

An SNMP Extension with Embedded Management Script*

Tawatchai Srirattanawong¹ and Surasak Sanguanpong²
Department of Computer Engineering
Kasetsart University
Bangkok, Thailand
E-Mail: g40tasr¹, nguan²@ku.ac.th

Abstract: SNMP is a client/server protocol that uses a request/response model. Protocol operations defined in SNMP provide rather monitoring features than management functionalities. SNMP lacks features for networks that require a highly level of management. In this paper, we propose an SNMP extension, which makes use of embedded management script. This enhances SNMP agents to have capabilities to execute scripts submitted from managers without modification of protocol definitions. Scripts are embedded in SNMP messages and will be submitted for execution at agents. Particular MIB objects are defined to handle scripts and their correspondings parameters.

Key words: SNMP, Network Management, Embedded Script.

1. บทนำ

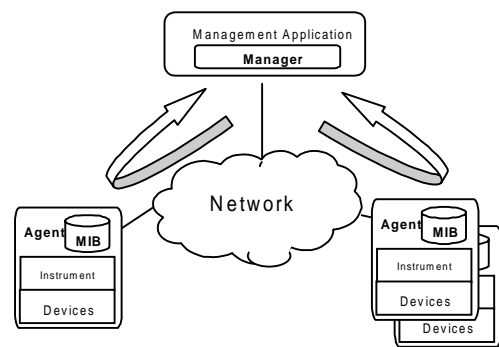
การจัดการเครือข่ายคือกรรมวิธีตรวจสอบและควบคุมเพื่อวางแผนการใช้ทรัพยากรคอมพิวเตอร์และเครือข่ายให้มีประสิทธิภาพและสามารถหาจุดบกพร่องที่เกิดขึ้นเพื่อแก้ไขปัญหาได้อย่างรวดเร็ว คอมพิวเตอร์อย่างน้อยเครื่องหนึ่งในเครือข่ายทำหน้าที่เป็น *แมนเนเจอร์* (manager) ภาระงานของแมนเนเจอร์คือตรวจสอบและควบคุม *เอเจนต์* (agent) ซึ่งเป็นอุปกรณ์ที่มีฟังก์ชันภายในให้ตรวจสอบและปรับเปลี่ยนการทำงานได้ เช่น สับ, เรเตอร์, สวิตช์, หรือพีซี เป็นต้น

เอเจนต์จะส่งข้อมูลเมื่อแมนเนเจอร์ร้องขอหรือปรับการทำงานเมื่อแมนเนเจอร์สั่ง ในเอเจนต์มีฐานข้อมูลสารสนเทศ (management information base) หรือ *มิบ* (MIB) สำหรับเก็บตัวแปรและค่ากำหนดการทำงานองค์ประกอบของเอสเอ็นเอ็มพีแสดงได้ดังรูปที่ 1

เอสเอ็นเอ็มพีมีการจัดการแบบรวมศูนย์ไว้ที่แมนเนเจอร์ ค่าในมิบโดยมากเป็นข้อมูลคิประจําเอเจนต์ การดำเนินการของแมนเนเจอร์ส่วนใหญ่เป็นการสอบถามข้อมูลจากเอเจนต์มากกว่าจะสั่งให้เอเจนต์ปฏิบัติงานใดๆ

ประโยชน์ของใช้เอสเอ็นเอ็มพีจึงเป็นการตรวจสอบและเพื่อวางแผนปรับปรุงเครือข่ายมากกว่าที่จะควบคุมการทำงานหรือสั่งงานให้อุปกรณ์ในเครือข่ายทำงานตอบสนองกับภาวะหรือเหตุการณ์เฉพาะหน้า

หัวข้อถัดไปของบทความนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้อง ส่วนต่อไปกล่าวถึงโครงสร้างระบบที่ออกแบบโดยอธิบายถึงการแฝงสคริปต์และการออกแบบมิบ โมดูลรวมทั้งการสื่อสารระหว่างแมนเนเจอร์และเอเจนต์เพื่อจัดการกับสคริปต์ และหัวข้อสุดท้ายเป็นบทสรุป



รูปที่ 1 : องค์ประกอบการจัดการเครือข่ายในเอสเอ็นเอ็มพี

* งานวิจัยนี้ได้รับทุนสนับสนุนจาก โครงการวิจัย SRU2.41 จากสถาบันวิจัยและพัฒนาแห่งมหาวิทยาลัยเกษตรศาสตร์

2. งานวิจัยที่เกี่ยวข้อง

งานวิจัยด้านการจัดการเครือข่ายโดยเฉพาะเรื่องที่เกี่ยวข้องโดยตรงกับเอสเอ็นเอ็มพีมีมุมมองเน้นที่การพัฒนา รูปแบบและเพิ่มฟังก์ชันของโปรโตคอล พัฒนาการของ เอสเอ็นเอ็มพีมีมาเป็นลำดับจากรุ่น 1 [1] และรุ่น 2 [2,3,4,5] ที่ใช้งานอย่างแพร่หลาย และการกำหนดเป็นโปรโตคอลรุ่นที่ 3 [10] โดยไออีทีเอฟ

งานวิจัยอีกแนวหนึ่งของเอสเอ็นเอ็มพีซึ่งเป็นที่สนใจได้แก่ การพัฒนาให้อเจนท์มีความสามารถในการจัดการตัวเอง [8] เพื่อลดการผูกติดด้านการควบคุมที่ขึ้นตรงกับแมนเนเจอร์ หรือเพิ่มสมรรถนะเอเจนท์โดยใช้ภาษาเอสคิวเอล [16] เพิร์ลสคริปต์ [18] หรือชุดซอฟต์แวร์เพื่อจัดการเครือข่ายเช่น สก็อตต์ [13] แพคเกจสก็อตต์นอกจากจะมีฟังก์ชันการจัดการเครือข่ายแล้วยังให้ผู้ใช้สามารถสร้างสคริปต์ทีซีเอล (TCL) [17] เพื่ออินเทอร์เฟซเข้ากับแพคเกจได้

งานวิจัยส่วนใหญ่ที่มีการนำสคริปต์มาใช้กับเอสเอ็นเอ็มพีมุ่งเน้นการใช้สคริปต์เพื่อช่วยงานด้านตรวจสอบข้อมูลเช่นการใช้ภาษาสก็ม [7] เพิ่มความสามารถของอาร์มอนเอเจนท์ [14,15] ให้มีความคล่องตัวและสามารถตรวจสอบสถานะของเครือข่ายตามสคริปต์ที่กำหนด

วิทยานิพนธ์ปริญญาเอกของโกลซมิดท์ [6] ใช้โมบายล์เอเจนท์ที่สามารถรองรับคำสั่งซึ่งอาจเป็นสคริปต์หรือชุดคำสั่งได้หลายภาษา และขยายรูปแบบเป็นการจัดการเครือข่ายแบบกระจายโดยไม่ได้อิงอยู่กับเอสเอ็นเอ็มพีเพียงอย่างเดียว แต่วิธีการดังกล่าวต้องการสภาพแวดล้อมที่มีความซับซ้อนและมีลักษณะเฉพาะจึงยังไม่เป็นที่นิยมในการไปใช้ทางปฏิบัติ

กลุ่มงาน DISMAN (Distributed Management Working Group) ของไออีทีเอฟ กำลังพัฒนากรอบความคิดในการจัดการออบเจ็กต์โดยใช้สคริปต์เพื่อกำหนดแนวทางที่เป็นมาตรฐานขึ้น [9]

แนวทางการใช้สคริปต์ที่มุ่งเน้นการพัฒนาให้อเจนท์มีสมรรถนะเพิ่มขึ้น และสามารถจัดการตนเองได้ยังเป็นแนวคิดใหม่ที่ยังไม่พบจากเอกสารอ้างอิง ด้านการจัดการเครือข่าย การขยายความสามารถของเอเจนท์ที่จะ

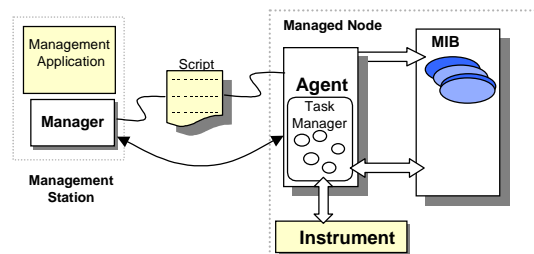
กล่าวในบทความนี้จะนำเสนอใช้วิธีจัดเก็บและงสคริปต์ระหว่างแมนเนเจอร์และเอเจนท์ผ่านมิบ โดยยังคงหลักการพื้นฐานของโปรโตคอลเดิมไว้โดยไม่เปลี่ยนแปลงเพื่อให้ใช้กับเอสเอ็นเอ็มพีได้ทุกรุ่น

3. โครงสร้างระบบ

แนวทางการออกแบบมุ่งเน้นการขยายความสามารถของเอสเอ็นเอ็มพี โดยไม่ผูกติดกับภาษาสคริปต์หรือระบบปฏิบัติการใด ระบบต้องสามารถที่จะส่งสคริปต์จากเอเจนท์หรือแมนเนเจอร์ไปยังหน่วยประมวลผลอื่นจากระยะไกลได้ รวมทั้งสามารถจัดการกับผลลัพธ์ที่ส่งกลับคืนมาในรูปของการแจ้งเตือน และการเก็บค่ากลับเข้าไว้ในมิบได้ การแผ่สคริปต์เป็นการขยายฟังก์ชันของข่าวสารการเพิ่มขึ้นไปเป็นเซตของกลุ่มคำสั่งทำให้ใช้ประโยชน์จากข่าวสารการจัดการเพิ่มขึ้น

องค์ประกอบหลักของระบบสามารถแบ่งเป็น 5 ส่วนและทำงานสัมพันธ์ดังรูปที่ 2 แต่ละส่วนมีหน้าที่การทำงานดังต่อไปนี้คือ

- **แมนเนเจอร์** ทำหน้าที่เช่นเดียวกับแมนเนเจอร์ตามมาตรฐานของเอสเอ็นเอ็มพี และนำส่งสคริปต์ไปให้อเจนท์ รวมทั้งการตรวจสอบและควบคุมงานบนเอเจนท์ที่ได้ส่งผ่านสคริปต์
- **เอเจนท์** ทำหน้าที่เช่นเดียวกับเอเจนท์มาตรฐาน โดยรับสคริปต์มาจากแมนเนเจอร์เพื่อจัดเก็บในรูปของมิบ นอกจากนี้ยังทำหน้าที่เป็นทาสค์แมนเนเจอร์ของสคริปต์ และควบคุมสถานะของงาน



รูปที่ 2 : องค์ประกอบของการจัดการเครือข่ายส่วนขยาย

- **มิม** เก็บมิมอบเจ็คและข้อมูลระบบทั้งหมด โครงสร้างมิมที่ออกแบบจะกล่าวในหัวข้อ 3.2
- **โปรโตคอลจัดการ** โปรโตคอลจัดการซึ่งยังคำสั่งใช้รูปแบบและคำสั่งตามมาตรฐานเอสเอ็นเอ็มพี
- **สคริปต์จัดการ** ตัวสคริปต์เพื่อการจัดการ

```
# create snmp session
set sess1 [snmp session -address 158.108.5.71 /
-port 161 -community private]
# get request
$sess1 get sysDescr.0
```

3.1 สคริปต์จัดการ

เพื่อให้ระบบผูกติดอยู่กับภาษาใดจึงเลือกใช้ภาษาสคริปต์ที่สะดวกต่อการสร้างชุดคำสั่ง สคริปต์ที่เหมาะสมกับระบบที่ออกแบบนี้ควรมีความสามารถในการเข้าถึงมิมอบเจ็คได้ ตัวอย่างของสคริปต์เหล่านี้ได้แก่ ทีซีเอล, สกิม, เพิร์ล หรือ จาวา ข้อดีของการใช้สคริปต์คือไม่ต้องสร้างตัวแปลสคริปต์ อีกทั้งระบบจะคล่องตัวเมื่อจำเป็นต้องแก้ไขสคริปต์ซึ่งอยู่ในรูปของเท็กซ์ไฟล์

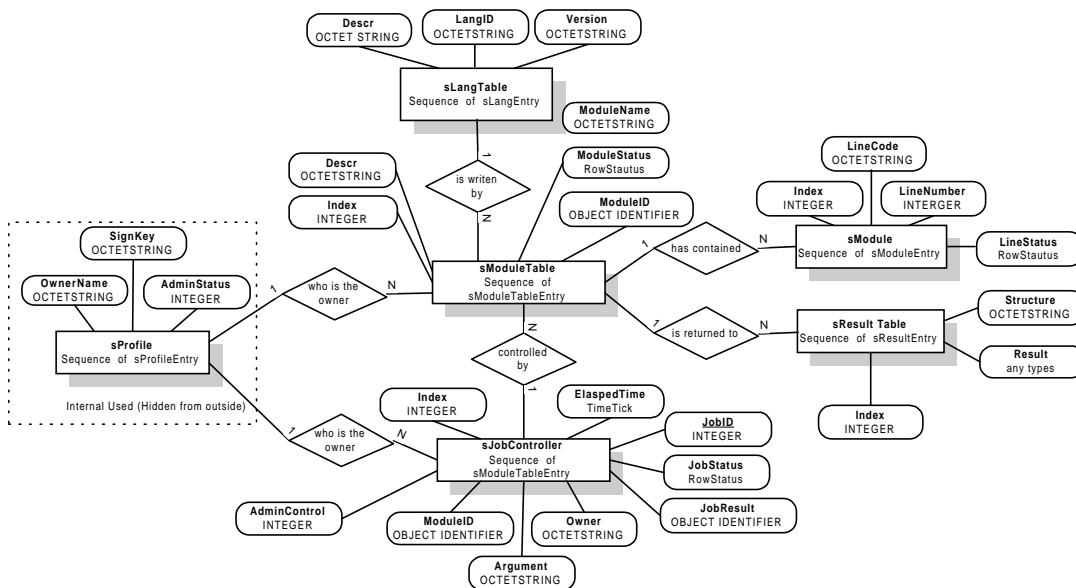
ในงานวิจัยนี้เลือกใช้ทีซีเอล (สามารถใช้ภาษาสคริปต์อื่นได้) เนื่องจากความแพร่หลายบนคอมพิวเตอร์และระบบปฏิบัติการหลายระบบ ซึ่งช่วยให้สามารถศึกษาการทำงานข้ามระบบได้สะดวก

การเข้าถึงมิมอบเจ็คด้วยทีซีเอลจะอาศัยส่วนขยายสคริปต์จากแพ็คเกจสก็อตต์ [14] ตัวอย่างต่อไปนี้แสดงสคริปต์ทีซีเอลที่แมนเนเจอร์ส่งไปร้องขอค่า sysDescr.0 จากเอเจนต์

3.2 มิม

โครงสร้างมิมที่ออกแบบประกอบด้วยตาราง 6 กลุ่มดังรูปที่ 3 แต่ละตารางมีหน้าที่เก็บข้อมูลดังนี้

- **sLangTable** : เก็บรายละเอียดของภาษาสคริปต์ที่เอเจนต์สนับสนุนการทำงาน ทาสก์แมนเนเจอร์จะใช้ค่าใน sLangTable เพื่อเตรียมสภาพแวดล้อมที่จำเป็นให้แก่สคริปต์
- **sProfileTable** : ทำหน้าที่เก็บข้อมูลประจำแมนเนเจอร์ที่มีอยู่ในระบบเครือข่ายนั้น เอเจนต์ใช้ตารางนี้เพื่อตรวจสอบความเชื่อถือได้ของข้อความที่ส่งมาจากแมนเนเจอร์ ข้อมูลมิมส่วนนี้ถือเป็นส่วนสำคัญในการพิสูจน์ตัวตนเพื่อเข้าถึงเอเจนต์
- **sExecModuleTable** : เป็นระบบทะเบียนของสคริปต์แยกตามโมดูลที่มีอยู่ในเอเจนต์โดยจัดเก็บรายละเอียดของโมดูลเช่น ภาษาที่ใช้, ชื่อโมดูล, พอยเตอร์ชี้ไปยังที่เก็บสคริปต์ (OID ชี้ไปยัง sCodeTable) เป็นต้น



รูปที่ 3 : โครงสร้างมิมสำหรับรองรับการใช้สคริปต์

- **sCodeTable** : ใช้เก็บตัวสคริปต์ที่ได้รับมาจากแมนเนเจอร์ในรูปแบบของสายอักขระ สคริปต์แต่ละบรรทัดจะแยกเก็บในแต่ละช่องอาเรย์
- **sResultTable** : ใช้เก็บค่าผลลัพธ์ของงานที่ได้ดำเนินการเสร็จสิ้นและต้องส่งค่าคืนให้แมนเนเจอร์ ตารางมีบนี้สร้างขึ้นในขณะที่ทำงาน ค่าตัวแปรที่ส่งคืนกลับจะต้องนิยามไว้ในสคริปต์ ซึ่งจะอธิบายในหัวข้อ 3.4
- **sJobControllerTable** : ทำหน้าที่จัดเก็บรายละเอียดของกระบวนการทำงานของแต่ละงานเช่น หมายเลขโมดูล (*ModuleID*), อาร์กิวเมนต์ (*Argument*), สถานะงาน (*JobStatus*) และผลการปฏิบัติงาน (*JobResult*) เป็นต้น

3.3 การส่งงานสคริปต์

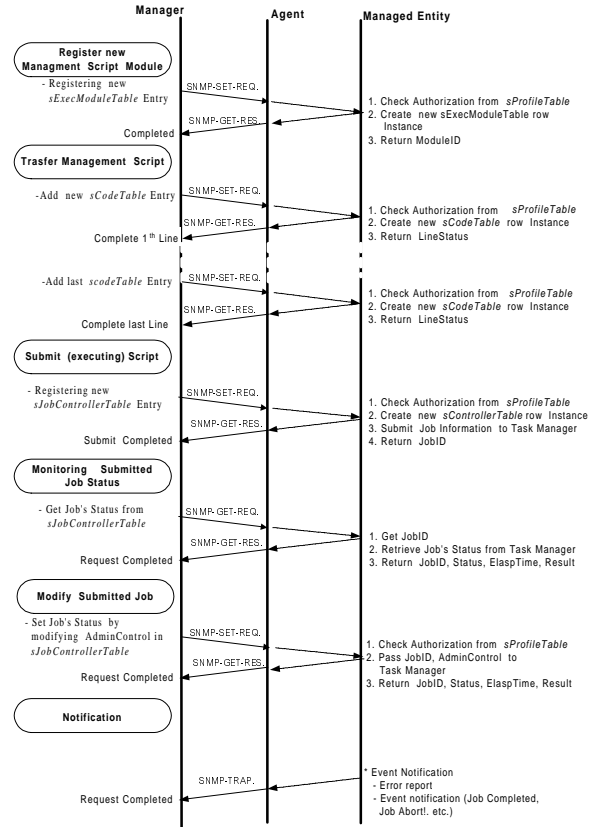
รูปที่ 4 แสดงกระบวนการสื่อสารระหว่างระหว่างแมนเนเจอร์และเอเจนต์ ในที่นี้จะอธิบายกระบวนการส่งงานสคริปต์โดยสรุปซึ่งมีขั้นตอนการทำงานสองส่วนหลักคือ การส่งสคริปต์จากแมนเนเจอร์ไปยังเอเจนต์ และการสั่งให้เอเจนต์ปฏิบัติงานสคริปต์ แต่ละส่วนมีรายละเอียดดังนี้

3.3.1 การส่งสคริปต์ใหม่จากแมนเนเจอร์ไปยังเอเจนต์

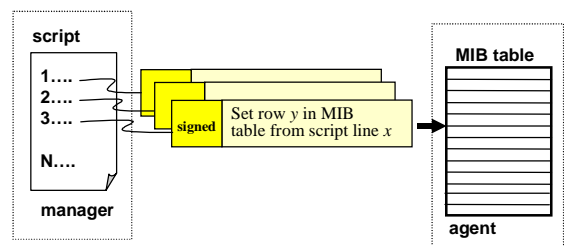
มีขั้นตอนเริ่มต้นด้วย

- (1) การลงทะเบียนสคริปต์ใหม่ แมนเนเจอร์จะเตรียมชื่อโมดูลและรายละเอียดของสคริปต์ที่จะลงทะเบียนใหม่ผ่าน *sExecModuleTable* ด้วยคำสั่ง **SNMP Set Request Message (Create New Row)** เอเจนต์จะตรวจสอบสิทธิการร้องขอ (ตรวจสอบกับ *SignKey* ใน *sProfileTable*) และสร้าง *ModuleID*
- (2) การถ่ายสคริปต์สู่ตารางมีบ แมนเนเจอร์ถ่ายโอนสคริปต์ไปยังเอเจนต์ทีละบรรทัดด้วยคำสั่ง **Create New Row Request** เพื่อเก็บใน *sCodeTable* สคริปต์ทุกบรรทัดจะมีลายเซ็นกำกับตามขั้นตอนวิธีแบบ MD5 ด้วยคีย์ขนาด

128 บิต เมื่อเอเจนต์ตรวจสอบว่าลายเซ็นของแมนเนเจอร์ถูกต้องก็จะรับสคริปต์ไปเก็บไว้ดังแสดงในรูปที่ 5



รูปที่ 4 : การสื่อสารระหว่างแมนเนเจอร์และเอเจนต์แยกตามกระบวนการย่อย



รูปที่ 5 : การถ่ายโอนสคริปต์จากแมนเนเจอร์ไปยังเอเจนต์

3.3.2 การสั่งการให้เอเจนต์ปฏิบัติงานตามสคริปต์

มีขั้นตอนดังนี้คือ

- (1) การสั่งให้ปฏิบัติงาน แมนเนเจอร์ส่งคำสั่งร้องขอด้วย **Request New Row Entry** โดยอ้างอิงถึงสคริปต์โมดูล และพารามิเตอร์อื่นเช่น *Owner*,

Argument ต่อจากนั้นเอเจนต์จะตรวจสอบสิทธิของคำสั่งร้องขอและป้อนงานนี้ไปให้ทาสค์แมนเจอร์ เพื่อเปิดตาราง *sLangTable* สำหรับเตรียมสภาพแวดล้อมให้แก่งานดังกล่าว เอเจนต์จะปฏิบัติงานนั้นทันทีหากค่า *AdminControl* ใน *sJobControllerTable* มีค่าเท่ากับ *Restart*

- (2) การตรวจสอบและควบคุมงาน แมนเจอร์สามารถตรวจสอบสถานะของแต่ละงานที่สั่งทำได้โดยขอค่า *JobStatus* ใน *sJobControllerTable* ค่าที่เป็นไปได้คือ *Running*, *Completed*, *Suspend* หรือ *Aborted* แมนเจอร์สามารถควบคุมสถานะของแต่ละงานโดยส่งคำสั่งกำหนดค่าให้แก่ *AdminControl* ค่าที่เป็นไปได้คือ *Restart*, *Suspend*, *Resume* และ *Abort*

3.4 การแจ้งผลการทำงานของสคริปต์

เอเจนต์แจ้งผลการทำงานผ่านแทรีพ เช่น การหยุดงานกลางคัน (*Aborted*), งานเสร็จสมบูรณ์ (*Completed*) หรือ เกิดข้อผิดพลาด (*Error*) เป็นต้น

3.5 ผลลัพธ์ที่ส่งคืนทางตัวแปร

เพื่อให้แมนเนเจอร์นำผลลัพธ์จากสคริปต์ที่ดำเนินการเสร็จสิ้นแล้วไปใช้ประโยชน์ได้ จึงจำเป็นที่จะต้องมีการนิยามตัวแปรเพื่อแจ้งให้ทาสค์แมนเจอร์ทราบว่าต้องผูกค่าตัวแปรใดในสคริปต์กับตัวแปรในผลลัพธ์ที่ได้จากการทำงานจะเก็บอยู่ในฟิลด์ *JobResult* ของ *sJobControllerTable* (ดูรูปที่ 3 ประกอบ)

ตัวแปรที่ต้องผูกค่าเพื่อส่งกลับจะต้องประกาศไว้ที่ *ส่วนประกาศค่า* ซึ่งอยู่ส่วนบนสุดของแต่ละสคริปต์ โดยใช้ *oVAR* และ *oEND* เป็นตัวกำกับดังรูปที่ 6

3.6 ความปลอดภัยระบบ

ระบบรักษาความปลอดภัยในเอสเอ็นเอ็มพีอาศัยไชรหัสผ่านในรูปแบบสายอักขระที่กำหนดมาในเฟรมเพื่อให้เอเจนต์ใช้ตรวจสอบ ในงานได้ออกแบบให้ระบบมีกลไกตรวจสอบความปลอดภัยของสคริปต์อีกชั้นหนึ่ง

ก่อนที่แมนเนเจอร์จะส่งสคริปต์ไปให้เอเจนต์ แมนเนเจอร์จะนำสคริปต์ไปสร้างลายเซ็นตามขั้นตอนวิธี MD5 [11] และส่งลายเซ็นกำกับท้ายสคริปต์ ทางด้านเอเจนต์เมื่อรับสคริปต์มาแล้วต้องหาลายเซ็นเปรียบเทียบกับลายเซ็นที่ได้รับมาก่อนดำเนินการต่อไป

```
# declaration part
oVAR
  var1 : INTEGER;
  var2 : UNSIGNED32;
  var3 : TIMETICKS;
  var4 : OCTET STRING;
  var5 : OBJECT IDENTIFIER;
  var6 : IPADDRESS;
  var7 : NETWORKADDRESS;
oEND

# coding
set s [snmp session - port 161]
set var1 100
while ($var1 != var2) {
  $s get sysDescr.0 {set var4 [lindex [lindex "%V" 0] 2]}
  incr var1
}
```

รูปที่ 6 : การนิยามตัวแปรเพื่อใช้ส่งค่ากลับ

4. การสร้างซอฟต์แวร์เอเจนต์

เอเจนต์ที่พัฒนาขึ้นตามวิธีการแปลงสคริปต์โดยมีมิมิอ็อบเจ็กต์เก็บอยู่ใน *iso.org.dod.internet.experimental* รูปที่ 7 แสดงบางส่วนของมิมิไฟล์ซึ่งนิยามมิมิอ็อบเจ็กต์ที่ใช้งานตามที่กล่าวในหัวข้อ 3.2

รูปที่ 8 แสดงถึงหน้าจอหลักของการสั่งงานซึ่งแสดงถึงมิมิอ็อบเจ็กต์และรายละเอียดประจำตัวพร้อมทั้งส่วนการควบคุมเพื่อส่งถ่ายสคริปต์และการตรวจสอบระบบ

รูปที่ 9 แสดงจอภาพการทำงานอื่นของระบบซึ่งได้แก่ การตรวจสอบสคริปต์แต่ละโมดูลที่ส่งไปปฏิบัติงาน (จอ ExecModule Explorer) ในเอเจนต์หนึ่งๆว่าประกอบด้วยสคริปต์ใดและมีสภาพแวดล้อมการทำงานอย่างไร รวมทั้งสามารถตรวจสอบสถานะการทำงานปัจจุบันของแต่ละสคริปต์ (จอ JobController Explorer)

5. บทสรุป

การนำสคริปต์แฟงไปกับเฟรมเอสเอ็นเอ็มพีเป็นการขยายบทบาทและหน้าที่ให้อเจนต์มีความสามารถจัดการระบบเครือข่ายได้มากขึ้น ซึ่งแตกต่างจากการใช้เอสเอ็นเอ็มพีเพียงการตรวจค่าในระบบเครือข่ายเท่านั้น กรรมวิธีนี้ทำให้ระบบจัดการเครือข่ายโดยรวมปรับเปลี่ยนรูปแบบจากการจัดการแบบรวมศูนย์ไปสู่การจัดการแบบกระจาย ระบบจัดการเครือข่ายมีความคล่องตัวสูง และเปิดโอกาสให้ปรับแต่งการทำงานและมีประสิทธิภาพ ขึ้น

ภาวะการจัดการเครือข่ายที่แมนเจอร์ จะลดลงเนื่องจากมีอเจนต์ที่ฉลาดขึ้น ระบบจัดการเครือข่ายจะมีความสามารถในการจัดการตนเองได้มากขึ้นซึ่งมีประโยชน์เมื่อระบบสื่อสารระหว่างแมนเจอร์และอเจนต์เกิดปัญหา ในขณะปัจจุบันได้พัฒนาซอฟต์แวร์อเจนต์ที่รองรับสคริปต์และแมนเจอร์สำหรับการตรวจสอบระบบ

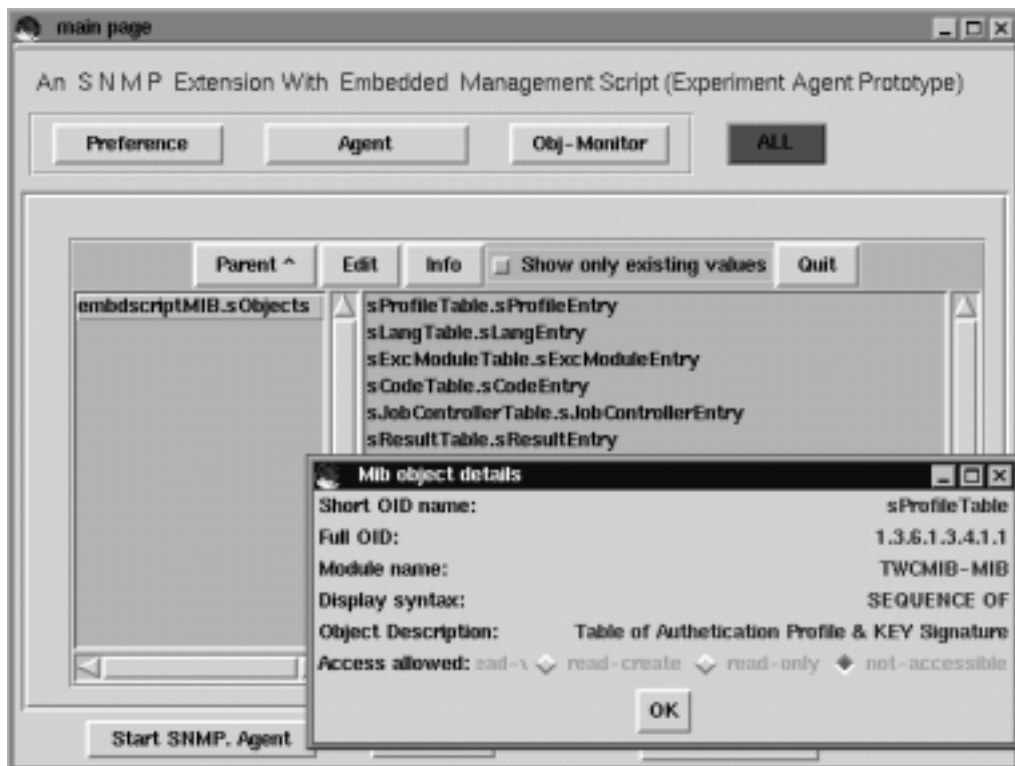
```
rockmin
DEFINITIONS ::= BEGIN
IMPORTS
MODULE-IDENTITY, OBJECT-TYPE, Unsigned32, TimeTicks, experimental
FROM SNMPv2-SMI
RowStatus
FROM SNMPv2-TC;

embedscriptMIB MODULE-IDENTITY
LAST-UPDATED "2001103000Z"
ORGANIZATION "TWSC"
CONTACT-INFO
    "Teewchai Sriratamwong, Suresak Samsuwan,
    Department of Computer Engineering,
    Faculty of Engineering, Kasetsart University,
    Bangkok,
    Thailand,
    E-mail: gskcasr@cestr.ku.ac.th"
DESCRIPTION
    "This MIB module defines the experimental of
    an Extended SNMP Protocol by Embedded Script Language"
 ::= { experimental 4 }

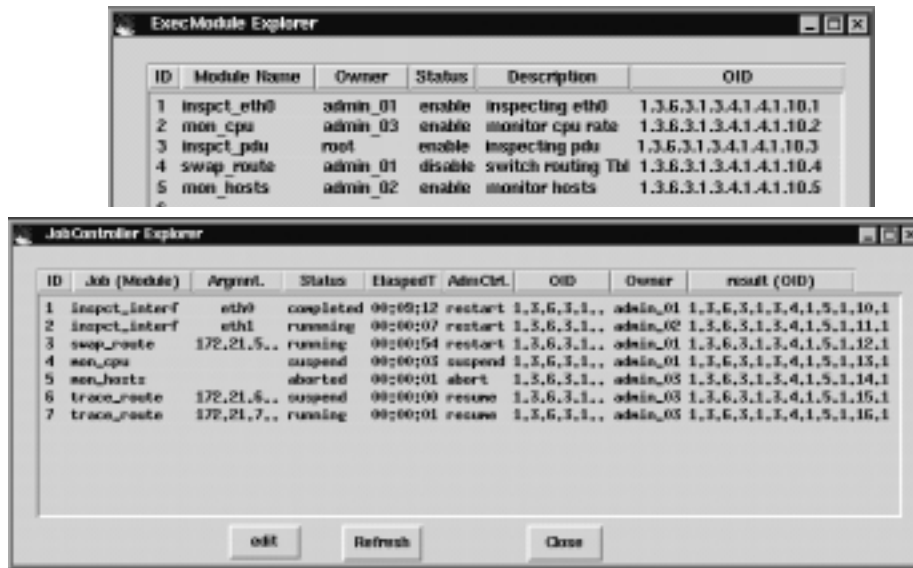
--
-- Groups which was defined in this module
--
objects OBJECT IDENTIFIER ::= { embedscriptMIB 1 }
authentications OBJECT IDENTIFIER ::= { embedscriptMIB 2 }
performance OBJECT IDENTIFIER ::= { embedscriptMIB 3 }

-----
sProfileTable OBJECT-TYPE
SYNTAX SEQUENCE OF SProfileEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Table of Authentication Profile & KEY Signature"
 ::= { objects 1 }
-- example of experimental mib
```

รูปที่ 7 : มิบไฟล์ซึ่งนิยามมิบอ็อบเจ็กต์



รูปที่ 8 : หน้าจอหลักของซอฟต์แวร์จัดการระบบ



รูปที่ 9 : การตรวจสอบสคริปต์และสถานะการทำงานแต่ละสคริปต์

6. เอกสารอ้างอิง

- [1] J. Case, M. Fedor, M. Schoffstall and C. Davin. Simple Network Management Protocol (SNMP). *RFC 1157*, May 1990.
- [2] J. Case, K. McCloghrie, M. Rose and S. Waldbusser. Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2). *RFC 1902*, January 1996.
- [3] J. Case., K. McCloghrie, M. Rose, S. Waldbusser. Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2). *RFC 1905*, January 1996.
- [4] J. Case, K. McCloghrie, M. Rose and S. Waldbusser. Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2). *RFC 1907*, January 1996.
- [5] J. Case, K. McCloghrie, M. Rose and S. Waldbusser. Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework. *RFC 1908*, January 1996.
- [6] G.Goldszmidt, *Distributed Management by Delegation*. PhD. Thesis, Columbia University, New York, December 1996.
- [7] R. Kooijman. *Divide and conquer in network management using event-driven network area agents*. Technical University of Delft, May 1995.
- [8] A. Konstantinou. *NESTOR: Network Self Management and Organization*. <http://www.cs.columbia.edu/dcc/nestor/>.
- [9] D. Levi. Definition of Management Object for the delegation of Management Scripts, *RFC 2592*, May 1999.
- [10] D. Levi, P. Meyer, and B. Stewart. SNMPv3 Applications. *RFC 2273*, January 1998.
- [11] R. Rivest. The MD5 Message-Digest algorithm, *RFC 1321*, April 1992.
- [12] M.T. Rose and K. McCloghrie. Structure and Identification of Management Information for TCP/IP-based Internet. *RFC 1155*, May 1990.
- [13] J. Schonwalder and H. Langendorfer. *Tcl Extensions for Network Management Application*. In Proc. 3rd Tcl/TK workshop, July 1995.
- [14] S. Waldbusser. Remote Network Monitoring Management Information Base. *RFC 1757*, February 1995.
- [15] S. Waldbusser, Remote Network Monitoring Management Information Base Version 2 using SMIV2. *RFC 2021*, January 1997.
- [16] S. Zbaty, M. Lion, and J. Hubaux. *DEAL: A Delegation Agent Language for Development Network Management Functions*. Swiss Federal Institute of Technology, April 1995.
- [17] D. Zeltserman, G. Puopolo. *Building Network Management Tools with Tcl/TK*. Prentice Hall, 1998.
- [18] A. Zwemmer, E. Hengstum, and M. Sinderen. Developing an SNMP agent protocol entity with object oriented Perl. *ECOOP'96, Workshop on Object Oriented Technology for Service and Network Management*, Linz, Austria, July 1996.