบบต่อชั่วโมง เปรียบเทียบกันในแต่ละอินเทอร์เฟซ ของสวิทช์ แสดงอยู่ในรูปของกราฟแท่ง

สำหรับในส่วนที่เป็น จาวาเซิร์ฟเล็ต ภายในโครงงานนี้ จะอยู่ใน 2 ลักษณะคือ

1. เรียกใช้ผ่านเอกสาร HTML
2. อยู่ในรูปแบบของ Server Side Include (SSI)

คือ จะใช้แท็ก (Tag) Servlet ภายในเอกสารนามสกุล .shtml ดังตัวอย่างต่อไปนี้

```
....
พื้นที่ตรงนี้ สามารถใส่แท็ก html ได้ตามปกติ
<SERVLET CODE=pcview>
    <PARAM NAME=ip VALUE=161.246.6.117>
</SERVLET>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากกดปุ่ม Start จาваแอปพลิเคชันจะทำการตรวจสอบ ตัวอุปกรณ์ และ เก็บข้อมูลจากมิบของอุปกรณ์ ลงในฐานข้อมูลไปเรื่อยๆ จนกว่าจะมีการกดปุ่ม Stop

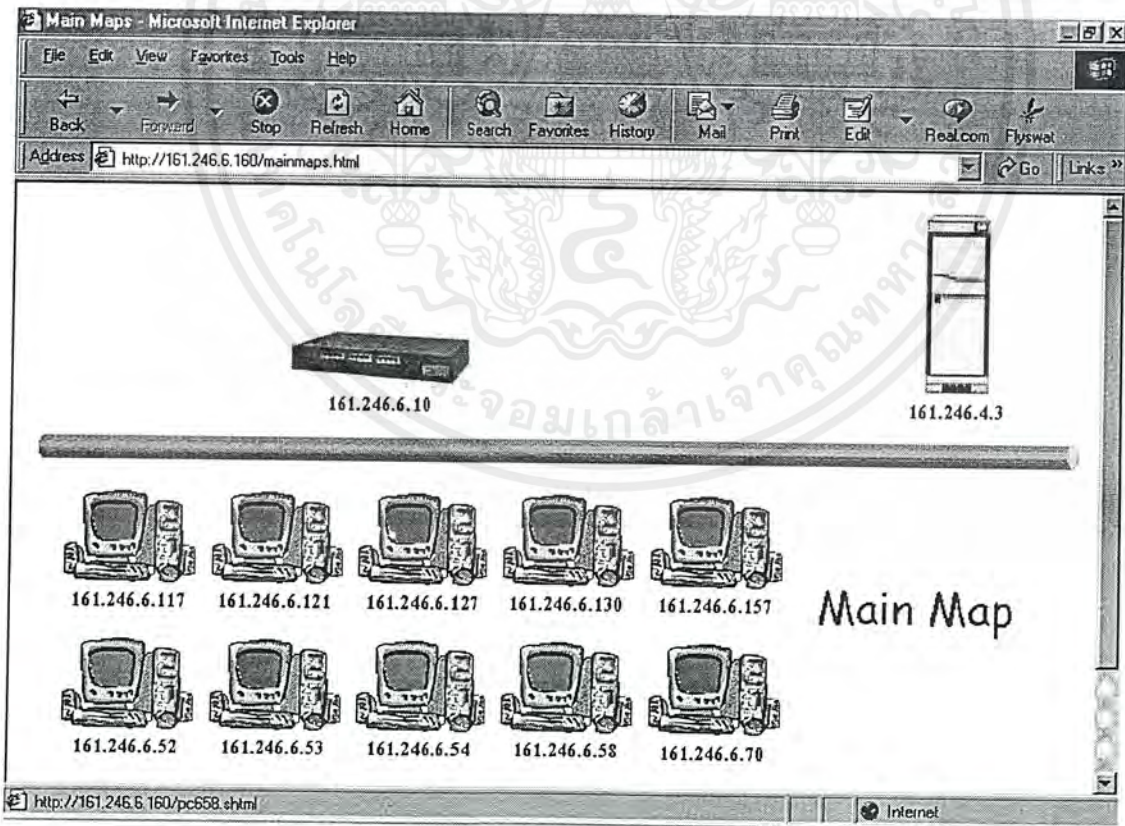
- กดปุ่ม Stop

เมื่อกดปุ่ม Stop จาवाแอปพลิเคชันจะหยุดการตรวจสอบตัวอุปกรณ์และหยุดเก็บข้อมูลจากมิบของอุปกรณ์ และทำการลบข้อมูลที่อยู่ในทุกตารางของฐานข้อมูล

5.4.2 ส่วนที่เป็น จาवाเซิร์ฟเล็ต ทำหน้าที่ประมวลผลข้อมูล แล้วส่งกลับไปยังเว็บเพจ

5.4.2.1 กลาส viewall

สำหรับกลาส viewall นั้น จะถูกเรียกเพื่อแสดงผล สถานะของอุปกรณ์โดยรวม ดังรูป



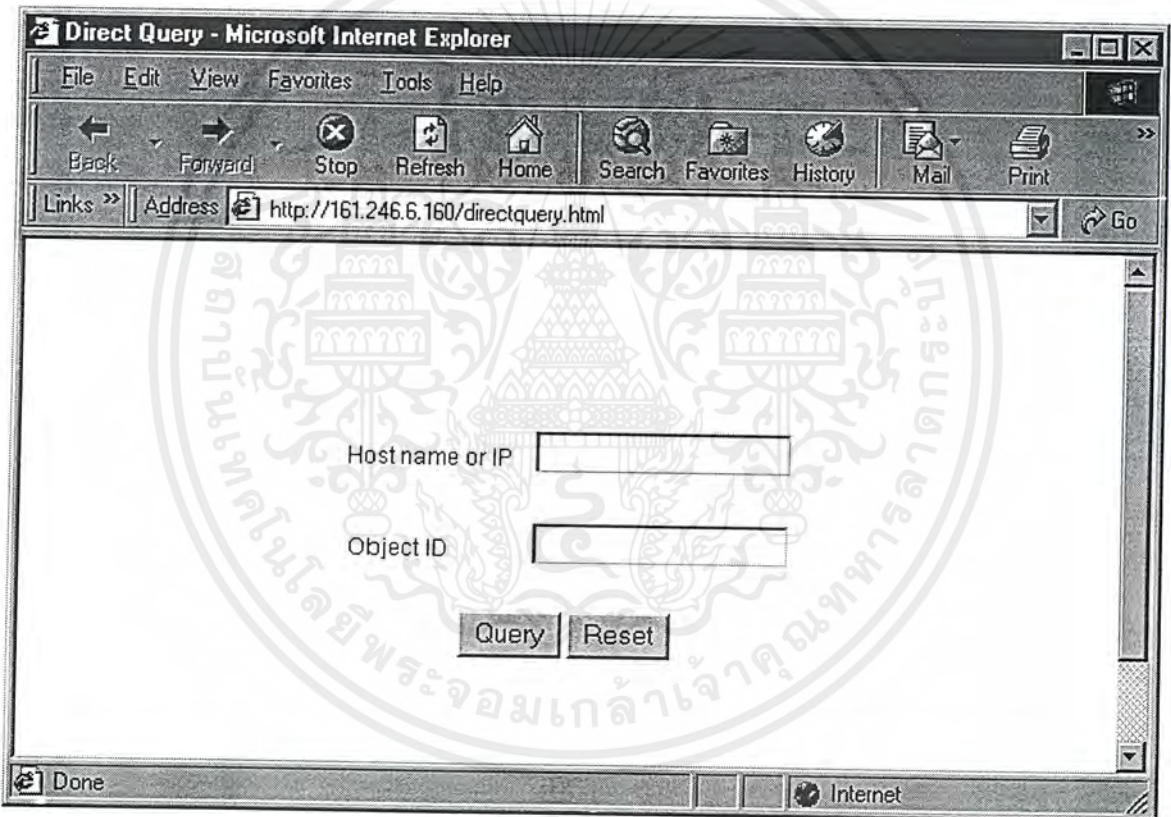
รูปที่ 5-7 สถานะของอุปกรณ์โดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เว็บเพจแสดง Main Map นี้ หากมีอุปกรณ์ใดหยุดทำงานลง รูปภาพที่อยู่บนเว็บเพจก็จะเปลี่ยนไป โดยอัตโนมัติ เช่น ในกลุ่มของพีซี สีของภาพก็จะเปลี่ยนจากสีเขียวเป็นสีแดง และในทางตรงกันข้าม หากเดิมอุปกรณ์ไม่ได้ทำงาน แล้วต่อมาอุปกรณ์ทำงานขึ้น ในกลุ่มของพีซี สีของภาพก็จะเปลี่ยนจากสีแดงเป็นสีเขียว ให้โดยอัตโนมัติเช่นกัน

5.4.2.2 กลาส snmpServlet4

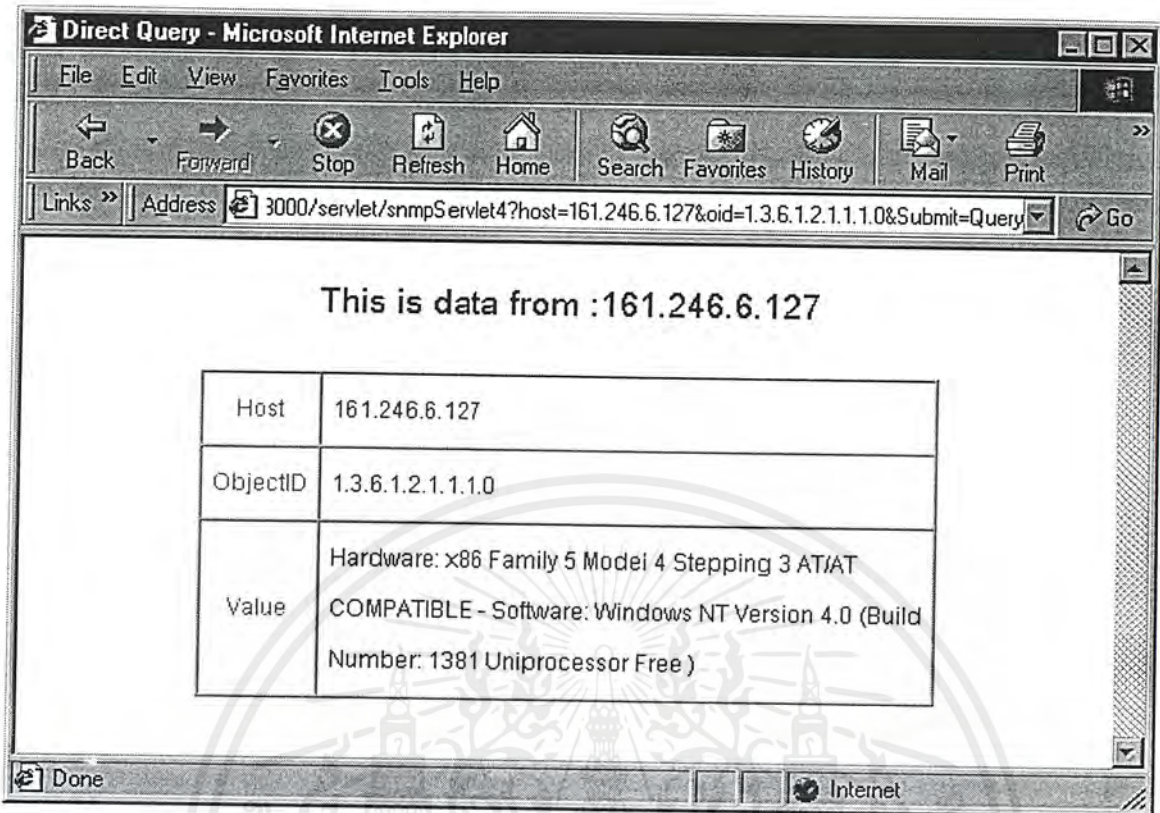
จาก Main Map ดังรูป 5-7 ให้คลิก Click here for Direct Query (อยู่ล่างสุดของหน้า) ซึ่งจะปรากฏหน้าเว็บเพจดังรูป



รูปที่ 5-8 หน้าจอ Direct Query

ใส่ IP และ Object ID ตามต้องการ แล้วกดปุ่ม Query กลาส snmpServlet4 จะถูกเรียกใช้ และแสดงผลสู่เว็บเพจ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-9 ผลลัพธ์ที่ได้จากการเรียกค้น (Query)

5.4.2.3 กลาส pview

จาก Main Map ดังรูป 5-7 หากต้องการทราบข้อมูลที่เกี่ยวข้องกับอุปกรณ์ใด ก็ให้คลิกที่ภาพอุปกรณ์นั้นๆ เช่น เมื่อคลิกที่ภาพพีซี ไอพี 161.246.6.70 ก็จะมีปรากฏ ดังรูป (กลาส pview จะถูกใช้ในขณะนี้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Personal Computer : 161.246.6.70 - Microsoft Internet Explorer

Address <http://161.246.6.160/pc670.shtml>

IP : 161.246.6.70

System Group

Variable	Value
Description	Hardware: x86 Family 5 Model 4 Stepping 3 AT/AT COMPATIBLE - Software: Windows NT Version 4.0 (Build Number: 1381 Uniprocessor Free)
ObjectID	1.3.6.14.1.311.1.1.3.1.1
Contact	Computer Digit20

รูปที่ 5-10 ข้อมูลของ 161.246.6.70 ส่วนบน

Personal Computer : 161.246.6.70 - Microsoft Internet Explorer

Address <http://161.246.6.160/pc670.shtml>

Interface 1	16598	16598
Interface 2	5029066	600416

[In-Out Statistics](#) | [Octet Statistics](#) |
[Use per Day Statistics](#) |
[InOctet in Interface 1 Statistics \(per Day\)](#) | [OutOctet in Interface 1 Statistics \(per Day\)](#) |
[Comparing Octet In Interface 1 \(per Day\)](#) |
[InOctet in Interface2 Statistics \(per Day\)](#) | [OutOctet in interface2 Statistics \(per Day\)](#) |
[Comparing Octet in Interface2 \(per Day\)](#) |

รูปที่ 5-11 ข้อมูลของ 161.246.6.70 ส่วนล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2.4 กลาส inout

จากรูปที่ 5-11 หากต้องการทราบสถิติการใช้งานของอุปกรณ์ ก็ให้คลิกที่ In-Out Statistics ซึ่งจะปรากฏ ข้อมูลในรูปแบบของตาราง ดังรูป (กลาส inout จะถูกใช้ในขณะนี้)

No.	In	Out	Total of Times
1	Tue Mar 28 21:04:56 GMT+07:00 2000 (backing)	Tue Mar 28 23:30:27 GMT+07:00 2000	2h 25m 31s
2	Tue Mar 28 23:55:30 GMT+07:00 2000	Wed Mar 29 03:21:13 GMT+07:00 2000	3h 25m 43s
3	Wed Mar 29 11:54:17 GMT+07:00 2000	Wed Mar 29 16:42:06 GMT+07:00 2000	4h 47m 49s
4	Wed Mar 29 16:44:00 GMT+07:00 2000	Wed Mar 29 17:47:43 GMT+07:00 2000	1h 3m 43s
5	Wed Mar 29 19:04:07 GMT+07:00 2000	Wed Mar 29 20:45:04 GMT+07:00 2000 (playing)	1h 40m 57s

รูปที่ 5-12 สถิติการเปิดปิดอุปกรณ์ ไอพี 161.246.6.70

backing

หมายถึง อุปกรณ์นั้น เริ่มทำงานก่อนที่ จาวาแอปพลิเคชัน ซึ่งมีหน้าที่เก็บข้อมูลจากมิบ จะทำงาน

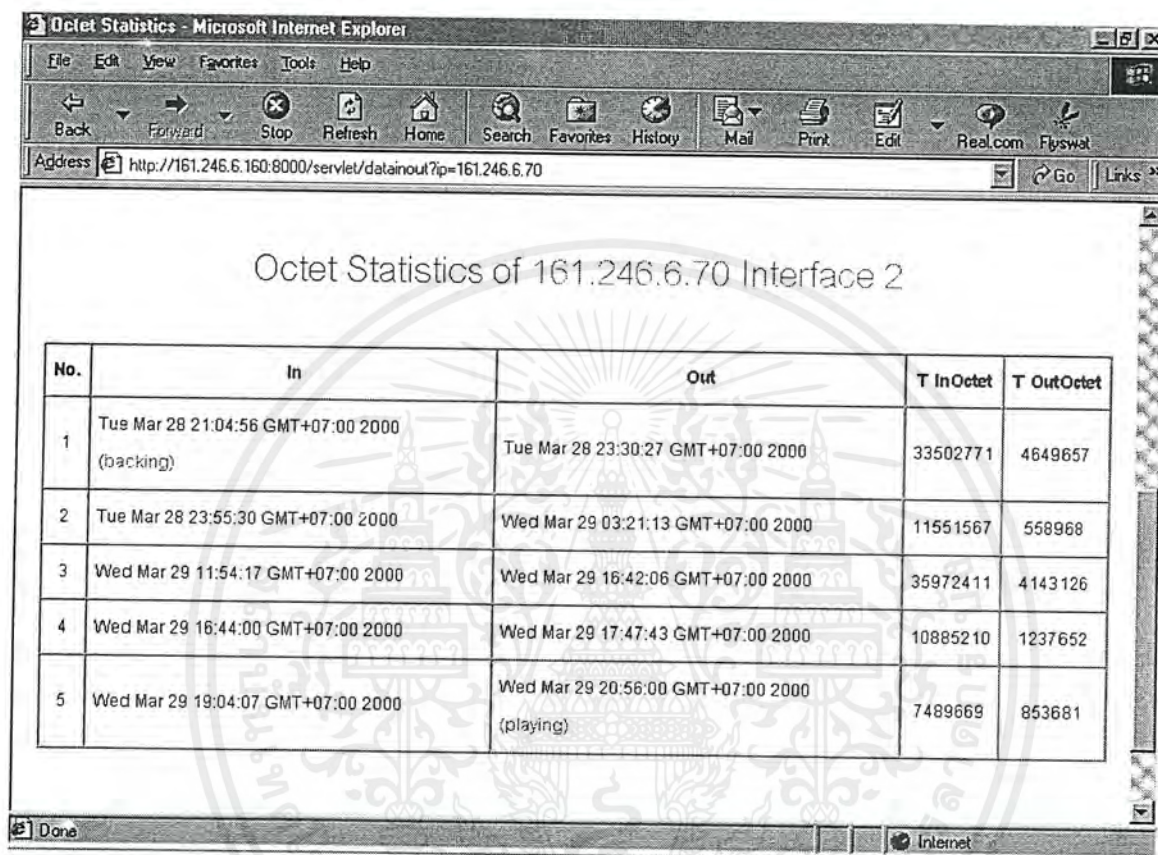
Playing

หมายถึง อุปกรณ์นั้น ในขณะนี้ยังคงทำงานอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2.5 กلاس datainout

จากรูปที่ 5-11 หากต้องการทราบจำนวนข้อมูลที่เข้าและจำนวนข้อมูลที่ออกในแต่ละครั้งของการเปิดและปิดอุปกรณ์ ก็ให้คลิกที่ Octet Statistics ซึ่งจะปรากฏข้อมูลในรูปแบบของตาราง ดังรูป (กلاس datainout จะถูกใช้ในขณะนี้)



No.	In	Out	T InOctet	T OutOctet
1	Tue Mar 28 21:04:56 GMT+07:00 2000 (backing)	Tue Mar 28 23:30:27 GMT+07:00 2000	33502771	4649657
2	Tue Mar 28 23:55:30 GMT+07:00 2000	Wed Mar 29 03:21:13 GMT+07:00 2000	11551567	558968
3	Wed Mar 29 11:54:17 GMT+07:00 2000	Wed Mar 29 16:42:06 GMT+07:00 2000	35972411	4143126
4	Wed Mar 29 16:44:00 GMT+07:00 2000	Wed Mar 29 17:47:43 GMT+07:00 2000	10885210	1237652
5	Wed Mar 29 19:04:07 GMT+07:00 2000	Wed Mar 29 20:56:00 GMT+07:00 2000 (playing)	7489669	853681

รูปที่ 5-13 สถิติของจำนวนข้อมูลที่เข้า และ จำนวนข้อมูลที่ออก ในแต่ละอินเทอร์เน็ตเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 ตรีพมบของอุปกรณ (ที่เลือกนำมาแสดงผล)

5.5.1 กลุ่ของพีซี (PC)

ประกอบไปด้วย

➤ System group

- Description

คำบรรยาย ของอุปกรณ เช่น ฮาร์ดแวร์ ระบบปฏิบัติการ ที่ใช้ เป็นต้น

- ObjectID

ผู้ผลิตของอุปกรณ ซึ่งจะแสดงอยู่ในรูปแบบของอ็อบเจกต์ไอดี (Object ID)

- Uptime

จำนวนเวลาที่อุปกรณทำงานในครั้งปัจจุบัน

- Contact

ข้อมูลในการติดต่อ โดยอาจจะอยู่ในรูปของอีเมล (e-mail)

- Name

ชื่อของอุปกรณ

- Location

ตำแหน่งที่อยู่ของอุปกรณ

- Service

ค่าตัวเลข ซึ่งจะทำให้ทราบได้ว่า อุปกรณนี้มีการบริการอยู่ใน เลเยอร์ (layer) ไตบ้าง

➤ Interface group

- Description

คำบรรยาย ของอินเทอร์เฟซ เช่น PCI Bus Master Adapter เป็นต้น

- Type

ชนิดของอินเทอร์เฟซ เช่น ethernet-csmacd เป็นต้น

- Speed

ความเร็วของอินเทอร์เฟซ เช่น 10000000 เป็นต้น

- Inoctet

ผลรวมค่าอ็อกเตตเข้า จนถึงเวลาปัจจุบัน

- Outoctet

ผลรวมค่าอ็อกเตตออก จนถึงเวลาปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.2 กลุ่มของเซิร์ฟเวอร์ (Server)

ประกอบไปด้วย

- **System group**
มีรายละเอียดเช่นเดียวกับกลุ่มของพีซี
- **Interface group**
มีรายละเอียดเช่นเดียวกับกลุ่มของพีซี
- **Ethernet Interface MIB**
 - **SingleCollisionFrames**
จำนวนเฟรมที่มีการชน (collision) 1 ครั้ง ก่อนการส่งจะเสร็จสมบูรณ์
 - **MultipleCollisionFrames**
จำนวนเฟรมที่มีการชนมากกว่า 1 ครั้ง ก่อนการส่งจะเสร็จสมบูรณ์
 - **DeferredTransmissionsFrames**
จำนวนเฟรมที่การพยายามในการส่งครั้งแรกถูกดีเลย์ (delay) ไป เนื่องจากตัวกลาง (medium) ไม่ว่าง

5.5.3 กลุ่มของสวิตช์ (Switch)

ประกอบไปด้วย

- **System group**
มีรายละเอียดเช่นเดียวกับกลุ่มของพีซี
- **Interface group**
มีรายละเอียดเช่นเดียวกับกลุ่มของพีซี
- **RMON group**
 - **Packets**
จำนวนรวมของแพ็กเก็ต (Packet) ภายใต้อินเทอร์เฟซ
 - **CRCAAlignErrors**
จำนวนรวมของแพ็กเก็ต ที่มีขนาดถูกต้อง (ขนาด 64 ถึง 1,518 อ็อกเตต) แต่มี CRC error หรือ alignment error (ไม่เป็นจำนวนเต็มของอ็อกเตต)
 - **Collisions**
การประมาณค่าของจำนวนการชน (Collision) ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

เอสเอ็นเอ็มพีเอเจนต์ (SNMP Agent)

ในส่วนของระบบบริหารเครือข่ายส่วนที่สำคัญ มี 2 ส่วนคือ ส่วนของสถานีบริหารและส่วนของ เอเจนต์ ซึ่งใช้เป็นตัวจัดการอุปกรณ์ในระบบเครือข่าย โดยใช้ส่งข้อมูลที่จำเป็นในการบริหารผ่านมาทางเอสเอ็นเอ็มพี โพรโตคอล ซึ่งช่วยให้ทำงานเข้ากับ สถานีบริหารต่างๆ ที่ใช้งานอยู่ในระบบเครือข่าย

ตัวเอเจนต์จะใช้ระบบที่คล้ายคลึงกับส่วนของสถานีบริหารเช่นการเข้ารหัสและการถอดรหัส โดยเอเจนต์ จะต่างกับส่วนบริหารตรงที่จำเป็นต้องทำงานอยู่ตลอดเวลาเพื่อทำการตรวจสอบ และรอรับสัญญาณขอข้อมูลของ สถานีบริหาร ตรงกับที่เอเจนต์ตัวนั้นๆ ใบริการอยู่หรือไม่

6.1 ส่วนประกอบของ SNMP Agent

ตัวเอเจนต์จะทำการตัดสินใจว่าควรส่งข้อมูลใดกลับไป โดยดูจากสิ่งที่ได้จากสถานีบริหารส่งมาให้

1. ชนิดของสัญญาณขอข้อมูล
2. Object ID
3. Community name

ซึ่งขั้นตอนการประมวลผลของเอสเอ็นเอ็มพีเอเจนต์จะประกอบด้วย

6.1.1 การประมวลผลข้อมูล UDP (UDP Process)

กระบวนการติดต่อดสื่อสารของเอเจนต์กับ สถานีบริหารในที่นี้ใช้ TCP/IP เป็นหลัก โดยใช้การเชื่อมต่อแบบ UDP ซึ่งอยู่ในชั้น Transport Layer กระบวนการประมวลผลแพ็กเกจ UDP จะถูกมองว่าเป็นโพรเซสแม่ (Server Process) ในทางตรงข้าม สถานีบริหารจะถูกมองเป็นโพรเซสลูก (Client Process) ที่ขอข้อมูลจากเอสเอ็นเอ็มพีเอเจนต์ดังนั้นกระบวนการก็จำเป็นที่จะต้องให้สถานีบริหารทำการวนลูปรับข้อมูลร้องขอ เมื่อใดที่มีแพ็กเกจส่งเข้ามาจะทำการกระตุ้นการทำงานของเอสเอ็นเอ็มพีเอเจนต์ให้ประมวลผลข้อมูลที่ได้รับเข้ามา แปลความหมายของคำสั่งที่ได้รับ ประมวลผลคำสั่งที่ได้รับมาแล้ว ส่งข้อมูลตอบสนองไปยังสถานีบริหาร อย่างไรก็ตามหากเกิดการผิดพลาดขึ้นมาเอสเอ็นเอ็มพีเอเจนต์นั้นๆจะต้องส่งข้อมูลตอบสนองความผิดพลาดให้สถานีบริหารทราบด้วย

6.1.2 กระบวนการอ่านค่าเริ่มต้นของเอสเอ็นเอ็มพีเอเจนต์

กระบวนการอ่านค่าเริ่มต้นหรือ กระบวนการปรับแต่ง (Configuration Process) คือการกำหนดค่าเริ่มต้นให้กับระบบ ในที่นี้คือการกำหนดค่าตัวแปรที่เอสเอ็นเอ็มพีเอเจนต์ต้องใช้ในตอนแรกที่เริ่มทำงาน ตัวอย่างเช่น ค่า Community name ที่ต้องมีการกำหนดให้ก่อนทุกครั้งเพื่อใช้ในการตรวจสอบสิทธิ์ในการขอข้อมูล หรือในตอนแรกที่เริ่มทำงานเอสเอ็นเอ็มพีเอเจนต์จะเก็บค่าเวลาของระบบไว้ใช้คำนวณค่า sysUptime เป็นต้น ทั้งนี้การอ่านค่าเริ่มต้นของเอสเอ็นเอ็มพีเอเจนต์บนอุปกรณ์แต่ละชนิดอาจมีการเริ่มต้นที่แตกต่างกัน ขึ้นอยู่กับอุปกรณ์ที่เราทำการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น เราเตอร์มีการอ่านค่าและ กำหนดค่าตารางการเรดต์ (Routing Table) ในตอนเริ่มต้นการทำงานของระบบ (Boot Strap) ก่อนที่เอสเอ็นเอ็มพีเอเจนท์ในตัวเราเตอร์จะเริ่มทำงาน จำเป็นต้องได้ค่าจากตารางการเรดต์ เพื่อใช้เป็นค่าเริ่มต้นให้กับโหนดภายในฐานข้อมูลการจัดการ (MIB)

6.1.3 กระบวนการอ่านค่าตัวแปรและกำหนดค่ากับอุปกรณ์

หัวใจหลักของการประมวลผล SNMP คือการอ่านค่าอุปกรณ์ที่เราสนใจ ผ่านโปรแกรมย่อยที่ควบคุมอุปกรณ์นั้นๆอยู่ โปรแกรมย่อยที่กล่าวถึงนี้ จะเป็นได้ทั้งการกำหนดค่าตัวแปรให้อุปกรณ์และการอ่านค่าของตัวแปรของอุปกรณ์นั้นๆ เช่น โปรแกรมสำหรับอ่านและตั้งค่าเวลาของระบบ โดยโปรแกรมเหล่านี้จะถูกแปลงเป็นคำสั่ง SNMP อีกทีหนึ่ง โดยโปรแกรมที่เกี่ยวกับการควบคุมจะถูกเปลี่ยนเป็นคำสั่ง SetRequest และไม่ว่ากรณีใดๆ สถานีบริหารจะตอบรับผลที่เกิดขึ้นจากการใช้คำสั่งนั้นๆด้วยคำสั่ง GetRespond ทุกครั้ง เช่นเดียวกันกับการขอข้อมูลด้วยการเรียกคำสั่ง GetRequest ฎจะตอบสนองกลับมาด้วย GetRespond ทุกครั้ง ไม่ว่าจะเป็นการตอบรับข้อมูลที่ส่งให้หรือการบอกความผิดพลาดที่เกิดขึ้น

6.1.4 กระบวนการหาค่าตัวแปรและ Object ID

การจัดการ Object ID เป็นการกำหนดตัวแปรข้อมูลต่างๆออกมาเป็นฐานข้อมูล ซึ่งใช้เก็บข้อมูลชนิดต่างๆ รวมถึง Object ID ด้วย ซึ่งถูกมองได้เป็นฐานข้อมูลในระดับลอจิกคอล ซึ่งฐานข้อมูลการจัดการดังกล่าวอาจถูกเก็บในฐานข้อมูลที่เห็นได้จริง (Real Database) หรือฐานข้อมูลที่มาจากการจัดการตัวแปรที่มาจากหน่วยความจำที่เป็นตัวแปรแบบอาร์เรย์ แล้วนำข้อมูลดังกล่าวมาทำการเทียบเคียงกับฐานข้อมูลจัดการ ทำให้มองเสมือนว่ามีฐานข้อมูลอยู่จริง ซึ่งในความเป็นจริงอาจมีหรือไม่มีก็ได้ ทั้งนี้ขึ้นอยู่กับผู้ออกแบบเอสเอ็นเอ็มพีเอเจนท์ที่ว่าต้องการให้มีหรือไม่ ตัวอย่าง Agent ที่มีฐานข้อมูลอยู่จริงได้แก่ เอสเอ็นเอ็มพีเอเจนท์ของ Oracle และกรณีที่ไม่มีฐานข้อมูลอยู่จริงได้แก่ SNMP Agent Service ของ Windows NT

6.2 ขั้นตอนการทำงานของเอสเอ็นเอ็มพีเอเจนท์

ขั้นตอนการทำงานของเอสเอ็นเอ็มพีเอเจนท์แบ่งย่อยออกได้เป็นขั้นตอนได้ดังนี้

6.2.1 **ขั้นเตรียมการ (Initialization)** จะเป็นการเริ่มการทำงานของเอเจนท์โดยการสั่งให้ทำงานเอเจนท์ จะทำการกำหนดค่าเริ่มต้นให้เอสเอ็นเอ็มพีเอเจนท์โปรแกรม แล้วทำการเตรียมการเชื่อมต่อกับระบบเครือข่ายโดยกำหนดให้เตรียมการเชื่อมต่อแบบเซิร์ฟเวอร์ (Server Socket) จากนั้นระบบจะรอการเรียกใช้จากเครื่องลูกที่เป็นสถานีบริหาร

6.2.2 **ขั้นการรับการร้องขอข้อมูล (GetRequest)** เอสเอ็นเอ็มพีเอเจนท์จะทำการรับข้อมูลที่เป็นแพ็กเกจ UDP มาทำการเปลี่ยนให้เป็นข้อมูลคิบบจำนวนหนึ่ง ซึ่งถูกเข้ารหัสแบบ BER (Basic Encoding Rule) เอาไว้

6.2.3 **ขั้นการถอดรหัส (BER decoding)** จะทำการถอดรหัส BER ให้เป็นภาษา ASN.1 ที่บรรจุข้อมูล และคำสั่งที่ใช้ในระบบ SNMP เอาไว้

6.2.4 **การแปลความหมายของภาษา ASN.1 (ASN.1 decoding)** จะทำการแปลความหมายของภาษา ASN.1 ที่ได้รับมา โดยการเปรียบเทียบกับตารางคำสั่งที่มีอยู่ในโปรแกรม

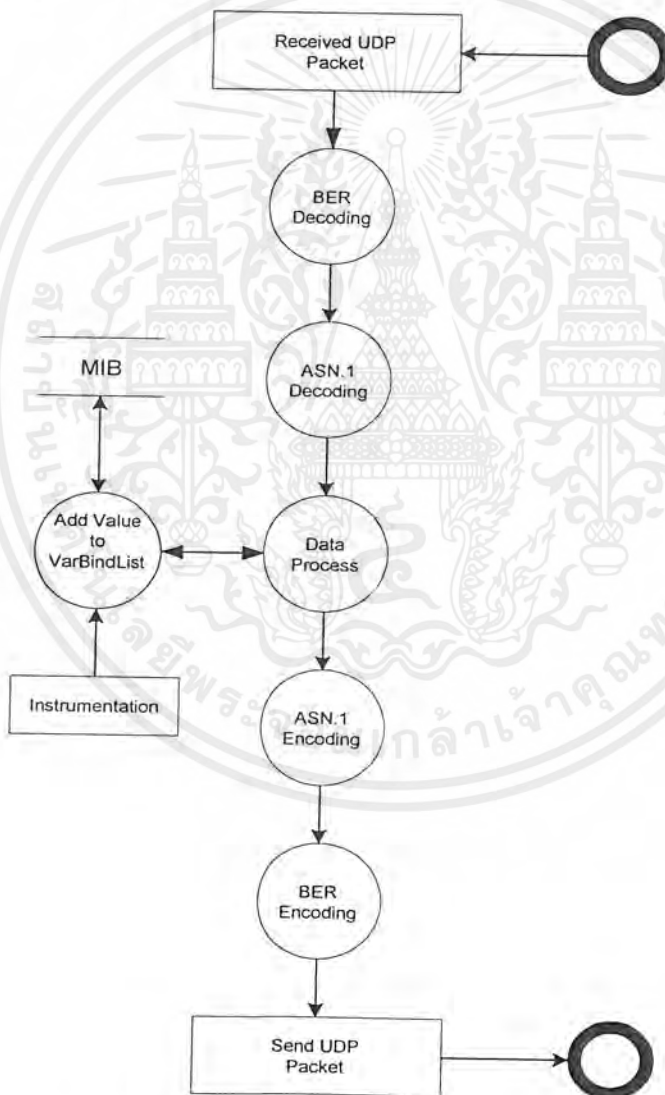
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.5 ขั้นตอนการปฏิบัติงาน (Datagram Processing) เป็นการรับคำสั่งจากที่ระบุใน ASN.1 ที่ได้รับมา จากนั้นจึงนำคำสั่งที่ได้รับมา ไปประมวลผลให้ได้ข้อมูลที่ต้องการ การตรวจสอบชนิดของข้อมูลที่ต้องการดูได้จาก Object ID ที่ส่งมาพร้อมกับคำสั่ง เมื่อได้ข้อมูลที่ต้องการแล้วจึงส่งไปยังส่วนปฏิบัติงานเพื่อเตรียมจัดส่งข้อมูลที่ได้รับต่อไป

6.2.6 การเตรียมชุดคำสั่งตอบกลับ (ASN.1 Encoding) จะทำการเตรียมข้อมูลที่ได้มาจากการขั้นตอน 6.2.5 มาทำการจัดลงภาษา ASN.1 พร้อมคำสั่ง GetRespond เพื่อให้ตัวสถานีบริหารทำการรับข้อมูลที่ได้

6.2.7 ขั้นตอนการเข้ารหัส (BER Encoding) จะนำภาษาที่เตรียมไว้ในขั้นที่ 6.2.6 มาทำการเข้ารหัสเพื่อเตรียมส่งไปยังสถานีบริหาร

6.2.8 การส่งกลับแพ็คเกจ (GetRespond) จะทำการส่งข้อมูลที่เข้ารหัสแล้วไปยังสถานีบริหารซึ่งรอรับข้อมูลอยู่แล้ว เพื่อให้สถานีบริหารนำข้อมูลที่ได้รับไปจัดการตามขั้นตอนต่อไป



รูปที่ 6-1 แสดงขั้นตอนการทำงานของเอสเอ็นเอ็มพีเอเจนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การพัฒนาเอสเอ็นเอ็มพีเอเจนต์โดยใช้ JAVA

หากแบ่งตามหน้าที่แล้ว อาจแบ่งได้เป็น

6.3.1 ขั้นตอนการรับแพ็กเกจ PDU

ใช้ DatagramSocket ในการรอรับแพ็กเกจที่เข้ามาโดยการกำหนดหมายเลขพอร์ตเป็น 161 เมื่อรับแล้วก็จะนำไปเก็บไว้ใน DatagramPacket เพื่อรอการนำไปใช้

```
//peer.port = 161 peer.address = localhost
//peer.mtu = 1500
DatagramSocket socket = new DatagramSocket(peer.port, peer.address);
byte[] buffer = new byte[peer.mtu];
DatagramPacket request = new DatagramPacket(buffer, buffer.length);
socket.receive(request);
```

6.3.2 ขั้นตอนการถอดรหัสแปลความหมาย

ใช้ Class ASN1Value ในการถอดรหัส BER coding ออกมาเป็นภาษา ASN1

```
public Asn1Value service(Asn1Value request)
    throws IOException, SnmpException
{
    err.print("request: "); request.print(err); err.println();
    Asn1Value response = service(SnmpMessage.decode(request)).encode();
    err.print("response: "); response.print(err); err.println();
    return response;
}
```

6.3.3 ขั้นตอนการเตรียมข้อมูล

ใช้ Class VarBind ในการเตรียมข้อมูลลงใน Variable Binding List

```
public Varbind service(int requestTag, Varbind varbind)
{
    //return new Varbind(varbind.name, new Asn1Value("kai"));
    return new Varbind(new OID("1.3.6.1.2.1.1.1"), new Asn1Value("kai"));
}
```

6.3.4 ขั้นตอนการเข้ารหัส

ใช้ Class ASN1Value method Service ในการเข้ารหัส

```
public byte[] service(InputStream request)
    throws IOException, SnmpException
{
    return service(Asn1Value.decode(request)).encode();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.5 การส่งข้อมูลกลับ

ใช้ DatagramPacket ในการส่งข้อมูลกลับ

```
byte[] response = service(in);
    DatagramPacket responsePacket =
        new DatagramPacket(response, response.length, request.getAddress(), request.getPort());
    socket.send(responsePacket);
```

6.4 ตัวอย่างเอสเอ็นเอ็มพีเอเจนต์

```
//Agent
import jmgmt.asn1.*;
import jmgmt.snmp.io.*;
import java.net.*;
import java.io.*;
import java.util.*;

public class Agent implements Runnable {
    static PrintWriter err = new PrintWriter(System.err);
    /** the server peer for this agent instance.
     */
    SnmpPeer peer;
    public Agent (SnmpPeer peer)
    {
        this.peer=peer;
    }
    public byte[] service(InputStream request)
        throws IOException, SnmpException
    {
        return service(Asn1Value.decode(request)).encode();
    }
    /** decode a SNMP message and service it.
     */
    public Asn1Value service(Asn1Value request)
        throws IOException, SnmpException
    {
        err.print("request: "); request.print(err); err.println();
        Asn1Value response = service(SnmpMessage.decode(request)).encode();
        err.print("response: "); response.print(err); err.println();
        return response;
    }
    /** service a message.
     */
    public SnmpMessage service(SnmpMessage request)
        throws SnmpException
    {
        if (request.community != peer.community) {
            // wrong community: NO_SUCH_NAME
            // ... ..
        }
        SnmpPdu responsePdu = service(request.pdu);
        return new SnmpMessage( request.community, responsePdu );
    }
    /** service a PDU.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*/
public SnmpPdu service(SnmpPdu request)
{
    Varbind[] varbinds = request.getVarbindArray();
    SnmpPdu response = new SnmpPdu(SMI.GET_RESPONSE_ID, request.requestId);
    response.addVarbinds(service(request.id, varbinds));
    return response;
}
/** service a sequence of varbinds.
*/
public Varbind[] service(int requestTag, Varbind[] varbinds)
{
    Varbind[] responseVarbinds = new Varbind[varbinds.length];
    for (int i=0; i<varbinds.length; i++)
        responseVarbinds[i] = service(requestTag, varbinds[i]);
    return responseVarbinds;
}
/** service a single variable of a request.
*/
public Varbind service(int requestTag, Varbind varbind)
{
    //return new Varbind(varbind.name, new Asn1Value("kai"));
    return new Varbind(new OID("1.3.6.1.2.1.1.1"), new Asn1Value("kai"));
}
/** answer requests. */
public void run()
{
    try {
        DatagramSocket socket = new DatagramSocket(peer.port, peer.address);
        byte[] buffer = new byte[peer.mtu];
        while (true) {
            // must create a new DatagramPacket each time. otherwise getLength is not correct.
            DatagramPacket request = new DatagramPacket(buffer, buffer.length);
            socket.receive(request);
            byte[] data = request.getData();
            int length = request.getLength();
            err.print("received bytes = " );
            err.println(length);
            Asn1Value.printHex(err, request.getData(), length);
            err.println();
            err.flush();
            ByteArrayInputStream in = new ByteArrayInputStream(data, 0, length);
            byte[] response = service(in);
            DatagramPacket responsePacket =
                new DatagramPacket(response, response.length, request.getAddress(), request.getPort());
            socket.send(responsePacket);
            //err.flush();
        }
    } catch (Exception e) {
        err.println("Agent: caught Exception, terminating.");
        e.printStackTrace();
    }
}

/** run an agent demo on the standard port. */
public static void main(String args[])
    throws IOException
{

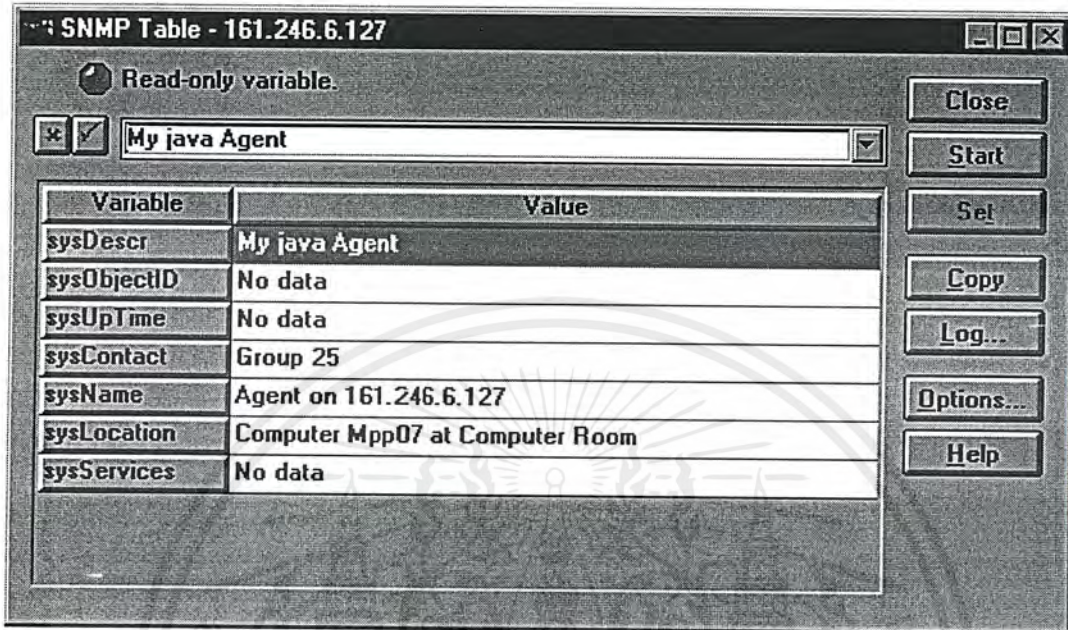
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

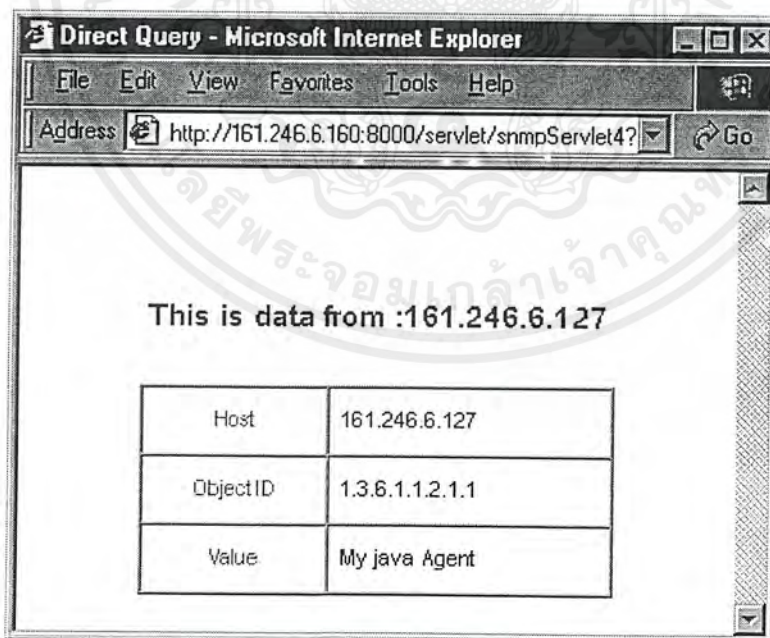
```

SnmPeer peer = new SnmPeer("Mpp07", InetAddress.getByName("161.246.6.127"), SMI.PUBLIC
);
new Agent(peer).run();
}
}

```



รูปที่ 6-2 แสดงการทดสอบเอเจนต์ด้วย HP Openview



รูปที่ 6-3 แสดงการทดสอบเอเจนต์ผ่านโปรแกรม snmpServlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

บทวิจารณ์และสรุปผล

7.1 สรุปผลการดำเนินงาน

สำหรับโครงการในส่วนแรก นั้น จะช่วยให้ผู้ใช้สามารถที่จะดูข้อมูลต่างๆ ที่เกี่ยวข้องกับการบริหารเครือข่ายได้ โดยผ่านทางเว็บเพจ บนเครือข่ายอินเทอร์เน็ต โดยข้อมูลสำหรับการบริหารเครือข่ายที่ได้นี้ จะใช้ภาษาจาวา เขียนเป็นแอปพลิเคชัน ดึงข้อมูลมาเก็บในฐานข้อมูล โดยผ่านทาง SNMP (Simple Network Management Protocol) สำหรับฝั่งผู้ใช้นั้น จะดูการแสดงผลบนหน้าเว็บเพจ โดยการแสดงผลนี้ จะมาจากการใช้ จาวาเซิร์ฟเล็ต (Java Servlet) ทำการประมวลผลค่าข้อมูลที่อยู่ในฐานข้อมูล ออกมาแสดงผลในรูปแบบ ลักษณะต่างๆ เป็นส่วนใหญ่

กล่าวโดยสรุป สำหรับโครงการในส่วนนี้ สามารถตอบสนองการบริหารเครือข่าย (Network Management) ได้ดังนี้

- การบริหารทางด้านคอนฟิก (Configuration Management)

ในแง่ของ การบริหารทางด้านคอนฟิก นั้น เราสามารถขอรายละเอียดของอุปกรณ์แต่ละตัวได้ เช่น อุปกรณ์แต่ละตัวนั้น มีฮาร์ดแวร์ที่ใช้ ระบบปฏิบัติการที่ใช้ เป็นอย่างไร มีอินเทอร์เน็ตประเภทใดบ้าง เป็นต้น

- การบริหารทางด้านประสิทธิภาพ (Performance Management)

ในแง่ของ การบริหารทางด้านประสิทธิภาพ นั้น เราสามารถที่จะทำการมอนิเตอร์ดู อ็อกเทต (octet) เข้า-ออก ในวันเวลาต่างๆ ตลอดจนความผิดพลาดที่เกิดขึ้นบนอินเทอร์เน็ตชนิดต่างๆ ของสวิตช์ได้ โดยเฉพาะอย่างยิ่งในการใช้ RMON จะทำให้เราทราบถึงสาเหตุความผิดพลาด ของแพ็กเก็ตในรูปแบบต่างๆ ตลอดจนผลลัพธ์โดยรวมของเครือข่ายย่อย (subnetwork) นั้นๆ ได้เป็นอย่างดี

ข้อมูลเหล่านี้ จะเป็นข้อมูลที่จะช่วยในการพิจารณาเพื่อปรับปรุงแก้ไข อันจะนำมาซึ่งประสิทธิภาพ ที่ดีขึ้นต่อไป (ตัวอย่างเช่น เมื่อตรวจสอบดูจากสถิติการเกิดความผิดพลาดของแพ็กเก็ต และพบว่า มีการเกิดความผิดพลาดที่เพิ่มมากขึ้นผิดปกติ ตรงนี้ก็จะนำไปสู่การตรวจสอบหาสาเหตุของการเกิดความผิดพลาดที่ผิดปกติได้)

- การบริหารทางด้านความปลอดภัย (Security Management)

โครงการนี้ ยังไม่จัดทำในด้านนี้

- การบริหารทางด้านความผิดพลาด (Fault Management)

ในแง่ของ การบริหารทางด้านความผิดพลาด นั้น เราสามารถที่จะตรวจสอบการทำงานของอุปกรณ์แต่ละอุปกรณ์บนหน้าเว็บเพจ ได้ว่า ขณะนี้อุปกรณ์ยังทำงานอยู่หรือไม่ ซึ่งหากมีการดาวน์โหลด สีของภาพอุปกรณ์บนหน้าเว็บเพจก็จะเปลี่ยนเป็นสีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พร้อมกับส่งสัญญาณเตือนทางเพจเจอร์ ให้แก่ผู้บริหารเครือข่าย อันเป็นการแจ้งเตือนว่าขณะนี้ อุปกรณ์นั้นๆ เกิดการดาวน์โหลดขึ้น โดยอัตโนมัติ

- การบริหารทางการเก็บบันทึก (Accounting Management)

ในแง่ของการบริหารทางการเก็บบันทึกนั้น เราจะทำการบันทึกสถิติต่างๆ ภายในระบบ เก็บไว้ในฐานข้อมูล สำหรับตัวอย่างของสถิติดังกล่าวนี้ ก็เช่น สถิติการใช้เครื่องพีซี (จะทราบว่า พีซีเครื่องนี้ ถูกใช้งานวันไหน เวลาเท่าไรบ้าง) สถิติอินเทอร์เน็ตที่ได้รับและส่งออก ในอินเทอร์เน็ตต่างๆ ต่อวัน ของอุปกรณ์ เป็นต้น

สำหรับโครงการส่วนที่ 2 การพัฒนาในส่วนของเอสเอ็นเอ็มพีเอเจนต์นั้น พบว่าในการทดสอบการทำงานของเอเจนต์ในระดับพื้นฐานสามารถทำได้ มีการเข้ารหัสถูกต้อง และสามารถกำหนดอ็อบเจกต์ไอดี (Object ID) เป็นไปตามข้อกำหนดของ SNMPv1 ทุกประการ จึงสามารถผ่านการทดสอบกับ สถานีบริหารเครือข่าย ทั้งที่เป็น โปรแกรมสำเร็จรูปและที่สร้างขึ้นมาเอง แต่เนื่องด้วยข้อจำกัดของจาวา ที่ไม่สามารถเข้าถึงการทำงานในระดับต่ำ หรือการควบคุมทางฮาร์ดแวร์ จึงทำให้ไม่สามารถทำงานด้านเอเจนต์ได้อย่างเต็มที่ และเอเจนต์ตัวนี้ยังไม่สามารถทำงานพร้อมกับเอเจนต์ตัวอื่นๆ ได้เนื่องจากการกำหนดพอร์ตที่ 161 นั้นได้ถูกใช้งานไปก่อนแล้ว

7.2 แนวทางการพัฒนาต่อ

สำหรับโครงการในส่วนแรก นั้น ในส่วนที่เป็นแอปพลิเคชัน ที่ทำการเก็บข้อมูลจาก MIB ของอุปกรณ์อยู่เรื่อยๆ นั้นจะต้องมีการส่ง message ออกไป query ที่ตัวอุปกรณ์อยู่เป็นระยะๆ เช่น ทุกๆ 30 วินาที หากในส่วนนี้ สามารถทำเป็นลักษณะของการ trap คือ เมื่ออุปกรณ์หยุดทำงาน ไป ก็ให้มีการส่ง message ตลอดจนค่าต่างๆ ที่อยู่ใน MIB ณ ขณะนั้น มาให้โดยอัตโนมัติได้ ก็จะทำให้เกิดประสิทธิภาพที่มากขึ้น ต่อระบบเครือข่าย

สำหรับส่วนที่ 2 นั้น เนื่องจากการที่จาวาไม่สนับสนุนการทำงานโดยใช้ภาษาระดับต่ำซึ่งเข้าถึงฮาร์ดแวร์ได้ดีกว่าจาวาและความเร็วในการทำงานที่ยังคงต่ำอยู่มาก จึงไม่แนะนำให้ใช้จาวาในการพัฒนาเอสเอ็นเอ็มพีเอเจนต์

บทที่ 8

การติดตั้งองค์ประกอบของโครงการ

8.1 ส่วนที่เป็นของการบริหารเครือข่าย (Network Management)

8.1.1 JAVA Development Kit หรือ JDK

JDK เป็นเครื่องมือสำหรับพัฒนาภาษาจาวาของบริษัท SUN Microsystems inc. ซึ่งใช้เป็นเครื่องมือหลักในการพัฒนาโครงการนี้ ตัว JDK มี 2 ประเภทคือ

1. JAVA™ 2 SDK, Standard Edition, v1.2.2 ในชุดโปรแกรมจะประกอบด้วย 2 ส่วนคือ Development System ที่ใช้สำหรับการแปลภาษาจาวา และส่วนของ Runtime Environment ที่ใช้ในการทำงานกับโปรแกรมภาษาจาวา
2. JAVA™ 2 Runtime Environment, v1.2.2 จะต่างกับชุด Standard Edition ตรงที่มีเฉพาะส่วนของ Runtime Environment เท่านั้น ทำให้ตัวโปรแกรมติดตั้งนี้มีขนาดเล็กกว่ามาก

ในระบบวินโดวส์การติดตั้งจะไม่แตกต่างจากการติดตั้งโปรแกรมทั่วไปเพราะ JDK มีโปรแกรมติดตั้งที่ทำงานอยู่บนวินโดวส์แล้ว โดยมีข้อควรระวังคือ หากมีการติดตั้ง JDK นี้ในเวอร์ชันอื่นๆ ควรทำการถอนโปรแกรมออกจากระบบให้เรียบร้อยก่อนที่จะทำการติดตั้งเวอร์ชันที่ใหม่กว่าลงไป

- 1 ส่วน Development System เป็นส่วนที่ใช้สำหรับเขียนภาษา JAVA ประกอบด้วย
 - javac เป็น JAVA compiler ใช้แปลตัวภาษาจาวา
 - javap เป็น JAVA dissembler ใช้ดูรายละเอียดเกี่ยวกับคลาสและ เมธอดในแฟ้มสกุล .class
 - javah ใช้สร้าง แฟ้ม Header ของภาษา C ที่ต้องการเรียกใช้คลาสของจาวา
 - javadoc ใช้สร้างแฟ้ม HTML ที่แสดงรายละเอียดของคลาสทั้งหมด
- 2 ส่วน Runtime System เป็นส่วนที่ใช้ให้จาวาทำงานประกอบด้วย
 - java เป็น โปรแกรมที่ใช้เรียกโปรแกรมภาษาจาวาขึ้นมาทำงาน โดยจะใช้เรียกโปรแกรมที่เขียนขึ้นเพื่อใช้เป็นแอปพลิเคชัน
 - javaw การทำงานเช่นเดียวกับ java.exe แต่จะแตกต่างกันที่ javaw.exe จะสร้าง โปรเซสของตัวเองขึ้นมาโดยไม่จำเป็นต้องทำงานในโหมมดอส
 - Applet viewer เป็นโปรแกรมสำหรับสร้างสภาพแวดล้อมในการรันแอปพลิเคชัน

การติดตั้งในส่วนของ JDK จะเป็นการติดตั้งสำหรับระบบวินโดวส์ ซึ่งจะเป็นแฟ้มโปรแกรมติดตั้งจะติดตั้งส่วนประกอบพื้นฐานที่จำเป็นทั้งหมดของจาวา รวมถึงแฟ้มระบบ (Registry File) ที่ใช้ใน JDK แต่การติดตั้งนี้ยังขาดส่วนของการปรับแต่งระบบให้อำนวยความสะดวกต่อการเขียนโปรแกรม และการใช้งานจะพบข้อปลีกย่อยจำนวนหนึ่ง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1.1.1 การติดตั้ง PATH ของระบบ

เนื่องด้วย JDK เป็นโปรแกรมพัฒนาที่ทำงานบนคอมพิวเตอร์ เป็นหลักดังนั้นการเรียกใช้โปรแกรมจึงยังต้องผ่านตัวรับคำสั่ง (Command Line) ซึ่งจะประสบปัญหาขุ่นยากในการเรียกใช้งานโปรแกรม จึงต้องทำการปรับแต่งตัวแปร PATH ให้สามารถหาตำแหน่งของโปรแกรมจาวาที่จำเป็นต้องใช้งานได้

- Windows 95/98 เราสามารถปรับแต่งตัวแปร PATH ได้โดยใช้ Sysedit.exe ซึ่งเป็นโปรแกรมในวินโดวส์ โดยเลือกแก้ไขที่เพิ่ม AUTOEXEC.BAT เข้าไปแก้ไขโดยให้ชี้ไปที่ที่เก็บโปรแกรมสำหรับจาวาวีธีนี้จะช่วยให้การเรียกใช้ครั้งต่อไปง่ายขึ้น



รูปที่ 8-1 การใช้ Sysedit ในการแก้ไขเพิ่มระบบ

- Windows NT เช่นเดียวกับใน Windows95/98 การแก้ไขตัวแปรระบบ PATH ให้ใช้ค่าเดียวกัน แต่การแก้ไขให้ใช้ System Properties ซึ่งอยู่ใน Control Panel => System ให้ชี้ไปที่ Environment

8.1.1.2 การแก้ไข Classpath

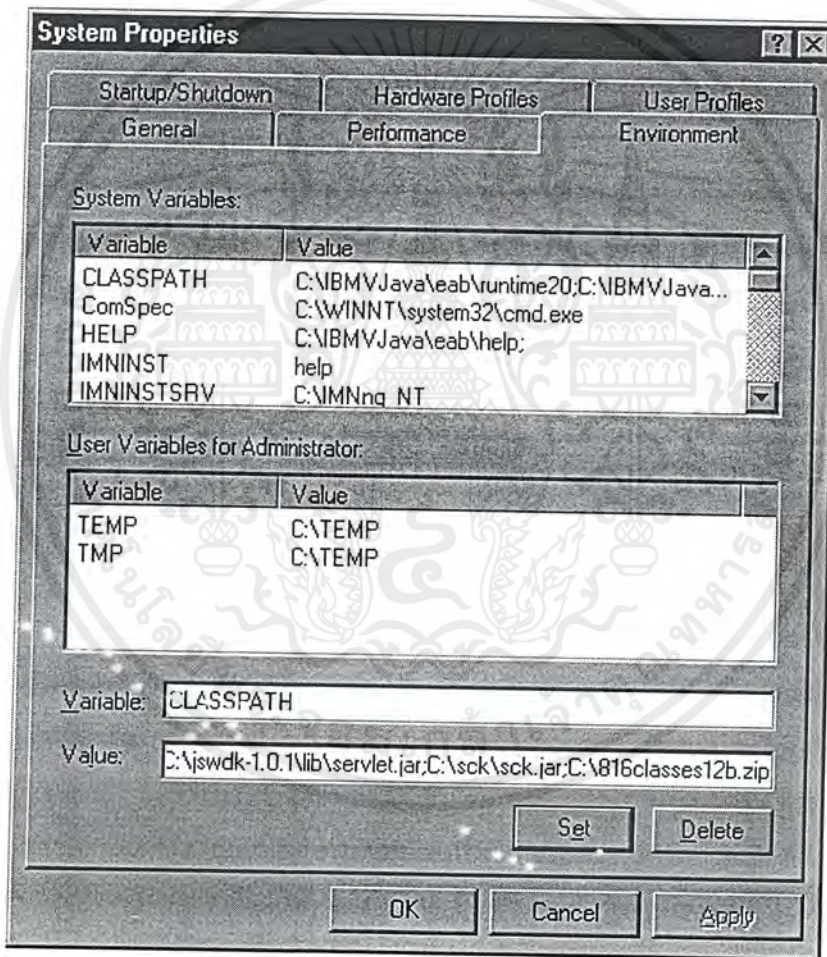
ในการพัฒนาภาษา JAVA มีการใช้คลาสจำนวนมากในการทำงานซึ่งคลาสต่างๆอาจแบ่งออกได้เป็น

- Bootstrap classes เป็นเพิ่มที่เก็บคลาสมมาตรฐานของจาวาไว้ทั้งหมดในการใช้งาน สำหรับในกรณีของ JDK 1.2 คลาสนี้จะถูกเก็บอยู่ใน rt.jar และ i18n.jar ซึ่งในกรณีนี้ JDK จะสามารถหาพบได้โดยไม่ต้องทำการปรับแต่งใดๆอีก แต่สำหรับกรณีของ JDK 1.1 หรือต่ำกว่าเพิ่มนี้จะถูกเก็บอยู่ในชื่อ Class.zip และต้องกำหนดอยู่ในตัวแปรระบบ CLASSPATH ด้วยเสมอ ในกรณีที่ต้องการปรับเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่นของ JDK ที่ใช้งานจาก JDK 1.1 ไปเป็น JDK 1.2 ให้ทำการยกเลิกการประกาศ CLASSPATH ของ Class.zip นี้ด้วย

- Extension classes เป็นคลาสที่ไม่ได้มีอยู่ในการติดตั้งปกติของ JDK แต่เป็นคลาสของ SUN ที่จำเป็นต้องติดตั้งเพิ่มเติมหากต้องการใช้ เช่น เจิร์ฟเล็คเป็นต้น ซึ่งจะอยู่ในรูปของ แพ้ม .jar และถูกเก็บอยู่ใน Extension directory การใช้งานเมื่อติดตั้งให้แตกแพ้นี้ด้วย jar.exe ลงใน Extension directory ก็จะสามารถใช้งานได้ (Extension Mechanism)
- User classes เป็นคลาสที่พัฒนาขึ้นมาเองหรือหากจากที่มีผู้พัฒนาไว้แล้ว ซึ่งจะไม่สามารถใช้ความสามารถของ Extension Mechanism ได้ จำเป็นจะต้องระบุตำแหน่งของคลาสเหล่านั้นโดยระบุในตัวแปรระบบ CLASSPATH หรือกำหนดจากสวิตซ์ -classpath ซึ่งเป็นส่วนหนึ่งในคำสั่งของ JAVA



รูปที่ 8-2 การแก้ไขตัวแปรระบบใน Windows NT

เนื่องด้วยการทำงานของระบบที่จำเป็นต้องใช้คลาสร่วมกันจากหลายๆ แหล่งจึงมีการกำหนดการปรับแต่งตัวแปร CLASSPATH ไว้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในกรณีที่คลาสนั้นไม่ได้ถูกบีบข้อเก็บไว้ในแฟ้มใดๆ จะกำหนด CLASSPATH ให้ชี้ไปยังที่ที่เก็บคลาสนั้นๆ โดยไม่จำเป็นต้องกำหนดชื่อของคลาส
- ในกรณีที่คลาสนั้นไม่ได้ถูกบีบข้อเก็บไว้ในแฟ้มใดๆ แต่มีการกำหนดเป็นแพคเกจ (Package) ให้กำหนด CLASSPATH ชี้ไปที่ไดเรกทอรีที่เก็บไดเรกทอรีรากของแพคเกจนั้นๆ โดยไม่จำเป็นต้องกำหนดชื่อคลาสหรือชื่อ ไดเรกทอรี
- ในกรณีที่มีการรวมคลาส หรือทำการบีบอัดแฟ้มคลาสต่างๆ ไว้รวมกัน ให้กำหนด CLASSPATH ให้ชี้ไปที่แฟ้มนั้นและให้ระบุชื่อด้วย

ในการใช้งาน JDK หากไม่ได้กำหนด CLASSPATH ไว้ล่วงหน้าสามารถกำหนดผ่านสวิตช์ `-classpath` ซึ่งจะมีอยู่ในจาวาทูลทุกตัวด้วยเงื่อนไขเดียวกับกรณีกำหนดตัวแปร CLASSPATH โดยหากกำหนดที่มาของคลาสมากกว่าหนึ่งตัวให้คั่นด้วย ; โดยสามารถนำแต่ละเงื่อนไขมารวมกันได้เช่น

```
java -classpath c:\sck\sck.jar;c:\jmgmt; snmpPing
```

ในกรณีที่ใช้คลาสจำนวนมากและเก็บอยู่ในหลายแฟ้มหรือมีการใช้งานในคลาสหลายๆคลาส จนอาจทำให้ตัวแปรระบบที่กำหนดไว้ไม่พอรองรับความยาวของ CLASSPATH แก้ไขได้โดยการแตกแฟ้ม .jar นั้นออกเป็นแฟ้มย่อยซึ่งโดยทั่วไปภายในจะกำหนดเป็นไดเรกทอรีไว้แล้ว นำมารวมไว้ในที่เดียวกันแล้วกำหนด CLASSPATH ให้ชี้ไปยังที่เก็บที่กำหนดให้ การกระจายแฟ้ม .jar ใช้คำสั่ง

```
jar xvf class.jar
```

ในกรณีที่ต้องการตรวจสอบข้อมูลให้แฟ้มก่อนให้ใช้คำสั่ง

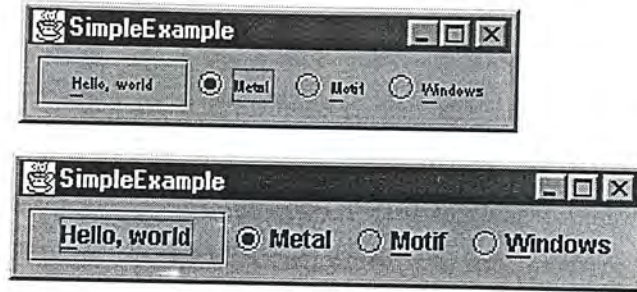
```
jar tvf class.jar
```

8.1.1.3 การปรับแต่งฟอนต์

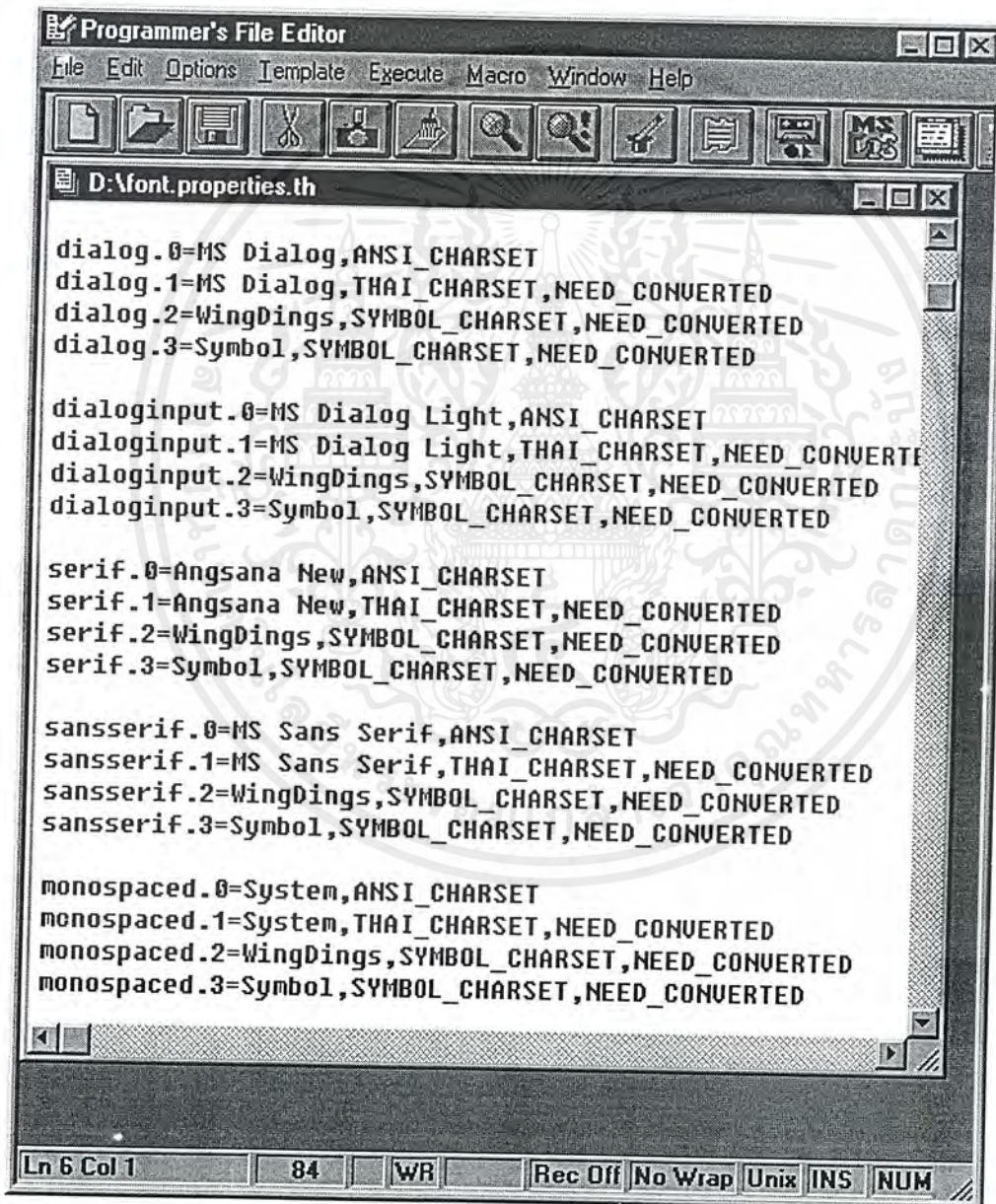
ในการเขียนโปรแกรมด้วยภาษาจาวา ให้ทำงานบนวินโดวส์ หรือใช้ตัวพัฒนาภาษาจาวา ที่ใช้คลาสของจาวา ในการทำงานจะพบว่าตัวหนังสือที่ปรากฏจะมีขนาดเล็กมาก เหตุผลเพราะวินโดวส์ภาษาไทยถูกออกแบบมาให้รองรับภาษาไทยซึ่งมีหลายระดับในหนึ่งบรรทัด ดังนั้นฟอนต์ที่ใช้จึงต้องปรับขนาดลงเพื่อให้บรรทัดอยู่ในหนึ่งบรรทัดได้และจำเป็นต้องปรับขนาดทั้งภาษาไทยและภาษาอังกฤษเพื่อให้ตัวอักษรสมมูลกันและสามารถแสดงร่วมกันในบรรทัดเดียวได้ จึงทำให้ขนาดเมื่อเทียบกับฟอนต์มาตรฐานมีขนาดต่างกันมาก จึงทำให้โปรแกรมภาษา JAVA ที่ใช้ฟอนต์ของระบบตัวอักษรจะเล็กมาก

การแก้ไขทำได้โดยการปรับเปลี่ยน ไปใช้ฟอนต์ที่ปรับขนาดแล้ว โดยเข้าไปแก้ไขแฟ้ม `font.properties.th` แก้ไขเฉพาะฟอนต์หลักที่ใช้ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-3 การทำงานของโปรแกรมก่อนและหลังปรับแต่งฟอนต์



รูปที่ 8-4 ส่วนที่ต้องแก้ไขใน font.properties.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

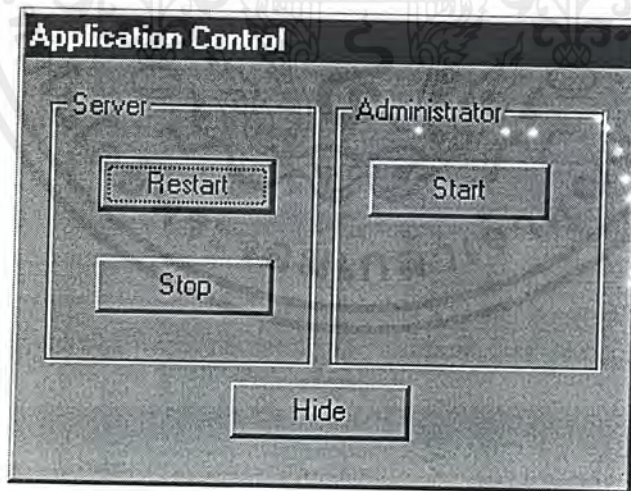
การแก้ไขนี้ สามารถแก้ไขระบบที่ใช้กับจาวาทั้งหมด โดยการนำแฟ้มที่แก้ไขแล้ว เข้าไปแทนที่ทุกๆ ที่ปรากฏแฟ้มนี้อยู่ การแก้ไขนี้ไม่จำกัดรุ่นของ JDK สามารถใช้แทนกันได้ทั้งหมด

8.1.2 การติดตั้งเซิร์ฟเล็ต

เป็นการติดตั้งระบบให้สามารถใช้งานและพัฒนาเซิร์ฟเล็ตได้ การติดตั้งมีเงื่อนไขดังนี้

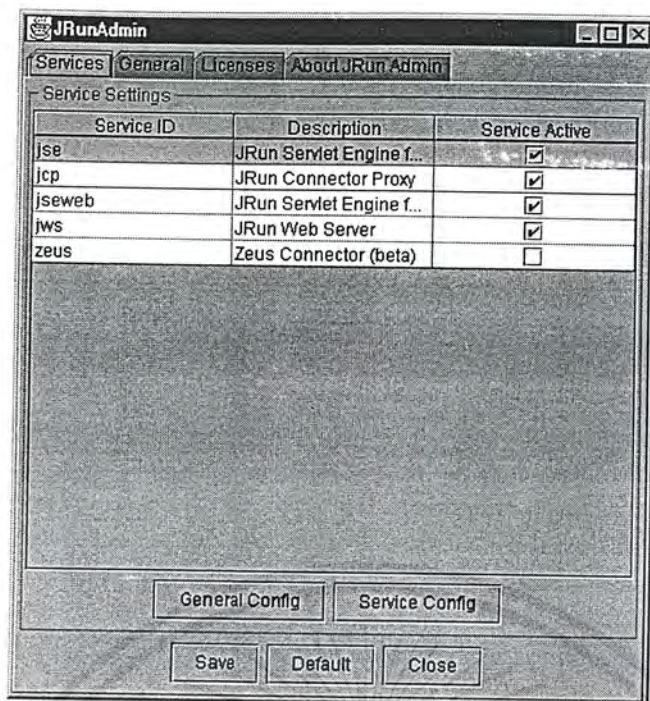
- ควรติดตั้งเว็บเซิร์ฟเวอร์ก่อนการติดตั้ง Servlet Engine ใดๆ และควรใช้เว็บเซิร์ฟเวอร์ที่สามารถติดตั้งโมดูลเพิ่มได้ เพื่อเพิ่มความสามารถของเว็บเซิร์ฟเวอร์ให้ทำงานร่วมกับ Servlet Engine ที่ใช้ได้
- ควรติดตั้ง JDK ก่อนและหลังจากติดตั้ง Servlet Engine แล้วให้ทำการแก้ไข CLASSPATH เพื่อให้รู้จักเซิร์ฟเล็ต คลาส

การที่เซิร์ฟเล็ตจะทำงานได้ระบบจะต้องรู้จักเซิร์ฟเล็ตคลาสเสียก่อน ทั้งในระบบที่ใช้งาน และระบบที่ใช้ในการพัฒนา โดยการตั้ง CLASSPATH ให้ชี้ไปที่ servlet.class ซึ่งจะมียู่ใน JAVA Servlet Development Kit (JSDK) หรือ Servlet Engine ทั่วไป ซึ่งในโครงการใช้ IIS เว็บเซิร์ฟเวอร์ของ Microsoft ร่วมกับ JRUN ของ Live software JRUN เป็น Servlets Engine แบบ Add-on จำเป็นต้องทำงานร่วมกับเว็บเซิร์ฟเวอร์ เมื่อติดตั้ง JRUN ให้ระบุชนิดของเว็บเซิร์ฟเวอร์ที่ติดตั้งอยู่ให้ถูกต้องด้วย เมื่อทำการติดตั้งก็เข้าสู่การปรับแต่ง



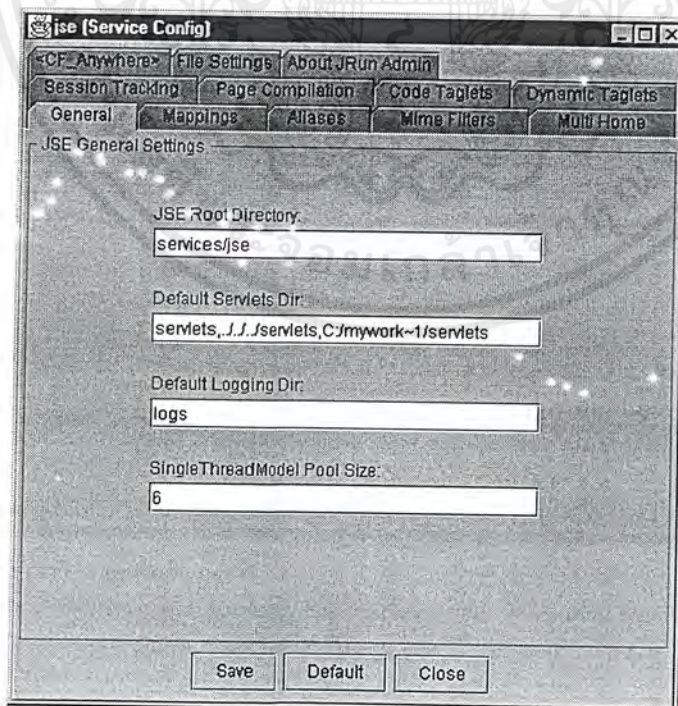
รูปที่ 8-5 หน้าจอเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-6 หน้าจอ Administrator

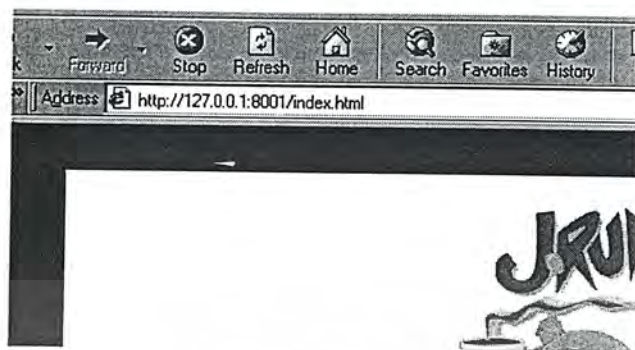
จากรูปที่ 8-2 จะเห็นหน้าจอควบคุม ในครั้งแรกจะต้องทำการปรับแต่งก่อนโดยเข้าสู่หน้าจอ Administrator ส่วนสำคัญส่วนที่จำเป็นคือ Java Servlets Engine (jse) ให้เข้าไปในส่วนของ Service Config หัวข้อ General เป็นการเพิ่ม Directory ที่เป็นที่ยอมรับที่เก็บเซิร์ฟเล็ตที่ใช้งานอยู่



รูปที่ 8-7 หน้าจอการเพิ่ม Servlets Directory

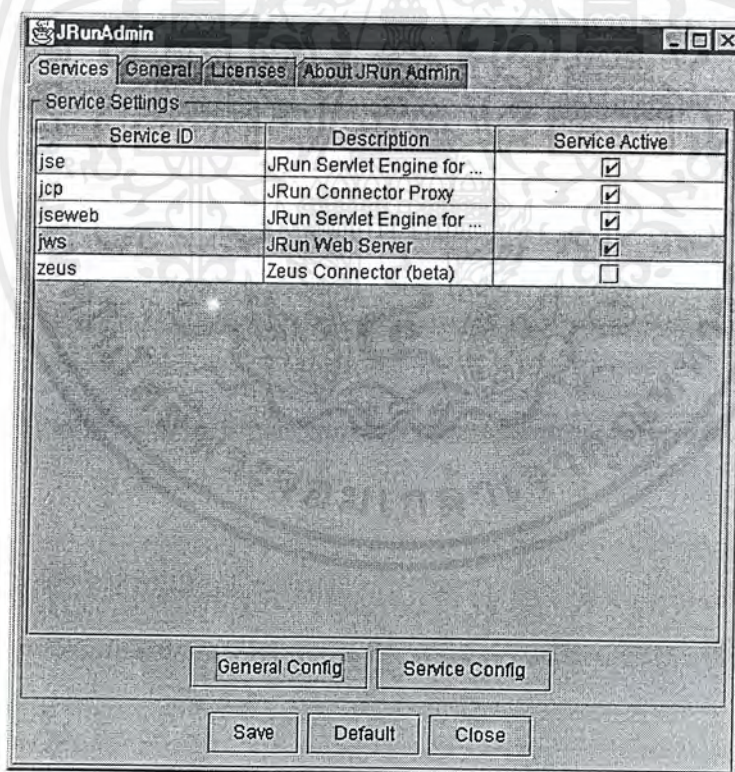
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัว Servlets Engine จำเป็นต้องมี พอร์ต ที่ใช้งานเรียกเซิร์ฟเล็ตในกรณีปกติ JRUN จะกำหนดให้ที่ 8000 แต่ในบางครั้งพอร์ตอาจถูกใช้โดย โปรแกรมอื่น แต่เราสามารถ กำหนดพอร์ตได้ตามความสะดวก



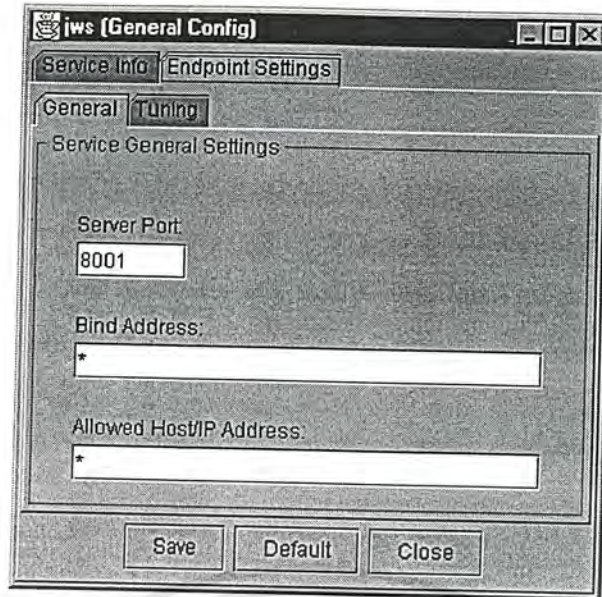
รูปที่ 8-8 หมายเลขพอร์ตที่ใช้

การกำหนดหมายเลขพอร์ตทำได้โดยเข้าไปในส่วน JRUN Web Server เข้าส่วน General Config



รูปที่ 8-9 การปรับแต่ง Jrun Web Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-10 การปรับแต่งค่าพอร์ต

8.2 ส่วนของเอสเอ็นเอ็มพีเอเจนต์ (SNMP Agent)

ในส่วนของเอสเอ็นเอ็มพีเอเจนต์ นั้นใช้จาวาในโครงการนี้เช่นกัน การติดตั้ง JDK ก็กระทำแบบเดียวกัน ส่วนการติดตั้งในส่วนของเอสเอ็นเอ็มพีเอเจนต์ มีวิธีการติดตั้งดังนี้

1. ตรวจสอบว่ามีเอสเอ็นเอ็มพีเอเจนต์ หรือ โปรแกรมอื่นๆ ที่มีการใช้พอร์ตหมายเลข 161 หากไม่ปิดโปรแกรมดังกล่าวแล้วจะไม่สามารถสั่งให้เอสเอ็นเอ็มพีเอเจนต์ ตัวนี้ทำงานได้
2. ทำการติดตั้งส่วนของคลาสที่ใช้ โดยการประกาศให้เป็นส่วนหนึ่งของ CLASSPATH เช่น

```
C:\>SET CLASSPATH=%CLASSPATH%;imgmt.jar
```

3. สามารถสั่งให้เอสเอ็นเอ็มพีเอเจนต์ ทำงานได้ในการสั่งงานแบบจาวาแอปพลิเคชัน โดยหน้าต่างที่เปิดทิ้งไว้จะเป็นตัวแสดงผลด้วย

```
C:\>java Agent
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 ความต้องการของระบบ (System Requirement)

ความต้องการของระบบที่ใช้ JDK และ Jrun Servlet Engine

- หน่วยประมวลผล Pentium 133 (ขั้นต่ำ), แนะนำให้ใช้ Pentium II 233 หรือมากกว่า
- หน่วยความจำ 32 เมกะไบต์ (ขั้นต่ำ), แนะนำให้ใช้ 128 เมกะไบต์
- ความจุฮาร์ดดิส 40 เมกะไบต์ (ขั้นต่ำ), แนะนำให้ใช้ 100 เมกะไบต์
- ระบบปฏิบัติการ Windows 95/98/NT ,Linux
- ไดรเวอร์
 - Winsock 1.1 หรือ 1.2 ไม่สามารถใช้ได้กับ Winsock ใน Windows 3.11
 - TCP/IP



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

HP Openview

HP Open View

สำหรับเครื่องมือหรือว่าโปรแกรมที่ใช้ในการตรวจสอบเครือข่ายในปัจจุบันนี้ก็มียู่อหลากหลาย ในที่นี้เราจะใช้โปรแกรม HP Open View ซึ่งมีความสามารถในการตรวจสอบการเชื่อมต่อเครือข่ายในลักษณะตรรกะ (Logical) มีการตรวจสอบสถานะของอุปกรณ์ พร้อมทั้งแจ้งเตือนได้หากว่าอุปกรณ์ใดๆ ทำงานบกพร่อง และยังสามารถเข้าไปจัดการแก้ไขอุปกรณ์ที่เกิดปัญหาได้ด้วย โดยผ่านทาง SNMP Service และ DMI ที่จะต้องติดตั้งไว้ที่ตัวอุปกรณ์ดังกล่าว

DMI ย่อมาจาก Desktop Management Information เป็นมาตรฐานที่ถูกจัดเตรียมโดย Desktop Management Task Force (DMTF) ซึ่งก่อตั้งขึ้นในปี 1992 โดยมีวัตถุประสงค์เพื่อกำหนดมาตรฐานแบบเปิด ให้มีข้อกำหนดเกี่ยวกับการจัดการคอมพิวเตอร์ (Desktop) ทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ เพื่อให้สามารถตรวจสอบ-เข้าถึงข้อมูลเกี่ยวกับฮาร์ดแวร์/ซอฟต์แวร์ของคอมพิวเตอร์ โดยใช้ Remote Control หรือการเข้าถึงระยะไกลได้ หรือโดยที่ไม่ต้องเปิดเคสของคอมพิวเตอร์ วิศวกรหรือช่างผู้ตรวจสอบก็สามารถทราบได้ว่าคอมพิวเตอร์เครื่องนั้นมีส่วนประกอบฮาร์ดแวร์ต่างๆ อย่างไร หรือผู้ผลิตคอมพิวเตอร์เองก็สามารถเพิ่มข้อมูลเกี่ยวกับรุ่นการผลิตคอมพิวเตอร์ วันที่ผลิต ส่วนผู้ที่ซื้อคอมพิวเตอร์มาใช้ก็สามารถเพิ่มรายละเอียดเกี่ยวกับสถานที่ติดตั้งและการติดตั้งได้ จะเห็นได้ว่า DMI นั้นมีประโยชน์ในการจัดการต่อทุกๆ ฝ่ายที่เกี่ยวข้อง ในการจัดการเกี่ยวกับคอมพิวเตอร์ ประโยชน์ของ DMI คือ ลดค่าใช้จ่ายในการจัดการและการติดตั้งคอมพิวเตอร์

การใช้งานโปรแกรม Hp Open View

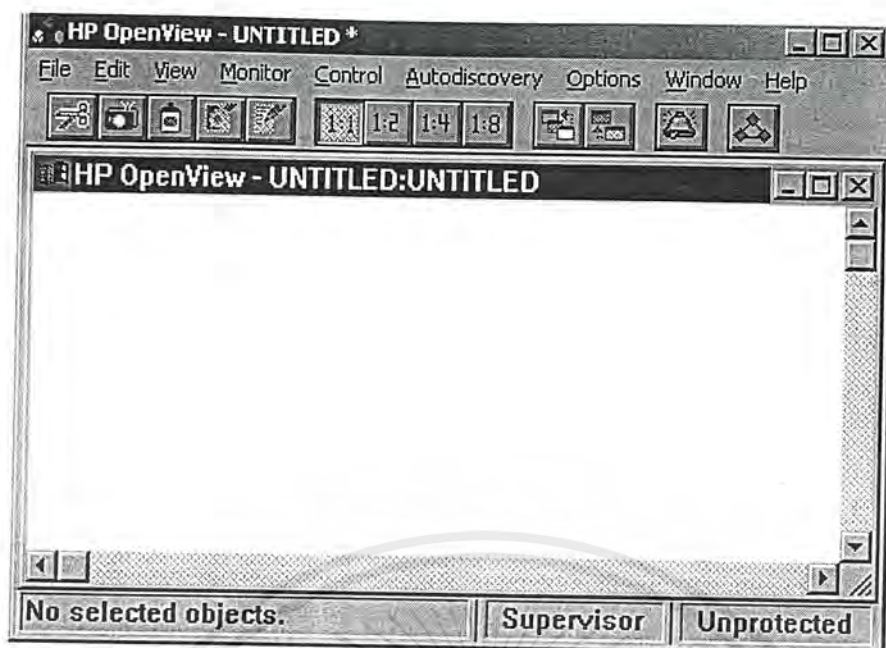
เมื่อเราทำการติดตั้งโปรแกรมเสร็จเรียบร้อยแล้ว ก็ให้ทำการเปิดโปรแกรม HP Open View ขึ้นมา โดยจะมีหน้าต่างของโปรแกรมดังรูปที่ 1

สำหรับเมนูการทำงานต่างๆ ของโปรแกรม HP Open View จะมีอยู่ดังนี้

1. File Menu

- New คือ สร้างเพิ่มข้อมูลและ Map ขึ้นมาใหม่
- Open คือ เปิดเพิ่มข้อมูลเก่าที่ได้มีการบันทึกไว้แล้ว
- Save คือ บันทึกข้อมูลของ Map และเพิ่มการทำงานที่ทำงานอยู่ขณะนั้น
- Save As คือ บันทึกข้อมูลของ Map และเพิ่มการทำงานที่ทำงานอยู่ขณะนั้นเป็นเพิ่มอื่น
- Check Map คือ ตรวจสอบ Map และอุปกรณ์ต่างๆ ที่มีอยู่ใน Map ให้ถูกต้อง
- Print Submap คือ พิมพ์ Submap ที่ได้ทำการเลือกเอาไว้จากทางเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 แสดงหน้าต่างของโปรแกรม HP Open View

- Print Object List คือ การพิมพ์รายการของ Object ที่มีอยู่ใน Map ทั้งหมดออกเครื่องพิมพ์
- Print Setup คือ กำหนดและเปลี่ยนแปลงค่าติดตั้งของเครื่องพิมพ์
- Load Selection List คือ โหลดเอารายการของอุปกรณ์ที่ได้ทำการบันทึกไว้ขึ้นมา
- Save Selection List คือ บันทึกค่ารายการของอุปกรณ์ที่ได้ทำการเลือกไว้ใน Map
- Exit คือ ออกจากโปรแกรม HP Open View

2. Edit Menu

- Add คือ ทำการเพิ่มสัญลักษณ์และอุปกรณ์อื่นๆ เพิ่มเติมลงใน Map
- Cut คือ ตัดสัญลักษณ์ที่ได้ทำการเลือกออกจาก Map ไปยัง Clipboard
- Copy คือ คัดลอกสัญลักษณ์ที่ได้ทำการเลือกออกจาก Map ไปยัง Clipboard
- Paste คือ วางสัญลักษณ์ที่มีอยู่ใน Clipboard ลงใน Map และตำแหน่งที่เลือกไว้
- Delete คือ ลบสัญลักษณ์ที่ได้ทำการเลือกออกจาก Map
- Describe คือ แสดงรายละเอียดของสัญลักษณ์อุปกรณ์ที่ได้ทำการเลือกไว้ออกมา
- New Submap คือ สร้าง Submap ใหม่ขึ้นมาภายใต้ Map ที่กำลังทำงานอยู่
- Rename Submap คือ เปลี่ยนชื่อของ Submap ที่ได้ทำการเลือกไว้ขณะนั้น
- Clear Submap คือ การเคลียร์สัญลักษณ์อุปกรณ์ที่มีทั้งหมดใน Submap ที่เลือกไว้ขณะนั้น
- Delete Submap คือ ลบ Submap ที่ได้ทำการเลือกไว้ขณะนั้นออกไปจาก Map
- Set Background Image คือ การกำหนดรูปภาพพื้นหลังของ Map ที่ได้ทำการเลือกไว้ขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Clear Background Image คือ ลบภาพที่เป็นพื้นหลังของ Map ที่ได้ทำการเลือกไว้ขณะนั้น
- Set Home Submap คือ กำหนดค่า Map เริ่มต้นหรือ Home ของ Submap ที่เลือกไว้

3. View Menu

- Zoom 1 คือ กำหนดอัตราการแสดงผลอยู่ที่ 1 เท่า
- Zoom ½ คือ กำหนดอัตราการแสดงผลอยู่ที่ ½ เท่า
- Zoom ¼ คือ กำหนดอัตราการแสดงผลอยู่ที่ ¼ เท่า
- Zoom 1/8 คือ กำหนดอัตราการแสดงผลอยู่ที่ 1/8 เท่า
- Pan คือ ย่อมุมมองโฟกัสของภาพรวมทั้งหมดใน Map
- Set Grid คือ กำหนดตำแหน่งระยะห่างระหว่างเส้น Grid ทั้งแนวนอนและแนวตั้ง
- Show Grid/Hide Grid คือ กำหนดการแสดงผลว่าใช้แสดงหรือซ่อนเส้น Grid

4. Monitor Menu

- Describe คือ แสดงรายละเอียดของสัญลักษณ์อุปกรณ์ที่ได้ทำการเลือกไว้ออกมา
- Status Legend คือ สีที่ใช้ในการบ่งบอกสถานะของสัญลักษณ์อุปกรณ์ต่างๆ ที่มีอยู่
- Alarm Log คือ บันทึกการแจ้งเตือนจากสัญลักษณ์อุปกรณ์ต่างๆ ที่มีอยู่ใน Map
- Ping คือ การทดสอบการตอบสนองกลับของอุปกรณ์ ว่ายังทำงานอยู่หรือไม่
- Remote Power On คือ สั่งให้เปิดอุปกรณ์พีซีจากระยะไกล จะมีเมนูย่อยคือ
 - Wake Up PC คือ สั่งให้เครื่องพีซีที่ถูกเลือกนั้นทำการเปิดเครื่องตัวเอง
 - Schedule Wake Up คือ กำหนดตารางวันและเวลาในการเปิดเครื่องพีซีที่เลือก
- HP Top Tools คือ โปรแกรมที่ให้ทำงานบนอุปกรณ์ที่ต้องการ เพื่อการจัดการดูแลเกี่ยวกับ DMI ภายในอุปกรณ์นั้นๆ และตรวจดูค่า Configure ภายในด้วย
- Polling คือ ระบบอัตโนมัติที่จะทำงานเมื่อเริ่มต้นโปรแกรม HP Open View โดยจะเริ่มทำการ Discovery ระบบเครือข่ายแล้วทำการสร้าง Map ของอุปกรณ์ต่างๆ อุปกรณ์ที่ถูกสร้างไว้ใน Map จะถูกเก็บไว้ใน Polling List

5. Control Menu

- SNMP Manager คือ การจัดการเกี่ยวกับด้าน SNMP ของอุปกรณ์ มีเมนูย่อยคือ
 - Define Query คือ การกำหนดเลือกค่า MIB ต่างๆ ของอุปกรณ์ที่ถูกเลือกในขณะนั้น
 - Select Query คือ การเลือกค่า MIB ของอุปกรณ์ต่างๆ โดยจะมีรายการของอุปกรณ์ให้เลือก
 - Manage Database คือ จัดการรายการเพิ่มเติมของ MIB ที่ต้องการให้ทำการ Define เพิ่มเติม
- DMI Manager
 - Define Query คือ การกำหนดเลือกค่า DMI ต่างๆ ของอุปกรณ์ที่ถูกเลือกในขณะนั้น
 - Select Query คือ การเลือกค่า DMI ของอุปกรณ์ต่างๆ โดยจะมีรายการของอุปกรณ์ให้เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Brows Web Home Page คือ เข้าไปดู Home Page ของอุปกรณ์ดังกล่าวในกรณีที่เครื่องนั้นเปิดบริการ WWW ไว้
- Brows Web Management Page คือ เข้าไปแก้ไขจัดการเกี่ยวกับ Home Page หรือเป็น Home Page ของผู้ให้บริการสำหรับอุปกรณ์นั้นๆ

6. Autodiscovery Menu จะประกอบด้วยเมนูย่อย 3 เมนูคือ

- Configure Menu
 - Discovery Schedule คือ ตารางการกำหนดระยะเวลาในการทำ IP Discovery, Topology Discovery และ Extended (IPX, Other) Discovery
 - Discovery Networks คือ การกำหนดค่าต่างๆของระบบเครือข่ายของตัวเองคือ IP Address, Subnet Mask และ Gateway รวมไปถึงส่วนที่ต้องการให้ทำการ Discovery คือ IP และ IPX
 - User-Specified Devices คือ เพิ่มเติมอุปกรณ์ที่ไม่ได้อยู่ในรายการที่มีอยู่การทำ Autodiscovery
 - Ignored Devices คือ กำหนดรายการอุปกรณ์ที่ไม่ต้องการให้ทำ Autodiscovery
 - Device Classes คือ รายการของ Classes อุปกรณ์ต่างๆที่มีอยู่ในโปรแกรม HP Open View
- Discovery Menu
 - Discovery Manager คือ ศูนย์กลางหลักในการจัดการเกี่ยวกับการ Discovery ซึ่งจะกำหนดการเพิ่มอุปกรณ์ที่ต้องการ Discovery ลงใน Database สั่งการ Discovery ให้เริ่มและหยุดและสั่งให้ Discovery ในส่วนของ IP และ IPX
 - Discovery Routers คือ ฟังก์ชันในการกำหนดและแก้ไขค่าสำหรับ SNMP Router
 - Update Host Names คือ การปรับปรุงชื่อ Host Name โดยจะไปอ่านค่ามาจาก Domain Name Server เพื่อทำการปรับปรุง
 - Clear Discovery Database คือ การเคลียร์ค่า Database ที่มีทั้งหมดที่ได้มาจากการ Discovery
 - Dump Discovery Database คือ คือการบันทึกค่า Database ที่ได้มาจากการ Discovery ลงไฟล์ โดยจะเก็บไว้ในไฟล์ที่ชื่อว่า OVDUMPIT
 - Discovery Topology คือ เรียกการทำ Discovery อัตโนมัติเพื่อทำการตรวจหาการเชื่อมต่อของอุปกรณ์ในเครือข่าย
- Layout Menu
 - Do Basic Layout คือ การสั่งให้สร้าง Map ที่ได้มาจากการ Discovery แล้วออกมา
 - Basic Layout Options คือ การกำหนดระยะห่าง ขนาด และความกว้างของ Map ที่จะสร้างหรือกำหนดให้เป็น Auto Arrange ก็ได้
 - Configure Preferred Name คือ กำหนดชื่อให้กับ Network Address ของอุปกรณ์นั้นๆ

7. Option Menu

- Protect Map คือ การกำหนดรหัสผ่านเพื่อที่จะทำการเข้าไปแก้ไขหรือเปลี่ยนแปลง Map

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Log In คือ การ login เข้าไปใช้งานโปรแกรม HP Open View โดยมีระดับของผู้ใช้ต่างๆ 3 ระดับ คือ ผู้สังเกตการณ์ ผู้ปฏิบัติการ และผู้ดูแลระบบ
- Log Out คือ การ Logout ออกจากโปรแกรม HP Open View
- Set Password คือ การตั้งรหัสผ่านสำหรับผู้ใช้ในแต่ละระดับเพื่อเข้าใช้โปรแกรม HP Open View
- Customize HP Open View คือ การกำหนดค่าต่างๆ ในการแสดงผลของโปรแกรม HP Open View
- Customize Alarm คือ การกำหนดค่าต่างๆ ในการแจ้งเตือนเหตุของอุปกรณ์ต่างๆ ใน Map
- Customize Device Access คือ การกำหนดค่าในการเข้า Access ยังอุปกรณ์ที่อยู่ใน Map
- Customize Traps คือ กำหนดค่า SNMP Message พิเศษที่ต้องการให้ส่งไปยังระบบเครือข่าย

8. Windows Menu

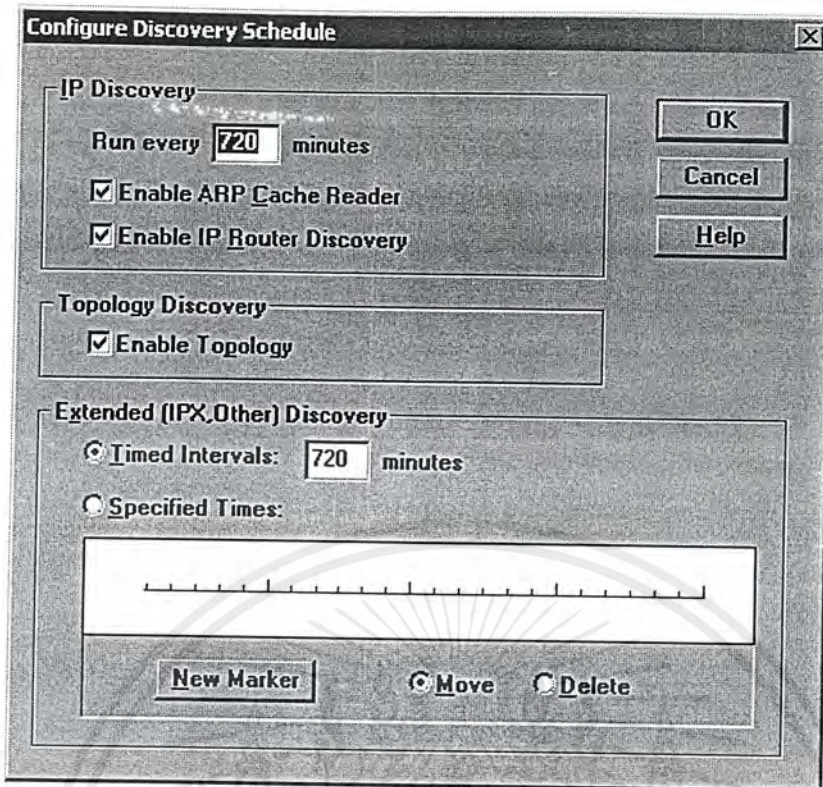
- Home Submap คือ ไปยังหน้าต่างของ Home Map ที่ได้กำหนดไว้
- Previous Submap คือ ไปยังหน้าต่างก่อนหน้า หน้าต่างที่กำลังใช้งานอยู่
- Locate Submap คือ ไปยังหน้าต่างของ Submap ที่แสดงไว้ในรายการที่มีอยู่
- Locate Object คือ ไปยังหน้าต่าง Submap ที่มีรายการของสัญลักษณ์อุปกรณ์นั้นแสดงอยู่
- Extended Locate คือ การดูรายละเอียดภายใน Map หรือว่า Component ดังกล่าวว่ามี Submap อะไรบ้าง มีอุปกรณ์และสัญลักษณ์อะไรบ้าง ซึ่งจะมีลักษณะฟังก์ชันการทำงานเหมือนกับ Locate Object แต่ว่า Extended Locate จะมีส่วนที่เพิ่มเติมมากกว่า

ขั้นตอนการใช้งานโปรแกรม HP Open View

สำหรับการใช้งานโปรแกรม HP Open View ในเบื้องต้น เราจะใช้วิเคราะห์ระบบเครือข่ายและเข้าไปดูรายละเอียด ตรวจสอบค่าต่างๆ ในอุปกรณ์แต่ละตัว ซึ่งจะมีขั้นตอนการใช้งานดังนี้

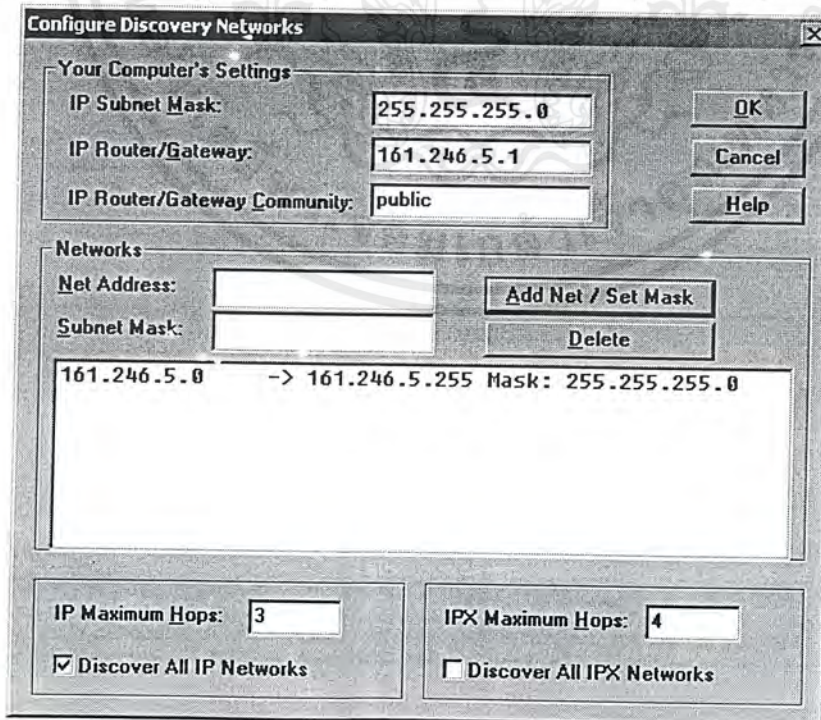
1. เมื่อทำการเปิดโปรแกรมขึ้นมาแล้วดังรูปที่ 1 ให้เข้าไปแก้ไขค่าในเมนู AutoDiscovery เพื่อที่จะทำการ Discovery ในส่วนของ IP ที่เราจะใช้เท่านั้น
2. ให้ไปที่เมนู Discovery Schedule เพื่อที่จะทำการกำหนดค่าเวลาในการทำ Discovery แต่ละครั้งให้เหมาะสมว่าต้องการตรวจสอบบ่อยแค่ไหน ตามแต่ความต้องการของระบบ ดังรูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 แสดงหน้าต่างของ Discovery Schedule

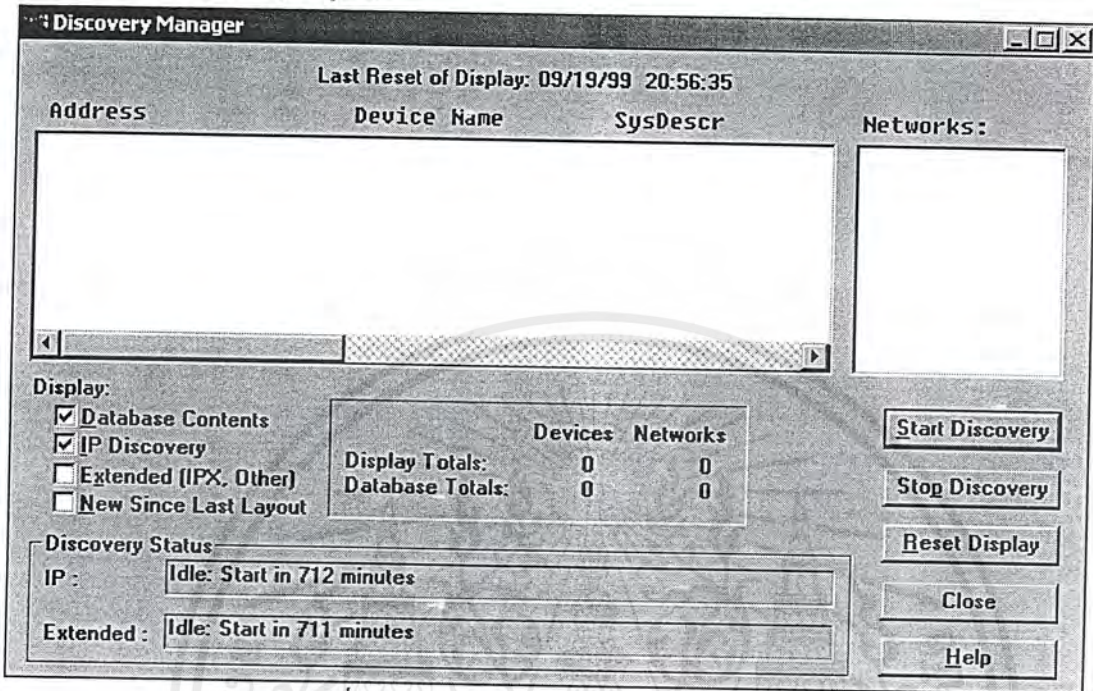
3. ให้ไปที่เมนู Discovery Networks เพื่อที่จะทำการกำหนดค่าของเครื่องที่ใช้โปรแกรม คือ IP Subnet mask, IP Router/Gateway และเลือกพร้อมทั้งกำหนดค่าจำนวนของ Hop ทั้ง IP และ IPX ที่ต้องการ จะให้ทำการตรวจสอบ ดังรูปที่ 3



รูปที่ 3 แสดงหน้าต่าง Discovery Networks

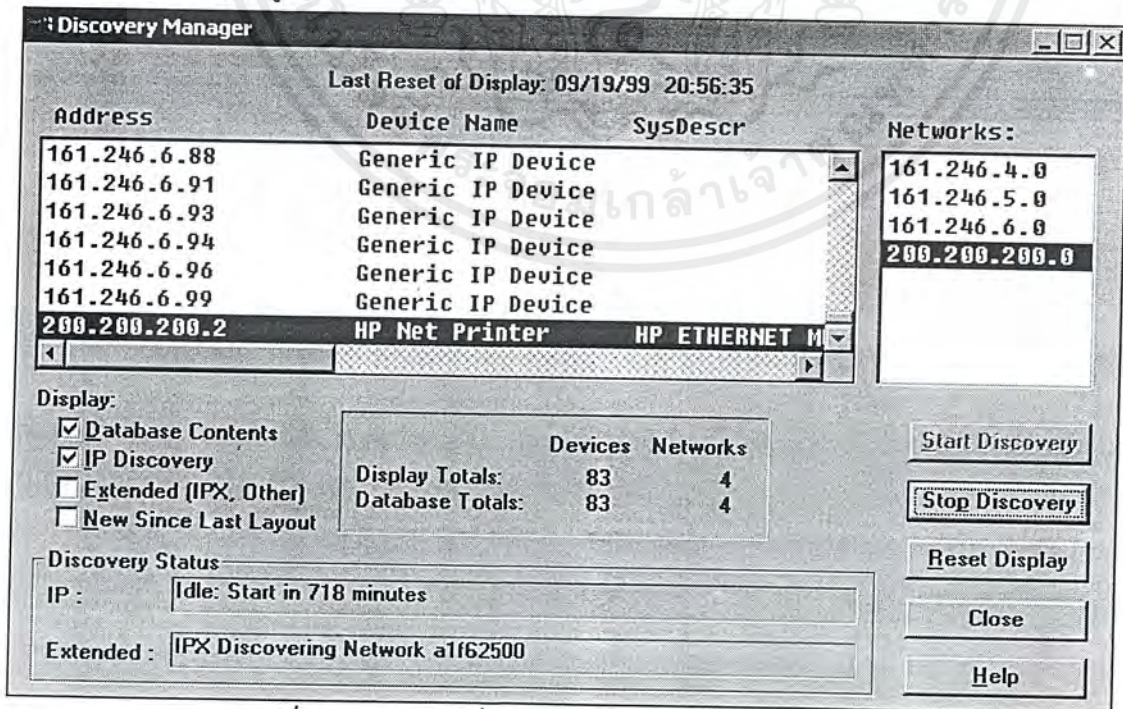
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ไปที่เมนู Discovery Manager เพื่อที่จะทำการเริ่มการทำ Discovery โดยกำหนดให้ทำการ Discovery ในส่วนของ IP และ Database Contents (หากว่าต้องการ Discovery ในส่วนของ IPX ก็เลือกได้) แล้วจึงเริ่ม Start Discovery ดังรูปที่ 4



รูปที่ 4 แสดงหน้าต่างของ Discovery Manager

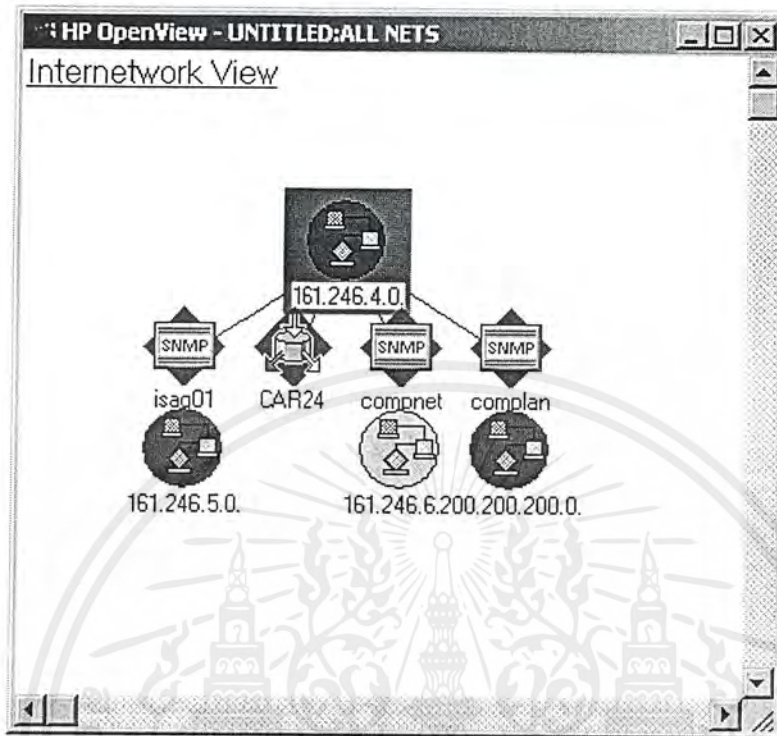
5. จากนั้นรอให้สถานะของโปรโตคอลที่เราเลือกไว้เป็นสถานะ Idle เสียก่อน (ในที่นี้คือ IP ถ้าเป็น IPX ก็คือ Extended) ดังรูปที่ 5



รูปที่ 5 แสดงสถานะเมื่อทำการ Discovery IP เสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ต่อไปให้ไปที่เมนู Do Basic Layout เพื่อให้โปรแกรม HP Open View ทำการสร้าง Map ของอุปกรณ์ ออกมาจะได้ดังรูปที่ 6



รูปที่ 6 แสดง Map ที่ได้มาจากการ Discovery

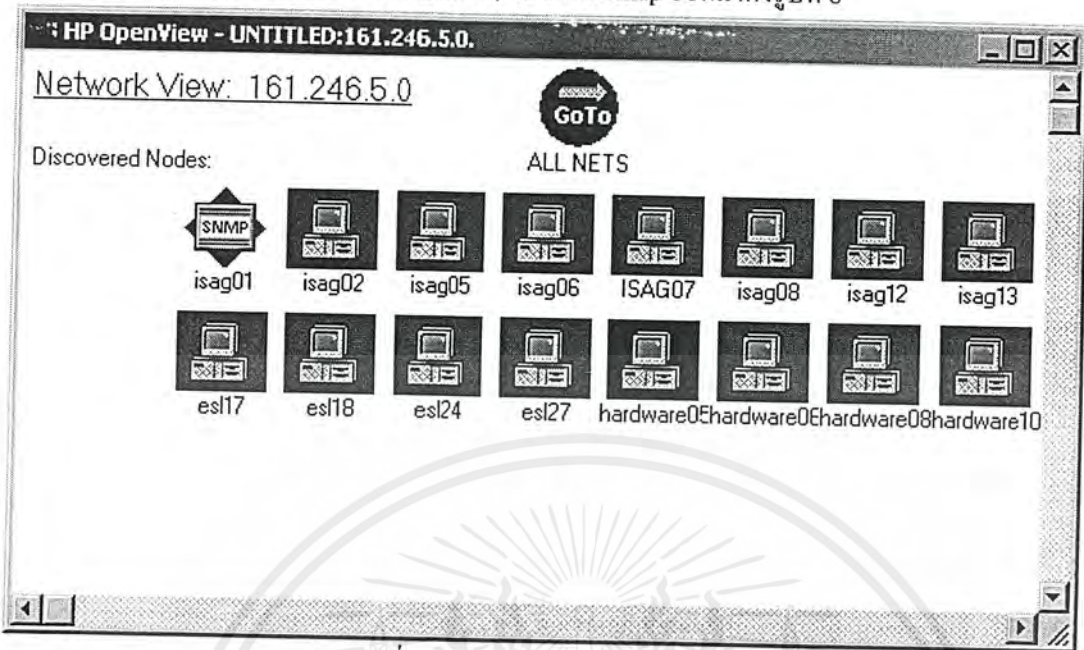
7. จากรูปที่ได้จะเห็นได้ว่าสถานะของแต่ละสัญลักษณ์นั้นมีสีแตกต่างกันไป สำหรับสีแต่ละสีนั้นมีความหมายแตกต่างกันไป เราสามารถเรียกดูสถานะของแต่ละสีได้ที่เมนู Monitor -> Status Legend จะได้ดังรูปที่ 7



รูปที่ 7 แสดงสีและสถานะที่บ่งบอกใน Map

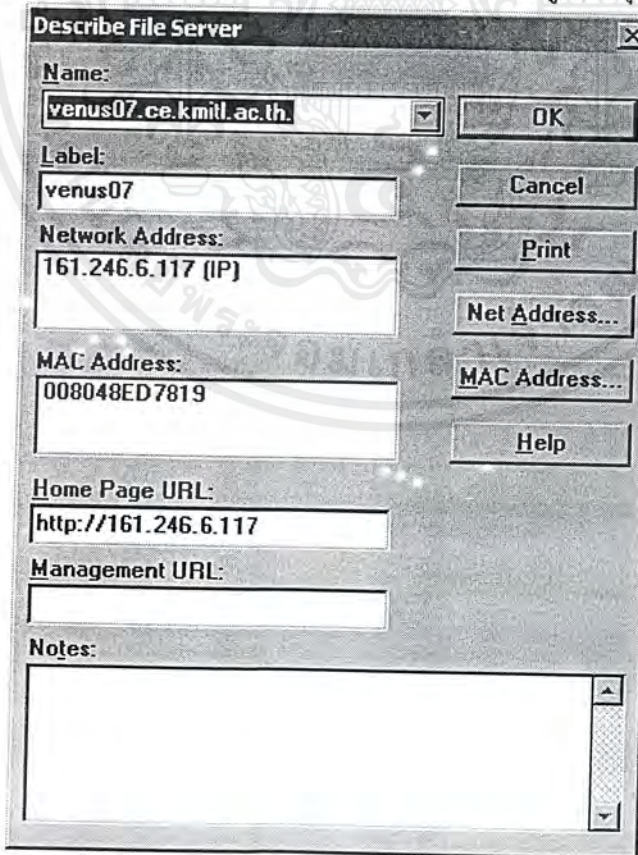
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. จาก Map ที่เห็นนั่นคือ Home Map หากต้องการที่จะเข้าไปดูภายในเครือข่ายย่อย ก็ให้ทำการ Double Click ที่สัญลักษณ์ (ณ ที่นี้ก็คือ 161.246.5.0) จะได้ Submap ออกมาดังรูปที่ 8



รูปที่ 8 แสดงเครือข่าย Submap ที่ได้

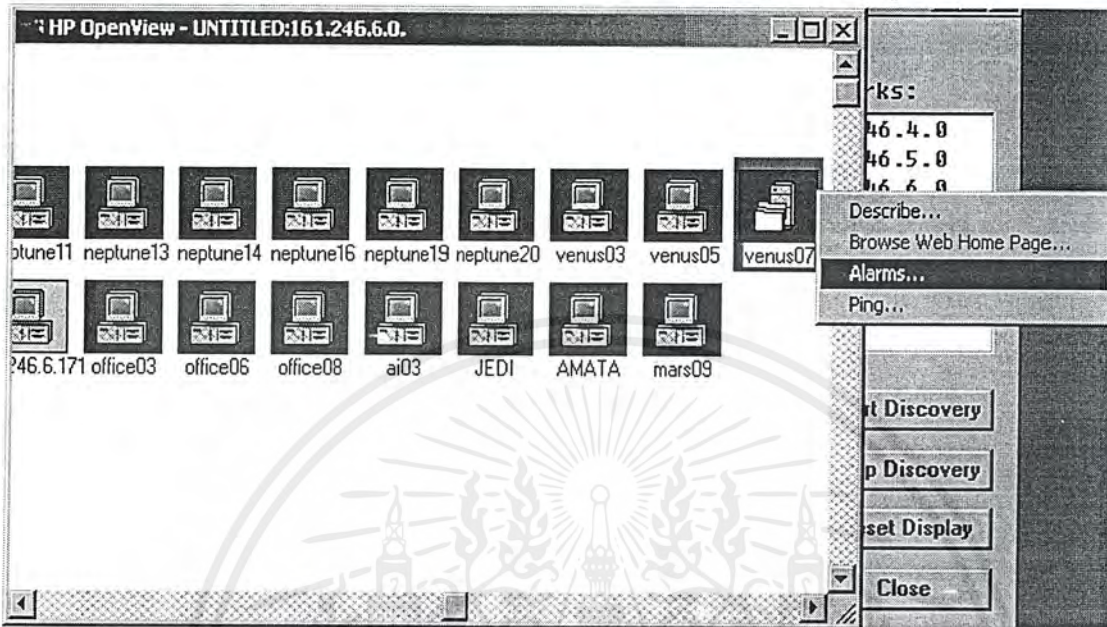
9. ในกรณีที่เรต้องการดูข้อมูลของอุปกรณ์นั้นๆ ว่ามีอะไรอย่างไรบ้างเราสามารถดูได้จากการ Click ขวาแล้วเลือก Describe หรือไปที่เมนู Monitor -> Describe จะ ได้ข้อมูลของอุปกรณ์นั้นดังรูปที่ 9



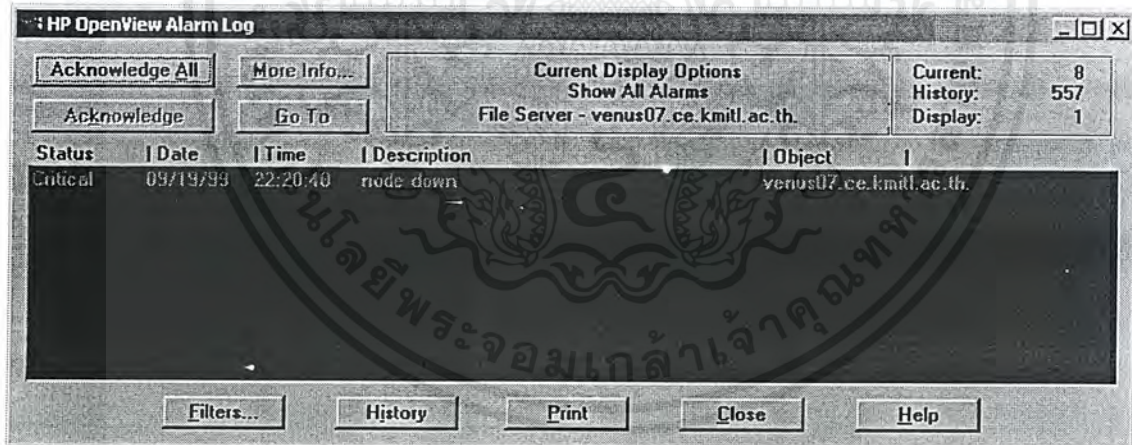
รูปที่ 9 แสดงหน้าต่าง Describe ที่ได้จากอุปกรณ์ที่เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. หากว่าใน Submap พบว่าอุปกรณ์ตัวใดที่ไม่ได้มีสถานะปกติคือเป็นสีเขียว เราสามารถไปดูสถานะการแจ้งเตือนได้โดยการ Click ขวาที่อุปกรณ์ดังกล่าวแล้วไปที่เมนู Alarm หรือไปที่เมนู Monitor -> Alarm Log ก็ได้ ดังรูปที่ 10 และจะได้หน้าต่างแสดงรายละเอียดการแจ้งเตือนดังรูปที่ 11



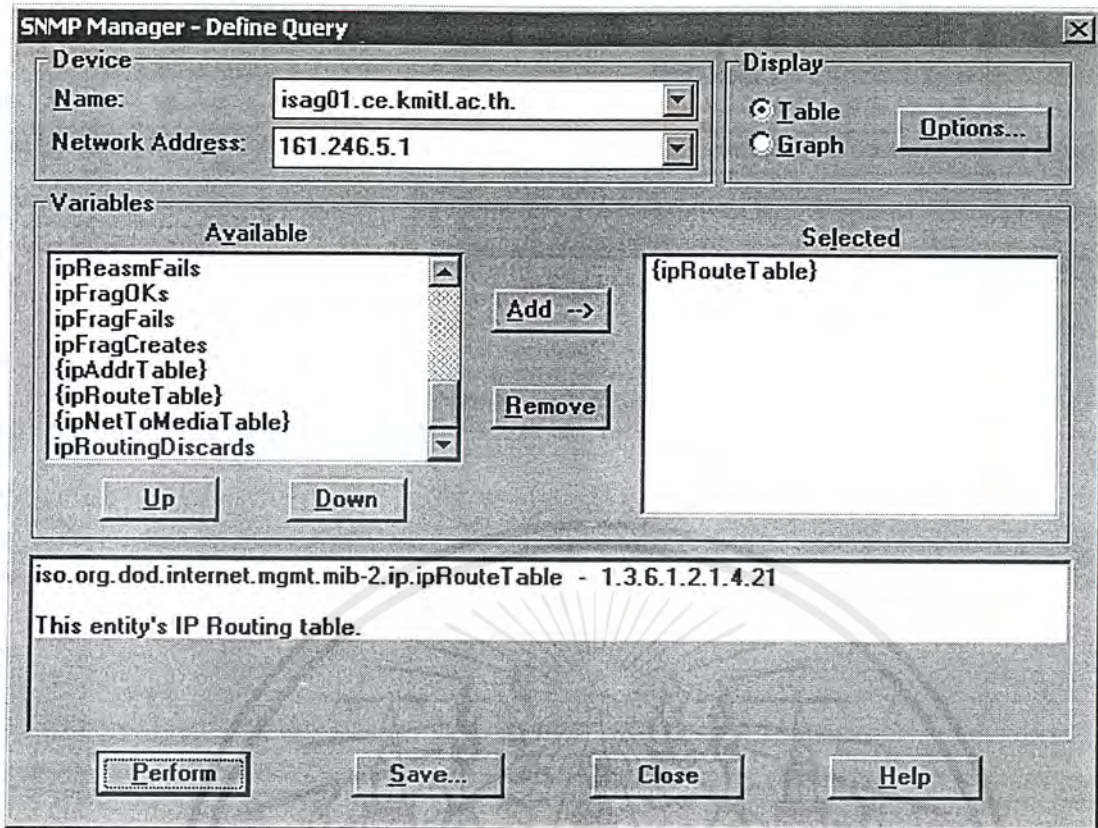
รูปที่ 10 แสดงการเลือกตรวจสอบ Alarm ของอุปกรณ์ที่เลือก



รูปที่ 11 แสดงรายการการแจ้งเตือนของเครื่องที่ได้ทำการเลือกไว้

11. ถ้าหากว่าเรารับรู้กับการแจ้งเตือนที่เกิดขึ้นแล้ว ก็ให้ไปที่รายการที่แสดงเพื่อทำการ Acknowledge หรือว่าจะ Acknowledge All เพื่อรับทราบการแจ้งเตือนทั้งหมดก็ได้
12. สำหรับสัญลักษณ์ที่มีคำว่า SNMP นั้นเราสามารถทำการเข้าไปตรวจดูค่า MIB ของอุปกรณ์นั้นๆ ได้ โดยไปที่เมนู Control -> SNMP Manager -> Define Query จะได้หน้าต่างออกมาดังรูปที่ 12 เพื่อที่จะทำการเลือกรายการที่เราต้องการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 แสดงหน้าต่างสำหรับการกำหนดรายการเพื่อเลือกค่าที่ต้องการ

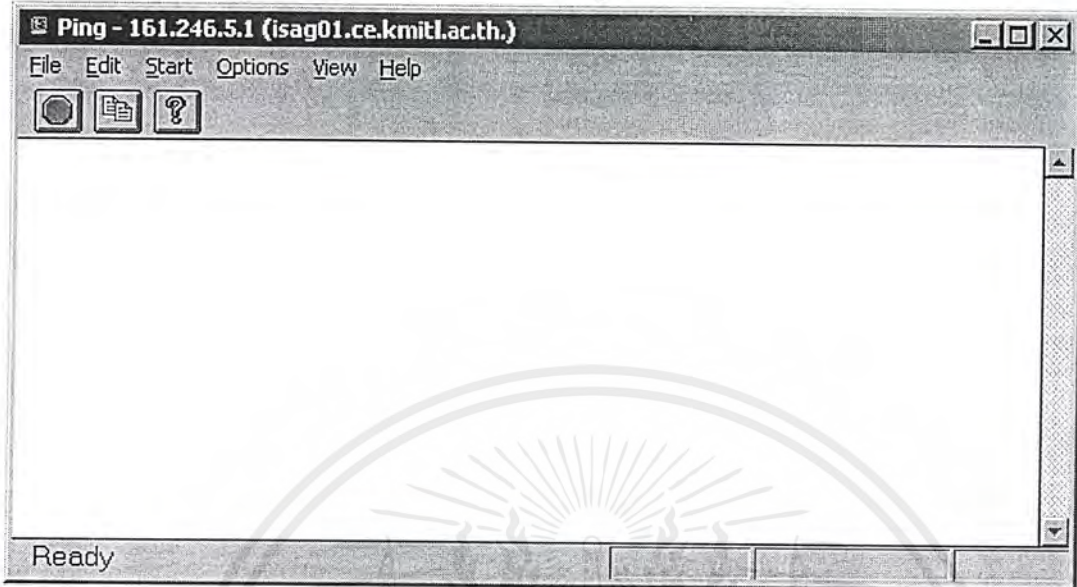
13. เราจะทำการเลือกเพื่อเป็นตัวอย่างในที่นี้คือ {ipRouteTable} ให้เลือกที่รายการดังกล่าวแล้วทำการ Add หากว่าต้องการดูรายการอื่นๆ ก็สามารถเพิ่มเติมเข้าไปได้ แล้วทำการ Perform จะได้ดังรูปที่ 13 ซึ่งแสดงค่าที่เราได้เลือกไว้

	ipRouteNextHop	ipRouteType	ipRouteProto	ipRouteAge
0.0.0.0	161.246.4.5	indirect	local	866515
161.246.1.0	161.246.4.5	indirect	rip	17
161.246.2.0	161.246.4.5	indirect	rip	17
161.246.3.0	161.246.4.5	indirect	rip	17
161.246.4.0	161.246.4.7	direct	local	913795
161.246.5.0	161.246.5.1	direct	local	913795
161.246.6.0	161.246.4.1	indirect	local	866516
161.246.7.0	161.246.4.5	indirect	rip	17
161.246.9.0	161.246.4.5	indirect	rip	17
161.246.10.0	161.246.4.5	indirect	rip	17
161.246.11.0	161.246.4.5	indirect	rip	17

รูปที่ 13 แสดงค่าที่เลือกไว้ในหลังจาก Perform

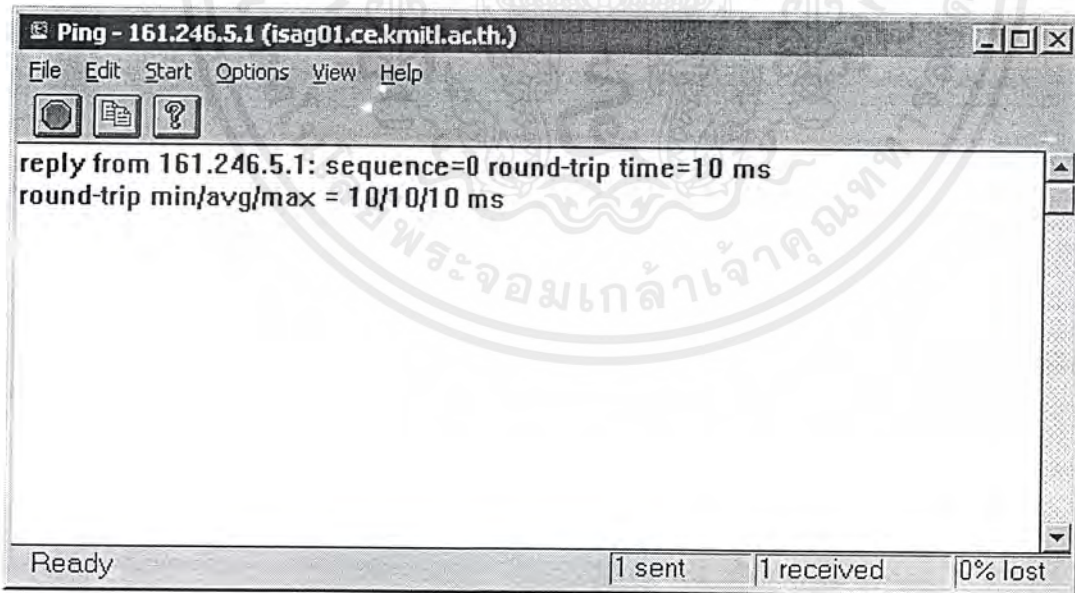
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. เราสามารถทำการ Ping ไปยังเครื่องที่เราต้องการทราบสถานะได้ โดยการ Click ขวาแล้วไปที่เมนู Ping หรือไปที่เมนู Monitor -> Ping จะได้นหน้าต่างดังรูปที่ 14



รูปที่ 14 แสดงหน้าต่างการ Ping ของอุปกรณ์ที่เลือก

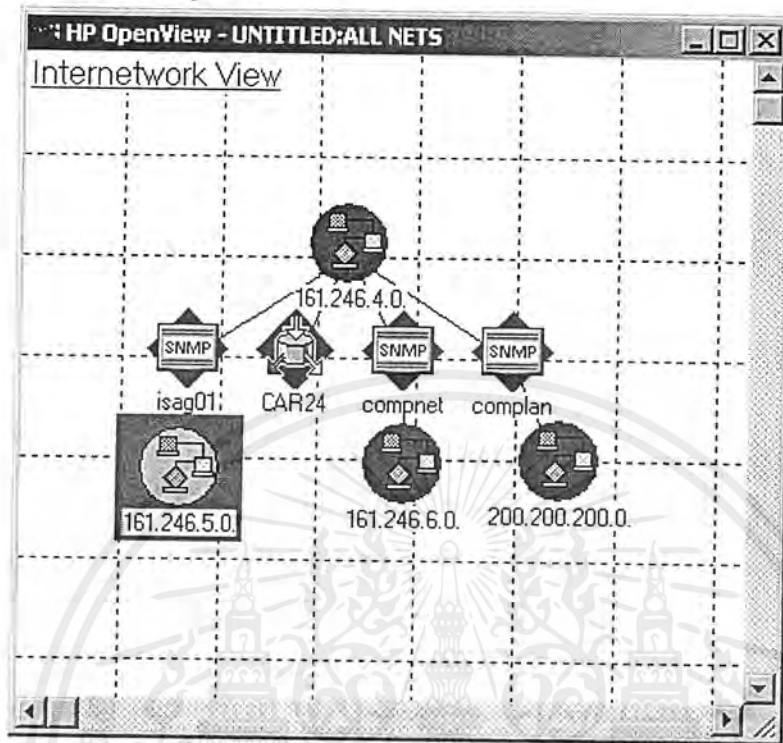
15. เมื่อได้นหน้าต่างดังรูปที่ 14 แล้วให้ทำการ Start การ Ping โดยการกดที่ปุ่มสีเขียวหรือว่า ไปที่เมนู Start จะมีข้อมูลของการ Ping นั้นออกมาว่าส่งหรือว่ารับ packet ที่ส่งได้หรือไม่คิดเป็นกิโลเปอร์เซ็นต์ ดังรูปที่ 15



รูปที่ 15 แสดงสถานะของการ Ping อุปกรณ์ที่ได้เลือกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

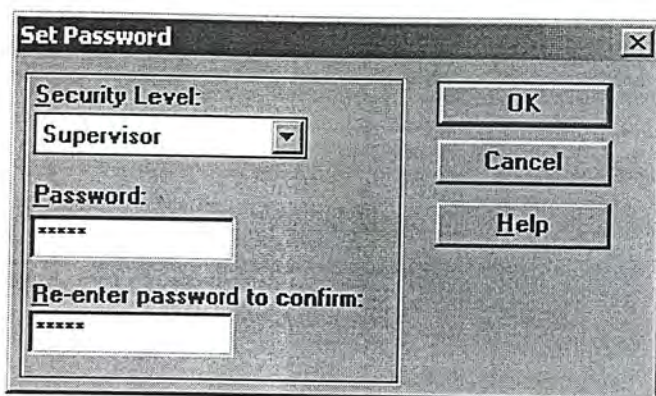
16. ถ้าหากว่าในการจัด Map ไม่ได้กระทำในแบบ Auto Arrange ต้องมาทำการจัด Map เอง เราจะสั่งให้ทำการแสดงเส้น Grid ได้จากเมนู View -> Show Grid / Hide Grid เพื่อสั่งให้ซ่อนหรือแสดงเส้น Grid จะได้ผลการแสดง Map ดังรูปที่ 16



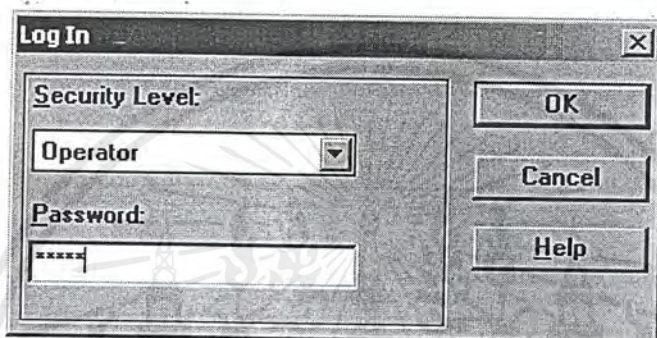
รูปที่ 16 แสดงการแสดงผลเส้น Grid ใน Map ที่เลือก

17. การกำหนดระบบรักษาความปลอดภัยในการเข้าใช้โปรแกรม HP Open View เพื่อกำหนดระดับความปลอดภัยในการใช้งานให้เหมาะสมกับผู้ใช้ในแต่ละระดับ
- ในการกำหนดระบบรักษาความปลอดภัยนั้นก่อนอื่นเราจะต้องกำหนดรหัสผ่านของผู้ใช้ในแต่ระดับเสียก่อน โดยไปที่เมนู Option -> Set Password ดังรูปที่ 17
 - ในการเข้าใช้โปรแกรมก็ให้ทำการ Login เข้าใช้งาน โปรแกรมที่เมนู Option -> Log In แล้ว Login ดังรูปที่ 18 ในการเข้าใช้งานในแต่ละระดับของผู้ใช้ก็จะมีคามสามารถในการใช้งานเมนูนั้นแตกต่างกันไปตามลำดับความสำคัญของผู้ใช้โปรแกรม
 - เมื่อใช้งานเสร็จเรียบร้อยแล้วต้องการยกเลิกการใช้งานระบบในโปรแกรม ก็ให้ทำการ Logout ออก โดยไปที่เมนู Option -> Log Out เพื่อเลิกการใช้งานโปรแกรม ตัวโปรแกรมจะมีเมนูที่เหลือให้ใช้งานได้ดังรูปที่ 19 เท่านั้น

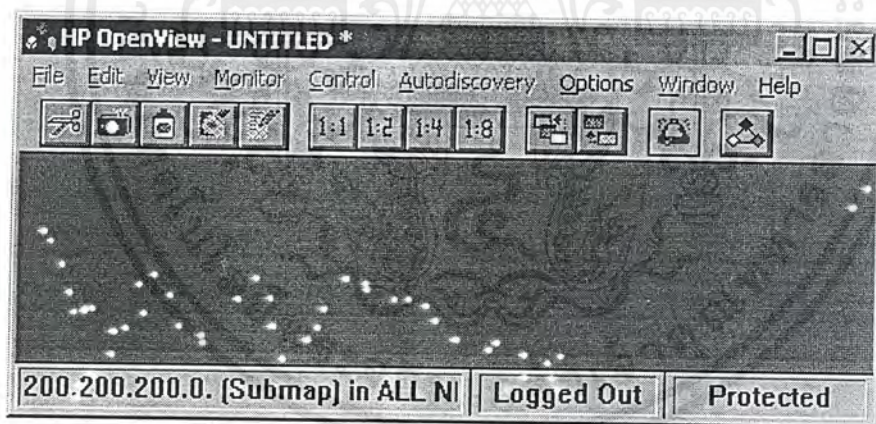
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 17 แสดงการกำหนดรหัสผ่านสำหรับผู้ใช้งานโปรแกรม HP Open View



รูปที่ 18 แสดงการเข้าใช้งานโปรแกรม HP Open View ของผู้ใช้งาน



รูปที่ 19 แสดงตัวโปรแกรมหลังจาก Log Out ออกแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

Java Class สำหรับ SNMP

มีการพัฒนา JAVA tool ในแบบ Freeware Shareware และ แบบ Commercial ออกมาสนับสนุน SNMP ซึ่งในปัจจุบันได้พัฒนาออกมาเป็น SNMP V.3 แต่ที่เป็นที่นิยมใช้กันมากในปัจจุบันคือ SNMP V.2 และ RMON

ในโปรแกรม Network Management อาจแบ่งส่วนหลักๆ ออกเป็น

1. **Structure of Management Interface (SMI)** มีหน้าที่กำหนดชนิดของข้อมูลที่ใช้อยู่ใน MIB ให้ตัวโปรแกรมเข้าใจและจัดเก็บได้อย่างถูกต้อง ส่วนนี้ถูกกำหนดอยู่ใน RFC: RFC1155, RFC1212, RFC1215
2. **Basic Encoding Rules (BER)** เป็นการกำหนดการเข้ารหัสข้อมูลที่ใช้ในการส่งผ่านในระบบ Network
3. **Simple Network Management Protocol (SNMP)** กำหนดคำสั่งในการบริหาร Object เช่น GET, GET_NEXT, SET หรือ TRAP
4. **Management Information Base (MIB)** เป็นที่รวบรวมข้อมูลที่ใช้ในการบริหาร Object

ได้มีผู้พัฒนา TOOL ที่ให้การทำงานครอบคลุมทั้ง 4 หัวข้อหลักไว้โดยในภาษา JAVA มีด้วยกัน 2 แบบคือ

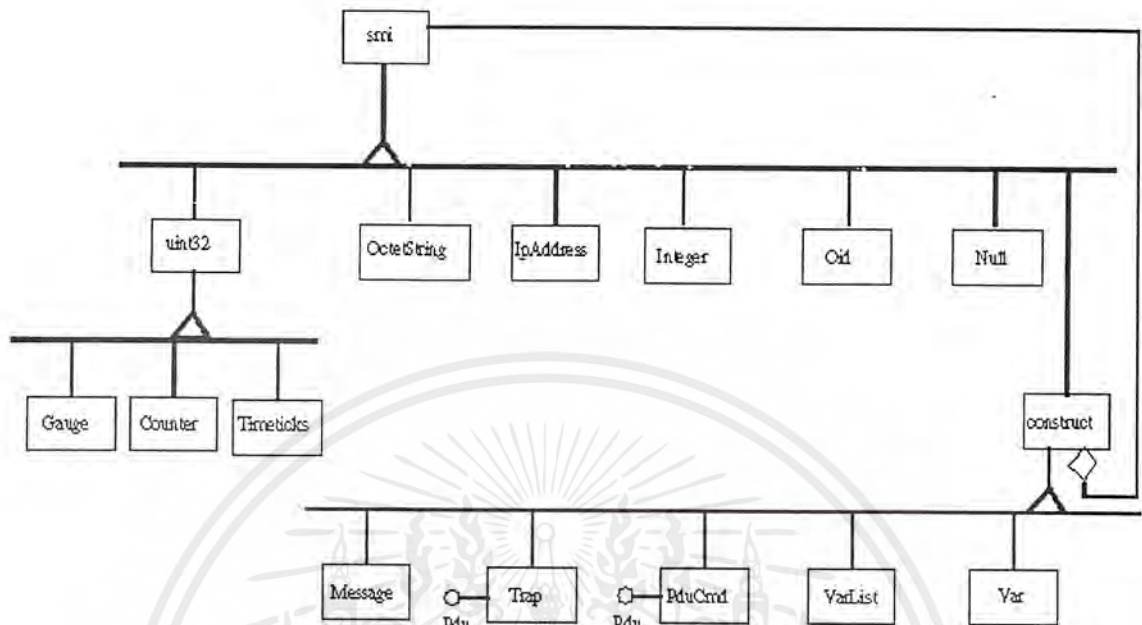
1. **Package Class** เป็นการรวบรวม Class ที่จำเป็นมารวมกันไว้ให้โปรแกรมหลักสามารถเรียกใช้ได้
2. **JAVA bean component** เป็น component class ที่เรียกใช้โดยการส่ง even และ Message มีข้อดีคือสามารถเขียนโปรแกรมในแบบ Visual ได้ ซึ่งในปัจจุบันมักพัฒนาแจกจ่ายกันในรูปแบบของ JAVA bean นี้

ตัวอย่างของ SNMP JAVA Class

SNMP Construction Kit (SCK) เป็น Class ขนาดเล็กที่มีความสามารถไม่มากนักแต่สามารถทำงานครอบคลุมได้ทั้งหมด มีให้ทั้งแบบ Package Class และ JAVA bean ทั้งหมดเก็บอยู่ในรูปของ Jar file เป็น Freeware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตัวของ SCK จะประกอบด้วย Package ย่อยหลายๆ ตัว



รูป แสดง Class Diagram ของ SCK

- SMI เป็นการกำหนดค่าเริ่มต้นสำหรับ Class ทั้งหมด
 - **Unit32** ที่รองรับสัญญาณที่มาจาก BER โดยจะ Inherit ออกมาเป็น 3 class เพื่อรองรับข้อมูลบางอย่างเช่น จำนวนเต็มที่มี 32 บิต หรือ ข้อมูลเกี่ยวกับเวลา
 - **OctetString** ที่มีหน้าที่จัดการกับ Octet String จะมี Class ที่ใช้ convert String หรือเปรียบเทียบ Octet String
 - **IpAddress** แปลงข้อมูลที่เป็น IP Address จาก String ให้อยู่ในรูป Array ที่โปรแกรมสามารถเรียกไปใช้ได้
 - **Integer** จัดการข้อมูล Int
 - **Oid** จัดการ Object ID จาก String ให้อยู่ในรูปที่โปรแกรมสามารถเรียกไปใช้ได้
 - **Null** รองรับข้อมูลที่เป็น Null ที่อาจส่งมาจาก Agent แจ้งให้ โปรแกรมรู้
 - **Construct** เป็น Abstract class ที่จัดการเรื่องของการรับส่ง PDU ที่ใช้ในการบริหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน SCK ยังประกอบด้วย JAVA bean ที่เป็นแบบ visible component ซึ่งสามารถใช้ Developer Software หรือ ใช้โปรแกรม Bean box ในการนำ Bean มารวมกันทำการทดสอบหรือแปลงเป็น Applet เพื่อนำไปใช้งานต่อได้

*Bean Box เป็น Software ที่ใช้ในการพัฒนา JAVA Bean ของ SUN Microsystem Inc. ใช้ทดสอบการทำงานของ bean หรือ เปลี่ยนให้เป็น Application หรือ Applet ได้

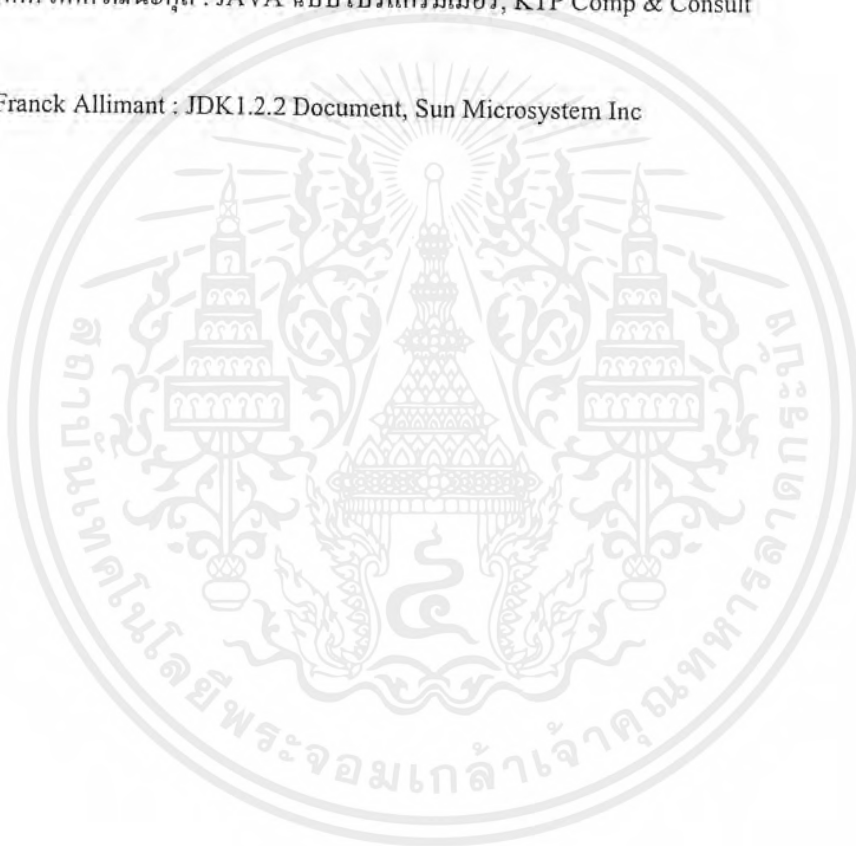
ในโปรแกรม SCK ประกอบด้วย Beans จำนวนหนึ่ง คือ

- **Smi Display** ใช้แสดงข้อมูล เช่น Var หรือ PDU bean ตัวนี้ใช้สำหรับรับ Event ที่ส่งมาและนำข้อมูลออกแสดงในรูปของ ASCII หรือ BER
- **VarEditor** ใช้สร้างหรือแก้ไข Smi object
- **ButtonVar** ใช้สร้างและส่ง smiVar
- **SnmpTube** ใช้ในการกำหนด Address ที่ใช้ในการส่ง SNMP Message
- **GREDLight** เป็นbean ที่ช่วยให้การแสดงผลดูสวยงาม
- **Parser** เป็นตัวกำหนดรูปแบบการแสดงผล
- **Plotter** เป็นตัวแสดงผลข้อมูลในรูปกราฟแท่ง
- **MibBrowser** เป็นตัวเลือก object mib จาก tree

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- 1 William Stalling : SNMP, SNMPv2, and RMON Practical Network Management, Addison Wesley Publishing Company
- 2 Elliott Rusty Harold : JAVA Network Programming, O'Reilly & Associate, Inc
- 3 Jason Hunter : JAVA Servlet Programming, O'Reilly & Associate, Inc
- 4 กิตติ ภัคดีวัฒนะกุล : JAVA ฉบับโปรแกรมเมอร์, KTP Comp & Consult
- 5 Franck Allimant : JDK 1.2.2 Document, Sun Microsystem Inc



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้