

ระบบแบบจำลองในกระบวนการผลิต  
Simulation System for Manufacturing Process



\*H001889\*



โดย

นายปริญญา งามเกียรติขจร

รหัส 43067075

อาจารย์ที่ปรึกษา

ดร.จันทร์บูรณ์ สถิตวิริยวงศ์

วัน เดือน ปี.....	10 ส.ค. 2550
เลขทะเบียน.....	01889
เลขเรียกหนังสือ.....	คทท 4565 2544
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาคเรียนที่ 2 ปีการศึกษา 2544  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

11/2/11

หัวข้อเรื่อง	ระบบแบบจำลองในกระบวนการผลิต
นักศึกษา	นายปริญญา งามเกียรติขจร
อาจารย์ที่ปรึกษา	ดร.จันทบูรณ์ สติตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

### บทคัดย่อ

ระบบแบบจำลองในกระบวนการผลิตเป็นระบบที่จะช่วยอำนวยความสะดวกในการออกแบบ และการทดสอบการทำงานของกระบวนการผลิต เพื่อลดเวลา และค่าใช้จ่ายในการปรับเปลี่ยนรูปแบบการทำงานให้มีประสิทธิภาพมากขึ้น รายงานฉบับนี้เป็นการนำเสนอการพัฒนากระบวนการแบบจำลองในกระบวนการผลิตเพื่อช่วยในการสร้างแบบจำลอง การปรับเปลี่ยนรูปแบบการทำงานของแบบจำลอง รวมทั้งการใช้งานแบบจำลองให้มีความสะดวกรวดเร็ว และสามารถประเมินสมรรถนะการทำงานของระบบได้ โดยอาศัยภาพกราฟฟิกในการติดต่อกับผู้ใช้งาน และใช้ CSIM เป็น Simulation Library ในการพัฒนาส่วนประมวลผลของโปรแกรม

Title	Simulation System for Manufacturing Process
Student	Mr.Parinya Ngamkiatkhajhorn
Advisor	Chanboon Sathitwiriawong, Ph. D.
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2001

## ABSTRACT

Simulation for manufacturing process is the "simulation model" used to facilitate the design and testing of manufacturing processes. It could save time and money spent on design or improvement of the manufacturing processes. This report presents the development of the simulation for manufacturing process, starting from building the simulation model, modifying the existing model, and using the model in the real application. The model uses graphic user interface so it is easy and more convenient to evaluate the performance of the system. The development of the simulation model uses CSIM as the simulation library for the simulation program.

## กิตติกรรมประกาศ

การที่โครงการนี้ประสบความสำเร็จไปด้วยดีได้นั้น ผู้เขียนต้องขอขอบคุณบุคคลต่าง ๆ ที่มีส่วนในการสนับสนุนให้การจัดทำโครงการดังกล่าวสำเร็จตามวัตถุประสงค์ที่ต้องการ ได้แก่

1. คุณพ่อ คุณแม่ ผู้ซึ่งให้ความสำคัญกับการศึกษาของลูก ๆ ทุกคน รวมทั้งคนในครอบครัวที่คอยเป็นกำลังใจที่สำคัญตลอดมา
2. ดร.จันทร์บุรณ สติตวิริยวงศ์ อาจารย์ที่ปรึกษาโครงการ ผู้ให้คำปรึกษาและชี้แนะทางในการจัดทำโครงการ รวมทั้งให้ความช่วยเหลือตลอดเวลาในการศึกษา
3. ดร.อาริต ธรรมโน อาจารย์ที่ปรึกษา ผู้ให้คำแนะนำและช่วยเหลือในการศึกษาเป็นอย่างดี

นายปริญญา งามเกียรติขจร

ผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่	
1. บทนำ.....	1
1.1 วัตถุประสงค์ของการพัฒนาโครงการ.....	1
1.2 แนวทางการพัฒนา.....	1
1.3 เครื่องมือที่ใช้ในการพัฒนา.....	1
1.4 ขอบเขตการพัฒนา.....	2
2. ความรู้พื้นฐานในการพัฒนาระบบ.....	3
2.1 กระบวนการผลิต.....	3
2.2 เทคนิคในการจำลองแบบปัญหา.....	3
2.3 พื้นฐานเกี่ยวกับฟังก์ชันวินโดวส์ API-32 บิต.....	6
2.4 พื้นฐานเกี่ยวกับไลบรารี CSIM.....	13
3. การออกแบบระบบ.....	15
3.1 การออกแบบการทำงานของส่วน Model.....	24
3.2 การออกแบบการทำงานของส่วน Simulation.....	27
3.3 การออกแบบการทำงานของส่วน Report.....	30
4. การพัฒนาระบบแบบจำลองในกระบวนการผลิต.....	31
4.1 เครื่องมือที่ใช้ในการพัฒนาระบบ.....	31
4.2 การพัฒนาส่วน Model.....	31
4.3 การพัฒนาส่วน Simulation.....	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
4.4 การพัฒนาส่วน Report.....	41
5. การทดสอบระบบ.....	44
5.1 วิธีการทดลอง.....	44
5.2 ผลการทดลอง.....	45
6. สรุป.....	50
6.1 สรุปผลการพัฒนาโครงการ.....	50
6.2 ข้อควรปรับปรุงเพิ่มเติม.....	50
6.3 ปัญหาและข้อเสนอแนะ.....	51
7. บรรณานุกรม.....	52



## สารบัญตาราง

ตารางที่	หน้า
2.1 ไฟล์ไลบรารีของวิน โควส์ 32 บิต.....	7
2.2 ชนิดของข้อมูลสำหรับภาษา C.....	8
2.3 การประกาศพารามิเตอร์ของข้อมูลชนิดตัวเลขขนาด 8 บิต.....	9
2.4 การประกาศพารามิเตอร์ของข้อมูลชนิดตัวเลขขนาด 16 บิต.....	10
2.5 การประกาศพารามิเตอร์ของข้อมูลชนิดตัวเลขขนาด 32 บิต.....	11
2.6 การประกาศฟังก์ชันวิน โควส์ API.....	12
2.7 ประเภทของการประกาศตัวแปร.....	14
2.8 ตัวแปรของ CSIM Library.....	14
3.1 จำนวนองค์ประกอบในการเชื่อมต่อ.....	25
3.2 เส้นใยในการเชื่อมต่อ.....	25
3.3 ข้อมูลเพิ่มเติมสำหรับแต่ละองค์ประกอบ.....	26

## สารบัญภาพ

ภาพที่	หน้า
2.1 กราฟของการกระจายแบบสมมาตร	4
2.2 กราฟของการกระจายแบบเบ้ขวา	5
2.3 กราฟของการกระจายแบบเอกโพเนนเชียล	6
2.4 กราฟของการกระจายแบบนอร์มอล	7
3.1 Context Diagram ของระบบแบบจำลองในกระบวนการผลิต	15
3.2 DFD Level 1 ของระบบแบบจำลองในกระบวนการผลิต	17
3.3 DFD Level 2 การออกแบบ Model	18
3.4 DFD Level 2 การประมวลผลแบบจำลอง	19
3.5 DFD Level 2 การออกแบบ Report	20
3.6 การทำงานขององค์ประกอบต่าง ๆ ในแบบจำลอง	29
4.1 Menu ที่ใช้สำหรับเพิ่มองค์ประกอบให้กับตัวแบบจำลอง	32
4.2 Toolbar ที่ใช้สำหรับเพิ่มองค์ประกอบให้กับตัวแบบจำลอง	33
4.3 รูปแบบการเชื่อมต่อขององค์ประกอบ	34
4.4 รายละเอียดขององค์ประกอบ	35
4.5 แสดงคุณสมบัติขององค์ประกอบประเภท Generator	36
4.6 คุณสมบัติขององค์ประกอบประเภท Server	36
4.7 แสดงคุณสมบัติขององค์ประกอบประเภท Queue	37
4.8 แสดงคุณสมบัติ Predecessor ขององค์ประกอบประเภท Assembly Operation	37
4.9 แสดงคุณสมบัติเพิ่มเติมขององค์ประกอบประเภท Assembly Operation	38
4.10 แสดงคุณสมบัติขององค์ประกอบประเภท Split Point	38
4.11 แสดงคุณสมบัติเพิ่มเติมขององค์ประกอบประเภท Split Point	39
4.12 กำหนดระยะเวลาในการประมวลผล	39
4.13 ข้อมูลในการประมวลผล	40
4.14 รายงานจากการประมวลผล	43
5.1 แบบจำลองในการทดสอบ	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา VII ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
5.2	สรุปการทำงานของ Assembly ในระบบที่ 1..... 45
5.3	Mean Waiting Time of Parts in Queue 2003..... 46
5.4	Mean Waiting Time of Parts in Queue 2004..... 47
5.5	Mean Number of Parts in Queue 2003..... 47
5.6	Mean Number of Parts in Queue 2004..... 48
5.7	สรุปการทำงานของ Assembly ในระบบที่ 2..... 49
5.8	เปรียบเทียบการทำงานของ Assembly ในระบบที่ 1 และระบบที่ 2..... 49



# บทที่ 1

## บทนำ

กระบวนการผลิตเป็นกระบวนการพื้นฐานในภาคอุตสาหกรรม ซึ่งจะมีขั้นตอนและวิธีการทำงานแตกต่างกันไปตามรูปแบบการผลิตและขนาดของระบบงาน ปัจจุบันเทคโนโลยีมีความก้าวหน้ามากและตลาดมีการแข่งขันสูง แต่ละบริษัทจึงจะต้องมีการพัฒนาปรับปรุงกระบวนการผลิตเพื่อให้มีประสิทธิภาพมากขึ้นสามารถแข่งขันในตลาดได้ แต่การปรับปรุงหรือขยายระบบงานมักจะมีค่าใช้จ่ายมาก รวมทั้งยังต้องใช้เวลาในการเปลี่ยนแปลงระบบงานและอาจมีผลกระทบต่อระบบงานเดิม ทำให้เกิดความเสี่ยงที่จะไม่ประสบความสำเร็จ ดังนั้นจึงมีแนวความคิดในการพัฒนาระบบแบบจำลองในกระบวนการผลิตที่มีความยืดหยุ่นสามารถปรับเปลี่ยนรูปแบบของระบบงานได้ตามความต้องการ

### 1.1 วัตถุประสงค์ของการพัฒนาโครงการ

เพื่อเป็นการพัฒนาเครื่องมือที่ช่วยในการสร้างและทดสอบประสิทธิภาพแบบจำลองที่ใช้ในกระบวนการผลิต เพื่อทำการวิเคราะห์เปรียบเทียบรูปแบบของระบบงานในแบบต่างๆ ทำให้สามารถประเมินประสิทธิภาพของระบบงาน โดยไม่ต้องติดตั้งใช้งานกับระบบงานจริง

### 1.2 แนวทางการพัฒนา

แบ่งการพัฒนาโปรแกรมออกเป็น 2 ส่วนคือ

1. พัฒนาในส่วนติดต่อกับผู้ใช้งาน โดยทำหน้าที่ในการอำนวยความสะดวกในการสร้างรูปแบบของระบบงานในลักษณะของแผนภูมิรูปภาพ และแสดงรายงานผลของการจำลองการทำงานของระบบงาน รวมทั้งการบันทึก การพิมพ์ และการเรียกใช้รูปแบบของระบบงาน

2. พัฒนาในส่วนการประมวลผลแบบจำลอง ซึ่งทำหน้าที่ทดลองการทำงานของแบบจำลอง โดยจะนำข้อมูลของแบบจำลองที่ได้ออกแบบไว้และเงื่อนไขในการทดลองมาประมวลผล จากนั้นจะส่งผลลัพธ์ที่ได้ส่งกลับไปยังส่วนแสดงผล เพื่อรายงานการทำงานของส่วนต่าง ๆ ในแบบจำลอง

### 1.3 เครื่องมือที่ใช้ในการพัฒนา

เครื่องมือที่ใช้ในการพัฒนา Simulation System of Manufacturing Process นั้นประกอบด้วย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Microsoft Visual Basic 6.0 ใช้สำหรับพัฒนาในส่วนติดต่อกับผู้ใช้งาน ที่ทำหน้าที่ในการสร้างแบบจำลอง การเรียกใช้งานส่วนประมวลผลแบบจำลอง และรายงานผลลัพธ์จากการประมวลผลแบบจำลอง
2. Microsoft Visual C++ 6.0 ใช้สำหรับพัฒนาในส่วนการประมวลผลแบบจำลอง เนื่องจากมีการเรียกใช้งาน Simulation Library ซึ่งเขียนจากภาษา C โดยพัฒนาในรูปแบบของ DLL (Dynamic-Link Library) เพื่อให้สามารถเรียกใช้งานได้จากโปรแกรมอื่น ๆ ในลักษณะของ วินโดวส์ API
3. CSIM ซึ่งเป็น Simulation Library ที่มีประสิทธิภาพ ทำให้สามารถเรียกใช้งาน Function การทำงานในการทำ Simulation โดยไม่ต้องเขียนโปรแกรมในการทำงานมากนัก
4. เครื่อง Computer

#### 1.4 ขอบเขตการพัฒนา

ส่วนเงื่อนไขและขอบเขตของการพัฒนานั้น ได้กำหนดขอบเขตของการพัฒนาไว้เป็นดังนี้

1. พัฒนาโปรแกรมโดยใช้ Simulation Library ของ CSIM
2. พัฒนาในส่วนติดต่อกับผู้ใช้งานให้สามารถสร้างแบบจำลองได้ง่ายโดยใช้ GUI
3. พัฒนาการประมวลผลแบบจำลองให้ใช้งานกับกระบวนการผลิตที่มีผลิตภัณฑ์เดียว และมีทิศทางเดียว เพื่อให้ระบบไม่ซับซ้อนเกินไป
4. กำหนดจำนวนการเชื่อมต่อในแต่ละส่วนไม่เกิน 5 เส้นทาง เพื่อให้ดูแลจัดการกับแบบจำลองได้ง่าย
5. ใช้งานบนเครื่องคอมพิวเตอร์แบบ Standalone และมีระบบปฏิบัติการ Microsoft windows 95/98 หรือสูงกว่า

## บทที่ 2

### ความรู้พื้นฐานในการพัฒนาระบบ

เนื่องจากการพัฒนานั้นประกอบด้วย การจองแบบปัญหา และจำเป็นต้องใช้ Windows API โดยใช้ Microsoft Visual Basic ดังนั้นจึงจำเป็นต้องศึกษาความรู้พื้นฐานที่เกี่ยวข้องก่อนเพื่อประโยชน์ในการพัฒนาระบบให้มีประสิทธิภาพมากขึ้น

#### 2.1 กระบวนการผลิต

การกำหนดงานผลิตขึ้นอยู่กับชนิดของกระบวนการผลิต โดยทั่วไปการกำหนดงานผลิตตามกระบวนการผลิตแบ่งออกได้ 3 ประเภท ดังนี้

1. การกำหนดงานผลิตที่มีผลิตภัณฑ์ชนิดเดียวหรือไม่กี่ชนิด มีกระบวนการผลิตที่ต่อเนื่องกันไป เช่น โรงงานผลิตน้ำตาล โรงงานผลิตน้ำอัดลม โรงงานปิโตรเคมี เป็นต้น เครื่องจักรที่นำมาใช้ในการผลิตส่วนมากเป็นเครื่องจักรเอกประสงค์ และมีการจัดลำดับการผลิตตายตัวแล้วจึงไม่ต้องจัดลำดับการผลิตใหม่อีก สำหรับการกำหนดงานผลิตสามารถทำได้ง่ายกว่ากระบวนการผลิตประเภทอื่น
2. การกำหนดงานผลิตตามกระบวนการผลิตแบบไม่ต่อเนื่อง หมายถึง การกำหนดงานผลิตสำหรับกระบวนการผลิตที่มีผลิตภัณฑ์หลายชนิด แต่ละชนิดมีจำนวนการผลิตไม่เท่ากันและไม่แน่นอน โดยมีจำนวนการผลิตครั้งละไม่มาก เครื่องจักรที่ใช้เป็นแบบเอกประสงค์ โรงงานประเภทนี้มีอยู่จำนวนมาก เช่น โรงกลึง โรงพิมพ์ เป็นต้น งานแต่ละชนิดที่เข้ามาจะมีขั้นตอนการทำงานที่แตกต่างกัน ดังนั้นน้อยสุด กระบวนการผลิตแบบนี้บางครั้งเรียกว่ากระบวนการผลิตแบบสั่งทำ
3. การกำหนดงานผลิตที่มีผลิตภัณฑ์ชนิดเดียวกันหรือไม่กี่ชนิด มีลักษณะการผลิตเป็นสายงานผลิต และใช้แรงงานมาก เครื่องจักรที่ใช้มีทั้งแบบเอกประสงค์และเอกประสงค์ เช่น โรงงานประกอบรถยนต์ การกำหนดงานผลิตจะเน้นไปที่การจัดสมดุลของสายงานผลิตเพื่อลดเวลาสูญเสียเปล่าอันเกิดจากรองาน การล่าช้าของงานในสายงานผลิตให้มีค่าน้อยสุดเท่าที่จะเป็นไปได้

#### 2.2 เทคนิคในการจองแบบปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแบบจำลองที่สามารถแทนองค์ประกอบและความสัมพันธ์ระหว่างองค์ประกอบด้วยค่าเชิงปริมาณนั้น โดยปกติข้อมูลเชิงปริมาณมักจะมีค่าไม่แน่นอนตายตัว ดังนั้นจึงเป็นปัญหาว่าในการคำนวณนั้นเราจะใช้ค่าใดจึงจะทำให้ผลลัพธ์ที่ได้จากแบบจำลองคล้ายคลึงใกล้เคียงกับผลลัพธ์ที่ได้จากระบบงานจริง เทคนิคที่ใช้แปลงตัวแปรแบบสุ่มที่มีลักษณะการกระจายของความน่าจะเป็นแบบสม่ำเสมอไปอยู่ในรูปของตัวแปรแบบสุ่มที่มีลักษณะการกระจายของความน่าจะเป็นแบบอื่นๆ โดยที่ตัวเลขทุกตัวมีโอกาสถูกเลือกได้เท่าๆกัน ซึ่งจะมีการกระจายแบบต่างๆดังนี้ [4]

**2.2.1 การกระจายแบบสม่ำเสมอ (Uniform Distribution)**

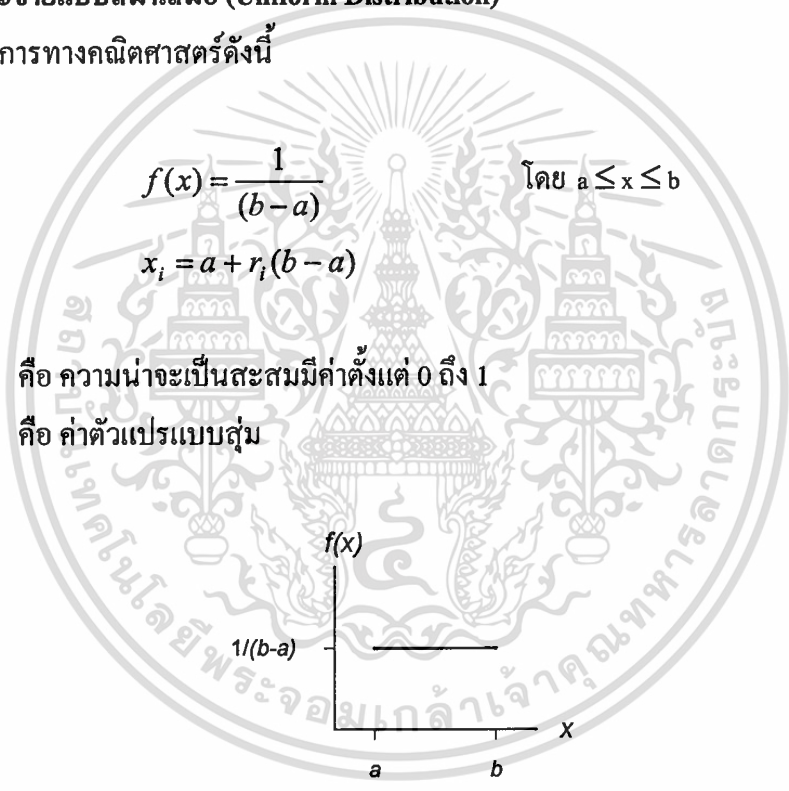
มีสมการทางคณิตศาสตร์ดังนี้

$$f(x) = \frac{1}{(b-a)} \quad \text{โดย } a \leq x \leq b$$

$$x_i = a + r_i(b-a)$$

เมื่อ  $r$  คือ ความน่าจะเป็นสะสมมีค่าตั้งแต่ 0 ถึง 1

$x$  คือ ค่าตัวแปรแบบสุ่ม



ภาพที่ 2.1 กราฟของการกระจายแบบสม่ำเสมอ

**2.2.2 การกระจายแบบปัวซอง (Poisson Distribution)**

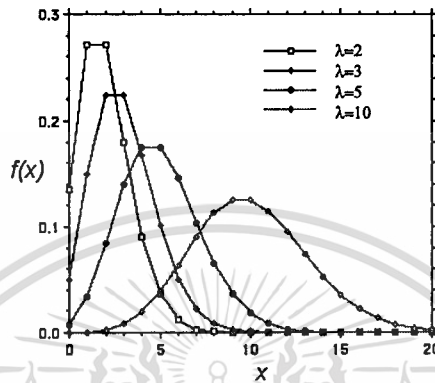
มีสมการทางคณิตศาสตร์ดังนี้

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad \text{โดย } x \geq 0$$

$$\prod_{i=0}^x r_i \geq e^{-\lambda} \geq \prod_{i=0}^{x+1} r_i$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ  $\lambda$  คือ อัตราการเกิดเฉลี่ย  
 $x$  คือ ค่าตัวแปรแบบสุ่ม



ภาพที่ 2.2 กราฟของการกระจายแบบบิวซ์ของ

### 2.2.3 การกระจายแบบเอกโปเนนเชียล (Exponential Distribution)

มีสมการทางคณิตศาสตร์ดังนี้

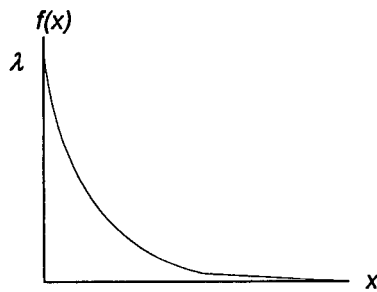
$$f(x) = \lambda e^{-\lambda x}$$

โดย  $x \geq 0$

$$x_i = -\theta \ln(r_i)$$

โดย  $\theta = \frac{1}{\lambda}$

เมื่อ  $r$  คือ ความน่าจะเป็นสะสมมีค่าตั้งแต่ 0 ถึง 1  
 $\theta$  คือ ค่าความแปรปรวน  
 $\lambda$  คือ อัตราการเกิดเฉลี่ย  
 $x$  คือ ค่าตัวแปรแบบสุ่ม



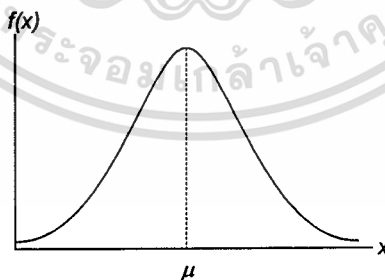
ภาพที่ 2.3 กราฟของการกระจายแบบเอกโปเนนเชียล

### 2.2.4 การกระจายแบบนอร์มอล (Normal Distribution)

มีสมการทางคณิตศาสตร์ดังนี้

$$f(x) = \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \quad \text{โดย} \quad -\infty < x < \infty$$

เมื่อ  $x$  คือ ค่าตัวแปรแบบสุ่ม  
 $\mu$  คือ ค่าเฉลี่ย  
 $\sigma$  คือ ค่าของส่วนเบี่ยงเบนมาตรฐาน



ภาพที่ 2.4 กราฟของการกระจายแบบนอร์มอล

### 2.3 พื้นฐานเกี่ยวกับฟังก์ชันวินโดวส์ API-32 บิต

ฟังก์ชันวินโดวส์ API หมายถึง ฟังก์ชันงานทั้งหมดที่อยู่ในไฟล์ไลบรารี .dll ของวินโดวส์ เช่น User32.dll Gdi32.dll Kernel32.dll เป็นต้น ซึ่งไฟล์ไลบรารี .dll เหล่านี้จะเป็นแกนหลักของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบปฏิบัติการวินโดวส์ เพราะการทำงานของวินโดวส์ จะอาศัยความสามารถของฟังก์ชันที่ปรากฏในไฟล์ไลบรารี .dll

### 2.3.1 ไฟล์ไลบรารี .dll ของวินโดวส์ 32 บิต

เป็นไฟล์ที่โดยปกติจะถูกกำหนดให้มีนามสกุล .dll และจะถูกจัดเก็บเอาไว้ใน Directory ย่อย \System ของวินโดวส์ และการทำงานของวินโดวส์ก็คือองอาศัยฟังก์ชันวินโดวส์ API ที่มีในแต่ละไฟล์ไลบรารี .dll นั้นเอง สำหรับวินโดวส์ 32 บิต (วินโดวส์ 98) สนับสนุนฟังก์ชันวินโดวส์ API มาตรฐานมากกว่า 1,500 ฟังก์ชัน โดยที่จะแยกจัดเก็บไว้ในไฟล์ไลบรารีที่ถือได้ว่าเป็นแกนหลักของวินโดวส์ 32 บิต ดังต่อไปนี้

ไฟล์ .dll	รายละเอียด
Gdi32.dll	ไลบรารีสำหรับการติดต่อสื่อสารกับดีไวซ์กราฟิก (Graphic Device Interface) ซึ่งประกอบด้วยฟังก์ชันต่างๆสำหรับงานด้าน อุปกรณ์แสดงผล การวาดกราฟิก ไฟล์เมต้าฟอนต์ (.wmf) และการกำหนดตำแหน่งโคออร์ดิเนต
Comdlg32.dll	ไลบรารีสำหรับสนับสนุนการสร้างและแสดงผล Dialog box ซึ่ง Visual Basic 6.0 ก็อาศัยความสามารถจากไลบรารีดังกล่าวนี้ ร่วมกับคอนโทรล Comdlg32.ocx ในการแสดงผล Dialog box มาตรฐาน เช่น Open / Save As หรือ Font เป็นต้น และสำหรับการปรับแต่ง Form สำหรับ Visual Basic 6.0
Version.dll	ไลบรารีสำหรับการสนับสนุนการควบคุม และอ่านหมายเลขเวอร์ชันจากไฟล์ .exe .dll .ocx ฯลฯ ซึ่งจะมีประโยชน์อย่างมากในขณะทำการติดตั้งไฟล์ต่างๆเนื่องจากเวอร์ชันของไฟล์เพื่อป้องกันความไม่เข้ากันของ Application และคอนโทรล ActiveX หรือไฟล์ไลบรารีต่างๆ
Dllz32.dll	ไลบรารีสำหรับการสนับสนุนการขยายขนาดของไฟล์ที่ถูกบีบอัด อย่างเช่น ในการสร้างแผ่น setup นั้น ไฟล์ต่างๆจะถูกบีบอัดด้วยโปรแกรม compress.exe ซึ่งโดยปกติไฟล์ที่ถูกบีบอัดก็จะมีนามสกุลตัวสุดท้ายเป็นเครื่องหมาย underscore( ) โดยขณะทำการติดตั้ง ไฟล์เหล่านี้ก็จะถูกขยายขนาดอัตโนมัติโดยวินโดวส์ ซึ่งจะอาศัยฟังก์ชันวินโดวส์ API จากไฟล์ไลบรารี นั้นเอง
Kernel32.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการติดต่อระดับล่าง (low-level) กับระบบปฏิบัติการ การจัดการงาน การควบคุมทรัพยากรของระบบ และ การจัดการระบบการใช้งานหน่วยความจำ เป็นต้น

User32.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการจัดการกับเมนู Cursor,Caret การจัดการกับเวลาของระบบ การจัดการกับการส่ง message ระหว่าง Application หรือระบบปฏิบัติการ เป็นต้น
Netapi32.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการจัดการด้านระบบเครือข่าย
Odbc32.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการควบคุมการติดต่อสื่อสารระหว่างระบบฐานข้อมูล ที่รูปแบบต่างๆกัน โดยอาศัยเทคโนโลยีที่เรียกว่า Open Database Connectivity (ODBC) ซึ่งจะช่วยให้การแลกเปลี่ยนข้อมูลระหว่างฐานข้อมูล ที่ถูกจัดการ โดยเซิร์ฟเวอร์ต่างๆกัน เป็นไปได้โดยสะดวกและง่ายมากขึ้น
Winmm.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการจัดการด้านมัลติมีเดีย เช่น การจัดการหรือควบคุมการเล่นกับไฟล์ .wav .mid หรือแม้กระทั่งการจัดการกับ CD-Audio เป็นต้น

### ตารางที่ 2.1 ไฟล์ไลบรารีของวินโดวส์ 32 บิต

#### 2.3.2 การเขียนฟังก์ชันวินโดวส์ API ด้วย Microsoft Visual Basic

เนื่องจากฟังก์ชันวินโดวส์ API ส่วนใหญ่จะถูกเขียนขึ้นมาเพื่อสนับสนุนภาษา C หรือ Microsoft Visual C++ ดังนั้นชนิดของข้อมูลบางส่วนจึงไม่ Compatible กับชนิดข้อมูลของ Microsoft Visual Basic 6.0 ซึ่งในการส่งผ่านตัวแปรให้กับแต่ละพารามิเตอร์ของฟังก์ชันวินโดวส์ API จึงต้องมีการตรวจสอบความเข้ากันได้ของชนิดข้อมูลเสียก่อนทุกครั้ง และความหมายของชนิดข้อมูลสำหรับภาษา C หรือ Microsoft Visual C++ ดังต่อไปนี้

ชนิดข้อมูล	รายละเอียด
BOOL	ข้อมูลชนิด Boolean ขนาด 32 บิต โดยที่ 0 หมายถึง False และค่าไม่เท่ากับ 0 หมายถึง True
BYTE	ข้อมูลชนิดไบต์ ขนาด 8 บิต
Char	ข้อมูลชนิดตัวเลขเต็ม Unsigned ขนาด 8 บิต
FARPROC	ข้อมูลชนิด Pointer ไปยัง Procedure หรือฟังก์ชัน ขนาด 32 บิต
HANDLE	ข้อมูลชนิดเลขจำนวนเต็ม Unsigned ขนาด 32 บิต เพื่อเป็น handle ไปยัง Object
Int	ข้อมูลชนิดเลขจำนวนเต็ม Signed ขนาด 32 บิต
Short	ข้อมูลชนิดเลขจำนวนเต็ม Signed ขนาด 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ บริษัท เทคโนโลยี จำกัด ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Long	ข้อมูลชนิดเลขจำนวนเต็ม Signed ขนาด 32 บิต
LPINT	ข้อมูลชนิด Pointer ขนาด 32 บิต ซึ่งไปยังข้อมูลชนิดเลขจำนวนเต็ม signed ขนาด 32 บิต
LPSTR	ข้อมูลชนิด Pointer ขนาด 32 บิต ซึ่งไปยังข้อมูลชนิดสตริงชนิด ASCIIZ
UINT	ข้อมูลชนิดเลขจำนวนเต็ม Unsigned ขนาด 32 บิต
WORD	ข้อมูลชนิดเลขจำนวนเต็ม Unsigned ขนาด 16 บิต
DWORD	ข้อมูลชนิดเลขจำนวนเต็ม Unsigned ขนาด 32 บิต
Flags	ข้อมูลชนิด Flag ขนาด 32 บิต
Float	ข้อมูลชนิดเลขจำนวนจริง ขนาด 32 บิต
Double	ข้อมูลชนิดเลขจำนวนจริง ขนาด 32 บิต
LPUNKNOWN	Pointer ขนาด 32 บิต ไปยัง Interface IUnknown OLE 2.0

## ตารางที่ 2.2 ชนิดของข้อมูลสำหรับภาษา C

จากข้างต้นจะเห็นว่าชนิดข้อมูลของภาษา C หรือ Visual C++ นั้นมีความหลากหลายมาก โดยเฉพาะชนิดชนิดตัวเลขเพราะภาษา C สนับสนุนทั้งข้อมูลตัวเลขที่มีเครื่องหมายและไม่มีเครื่องหมาย แต่สำหรับ Visual Basic 6.0 จะสนับสนุนเฉพาะข้อมูลตัวเลขที่มีเครื่องหมายเท่านั้น ยกเว้นข้อมูลชนิด byte ของ Visual Basic 6.0 เท่านั้น ที่สนับสนุนข้อมูลชนิดไม่มีเครื่องหมาย ดังนั้นเพื่อให้สามารถเข้าใจถึงการเปรียบเทียบชนิดข้อมูลระหว่างภาษาซี และ Visual Basic 6.0 ได้ง่ายขึ้น ก็สามารถพิจารณาได้ดังนี้

1.5 ข้อมูลชนิดตัวเลขขนาด 8 บิต ข้อมูลชนิดตัวเลขขนาด 8 บิต สำหรับภาษาซีนั้นมีอยู่ด้วยกัน 2 ประเภท ดังนี้ Char และ BYTE ซึ่งจะ Compatible กับข้อมูลชนิด byte ของ Visual Basic 6.0 โดยจะต้องมีการประกาศให้กับ พารามิเตอร์ ดังนี้

C, C++	Visual Basic 6.0
Char posval	By Val posval As Byte
BYTE posval	By Val posval As Byte

## ตารางที่ 2.3 การประกาศพารามิเตอร์ของข้อมูลชนิดตัวเลขขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับข้อมูลชนิด Char นั้นสามารถเก็บค่าได้ตั้งแต่ -128-ถึง 127-แต่ข้อมูลชนิด byte ของ Visual Basic 6.0 เป็นข้อมูลชนิดตัวเลขจำนวนเต็ม unsigned ดังนั้นถ้าหากค่าของข้อมูลชนิด Char มีค่าเป็นลบ ต้องทำการเขียน Code เพื่อแปลงค่าของข้อมูลเอาเอง ส่วนข้อมูลชนิด BYTE นั้น Compatible กับข้อมูลชนิด Byte ของ Visual Basic 6.0 แต่เนื่องจากวินโดวส์ 32 บิต เน้นหนักด้านระบบปฏิบัติการ 32 บิต ดังนั้นข้อมูลชนิด char และ BYTE มักจะ ไม่มีการใช้งานในฟังก์ชัน วินโดวส์ API

1.6 ข้อมูลชนิดตัวเลขขนาด 16 บิต สำหรับภาษาซีนั้นมีอยู่ด้วยกัน 2 ประเภท ดังนี้ short และ WORD ซึ่งจะ Compatible กับข้อมูลชนิด integer ของ Visual Basic 6.0 โดยจะต้องมีการประกาศให้กับพารามิเตอร์ ดังนี้

C, C++	Visual Basic 6.0
Short posval	ByVal posval As Integer
WORD posval	ByVal posval As Integer

#### ตารางที่ 2.4 การประกาศพารามิเตอร์ของข้อมูลชนิดตัวเลขขนาด 16 บิต

สำหรับข้อมูลชนิด short นั้นเป็นข้อมูลชนิดตัวเลข signed ขนาด 16 บิต ซึ่งจะเหมือนกับข้อมูลชนิด integer ของ Visual Basic 6.0 ทุกประการ แต่สำหรับข้อมูลชนิด WORD นั้นเป็นข้อมูลชนิดตัวเลข unsigned ขนาด 16 บิต ซึ่งสามารถจัดเก็บค่าตัวเลขได้ตั้งแต่ 0 ถึง 65,535 ดังนั้นจึงไม่เหมือนกับข้อมูลชนิด Integer ของ Visual Basic 6.0 แต่เนื่องจากขนาดของข้อมูลเท่ากัน จึงมีความ Compatible กัน และในการส่งค่าตัวเลขหรืออ่านค่ากลับจากฟังก์ชันวินโดวส์ API สำหรับข้อมูลชนิดตัวเลข unsigned ขนาด 16 บิต ที่มีค่าตั้งแต่ 32,768 ถึง 65,535 จึงต้องอาศัยเทคนิคในการแปลงข้อมูลตัวเลข

1.7 ข้อมูลชนิดตัวเลขขนาด 32 บิต สำหรับภาษาซีนั้นมีอยู่ด้วยกัน 5 ประเภทดังนี้ BOOL DWORD Int UINT และ LONG ซึ่งจะ Compatible กับข้อมูลชนิด long ของ Visual Basic 6.0 โดยจะต้องมีการประกาศให้กับพารามิเตอร์ ดังนี้

C, C++	Visual Basic 6.0
BOOL posval	ByVal posval As Long
DWORD posval	ByVal posval As Long

Int posval	ByVal posval As Long
LONG posval	ByVal posval As Long
UINT posval	ByVal posval As Long

## ตารางที่ 2.5 การประกาศพารามิเตอร์ของข้อมูลชนิดตัวเลขขนาด 32 บิต

ข้อมูลชนิด Int และ LONG นั้นเป็นข้อมูลชนิดตัวเลข Signed ขนาด 32 บิต ซึ่งจะเหมือนกับข้อมูลชนิด Long ของ Visual Basic 6.0 ทุกประการ แต่สำหรับข้อมูลชนิด DWORD และ UINT นั้นเป็นข้อมูลชนิดตัวเลข Unsigned ขนาด 32 บิต และสามารถจัดเก็บค่าตัวเลขได้ตั้งแต่ 0 ถึง 4,294,967,295 ดังนั้นจึงไม่เหมือนกับข้อมูลชนิด Long ของ Visual Basic 6.0 แต่เนื่องจากขนาดของข้อมูลเท่ากัน จึงมีความ Compatible กัน และในการส่งค่าตัวเลขหรืออ่านค่ากลับจากฟังก์ชัน วินโดวส์ API สำหรับข้อมูลชนิดตัวเลข Unsigned ขนาด 32 บิต ที่มีค่าตั้งแต่ 2,147,483,647 ถึง 4,294,967,295 นั้นจึงต้องมีเทคนิคในการแปลงข้อมูล

1.8 ข้อมูลชนิดบูลีน (BOOL) นั้น สำหรับ Visual Basic 6.0 จะมีอยู่เพียง 2 ความหมายคือ 0 หมายถึง False ส่วนค่าอื่นๆ ที่ไม่เท่ากับ 0 หรือโดยปกติค่าของ -1 จะหมายถึง True แต่สำหรับภาษาซีมีข้อแตกต่างกับ Visual Basic 6.0 ตรงที่ค่าของ True นั้นจะเท่ากับ 1 ส่วนค่าของ False นั้นจะเท่ากับ 0

1.9 ข้อมูลชนิดตัวเลขจำนวนจริงขนาด 32 บิต สำหรับภาษาซีนั้นมีอยู่ด้วยกันเพียงประการเดียว คือ Float ซึ่งจะเหมือนกับข้อมูลชนิด Single ของ Visual Basic 6.0 ทุกประการ ดังนั้นจึงไม่จำเป็นต้องมีการแปลงข้อมูลที่ได้

1.10 ข้อมูลชนิดตัวเลขจำนวนจริงขนาด 64 บิต สำหรับภาษาซีนั้นมีอยู่ด้วยกันเพียงประการเดียว คือ Double ซึ่งจะเหมือนกับข้อมูลชนิด Double ของ Visual Basic 6.0 ทุกประการ ดังนั้นจึงไม่จำเป็นต้องมีการแปลงข้อมูลที่ได้ และสามารถประกาศให้กับพารามิเตอร์

1.11 ข้อมูลชนิด Handle สำหรับวินโดวส์ 32 บิต จะมีขนาดเท่ากับ 32 บิต สำหรับใช้กำหนดหรืออ้างอิง (Reference) ชนิดของ Object ของวินโดวส์ ดังนั้นถ้าหากมีการกำหนดตัวแปรสำหรับรับหรือส่งข้อมูลชนิด Handle นี้ ก็ต้องใช้ข้อมูลชนิด Long ของ Visual Basic 6.0

1.12 ข้อมูลชนิดสตริง (String) ที่สนับสนุนโดย C หรือ C++ นั้นเป็น null-terminated ซึ่งแตกต่างจากสตริงของ Visual Basic 6.0 โดยสิ้นเชิง ดังนั้น จึงไม่สามารถส่งผ่านสตริงของ Visual Basic 6.0 ให้กับฟังก์ชันวินโดวส์ API โดยตรง แต่ต้องการแปลงสตริงของ Visual Basic 6.0 ให้อยู่ในรูปแบบของ null-terminated

### 2.3.3 การประกาศฟังก์ชันวินโดวส์ API

ก่อนที่จะสามารถใช้งานฟังก์ชันวินโดวส์ API ในการเขียนโปรแกรม จะต้องทำการประกาศฟังก์ชันนั้นๆเสียก่อน เพื่อเป็นการกำหนด Prototype ในการติดต่อระหว่างฟังก์ชันวินโดวส์ API กับ Visual Basic 6.0 สำหรับการประกาศฟังก์ชันวินโดวส์ API ซึ่งมีอยู่ด้วยกัน 2 รูปแบบ คือ การประกาศ Procedure ประเภท Sub Routine และ การประกาศ Procedure ประเภทฟังก์ชัน

#### รูปแบบที่ 1

[Public | Private] Declare Sub *name* Lib "*libname*" [Alias "*aliasname*"][(arglist)]

#### รูปแบบที่ 2

[Public | Private] Declare Function *name* Lib "*libname*" [Alias "*aliasname*"][(arglist)] [As *type*]

Public	กำหนดให้ Procedure ที่ถูกประกาศสามารถถูกเรียกใช้ได้จาก Module หรือ Form อื่นใน Project ปัจจุบัน แต่การประกาศ Procedure ด้วย Public จะต้องทำใน Module มาตรฐาน (.bas) เท่านั้น
Private	กำหนดให้ Procedure ที่ถูกประกาศสามารถถูกเรียกใช้ได้ เฉพาะจาก Module หรือ Form ที่ประกาศ Procedure นี้เท่านั้น
Sub	กำหนดให้ Procedure เป็นชนิด Routine ซึ่งจะไม่มีค่าส่งกลับ
Function	กำหนดให้ Procedure เป็นชนิดฟังก์ชัน ซึ่งจะมีค่าส่งกลับ ดังนั้นจึงสามารถนำค่าที่ได้จากฟังก์ชัน ไปใช้ในลักษณะของตัวแปรต่างๆ ในนิพจน์ (expression) เช่นนิพจน์ การคำนวณ
Name	กำหนดชื่อของ Procedure สำหรับ Visual Basic 6.0 ยอมให้ชื่อของ Procedure ความยาวทั้งหมดเพียง 40 ตัวอักษรเท่านั้น
Lib	กำหนดให้สตริงที่ตาม (" <i>libname</i> ") จะบ่งบอกถึงชื่อของไฟล์ไลบรารี .dll ซึ่งในการประกาศฟังก์ชันวินโดวส์ API จะต้องมีการกำหนดชื่อของไฟล์ไลบรารี .dll ทุกครั้ง
Alias	กำหนดให้ Procedure ชื่อ " <i>aliasname</i> " สามารถถูกเรียกใช้ได้โดยใช้ชื่ออื่นๆ ตามที่กำหนดโดย name ทั้งนี้เนื่องจากในบางกรณีชื่อของฟังก์ชันวินโดวส์ API นั้นอาจจะซ้ำกับประโยคหรือคำสั่งของ Visual Basic 6.0 หรืออาจจะประกอบด้วยตัวอักษรที่ Visual Basic 6.0 ไม่ยอมรับในการกำหนดเป็นชื่อของ Procedure

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arglist	รายการของพารามิเตอร์ที่จะส่งให้กับ Procedure
ByVal	กำหนดให้พารามิเตอร์มีการส่งค่าแบบ passed by value แต่ถ้าหากนำไปใช้ประกาศฟังก์ชันวินโดวส์ API ก็จะมีควมหมายตามแต่ชนิดข้อมูลของพารามิเตอร์ตามที่เคยกล่าวรายละเอียดมาแล้วข้างต้น
ByRef	กำหนดให้พารามิเตอร์มีการส่งค่าแบบ passed by reference
ParamArray	ใช้กำหนดให้พารามิเตอร์ดังกล่าวเป็น array ของข้อมูลชนิด variant ซึ่งต้องเป็นพารามิเตอร์ตัวสุดท้ายของ Procedure เท่านั้น โดยทั่วไปส่วนนี้มักจะไม่สามารถใช้ได้กับฟังก์ชันวินโดวส์ API มาตรฐาน ยกเว้นฟังก์ชันถูกออกแบบเฉพาะกับ Visual Basic 6.0 เท่านั้น

## ตารางที่ 2.6 การประกาศฟังก์ชันวินโดวส์ API

### 2.4 พื้นฐานเกี่ยวกับไลบรารี CSIM

CSIM เป็น Library สำหรับการทำให้ Simulation ซึ่งเขียนด้วยภาษา C สามารถใช้ได้ทั้งในระบบปฏิบัติการ DOS Unix และ Microsoft Window มีคำสั่งต่าง ๆ ที่จะช่วยในการเขียนโปรแกรมเพื่อทำ Simulation ได้ง่ายขึ้น ในชุดของ CSIM จะประกอบด้วย

1. SIM\_LIB.H ใช้ในการประกาศสำหรับ Simulation Library และการจัดการกับ Entities และ Queues
2. SIM\_LIB.C ใช้ในการทำ Event Processing, Queue Handling, Entity Creation
3. RANDOM.H ใช้ในการประกาศสำหรับการสร้างชุดของตัวเลขแบบสุ่ม
4. RANDOM.C ใช้ในการสร้างตัวเลขแบบสุ่มจากการกระจายแบบต่าง ๆ
5. RESULTS.H ใช้ในการประกาศสำหรับการจัดการกับผลลัพธ์ของการทำ Simulation
6. RESULTS.C ใช้ในการสร้างตัวแปรสำหรับบันทึกผลลัพธ์ การคำนวณทางสถิติ และการสร้าง Histograms
7. RESOURCE.H ใช้ในการประกาศสำหรับการจัดการกับ Resource ที่ใช้ในแบบจำลอง
8. RESOURCE.C ใช้ในการสร้างและจัดการเกี่ยวกับ Resource ที่ใช้ในแบบจำลอง
9. STATS.H ใช้ในการประกาศสำหรับ Function ทางสถิติ
10. STATS.C ใช้ในการคำนวณ Confidence intervals
11. ERROR.H ใช้ในการกำหนด Error และ Warning
12. ERROR.C ใช้ในการจัดการกับ Error และ Warning ที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ญาติเห็นนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งภายใน Simulation.Library ก็จะมีชุดคำสั่งในการสร้างโปรแกรม รวมทั้งประเภทของตัวแปร และตัวแปรที่ใช้ในการทำ Simulation ดังนี้

#### 2.4.1 Type declarations

Library	Type
RANDOM.H	Stream
RESOURCE.H	Resource
RESULTS.H	table histogram summary
SIM_LIB.H	entity queue

#### ตารางที่ 2.7 ประเภทของการประกาศตัวแปร

#### 2.4.2 ตัวแปร

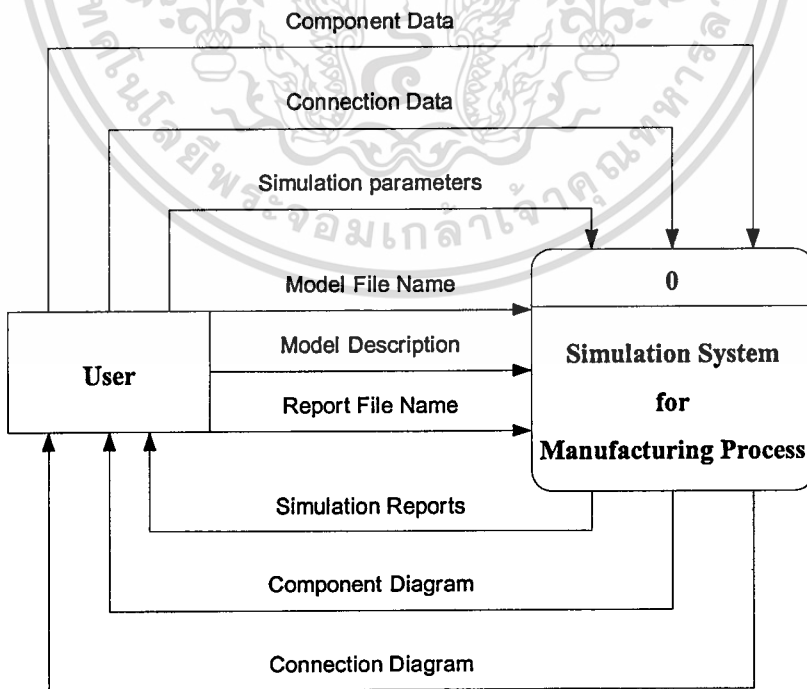
Library	Variables
SIM_LIB.H	Active current sim_clock

#### ตารางที่ 2.8 ตัวแปรของ CSIM Library

### บทที่ 3

#### การออกแบบระบบ

ในขั้นตอนการออกแบบได้ออกแบบโดยแบ่งการทำงานออกเป็น 3 ส่วนตามลักษณะหน้าที่การทำงาน โดยในระบบประกอบไปด้วยส่วน Model ที่ทำหน้าที่สร้าง หรือแก้ไขแบบจำลองโดยใช้รูปภาพแทนองค์ประกอบต่าง ๆ ในระบบ รวมทั้งการเก็บบันทึก หรือเรียกใช้งานแบบจำลองที่ได้เคยออกแบบไว้ ส่วน Report ที่ทำหน้าที่ในการแสดงรายงานผลของแบบจำลองจากการประมวลผล หรือจากรายงานที่ได้เก็บบันทึกไว้ นอกจากนี้ในระบบยังประกอบด้วยส่วน Simulation ที่พัฒนาแยกออกจาก 2 ส่วนแรก โดยมีหน้าที่ในการรับข้อมูลของแบบจำลองจากส่วน Model เพื่อนำมาประมวลผลแบบจำลองแล้วส่งผลลัพธ์ที่ได้กลับไปยังส่วน Report เพื่อทำเป็นรายงานแสดงให้แก่ผู้ใช้งาน ทั้งนี้ได้ออกแบบระบบให้ส่วน Simulation ติดต่อกับส่วนอื่น ๆ ในรูปแบบของวินโดวส์ API และอยู่ในเครื่องคอมพิวเตอร์แบบ Stand-Alone System ซึ่งสามารถแสดงด้วย Context diagram และแผนภาพการไหลข้อมูล (Data Flow Diagram) ได้ดังนี้

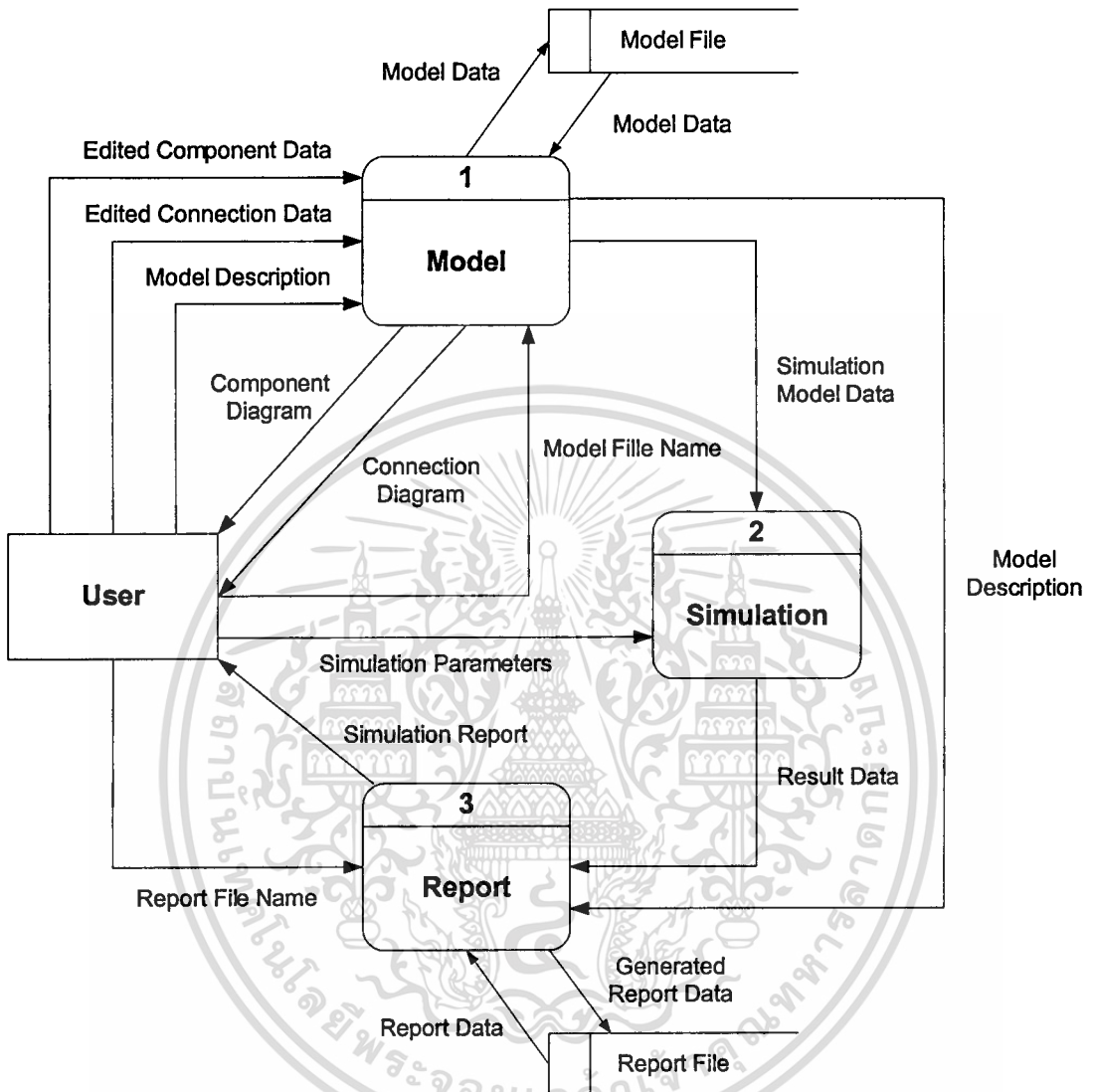


ภาพที่ 3.1 Context Diagram ของระบบแบบจำลองในกระบวนการผลิต

จากภาพที่ 3.1 สามารถอธิบายได้ว่าในระบบแบบจำลองในกระบวนการผลิตนั้นจะทำการติดต่อกับผู้ใช้โดยตรง เพื่อรับและส่งข้อมูล ได้แก่

- Component Data เป็นรายละเอียดและข้อกำหนดขององค์ประกอบต่าง ๆ ที่ประกอบเป็นแบบจำลองในกระบวนการผลิต เช่น ชนิดขององค์ประกอบในแบบจำลอง ชื่อขององค์ประกอบในแบบจำลอง
- Connection Data เป็นความสัมพันธ์ระหว่างองค์ประกอบต่าง ๆ ในแบบจำลอง เพื่อแสดงการเชื่อมต่อไปยังองค์ประกอบอื่น ๆ
- Simulation Parameters เป็นข้อกำหนดในการทดสอบแบบจำลอง เช่น ระยะเวลาที่ใช้ในการทดสอบแบบจำลอง
- Model File Name เป็นชื่อเพิ่มข้อมูลของแบบจำลองที่ผู้ใช้ต้องการบันทึกหรือต้องการเรียกขึ้นมาใช้งาน
- Model Description เป็นรายละเอียดของแบบจำลองที่ต้องการเก็บบันทึกไว้
- Report File Name เป็นชื่อเพิ่มข้อมูลของผลการประมวลผลจากแบบจำลองที่ผู้ใช้ต้องการบันทึกหรือต้องการเรียกขึ้นมาใช้งาน
- Component Diagram เป็นรูปภาพแสดงองค์ประกอบต่าง ๆ ของแบบจำลองในกระบวนการผลิต
- Connection Diagram เป็นรูปภาพแสดงทิศทางการเชื่อมต่อขององค์ประกอบต่าง ๆ ของแบบจำลองในกระบวนการผลิต
- Simulation Report เป็นรายงานแสดงผลลัพธ์ที่ได้จากการประมวลผลแบบจำลองในกระบวนการผลิต

ส่วนการออกแบบการทำงานของระบบแบบจำลองในกระบวนการผลิตนั้นสามารถแบ่งการทำงานออกเป็น 3 ส่วนคือ ส่วน Model ที่ทำหน้าที่รับและแสดงข้อมูลของแบบจำลอง ส่วน Simulation ที่ทำหน้าที่ในการประมวลผลแบบจำลอง และส่วน Report ที่ทำหน้าที่ในการสร้างรายงานผลของการประมวลผล การออกแบบการทำงานของระบบแบบจำลองในกระบวนการผลิตสามารถเขียนเป็น Data Flow Diagram (DFD Level 1) ได้ดังนี้

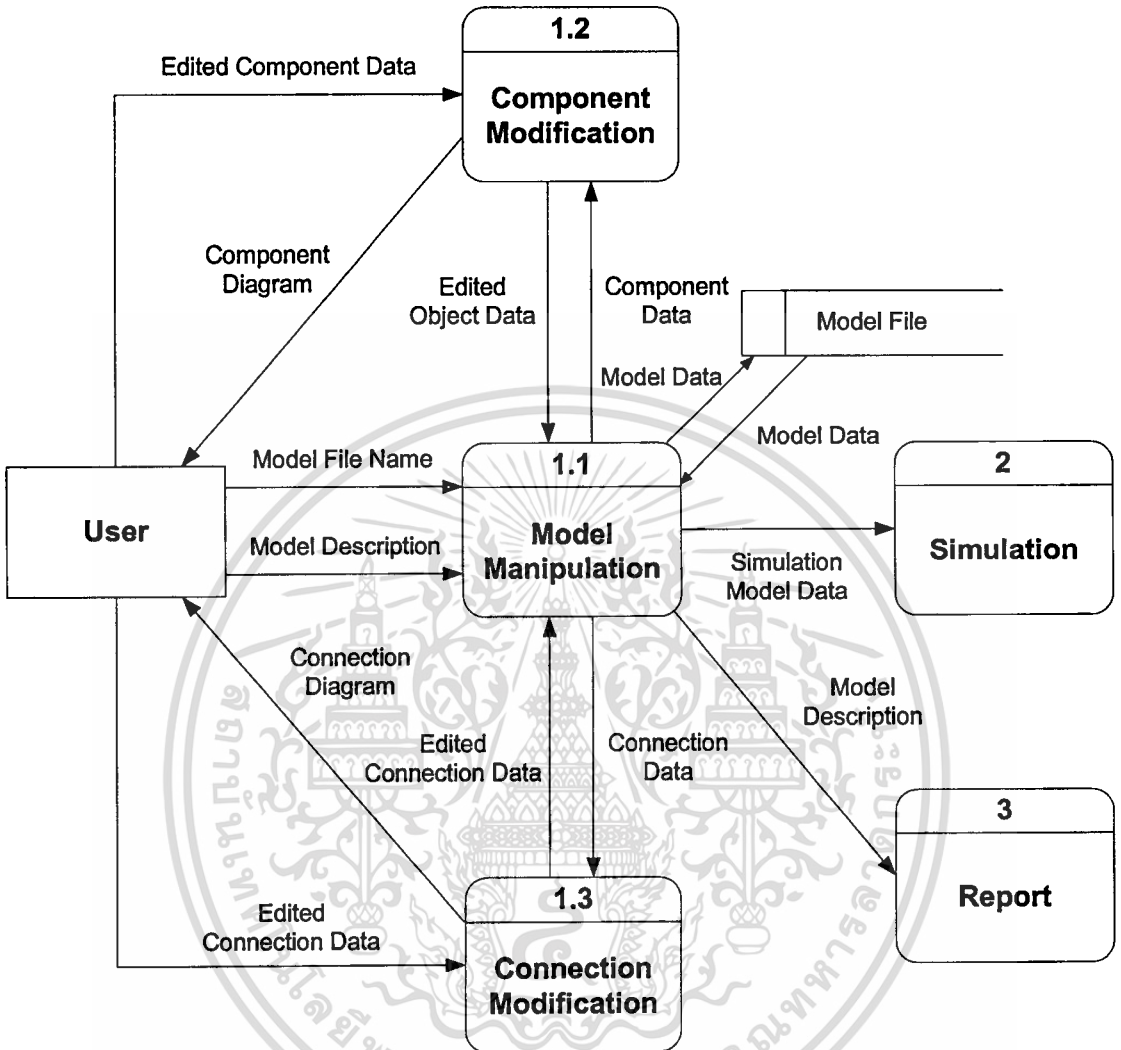


ภาพที่ 3.2 DFD Level 1 ของระบบแบบจำลองในกระบวนการผลิต

จากภาพที่ 3.2 สามารถอธิบายได้ว่าในระบบแบบจำลองในกระบวนการผลิตนั้นเราแบ่งการทำงานออกเป็น 3 ส่วนใหญ่ ๆ คือ

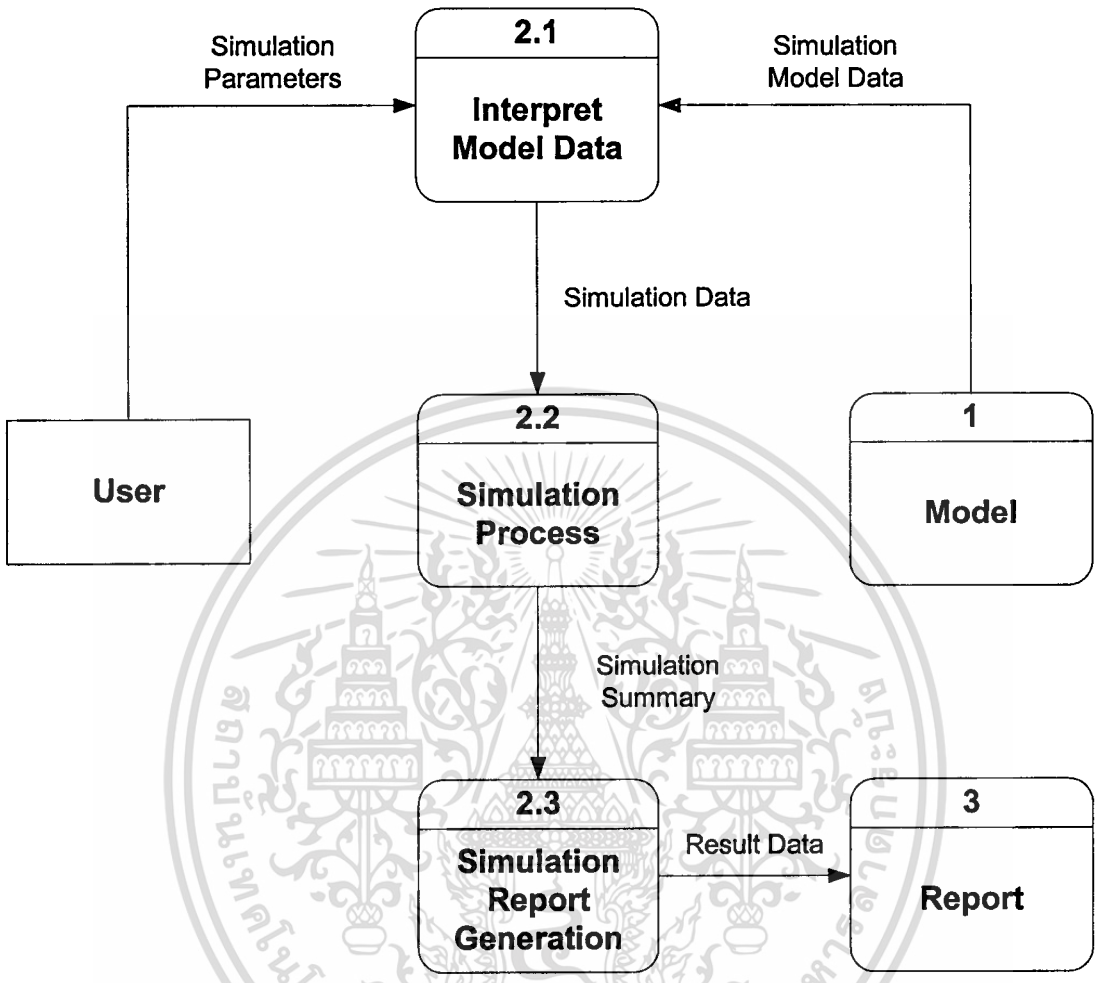
1. Model มีหน้าที่ในการจัดการเกี่ยวกับการสร้างแบบจำลอง
2. Simulation มีหน้าที่ในการ Simulate แบบจำลองที่เราสร้างขึ้น
3. Report มีหน้าที่ในการจัดการเกี่ยวกับผลลัพธ์ที่ได้จากการทำ Simulation

ซึ่งสามารถแสดงรายละเอียดแต่ละส่วนจาก Data Flow Diagram (DFD Level 2) ได้ดังนี้



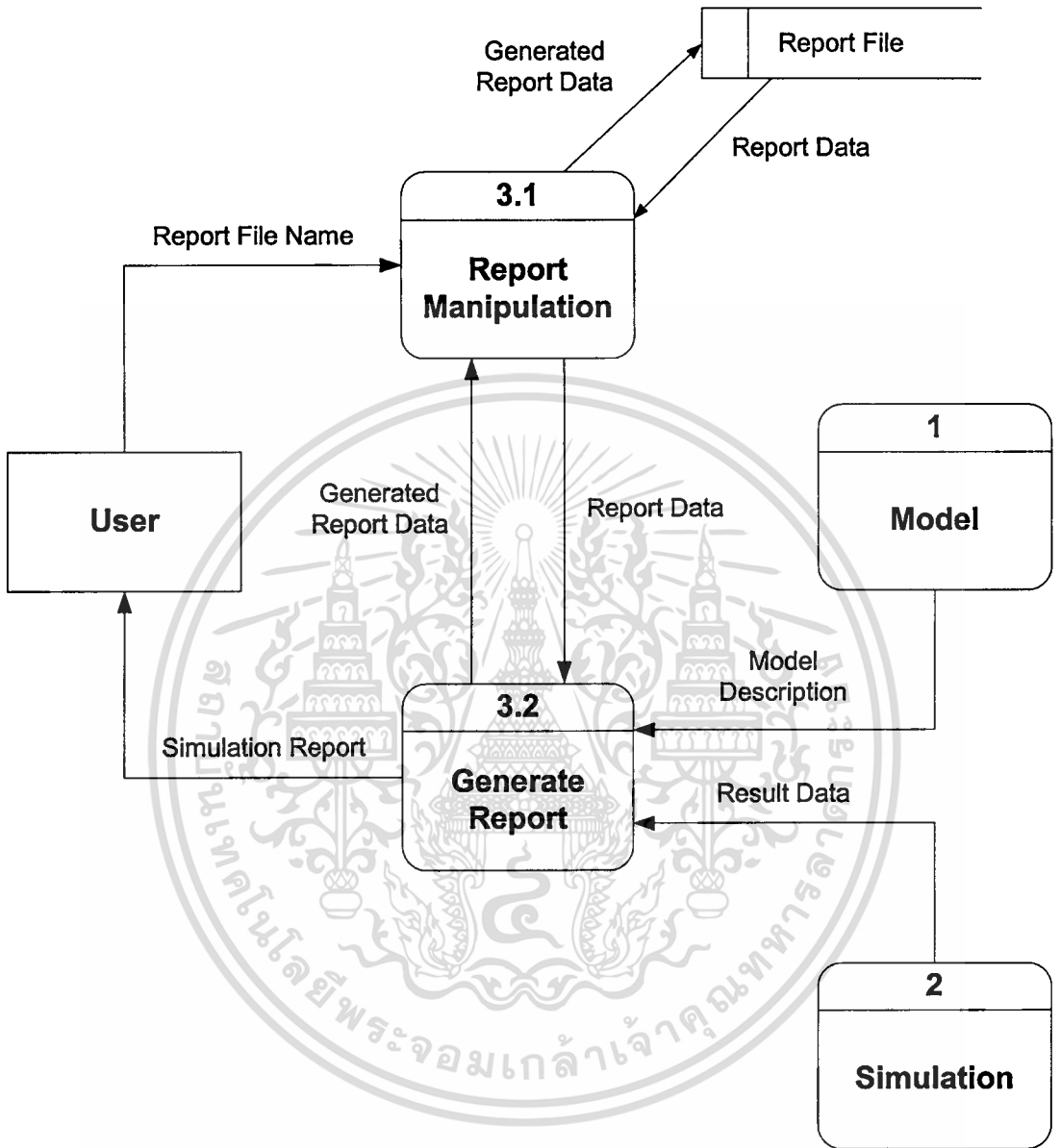
ภาพที่ 3.3 DFD Level 2 การออกแบบ Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.4 DFD Level 2 การประมวลผลแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.5 DFD Level 2 การแสดง Report

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ELEMENTARY PROCESS DESCRIPTION

Process ID	1.0
Process Name	Model
Description	สร้าง แก้ไข และจัดการแบบจำลองในกระบวนการผลิต
Input Data Flows	Component Data เป็นข้อมูลองค์ประกอบของแบบจำลอง Connection Data เป็นข้อมูลการเชื่อมต่อภายในแบบจำลอง Model Description เป็นรายละเอียดของแบบจำลอง Model File Name เป็นชื่อแฟ้มข้อมูลในการเก็บแบบจำลอง Model Data เป็นข้อมูลโครงสร้างของแบบจำลอง
Output Data Flows	Component Diagram เป็นแผนภาพของแบบจำลอง Connection Diagram เป็นแผนภาพแสดงการเชื่อมต่อภายในแบบจำลอง Model Data เป็นข้อมูลโครงสร้างของแบบจำลอง

Process ID	2.0
Process Name	Simulation
Description	ประมวลผลแบบจำลอง
Input Data Flows	Simulation Model Data เป็นข้อมูลแบบจำลองสำหรับการประมวลผล Simulation Parameters เป็นเงื่อนไขในการประมวลผล
Output Data Flows	Result Data เป็นผลลัพธ์ที่ได้จากการประมวลผล

Process ID	3.0
Process Name	Report
Description	แสดง และจัดการรายงานจากการประมวลผล
Input Data Flows	Result Data เป็นผลลัพธ์ที่ได้จากการประมวลผล Model Descriptions เป็นรายละเอียดของแบบจำลอง Report File Name เป็นชื่อแฟ้มข้อมูลของรายงาน Report Data เป็นข้อมูลของรายงาน
Output Data Flows	Simulation Report เป็นรายงานผลของการทำ Simulation Generated Report Data เป็นข้อมูลของรายงานที่ได้จากการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้เผยแพร่ภายนอกองค์กร

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Process ID	1.1
Process Name	Model Manipulation
Description	จัดการเกี่ยวกับการบันทึก แก้ไข และการเรียกใช้แบบจำลอง
Input Data Flows	Edited Component Data Connection Data Model Data Model File Name Model Description
Output Data Flows	Simulation Model Data Component Data Connection Data Model Data Model Description

Process ID	1.2
Process Name	Component Modification
Description	จัดการเกี่ยวกับการสร้าง แก้ไข และแสดงรูปองค์ประกอบ
Input Data Flows	Edited Component Data Component Data
Output Data Flows	Edited Component Data Component Diagram

Process ID	1.3
Process Name	Connection Modification
Description	จัดการเกี่ยวกับการสร้าง แก้ไข และแสดงการเชื่อมต่อ
Input Data Flows	Edited Connection Data Connection Data
Output Data Flows	Edited Connection Data Connection Diagram

Process ID	2.1
Process Name	Interpret Model Data
Description	ตีความข้อมูลจากแบบจำลอง
Input Data Flows	Simulation Parameter Simulation Model Data
Output Data Flows	Simulation Data

Process ID	2.2
Process Name	Simulation Process
Description	ประมวลผลแบบจำลอง
Input Data Flows	Simulation Data
Output Data Flows	Simulation Summary

Process ID	2.3
Process Name	Simulation Report Generation
Description	จัดรูปแบบของผลลัพธ์ที่ได้จากการประมวลผล
Input Data Flows	Simulation Summary
Output Data Flows	Result Data

Process ID	3.1
Process Name	Report Manipulation
Description	จัดการเกี่ยวกับการแสดง และการเรียกใช้รายงาน
Input Data Flows	Generated Report Data Report Data Report File Name
Output Data Flows	Generated Report Data Report Data

Process ID	3.2
------------	-----

เอกสารที่สงวนไว้สำหรับ 3.2 การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Process Name	Generate Report
Description	สร้างรายงานผลของการประมวลผลแบบจำลอง
Input Data Flows	Model Description Result Data Report Data
Output Data Flows	Generated Report Data Simulation Report

### 3.1 การออกแบบการทำงานของส่วน Model

เพื่อให้เห็นภาพรวมของระบบได้ง่ายและมีความสะดวกรวดเร็วในการสร้างและการปรับเปลี่ยนแบบจำลองจึงมีการใช้รูปภาพเพื่อแทนองค์ประกอบต่าง ๆ ในระบบ ซึ่งเราสามารถแบ่งองค์ประกอบของแบบจำลองทางคอมพิวเตอร์ในกระบวนการผลิตออกเป็น 6 ประเภทด้วยกันคือ

1. Generator ใช้แทนจุดเริ่มต้นในกระบวนการผลิต ซึ่งจะเป็นวัตถุดิบที่ใช้ในกระบวนการผลิต โดยปกติจะมีการใช้วัตถุดิบหลายๆชนิด ซึ่งบางชนิดอาจจะต้องมีการแปรรูปก่อน หรือบางชนิดอาจจะนำไปใช้ในขั้นตอนการผสมหรือการประกอบเป็นผลิตภัณฑ์ได้เลย

2. Server ใช้แทนบริการต่างๆที่เกิดขึ้นในกระบวนการผลิต เช่นการแปรรูปวัตถุดิบ การขนย้าย หรือเตรียมชิ้นส่วนก่อนนำไปประกอบรวมกัน

3. Queue ใช้เป็นที่พักของวัตถุดิบหรือชิ้นส่วนก่อนนำเข้าสู่กระบวนการต่างๆ ในกรณีที่กระบวนการไม่สามารถทำงานได้เร็วกว่าการมาถึงของวัตถุดิบหรือชิ้นส่วน ซึ่งมีวิธีในการจัดการข้อมูลใน Queue ที่พบบ่อยๆ ได้แก่ มาก่อน-ใช้ก่อน (First in – First out, FIFO) มาหลัง-ใช้ก่อน (Last in – First out, LIFO) เลือกแบบสุ่ม (Random) หรือการใช้กฎอภิสัทย์ (Priority)

4. Assembly operation ใช้แทนการประกอบหรือการผสมชิ้นส่วนต่างๆเข้าด้วยกัน โดยปกติจะทำงานโดยใช้เครื่องจักรซึ่งผลลัพธ์ที่ได้จะขึ้นอยู่กับกำลังการผลิตของเครื่องจักรนั้น แต่ในกรณีที่วัตถุดิบหรือชิ้นส่วนเข้ามามากกว่ากำลังการผลิตจะทำให้เครื่องจักรไม่สามารถทำงานได้ทันทีจะเกิดการรอคอยขึ้นก่อนเข้าสู่การทำงาน หรือในกรณีที่วัตถุดิบหรือชิ้นส่วนเข้ามาช้ากว่ากำลังการผลิตก็จะทำให้เครื่องจักรรอคอยเพื่อที่จะทำงาน ไม่ว่าจะในกรณีใดก็ตามจะเห็นได้ว่าทำให้ระบบทำงานได้ไม่เต็มประสิทธิภาพ

5. Split point เป็นส่วนที่ใช้ในการแยกเส้นทางในกระบวนการผลิต ซึ่งอาจจะมีได้ในกรณีที่มียุทธศาสตร์ผลิตหลายๆสายการผลิต

6. Terminal point เป็นจุดสิ้นสุดของกระบวนการผลิต ซึ่งจะได้เป็นผลิตภัณฑ์ของการผลิตนั้น

นอกจากองค์ประกอบที่กล่าวมาแล้วยังมี Connector Lines ที่ใช้เชื่อมต่อระหว่างองค์ประกอบต่าง ๆ เพื่อแสดงลำดับและขั้นตอนการทำงานของระบบ ซึ่งจะต้องมีการตรวจสอบความสัมพันธ์ระหว่างองค์ประกอบเริ่มต้นและองค์ประกอบในขั้นตอนถัดไป ทั้งในเรื่องของจำนวนองค์ประกอบในการเชื่อมต่อ และชนิดขององค์ประกอบดังนี้

### 3.1.1 จำนวนองค์ประกอบในการเชื่อมต่อ

No.	Component Types	Number of Inputs	Number of Outputs
1.	Generator	-	1
2.	Server	1	1
3.	Queue	1	1
4.	Assembly operation	N	1
5.	Split point	1	N
6.	Terminal point	1	-

ตารางที่ 3.1 จำนวนองค์ประกอบในการเชื่อมต่อ

### 3.1.2 เงื่อนไขในการเชื่อมต่อ

Destination Source	Generator	Server	Queue	Assembly	Split	Terminal
Generator	-	✓	✓	✓	-	-
Server	-	-	✓	✓	✓	✓
Queue	-	✓	-	✓	-	-
Assembly	-	✓	✓	-	✓	✓
Split	-	✓	✓	-	-	✓
Terminal	-	-	-	-	-	-

ตารางที่ 3.2 เงื่อนไขในการเชื่อมต่อ

### 3.1.3 ข้อมูลของแบบจำลอง

- Number of Components เป็นจำนวนของ Component ในแต่ละประเภท
- Model Description เป็นรายละเอียดของแบบจำลอง

### 3.1.4 ข้อมูลขององค์ประกอบ

1. Component ID: "ONNN" (Component: 0 – 5, NNN: Running Number)
2. Type: 0 - 5
3. Name: ชื่อขององค์ประกอบที่จะแสดงได้ภาพขององค์ประกอบ
4. Label: ข้อมูลขององค์ประกอบที่จะแสดงได้ Name
5. Location: เป็นตำแหน่งและขนาดในการแสดงองค์ประกอบ (X, Y, H, W)
6. Input list: เก็บ ID ของ Input Components
7. Output list: เก็บ ID ของ Output Components

### 3.1.5 ข้อมูลเพิ่มเติมสำหรับแต่ละองค์ประกอบ

No.	Component Types	Tag	Model Attributes	Result Attributes
1.	Generator	0	Distribution Material Type Mean Interarrival Time	Count
2.	Server	1	Distribution Additional Delay Mean Service Time	Service Time Idle Time Idle % Lost Served
3.	Queue	2	Queue Type Length	Avg. waiting time LMAX LMIN Avg. Length Lost
4.	Assembly operation	3	Distribution Additional Delay	Service Time Idle Time

			Mean Process Time	Idle %
				Lost
				Served
5.	Split point	4	Flow Decision	-
6.	Terminal point	5	-	Count

### ตารางที่ 3.3 ข้อมูลเพิ่มเติมสำหรับแต่ละองค์ประกอบ

โดยกำหนดค่าต่าง ๆ ดังนี้

- Distribution: 0=None, 1=Exponential, 2=Poisson
- Flow Decision: 0=Cyclic
- Queue Type: 0=FIFO

### 3.2 การออกแบบการทำงานของส่วน Simulation

ระบบแบบจำลองของกระบวนการผลิตจะครอบคลุมตั้งแต่การเข้ามาของวัตถุดิบจนกระทั่งผลิตได้เป็นผลิตภัณฑ์ ซึ่งแบบจำลองจะประกอบด้วย

- Entities ได้แก่ วัตถุดิบหรือชิ้นส่วนที่ถูกส่งเข้าสู่ Process ต่างๆ ซึ่งตอนเริ่มต้นการทำงานจะสร้างมาจาก Generator ก่อนเสมอ
- Resources ได้แก่ เครื่องจักรหรือคนงาน ซึ่งถูกแทนด้วย Server และ Assembly
- Queues ซึ่งอาจจะมีหรือไม่มีก็ได้ขึ้นอยู่กับการสร้างแบบจำลอง

ในการออกแบบการทำงานของส่วน Simulation นั้นจะประกอบด้วย 3 ส่วนหลักคือ ส่วนตีความข้อมูล, ส่วนประมวลผล และส่วนรายงานผลของการทำ Simulation ผ่านวินโดวส์ API

#### 3.2.1 การออกแบบส่วนตีความข้อมูล

เนื่องจากการแยกส่วนประมวลผลแบบจำลองออกจากส่วนติดต่อกับผู้ใช้งาน ทำให้ต้องมีการรับส่งข้อมูลระหว่างกัน โดยใช้ลักษณะฟังก์ชันของวินโดวส์ API ซึ่งจะมี Arguments ที่ใช้ 3 Arguments คือ

1. ข้อมูลของการทำ Simulation
2. ข้อมูลของแบบจำลอง
3. ผลลัพธ์ของการประมวลผลจากแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเหล่านี้จะมีรูปแบบในการเก็บเพื่อใช้ในการรับส่ง ดังนั้นก่อนจะนำไปใช้งานได้จะต้องมีการตีความข้อมูลก่อน ซึ่งสามารถออกแบบ Module หลัก ๆ ได้ดังนี้

1. Module ตีความข้อมูลของการทำ Simulation ซึ่งจะได้ Duration time, Transient time ในการทำ Simulation
2. Module ตีความข้อมูลของแบบจำลอง ซึ่งจะได้ Array ของข้อมูลองค์ประกอบต่าง ๆ ในแบบจำลอง เช่น ประเภท, ขนาดของ Queue, เวลาเฉลี่ยที่ใช้ทำงาน เป็นต้น เพื่อใช้เป็นที่อ้างอิงข้อมูลทั้งก่อนและระหว่างการประมวลผล
3. Module สร้างความสัมพันธ์ระหว่างองค์ประกอบต่าง ๆ เพื่อให้แต่ละส่วนสามารถอ้างอิงกับองค์ประกอบอื่นได้ทั้งด้านหน้าและด้านหลัง
4. Module เก็บข้อมูลของ Server, Assembly ซึ่งทำหน้าที่เป็นผู้ให้บริการในระบบ สำหรับการใช้ในการประมวลผลแบบจำลอง

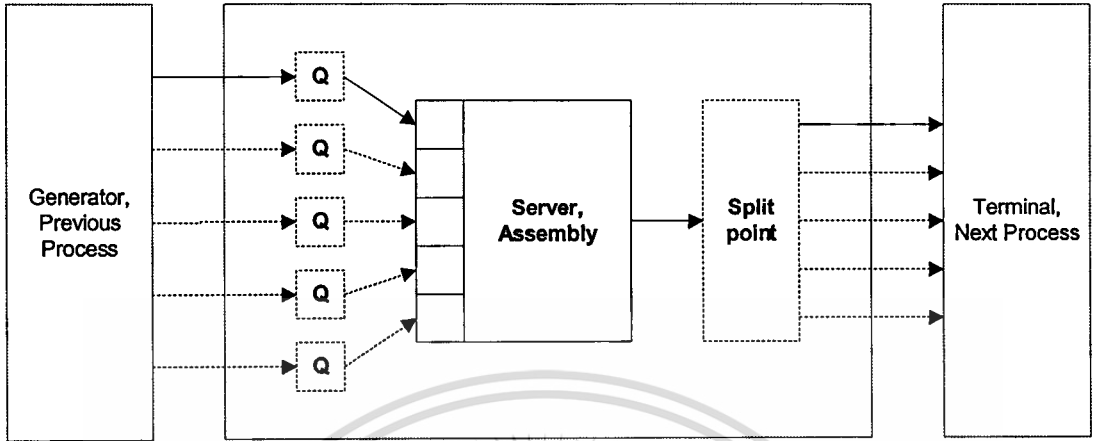
### 3.2.2 การออกแบบส่วนประมวลผลในแบบจำลอง

ในการออกแบบส่วนประมวลผลเราจะแบ่งการประมวลผลออกเป็น 2 ช่วงเวลาคือ Transient time และ duration time โดยจะนำเฉพาะผลที่ได้ในช่วง duration time เท่านั้นมาคำนวณทางสถิติเพื่อตัดข้อมูลที่ได้ในขณะที่ระบบยังไม่เสถียรออกไป ทำให้ผลลัพธ์ที่ได้มีความถูกต้องมากขึ้น ในส่วนของการทำงานเราสามารถแบ่งองค์ประกอบของแบบจำลองออกได้เป็น 3 กลุ่ม ได้แก่

1. กลุ่มที่ใช้ในการทำงาน ได้แก่ Server, Assembly, Queue และ Split point
2. กลุ่มที่ใช้ในการสร้าง Input ของระบบ ได้แก่ Generator
3. กลุ่มที่ใช้ในการเก็บ Output ของระบบ ได้แก่ Terminal

ซึ่งกลุ่มที่ 2 และกลุ่มที่ 3 ไม่ได้เป็นส่วนทำงานในระบบ ดังนั้นในการประมวลผลเราจะพิจารณากลุ่มที่ทำงานเป็นหลัก โดยจะแยกแบบจำลองออกเป็นกลุ่ม ๆ จากนั้นจะให้ระบบทำการตรวจสอบกลุ่มที่ใช้ในการทำงานตลอดเวลาเพื่อจัดการกับ Entities ที่อยู่ในกลุ่ม หลังจากสร้างการตรวจสอบแล้วจึงเริ่มให้ทุก Generator ทำการสร้าง Entities เข้าสู่ระบบไปยังกลุ่มที่ใช้ในการทำงานตามลำดับของการทำงาน ดังภาพที่ 3.6 โดยเส้นประในรูปจะหมายถึง Optional components ซึ่งอาจจะมีหรือไม่มีในระบบก็ได้

### Manufacturing Process



ภาพที่ 3.6 การทำงานขององค์ประกอบต่าง ๆ ในแบบจำลอง

การทำงานของระบบจะเริ่มจากส่วน Generator ก่อนเสมอเพื่อส่งวัตถุดิบเข้าสู่ระบบ จากนั้นจะเข้าสู่กระบวนการทำงาน ซึ่งในการทำงานแต่ละ Server อาจจะมี Queues เพื่อใช้ในการเก็บชิ้นส่วนที่เข้ามาในขณะที่ Server ไม่พร้อมที่จะทำงาน แต่ถ้า Server ว่างอยู่ Server ก็จะตรวจสอบ Buffer ของตน ในกรณีที่มี Buffer ว่างและมี Queue อยู่ Server ก็จะนำชิ้นส่วนจาก Queue เข้ามาเก็บใน Buffer แล้วเริ่มทำงาน ในกรณีของ Assembly ซึ่งอาจมีชิ้นส่วนหลายชิ้นจะมีการตรวจสอบทุก Buffer ที่มีการเชื่อมต่อ เมื่อทำงานเสร็จก็จะส่งชิ้นส่วนชิ้นอันดับแรกออกไปเสมือนว่าเป็นชิ้นส่วนที่ได้รับการประกอบแล้ว ในการส่งไปยัง Process อื่นจะมีการตรวจสอบช่องทางที่จะส่งออกก่อน ถ้ามีการ Split เส้นทางจะมีการเลือกเส้นทางที่ควรส่งต่อไปตามเงื่อนไขที่กำหนดไว้แล้วส่งต่อไปยัง Process อื่นหรือ Terminal ในกรณีที่เสร็จสิ้นการผลิตแล้ว

### 3.2.3 การออกแบบส่วนรายงานผลของการทำ Simulation

การรายงานผลของการทำ Simulation กลับไปยังส่วนแสดงผล โดยผ่าน Arguments ของ วินโดวส์ API นั้นจะมีการจัดรูปแบบของข้อมูลก่อน เพื่อให้สามารถนำข้อมูลเหล่านั้นกลับไปใช้งานได้ ซึ่งสามารถออกแบบ Module หลัก ๆ ได้ดังนี้

1. Module สรุปผลทางสถิติจากการประมวลผล ซึ่งทำหน้าที่ในการรวบรวมข้อมูลขององค์ประกอบในแบบจำลอง แล้วทำการคำนวณเพื่อหาค่าต่างๆที่จะแสดงแก่ผู้ใช้งาน

2. Module จัดรูปแบบของข้อมูลที่จะส่งเป็นข้อความ มีหน้าที่ในการเรียบเรียงข้อมูลของแต่ละองค์ประกอบในแบบจำลองโดยใช้เครื่องหมายแยกระหว่างองค์ประกอบและข้อมูลในองค์ประกอบของแบบจำลอง

### 3.3 การออกแบบการทำงานของส่วน Report

ส่วน Report มีหน้าที่หลักในการรายงานผลที่ได้จากการประมวลผลแบบจำลอง รวมทั้งทำการเก็บผลที่ได้ในรูปแบบ Text File (\*.smr) เพื่อนำกลับมาแสดงผลได้อีกในภายหลัง ดังนั้นสามารถออกแบบการทำงานได้ดังนี้

1. Report Manipulations ทำหน้าที่ในการจัดการเกี่ยวกับ File ที่เก็บผลที่ได้การประมวลผล ซึ่งประกอบด้วยการทำงานต่าง ๆ ดังนี้

- บันทึกผลจากการทำ Simulation เก็บไว้
- บันทึกรายงานที่ได้จากการทำ Simulation
- เรียกใช้รายงานที่ได้บันทึกไว้

2. Generate Report ทำหน้าที่ในการจัดรูปแบบรายงานเพื่อแสดงต่อผู้ใช้งาน โดยอาศัยข้อมูลที่ได้จากการประมวลผล และข้อมูลจากรายละเอียดของ Model

## บทที่ 4

### การพัฒนาระบบแบบจำลองในกระบวนการผลิต

ในขั้นตอนของการพัฒนาระบบแบบจำลองในกระบวนการผลิตนั้น จะพัฒนาโปรแกรมในลักษณะเป็น Multiple Document Interface (MDI) คือ โปรแกรมที่มีหลายวินโดว์ย่อยอยู่ในวินโดว์หลักอันเดียวกัน และพัฒนาส่วนงานต่าง ๆ ตามที่ได้ออกแบบไว้ ซึ่งจะมีรายละเอียดดังนี้

#### 4.1 เครื่องมือที่ใช้ในการพัฒนาระบบ

ในการพัฒนาระบบจะใช้ Hardware และ Software ดังนี้

1. เครื่องคอมพิวเตอร์ AMD Thunderbird 1 GHz หน่วยความจำ 256 Mbytes ระบบปฏิบัติการ Microsoft Window 2000 Professional
2. CSIM Library
3. Microsoft Visual Basic 6.0
4. Microsoft Visual C++ 6.0
5. Adobe Photoshop 6.0

#### 4.2 การพัฒนาส่วน Model

เป็นการพัฒนาส่วนที่มีหน้าที่ในการสร้างแบบจำลอง ซึ่งจะแสดงองค์ประกอบและลำดับขั้นตอนการทำงานของระบบ นอกจากนี้ยังทำหน้าที่ในการจัดการข้อมูลของแบบจำลอง

##### 4.2.1 การพัฒนาในส่วนของการบันทึกแบบจำลอง

เนื่องจากในการใช้งานจริง ผู้ใช้ระบบอาจมีความต้องการที่จะนำแบบจำลองที่เคยสร้างไว้ นำกลับมาปรับปรุงแก้ไข หรือเปรียบเทียบกับแบบจำลองใหม่ที่พัฒนาขึ้น ซึ่งจะช่วยให้ผู้ใช้ประหยัดเวลาในแง่ของการพัฒนาแบบจำลอง และการเปรียบเทียบแบบจำลอง ซึ่งระบบจะจัดเก็บแบบจำลองที่ได้พัฒนาขึ้น ในรูปแบบของ Text File (\*.smm) โดยมีรูปแบบ ดังนี้

1. บรรทัดที่ 1 จะเก็บรายละเอียดซึ่งเป็นส่วนที่ใช้บอกว่าเป็นแบบจำลอง ของระบบใด
2. บรรทัดที่ 2 จะเก็บรายละเอียดของแบบจำลอง ได้แก่ ขนาด และตำแหน่งบนหน้าจอ
3. บรรทัดที่ 3 จะเก็บจำนวนขององค์ประกอบที่มีในแบบจำลอง โดยแบ่งตามประเภท

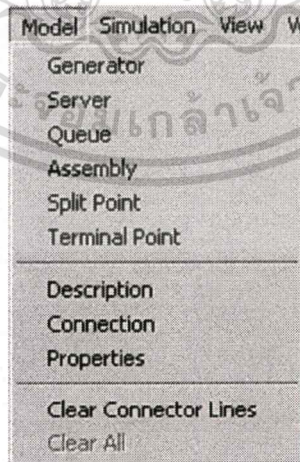
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. บรรทัดที่ 4 จนถึงท้ายสุด จะเป็นรายละเอียดองค์ประกอบที่มีในแบบจำลอง โดยมีรูปแบบในการจัดเก็บตามลำดับ ดังนี้

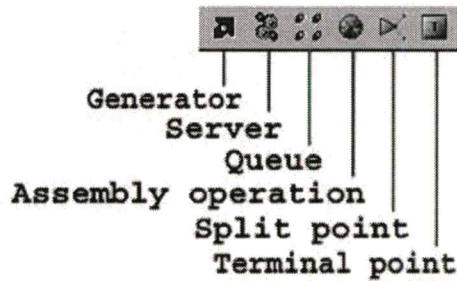
1. Component ID เป็นหมายเลขที่ระบบสร้างขึ้น โดยจะอยู่ในรูปแบบของ “ONNN” (O : เป็นประเภทขององค์ประกอบโดยจะมีค่าเป็น 0-5, NNN : เป็นหมายเลขลำดับขององค์ประกอบ)
2. Component Type เป็นชื่อประเภทขององค์ประกอบ
3. Name เป็นชื่อขององค์ประกอบซึ่งผู้ใช้สามารถที่จะ แก้ไขได้ภายหลัง
4. Label เป็นรายละเอียดขององค์ประกอบซึ่งผู้ใช้สามารถที่จะ แก้ไขได้ภายหลัง
5. Location (X, Y, H, W) เป็นตำแหน่งขององค์ประกอบ โดยจะประกอบด้วย ตำแหน่ง และขนาด
6. Input List เป็นส่วนที่ใช้เก็บ Component ID ที่เป็น Input ขององค์ประกอบ
7. Output List เป็นส่วนที่ใช้เก็บ Component ID ที่เป็น Output ขององค์ประกอบ

#### 4.2.2 การพัฒนาในส่วนของการเพิ่มองค์ประกอบให้กับแบบจำลอง

เป็นส่วนที่พัฒนาขึ้นเพื่อให้ผู้ใช้เพิ่มองค์ประกอบให้กับตัวแบบจำลอง โดยผู้พัฒนาได้ออกแบบให้ง่ายต่อการใช้งานของผู้ใช้ คือ สามารถเลือกได้จากทั้ง Menu และ Menu Bar ดังแสดงในภาพที่ 4.1 และภาพที่ 4.2



ภาพที่ 4.1 Menu ที่ใช้สำหรับเพิ่มองค์ประกอบให้กับตัวแบบจำลอง



ภาพที่ 4.2 Toolbar ที่ใช้สำหรับเพิ่มองค์ประกอบให้กับตัวแบบจำลอง

โดยเมื่อผู้ใช้เลือกองค์ประกอบประเภทต่าง ๆ โปรแกรมจะแสดงภาพสัญลักษณ์ ที่ใช้แทนองค์ประกอบนั้น ๆ บน แผนภาพแบบจำลองที่อยู่บนสุด โดยองค์ประกอบประกอบไปด้วย

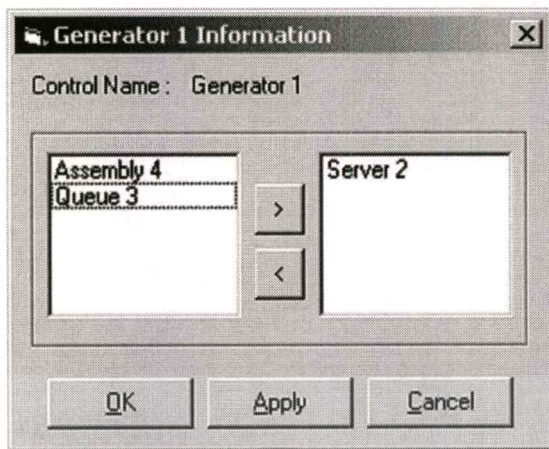
1. Generator
2. Server
3. Queue
4. Assembly Operation
5. Split Point
6. Terminal Point

เพื่อที่ผู้ใช้จะได้นำไปพัฒนาสู่ขั้นตอน การออกแบบรูปแบบการเชื่อมต่อขององค์ประกอบ และการทดสอบแบบจำลองต่อไป

#### 4.2.3 การพัฒนาในส่วนของการออกแบบรูปแบบการเชื่อมต่อขององค์ประกอบ

เป็นส่วนที่พัฒนาขึ้นเพื่อใช้ออกแบบรูปแบบการเชื่อมต่อขององค์ประกอบเพื่อนำไปใช้ในการจำลองค่าต่อไป เมื่อผู้ใช้เลือกองค์ประกอบที่ต้องการสร้างการเชื่อมต่อ ระบบจะตรวจสอบว่าองค์ประกอบที่เลือกสามารถที่จะเชื่อมต่อกับองค์ประกอบใดในแบบจำลองได้บ้าง หลังจากนั้นระบบจะนำองค์ประกอบที่สามารถเชื่อมต่อได้ มาแสดงใน List เพื่อให้ผู้ใช้เลือก ดังแสดงในภาพที่

4.3



ภาพที่ 4.3 รูปแบบการเชื่อมต่อขององค์ประกอบ

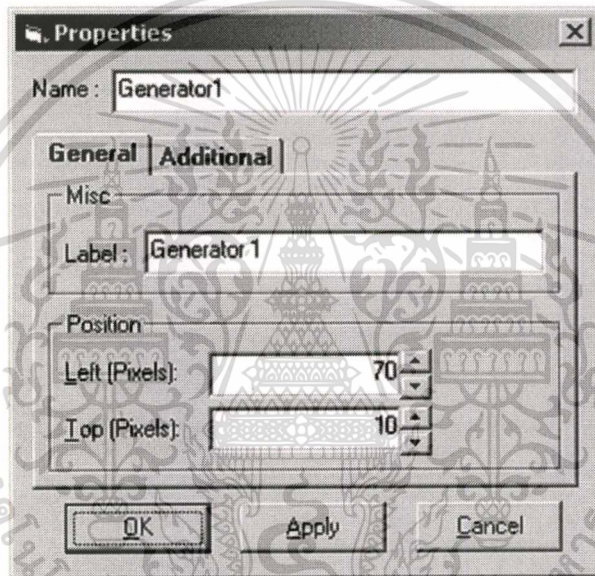
วิธีการทำงานของส่วนการออกแบบรูปแบบการเชื่อมต่อขององค์ประกอบ

1. เลือกองค์ประกอบที่ต้องการออกแบบการเชื่อมต่อ
2. ผู้ใช้เลือก Model Connection ซึ่งผู้ใช้จะสามารถเลือกได้ 3 วิธี ซึ่ง โปรแกรมจะแสดงส่วนของ Components Information ซึ่งจะประกอบไปด้วย Components list (ด้านซ้าย) เป็นส่วนที่แสดงชื่อองค์ประกอบที่สามารถเลือกเป็น Output ขององค์ประกอบที่ต้องการเชื่อมต่อ และ Output model list (ด้านขวา) เป็นส่วนที่แสดงชื่อองค์ประกอบที่เลือกเป็น Output ขององค์ประกอบที่ต้องการเชื่อมต่อเอาไว้แล้ว
  - จาก Main Menu โดยเลือก Model จากนั้นเลือก Connection
  - จาก Menu Bar โดยเลือกที่ปุ่ม Connection
  - จากเมนูลัด โดยเลือกที่องค์ประกอบที่ต้องการ เลือก Mouse ขวา โปรแกรมจะแสดงเมนูลัด จากนั้นให้เลือกที่ Connection
3. ผู้ใช้เลือกองค์ประกอบที่ต้องการเชื่อมต่อจาก List ทางด้านซ้าย (Model List) จากนั้นเลือกที่ปุ่มเพิ่ม หรือ Double Click ที่องค์ประกอบที่ต้องการเลือก ทำให้องค์ประกอบที่เลือกจะปรากฏทางด้านขวา (Output Model List) แทน
4. และถ้าหากผู้ใช้ต้องการยกเลิกการเลือกองค์ประกอบที่ต้องการยกเลิกจาก List ทางด้านขวา (Output Model List) จากนั้นเลือกที่ปุ่มลบ หรือ Double Click ที่องค์ประกอบที่ต้องการยกเลิก ทำให้องค์ประกอบที่ยกเลิกจะปรากฏทางด้านซ้าย (Model List) แทน

#### 4.2.4 การพัฒนาส่วนของการกำหนดคุณสมบัติขององค์ประกอบ

เป็นส่วนที่พัฒนาขึ้นเพื่อให้ผู้ใช้กำหนดรายละเอียด และคุณสมบัติต่าง ๆ ขององค์ประกอบแต่ละประเภทในแบบจำลองที่ผู้ใช้พัฒนาขึ้น โดยในส่วนของรายละเอียดซึ่งเป็นส่วนที่มีอยู่ใน Components ทุกประเภทดังแสดงในภาพที่ 4.4 ประกอบด้วย

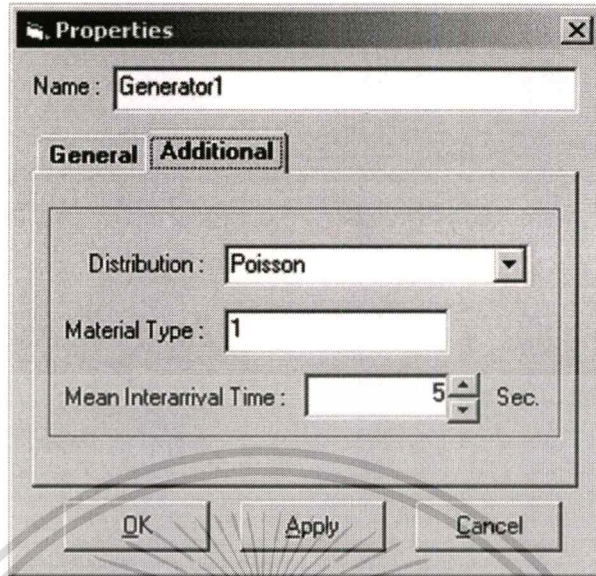
- ชื่อของ Components
- รายละเอียดของ Components
- ตำแหน่งของ Components ในแผนภาพของแบบจำลอง



ภาพที่ 4.4 รายละเอียดขององค์ประกอบ

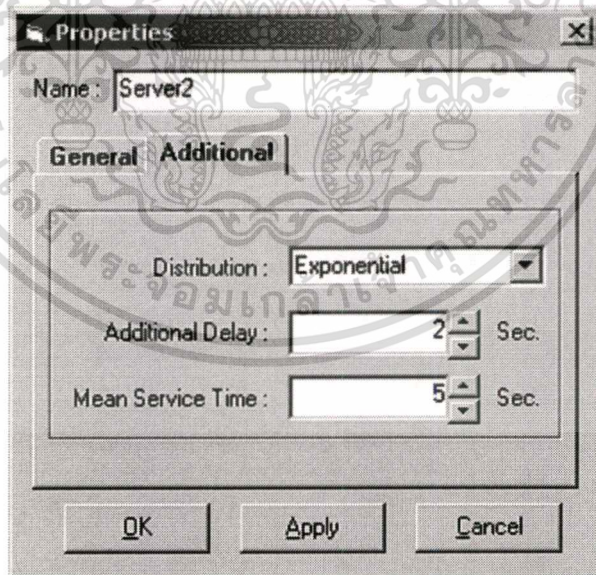
โดยในส่วนของคุณสมบัติ ต่าง ๆ จะแตกต่างกันไป ตามประเภทขององค์ประกอบดังนี้

1. Generator ประกอบด้วยคุณสมบัติต่าง ๆ ดังแสดงในภาพที่ 4.5



ภาพที่ 4.5 คุณสมบัติขององค์ประกอบประเภท Generator

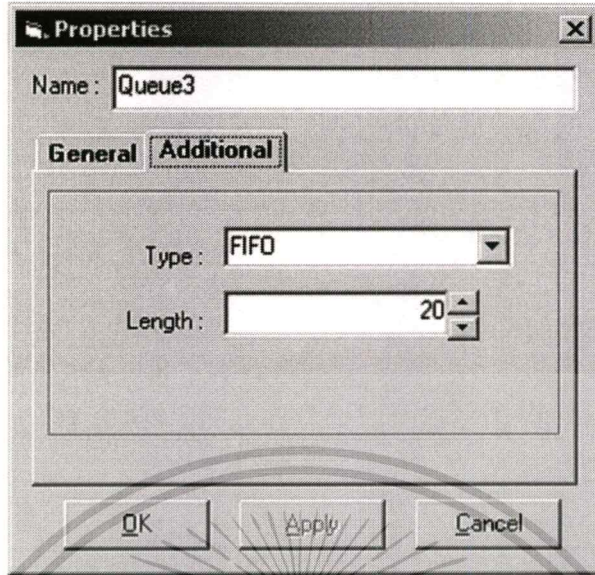
2. Server ประกอบด้วยคุณสมบัติต่าง ๆ ดังแสดงในภาพที่ 4.6



ภาพที่ 4.6 คุณสมบัติขององค์ประกอบประเภท Server

3. Queue ประกอบด้วยคุณสมบัติต่าง ๆ ดังแสดงในภาพที่ 4.7

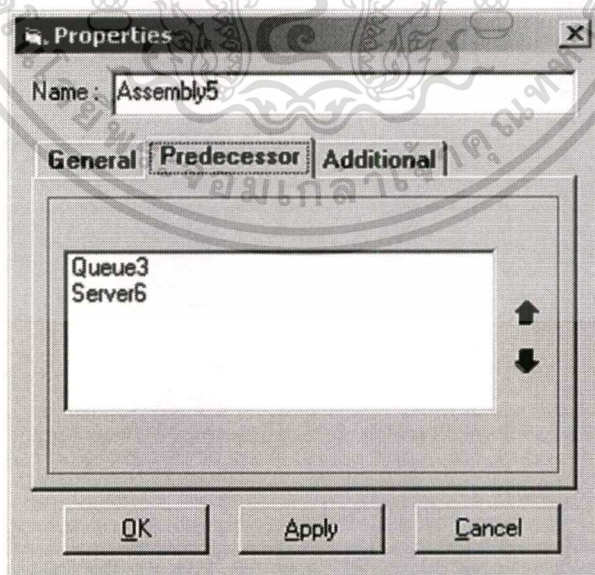
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.7 คุณสมบัติขององค์ประกอบประเภท Queue

#### 4. Assembly Operation ประกอบด้วย 2 ส่วนหลัก คือ

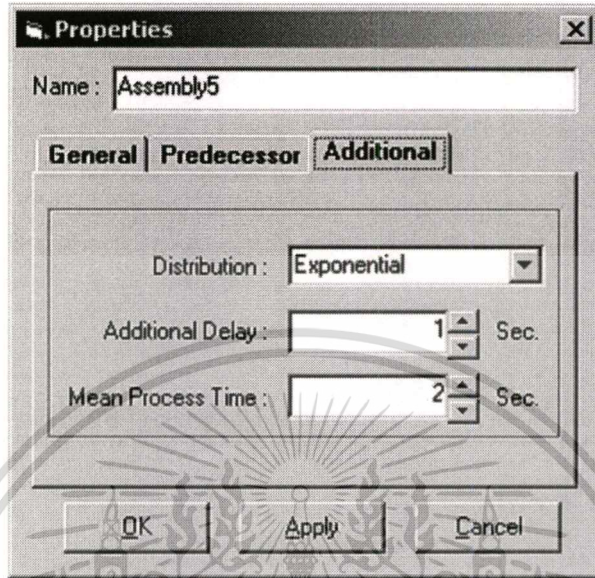
4.1 Predecessor ใช้ในการจัดลำดับการผ่านเข้ามาในองค์ประกอบ โดยให้ผู้ใช้เลือกองค์ประกอบที่ต้องการ จากนั้นให้เลือกปุ่มลูกศรขึ้น หรือลง เพื่อจัดลำดับ ดังภาพที่ 4.8



ภาพที่ 4.8 คุณสมบัติ Predecessor ขององค์ประกอบประเภท Assembly Operation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

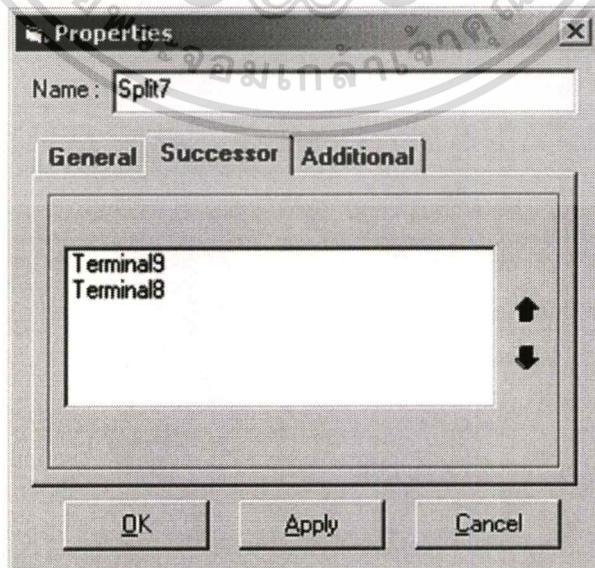
#### 4.2. คุณสมบัติต่าง ๆ ดังแสดงในภาพที่ 4.9 ประกอบด้วย



ภาพที่ 4.9 คุณสมบัติเพิ่มเติมขององค์ประกอบประเภท Assembly Operation

#### 5. Split Point ประกอบด้วย 2 ส่วนหลัก คือ

5.1. Successor ใช้ในการจัดลำดับการผ่านออกจากองค์ประกอบโดยให้ผู้ใช้เลือกองค์ประกอบที่ต้องการ จากนั้นให้เลือกปุ่มลูกศรขึ้น หรือลง เพื่อจัดลำดับ ดังภาพที่ 4.10



เอกสารนี้เป็นเอกสารที่สงวนภาพที่ 4.10 คุณสมบัติขององค์ประกอบประเภท Split Point ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

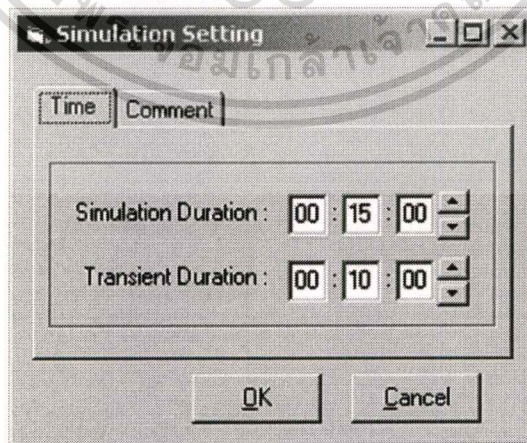
## 5.2. คุณสมบัติต่าง ๆ ดังแสดงในภาพที่ 4.11



ภาพที่ 4.11 คุณสมบัติเพิ่มเติมขององค์ประกอบประเภท Split Point

### 4.2.5 การพัฒนาส่วนของการกำหนดการประมวลผล

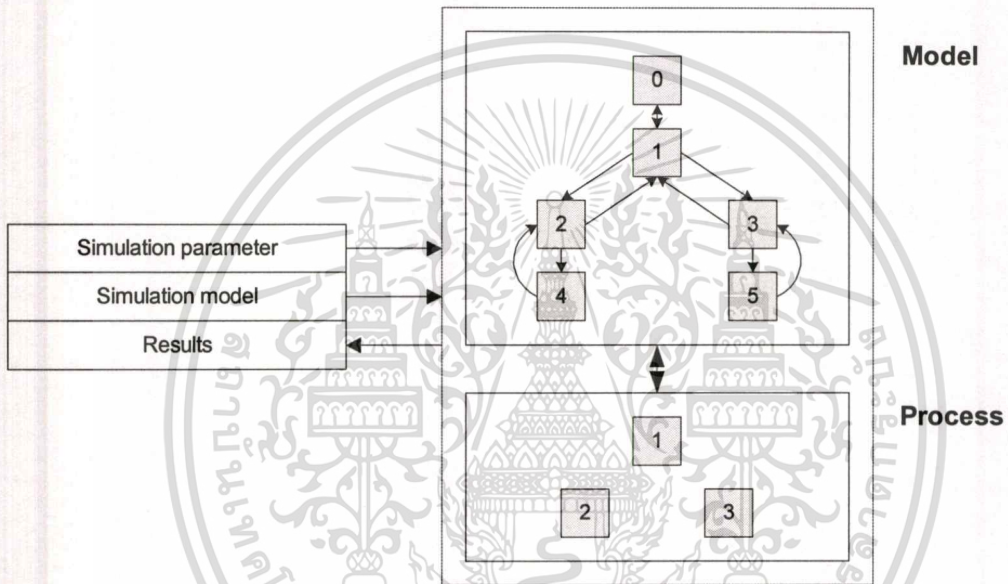
เป็นส่วนที่พัฒนาขึ้นเพื่อให้ผู้ใช้กำหนดระยะเวลา และหมายเหตุในการประมวลผลดังภาพ



ภาพที่ 4.12 กำหนดระยะเวลาในการประมวลผล

### 4.3 การพัฒนาส่วน Simulation

ในการพัฒนาส่วน Simulation นั้นจะใช้ Simulation Library ของ CSIM โดยแยกประเภทของข้อมูลเป็น 2 ส่วนคือ ส่วนที่เก็บข้อมูลขององค์ประกอบ โดยจะอาศัยข้อมูลจาก Simulation model ในการสร้างความสัมพันธ์ขององค์ประกอบ และส่วนที่ใช้เก็บข้อมูลของ Process ซึ่งจะใช้ในการประมวลผล โดยอาศัย Simulation parameter ในการแบ่งช่วงเวลาในการประมวลผล ดังแสดงในภาพที่ 4.13



ภาพที่ 4.13 ข้อมูลในการประมวลผล

จากนั้นจะทำการสร้าง Conditional Entities เพื่อใช้ในการตรวจสอบสถานะของ Process ซึ่งมี 3 สถานะคือ

1. Idle เป็นสถานะที่ Process ว่างและพร้อมที่จะทำงาน โดยจะทำการตรวจสอบ Buffer ของตนที่มีการเชื่อมต่อ ถ้ามีการเชื่อมต่อกับ Queue และ Buffer ว่างก็จะนำชิ้นส่วนจาก Queue มาเก็บไว้ใน Buffer และถ้ามีชิ้นส่วนครบถ้วนก็จะเปลี่ยนสถานะเป็นสถานะ Process

2. Process เป็นสถานะของการทำงาน ซึ่งจะเปลี่ยนสถานะเป็นสถานะ Finish โดยเว้นระยะเวลาเท่ากับ Process time รวมกับ Delay time

3. Finish เป็นสถานะในการส่งชิ้นส่วนไปยัง Process ต่อไป โดยจะทำการเลือกเส้นทางที่ควรจะส่งต่อไปในกรณีที่มีการแยกเส้นทาง และตรวจสอบ Buffer ในกรณีที่เชื่อมต่อกับ Server โดยตรง หรือตรวจสอบขนาดของ Queue ในกรณีที่เชื่อมต่อกับ Queue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นจะทำการสร้าง Schedule Entities เพื่อส่งวัตถุคิบบหรือชิ้นส่วนเข้าสู่แบบจำลองตลอดระยะเวลาที่ได้กำหนดไว้ในส่วน Simulation parameters ในระหว่างที่ Entities ผ่านไปยัง Process ต่าง ๆ ระบบก็จะเก็บข้อมูลที่เกิดขึ้นไว้ เพื่อนำไปคำนวณหลังจากครบกำหนดเวลาในการประมวลผลแล้ว

#### 4.4 การพัฒนาส่วน Report

ส่วน Report มีหน้าที่ในการจัดการเกี่ยวกับรายงานผลจากการประมวลผล ซึ่งจะรับข้อมูลมาจากส่วน Model และส่วน Simulation นอกจากนั้นยังจัดการเกี่ยวกับเพิ่มข้อมูลรายงาน ซึ่งจะรับชื่อของเพิ่มข้อมูลรายงานมาจากผู้ใช้งาน ซึ่งได้แบ่งออกเป็น 2 ส่วนหลักคือส่วนจัดการเพิ่มข้อมูลรายงานและส่วนสร้างและแสดงรายงาน

##### 4.4.1 การพัฒนาส่วนจัดการเพิ่มข้อมูลรายงาน

เนื่องจากการใช้งานจริง ผู้ใช้ระบบอาจมีความต้องการที่จะนำรายงานที่เคยได้รับ นำกลับมาใช้เปรียบเทียบ กับรายงานที่ได้รับจากแบบจำลองใหม่ที่พัฒนาขึ้น ซึ่งจะช่วยให้ผู้ใช้สามารถนำรายงานของแบบจำลองที่เคยพัฒนา มาเปรียบเทียบกับแบบจำลองใหม่ ซึ่งระบบจะจัดเก็บแบบจำลองที่ได้พัฒนาขึ้น ในรูปแบบของ Text File (\*.smr) โดยมีรูปแบบ ดังนี้

บรรทัดที่ 1 จะเก็บรายละเอียดซึ่งเป็นส่วนที่ใช้บอกว่าเป็นแบบจำลอง ของระบบใด

บรรทัดที่ 2 จะเก็บรายละเอียดของแบบจำลอง

บรรทัดที่ 3 จะเก็บค่า ระยะเวลาที่ใช้ในการทำงานทั้งหมด, ค่าใช้จ่ายรวม และ เวลาที่ใช้ในการทำงาน โดยเฉลี่ย

บรรทัดที่ 4 จนถึงท้ายสุด จะเป็นรายละเอียด Component ที่มีในรายงาน ซึ่งจะประกอบไปด้วย Queue, Server, Assemble และ Terminal Point โดยมีรูปแบบในการจัดเก็บตามลำดับ ดังนี้

1. Component ID เป็นหมายเลขที่ระบบสร้างขึ้น โดยจะอยู่ในรูปแบบของ "ONNN" (O : เป็นประเภทขององค์ประกอบ โดยจะมีค่าเป็น 0-5, NNN : เป็นหมายเลขลำดับขององค์ประกอบ)
2. Name เป็นชื่อขององค์ประกอบซึ่งผู้ใช้สามารถที่จะ แก้ไขได้ภายหลัง
3. Label เป็นรายละเอียดขององค์ประกอบซึ่งผู้ใช้สามารถที่จะ แก้ไขได้ภายหลัง
4. ประกอบด้วยค่าได้จากแบบจำลอง โดยจะจำแนกตามประเภทขององค์ประกอบได้ ดังนี้

- Queue ประกอบด้วย เวลารอคอยเฉลี่ย, ขนาดสูงสุดของ Queue, ขนาดต่ำสุดของ Queue, ขนาดเฉลี่ยของ Queue และ จำนวนที่ Lost ตามลำดับ
- Server ประกอบด้วย เวลาที่ให้บริการ, เวลาที่ว่าง, เปอร์เซนต์ที่ว่าง, จำนวนที่ Lost และ จำนวนที่ Served ตามลำดับ
- Assembly ประกอบด้วย เวลาที่ให้บริการ, เวลาที่ว่าง, เปอร์เซนต์ที่ว่าง, จำนวนที่ Lost และ จำนวนที่ Served ตามลำดับ
- Terminal Point ประกอบด้วย จำนวนผลิตภัณฑ์, จำนวนชิ้นส่วนที่ใช้ไป

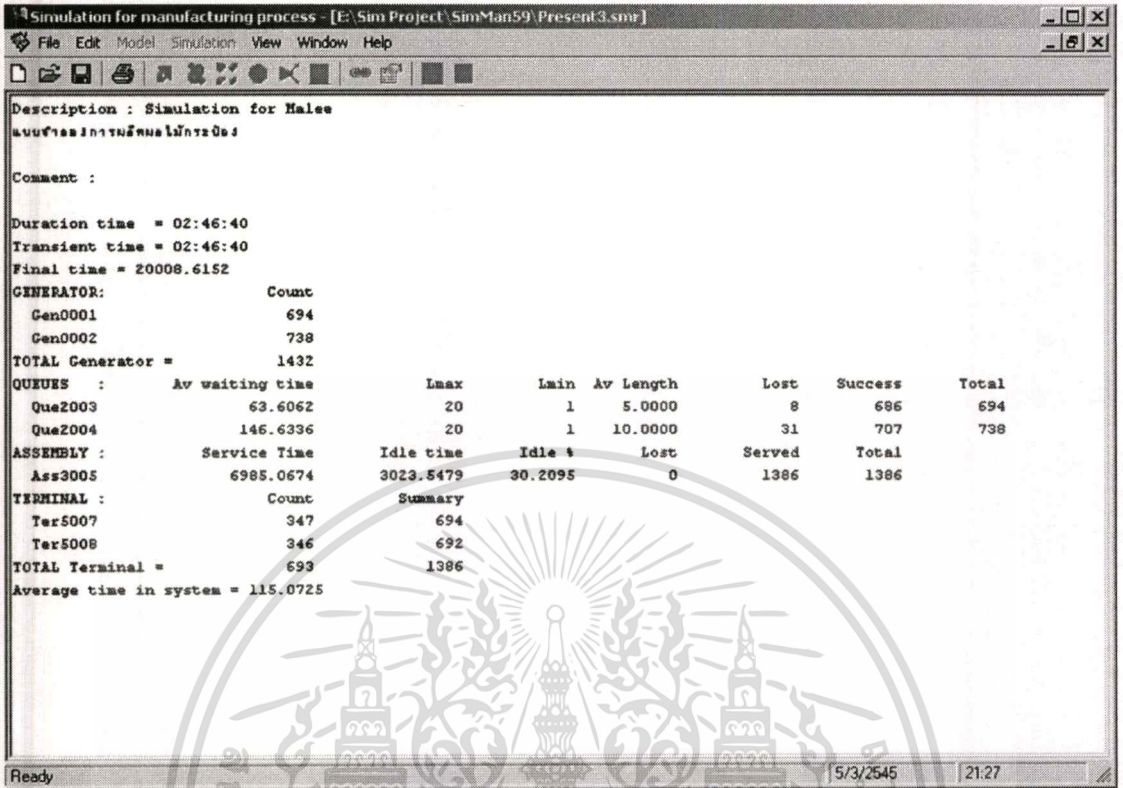
นอกจากการจัดเก็บรายงานเพื่อใช้ในการเปรียบเทียบ กับระบบงานที่พัฒนาใหม่แล้ว ในบางครั้งผู้ใช้ระบบอาจมีความต้องการที่จะนำรายงานที่ได้จากแบบจำลอง ไปใช้ในการนำเสนอในรูปแบบของไฟล์ด้วย ดังนั้นระบบจึงได้ออกแบบให้สามารถรองรับการทำงานดังกล่าวได้ ซึ่งสามารถทำได้โดย

1. เลือกรายงานของแบบจำลองที่ต้องการสร้างรายงานการนำเสนอ
2. เลือกที่ Menu File แล้วเลือกที่ Export Report
3. เลือกชื่อ และ โฟลเดอร์ที่ต้องการจัดเก็บ จากนั้นเลือก OK

โปรแกรมจะทำการจัดเก็บรายงานดังกล่าวลงในไฟล์ที่เลือกไว้ ในรูปแบบของ รายงานที่ได้รับจากแบบจำลอง

#### 4.4.2 การพัฒนาส่วนสร้างและแสดงรายงาน

เป็นการพัฒนาส่วนที่ใช้ในการแสดงผลของการทำ Simulation ในรูปแบบของรายงาน โดยจะนำข้อมูลขององค์ประกอบที่ได้รับมาจากการ Simulation หรือข้อมูลที่ได้เก็บบันทึกไว้มาจัดเรียงให้เป็นรูปแบบรายงาน ดังภาพที่ 4.14



ภาพที่ 4.14 รายงานจากการประมวลผล

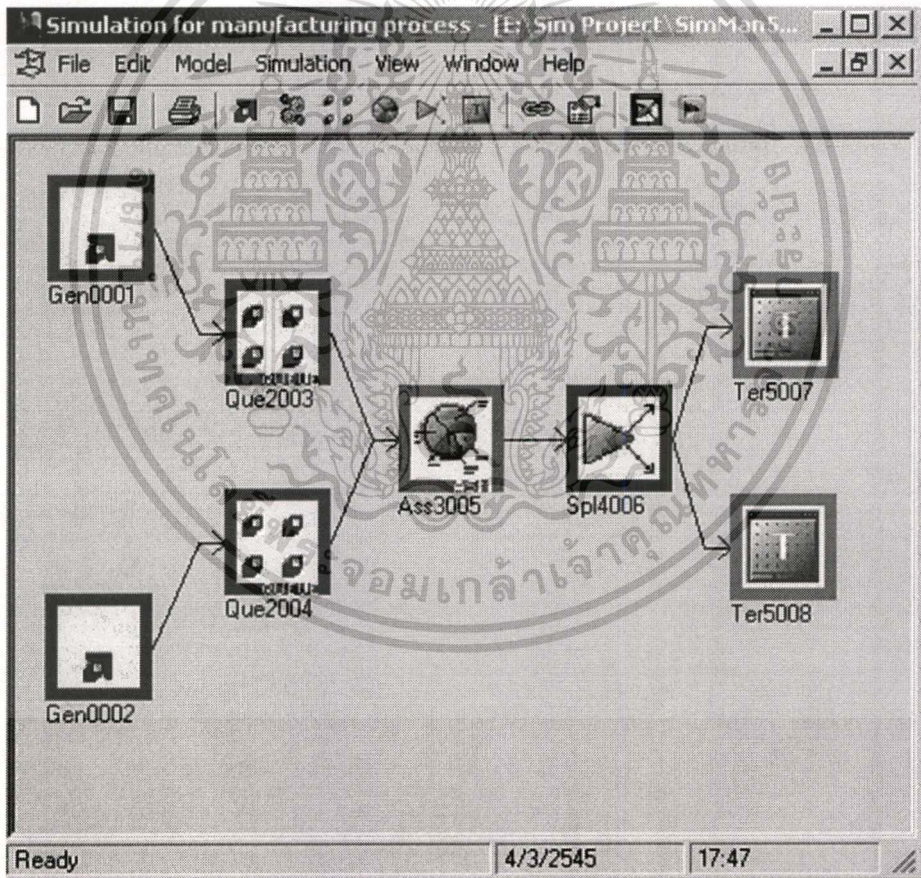
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### การทดสอบระบบ

#### 5.1 วิธีการทดลอง

จากการพัฒนาโปรแกรมระบบแบบจำลองในกระบวนการผลิตแล้ว จะทำการทดสอบโปรแกรมโดยการสร้างแบบจำลองในแบบ Queuing Model และการแยกสายการผลิต เพื่อตรวจสอบการทำงานของ Assembly ซึ่งเป็นผู้ให้บริการในระบบ ดังภาพที่ 5.1



ภาพที่ 5.1 แบบจำลองในการทดสอบ

โดยกำหนดเงื่อนไขในการประมวลผลดังนี้

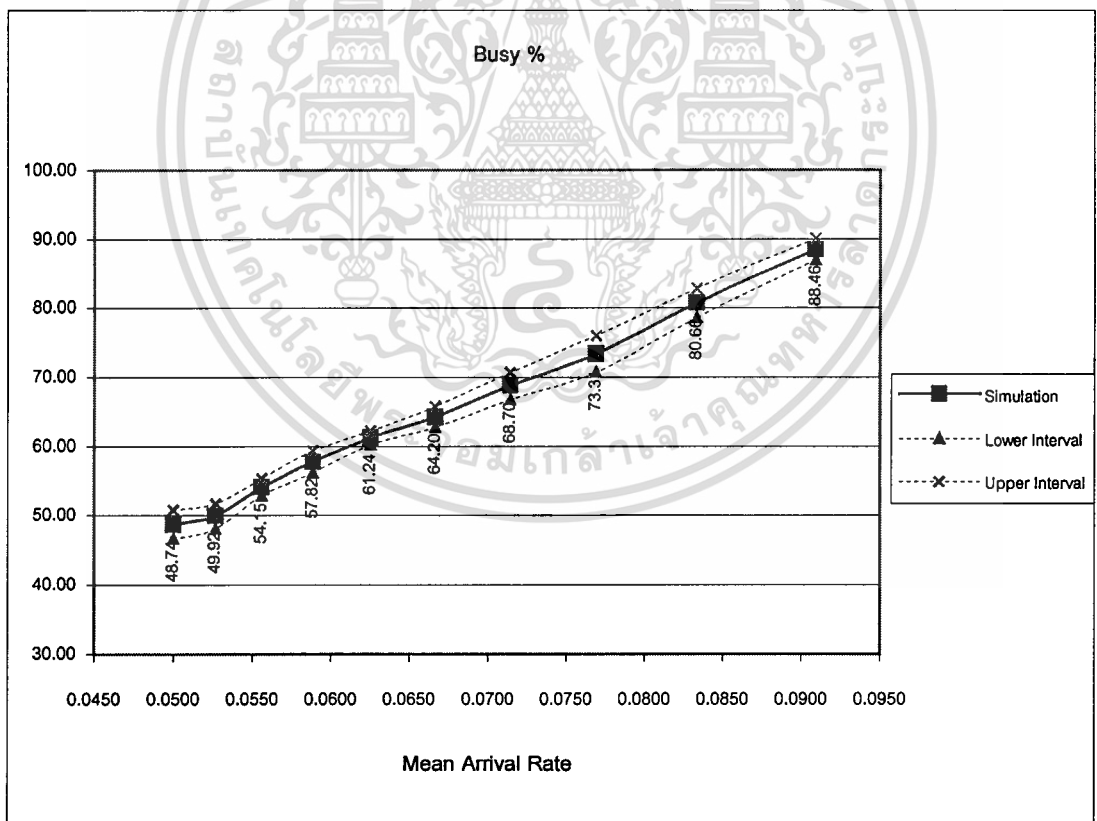
1. Simulation time เท่ากับ 5 ชั่วโมง 33 นาที 20 วินาที (20000 วินาที)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Duration time เท่ากับ 2 ชั่วโมง 46 นาที 40 วินาที (10000 วินาที)
3. Transient time เท่ากับ 2 ชั่วโมง 46 นาที 40 วินาที (10000 วินาที)
4. Mean interarrival time เท่ากับ 11 วินาที
5. Mean process time เท่ากับ 10 วินาที ในระบบที่ 1 และ 8 วินาทีในระบบที่ 2
6. ทำการทดสอบทั้งหมด 10 Cases โดยจะเพิ่ม Mean interarrival time ขึ้นครั้งละ 1 วินาที มีค่าตั้งแต่ 11 วินาทีถึง 20 วินาที และทำการเก็บข้อมูล 10 ครั้งในแต่ละ Case

## 5.2 ผลการทดลอง

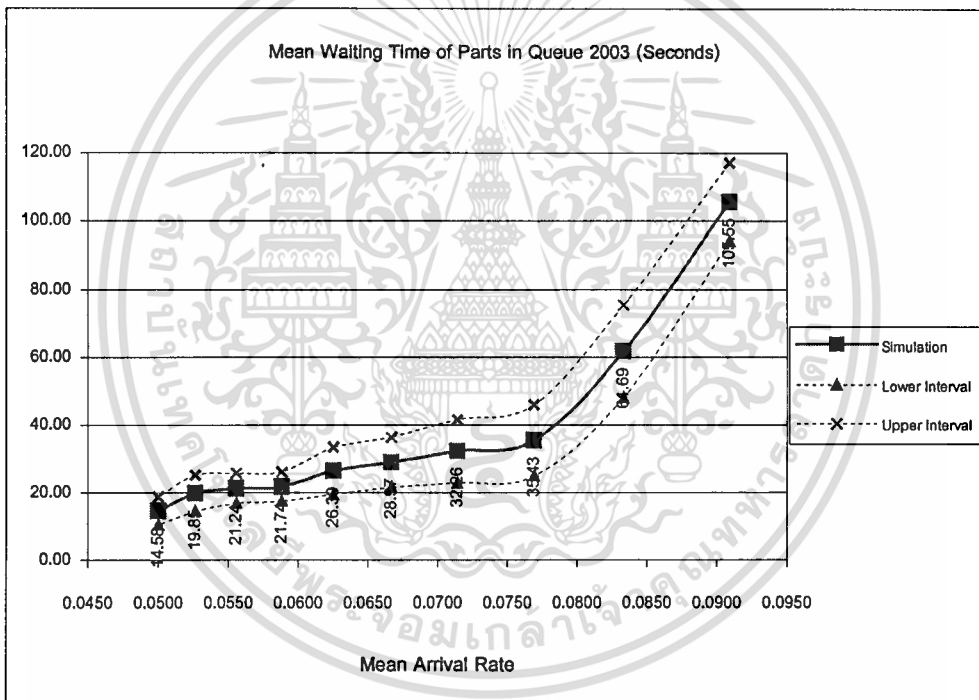
เมื่อนำข้อมูลของ Assembly ที่ได้จากการประมวลผลในระบบที่ 1 มาคำนวณหาค่าเฉลี่ยของ Busy%, Variance, Standard deviation และ Confident interval ที่ความเชื่อมั่น 90 เปอร์เซ็นต์ แล้วนำมาสร้างกราฟระหว่าง Busy% กับ Mean arrival rate จะได้ดังภาพที่ 5.2



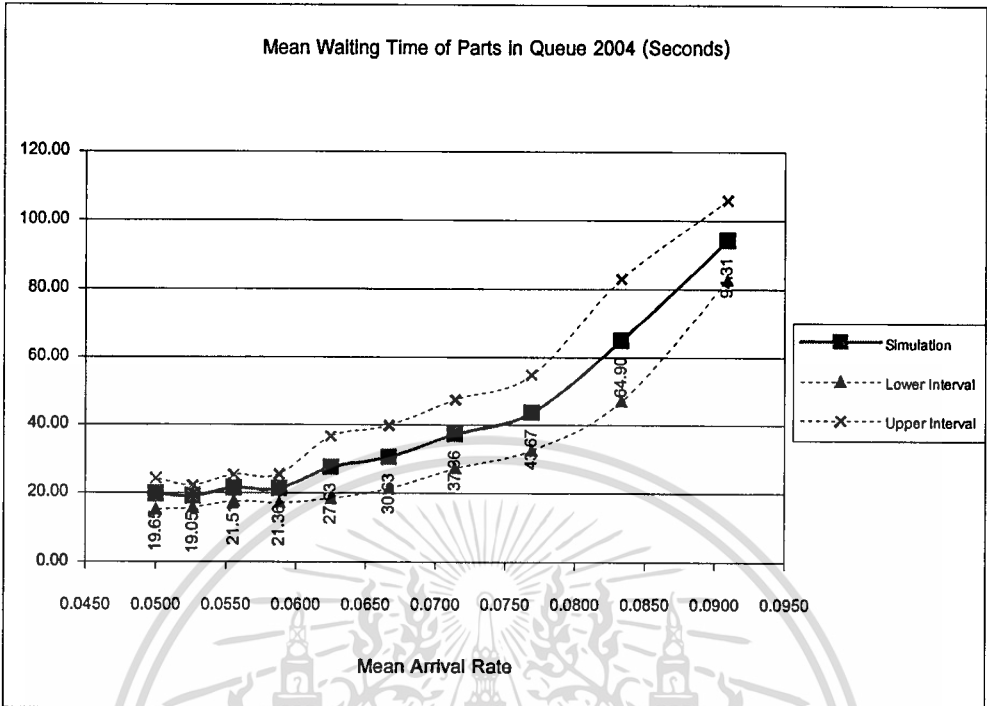
ภาพที่ 5.2 สรุปการทำงานของ Assembly ในระบบที่ 1

จากภาพแสดงให้เห็นว่า Mean arrival rate มีความสัมพันธ์แบบแปรผันตรงกับการอัตราการทำงานของ Server เมื่อค่าของ Mean arrival rate ยิ่งมากอัตราการการทำงานก็จะยิ่งมากขึ้น ซึ่งถ้าเราต้องการให้ Server ไม่ทำงานหนักจนเกินไป เราสามารถมั่นใจได้ 90 เปอร์เซ็นต์ที่ Mean arrival rate มีค่าประมาณ 0.0775 ว่า Server จะทำงานไม่เกิน 70 เปอร์เซ็นต์

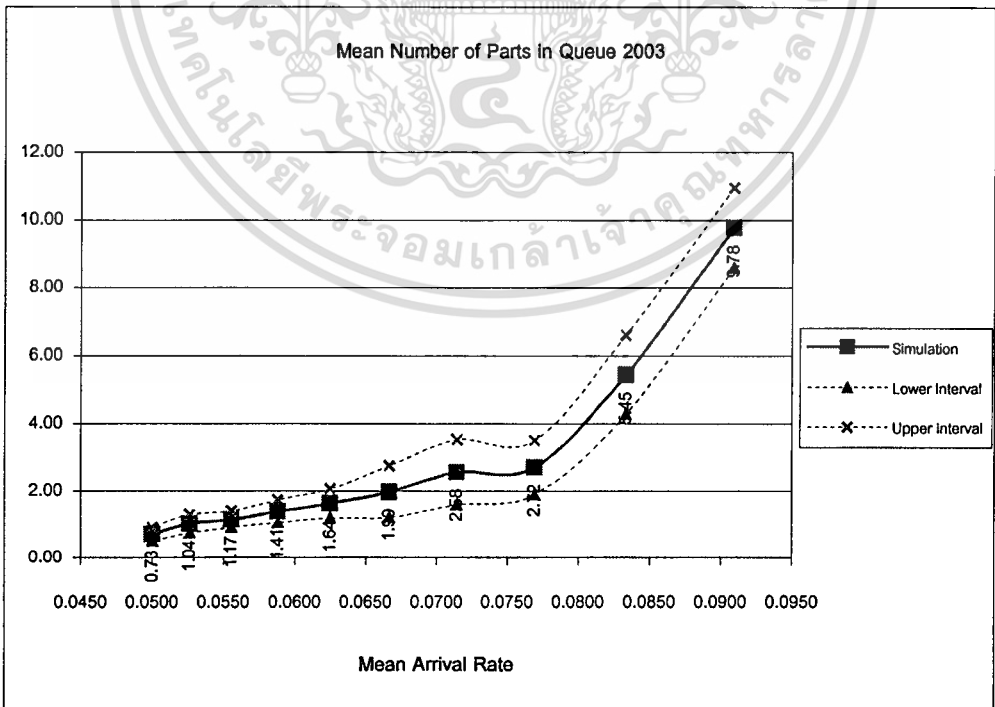
ในส่วนของ Queue 2003 และ Queue 2004 เราสามารถนำผลที่ได้จากการประมวลผลของระบบที่ 1 มาคำนวณหาค่าเฉลี่ยของ Average waiting time of parts in queue, Number of parts in queue, Variance, Standard deviation และ Confident interval ที่ความเชื่อมั่น 90 เปอร์เซ็นต์ แล้วนำมาสร้างกราฟเทียบกับ Mean arrival rate ได้ดังภาพ



ภาพที่ 5.3 Mean Waiting Time of Parts in Queue 2003

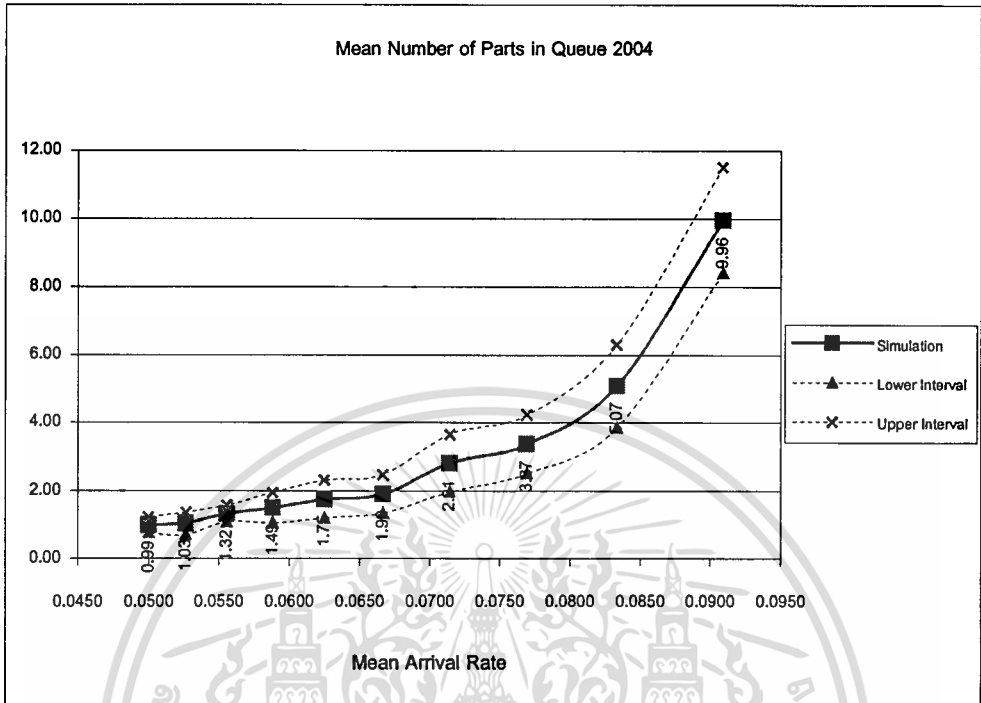


ภาพที่ 5.4 Mean Waiting Time of Parts in Queue 2004



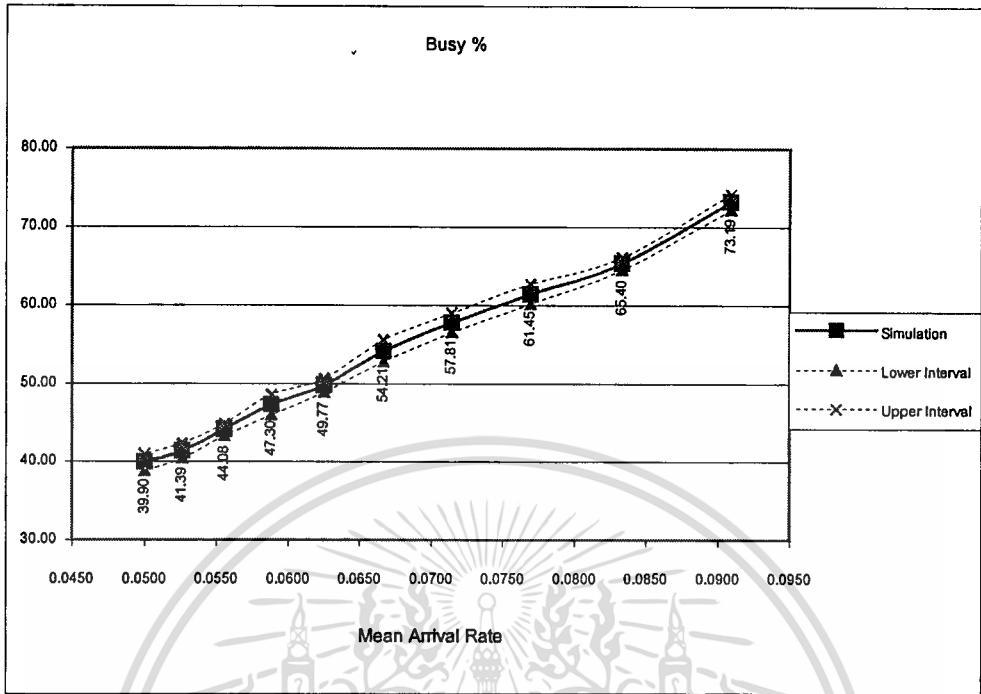
ภาพที่ 5.5 Mean Number of Parts in Queue 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

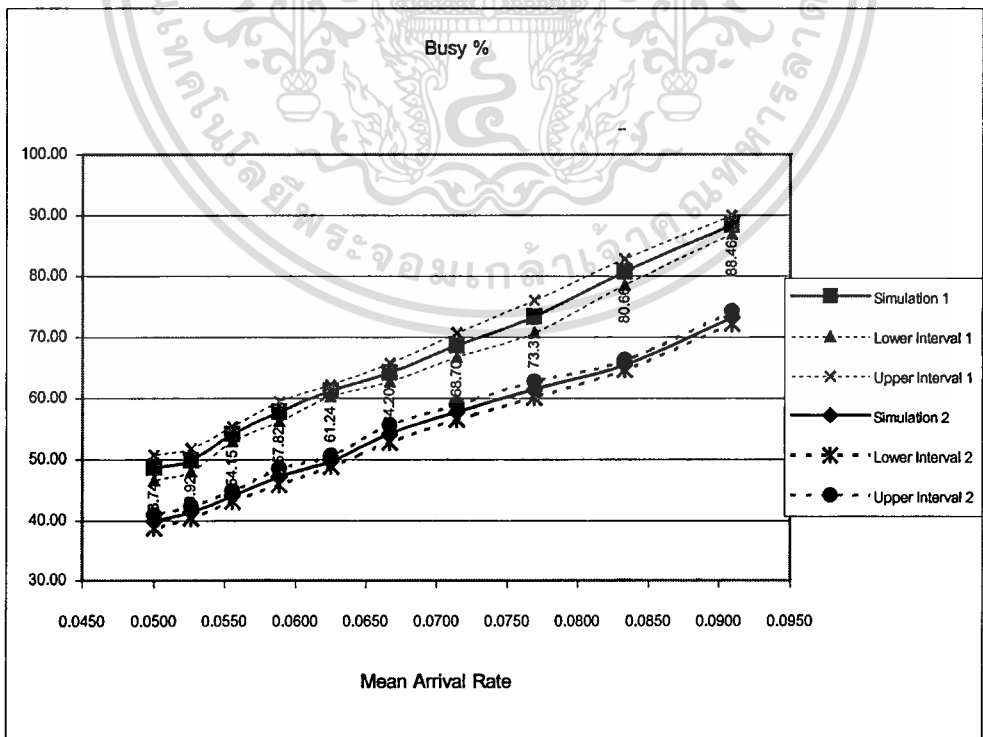


ภาพที่ 5.6 Mean Number of Parts in Queue 2004

เมื่อนำข้อมูลของ Assembly ที่ได้จากการประมวลผลในระบบที่ 2 มาคำนวณหาค่าเฉลี่ยของ Busy%, Variance, Standard deviation และ Confident interval ที่ความเชื่อมั่น 90 เปอร์เซ็นต์ แล้วนำมาสร้างกราฟระหว่าง Busy% กับ Mean arrival rate จะได้ดังภาพที่ 5.7 และเมื่อนำข้อมูลของทั้ง 2 ระบบมาเปรียบเทียบกันดังภาพที่ 5.8 จะได้ว่าเมื่อทำการปรับปรุงระบบโดยการลด Mean process time ลงจะทำให้ระบบสามารถรองรับงานได้มากขึ้น ซึ่งถ้าเราต้องการให้ Server ทำงานไม่เกิน 70 เปอร์เซ็นต์ในระบบที่ 1 จะทำงานได้ที่ Mean arrival rate ประมาณ 0.0775 แต่ในระบบที่ 2 จะสามารถทำงานได้ที่ Mean arrival rate ประมาณ 0.0900



ภาพที่ 5.7 สรุปการทำงานของ Assembly ในระบบที่ 2



ภาพที่ 5.8 เปรียบเทียบการทำงานของ Assembly ในระบบที่ 1 และระบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### สรุป

#### 6.1 สรุปผลการพัฒนาโครงการ

ระบบแบบจำลองในกระบวนการผลิตสามารถแบ่งการทำงานออกเป็น 3 ส่วนคือ

ส่วนที่ 1 ส่วนติดต่อผู้ใช้งาน เพื่อสร้างและแสดงแบบจำลองของระบบ

ส่วนที่ 2 ส่วนประมวลผลแบบจำลอง เพื่อจำลองการทำงานและสรุปผลการทำงานของระบบ

ส่วนที่ 3 ส่วนรายงานผลที่ได้จากการประมวลผลแบบจำลอง

6.1.1 การวิเคราะห์และการออกแบบระบบใช้ Data Flow Diagram ในการอธิบายการทำงานของแต่ละส่วนต่าง ๆ

6.1.2 การพัฒนาระบบเป็นแบบ Standalone โดยใช้ GUI ในส่วนติดต่อกับผู้ใช้งานและเรียกใช้งานส่วนประมวลผลแบบจำลองในรูปแบบของ DLL

6.1.3 โปรแกรมที่พัฒนาขึ้นสามารถนำมาใช้ในการทดสอบประสิทธิภาพของระบบ โดยทำการเปรียบเทียบผลของการทำงานในแต่ละองค์ประกอบจากรายงานที่ได้ และสามารถสรุปการทำงานโดยรวมได้ดังนี้

1. ระบบสามารถสร้าง, เรียกใช้งาน, แก้ไข และบันทึกแบบจำลองได้อย่างถูกต้อง
2. ระบบสามารถบันทึก, เรียกใช้งานรายงานได้อย่างถูกต้อง
3. ระบบสามารถบันทึก Entities ในการประมวลผลได้อย่างถูกต้อง
4. ระบบสามารถพิมพ์แบบจำลอง และรายงานได้อย่างถูกต้อง

#### 6.2 ข้อควรปรับปรุงเพิ่มเติม

1. เนื่องจากข้อจำกัดในเรื่องของเวลาในการพัฒนาทำให้ Source Code ของโปรแกรมบางส่วนยังไม่มีคุณภาพ และในส่วนสร้างแบบจำลองยังใช้งานได้ไม่สะดวกเท่าที่ควร เช่นการสร้างความสัมพันธ์ ดังนั้นควรมีการทำ Optimize Source Code และปรับปรุงในส่วน User Interface ให้ใช้งานง่ายขึ้น

2. ควรพัฒนาส่วนวิเคราะห์ และเปรียบเทียบแบบจำลอง เพื่อใช้พิจารณาความเหมาะสมของแบบจำลอง รวมทั้งการปรับปรุงประสิทธิภาพการทำงานของระบบ

3. ควรพัฒนาในส่วนควบคุมการประมวลผลให้สามารถติดตามการทำงานได้ เพื่อช่วยในการศึกษาการทำงาน of ระบบ

### 6.3 ปัญหาและข้อเสนอแนะ

1. เนื่องจากพัฒนาในรูปแบบของ DLL (Dynamic-Link Library) ทำให้การตรวจสอบข้อผิดพลาดทำได้ลำบาก ซึ่งอาจจะพัฒนาในรูปแบบอื่นที่สามารถติดตามการทำงานได้เช่น พัฒนาเป็น ActiveX Component, พัฒนาทั้งส่วนติดต่อผู้ใช้งานและส่วนประมวลผลด้วย MS Visual C++
2. ระบบมีการเรียกใช้งานหน่วยความจำในการประมวลผล ซึ่งภายหลังจากการประมวลผลแล้วจะคืนหน่วยความจำได้ไม่ทั้งหมด ทำให้เมื่อประมวลผลในระยะเวลาต่างๆ จะทำให้หน่วยความจำของระบบหมดไป จึงควรตรวจสอบวิธีการเรียกใช้ DLL เพื่อให้คืนหน่วยความจำทั้งหมดหลังจากเลิกใช้งาน DLL แล้ว



## บรรณานุกรม

สุรศักดิ์ พงศ์ธนาพานิช. 2542. การเขียนโปรแกรมด้วย Visual Basic 6.0 ระดับสูง. กรุงเทพฯ :  
โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย.

B. Wu. 1992. **Manufacturing Systems Design and Analysis**. New York : Chapman & Hall.

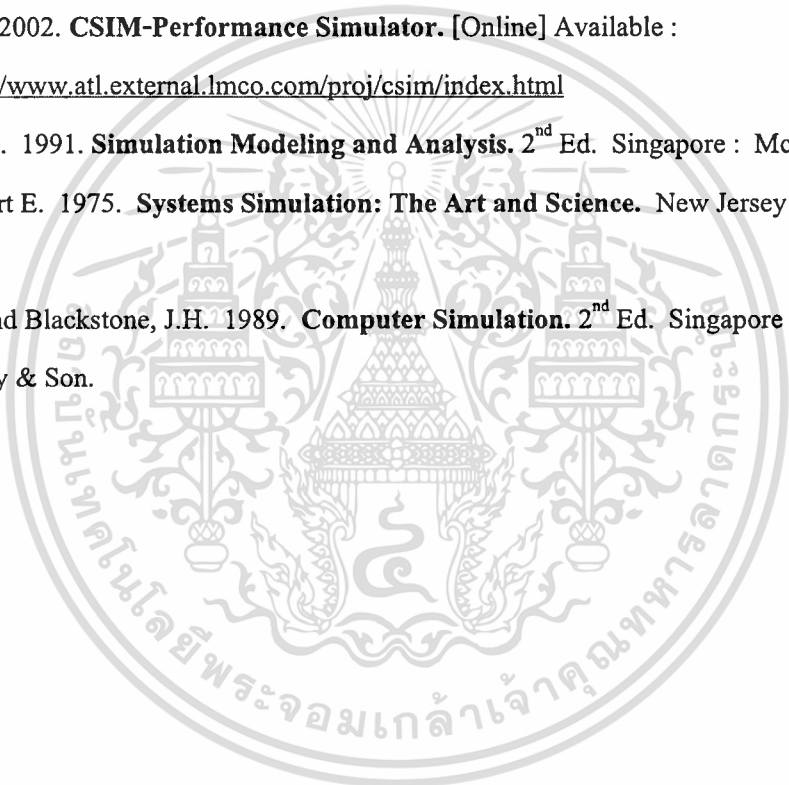
CSIM Library. 2002. **CSIM-Performance Simulator**. [Online] Available :

<http://www.atl.external.lmco.com/proj/csim/index.html>

Law, Averill M. 1991. **Simulation Modeling and Analysis**. 2<sup>nd</sup> Ed. Singapore : McGraw-Hill.

Shannon, Robert E. 1975. **Systems Simulation: The Art and Science**. New Jersey : Prentice-  
Hall.

Watson, H.J. and Blackstone, J.H. 1989. **Computer Simulation**. 2<sup>nd</sup> Ed. Singapore : John  
Wiley & Son.



## ประวัติผู้เขียน

- ชื่อ** นายปริญญา งามเกียรติขจร
- สถานที่เกิด** นครปฐม
- การศึกษา** ระดับมัธยมศึกษา โรงเรียนสาธิตมหาวิทยาลัยศิลปากร  
ระดับปริญญาตรี สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยศิลปากร
- การทำงาน** พนักงานควบคุมงานอาวุโส ธนาคารไทยพาณิชย์ จำกัด มหาชน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้