

การจำลองกับ SLAM II

กฤษณา ศุขุมอก*

1. คำนำ

ในเรื่องของการสร้างตัวแบบการจำลอง เพื่อใช้ในการศึกษาหรือแก้ปัญหาระบบงานนั้น จำเป็นที่ต้องใช้โปรแกรมคอมพิวเตอร์ในการสร้างตัวแบบให้สอดคล้องกับระบบการทำงานจริง โปรแกรมดังกล่าว ผู้ศึกษาอาจเลือกใช้ภาษาทางคอมพิวเตอร์ ภาษาใด ภาษาหนึ่ง ในการเขียนโปรแกรม เช่น ภาษา C ภาษา PASCAL หรือภาษา FORTRAN เป็นต้น ทั้งนี้ขึ้นอยู่กับความต้องการ ความรู้ หักษ์ หรือแม้กระทั่งความชอบของผู้ศึกษาเป็นสำคัญ นอกจากนี้ยังต้องคำนึงถึงความเหมาะสม สมของภาษาหนึ่งๆ กับตัวแบบการจำลองอีกด้วย อย่างไรก็ตามไม่ว่าจะใช้ภาษาใดในการสร้างตัวแบบ จากประสบการณ์ของผู้เขียนพบว่าเป็นงานที่ใช้เวลามาก ถึงแม้ว่าระบบงานที่นำมาสร้างเป็นตัวแบบนั้นจะไม่ยาก แต่โปรแกรมที่เขียนอาจประกอบไปด้วยคำสั่งหลายร้อยบรรทัด หรือบางครั้งเป็นพันหรือเป็นหมื่นบรรทัด ทั้งนี้ขึ้นอยู่กับความซับซ้อนของระบบงาน และเมื่อเป็นเช่นนี้ไม่ใช่พำนัชใช้เวลาที่ค่อนข้างมากเพียงอย่างเดียว หากจะพบว่าคำใช้จ่ายในการสร้างตัวแบบดังกล่าวก็มากไปด้วยเช่นกัน

ภาษาการจำลอง (simulation language) เป็นคีกทางเลือกหนึ่งของการสร้างตัวแบบการจำลองจากระบบงาน ซึ่งเป็นที่นิยมใช้กันมากในปัจจุบัน ทั้งนี้ เพราะว่าภาษาการจำลองได้ถูกพัฒนาขึ้นมาเพื่อใช้ในการจำลองโดยเฉพาะ โครงสร้างของภาษาถูกวางแผนเพื่อให้การสร้างตัวแบบการจำลอง เป็นไปได้โดยง่าย รวดเร็ว และมีประสิทธิภาพสูง ภาษาการจำลองส่วนมากจะมีฟังก์ชันการทำงาน ต่างๆ หรือโปรแกรมย่อย (subprograms) ที่จะนำมาช่วยในการสร้างตัวแบบการจำลองให้อายุ มากมาย คำสั่งหลายล้านหรือหลายร้อยบรรทัดของการเขียนโปรแกรมด้วยภาษาจะลดต่ำลง อาจเทียบได้กับคำสั่งของภาษาการจำลองเพียงไม่กี่บรรทัด นั่นหมายถึงว่าการสร้างตัวแบบการจำลองด้วยภาษา การจำลองสามารถสร้างด้วยความรวดเร็วกว่าการสร้างตัวแบบที่ไม่ใช้ภาษาการจำลอง นอกจากนี้จากประสบการณ์ของผู้เขียนพบว่า ตัวแบบการจำลองที่เขียนด้วยภาษาที่ไม่ใช้ภาษาการจำลองนั้น ส่วนใหญ่ไม่ได้คำนึงถึงโครงสร้างของข้อมูล (data structure) ซึ่งเป็นผลให้การนำตัวแบบการจำลองไป

* อาจารย์ คณะบริหารธุรกิจ มหาวิทยาลัยเกริก

ประยุกต์ใช้มีประสิทธิภาพไม่ดีเท่าที่ควร ซึ่งจุดนี้ดูเหมือนว่าจะเป็นสาเหตุให้เปรียบอีกข้อหนึ่งของภาษาการจำลอง เพราะได้มีการวางแผนสร้างของภาษาให้ใช้โครงสร้างของข้อมูลให้เหมาะสมในขณะที่ทำการจำลองตัวแบบ

SLAM II เป็นภาษาการจำลองภาษาหนึ่ง ที่しまากในการนำมาสร้างตัวแบบห้องทางด้านธุรกิจ การจัดการ คอมพิวเตอร์ อุตสาหกรรม เป็นต้น SLAM II เป็นภาษาการจำลองที่พัฒนาขึ้นมาจากการเขียนภาษา FORTRAN ซึ่งนอกเหนือจากคุณสมบัติของภาษาการจำลองทั่วๆ ไปแล้ว สิ่งที่นับว่าทำให้ SLAM II เป็นภาษาที่นิยมใช้มากก็คือ ความง่ายของการใช้ ห้องนี้เพื่อการสร้างตัวแบบการจำลองโดยใช้ SLAM II สามารถสร้างได้โดยใช้สัญลักษณ์ต่างๆ มาประกอบกันขึ้นเป็นตัวแบบ ที่เรียกว่าตัวแบบรูปภาพ (graphical models) จากนั้นตัวแบบรูปภาพจะถูกแปลเป็นโปรแกรมคำสั่งที่ใช้ในการจำลอง ได้เองโดยอัตโนมัติ และในทางกลับกัน ตัวแบบที่อยู่ในรูปโปรแกรมคำสั่งก็สามารถแปลให้อยู่ในรูปของตัวแบบรูปภาพได้โดยอัตโนมัติเช่นกัน นอกจากนี้ SLAM II ยังประกอบไปด้วยโปรแกรมย่อยที่สามารถใช้ได้ทั้งในตัวแบบที่เป็นเหตุการณ์ไม่ต่อเนื่อง (discrete events) และตัวแบบต่อเนื่อง (continuous models)

2. การสร้างตัวแบบการจำลองโดยใช้ SLAM II

การสร้างตัวแบบการจำลองสามารถแบ่งออกเป็น 2 ขั้นตอนใหญ่ๆ คือ การสร้างตัวแบบรูปภาพซึ่งจะเป็นการจำลองเหตุการณ์ หรือการทำงาน ต่างๆ ของระบบ และการสร้างคำสั่งควบคุม ซึ่งจะทำหน้าที่ควบคุมการทำงานของตัวแบบที่สร้างขึ้น ตัวอย่างเช่น การกำหนดระยะเวลาของการจำลอง การกำหนดค่าตัวแปรเริ่มต้น การกำหนดการทำงานที่สำคัญของการจำลอง เป็นต้น ใน การสร้างตัวแบบรูปภาพ และการสร้างคำสั่งควบคุม สามารถทำได้โดยใช้ตัวสร้างเครือข่าย (network builder) และตัวสร้างคำสั่งควบคุม (control builder) ที่อยู่ใน SLAM II ตามลำดับ

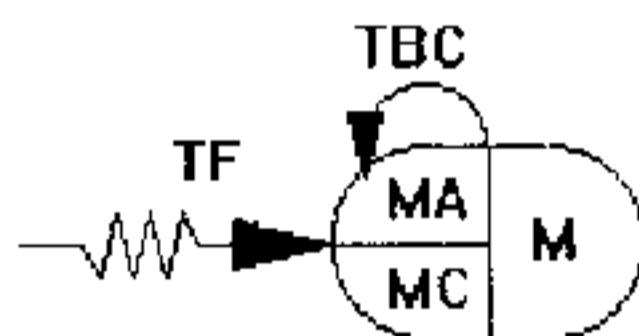
2.1 การสร้างตัวแบบรูปภาพ ดังที่ได้กล่าวมาแล้วข้างต้น ตัวแบบการจำลองใน SLAM II เริ่มต้นจากการสร้าง ตัวแบบรูปภาพโดยใช้ตัวสร้างเครือข่าย ตัวแบบรูปภาพจะประกอบด้วย สัญลักษณ์เครือข่าย (network symbols) ซึ่งมีอยู่หลายหลายแบบ แต่ละแบบจะแสดงฟังก์ชันการทำงานที่แตกต่างกัน อย่างไรก็ตาม มีสัญลักษณ์เพียง 7 แบบที่ถือว่าเป็นสัญลักษณ์พื้นฐานของ การสร้างตัวแบบการจำลอง สัญลักษณ์ดังกล่าวคือ

1. CREATE node
2. QUEUE node

3. TERMINATE node
4. ASSIGN node
5. ACTIVITY branch
6. GOON node
7. COLCT node

● **CREATE node** CREATE node เป็นสัญลักษณ์เครือข่าย ที่ใช้ในการสร้าง หรือผลิต entity (อาจเป็น คน สัตว์ สิ่งของ หรือ อื่นๆ) เข้าสู่ระบบ โดยผ่านเส้นทาง หรือกิจกรรม (activity) ที่เชื่อมต่อเข้ากับ CREATE node นี้ สัญลักษณ์ของ CREATE node แสดงในรูปที่ 1

รูปที่ 1 CREATE node



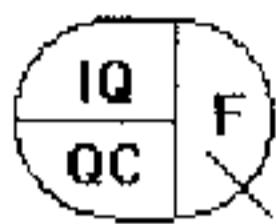
- | | |
|-----|--|
| TF | คือ จุดเวลาเริ่มต้นที่ entity แรกถูกสร้างขึ้น |
| TBC | คือ ระยะเวลาระหว่างการสร้างสอง entity ใดๆ |
| MC | คือ จำนวนมากที่สุดของ entity ที่ CREATE node สร้าง |
| MA | คือ เวลาที่ entity แต่ละตัวถูกสร้างขึ้น ซึ่งเป็นค่าประจำตัว (attribute) ของ entity นั้นๆ |
| M | คือ จำนวนเส้นทางหรือ จำนวนกิจกรรม ที่เชื่อมต่อเข้ากับ node |

สัญลักษณ์ของ CREATE node ดังกล่าวเมื่อเปลี่ยนค่าสั่งของภาษาการจำลอง จะเทียบเท่าได้กับ CREATE, TBC, TF, MA, MC, M;

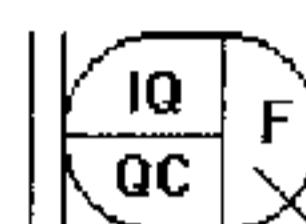
● **QUEUE node** QUEUE node เป็นสัญลักษณ์แสดงสถานที่ในระบบที่ entity มาขอรับบริการจากผู้ให้บริการ ซึ่งการบริการจะเห็นด้วยกิจกรรมที่เชื่อมต่อจาก QUEUE node เมื่อได้กีตامที่ entity มาถึง QUEUE node และขณะนั้นผู้ให้บริการว่าง entity จะผ่าน QUEUE node ตั้งกล่าว เช่นไปรับบริการทันที และถ้า entity มาถึง QUEUE node แต่ผู้ให้บริการไม่ว่าง entity ตั้งกล่าวจะต้องรอที่ QUEUE node และจะถูกเก็บไว้ในไฟล์ที่ได้กำหนดไว้ entity ที่อยู่ในไฟล์ หรืออีกหนึ่ง

อยู่ในແກວຄອຍຈະອອກຈາກແກວຄອຍທັນທີທີ່ຜູ້ໃຫ້ບໍລິການວ່າງ ສັນລັກຊົນຂອງ QUEUE node ແສດງໃນຮູບ
ທີ 2

ຮູບທີ 2 QUEUE node



ຮູບທີ 3 QUEUE node ແມ່ນມີການ BLOCK



IQ ດືອ ຈຳນວນ entity ເຮີມຕັ້ນເຖິງກຳຫັດໃຫ້ອູ່ໃນແກວຄອຍ

QC ດືອ ຈຳນວນສູງສຸດຂອງ entity ທີ່ຍອມໄຫຼມໃນແກວຄອຍ

F ດືອ ມາຍເລີຂອງໄຟລ໌ທີ່ເກີບ entity ທີ່ຕ້ອງຮອໃນແກວຄອຍ

ສັນລັກຊົນຂອງ QUEUE node ເນື້ອແປລເປັນການຫາກາຈຳລອງເລົ້ວຈະອູ່ໃນຮູບແບບດັ່ງນີ້

QUEUE (F), IQ, QC;

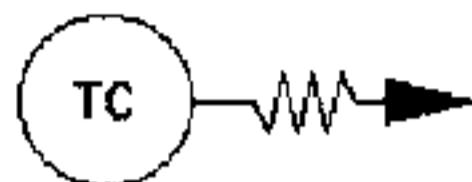
ນອກໄປຈາກນີ້ ຄ້າ entity ແພິມາເຖິງ QUEUE node ແລະປຣາກງູວ່າແກວຄອຍເຕີມ ດືອມ
ຈຳນວນ entity ກ່ອນທີ່ entity ນີ້ຄູ່ເຫັນ QC ແລ້ວ ເຮັດວຽກເລືອກກຳຫັດໃຫ້ entity ນີ້ ໄປໆ
node ອື່ນ (Balking) ຮີ້ວີ້ໄຫ້ຮອຈນກວ່າຈະມີທີ່ວ່າງໃນແກວຄອຍ (Blocking) ຮີ້ວີ້ໄຫ້ອອກຈາກຮະບນກີ່ໄດ້

ໃນການທີ່ແກວຄອຍເຕີມ ແລະ entity ຖຸກກຳຫັດໃຫ້ໄປທີ່ node ທີ່ຊື່ Q1 ຄໍາສັ່ງຕັ້ງກລ່າງຈະເປັນ
ດັ່ງນີ້ QUEUE (F), IQ, QC, BALK (Q1);

ແລະການທີ່ແກວຄອຍເຕີມ ແຕ່ entity ບຸກກຳຫັດໃຫ້ຮອ ຄໍາສັ່ງຈະເປັນດັ່ງນີ້ QUEUE (F), IQ,
QC, BLOCK;

TERMINATE node TERMINATE node ເປັນສັນລັກຊົນທີ່ໃຫ້ໃນການທຳລາຍ entity ຢີ້ວ່າ entity
ອອກຈາກຮະບນ ທັນທີ່ entity ຜ່ານ node ນີ້ ນອກຈາກນີ້ TERMINATE node ຍັງສາມາດໃຫ້ໃນການ
ກຳຫັດຮະຍະເວລາກາຈຳລອງທາງອ້ອມໄດ້ວັດວາຍ ກລ່າວຄົວກາຈຳລອງຈະສິ້ນສຸດລົງເນື້ອຈຳນວນຂອງ
entity ທີ່ຝ່ານ node ນີ້ມີຄ່າເທິກັນຄ່າທີ່ກຳຫັດ ຈຶ່ງຄົວຄ່າຂອງ TC ໃນສັນລັກຊົນຂອງ TERMINATE
node ທີ່ແສດງໃນຮູບທີ 4

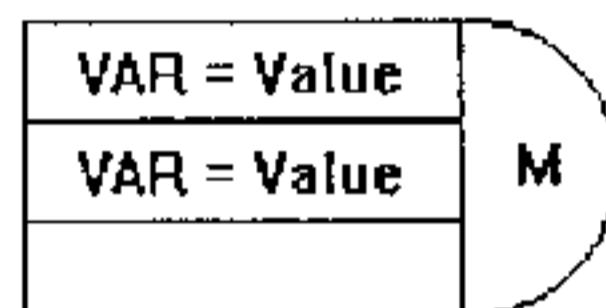
รูปที่ 4 TERMINATE node



สำหรับรูปแบบค่าสั่งของ TERMINATE node จะเป็นดังนี้ TERMINATE, TC;

- **ASSIGN node** ASSIGN node เป็นสัญลักษณ์ที่ใช้สำหรับ กำหนดค่าประจำตัวให้กับ entity ที่ผ่าน node นี้ หรือใช้ในการกำหนดค่าให้กับตัวแปรต่างๆ ในตัวแบบ สัญลักษณ์ของ ASSIGN node แสดงในรูปที่ 5

รูปที่ 5 ASSIGN node



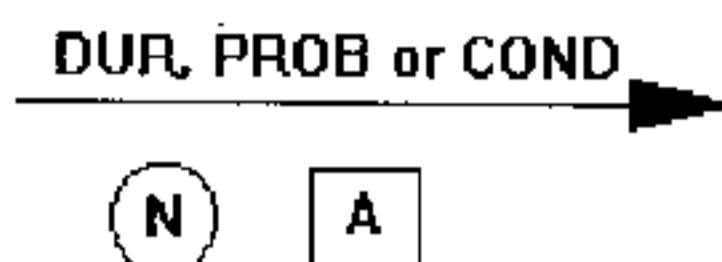
VAR = Value หมายถึงการกำหนดค่า Value ให้กับตัวแปร VAR ส่วนค่า M หมายถึงจำนวนกิจกรรม หรือเส้นทางเชื่อมต่อจาก node นี้ สัญลักษณ์ของ ASSIGN node เทียบได้กับคำสั่งของการจัดอย่างดังนี้ ASSIGN, VAR = Value, VAR = Value,..., M;

- **ACTIVITY branch** ใช้ในการแสดงกิจกรรมต่างๆ ในตัวแบบของระบบ ซึ่ง ACTIVITY branch นี้สามารถแบ่งออกเป็น 2 ลักษณะ คือ ACTIVITY branch ที่เชื่อมต่อมาจาก QUEUE node ซึ่งเทียบได้กับ กิจกรรมการให้บริการกับ entity ที่เข้ามาที่ QUEUE node ซึ่งกรณีนี้จำนวนของ entity ที่จะทำกิจกรรม ณ เวลาหนึ่ง จะถูกกำหนดโดยจำนวนผู้ให้บริการที่ได้กำหนดไว้ที่ ACTIVITY branch อีกลักษณะหนึ่งคือ ACTIVITY branch ที่เชื่อมต่อจาก node อื่นๆ ที่ไม่ใช่ QUEUE node จะไม่มีข้อจำกัดของจำนวน entity ที่จะทำกิจกรรมนั้น

ทุก ACTIVITY branch จะต้องมี node เริ่มต้น และ node ปลายทาง ระยะเวลาของการเดินทางของ entity ที่ผ่าน ACTIVITY branch นี้คือระยะเวลาในการทำกิจกรรมของ entity นั้น ระยะเวลาตั้งกล่าวอาจเป็นค่าคงที่ (constant) หรืออยู่ในรูปของฟังก์ชันการแจกแจงความน่าจะเป็น (probability distribution function) นอกจากนี้ในกรณีที่มีมากกว่าหนึ่ง ACTIVITY branch

เชื่อมสอง node เข้าด้วยกัน เราสามารถกำหนดความน่าจะเป็น หรือเมื่อไหร่การเลือก ACTIVITY branch หรือการเลือกกิจกรรมได้อีกด้วย สัญลักษณ์ของ ACTIVITY branch แสดงในรูปที่ 6

รูปที่ 6 ACTIVITY branch



N คือ จำนวนผู้ให้บริการ เมื่อ ACTIVITY branch นั้นเชื่อมต่อจาก QUEUE node

A คือ หมายเลขของกิจกรรม (activity number)

DUR คือ ระยะเวลาในการทำกิจกรรม

PROB คือ ความน่าจะเป็นของการเลือกทำกิจกรรมนั้น

COND คือ เมื่อไหร่ของการเลือกที่จะทำกิจกรรมนั้น

NLBL คือ ชื่อของ node ปลายทาง

สำหรับรูปแบบคำสั่งของ ACTIVITY branch นี้คือ

ACTIVITY (N)/A, DUR, PROB or COND, NLBL;

- **GOON node** เป็นสัญลักษณ์ที่ใช้สำหรับสร้างตัวแบบที่มีกิจกรรมเป็นขั้นตอนต่อเนื่องกัน (sequential activities) ทั้งนี้เนื่องจาก ACTIVITY branch หนึ่งๆ ต้องเชื่อมโยงสอง node เข้าด้วยกัน ดังนั้นถ้ามีกิจกรรมที่ต่อเนื่องกันไป เราจำเป็นต้องใช้ GOON node มาช่วยเพื่อให้เกิดความต่อเนื่องดังกล่าว สัญลักษณ์ของ GOON node แสดงในรูปที่ 7

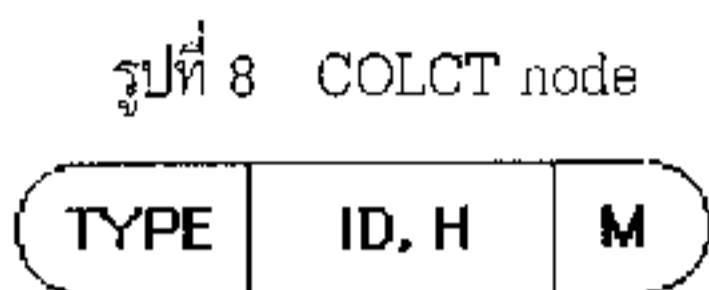
รูปที่ 7 GOON node



M คือ จำนวน ACTIVITY branch ที่เชื่อมโยงต่อจาก GOON node

สัญลักษณ์ของ GOON node เพียงได้กับคำสั่ง GOON, M;

- **COLCT node** COLCT node เป็นสัญลักษณ์ที่ใช้ในการเก็บค่าสถิติของตัวแปรแบบต่างๆ เช่น เวลาแรกที่มาถึง (time of first arrival) ของ entity ที่ node นี้ ระยะเวลาระหว่างการมา (time between arrivals) ของ entity หรือ จำนวนของ entity ทั้งหมดที่ผ่านมาถึง node นี้ เป็นต้น สัญลักษณ์ของ COLCT node แสดงในรูปที่ 8



- TYPE หมายถึง ชนิดของตัวแปรที่เก็บค่าสถิติว่าเป็นแบบใด เช่น FIRST คือเวลาแรกที่มาถึง หรือ BETWEEN คือ ระยะเวลาระหว่างการมาของ entity เป็นต้น
- ID หมายถึง ชื่อตัวแปรที่แสดงค่าสถิติ
- H หมายถึง การแสดงขิสโตแกรม (histograms) ของค่าสถิติตั้งกล่าว ซึ่งจะต้องกำหนดให้อยู่ในรูปของ NCEL/HLOW/HWID โดยที่ NCEL คือ จำนวนของเซล HLOW คือ ขีดจำกัดบน (upper limit) ของเซลแรก และ HWID คือ ความกว้างของเซล
- M คือ จำนวน ACTIVITY branch ที่ เชื่อมโยงต่อจาก COLCT node
- สัญลักษณ์ตั้งกล่าวเทียบได้กับค่าสั่ง COLCT, TYPE, ID, NCEL/HLOW/HWID, M;

2.2 การสร้างคำสั่งควบคุม คำสั่งควบคุม หมายถึงคำสั่งที่ใช้ในการกำหนด หรือให้ข้อมูลที่เกี่ยวกับตัวแบบการจำลอง ในคำสั่งควบคุมของตัวแบบหนึ่งๆ ประกอบไปด้วยคำสั่งควบคุมหลายคำสั่ง ซึ่งแต่ละคำสั่งทำหน้าที่แตกต่างกัน ทั้งนี้ขึ้นอยู่กับชนิดของตัวแบบ อย่างไรก็ตามอย่างน้อยที่สุด คำสั่งของตัวแบบ ต้องประกอบด้วย 3 คำสั่งต่อไปนี้

1. GENERAL
2. LIMITS
3. FINISH

- **GENERAL** คำสั่ง GENERAL เป็นคำสั่งที่ให้ในการให้ข้อมูลทั่วๆ ไปที่เกี่ยวกับการจำลอง เช่นผู้สร้างตัวแบบ (NAME of modeler) ชื่อของโครงการ (PROJECT title) วันที่ทำการ

จำลอง (MONTH/DAY/YEAR) และจำนวนการจำลองช้า (Number of Runs หรือ NNRNS) ซึ่งอยู่ในรูปแบบดังนี้

GEN, NAME, PROJECT, MONTH/DAY/YEAR, NNRNS;

นอกจากนี้ยังรวมถึง การเลือกที่จะกำหนด พิมพ์คำสั่งต่างๆ พิมพ์รายงาน หรือการกำหนดขนาดความกว้างของรูปแบบรายงาน เป็นต้น

- **LIMITS** คำสั่ง LIMITS ใช้ในการกำหนดข้อมูลเกี่ยวกับตัวแแบบ ซึ่งประกอบด้วย จำนวนสูงสุดของไฟล์ (maximum number of files หรือ MFIL) จำนวนสูงสุดของค่าประจำตัว (maximum number of attributes หรือ MATR) และจำนวนสูงสุดของ entity (maximum number of entities หรือ MNTRY) ที่ประเมินว่าจะอยู่ในไฟล์ต่างๆ ทั้งหมดในช่วงของการจำลอง โดยอยู่ในรูปแบบดังนี้

LIM, MFIL, MATR, MNTRY;

- **FIN** เป็นคำสั่งที่ใช้ในการปั่นออกถึง การสิ้นสุดของคำสั่งควบคุมของตัวแบบ ทั้งที่อยู่ในรูปแบบของ

FIN;

นอกจากนี้ยังมี คำสั่งควบคุมที่จัดไว้สำหรับ และพบมากในตัวแบบต่างๆ เช่น

INITIALIZE เป็นคำสั่งที่ใช้ในการกำหนดระยะเวลาของการจำลอง

TIMST เป็นคำสั่งที่ใช้ในการเก็บค่าสถิติของตัวแปรที่กำหนดในช่วงเวลาหนึ่ง

INTLC เป็นคำสั่งที่ใช้ในการกำหนดค่าเริ่มต้นให้กับตัวแปร ต. ๑๗

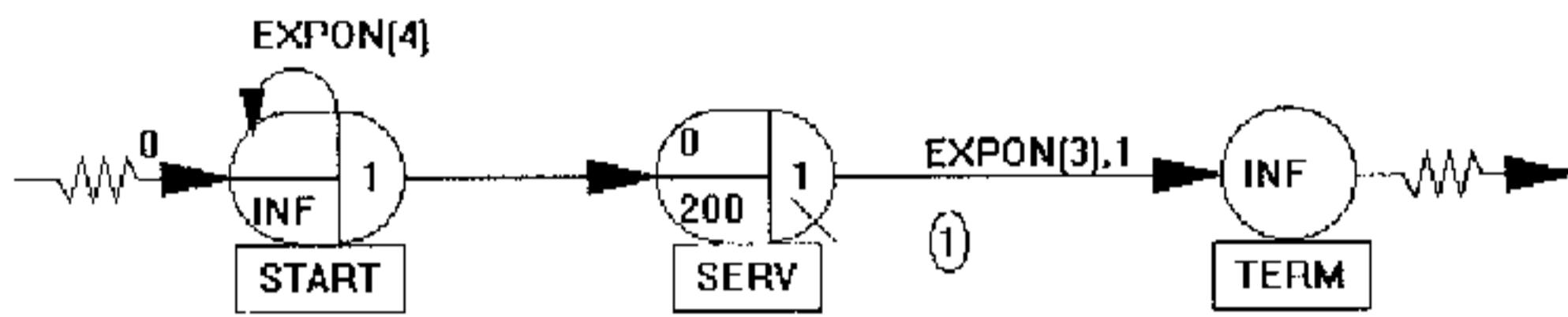
ENTRY เป็นคำสั่งที่ใช้ในไฟล์ entity เริ่มต้น เพื่อไปเก็บไว้ในไฟล์

2.3 ตัวอย่างการสร้างตัวแบบการจำลอง โดยใช้ SLAM II เพื่อให้เห็นภาพของการสร้างตัวแบบการจำลอง เราจะพิจารณาระบบ隊คอย หรือ Queue ที่มีผู้ให้บริการ 1 คน สมมติว่า ช่วงเวลาระหว่างการมาของลูกค้าที่มาเข้ารับบริการ มีการแจกแจงแบบ Exponential ที่มีค่าเฉลี่ยเท่ากับ 4 นาที และเวลาที่ได้รับบริการมีการแจกแจงแบบ Exponential เช่นกันและมีค่าเฉลี่ยเท่ากับ 3 นาที

ในที่นี้ เราสามารถสร้างตัวแบบรูปภาพของระบบได้อย่างง่าย โดยใช้สัญลักษณ์ที่ประกอบไปด้วย CREATE node QUEUE node TERMINATE node และ ACTIVITY branch ดังแสดงในรูปที่ 9 ซึ่งกำหนดให้จำนวน entity หรือ จำนวนลูกค้าที่เข้ามารับการบริการที่ CREATE node นั้นไม่จำกัด จำนวนคนที่มากที่สุดที่ยอมให้มีในແකคอยที่ QUEUE node เท่ากับ 200 คน และ

กำหนดให้จำนวนผู้ที่เสริจากงานให้บริการที่ฝ่ามามาก TERMINATE node ไม่จำกัด นั้นหมายถึง การหยุดการทำงานจำลองไม่สามารถกำหนดได้จาก TERMINATE node แต่สามารถกำหนดเวลาที่ใช้ในการทำการจำลองจากค่าสั่งควบคุม

รูปที่ 9 ตัวแบบรูปภาพของระบบແຄອຍ



สำหรับค่าสั่งควบคุมของตัวแบบ นอกเหนือจากค่าสั่ง GENERAL LIMITS และ FIN ที่ต้องใช้ในตัวแบบการจำลองแล้ว เราจำเป็นต้องเพิ่มค่าสั่ง INITIALIZE เพื่อเป็นการกำหนดระยะเวลา การจำลอง ซึ่งเนื่องจากเรากำหนดไว้ 100,000 นาที และยังต้องเพิ่มค่าสั่ง NETWORK เพื่อเป็นการกำหนดว่า มีตัวแบบรูปภาพที่ได้สร้างขึ้นมาจากการ ค่าสั่งควบคุมของตัวแบบตั้งกล่าวแสดงได้ดังนี้ ค่าสั่งควบคุม

GEN,GRIDSANA,SEMINAR,06/18/1997,1,Y,Y,Y/Y,Y,Y/1,132;

LIMITS,1,500;

INITIALIZE,,100000,Y;

NETWORK;

FIN;

เมื่อได้ตัวแบบรูปภาพ และค่าสั่งควบคุมเรียบร้อยแล้ว ขั้นตอนต่อไปคือการทดสอบตัวแบบ โดยการทำการจำลอง ซึ่งเริ่มจากการแปรตัวแบบที่ได้สร้างขึ้น ให้อยู่ในรูปของค่าสั่งของภาษาการจำลองเช่นก่อน จากนั้น run โปรแกรม หรือตัวแบบที่สร้างขึ้น ผลของการจำลองที่ได้จะออกมามาในรูปแบบของรายงาน ดังในรูปที่ 10

ผลจากการจำลองสรุปได้ว่า

1. จำนวนคนที่รอคิวยโดยเฉลี่ยที่เควคิว (Average Queue Length) เท่ากับ 2.262 คน
2. เวลารอคิวยโดยเฉลี่ย (Average Waiting Time) เท่ากับ 9.064 นาที
3. สัดส่วนของเวลาที่ผู้ให้บริการไม่ง่วง (Utilization Rate) เท่ากับ 0.745

รูปที่ 10 รายงานผลการจำลอง

1

SLAM II SUMMARY REPORT

SIMULATION PROJECT SEMINAR

BY GRIDSANA

DATE 6/18/1997

RUN NUMBER 1 OF 1

CURRENT TIME .1000E+06

STATISTICAL ARRAYS CLEARED AT TIME .0000E+00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1	SERV QUEUE	2.262	3.257	25	1	9.064
2	CALENDAR	1.745	.436	3	2	1.868

SERVICE ACTIVITY STATISTICS

ACT NUM	ACT START	LABEL OR NODE	SER CAP	AVERAGE UTIL	STD DEV	CUR UTIL	AVERAGE BLOCK	MAX IDL TME/SER	MAX BSY TME/SER	ENT CNT
0	SERV	QUEUE	1	.745	.44	1	.00	40.43	539.90	

SS-END

3. สรุป

จะเห็นได้ว่าในการสร้างตัวแบบการจำลองดังกล่าว สามารถทำได้อย่างไม่ยาก กล่าวคือสร้างตัวแบบรูปภาพ สร้างคำสั่งควบคุม ซึ่งใช้เพียงไม่กี่คำสั่งก็สามารถจำลองระบบได้ และถ้าจะเปรียบเทียบถึงเวลาที่ต้องใช้ในการเขียนโปรแกรมด้วยภาษาการจำลอง กับภาษาหารือทั่วไปจะเห็นว่าแตกต่างกันมาก ด้วยเหตุนี้ภาษาการจำลองจึงเป็นที่นิยมของผู้ที่ต้องใช้ตัวแบบในการ วิเคราะห์ระบบงานต่างๆ ในยุคปัจจุบัน

บรรณานุกรม

- Pritsker, A. Alan B. 1994. **Introduction to Simulation and SLAM II.** Fourth Edition, New York: John Wiley & Sons.
- Avrill, M. Law, Kelton, W. David 1991. **Simulation Modeling & Analysis.** Second Edition, New York: McGraw-Hill.