การสร้างแบบจำลองระบบขนส่งสำหรับเหมืองหิน

Haulage System Simulation in Quarries

เศกสรร บุญสำราญ

Sakeson Boonsamran

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมเหมืองแร่
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Mining Engineering
Prince of Songkla University
2556

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์ การสร้างแบบจำลองระบบขนส่งสำหรับเหมืองหิน

Haulage System Simulation in Quarries

ผู้เขียน นายเศกสรร บุญสำราญ

สาขาวิชา วิศวกรรมเหมืองแร่

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

….....

(ผู้ช่วยศาสตราจารย์ ดร.มนูญ มาศนิยม)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

….....

(รองศาสตราจารย์ ดร.สุรพล อารีย์กุล)

คณะกรรมการสอบ

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร.พิษณุ บุญนวล)

.....กรรมการ

(รองศาสตราจารย์ ดร.สุรพล อารีย์กุล)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.มนูญ มาศนิยม)

.....กรรมการ

(รองศาสตราจารย์ ดร.บุญสม ศิริบำรุงสุข)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาตรมหาบัณฑิต สาขาวิชาวิศวกรรมเหมืองแร่

.....

(รองศาสตราจารย์ ดร.ธีระพล ศรีชนะ)

คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้เป็นผลมาจากการศึกษาวิจัยของนักศึกษาเอง และขอขอบคุณผู้มีส่วนเกี่ยวข้องทุกท่านไว้ ณ ที่นี้

ลงชื่อ _____

(ผู้ช่วยศาสตราจารย์ ดร.มนูญ มาศนิยม)

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ลงชื่อ _____

(นายเศกสรร บุญสำราญ)

นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ _____

(นายเศกสรร บุญสำราญ)

นักศึกษา

ชื่อวิทยานิพนธ์ การสร้างแบบจำลองระบบขนส่งสำหรับเหมืองหิน
ผู้เขียน นายเศกสรร บุญสำราญ
สาขาวิชา วิศวกรรมเหมืองแร่
ปีการศึกษา 2555

บทคัดย่อ

งานวิจัยนี้เป็นการประเมินระบบการขนส่งหินจากหน้าเหมืองมายังเครื่องย่อยโดยการสร้างแบบจำลองด้วยโปรแกรมสำเร็จรูป Arena และสร้างเครื่องมือขึ้นมาใช้ประเมินร่วมอีกหนึ่งชิ้นโดยใช้ภาษาจาวาในการเขียน ลักษณะการทำงานจะพิจารณาเครื่องจักร 3 ชนิด ที่ทำงานต่อเนื่องกัน คือ รถตัก รถบรรทุกและเครื่องย่อย เพื่อให้กระบวนการตักหินของรถตักและกระบวนการย่อยหินของเครื่องย่อยเป็นไปอย่างต่อเนื่องในเวลาวงรอบของเครื่องจักรนั้น ๆ และลดปัญหาการรอคอยที่กระบวนการลำเลียงหินโดยรถบรรทุก ผู้วิจัยใช้กรณีศึกษาของบริษัทปูนซิเมนต์ไทย (ทุ่งสง) จำกัด และบริษัทปูนซิเมนต์ไทย (ท่าหลวง) จำกัด ผลการวิจัยพบว่า โรงงานทุ่งสง ได้จำนวนรถบรรทุกพอดีกับรถตักและเครื่องย่อย มีค่ามากที่สุดเท่ากับ 15 คัน ส่วนโรงงานท่าหลวงได้เท่ากับ 8 คัน โดยรถบรรทุกใช้เวลารอเฉลี่ยในแต่ละรอบที่รถตักและเครื่องย่อยน้อยที่สุดเท่ากับ 0.99 และ 1.24 นาที สำหรับโรงงานทุ่งสง และเท่ากับ 0.00 และ 1.00 นาที สำหรับโรงงานท่าหลวง นอกจากนี้ สัดส่วนการทำงานของรถตักมีค่าน่าพอใจสำหรับ โรงงานทุ่งสง คือ 97 เปอร์เซ็นต์ แต่สำหรับโรงงานท่าหลวงเท่ากับ 59 เปอร์เซ็นต์ เนื่องจากระยะทางระหว่างจุดตักและจุดเทค่อนข้างไกล

Thesis Title	Haulage System Simulation in Quarries
Author	Mr.Sakeson Boonsamran
Major Program	Mining Engineering
Academic Year	2012

ABSTRACT

This research evaluated the system of convey blasted rock from loading areas to crushers by simulations with Arena software and the Java programming evaluate tool. The research focuses on three types of machines that continuous working are loaders, trucks and crushers. This study require process of loading and crushing to be continuous in their cycle time, also reduce cycle time in transport by trucks. Researcher used case study of The Siam Cement (Thungsong) Co., Ltd. and The Siam Cement (Thaluang) Co., Ltd. The results shown that number of trucks for Thungsong Plant were match for loaders and crushers, the greater was 15 trucks and for Thaluang Plant was 8 trucks. The least waiting time of trucks at loaders and at crushers were 0.99 and 1.24 minutes for Thungsong Plant and 0.00 and 1.00 minutes for Thaluang Plant. Finally, the loader portion of working was better than regular incase of Thungsong Plant that was 97% maximum but for Thaluang Plant was 59%, because of distance between loading area and dumping area is greater than Thungsong Plant.

สารบัญ

	หน้า
หน้าอนุมัติ	(2)
บทคัดย่อ	(5)
กิตติกรรมประกาศ	(7)
สารบัญ	(8)
รายการตาราง	(10)
รายการภาพประกอบ	(12)
บทที่ 1 บทนำ	
1.1 ความสำคัญและที่มาของงานวิจัย	1
1.2 เอกสารและงานวิจัยที่เกี่ยวข้อง	2
1.3 วัตถุประสงค์	6
1.4 ข้อมูลเบื้องต้นสำหรับงานวิจัย	6
1.5 ทฤษฎีระบบแถวคอย	9
1.6 วิธีการวิจัย	18
บทที่ 2 วิธีการวิจัย	
2.1 การเก็บข้อมูลภาคสนาม	20
2.2 วิธีการสร้างแบบจำลองด้วย Arena	23
2.3 วิธีการสร้างแบบจำลองด้วยภาษาจาวา	32
2.4 วัสดุและอุปกรณ์	37

สารบัญ (ต่อ)

	หน้า
บทที่ 3 ผลการวิจัย	
3.1 กรณีที่ 1 : จำลองระบบด้วย Arean สำหรับทุ่งสง	42
3.2 กรณีที่ 2 : จำลองระบบด้วย Arean สำหรับท่าหลวง	52
3.3 กรณีที่ 3 : จำลองระบบด้วยจาวา สำหรับทุ่งสง	63
3.4 กรณีที่ 4 : จำลองระบบด้วยจาวา สำหรับท่าหลวง	67
บทที่ 4 วิเคราะห์ผลการวิจัย	68
บทที่ 5 สรุปผลการวิจัย	75
เอกสารอ้างอิง	77
ภาคผนวก ก โปรแกรม Arana	79
ภาคผนวก ข โปรแกรมจาวา	144
ประวัติผู้เขียน	215

รายการตาราง

หน้า

ตารางที่ 1.1 เปรียบเทียบรูปแบบที่นำเสนอกับข้อมูลย้อนหลัง (กรัณย์ กันพ้นภัย 2547) 2

ตารางที่ 1.2 สรุปผลแบบจำลองการสร้างลานวางตู้สินค้า (Kang และคณะ 2001) 5

ตารางที่ 1.3 ผลการประมวลผลด้วยแบบจำลอง จำนวน 100 เที่ยว (Cetin และคณะ 2001) 5

ตารางที่ 1.4 เครื่องจักรที่เกี่ยวข้องกับการสร้างแบบจำลองของบริษัท ปูนซิเมนต์ไทย (ทุ่งสง) จำกัด 7

ตารางที่ 1.5 เครื่องจักรที่เกี่ยวข้องกับการสร้างแบบจำลองของบริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด 8

ตารางที่ 2.1 ตารางสำหรับเก็บข้อมูลเวลาวงรอบของรถบรรทุก 21

ตารางที่ 2.2 ตารางสำหรับเก็บข้อมูลเวลามาถึงของรถบรรทุกที่รถตัก 22

ตารางที่ 2.3 ตารางสำหรับเก็บข้อมูลเวลามาถึงของรถบรรทุกที่เครื่องย่อย 22

ตารางที่ 3.1 ผลการวิจัยแบ่งได้เป็น 4 กรณี 42

ตารางที่ 3.2 ข้อมูลนำเข้าสำหรับเหมืองหินปูนทุ่งสง 44

ตารางที่ 3.3 จำนวนรถบรรทุกที่เหมาะสมจากการจำลองด้วย Arena สำหรับทุ่งสง 47

ตารางที่ 3.4 เวลารอของรถบรรทุกที่รถตักและเครื่องย่อยจากการจำลองด้วย Arena สำหรับทุ่งสง 49

ตารางที่ 3.5 สัดส่วนการทำงานของรถตักจากการจำลองด้วย Arena สำหรับทุ่งสง 51

ตารางที่ 3.6 ข้อมูลนำเข้าสำหรับเหมืองหินปูนท่าหลวง 53

ตารางที่ 3.7 จำนวนรถบรรทุกที่เหมาะสมจากการจำลองด้วย Arena สำาหรับท่าหลวง 57

รายการตาราง (ต่อ)

หน้า

ตารางที่ 3.8 เวลารอของรถบรรทุกที่รถตักและเครื่องย่อยจากการจำลองด้วย Arena สำหรับท่าหลวง 60

ตารางที่ 3.9 สัดส่วนการทำงานของรถตักจากการจำลองด้วย Arena สำหรับท่าหลวง 62

ตารางที่ 4.1 ข้อมูลนำเข้าสำหรับการประมวลผลด้วยโปรแกรมจาวาทุ่งสง 71

ตารางที่ 4.2 ข้อมูลนำเข้าสำหรับการประมวลผลด้วยโปรแกรมจาวาท่าหลวง 73

รายการภาพประกอบ

หน้า

รูปที่ 1.1 เครื่องจักรที่เกี่ยวข้องกับการสร้างแบบจำลองของบริษัท ปูนซิเมนต์ไทย (ทุ่งสง) จำกัด 7

รูปที่ 1.2 เครื่องจักรที่เกี่ยวข้องกับการสร้างแบบจำลองของบริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด 9

รูปที่ 1.3 ค่าตัวแปรต่าง ๆ ที่ใช้คำนวณสำหรับโมเดลของ Elbrond 15

รูปที่ 1.4 ค่า pP สำหรับสมการ (1.4) 17

รูปที่ 1.5 ค่า pA สำหรับสมการ (1.5) 17

รูปที่ 1.6 ขั้นตอนการวิจัย 18

รูปที่ 2.1 การเก็บข้อมูลภาคสนาม 20

รูปที่ 2.2 เวลาตักใส่รถบรรทุกจนเต็มของรถตักล้อยาง 25

รูปที่ 2.3 เปิดโปรแกรม Input Analyzer เพื่อวิเคราะห์การกระจายของข้อมูล 26

รูปที่ 2.4 สร้างไฟล์ใหม่โดยเลือก New 26

รูปที่ 2.5 เรียกข้อมูลเข้ามาใน Input Analyzer 27

รูปที่ 2.6 เปิดไฟล์ที่เคยบันทึกไว้เป็นนามสกุล .txt 27

รูปที่ 2.7 ผลลัพธ์จากการประมวลผลด้วย Input Analyzer 28

รูปที่ 2.8 หาค่าการกระจายของข้อมูลโดยเลือก Fit All 29

รูปที่ 2.9 ค่าการกระจายของข้อมูลที่ได้จากการประมวลผลโดย Input Analyzer 29

รูปที่ 2.10 เปิดโปรแกรม Arena เพื่อสร้าง Model 30

รูปที่ 2.11 โปรแกรม Arena 30

รายการภาพประกอบ (ต่อ)

หน้า

รูปที่ 2.12 การสร้างแบบจำลองด้วยโปรแกรม Arena 31

รูปที่ 2.13 ผลลัพธ์การทำงานของแบบจำลอง 32

รูปที่ 2.14 การจำลองระบบด้วย OpenOffice Calc ในระบบปฏิบัติการ Linux 33

รูปที่ 2.15 แผนผังการไหลของระบบ เขียนด้วย Dia 0.97.1 34

รูปที่ 2.16 สรุปขั้นตอนการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ 35

รูปที่ 2.17 รหัสภาษาจาวา ผู้ใช้สามารถเลือกจุดตัดได้ไม่เกิน 3 จุด 36

รูปที่ 2.18 ผลการประมวลผลรหัสภาษาจาวาในรูป 2.17 37

รูปที่ 2.19 โปรแกรมกระดาษคำนวณ LibreOffice Calc 3 38

รูปที่ 2.20 โปรแกรมเขียนข้อความ gedit รุ่น 3.4.1 39

รูปที่ 2.21 ภาษาจาวารุ่น 1.6 39

รูปที่ 2.22 IDE Eclipse SDK รุ่น 3.5.2 40

รูปที่ 2.23 Command line GNOME Terminal รุ่น 2.32 41

รูปที่ 3.1 แผนผังการไหลของรถบรรทุกจากการสร้างแบบจำลองด้วยโปรแกรม Arena สำหรับทุ่งสง 43

รูปที่ 3.2 แผนผังการไหลของรถบรรทุกจากการสร้างแบบจำลองด้วยโปรแกรม Arena สำหรับท่าหลวง 52

รูปที่ 3.3 โปรแกรมที่ใช้ประมวลผลผ่านคอมมานไลน์ชื่อว่า "เทอร์มินอล" 64

รูปที่ 4.1 ภาพถ่ายทางอากาศเหมืองหินปูนบริษัท ปูนซิเมนต์ไทย (ทุ่งสง) จำกัด 69

รูปที่ 4.2 ภาพถ่ายทางอากาศเหมืองหินปูนบริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด 70

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของงานวิจัย

กระบวนการทำเหมือง (Mining operation) ไม่ว่าจะเป็นเหมืองผิวดิน (Surface mining) หรือเหมืองใต้ดิน (Underground mining) ย่อมต้องมีการลำเลียงวัตถุดิบจากหน้าเหมืองมายังโรงแต่งแร่ เครื่องจักรที่ใช้ในการขนส่งวัตถุดิบมีหลายประเภท เช่น รถบรรทุก (Truck) สายพานลำเลียง (Belt conveyor) ปั้นจั่น (Crane) ลิฟต์ (Elevator) เป็นต้น การทำเหมืองหิน (Quarry) โดยทั่วไปใช้รถบรรทุกเป็นเครื่องจักรลำเลียง สาเหตุเพราะว่าอัตราการย้ายจุดตัก (Loading area) มีมากกว่าการทำเหมืองประเภทอื่น ดังนั้นรถบรรทุกจึงเหมาะสมที่สุด

ปัญหาหนึ่งของกระบวนการทำเหมืองหินในปัจจุบัน คือการจัดการระบบขนส่งหินระหว่างจุดตักกับจุดเท (Dumping area) เนื่องจากการวางแผนยังไม่เป็นขั้นตอนเท่าที่ควร ที่เห็นได้ชัดคือ กรณีจุดตักหลายจุด ผู้วางแผนอาจไม่แน่ใจว่าจะจัดให้รถบรรทุกเข้าไปรับหินที่จุดตักอย่างไร กล่าวคือ จะให้รถบรรทุกสุ่มจุดตัก (Random loading area) หรือตรึงจุดตัก (Fixed loading area) ภายในกะ (Shift) ทำให้เกิดปัญหาขึ้นเนื่องจากมีเวลารอ (Waiting time) ของรถบรรทุก รถตัก (Loader) และเครื่องย่อย (Crusher) ต้นทุนที่เกี่ยวข้องกับกระบวนการดังกล่าว ได้แก่ ต้นทุนการตักหินของรถตัก ต้นทุนการขนส่งหินของรถบรรทุกและต้นทุนการย่อยของเครื่องย่อย การจัดการที่ดีเกี่ยวกับกระบวนการดังกล่าวคือต้องให้การตัก การขนส่งและการย่อยหินเป็นไปอย่างต่อเนื่อง เพื่อลดเวลารอของเครื่องจักรทุกประเภทให้เหลือน้อยที่สุดและเครื่องจักรทุกประเภทในระบบทำงานเต็มกำลัง

ดังนั้น เพื่อเพิ่มประสิทธิภาพในการตักและการย่อยหิน กระบวนการระหว่างกลางคือ การขนส่งหินต้องมีประสิทธิภาพ งานวิจัยนี้จึงมุ่งไปที่การขนส่งหินโดยใช้รถบรรทุกเป็นเครื่องจักรขนส่งหินระหว่างรถตักถึงเครื่องย่อย ด้วยการสร้างเครื่องมือขึ้นมาเพื่อประเมินระบบขนส่งขององค์กร

1.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

เมื่อปี 2547 กรัณย์ กันพ้นภัย ใช้การโปรแกรมเชิงเส้น (Linear programming) และระบบแถวคอยวิเคราะห์การจัดรูปแบบการขนส่งหินปูนระหว่างหน้าเหมืองกับโรงโม่ ของบริษัท ทีพีไอ โพลีน จำกัด (มหาชน) โดยใช้โปรแกรมช่วยประเมินสองโปรแกรม คือ MATLAB และ Arena 3.0 ถ้าเปรียบเทียบระหว่างข้อมูลย้อนหลังกับรูปแบบที่นำเสนอ โดยโรงโม่มีกำลังการผลิตคงที่และวัตถุดิบมีคุณภาพเท่าเดิม จะเห็นผลชัดเจนในเรื่องของค่าเชื้อเพลิงที่ลดลงและต้นทุนรวมลดลง 22.20% ซึ่งเป็นตัวเลขที่น่าพอใจเพราะถ้ามองในภาพรวมของกระบวนการผลิตปูนซีเมนต์ทั้งระบบ การผลิตหินปูนถือเป็นกระบวนการขั้นต้นที่ต้องใช้ต้นทุนสูงมาก เมื่อเทียบกับกระบวนการถัดไปคือ การเตรียมวัตถุดิบ, การเผาปูนเม็ดและการบดปูนซีเมนต์

ผลจากการศึกษาวิจัย เป็นไปตามตารางที่ 1.1

ตารางที่ 1.1 เปรียบเทียบรูปแบบที่นำเสนอกับข้อมูลย้อนหลัง (กรัณย์ กันพ้นภัย 2547)

รายการ	ข้อมูลย้อนหลัง	รูปแบบที่นำเสนอ
กำลังการผลิต	1,450 ตัน/ชั่วโมง	1,450 ตัน/ชั่วโมง
คุณภาพวัตถุดิบ	34 %	34 %
แหล่งวัตถุดิบ	จากทั้ง 5 แหล่ง	แหล่งที่ 2 และ 5
รูปแบบการขนส่ง	ไม่มีรูปแบบที่แน่นอน	รูปแบบที่ 9
จำนวนรถตัก	เฉลี่ย 3.05 คัน	3 คัน
จำนวนรถบรรทุก	เฉลี่ย 8 คัน	7 คัน
ต้นทุนน้ำมันเชื้อเพลิง	6.08 บาท/ตัน	4.74 บาท/ตัน
ต้นทุนลดลง	-	22.20 %

การสร้างแบบจำลองระบบขนส่งนั้น ไม่ได้จำกัดอยู่เฉพาะการขนส่งภาคพื้นดินเท่านั้น สมชาย

วาณิชวศิน (2540) ได้ทำวิจัยของการขนส่งทางน้ำ เกี่ยวกับการจ่ายปูนซีเมนต์ผงลงเรือเพื่อส่งต่อไปยังสถานีสูบปลายทางของ บริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด โดยมีเป้าหมายคือ นำเอาผลลัพธ์ของอัตราการขนส่งปูนซีเมนต์ผงที่เพิ่มขึ้นมาใช้ปรับปรุงกระบวนการขนถ่ายปูนซีเมนต์ผงที่เป็นอยู่ในปัจจุบัน ใช้เครื่องมือคือ Arena รุ่น 3.10 งานวิจัยนี้ถ้าแบ่งกระบวนการทำงานตามประเภทเครื่องจักรสามารถแบ่งได้เป็น 3 กระบวนการ คือ การจ่ายปูนซีเมนต์ผงลงเรือ, การขนส่งและการสูบปูนซีเมนต์ผงขึ้นสถานี ในความเห็นของผู้วิจัยตัวแปรที่ควรจะมีผลต่อระบบมากที่สุดทั้งในแง่ของเวลาวงรอบและต้นทุนน่าจะเป็นกระบวนการต้นทางหรือไม่ก็เป็นกระบวนการปลายทางคล้าย ๆ กับระบบขนส่งของเหมืองหินที่ต้นทุนของกระบวนการต้นทางคือ รถตักมีค่ามากที่สุด ตัวแปรที่มีผลต่อระบบมากที่สุดในงานวิจัยนี้ก็คือ ความสามารถในการสูบปูนซีเมนต์ผงขึ้นจากเรือของชุดสูบปลายทางนั่นเอง มีผลทำให้อัตราการขนส่งเพิ่มขึ้นจากเดิมคิดเป็น 29.65%

งานวิจัยในเมืองไทยอีกชิ้นที่ใช้การโปรแกรมเชิงเส้นโดยวิธิซิมเพล็กซ์ เพื่อวิเคราะห์การลดค่าใช้จ่ายในการขนส่งสินค้าจากโรงงานไปยังแหล่งจำหน่ายหลายแห่งศึกษาโดย นิพนธ์ วงษ์พานิช (2525) จุดมุ่งหมายหลักก็คือต้องการลดค่าใช้จ่ายและเพิ่มกำไรให้ได้มากที่สุด โดยใช้เวลาวงรอบเป็นตัวแปรหลักในการวิเคราะห์ ผลลัพธ์ก็คือ ได้รูปแบบการขนส่งที่เป็นระบบมากขึ้นในรูปของสมการเชิงคณิตศาสตร์

ที่บราซิล Fioroni และคณะ (2008) ใช้โปรแกรมสำเร็จรูปสองโปรแกรมในการประเมินประสิทธิภาพของเครื่องจักร คือ Arena และ Lingo โดยมุ่งไปที่การทำงานของรถตักปุ๋งกี๋เสียและการจัดสรรจำนวนเที่ยวของรถบรรทุกเพื่อให้พอดีกับกำลังการผลิตของรถตัก และให้ระบบทำงานอย่างต่อเนื่อง ในการจัดการระบบขนส่งระหว่างจุดตักกับจุดเทนั้น ควรมุ่งเน้นไปที่เครื่องจักรที่ต้องใช้ต้นทุนมากที่สุด ในงานวิจัยนี้ผู้วิจัยมุ่งไปที่รถตักซึ่งถูกต้องแล้ว เพราะนอกจากอัตราการใช้เชื้อเพลิงที่สูงแล้วค่าซ่อมบำรุงก็สูงมาก ผลการวิจัย เมื่อเปรียบเทียบระหว่างปี 2005 และ 2006 ต้นทุนลดลง 4.97% คิดเป็นเงิน 1.27 ดอลลาร์/ตัน จากเดิม 1.33 ดอลลาร์/ตัน หรือสามารถประหยัดต้นทุนลงไป 4 ล้านดอลลาร์/ปี

Qing-hua และคณะ (2008) ทีมวิจัยชาวจีน นำระบบ GPS, GPRS และอินเตอร์เน็ตมาใช้ร่วมกัน เพื่อควบคุมการทำงานของรถบรรทุกและรถตักในเหมืองขนาดเล็กและขนาดกลางให้เป็นระบบมากยิ่งขึ้น ได้ผลการวิจัย คือ 1) เทคโนโลยีดังกล่าว สามารถตรวจสอบระบบขนส่งของรถบรรทุกและรถตักปุ้งกี๋เสีย โดยการตั้งค่าหลายทิศทาง ใช้ระบบ Real-time และระบบซอฟแวร์ที่มีคุณภาพสูง ระบบงานในเหมืองดีขึ้น ผลผลิตมากกว่าเดิมและต้นทุนในระบบขนส่งลดลง 2) จากข้อมูล GPS ระบบสามารถหาจำนวนเที่ยวที่รถบรรทุกวิ่งระหว่างจุดตักและจุดเทและจำนวนเที่ยวที่รถตักปุ้งกี๋เสียทำได้ ในระดับความถูกต้อง 99 – 100% ซึ่งทำให้การวางแผนทำเหมืองเป็นระบบขึ้นมาก 3) ระบบก่อให้เกิดการทำงานแบบอัตโนมัติในการจัดการระบบลำเลียงวัตถุดิบและการจัดการข้อมูลในเหมือง ซึ่งสามารถนำไปใช้ประยุกต์กับเหมืองอื่นได้ด้วย

สำหรับการทำเหมืองที่ประเทศแอฟริกาใต้ Krause และ Musingwini (2007) ใช้เครื่องมือ 5 ชนิดเพื่อเปรียบเทียบระบบระหว่างจุดตักและจุดเท คือ Elbrond, FPC, Winston, Arena และ Talpac โปรแกรมที่ได้ผลดีที่สุด คือ Talpac ได้รถบรรทุก CAT777D 7 คัน และ CAT776D 10 คัน และได้สัดส่วนระหว่างจำนวนรถบรรทุกกับรถตักทั้งระบบถึง 6:1 ได้กำลังการผลิตถึง 1,255 ตันต่อชั่วโมง ข้อดีของการใช้โปรแกรมหลาย ๆ โปรแกรมในการเปรียบเทียบก็คือ โปรแกรมบางตัวไม่สามารถตอบโจทย์ผู้วิจัยได้ทั้งหมด เพราะย่อมมีข้อจำกัดในตัวโปรแกรมเองถึงแม้จะเป็นโปรแกรมที่ใช้งานในลักษณะเดียวกันก็ตาม ดังนั้นถ้านำคำตอบของแต่ละโปรแกรมมารวมกันก็จะเป็นคำตอบสุดท้ายของผู้วิจัย ส่วนข้อเสียก็คือ ต้องใช้เวลานาน ในแง่ของการศึกษาโปรแกรมที่ยังไม่คุ้นเคย ดังนั้นผู้วิจัยต้องมีเวลาทุ่มเทเต็มที่

มีการนำการจัดการการลำเลียงหินระหว่างจุดตักและจุเทไปประยุกต์ใช้ในงานอื่น ซึ่งได้ผลลัพธ์เป็นที่น่าพอใจ ทีมนักวิจัยชาวเกาหลีใต้ นำโดย Kang (2006) ใช้ Arena จำลองระบบล่วงหน้าสำหรับการขุดขนหินจากการระเบิดภูเขาเพื่อทำเป็นลานวางตู้สินค้าสำหรับท่าเรือน้ำลึก ปริมาณหินที่ต้องขุดขนออกทั้งหมด 23 ล้านลูกบาศก์เมตร และต้องให้เสร็จภายในระยะเวลา 30 เดือน โดยมีรถบรรทุกทั้งหมด 100 คัน กับหน้างาน 10 จุด ได้ผลลัพธ์ คือ ใช้รถบรรทุกแค่ 83 คัน ดังตารางที่ 1.2

ตารางที่ 1.2 สรุปผลแบบจำลองการสร้างลานวางตู้สินค้า (Kang และคณะ 2006)

จุดตักที่	1	2	3	4	5	6	7	8	9	10	รวม
จำนวนรถบรรทุก	10	9	13	12	6	4	9	8	8	4	83
เวลารอของรถบรรทุก (นาที)	1.4	0.9	2.2	0.5	1.5	1.3	0.7	0.3	0.2	1.6	

งานวิจัยชิ้นสุดท้ายที่ผู้วิจัยได้ศึกษามาเป็นของ Cetin และคณะ (2001) เกี่ยวกับการสร้างแบบจำลองสำหรับเหมืองถ่านหินในตุรกี เป้าหมายของงานวิจัย คือ หาจำนวนนวนรถบรรทุกที่เหมาะสมในการขนถ่านหินจากหน้าเหมืองมายังโรงแต่ง โดยการสร้างโมเดลด้วยโปรแกรม GPSS/H ได้ผลลัพธ์ คือ จำนวนรถบรรทุกที่เหมาะสมที่สุดอยู่ระหว่าง 6 – 8 คัน ที่จำนวนรถบรรทุกเท่านี้จะได้สัดส่วนการทำงานของรถตักปุ้งกี๋เสย เท่ากับ 100.0 % ตามตารางที่ 1.3

ตารางที่ 1.3 ผลการประมวลผลด้วยแบบจำลอง จำนวน 100 เที่ยว (Cetin และคณะ 2001)

จำนวนรถบรรทุก	% การทำงานของรถตัก	เวลารอในแถวคอย (นาที)
1	24.2	0.00
2	47.8	0.03
3	69.8	0.10
4	89.5	0.29
5	99.3	0.90
6	100.0	1.86
7	100.0	2.86
8	100.0	3.86

งานวิจัยหลาย ๆ ชิ้นที่ผู้วิจัยได้ศึกษามาข้างต้นทั้งหมดเป็นการนำโปรแกรมสำเร็จรูปมาใช้งาน ในงานวิจัยนี้ผู้วิจัยต้องการสร้างเครื่องมือขึ้นมาใช้เอง ซึ่งอาจเป็นประโยชน์สำหรับผู้ที่ต้องการพัฒนาเครื่องมือชิ้นนี้ต่อในกรณีที่ยังไม่สมบูรณ์ ดังนั้นผู้วิจัยจึงตัดสินใจใช้โปรแกรมสำเร็จรูปหนึ่งโปรแกรม

และสร้างเครื่องมือขึ้นมาหนึ่งชิ้น คือ Arena 13.0 และโปรแกรมภาษาจาวา ตามลำดับ

1.3 วัตถุประสงค์

เพื่อให้กระบวนการตักหินของรถตักและการย่อยหินของเครื่องย่อยเป็นไปอย่างต่อเนื่อง คือ ลดปัญหาคอขวด (Bottle neck) ที่กระบวนการขนส่งหินโดยรถบรรทุก วัตถุประสงค์ของงานวิจัยนี้จึงแบ่งเป็น 3 ส่วน คือ

1. สร้างแบบจำลองระบบขนส่งหินโดยการเขียนโปรแกรมด้วยภาษาจาวา 1.6
2. สร้างแบบจำลองและประเมินระบบขนส่งหินขั้นต้นด้วยโปรแกรม Arena 13
3. ประเมินระบบขนส่งหินจากจุดตักถึงจุดเท

ทั้งนี้ องค์กรสามารถนำโปรแกรมภาษาจาวาและโมเดลการจำลองด้วยโปรแกรม Arena ไปใช้วางแผนการผลิตเพื่อเพิ่มความเป็นระบบให้กระบวนการลำเลียงหินได้

1.4 ข้อมูลเบื้องต้นสำหรับงานวิจัย

การจำลองระบบขนส่งหินในงานวิจัยนี้ ได้รับความอนุเคราะห์ข้อมูลภาคสนามจากเหมืองหินปูนของบริษัทปูนซิเมนต์ไทย (ทุ่งสง) จำกัด จังหวัดนครศรีธรรมราช และบริษัทปูนซิเมนต์ไทย (ท่าหลวง) จำกัด โรงงานเขาวง จังหวัดสระบุรี

เหมืองหินปูนของบริษัทปูนซิเมนต์ไทย (ทุ่งสง) จำกัด มีเครื่องจักรที่เกี่ยวข้องกับกระบวนการตัก ขนส่งและย่อยหิน คือ

1. รถบรรทุก 2 ประเภท ได้แก่ CAT777C/D น้ำหนักบรรทุก (Payload) 85 ตัน จำนวน 8 คัน และ EUCLID R90C น้ำหนักบรรทุก 90 ตัน จำนวน 3 คัน

2. รถตัก 2 ประเภท ได้แก่ รถตักล้อยาง (Wheel loader) รุ่น CAT992 น้ำหนักตักต่อปุ้งกี๋ 15 ตัน จำนวน 6 คัน และรถตักปุ้งกี๋เสย (Shovel) รุ่น HITACHI EX1800 น้ำหนักตักต่อปุ้งกี๋ 15 ตัน จำนวน 1 คัน และ

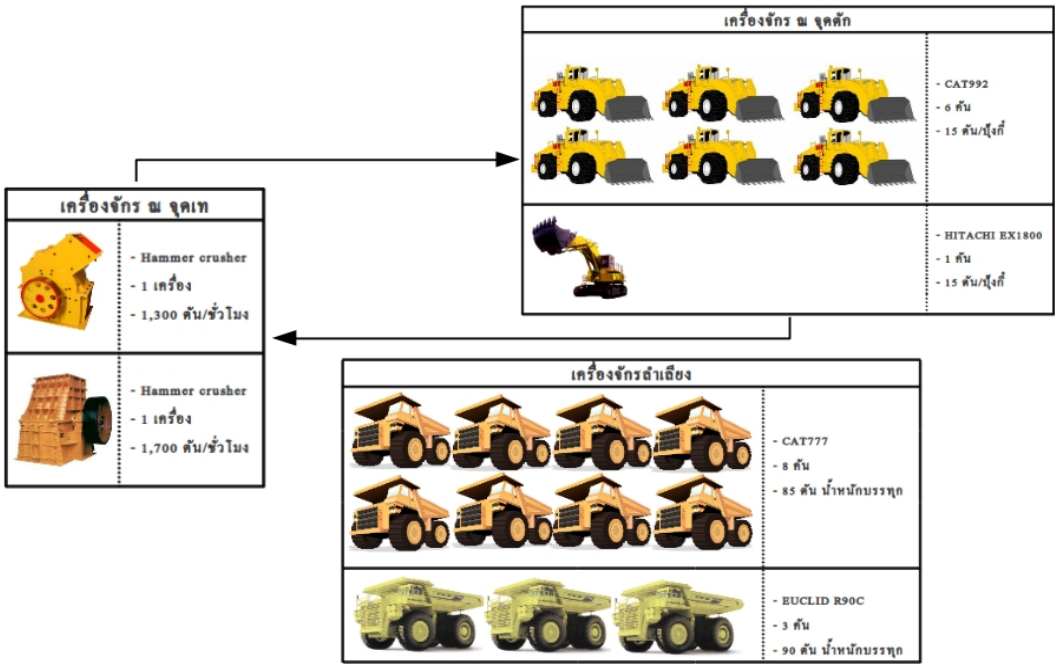
3. เครื่องย่อยหลัก 2 เครื่อง ประเภทค้อนทุบ (Hammer crusher) กำลังผลิต (Capacity) 1,300

และ 1,700 ตันต่อชั่วโมง ตามลำดับ

รายละเอียดตามตารางที่ 1.4 และรูปที่ 1.1

ตารางที่ 1.4 เครื่องจักรที่เกี่ยวข้องกับการสร้างแบบจำลองของบริษัทปูนซิเมนต์ไทย (ทุ่งสง) จำกัด

เครื่องจักร	ประเภท	รุ่น	จำนวน (คัน)	ความจุ/กำลังผลิต
รถบรรทุก	Off-highway truck	CAT777C/D	8	น้ำหนักบรรทุก 85 ตัน
		EUCLID R90C	3	น้ำหนักบรรทุก 90 ตัน
รถตัก	Wheel loader	CAT992	6	น้ำหนักต่อปุ้งกี๋ 15 ตัน
	Shovel	HITACHI EX1800	1	น้ำหนักต่อปุ้งกี๋ 15 ตัน
เครื่องย่อย	Hammer crusher	-	2	1,300 และ 1,700 ตัน/ชั่วโมง



รูปที่ 1.1 เครื่องจักรที่เกี่ยวข้องกับการสร้างแบบจำลองของบริษัทปูนซิเมนต์ไทย (ทุ่งสง) จำกัด

เหมืองหินปูนของบริษัทปูนซิเมนต์ไทย (ท่าหลวง) จำกัด โรงงานเขาวง มีเครื่องจักรที่เกี่ยวข้องกับกระบวนการตัก ขนส่งและย่อยหิน คือ

1. รถบรรทุกประเภท CAT777B/D น้ำหนักบรรทุก 85 ตัน จำนวน 12 คัน

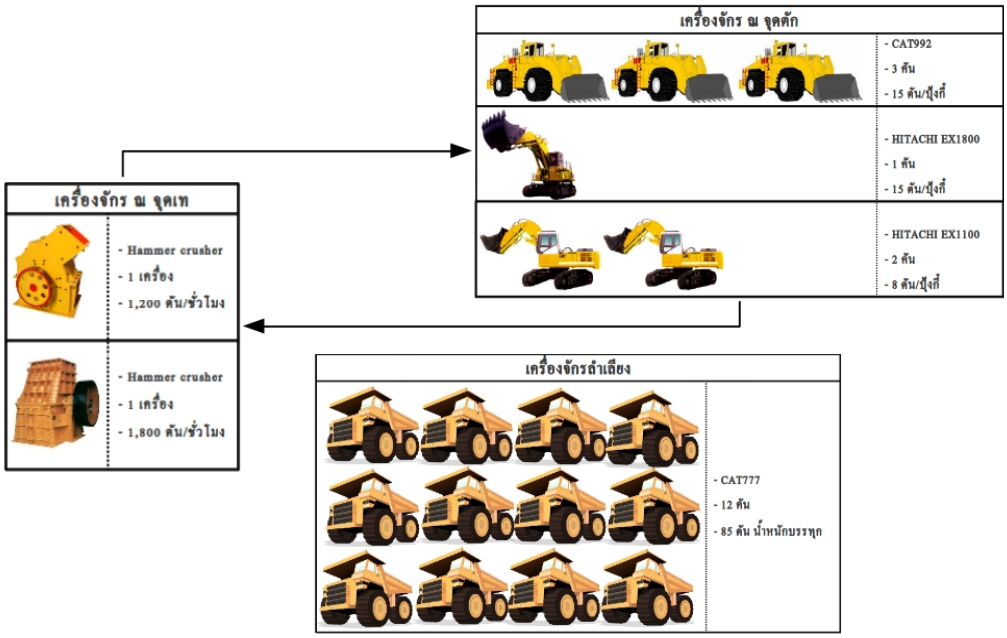
2. รถตัก 3 ประเภท คือ รถตักล้อยาง รุ่น CAT992 น้ำหนักตักต่อปุ้งกี๋ 15 ตัน จำนวน 3 คัน, รถตักปุ้งกี๋เสย รุ่น HITACHI EX1800 น้ำหนักตักต่อปุ้งกี๋ 15 ตัน จำนวน 1 คัน และรถตักปุ้งกี๋เสย รุ่น HITACHI EX1100 น้ำหนักตักต่อปุ้งกี๋ 8 ตัน จำนวน 2 คัน และ

3. เครื่องย่อยหลัก 2 เครื่อง ประเภทค้อนทุบ กำลังผลิต 1,200 และ 1,800 ตันต่อชั่วโมง ตามลำดับ

รายละเอียดตามตารางที่ 1.5 และรูปที่ 1.2

ตารางที่ 1.5 เครื่องจักรที่เกี่ยวข้องกับการสร้างแบบจำลองของบริษัทปูนซิเมนต์ไทย(ท่าหลวง) จำกัด

เครื่องจักร	ประเภท	รุ่น	จำนวน (คัน)	ความจุ/กำลังผลิต
รถบรรทุก	Off-highway	CAT777B/D	12	น้ำหนักบรรทุก 85 ตัน
รถตัก	Wheel loader	CAT992	3	น้ำหนักต่อปุ้งกี๋ 15 ตัน
	Shovel	HITACHI EX1800	1	น้ำหนักต่อปุ้งกี๋ 15 ตัน
		HITACHI EX1100	2	น้ำหนักต่อปุ้งกี๋ 8 ตัน
เครื่องย่อย	Hammer crusher	-	2	1,200 และ 1,800 ตัน/ชั่วโมง



รูปที่ 1.2 เครื่องจักรที่เกี่ยวข้องกับการสร้างแบบจำลองของบริษัทปูนซิเมนต์ไทย(ท่าหลวง) จำกัด

1.5 ทฤษฎีระบบแถวคอย

ทฤษฎีระบบแถวคอยทั่วไป (ธนิศา 2548)

ในระบบงานต่าง ๆ จะพบว่าส่วนใหญ่มีทรัพยากรจำกัด ทำให้มีลักษณะของระบบเป็นแบบแถวคอย (Queuing system) เนื่องจากมีความจำเป็นต้องใช้ทรัพยากรร่วมกัน การบริหารแถวคอยจะเป็นการกำหนดนโยบายว่าใครมีสิทธิได้ใช้ทรัพยากรก่อน หรือหลัง

ระบบแถวคอยเป็นระบบแบบพลวัตร (Dynamic system) คือ ระบบจะมีการเปลี่ยนแปลงสถานะตามเวลา คือเมื่อเวลาเปลี่ยนจะส่งผลให้เกิดการเปลี่ยนแปลงสถานะของระบบ โดยทั่วไปการเปลี่ยนแปลงสถานะของระบบแถวคอยจะเปลี่ยนแปลงแบบไม่ต่อเนื่อง หรือเป็นเหตุการณ์ที่เกิดขึ้นแบบไม่ต่อเนื่อง

โครงสร้างของระบบคิว (Structure of a Queuing System) โดยทั่วไปประกอบด้วย ผู้ใช้บริการ

หรือลูกค้า (Customers) เช่น คน สิ่งของ วัตถุดิบ อุปกรณ์ เครื่องจักร ฯลฯ เข้ามายังระบบบริการ โดยมาจากแหล่งต่าง ๆ ในช่วงเวลาต่าง ๆ กัน ถ้าขณะนั้นผู้ให้บริการ (Servers) ซึ่งอาจเป็นคน สิ่งของ อุปกรณ์ เครื่องจักรหรืออื่น ๆ ว่าง ผู้ใช้บริการจะเข้ารับบริการได้ทันทีโดยไม่ต้องคอย กรณีแบบนี้จะไม่มีแถวคอยเกิดขึ้น แต่ถ้าหากผู้ให้บริการขณะนั้นไม่ว่าง ผู้ใช้บริการจะต้องคอยทำให้เกิดระบบแถวคอย ระบบแถวคอยโดยทั่วไปประกอบด้วย 6 องค์ประกอบหลัก คือ

1.4.1 รูปแบบการเข้ามาของผู้รับบริการ (Arrival pattern of customers)

1.4.1.1 ค่าที่ใช้ในการพิจารณา ในการอธิบายรูปแบบการเข้ามาหรือกระบวนการเข้ามา (Input process) จะอธิบายด้วยค่า 2 ค่า คือ

(1) อัตราเข้ามาโดยเฉลี่ย (Average arrival rate) จำนวนผู้ใช้บริการที่เข้ามาโดยเฉลี่ยต่อหนึ่งหน่วยเวลา

(2) ระยะห่างระหว่างการเข้ามาโดยเฉลี่ย (Average arrival time)

1.4.1.2 ลักษณะการเข้ามา

(1) แน่นอน (Deterministic) การเข้ามาอธิบายด้วยอัตราการเข้าหรือระยะห่างระหว่างการเข้าที่คงที เช่น การเข้ามาห่างกันทุก ๆ 5 นาที

(2) สุ่ม (Random) การเข้ามาอธิบายด้วยอัตราการเข้าหรือระยะห่างระหว่างการเข้าที่เป็นค่าคาดหมายหรือค่าเฉลีย

1.4.1.3 จำนวนผู้เข้ารับบริการ พิจารณาจำนวนผู้เข้ารับบริการ ณ เวลาหนึ่ง ๆ

(1) แบบเดี่ยว (Single)

(2) แบบกลุ่ม (Batch/Bulk) มากกว่าหนึ่งหรือขนาดของกลุ่มอาจมีค่าไม่แน่นอน มีลักษณะเป็นค่าสุ่ม

1.4.1.4 ขนาดของผู้เข้ารับบริการ

(1) จำกัด (Finite)

(2) ไม่จำกัด (Infinite)

1.4.2 กระบวนการให้บริการ (The service process)

1.4.2.1 ค่าที่ใช้ในการพิจารณา ในการอธิบายกระบวนการให้บริการหรือรูปแบบการให้บริการจะอธิบายด้วยค่า 2 ค่า คือ

(1) อัตราการให้บริการโดยเฉลี่ย (Average service rate) จำนวนผู้ได้รับบริการแล้วเสร็จโดยเฉลี่ยต่อหนึ่งหน่วยเวลา

(2) เวลาบริการโดยเฉลี่ย (Average service time) เวลาที่ใช้ในการให้บริการแก่ผู้รับบริการแต่ละคน

1.4.2.2 ลักษณะการบริการ

(1) แน่นอน (Deterministic) ค่าคงที่

(2) สุ่ม (Random) ค่าคาดหมายหรือค่าเฉลี่ย

1.4.2.3 จำนวนผู้เข้ารับบริการ

(1) แบบเดี่ยว (Single) ครั้งละหนึ่งผู้ใช้บริการ

(2) แบบกลุ่ม (Batch/Bulk) ให้บริการมากกว่าหนึ่งผู้ใช้ในเวลาเดียวกัน โดยผู้ให้บริการคนเดียวกัน เช่น ลิฟท์ มัคคุเทศก์ เป็นต้น

1.4.2.4 อัตราความเร็วในการให้บริการ อาจไม่คงที่ขึ้นกับสถานะของระบบ หากในช่วงเวลาใดที่แถวคอยยาว ผู้ให้บริการอาจเร่งการบริการให้เร็วขึ้นทำให้อัตรา การให้บริการสูงขึ้นแต่ถ้าแถวคอยสั้น ผู้ให้บริการอาจมีความเร็วในการบริการต่ำลง ในกรณีที่มีผู้ให้บริการหลายคนที่ต่างให้บริการอย่างเดียวกัน แต่ละคนอาจมีความสามารถหรือความรวดเร็วต่างกัน

1.4.2.5 รูปแบบ การให้บริการสามารถจัดได้หลากหลายรูปแบบ

(1) หน่วยบริการช่องเดียว

(2) หน่วยบริการหลายช่องแบบขนาน มีผู้ให้บริการมากกว่าหนึ่ง แต่มีแถวคอยแถวเดียว

(3) หน่วยบริการหลายช่อง มีผู้ให้บริการมากกว่าหนึ่งและมีแถวคอยหลายแถว

(4) หน่วยบริการแบบอนุกรม การเข้ารับบริการมีหลายขั้นตอนต่อเนื่องกัน

1.4.3 นโยบายการให้บริการ (Service discipline) วิธีการที่ใช้ในการเลือกผู้ใช้บริการที่คอยอยู่ในแถวเพื่อเข้ารับบริการ

1.4.3.1 มาก่อน-ออกก่อน (First In First Out, FIFO)

1.4.3.2 มาหลัง-ออกก่อน (Last In First Out, LIFO)

1.4.3.3 เวลาน้อยที่สุด (Shortest Job First, SJF) เรียงลำดับตามระยะเวลาในการทำงาน งานที่ใช้เวลาทำงานน้อยที่สุดจะออกก่อน

1.4.3.4 อภิสิทธิ์ (Priorities) กำหนดลำดับว่ากิจกรรมใดต้องทำก่อน/หลัง สำหรับกิจกรรมที่มีระดับอภิสิทธิ์เท่ากันให้พิจารณาจากเวลาที่มาถึง

1.4.3.5 Round robin กำหนดระยะเวลาในการทำงานสำหรับแต่ละกิจกรรมเท่ากัน เรียกว่า Time slice เมื่อครบกำหนดเวลาถ้ากิจกรรมยังไม่เสร็จสิ้น จะใส่กลับลงในแถวคอยเพื่อรอการทำงานรอบต่อไป ในการเลือกว่าจะทำกิจกรรมใดก่อน อาจใช้วิธี FIFO หรืออภิสิทธิ์

1.4.3.6 Deadline เรียงลำดับตามกำหนดการที่กิจกรรมต้องเสร็จสิ้น

1.4.4 ขีดความสามารถของระบบในการให้บริการ (System capacity) จำนวนผู้ใช้บริการซึ่งรวมทั้งจำนวนผู้กำลังรับบริการและผู้ที่กำลังคอยที่ระบบสามารถรับได้ ณ เวลาหนึ่ง ๆ ในบางระบบจะไม่มีแถวคอย บางระบบจะมีความยาวแถวคอยจำกัด (Finite queue) และบางระบบจะมีความยาวแถวคอยไม่จำกัด (Infinite queue) หรือมีความยาวมากจนกำหนดได้ว่ามีความยาวไม่จำกัด ในการพิจารณาจะดูจากอัตราการเข้ามาถึงของผู้รับบริการและอัตราการให้บริการซึ่งสามารถแบ่งออกได้เป็น 3 ลักษณะ คือ

1.4.4.1 อัตราการเข้ามาเท่ากับอัตราการให้บริการ เช่น ลูกค้าเข้ามาในแถวคอยทุก ๆ 10 นาที และเวลาในการให้บริการสำหรับลูกค้าแต่ละคนเท่ากับ 10 นาที เช่นกัน ในลักษณะนี้หน่วยบริการจะถูกใช้ประโยชน์ต่อเนื่องกันและไม่มีแถวคอยเกิดขึ้น

1.4.4.2 อัตราการเข้ามามากกว่าอัตราการให้บริการ เช่น ลูกค้าเข้ามาในแถวคอยทุก ๆ 10 นาที (6 คน : ชั่วโมง) แต่อัตราการให้บริการเป็น 12 นาที ต่อ 1 คน (5 คน : ชั่วโมง) ในลักษณะนี้จะมีแถวคอยเกิดขึ้น เนื่องจากมีลูกค้าที่ไม่ได้รับบริการ

1.4.4.3 อัตราการเข้ามาน้อยกว่าอัตราการให้บริการ เช่น ลูกค้าเข้ามารับบริการ 6 คนต่อชั่วโมง แต่สามารถให้บริการได้ 8 คน ต่อชั่วโมง ในลักษณะนี้ หน่วยบริการจะถูกใช้ประโยชน์เพียงแค่ (6/8) * 100 = 75% และไม่มีแถวคอยเกิดขึ้น

1.4.5 จำนวนผู้ให้บริการคู่ขนาน (Number of parallel servers) ในบางระบบอาจมีผู้ให้บริการเพียงหนึ่งเดียว เช่น รูปแบบการให้บริการในแบบหน่วยบริการช่องเดียว หรือ

ในบางระบบอาจมีมากกว่าหนึ่ง เช่น รูปแบบการให้บริการในแบบหน่วยบริการหลายช่องและหน่วยบริการหลายช่องแบบขนาน นอกจากนี้จำนวนผู้ให้บริการอาจไม่คงที่ โดยจะแปรเปลี่ยนไปตามจำนวนผู้รับบริการในระบบหรือตามเวลา

1.4.6 จำนวนขั้นตอนการให้บริการ (Number of service stages) ระบบคิวบางระบบมีขั้นตอนบริการเพียงขั้นตอนเดียว แต่บางระบบอาจประกอบด้วยหลายขั้นตอน เช่น รูปแบบการให้บริการในแบบอนุกรมในระบบแถวคอยที่มีหลายขั้นตอน อาจมีกรณีวงรอบเกิดขึ้น เช่น ในระบบสายการผลิตบางระบบ ถ้าสินค้าชนิดใดไม่ได้มาตรฐานตามเกณฑ์ สินค้าชิ้นนั้นจะถูกส่งกลับเข้ากรรมวิธีการผลิตใหม่ ทำให้เกิดวงรอบ ภายในระบบ

1.4.7 ตัวแบบแถวคอย (Queuing models) ใช้หลักการทางคณิตศาสตร์และสถิติในการอธิบายและวิเคราะห์ผลลัพธ์ของระบบแถวคอย ได้แก่

1.4.7.1 ตัวแบบบริการเดี่ยวมาตรฐาน (Standard single server model) ตัวแบบแถวคอยสำหรับรูปแบบการให้บริการในแบบหน่วยบริการช่องเดียว

1.4.7.2 ตัวแบบบริการเดี่ยวมาตรฐานหลายช่อง (Standard multiserver model) ตัวแบบแถวคอยสำหรับรูปแบบการให้บริการในแบบหน่วยบริการหลายช่องแบบขนาน

1.4.8 การจำลองระบบแถวคอย ในการสร้างโมเดลเชิงหลักการ เพื่อแสดงการจำลองระบบงานที่มีลักษณะแถวคอย จะต้องกำหนดกิจกรรมต่าง ๆ ประกอบด้วย

1.4.8.1 การจัดลำดับการออกจากแถวคอยตามนโยบายการให้บริการ

1.4.8.2 การเพิ่มองค์ประกอบใหม่เข้าสู่แถวคอย

1.4.8.3 เวลาที่องค์ประกอบต้องคอยก่อนเข้ารับบริการ

1.4.8.4 จำนวนขององค์ประกอบในแถวคอย ณ เวลาใด ๆ

ระบบการขนส่งหินหรือแร่ด้วยรถบรรทุกถ้าพิจารณาขั้นตอนการให้บริการจะเป็นแบบวงรอบ กล่าวคือ ในระบบมีผู้ให้บริการคือรถตักและเครื่องย่อย และผู้ใช้บริการคือรถบรรทุก รถบรรทุกวิ่งวนแบบซ้ำ ๆ ตลอดช่วงระยะเวลาที่กำหนด โดยจำนวนของทั้งผู้ให้บริการและผู้ใช้บริการคงที่ การบริหารระบบในลักษณะแบบนี้ต้องพยายามจัดให้ ระบบทำงานในสถานะคงที่ (Steady state) โดยการลดเวลารอของเครื่องจักรทั้ง 3 ประเภท ดังกล่าว

การคำนวณเกี่ยวกับระบบแถวคอย (Krause 2006)

ในที่นี้จะกล่าวถึงการคำนวณระบบแถวคอยในงานเหมืองแร่โดยใช้ทฤษฎีของ Elbrond (1990) ความหมายของตัวแปรที่ใช้ในสมการ มีดังนี้

TS : Service Time คือ เวลาให้บริการของเครื่องจักรให้บริการ เช่น รถตัก

STS : Standard deviation of Service Time คือ ค่าเบี่ยงเบนมาตรฐานของเวลาให้บริการของเครื่องจักรให้บริการ

RT : Return Time คือ เวลาวกกลับของเครื่องจักรรับบริการ เช่น รถบรรทุก

SRT : Standard deviation of Return Time คือ ค่าเบี่ยงเบนมาตรฐานของเวลาวกกลับของเครื่องจักรรับบริการ

N : Number of clients คือ จำนวนของเครื่องจักรรับบริการ

k = TS/RT

pP : Utilization factor of Palm คือ ค่าแฟกเตอร์ของ Palm

pA : Utilization factor of Ashcroft คือ ค่าแฟกเตอร์ของ Ashcroft

WP : Palm's Waiting time คือ เวลารอของ Palm

WA : Ashcroft's Waiting time คือ เวลารอของ Ashcroft

WB : Waiting time in constant case คือ เวลารอกรณีตัวแปรอื่นเป็นค่าคงที่

WS : Adjusted Waiting time คือ เวลารอที่ปรับแก้แล้ว

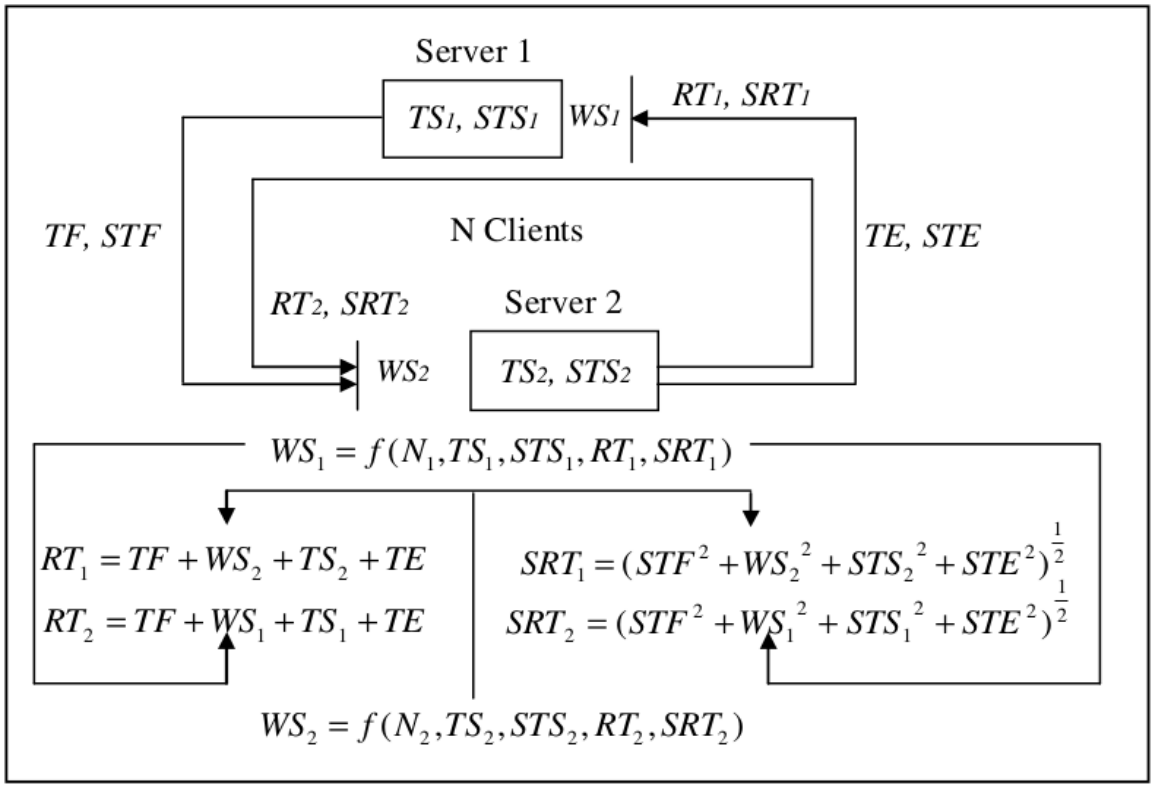
TF : Travel time Full คือ เวลาวิ่งรถหนัก

STF : Standard deviation of Travel time Full คือ ค่าเบี่ยงเบนมาตรฐานของเวลาวิ่งรถหนัก

TE : Travel time Empty คือ เวลาวิ่งรถเปล่า

STE : Standard deviation of Travel time Empty คือ ค่าเบี่ยงเบนมาตรฐานของเวลาวิ่งรถเปล่า

เพื่อให้เห็นภาพ รูปที่ 1.3 แสดงที่มาของสมการที่จะกล่าวในขั้นต่อไป



รูปที่ 1.3 ค่าตัวแปรต่าง ๆ ที่ใช้คำนวณสำหรับโมเดลของ Elbrond

รูปที่ 1.3 เป็นระบบอย่างง่ายที่สุด ประกอบด้วยรถตักปุ้งกี๋เสย 1 คัน เครื่องย่อย 1 เครื่อง และรถบรรทุกไม่จำกัดจำนวน โดย Elbrond อธิบายรูปที่ 1.3 ว่า

" WS_1 ที่รถตัก คำนวณโดยใช้ TS_1 และ RT_1, รถบรรทุกวิ่งรถหนักไปยังเครื่องย่อยด้วย TF, เวลาบริการของเครื่องย่อยเป็น TS_2, รถบรรทุกวิ่งรถเปล่ากลับไปยังรถตักด้วยเวลา TE โดยเป็นเวลารวมของทุกเที่ยว

ที่รถตักปุ้งกี๋เสย รู้ค่า STS_1 และ SRT_1 ได้จากผลบวกทางสถิติของแต่ละเที่ยว, STS_2 และ SRT_2 ที่เครื่องย่อยก็เช่นกัน

WS_2 ที่เครื่องย่อย คำนวณโดยใช้ TS_2 และ RT_2, รถบรรทุกวิ่งรถเปล่าไปยังรถตักด้วย TE, เวลาบริการของรถตักเป็น TS_1, รถบรรทุกวิ่งรถหนักกลับไปยังเครื่องย่อยด้วยเวลา TF โดยเป็นเวลารวมของทุกเที่ยว " โดยมีความสัมพันธ์ ดังนี้

$$WS = f(WP, TS, STS, WA, SRT, WB, RT)$$

เขียนเป็นสมการได้ดังนี้

$$WS = WP\frac{*STS}{TS} + \frac{2*WA*(\frac{1-STS}{TS})*SRT}{RT} + WB*(\frac{1-STS}{TS})*(1-\frac{2*SRT}{RT}) \quad (1.1)$$

$$\rho P = 1-[1+\sum_{n=1}^{N}\prod_{i=1}^{n}\{(N-i+1)*k\}]^{-1} \quad (1.2)$$

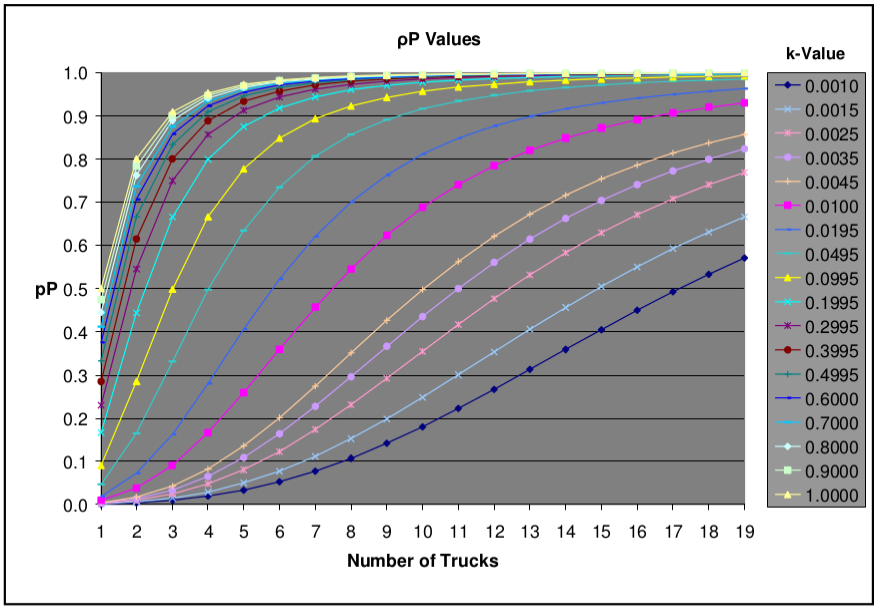
$$\rho A = 1-[1+\{N*k*(1+\sum_{n=1}^{n-1}\prod_{i=1}^{n}\frac{(N-i)*(e^{k*i}-1)}{i}\}^{-1}]^{-1} \quad (1.3)$$

$$WP = \frac{N*TS}{pP} - (TS+RT) \quad (1.4)$$

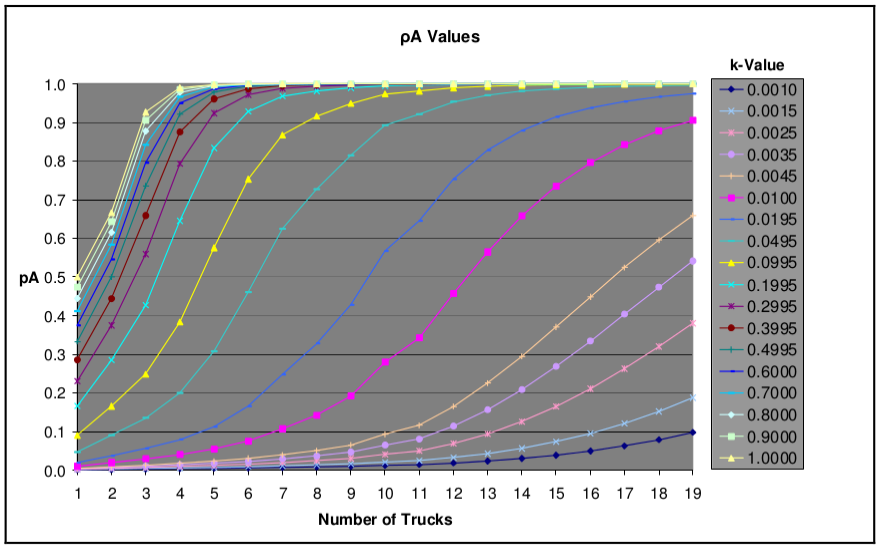
$$WA = \frac{N*TS}{pA} - (TS+RT) \quad (1.5)$$

$$WB = N*TS - (TS+RT) \quad (1.6)$$

สมการหลักคือสมการที่ (1.1) ตัวแปรที่เหลือหาได้จากสมการที่ (1.2) – (1.6) ค่า pP และ pA หาได้จากรูปที่ 1.4 และ 1.5 ตามลำดับ



รูปที่ 1.4 ค่า pP สำหรับสมการ (1.4)



รูปที่ 1.5 ค่า pA สำหรับสมการ (1.5)

ตัวอย่างเช่น ต้องการหาค่า pP ของระบบที่มีรถบรรทุก 15 คัน RT ของรถบรรทุกเท่ากับ 20 นาที TS ของรถตักล้อยางเท่ากับ 3.5 นาที ทำได้ดังนี้

จาก $k = TS/RT$

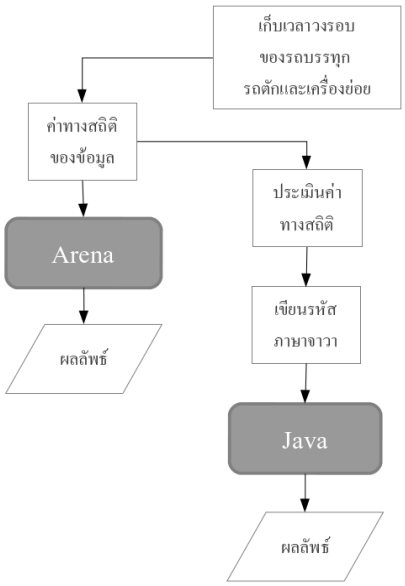
$= 3.5/20$

$= 0.175$

จากรูปที่ 1.4 เลือกค่า k ที่ใกล้เคียงที่สุด คือ 0.1995 และแกนนอนของกราฟที่จำนวนรถบรรทุกเท่ากับ 15 จะได้ค่า pP ประมาณ 1.0

1.6 วิธีการวิจัย

ขั้นตอนการดำเนินงานวิจัยเป็นไปตามรูปที่ 1.6



รูปที่ 1.6 ขั้นตอนการวิจัย

จากรูปที่ 1.6 ต้องมีการเก็บข้อมูลภาคสนามที่จำเป็นสำหรับการสร้างแบบจำลองในรูปแบบของเวลาการทำงานของเครื่องจักรแต่ละประเภท ได้แก่

1. เวลาวงรอบ (Cycle time) ของรถบรรทุก ประกอบด้วย 9 ช่วงเวลา คือ เวลารับหิน (Loading time) เวลาแล่นไปเท (Hauling time) เวลารอ ณ จุดเท (Waiting time at dumping area) เวลาเคลื่อนตัวไปเท (Spot at dump time) เวลารอหน้ายุ้ง (Waiting time at hopper) เวลาเท (Dumping time) เวลาแล่นไปรับหิน (Returning time) เวลารอ ณ จุดตัก (Waiting time at loading area) และเวลาเคลื่อนตัวไปรับหิน (Spot at load time)

2. เวลาวงรอบของรถตัก ประกอบด้วย 4 ช่วงเวลา คือ เวลาเท (Dumping time) เวลาเตรียมตัก (Spot at dig time) เวลาตัก (Digging time) และเวลาเตรียมเท (Spot at dump time)

3. เวลาวงรอบของเครื่องย่อย ประกอบด้วย 2 ช่วงเวลา คือ เวลาทำงาน (Crushing time) และเวลาว่างงาน (Idle time)

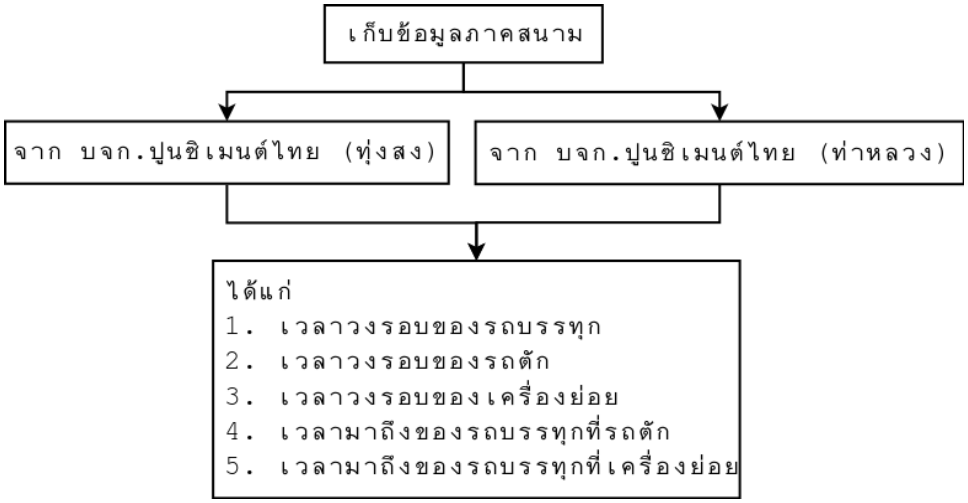
เมื่อเตรียมข้อมูลดังกล่าวครบแล้ว จึงเข้าสู่การสร้างแบบจำลองด้วย Arena โดยเริ่มต้นที่การหาค่าการกระจายของข้อมูล (Data distribution) ซึ่งจำเป็นที่สุดในการสร้างแบบจำลองด้วย Arena จากนั้นจึงนำข้อมูลเข้าเพื่อไปตั้งค่าหน่วย (Mode) ต่าง ๆ ในแบบจำลอง ผลของการจำลองจะได้จำนวนรถบรรทุกที่เหมาะสมที่สุดแต่ละจุดตักกรณีจุดตักหลายจุด ได้เวลารอของเครื่องจักรรับบริการ (Customer machine) คือ รถบรรทุก ที่เครื่องจักรให้บริการ (Server machine) คือ รถตักและเครื่องย่อย และได้สัดส่วนการทำงานของเครื่องจักรบริการ

ส่วนของจาวา ข้อมูลจากภาคสนามจะถูกนำมาประเมินเพื่อใช้เขียนรหัสภาษาจาวา (Java code) โดยแบ่งจำนวนจุดตักเป็น 3 กรณี คือ จุดตัก 1, 2 และ 3 จุด ผลจากการจำลองจะได้จำนวนรถบรรทุกที่เหมาะสมที่สุดในแต่ละจุดตักและผู้ใช้ (User) จะเห็นสถานะของรถบรรทุกแต่ละคันตลอดช่วงเวลา 1 กะ หรือ 8 ชั่วโมง

บทที่ 2
วิธีการวิจัย

2.1 การเก็บข้อมูลภาคสนาม

การสร้างและประเมินแบบจำลอง แบ่งเป็น 2 ขั้นตอนหลัก คือ การเก็บข้อมูลภาคสนามและการสร้างแบบจำลอง การเก็บข้อมูลภาคสนามแบ่งออกเป็นรายละเอียดข้อมูล 2 แหล่ง ได้แก่ บริษัท ปูนซิเมนต์ไทย (ทุ่งสง) จำกัด และบริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด โรงงานเขาวง ข้อมูลที่จำเป็นต้องเก็บจากภาคสนาม ได้แก่ เวลาวงรอบของรถบรรทุก รถตักและเครื่องย่อย เวลามาถึงของรถบรรทุกที่รถตักและเครื่องย่อย ดังแสดงในรูปที่ 2.1 เวลาวงรอบของบรรทุกประกอบด้วย เวลาแล่นไปรับหิน เวลารอที่จุดตัก เวลาเคลื่อนตัวไปรับหิน เวลารับหิน เวลาแล่นไปเท เวลารอที่จุดเท เวลาเคลื่อนตัวไปเท เวลารอที่หน้ายุ้งและเวลาเท ดังตารางที่ 2.1 ส่วนเวลาวงรอบของรถตักและเครื่องย่อยได้จากเวลารับหินของรถบรรทุกหารด้วยจำนวนปุ้งกี๋ที่รถตักตักให้รถบรรทุกต่อเที่ยวและเวลาที่เครื่องย่อยย่อยหินต่อเที่ยวของรถบรรทุก ตามลำดับ



รูปที่ 2.1 การเก็บข้อมูลภาคสนาม

ตารางที่ 2.1 ตารางสำหรับเก็บข้อมูลเวลาวงรอบของรถบรรทุก

เวลา (นาที)	รอบที่						
	1	2	3	4	5	6	ฯลฯ
แล่นไปรับหิน							
รอที่จุดตัก							
เคลื่อนตัวไปรับหิน							
รับหิน							
แล่นไปเท							
รอที่จุดเท							
เคลื่อนตัวไปเท							
รอหน้ายุ้ง							
เท							

เวลามาถึงของรถบรรทุกที่รถตัก บันทึกข้อมูลโดยระบุประเภทและหมายเลขรถบรรทุก บันทึกเวลาปัจจุบันที่รถบรรทุกมาถึงจุดตัก พร้อมทั้งระบุช่วงเวลารอในกรณีรถตักยังไม่พร้อมทำงาน ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 ตารางสำหรับเก็บข้อมูลเวลามาถึงของรถบรรทุกที่รถตัก

เที่ยวที่	หมายเลขรถบรรทุก	ถึงจุดตักเวลา	รอจนถึงเวลา
1			
2			
ฯลฯ			

เวลามาถึงของรถบรรทุกที่เครื่องย่อย บันทึกข้อมูลโดยระบุประเภทและหมายเลขรถบรรทุก บันทึกเวลาปัจจุบันที่รถบรรทุกมาถึงจุดเท พร้อมทั้งระบุช่วงเวลารอในกรณีเครื่องย่อยยังไม่พร้อมทำงาน ตามตารางที่ 2.3

ตารางที่ 2.3 ตารางสำหรับเก็บข้อมูลเวลามาถึงของรถบรรทุกที่เครื่องย่อย

เที่ยวที่	หมายเลขรถบรรทุก	ถึงจุดเทเวลา	รอจนถึงเวลา
1			
2			
3			
4			
5			
ฯลฯ			

ในการจับเวลานั้น ทุกกรณีจะใช้เวลาท้องถิ่นเพื่อเพิ่มความแม่นยำในการคำนวณ ความละเอียดถึงหน่วยวินาที ถ้าต้องการหาช่วงเวลาจะใช้วิธีการนำเวลามาหักลบกัน เช่น

t_1 = 08:05:00 นาฬิกา

t_2 = 08:30:02 นาฬิกา

ดังนั้นช่วงระหว่างเวลาดังกล่าว เท่ากับ

08:30:02 - 08:05:00 = 00:25:02 = 25 นาที 2 วินาที

การเก็บข้อมูลที่ดีควรเป็นไปในลักษณะที่ต่อเนื่อง กล่าวคือ ควรเก็บข้อมูลเวลาการทำงานของเครื่องจักรเพื่อให้เห็นความเป็นไปของทั้งระบบ เช่น ระบบเริ่มเวลา 08:00:00 นาฬิกา และหยุดเวลา 16:00:00 นาฬิกา ช่วงเวลาเท่ากับ 8 ชั่วโมง ในการสร้างแบบจำลองต้องใช้ข้อมูลของเครื่องจักร 3 ชนิด ได้แก่ รถบรรทุก รถตักและเครื่องย่อย ดังนั้นใน 8 ชั่วโมงดังกล่าวต้องได้ข้อมูลครบทั้ง 3 ชนิดเครื่องจักร ทั้งเวลาวงรอบของรถบรรทุก เวลามาถึงของรถบรรทุกที่รถตักและเวลามาถึงของรถบรรทุกที่เครื่องย่อย เพื่อข้อมูลจะได้เป็นชุดเดียวกันและความถูกต้องในการวิเคราะห์ข้อมูลมีมากขึ้น งานวิจัยชิ้นนี้ผู้วิจัยต้องเก็บข้อมูลภาคสนามเพียงลำพัง ซึ่งยากที่จะได้ข้อมูลครบทั้ง 3 ชุด ในเวลาเพียง 8 ชั่วโมง ดังนั้นผู้วิจัยจึงแบ่งการเก็บข้อมูล ออกเป็น

1) เก็บข้อมูลเวลาวงรอบของรถบรรทุกทุกประเภทที่จำเป็นต้องใช้ในการสร้างแบบจำลอง ใช้เวลาประมาณ 7 วัน

2) เก็บข้อมูลเวลามาถึงของรถบรรทุกที่รถตัก ระหว่างเวลา 08:00:00 นาฬิกา - 12:00:00 นาฬิกา และเวลามาถึงของรถบรรทุกที่เครื่องย่อย ระหว่างเวลา 12:00:00 นาฬิกา - 16:00:00 นาฬิกา ใช้เวลาประมาณ 7 วัน

ตัวอย่างการเก็บข้อมูลที่ไม่ดี คือ แยกเก็บข้อมูลของเครื่องจักรแต่ละชนิดต่างหาก เช่น วันที่ 1 ถึง 4 เก็บข้อมูลเวลาวงรอบของรถบรรทุก วันที่ 5 ถึง 9 เก็บข้อมูลเวลามาถึงของรถบรรทุกที่รถตักและวันที่ 10 ถึง 14 เก็บข้อมูลเวลามาถึงของรถบรรทุกที่เครื่องย่อย ถ้าเป็นแบบนี้การจัดการข้อมูลจะเป็นไปด้วยความยากลำบาก จะไม่สามารถจัดกลุ่มข้อมูลได้เลย และจะยิ่งสับสนตอนนำข้อมูลไปใช้กับ Arena 13.0

2.2 วิธีการสร้างแบบจำลองด้วย Arena

โปรแกรม Arena เป็นเครื่องมือตัวหนึ่งที่นิยมใช้งานกันอย่างแพร่หลายสำหรับสร้างตัวแบบจำลองและดำเนินการทดลองไปกับตัวแบบจำลอง โดยตัวแบบจำลองจะถูกทำการทดสอบทางความคิดในคอมพิวเตอร์ เพื่อศึกษาพฤติกรรมของระบบและนำไปสู่แนวทางในการวิเคราะห์ปรับปรุงระบบให้มี

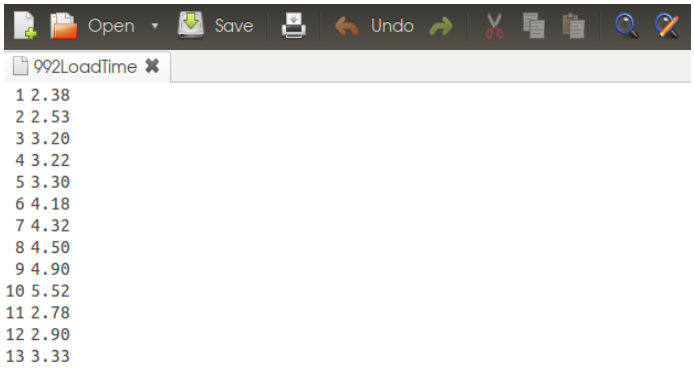
ประสิทธิภาพมากยิ่งขึ้น นอกจากนี้โปรแกรม Arena ยังสามารถสร้างภาพเคลื่อนไหวเสมือนจริงของระบบไว้บนจอคอมพิวเตอร์ได้ด้วย ตัวอย่างเช่น คนงาน เครื่องจักร อุปกรณ์ลำเลียง เป็นต้น โดยแต่ละรูปภาพสามารถแสดงสถานภาพของทรัพยากรได้ด้วย เช่น ว่างงาน ทำงาน หยุดงาน เป็นต้น

โปรแกรมสำเร็จรูป Arena เหมาะสมสำหรับการสร้างแบบจำลองของระบบงานที่มีเหตุการณ์เกิดเป็นช่วง (Discrete-Event Systems) ระบบงานที่มีเหตุการณ์เกิดขึ้นอย่างต่อเนื่อง (Continuous-Event Systems) และระบบงานที่มีทั้งเหตุการณ์เกิดขึ้นเป็นช่วงรวมกับเหตุการณ์ที่เกิดขึ้นอย่างต่อเนื่อง (Combined Continuous/Discrete-Event Systems) แต่โดยส่วนใหญ่แล้ว Arena มักถูกใช้เพื่อสร้างแบบจำลองของระบบงานที่มีเหตุการณ์เกิดขึ้นเป็นช่วง ๆ ด้วยการใช้แผนภาพรูปบล็อค (Block Diagrams) วิธีการของโปรแกรมนี้จะเป็นการกระทำที่มีต่อกันของกระบวนการหรือการไหลตัวของสิ่งที่ถูกให้ความสนใจ (Entity-flow) ที่ถูกสร้างขึ้น แบบจำลองมุ่งเน้นไปที่สิ่งที่ถูกให้ความสนใจหนึ่งอย่าง ณ ที่เวลาหนึ่ง เช่น ลูกค้าจำนวน 1 คน ในธนาคารแห่งหนึ่ง หรือชิ้นส่วนจำนวน 1 ชิ้น ในกระบวนการผลิต แล้วจะมีการติดตามการไหลผ่านตามเส้นทางที่ได้วางไว้ในแบบจำลอง โดยให้ความสนใจกับสิ่งสำคัญ ๆ ทั้งหมดที่เกิดขึ้นตั้งแต่ต้นจนจบ

โปรแกรม Arena พัฒนามาจากภาษา FORTRAN ซึ่งอาจจะทำงานบนเครื่องไมโคร-คอมพิวเตอร์หรือคอมพิวเตอร์ขนาดใหญ่กว่า โดยไม่ตองมีการดัดแปลงแก้ไขเนื้อหาของแบบจำลองที่เขียนขึ้น ถึงแม้ว่า Arena ถูกสนับสนุนให้ใช้งานได้กับเครื่องคอมพิวเตอร์หลายชนิดอย่างกว้างขวาง แต่โครงสร้างและกฎการทำงานของเครื่องคอมพิวเตอร์ก็ยังคงเหมือนกัน แม้ว่า Arena จะมีความสามารถในการจำลองแบบสำหรับระบบงานที่มีความซับซ้อนมาก ๆ ก็จริง แต่ผู้ใช้ก็จะต้องระบุราย-ละเอียดของแบบจำลองให้ชัดเจน โปรแกรม Arena กำหนดให้มีการแบ่งแยกช่วงสำหรับการดำเนินการวิเคราะห์ของการจำลองปัญหาออกเป็น 3 ระยะหลัก คือ การให้คำจำกัดความของแบบจำลอง (Model Definition) การทดลองกับแบบจำลอง (Model Experimentation) และการวิเคราะห์ผลลัพธ์ที่ได้จากการจำลองแบบปัญหา (Output Analysis) ข้อได้เปรียบที่ได้จากการกระทำในลักษณะนี้ก็คือเป็นการทดลองหลาย ๆ ครั้งโดยไม่ต้องมีการดัดแปลงแก้ไขแบบจำลอง และสามารถดำเนินการวิเคราะห์ผลลัพธ์ได้หลาย ๆ ครั้ง โดยใช้แฟ้มข้อมูลของผลลัพธ์ที่ได้จากแบบจำลองตัวเดียวกัน

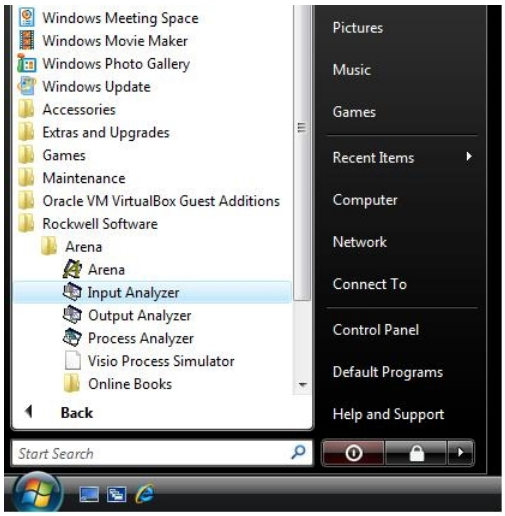
ในการสร้างแบบจำลองใด ๆ ด้วย Arena นั้น ผู้สร้างจำเป็นต้องมีข้อมูลดิบจากภาคสนาม เพื่อนำมาหาค่าการกระจายของข้อมูล โดยค่าการกระจายที่ได้ต้องมีความน่าเชื่อถือของข้อมูลไม่ต่ำกว่า 95

เปอร์เซ็นต์ หรือ ค่า p-value ไม่เกิน 0.05 หมายความว่า ถ้า p-value เท่ากับ 0.04 แสดงว่ามีความน่าเชื่อถือของข้อมูลเป็น 0.96 หรือ 96 เปอร์เซ็นต์ สามารถนำข้อมูลชุดนี้ไปใช้งานได้ แต่ถ้า p-value เท่ากับ 0.06 แสดงว่ามีความน่าเชื่อถือของข้อมูลเป็น 0.94 หรือ 94 เปอร์เซ็นต์ ไม่สามารถนำข้อมูลชุดนี้ไปใช้ได้ ต้องเก็บข้อมูลเพิ่มเติมเพื่อลดค่า p-value หรือหาข้อผิดพลาดจากการเก็บข้อมูล



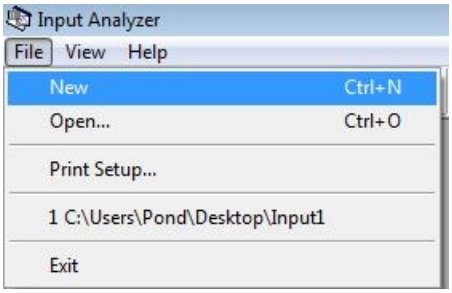
รูปที่ 2.2 เวลาตักใส่รถบรรทุกจนเต็มของรถตักล้อยาง

เมื่อมีข้อมูลดิบจากภาคสนามที่พร้อมแล้วสำหรับหาค่าการกระจาย โดยที่ข้อมูลต้องบันทึกอยู่ในไฟล์ข้อความนามสกุล .txt ดังตัวอย่างในรูปที่ 2.2 เป็นเวลาตักหินจนเต็มรถบรรทุกโดยรถตักล้อยาง CAT992 จำนวน 31 ข้อมูล ในที่นี้แสดงตัวอย่างให้เห็น 13 ข้อมูล



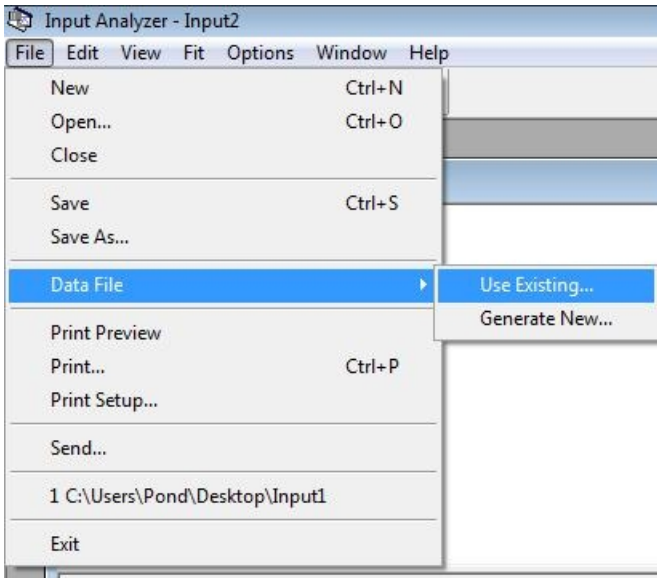
รูปที่ 2.3 เปิดโปรแกรม Input Analyzer เพื่อวิเคราะห์การกระจายของข้อมูล

เปิด Input Analyzer โดยไปที่ All Programes\Rockwell Software\Arena\Input Analyzer ตามรูปที่ 2.3

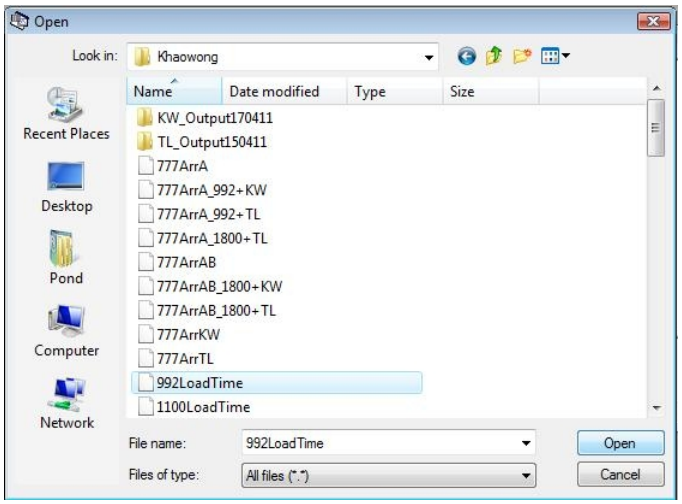


รูปที่ 2.4 สร้างไฟล์ใหม่โดยเลือก New

ใน Input Analyzer ให้สร้างไฟล์ใหม่โดยไปที่ File\New ตามรูปที่ 2.4

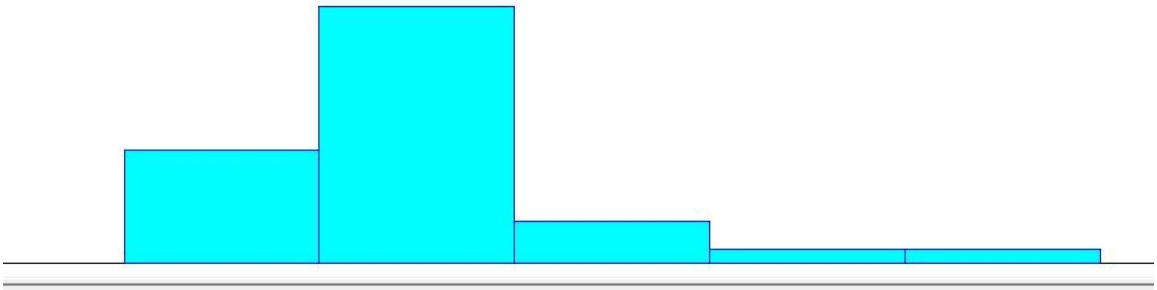


รูปที่ 2.5 เรียกข้อมูลเข้ามาใน Input Analyzer



รูปที่ 2.6 เปิดไฟล์ที่เคยบันทึกไว้เป็นนามสกุล .txt

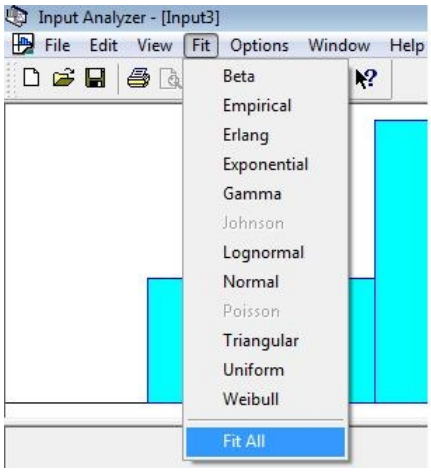
จากนั้นให้เรียกข้อมูลที่บันทึกเป็นไฟล์ข้อความเข้ามา โดยเลือก File\Data File\Use Existing... ดังในรูปที่ 2.5



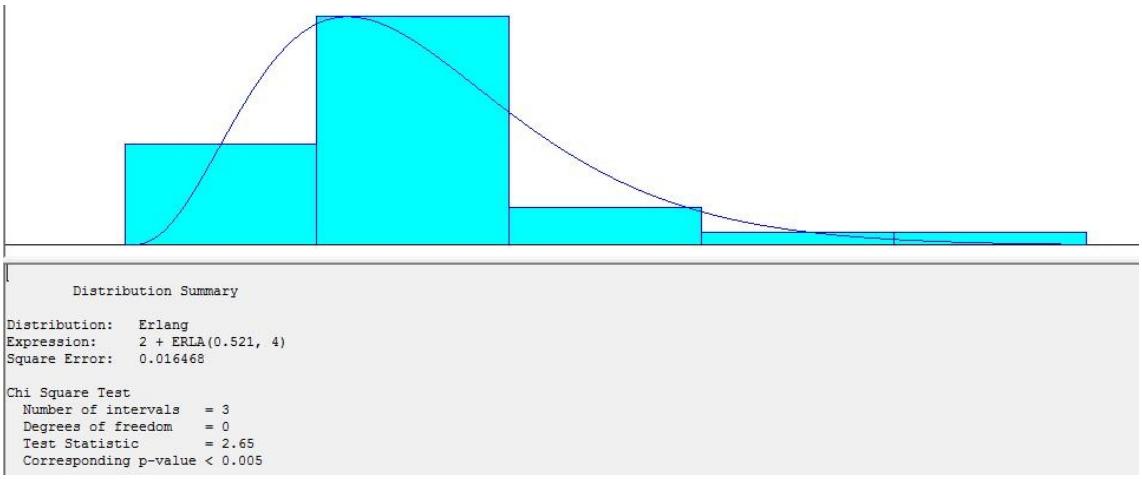
```
       Data Summary

Number of Data Points   = 31
Min Data Value          = 2.38
Max Data Value          = 8.17
Sample Mean             = 4.08
Sample Std Dev          = 1.14

       Histogram Summary

Histogram Range         = 2 to 8.75
Number of Intervals     = 5
```

รูปที่ 2.7 ผลลัพธ์จากการประมวลผลด้วย Input Analyzer

จากรูปที่ 2.6 ให้เปิดไฟล์ที่ต้องการหาค่าการกระจายขึ้นมา ในที่นี้เป็นไฟล์ที่ชื่อว่า 992LoadTime.txt และเมื่อได้ผลลัพธ์ดังรูปที่ 2.7 แล้ว ให้ทำการหาค่าการกระจายของข้อมูลโดยไปที่ Fit\Fit All ตามรูปที่ 2.8



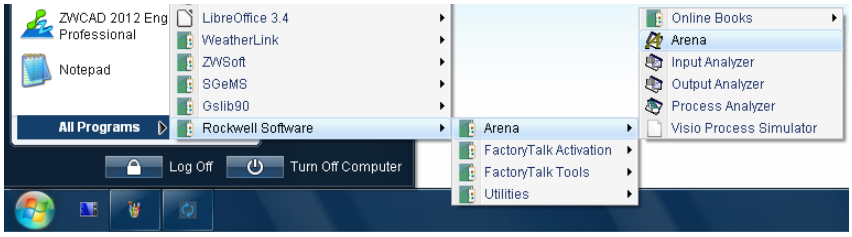
รูปที่ 2.8 หาค่าการกระจายของข้อมูลโดยเลือก Fit All



รูปที่ 2.9 ค่าการกระจายของข้อมูลที่ได้จากการประมวลผลโดย Input Analyzer

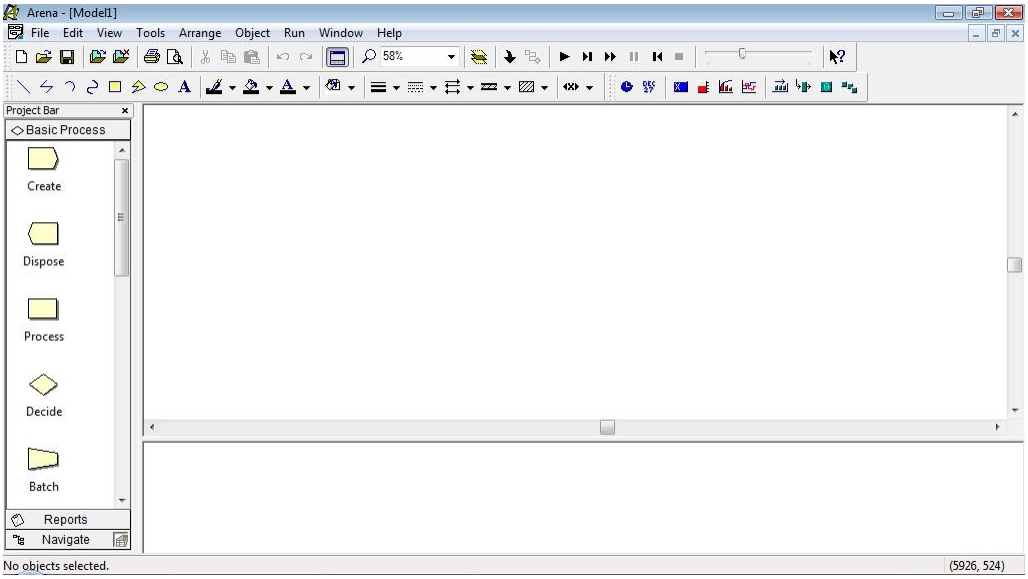
รูปที่ 2.9 เป็นผลลัพธ์สุดท้ายของการประมวลผลซึ่งได้ค่าการกระจายเป็นประเภท Erlang มีค่าเป็น 2 + ERLA(0.521, 4) และค่า p-value ไม่เกิน 0.05 ในที่นี้น้อยกว่า 0.005 ซึ่งค่าที่ต้องนำไปใช้ในขั้นตอนถัดไปคือ 2 + ERLA(0.521, 4)

เสร็จจากขั้นตอนนี้แล้วจะเป็นการสร้างโมเดล โดยมีลำดับการสร้างดังนี้



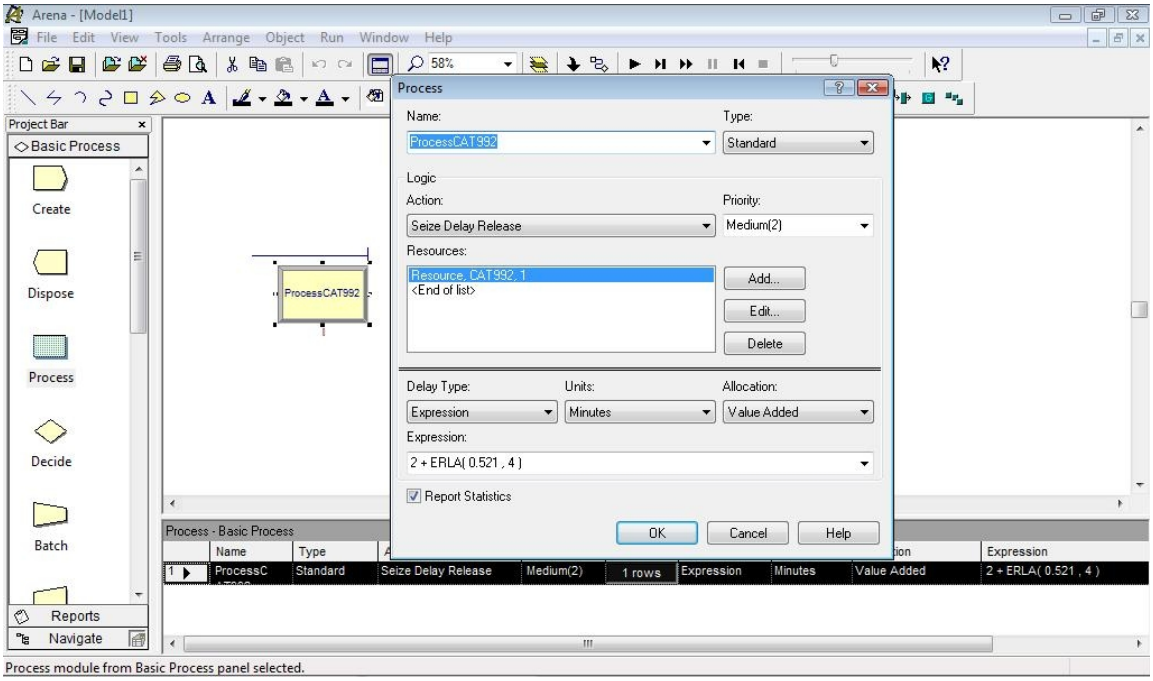
รูปที่ 2.10 เปิดโปรแกรม Arena เพื่อสร้าง Model

เปิด Arena โดยไปที่ All Programes\Rockwell Software\Arena\Arena ตามรูปที่ 2.10



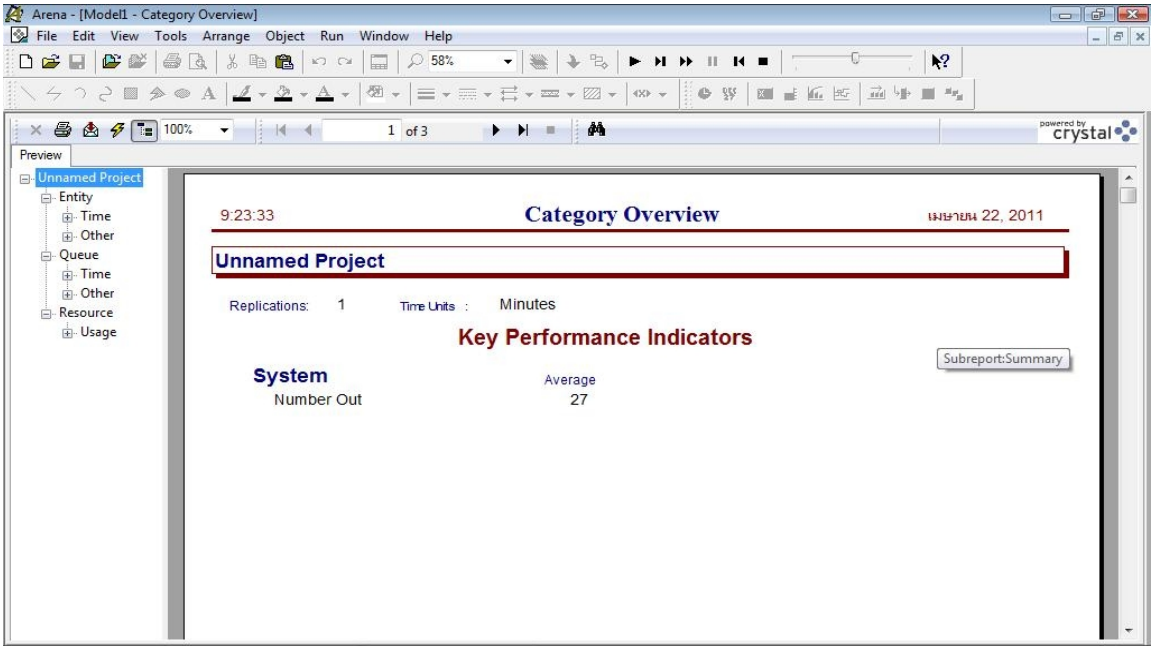
รูปที่ 2.11 โปรแกรม Arena

รูปที่ 2.11 เป็นหน้าแรกของโปรแกรม Arena ซึ่งพร้อมสำหรับสร้างแบบจำลองหรือโมเดล



รูปที่ 2.12 การสร้างแบบจำลองด้วยโปรแกรม Arena

รูปที่ 2.12 เป็นตัวอย่างการสร้างแบบจำลอง โดยสร้างหน่วยการทำงาน (Process module) ชื่อ ProcessCAT992 ข้อมูลในช่อง Expression: เป็นข้อมูลที่ได้มาจาก Input Analyzer



รูปที่ 2.13 ผลลัพธ์การทำงานของแบบจำลอง

ผลลัพธ์ของ Arena จะรายงานวัตถุที่ไหลไปในระบบ แถวคอยและเครื่องจักรให้บริการ ในเทอมของจำนวน เวลาและสัดส่วนการทำงาน ดังแสดงในรูปที่ 2.13

จากบทที่ 1 ผู้วิจัยได้กล่าวถึงการประเมินระบบขนส่งหินด้วยรถบรรทุก ณ เหมืองหิน 2 เหมือง ได้แก่ เหมืองหินปูน บริษัทปูนซิเมนต์ไทย (ทุ่งสง) จำกัด และเหมืองหินปูน บริษัทปูนซิเมนต์ไทย (ท่าหลวง) จำกัด ลักษณะการขนส่งหินของเหมืองหินปูนทุ่งสงเป็นแบบตรึงจุดตักและสุ่มจุดเทในกรณีที่ย่อย 2 เครื่องย่อย ส่วนเหมืองหินปูนท่าหลวงเป็นการทำงานแบบตรึงเครื่องย่อยและสุ่มจุดตัก กรณีที่พนักงานขับรถบรรทุกมองเห็นจุดตักมากกว่า 1 จุด ลำดับต่อไปจะเป็นรายละเอียดของการสร้างแบบจำลองด้วยภาษาจาวา สำหรับเหมืองหินปูนทั้ง 2 เหมือง

2.3 วิธีการสร้างแบบจำลองด้วยภาษาจาวา

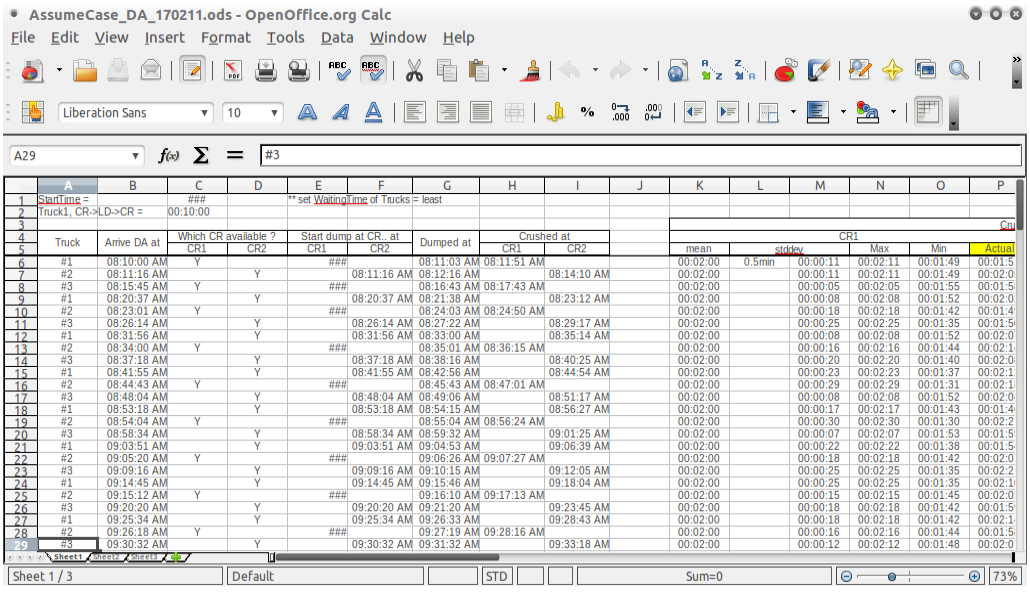
ในการสร้างแบบจำลองเพื่อประเมินเหตุการณ์ใดเหตุการณ์หนึ่ง ด้วยภาษาคอมพิวเตอร์ใด ๆ

นั้น ควรดำเนินการตามขั้นตอนดังนี้

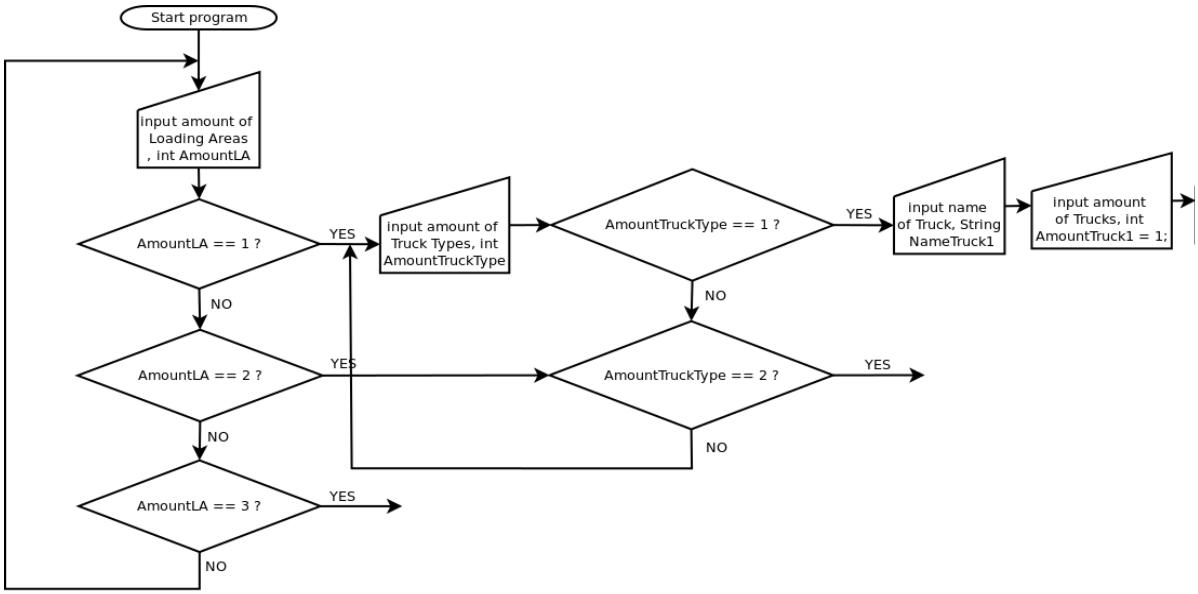
- จำลองเหตุการณ์ด้วยมือ เป็นขั้นตอนแรกในการสร้างแบบจำลองของระบบ ถือเป็นการลองผิดลองถูกเพราะระหว่างนี้การจำลองยังไม่เป็นขั้นตอนเท่าที่ควร แต่ทั้งนี้ผู้สร้างแบบจำลองต้องมีประสบการณ์หรือเคยสัมผัสกับระบบงานจริงมาก่อนเพื่อง่ายในการจินตนาการถึงระบบ

- จำลองเหตุการณ์ด้วยกระดาษคำนวณ (Spreadsheet) เป็นขั้นตอนการตรวจทานและเพื่อยืนยันว่าการจำลองด้วยมือนั้นถูกต้อง นอกจากนี้ตัวแปรต่าง ๆ ผู้สร้างแบบจำลองสามารถค้นพบได้จากขั้นตอนนี้ เครื่องมือที่ใช้ ได้แก่ Microsoft Excel สำหรับระบบปฏิบัติการ Windows และ OpenOffice Calc หรือ LibreOffice Calc สำหรับระบบปฏิบัติการ Linux เป็นต้น ตัวอย่างดังแสดงในรูปที่ 2.14

- เขียนแผนผังลำดับการทำงาน เป็นการสร้างแผนผังการไหลหรือแผนผังทางความคิดของระบบ เมื่อมาถึงขั้นตอนนี้ผู้สร้างแบบจำลองจะสามารถมองเห็นความเป็นไปได้ในการจำลองระบบได้เกือบสมบูรณ์ ตัวอย่างดังแสดงในรูปที่ 2.15



รูปที่ 2.14 การจำลองระบบด้วย OpenOffice Calc ในระบบปฏิบัติการ Linux



รูปที่ 2.15 แผนผังการไหลของระบบ เขียนด้วย Dia 0.97.1

- เขียนรหัสภาษาคอมพิวเตอร์ เป็นขั้นตอนสำคัญที่สุดในการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ทุกภาษา เป็นขั้นตอนที่มีความยุ่งยากที่สุด ถ้าติดขัดต้องย้อนกลับไปดูหรืออาจจะต้องแก้ไขขั้นตอนก่อนหน้า โดยที่ผู้สร้างแบบจำลองต้องมีความรู้เรื่องภาษาที่ใช้เขียนเป็นอย่างดี เมื่อเสร็จขั้นตอนนี้จะได้โปรแกรมหนึ่งโปรแกรมสำหรับจำลองระบบที่ต้องการ

จากขั้นตอนทั้ง 4 สามารถสรุปเป็นแผนผังได้ดังรูปที่ 2.16

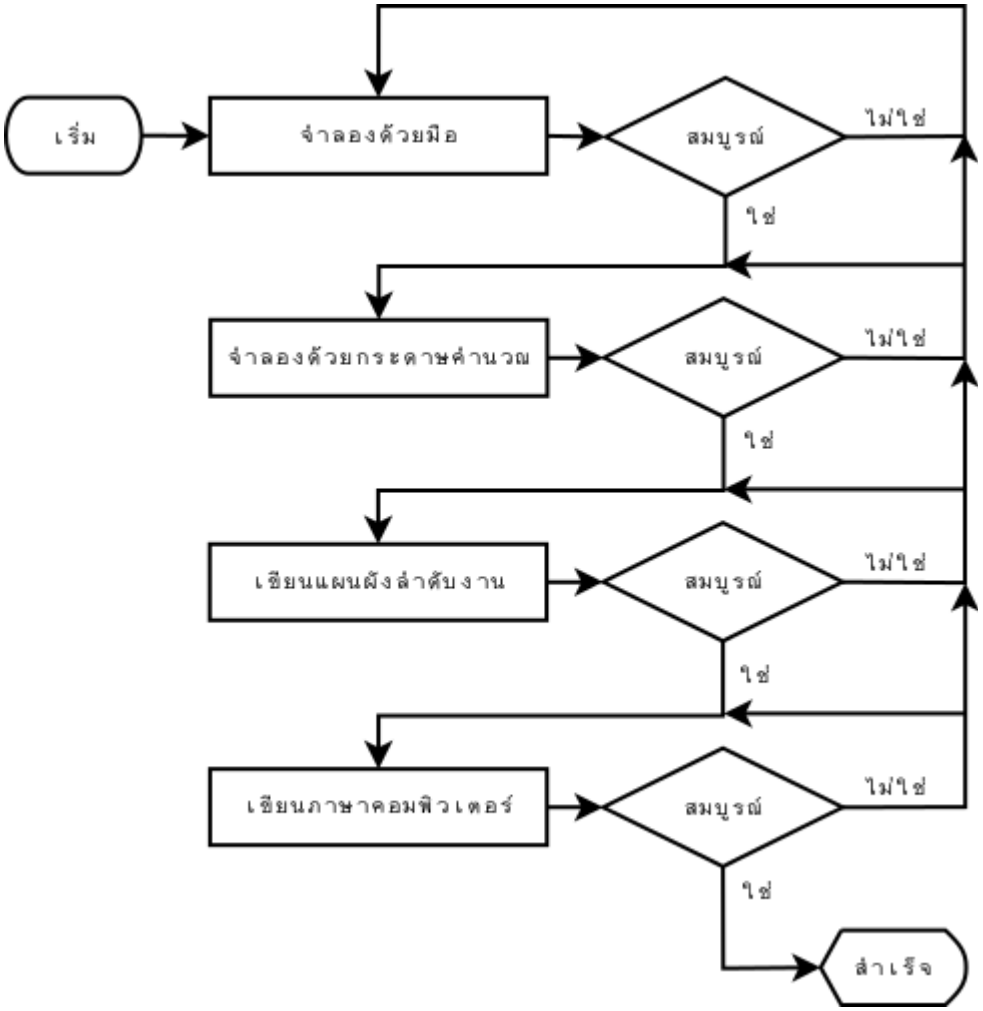
งานวิจัยชิ้นนี้ผู้วิจัยเลือกภาษาจาวาในการเขียนโปรแกรม เหตุผลที่เลือกใช้ภาษาจาวาเพราะมีข้อดีหลัก ๆ อยู่หลายประการ ดังนี้

1. การพัฒนาภาษาจาวาเป็นสิ่งที่ไม่ต้องเสียค่าใช้จ่ายไม่ว่าจะเป็นเครื่องมือ เอดิเตอร์หรือฐานข้อมูล เพียงมีความรู้ในภาษาจาวาก็สามารถเขียนและพัฒนาโปรแกรมจาวาได้

2. โปรแกรมจาวาที่เขียนขึ้นสามารถทำงานได้หลายแพลตฟอร์ม โดยไม่จำเป็นต้องแก้ไขหรือคอมไพล์ใหม่ ทำให้ลดค่าใช้จ่ายและเวลาที่ต้องเสียไปในการเชื่อมต่อ

3. จาวาเป็นภาษาเชิงวัตถุ (Object Oriented Programming, OOP) ซึ่งเหมาะ สำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุ จะช่วยให้สามารถนำคำหรือชื่อต่าง ๆ ที่มีอยู่ใน

ระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่ายขึ้น



รูปที่ 2.16 สรุปขั้นตอนการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์

4. มีความซับซ้อนน้อยกว่าภาษา C++ ทำให้ใช้งานได้ง่ายกว่าและลดความผิดพลาดได้มากขึ้น

5. มีการตรวจสอบข้อผิดพลาดทั้งตอนคอมไพล์ (compile time) และขั้นตอนการรัน (run time) ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรมและช่วยให้ดีบัก (debug) โปรแกรมได้ง่าย

6. ภาษาจาวาถูกออกแบบมาให้มีความปลอดภัยสูงตั้งแต่แรก ทำให้โปรแกรมที่เขียนขึ้นด้วย

จาวามีความปลอดภัยมากกว่าโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น

7. มี IDE (Integrated Development Environment) ช่วยอำนวยความสะดวกและมี Application servers and libraries ต่าง ๆ มากมายที่สามารถใช้งานได้โดยไม่ตองเสียค่าใช้จ่าย ทำให้ลดค่าใช้จ่ายที่ต้องเสียไปกับการซื้อเครื่องมือ (tools) และซอฟต์แวร์ต่าง ๆ

รหัสภาษาจาวาที่เขียนขึ้นมีขอบเขตการทำงาน ดังนี้

1. ใช้กับเหมืองหินที่มีจุดตักไม่เกิน 3 จุด

2. เครื่องย่อยทำงานพร้อมกันไม่เกิน 2 เครื่อง และอยู่ในพื้นที่เดียวกัน

3. เมื่อระบบเริ่มทำงานรถบรรทุกออกตัวจากจุดเทไปยังจุดตัก

4. รถบรรทุกมี 1 ประเภท คุณสมบัติของรถทุกคันเหมือนกันทุกประการ

5. มีรถตักทำงาน 1 คันต่อจุดตัก

การทำงานจะแบ่งเป็น 3 กรณี คือ ผู้ใช้เลือกจุดตัก 1, 2 หรือ 3 จุด รหัสภาษาจาวาดังกล่าวเป็นไปตามรูปที่ 2.17

```
1  import java.util.Scanner;
2  import static java.lang.System.out;
3
4  public class Assembly130611 {
5      public static void main(String[] pond) {
6          GetProcedures();
7      }
8      public static void GetProcedures() {
9          Scanner T = new Scanner(System.in);
10         out.println("\n\n\t\t ------------------ QUARRIES HAULAGE SYSTEM SIMULATION ON UBUNTU ------------
11                 "\n\n\n\t-Loading Areas-");
12         out.print("\t\tHow many loading areas ? : ");
13         int noOfLA = T.nextInt();
14         if (noOfLA == 1) {
15             OneLA130611.GetProcedures();
16         } else if (noOfLA == 2) {
17             TwoLA130611.GetProcedures();
18         } else if (noOfLA == 3) {
19             ThreeLA130611.GetProcedures();
20         } else {
21             out.println("\n\t\tWarning ! Excessive number of loading areas. Please, insert again.");
22         }
23     }
24 }
```

รูปที่ 2.17 รหัสภาษาจาวา ผู้ใช้สามารถเลือกจุดตักได้ไม่เกิน 3 จุด

จากรูปที่ 2.17 บรรทัดที่ 12 โปรแกรมจะถามผู้ใช้ว่า จุดตักที่จะทำงานมีกี่จุด ถ้าผู้ใช้เลือก

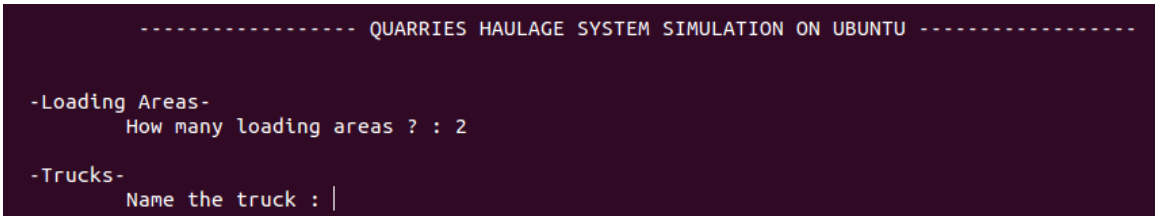
- "1" โปรแกรมจะทำบรรทัดที่ 15 แล้วเข้าสู่เงื่อนไขจุดตัก 1 จุด

- "2" โปรแกรมจะทำบรรทัดที่ 17 แล้วเข้าสู่เงื่อนไขจุดตัก 2 จุด

- "3" โปรแกรมจะทำบรรทัดที่ 19 แล้วเข้าสู่เงื่อนไขจุดตัก 3 จุด

- ตัวเลขอื่นนอกเหนือจาก 1-3 โปรแกรมจะทำบรรทัดที่ 21 และผู้ใช้ต้องป้อนค่าตัวเลขให้ถูกต้อง

ผลจากการประมวลผลรหัสดังกล่าวเป็นไปดังรูปที่ 2.18



รูปที่ 2.18 ผลการประมวลผลรหัสภาษาจาวาในรูป 2.17

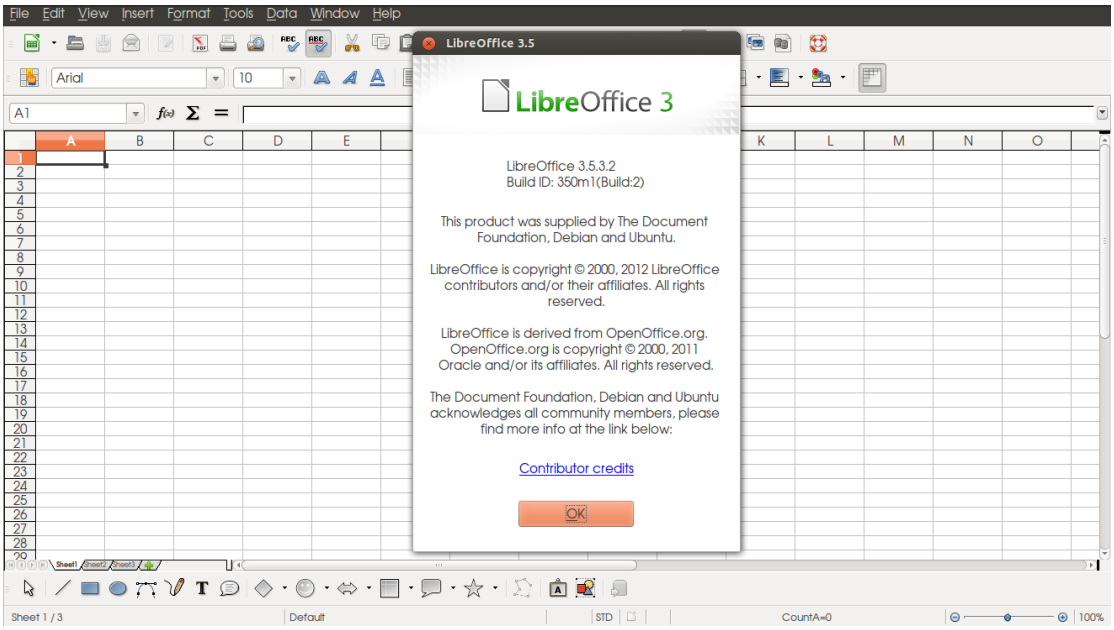
รูปที่ 2.18 ผู้ใช้เลือกจุดตักเท่ากับ 2 จุด ถัดไปผู้ใช้จะต้องระบุจำนวนรถบรรทุกที่พร้อมใช้งานสำหรับกะนั้น ๆ และระบุรายละเอียดจำเป็นอื่น ๆ โปรแกรมก็จะประมวลผลระบบขนส่งหินโดยรถบรรทุกออกมาให้

ที่สำคัญ เมื่อเสร็จขั้นตอนสุดท้ายคือ การเขียนภาษาคอมพิวเตอร์ตามรูปที่ 2.16 แล้ว ต้องมีการตรวจสอบความผิดพลาดเพื่อความสมบูรณ์ของโปรแกรม จึงเป็นอันเสร็จสิ้นการสร้างแบบจำลองด้วยการเขียนภาษาคอมพิวเตอร์

2.4 วัสดุและอุปกรณ์

การสร้างแบบจำลองด้วย Arena ประกอบด้วยเครื่องมือ ดังนี้

1. OpenOffice Calc หรือ LibreOffice Calc 3 เป็นโปรแกรมกระดาษคำนวณ สำหรับบันทึกข้อมูลดิบจากภาคสนามและคำณวนค่าเบื้องต้น ดังรูปที่ 2.19



รูปที่ 2.19 โปรแกรมกระดาษคำนวณ LibreOffice Calc 3

รูปที่ 2.19 เป็นโปรแกรมกระดาษคำนวณที่ไม่ต้องเสียค่าลิขสิทธิ์ (Opensource program) ซึ่งเทียบเท่ากับโปรแกรมลิขสิทธิ์ คือ Microsoft Excel ในระบบปฏิบัติการ Windows

2. gedit 3.4.1 เป็นโปรแกรมเขียนข้อความหรือ Text editor สำหรับเตรียมข้อมูลนำเข้า Input Analyzer ตามรูปที่ 2.20 ซึ่งเทียบเท่ากับโปรแกรม Notepad ในระบบปฏิบัติการ Windows

รูปที่ 2.20 โปรแกรมเขียนข้อความ gedit รุ่น 3.4.1

3. Arena 13.0 โปรแกรมหลักสำหรับสร้างแบบจำลอง ดังแสดงในรูปที่ 2.10-2.13

และเครื่องมือทั้งหมดที่ใช้ในการเขียนโปรแกรมภาษาจาวาสำหรับจำลองระบบ ได้แก่

1. Java รุ่น 1.6 ดังแสดงในรูปที่ 2.21

```
pond@POND:~/Eclipse/Ubulation/src$ java -version
java version "1.6.0_22"
OpenJDK Runtime Environment (IcedTea6 1.10.2) (6b22-1.10.2-0ubuntu1~11.04.1)
OpenJDK Server VM (build 20.0-b11, mixed mode)
pond@POND:~/Eclipse/Ubulation/src$
```

รูปที่ 2.21 ภาษาจาวารุ่น 1.6

จากรูปที่ 2.21 เป็นการทำงานในโปรแกรมคอมมานด์ไลน์ (Command line) ที่ชื่อว่า “Terminal” ในระบบปฏิบัติการลินุกซ์ (Linux) จากรูป

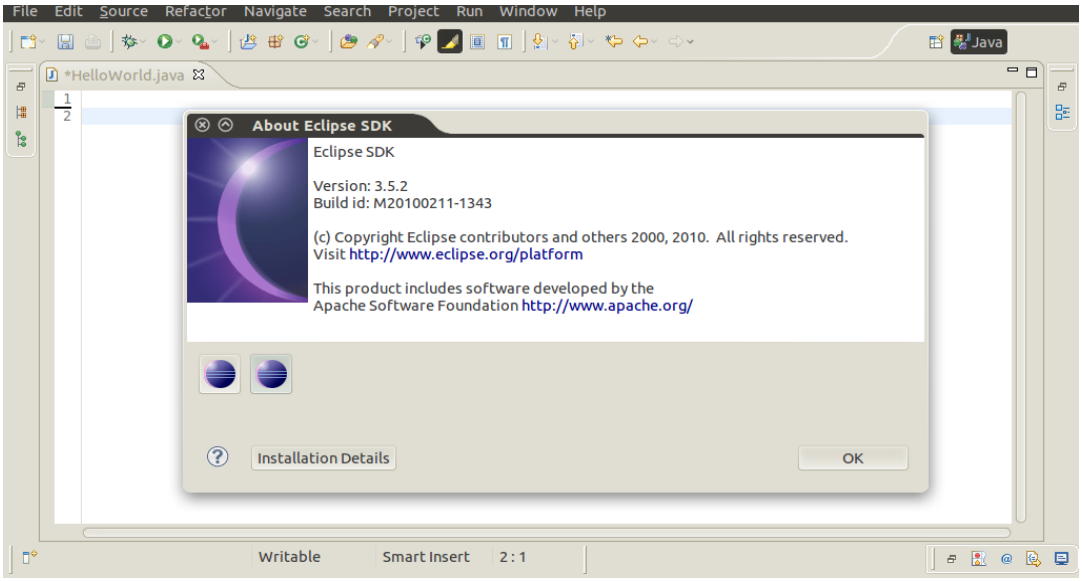
- บรรทัดที่ 1 “java -version” เป็นการสั่งให้ Terminal ตรวจสอบรุ่นของภาษาจาวาที่อยู่ในเครื่องคอมพิวเตอร์

- บรรทัดที่ 2 “java version “1.6.0_22”” คือ รุ่นของภาษาจาวาที่อยู่ในเครื่องคอมพิวเตอร์

- บรรทัดที่ 3 “OpenJDK Runtime Environment (IcedTea6 1.10.2) (6b22-1.10.2-0ubuntu1~11.04.1) ” คือ รุ่นของ JRE ที่มาพร้อมกับภาษาจาวา

- และบรรทัดที่ 4 “OpenJDK Server VM (build 20.0-b11, mixed mode) ” คือ รุ่นของ JVM ที่มาพร้อมกับภาษาจาวา

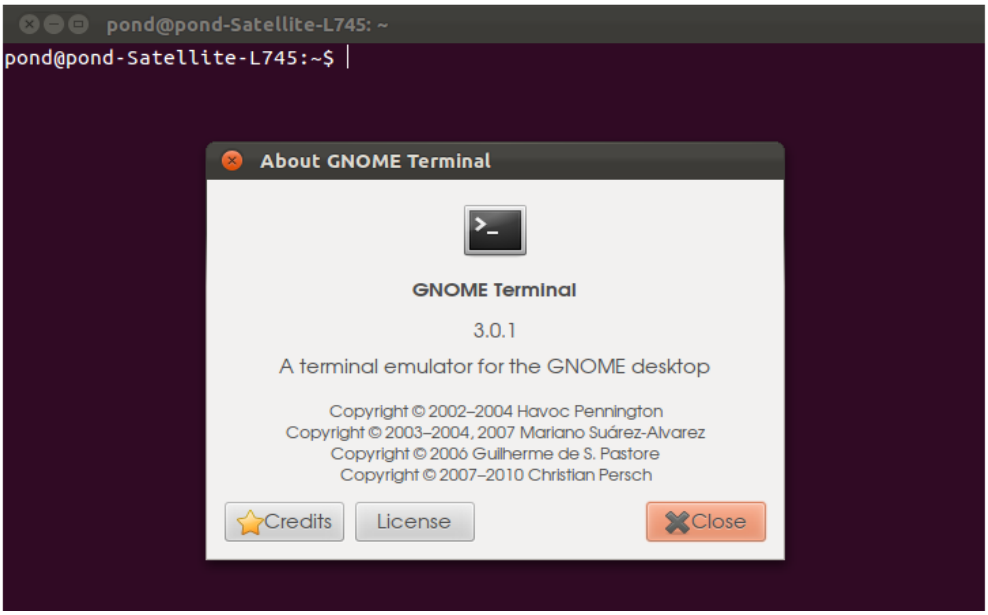
2. Eclipse SDK รุ่น 3.5 เป็น IDE ดังแสดงในรูปที่ 2.22



รูปที่ 2.22 IDE Eclipse SDK รุ่น 3.5.2

3. GNOME Terminal รุ่น 2.32 เป็น Command line ในระบบปฏิบัติการลีนุกซ์ ซึ่งเทียบเท่ากับ

DOS ในระบบปฏิบัติการวินโดวส์ (Windows) ดังแสดงในรูปที่ 2.23



รูปที่ 2.23 Command line GNOME Terminal รุ่น 2.32

บทที่ 3
ผลการวิจัย

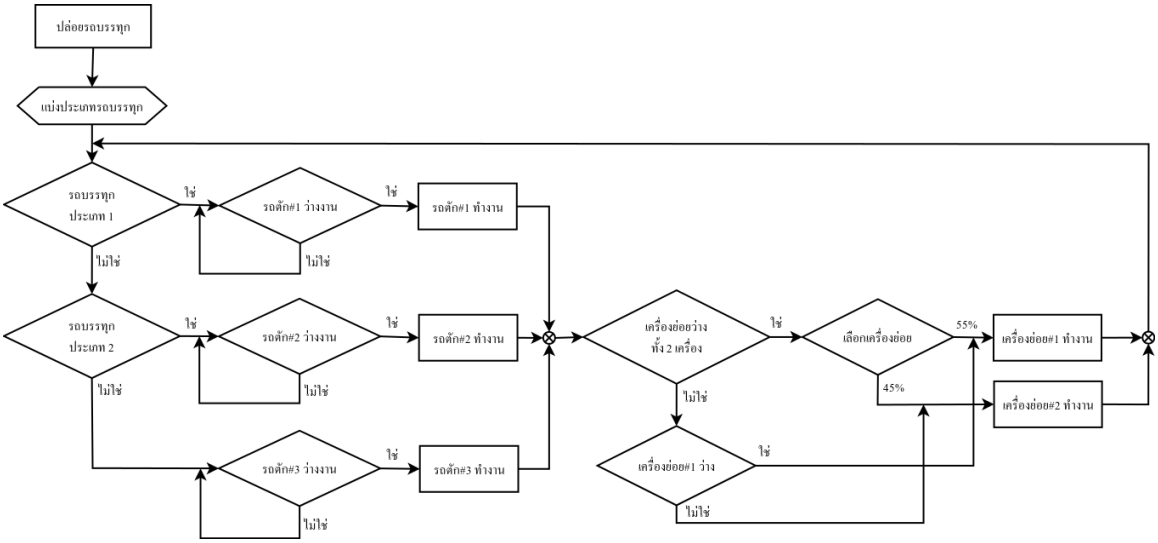
ผลการวิจัยแบ่งเป็น 4 กรณี คือ กรณีแรก ผลการจำลองระบบขนส่งหินปูนด้วยรถบรรทุกโดยใช้โปรแกรม Arena สำหรับเหมืองหินปูน บริษัท ปูนซิเมนต์ไทย (ทุ่งสง) จำกัด กรณีที่ 2 สำหรับ บริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด กรณีที่ 3 ผลการจำลองระบบขนส่งหินปูนด้วยรถบรรทุกโดยใช้โปรแกรมจาวา สำหรับเหมืองหินปูน บริษัท ปูนซิเมนต์ไทย (ทุ่งสง) จำกัด และกรณีที่ 4 สำหรับ บริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 ผลการวิจัยแบ่งได้เป็น 4 กรณี

เหมืองหินปูน	จำลองระบบด้วย	
บจก. ปูนซิเมนต์ไทย (ทุ่งสง)	กรณีที่ 1 Arena	กรณีที่ 3 จาวา
บจก. ปูนซิเมนต์ไทย (ท่าหลวง)	กรณีที่ 2 Arena	กรณีที่ 4 จาวา

3.1 กรณีที่ 1 : จำลองระบบด้วย Arena สำหรับทุ่งสง

การสร้างแบบจำลองด้วยโปรแกรม Arena สำหรับทุ่งสงนั้น แบ่งจุดตักเป็น 3 จุด มีรถตักประจำจุด ๆ ละ 1 คัน และมี 1 จุดเท ซึ่งมีเครื่องย่อย 2 เครื่อง อยู่ในบริเวณใกล้ ๆ กัน ระบบเป็นแบบตรึงจุดตักและสุ่มจุดเท ดังแผงผังการไหลของรถบรรทุกในระบบตามรูปที่ 3.1



รูปที่ 3.1 แผงผังการไหลของรถบรรทุกจากการสร้างแบบจำลองด้วยโปรแกรม Arena สำหรับทุ่งสง

การจำลองระบบด้วย Arena สำาหรับเหมืองหินปูนทุ่งสง มีข้อมูลนำเข้าในรูปของการกระจายของข้อมูลดังได้กล่าวมาแล้วในบทที่ 2 ดังนี้

1) หน่วย Create มี 3 หน่วย ได้แก่หน่วยที่ชื่อ Truck arr1stLA, Truck arr2ndLA และ Truck arr3rdLA เป็นการสร้างรถบรรทุกให้ไปรับหินที่จุดตักที่ 1, 2 และ 3 ตามลำดับ มีการกระจายเหมือนกัน คือ Exponential 2 นาที

2) หน่วย Route มี 9 หน่วย เป็นการสร้างเส้นทางให้กับรถบรรทุกเพื่อแล่นไปรับหินและเทหินเมื่อโมเดลเปลี่ยน ผู้ใช้จะต้องเปลี่ยนค่าการกระจายทั้ง 9 หน่วย เนื่องจากระยะทางระหว่างจุดตักและจุดเทจะต้องเปลี่ยนไป หน่วย Route ดังกล่าว ได้แก่ Route1 to Loaders, Route2 to Loaders, Route3 to Loaders, Route4 to Loaders, Route5 to Loaders, Route6 to Loaders, Route1 to CR, Route2 to CR และ Route3 to CR มีการกระจายแบบ Gamma = 3 + GAMM(0.282, 3.86), Lognormal = 4.21 + LOGN(0.612, 0.376), Normal = NORM(4.3, 0.2), Gamma = 3 + GAMM(0.282, 3.86), Beta = 2.65 + 1.62 * BETA(1.79, 1.95), Gamma = 3 + GAMM(0.282, 3.86), Lognormal = 4.62 + LOGN(0.60, 0.41), Normal = NORM(4.72, 0.63) และ Normal = NORM(5.38, 0.53) นาที ตามลำดับ

3) หน่วย Process มี 5 หน่วย เป็นหน่วยที่เกี่ยวกับการตักของรถตักและการย่อยของเครื่องย่อย ได้แก่หน่วยที่ชื่อ Load by EX1800 @1stLA, Load by CAT992 @2ndLA, Load by CAT992 @3rdLA, Crush by CR5 และ Crush by CR6 มีการกระจายแบบ Beta = 2.38+1.79*BETA(1.02, 1.25), Normal = NORM(3.66, 0.70), Normal = NORM(3.66, 0.70), Exponential = EXPO(1.58) และ Beta = 1+1*BETA(2.76, 1.94) นาที ตามลำดับ

ส่วนหน่วยที่ไม่จำเป็นต้องระบุค่าการกระจาย คือ หน่วย Assign เพราะเป็นการกำหนดค่าเฉพาะตัวให้กับรถบรรทุก หน่วย Station เป็นหน่วยปลายทางสำหรับรถบรรทุก โดยหน่วยต้นทาง คือ หน่วย Route และหน่วย Decide เป็นหน่วยตัดสินใจให้รถบรรทุกมีทางเลือก ตั้งแต่ 2 ทางขึ้นไป ดังตารางที่ 3.2

ตารางที่ 3.2 ข้อมูลนำเข้าสำหรับเหมืองหินปูนทุ่งสง

หน่วย	ชื่อ	ประเภทการกระจาย	ค่าการกระจาย (นาที)
Create	Truck arr1stLA	Exponential	EXPO(2.0)
	Truck arr2ndLA	Exponential	EXPO(2.0)
	Truck arr3rdLA	Exponential	EXPO(2.0)
Assign	Assign Truck1	-	-
	Assign Truck2	-	-
	Assign Truck3	-	-
Route	Route1 to Loaders	Gamma	3 + GAMM(0.282, 3.86)
	Route2 to Loaders	Lognormal	4.21 + LOGN(0.612, 0.376)
	Route3 to Loaders	Normal	NORM(4.3, 0.2)
	Route4 to Loaders	Gamma	3 + GAMM(0.282, 3.86)
	Route5 to Loaders	Beta	2.65 + 1.62 * BETA(1.79, 1.95)
	Route6 to Loaders	Gamma	3 + GAMM(0.282, 3.86)

หน่วย	ชื่อ	ประเภทการกระจาย	ค่าการกระจาย (นาที)
Route	Route1 to CR	Lognormal	4.62 + LOGN(0.60, 0.41)
	Route2 to CR	Normal	NORM(4.72, 0.63)
	Route3 to CR	Normal	NORM(5.38, 0.53)
Station	Station Loaders	-	-
	Station CR	-	-
Decide	Loader ?	-	-
	EX1800 available ?	-	-
	CAT992_1 available ?	-	-
	CAT992_2 available ?	-	-
	both CR available ?	-	-
	CR5 ?	-	-
	CR6 ?	-	-
	choose CR ?	-	-
	Return ?	-	-
Process	Load by EX1800 @1stLA	Beta	2.38+1.79*BETA(1.02, 1.25)
	Load by CAT992 @2ndLA	Normal	NORM(3.66, 0.70)
	Load by CAT992 @3rdLA	Normal	NORM(3.66, 0.70)
	Crush by CR5	Exponential	EXPO(1.58)
	Crush by CR6	Beta	1+1*BETA(2.76, 1.94)

เนื่องจากกรณีของทุ่งสงการทำงานเป็นแบบตรึงจุดตัก-สุ่มจุดเท เครื่องย่อยมี 2 เครื่อง คือ CR5 และ CR6 รถตักมี 2 ประเภท คือ CAT992 และ HITACHI EX1800 ดังนั้นจึงแบ่งการทำงานออกเป็น 10

กรณี ตามข้อมูลที่เก็บได้จากภาคสนาม โดยแต่ละกรณีประกอบด้วยจุดตัก 3 จุด ผลลัพธ์ที่ได้จากการประเมินโดย Arena แบ่งเป็น 3 ส่วน คือ จำนวนรถบรรทุกที่เหมาะสม เวลารอของรถบรรทุกที่รถตักและเครื่องย่อยและสัดส่วนการทำงานของรถตัก โดยอยู่ภายใต้เงื่อนไขที่ว่า จำนวนรถบรรทุกที่มารอต่อเที่ยวต้องไม่เกิน 1 คัน

ขั้นตอนการใช้งานโปรแกรม เป็นดังนี้

1) ให้ดับเบิ้ลคลิ๊กที่หน่วย Create ชื่อ Truck arr 1stLA ในช่อง Entities per Arrival: ใส่เลข 1 เพราะรถบรรทุกมาถึงจุกตักครั้งละ 1 คัน ใส่จำนวนรถบรรทุกตามต้องการในช่อง Max Arrivals: ซึ่งจำนวนรถบรรทุกค่านี้ จะใช้ไปตลอดช่วงเวลาที่ผู้ใช้กำหนด ทำเช่นเดียวกันกับหน่วย Create ชื่อ Truck arr 2ndLA และ Truck arr 3rdLA

2) ให้ดับเบิ้ลคลิ๊กที่หน่วย Route ชื่อ Route1 to CR ช่อง Route time: ใส่ค่าการกระจายที่ได้จากภาคสนาม เป็นค่าที่รถบรรทุกแล่นจากจุดตักไปยังจุดเท ซึ่งผู้ใช้จำเป็นต้องมีค่านี้และต้องเป็นค่าที่เที่ยงตรงด้วยเพื่อให้แบบจำลองมีความสมบูรณ์ที่สุดเท่าที่จะเป็นไปได้ ทำเช่นเดียวกันกับหน่วย Route ชื่อ Route2 to CR และ Route3 to CR

3) ให้ดับเบิ้ลคลิ๊กที่หน่วย Route ชื่อ Route4 to Loaders ช่อง Route time: ใส่ค่าการกระ-จายที่ได้จากภาคสนาม เป็นค่าที่รถบรรทุกแล่นจากจุดเทไปยังจุดตัก ทำเช่นเดียวกันกับหน่วย Route ชื่อ Route5 to Loaders และ Route6 to Loaders

4) คลิ๊ก Go เพื่อให้โปรแกรมเริ่มประมวลผล

สำหรับทุ่งสง รถตักปุ้งกี๋เสย EX1800 เคลื่อนย้ายลำบาก ดังนั้นจึงจัดให้ EX1800 อยู่ชั้นล่างทุกกรณี ถ้าพิจารณาจำนวนรถบรรทุกที่เหมาะสมใน 1 กะ จะได้

- กรณีที่ 1 จุดตัก 203 รถตัก EX1800, จุดตัก 221 รถตัก CAT992 และจุดตัก 239 รถตัก CAT992 เท่ากับ 5, 5 และ 5 คัน ตามลำดับ

- กรณีที่ 2 จุดตัก 203 รถตัก EX1800, จุดตัก 221 รถตัก CAT992 และจุดตัก 257 รถตัก CAT992 เท่ากับ 5, 5 และ 5 คัน ตามลำดับ

- กรณีที่ 3 จุดตัก 203 รถตัก EX1800, จุดตัก 221 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 เท่ากับ 5, 4 และ 4 คัน ตามลำดับ

- กรณีที่ 4 จุดตัก 203 รถตัก EX1800, จุดตัก 239 รถตัก CAT992 และจุดตัก 257 รถตัก

CAT992 เท่ากับ 5, 5 และ 4 คัน ตามลำดับ

- กรณีที่ 5 จุดตัก 203 รถตัก EX1800, จุดตัก 239 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 เท่ากับ 6, 4 และ 5 คัน ตามลำดับ

- กรณีที่ 6 จุดตัก 203 รถตัก EX1800, จุดตัก 257 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 เท่ากับ 5, 4 และ 4 คัน ตามลำดับ

- กรณีที่ 7 จุดตัก 221 รถตัก EX1800, จุดตัก 239 รถตัก CAT992 และจุดตัก 257 รถตัก CAT992 เท่ากับ 5, 4 และ 5 คัน ตามลำดับ

- กรณีที่ 8 จุดตัก 221 รถตัก EX1800, จุดตัก 239 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 เท่ากับ 5, 5 และ 5 คัน ตามลำดับ

- กรณีที่ 9 จุดตัก 221 รถตัก EX1800, จุดตัก 257 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 เท่ากับ 5, 5 และ 5 คัน ตามลำดับ และ

- กรณีที่ 10 จุดตัก 239 รถตัก EX1800, จุดตัก 257 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 เท่ากับ 5, 4 และ 5 คัน ตามลำดับ ดังแสดงในตารางที่ 3.3

ตารางที่ 3.3 จำนวนรถบรรทุกที่เหมาะสมจากการจำลองด้วย Arena สำหรับทุ่งสง

จำนวนรถบรรทุกที่เหมาะสม (คัน)					
กรณีที่	จุดตัก				
	203	221	239	257	275
1	EX1800 = 5	CAT992 = 5	CAT992 = 5		
2	EX1800 = 5	CAT992 = 5		CAT992 = 5	
3	EX1800 = 5	CAT992 = 4			CAT992 = 4
4	EX1800 = 5		CAT992 = 5	CAT992 = 4	
5	EX1800 = 6		CAT992 = 4		CAT992 = 5
6	EX1800 = 5			CAT992 = 4	CAT992 = 5

จำนวนรถบรรทุกที่เหมาะสม (คัน)					
กรณีที่	จุดตัก				
	203	221	239	257	275
7		EX1800 = 5	CAT992 = 4	CAT992 = 5	
8		EX1800 = 5	CAT992 = 5		CAT992 = 5
9		EX1800 = 5		CAT992 = 5	CAT992 = 5
10			EX1800 = 5	CAT992 = 4	CAT992 = 5

เวลารอของรถบรรทุกที่รถตักและเครื่องย่อย เป็นดังนี้

- กรณีที่ 1 จุดตัก 203 รถตัก EX1800, จุดตัก 221 รถตัก CAT992, จุดตัก 239 รถตัก CAT992, เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 1.74, 3.52, 2.69, 2.21 และ 1.92 นาที ตามลำดับ

- กรณีที่ 2 จุดตัก 203 รถตัก EX1800, จุดตัก 221 รถตัก CAT992, จุดตัก 257 รถตัก CAT992, เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 1.77, 3.43, 1.89, 1.92 และ 1.24 นาที ตามลำดับ

- กรณีที่ 3 จุดตัก 203 รถตัก EX1800, จุดตัก 221 รถตัก CAT992, จุดตัก 275 รถตัก CAT992, เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 2.12, 1.53, 1.32, 2.12 และ 1.51 นาที ตามลำดับ

- กรณีที่ 4 จุดตัก 203 รถตัก EX1800, จุดตัก 239 รถตัก CAT992, จุดตัก 257 รถตัก CAT992, เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 1.94, 2.02, 0.99, 2.37 และ 1.47 นาที ตามลำดับ

- กรณีที่ 5 จุดตัก 203 รถตัก EX1800, จุดตัก 239 รถตัก CAT992, จุดตัก 275 รถตัก CAT992, เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 3.09, 1.19, 3.30, 2.20 และ 1.98 นาที ตามลำดับ

- กรณีที่ 6 จุดตัก 203 รถตัก EX1800, จุดตัก 257 รถตัก CAT992, จุดตัก 275 รถตัก CAT992,

เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 1.66, 1.22, 3.01, 2.32 และ 1.92 นาที ตามลำดับ

- กรณีที่ 7 จุดตัก 221 รถตัก EX1800, จุดตัก 239 รถตัก CAT992, จุดตัก 257 รถตัก CAT992, เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 2.47, 1.36, 2.08, 1.89 และ 1.66 นาที ตามลำดับ

- กรณีที่ 8 จุดตัก 221 รถตัก EX1800, จุดตัก 239 รถตัก CAT992, จุดตัก 275 รถตัก CAT992, เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 2.38, 2.34, 3.16, 2.08 และ 3.09 นาที ตามลำดับ

- กรณีที่ 9 จุดตัก 221 รถตัก EX1800, จุดตัก 257 รถตัก CAT992, จุดตัก 275 รถตัก CAT992, เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 2.12, 2.05, 2.77, 2.25 และ 2.00 นาที ตามลำดับ และ

- กรณีที่ 10 จุดตัก 239 รถตัก EX1800, จุดตัก 257 รถตัก CAT992, จุดตัก 275 รถตัก CAT992, เครื่องย่อย CR5 และเครื่องย่อย CR6 รถบรรทุกมารอต่อเที่ยวเท่ากับ 1.81, 1.07, 3.10, 2.39 และ 1.52 นาที ตามลำดับ ดังแสดงในตารางที่ 3.4

ตารางที่ 3.4 เวลารอของรถบรรทุกที่รถตักและเครื่องย่อยจากการจำลองด้วย Arena สำหรับทุ่งสง

เวลารอของรถบรรทุกที่รถตักและเครื่องย่อย (นาที)							
กรณีที่	รถตักที่ระดับชั้น					เครื่องย่อย	
	203	221	239	257	275	CR5	CR6
1	EX1800 = 1.74	CAT992 = 3.52	CAT992 = 2.69			2.21	1.92
2	EX1800 = 1.77	CAT992 = 3.43		CAT992 = 1.89		1.92	1.24
3	EX1800 = 2.12	CAT992 = 1.53			CAT992 = 1.32	2.12	1.51
4	EX1800 = 1.94		CAT992 = 2.02	CAT992 = 0.99		2.37	1.47
5	EX1800 = 3.09		CAT992 = 1.19		CAT992 = 3.30	2.20	1.98
6	EX1800 = 1.66			CAT992 = 1.22	CAT992 = 3.01	2.32	1.92

จำนวนรถบรรทุกที่เหมาะสม (คัน)							
กรณีที่	จุดตัก					เครื่องย่อย	
	203	221	239	257	275	CR5	CR6
7		EX1800 = 2.47	CAT992 = 1.36	CAT992 = 2.08		1.89	1.66
8		EX1800 = 2.38	CAT992 = 2.34		CAT992 = 3.16	2.08	3.09
9		EX1800 = 2.12		CAT992 = 2.05	CAT992 = 2.77	2.25	2.00
10			EX1800 = 1.81	CAT992 = 1.07	CAT992 = 3.10	2.39	1.52

พิจารณาสัดส่วนการทำงานของรถตัก ถ้าเท่ากับ 1.00 หมายความว่า รถตักทำงานตลอดช่วงเวลาที่กำหนดคือ ตัก เตรียมเท เท เตรียมตัก อยู่ตลอดเวลา แต่ถ้าน้อยกว่า 1.00 หมายถึง รถตักมีเวลารอ ซึ่งสัดส่วนการทำงานของรถตักแต่ละกรณี เป็นดังนี้

- กรณีที่ 1 จุดตัก 203 รถตัก EX1800, จุดตัก 221 รถตัก CAT992 และจุดตัก 239 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.88, 0.97 และ 0.93 ตามลำดับ

- กรณีที่ 2 จุดตัก 203 รถตัก EX1800, จุดตัก 221 รถตัก CAT992 และจุดตัก 257 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.90, 0.98 และ 0.92 ตามลำดับ

- กรณีที่ 3 จุดตัก 203 รถตัก EX1800, จุดตัก 221 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.89, 0.84 และ 0.84 ตามลำดับ

- กรณีที่ 4 จุดตัก 203 รถตัก EX1800, จุดตัก 239 รถตัก CAT992 และจุดตัก 257 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.87, 0.90 และ 0.75 ตามลำดับ

- กรณีที่ 5 จุดตัก 203 รถตัก EX1800, จุดตัก 239 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.97, 0.75 และ 0.94 ตามลำดับ

- กรณีที่ 6 จุดตัก 203 รถตัก EX1800, จุดตัก 257 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.90, 0.72 และ 0.95 ตามลำดับ

- กรณีที่ 7 จุดตัก 221 รถตัก EX1800, จุดตัก 239 รถตัก CAT992 และจุดตัก 257 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.94, 0.83 และ 0.92 ตามลำดับ

- กรณีที่ 8 จุดตัก 203 รถตัก EX1800, จุดตัก 239 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.89, 0.91 และ 0.94 ตามลำดับ

- กรณีที่ 9 จุดตัก 221 รถตัก EX1800, จุดตัก 257 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.93, 0.91 และ 0.94 ตามลำดับ และ

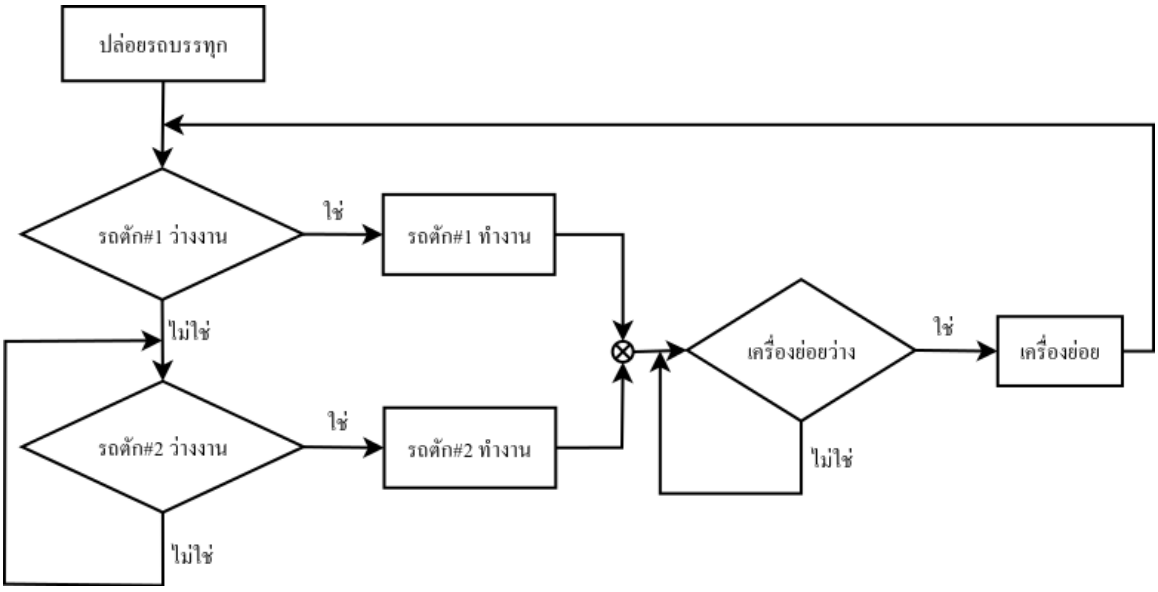
- กรณีที่ 10 จุดตัก 239 รถตัก EX1800, จุดตัก 257 รถตัก CAT992 และจุดตัก 275 รถตัก CAT992 ได้สัดส่วนการทำงานของรถตักเท่ากับ 0.88, 0.78 และ 0.94 ตามลำดับ ดังแสดงในตารางที่ 3.5

ตารางที่ 3.5 สัดส่วนการทำงานของรถตักจากการจำลองด้วย Arena สำหรับทุ่งสง

สัดส่วนการทำงานของรถตัก					
กรณีที่	จุดตัก				
	203	221	239	257	275
1	EX1800 = 0.88	CAT992 = 0.97	CAT992 = 0.93		
2	EX1800 = 0.90	CAT992 = 0.98		CAT992 = 0.92	
3	EX1800 = 0.89	CAT992 = 0.84			CAT992 = 0.84
4	EX1800 = 0.87		CAT992 = 0.90	CAT992 = 0.75	
5	EX1800 = 0.97		CAT992 = 0.75		CAT992 = 0.94
6	EX1800 = 0.90			CAT992 = 0.72	CAT992 = 0.95
7		EX1800 = 0.94	CAT992 = 0.83	CAT992 = 0.92	
8		EX1800 = 0.89	CAT992 = 0.91		CAT992 = 0.94
9		EX1800 = 0.93		CAT992 = 0.91	CAT992 = 0.94
10			EX1800 = 0.88	CAT992 = 0.78	CAT992 = 0.94

3.2 กรณีที่ 2 : จำลองระบบด้วย Arena สำหรับท่าหลวง

การสร้างแบบจำลองด้วยโปรแกรม Arena สำหรับท่าหลวงนั้น แบ่งจุดตักเป็น 2 จุด มีรถตักประจำจุด ๆ ละ 1 คัน และมี 1 จุดเท ซึ่งมีเครื่องย่อย 2 เครื่อง อยู่ในบริเวณใกล้ ๆ กัน ระบบเป็นแบบสุ่มจุดตักและตรึงจุดเท ดังแผงผังการไหลของรถบรรทุกในระบบตามรูปที่ 3.2



รูปที่ 3.2 แผงผังการไหลของรถบรรทุกจากการสร้างแบบจำลองด้วยโปรแกรม Arena สำหรับท่าหลวง

การจำลองระบบด้วย Arena สำหรับเหมืองหินปูนท่าหลวง มีข้อมูลนำเข้าในรูปของการกระจายของข้อมูลดังได้กล่าวมาแล้วในบทที่ 2 ดังนี้

1) หน่วย Create มี 6 หน่วย ได้แก่หน่วยที่ชื่อ Truck1ArrAB_1800, Truck2ArrAB_1100, Truck3ArrAB_992, Truck4ArrA_1800, Truck5ArrA_1100 และ Truck6ArrA_992 เป็นการสร้างรถบรรทุกเข้าสู่ระบบ ให้ไปรับหินที่จุดตัก เขา AB และเขา A ซึ่งแต่ละจุดตักแบ่งรถตักเป็น 3 ประเภท คือ HITACHI EX1800, HITACHI EX1100 และ CAT992 ตามลำดับ 3 หน่วยแรก หรือทีจุดตักเขา AB มีการกระจายเหมือนกัน คือ Beta = 4 + 23 * BETA(0.647, 3) นาที ส่วน 3 หน่วยถัดมา หรือที่จุดตักเขา A ก็มีการกระจายเหมือนกัน คือ Lognormal = 3 + LOGN(6.32, 4.88) นาที

2) หน่วย Route มี 6 หน่วย เป็นการสร้างเส้นทางให้กับรถบรรทุกเพื่อแล่นไปรับหินและเทหิน เมื่อโมเดลเปลี่ยน ผู้ใช้จะต้องเปลี่ยนค่าการกระจายทั้ง 6 หน่วย เนื่องจากระยะทางระหว่างจุดตักและจุดเทจะต้องเปลี่ยนไป หน่วย Route ดังกล่าว ได้แก่ RouteToLA_AB, RouteToDAfromAB, RouteToAB, RouteToLA_A, RouteToDAfromA และ RouteToA มีการกระจายแบบ Lognormal = 6 + LOGN(2.6, 2.5), Normal = NORM(10.5, 1.4), Lognormal = 6 + LOGN(2.6, 2.5), Normal = NORM(8.8, 1.47), Lognormal = 7 + LOGN(1.76, 1.15) และ Normal = NORM(8.8, 1.47) นาที ตามลำดับ

3) หน่วย Process มี 5 หน่วย เป็นหน่วยที่เกี่ยวกับการตักของรถตักและการย่อยของเครื่องย่อย ได้แก่หน่วยที่ชื่อ ProcessLoad 1800, ProcessLoad 1100, ProcessLoad 992, ProcessCrush TL และ ProcessCrush KW มีการกระจายแบบ Gamma = 2.28 + GAMM(0.287, 5.36), Normal = NORM(3.92, 0.971), Erlang = 2 + ERLA(0.521, 4), Weibull = 2.68 + WEIB(1.31, 2.06) และ Beta = 2.44 + 2.17 * BETA(2.46, 2.48) นาที ตามลำดับ

ส่วนหน่วยที่ไม่จำเป็นต้องระบุค่าการกระจาย คือ หน่วย Assign เพราะเป็นการกำหนดค่าเฉพาะตัวให้กับรถบรรทุก หน่วย Station เป็นหน่วยปลายทางสำหรับรถบรรทุก โดยหน่วยต้นทาง คือ หน่วย Route และหน่วย Decide เป็นหน่วยตัดสินใจให้รถบรรทุกมีทางเลือก ตั้งแต่ 2 ทางขึ้นไป ดังตารางที่ 3.6

ตารางที่ 3.6 ข้อมูลนำเข้าสำหรับเหมืองหินปูนท่าหลวง

หน่วย	ชื่อ	ประเภทการกระจาย	ค่าการกระจาย (นาที)
Create	Truck1ArrAB_1800	Beta	4 + 23 * BETA(0.647, 3)
	Truck2ArrAB_1100	Beta	4 + 23 * BETA(0.647, 3)
	Truck3ArrAB_992	Beta	4 + 23 * BETA(0.647, 3)
	Truck4ArrA_1800	Lognormal	3 + LOGN(6.32, 4.88)
	Truck5ArrA_1100	Lognormal	3 + LOGN(6.32, 4.88)
	Truck6ArrA_992	Lognormal	3 + LOGN(6.32, 4.88)
Assign	AssignToTruck1	-	-

หน่วย	ชื่อ	ประเภทการกระจาย	ค่าการกระจาย (นาที)
Assign	AssignToTruck2	-	-
	AssignToTruck3	-	-
	AssignToTruck4	-	-
	AssignToTruck5	-	-
	AssignToTruck6	-	-
Route	RouteToLA_AB	Lognormal	6 + LOGN(2.6, 2.5)
	RouteToDAfromAB	Normal	NORM(10.5, 1.4)
	RouteToAB	Lognormal	6 + LOGN(2.6, 2.5)
	RouteToLA_A	Normal	NORM(8.8, 1.47)
	RouteToDAfromA	Lognormal	7 + LOGN(1.76, 1.15)
	RouteToA	Normal	NORM(8.8, 1.47)
Station	StationLA_AB	-	-
	StationLA_A	-	-
	StationDA	-	-
Decide	Loader_AB ?	-	-
	Loader_A ?	-	-
	1800 available ?	-	-
	1100 available ?	-	-
	992 available ?	-	-
Decide	Route ?	-	-
	CR ?	-	-

หน่วย	ชื่อ	ประเภทการกระจาย	ค่าการกระจาย (นาที)
Decide	TL available ?	-	-
	KW available ?	-	-
	Return ?	-	-
Process	ProcessLoad 1800	Gamma	2.28 + GAMM(0.287, 5.36)
	ProcessLoad 1100	Normal	NORM(3.92, 0.971)
	ProcessLoad 992	Erlang	2 + ERLA(0.521, 4)
	ProcessCrush TL	Weibull	2.68 + WEIB(1.31, 2.06)
	ProcessCrush KW	Beta	2.44 + 2.17 * BETA(2.46, 2.48)

กรณีของท่าหลวงการทำงานเป็นแบบตรึงจุดเท เครื่องย่อยมี 2 เครื่อง คือ เครื่องย่อย TL และเครื่องย่อย KW ใน 1 กะ จะย่อยแค่ 1 เครื่อง รถตักมี 3 ประเภท คือ CAT992, HITACHI EX1100 และ HITACHI EX1800 ดังนั้น จึงแบ่งการทำงานออกเป็น 16 กรณี ตามข้อมูลที่เก็บได้จากภาคสนาม โดยแต่ละกรณีประกอบด้วยจุดตัก 2 จุด ผลลัพธ์ที่ได้จากการประเมินโดย Arena แบ่งเป็น 3 ส่วน คือ จำนวนรถบรรทุกที่เหมาะสม เวลารอของรถบรรทุกที่รถตักและเครื่องย่อยและสัด-ส่วนการทำงานของรถตักโดยอยู่ภายใต้เงื่อนไขที่ว่า จำนวนรถบรรทุกที่มารอต่อเที่ยวต้องไม่เกิน 1 คัน

ขั้นตอนการใช้งานโปรแกรม เป็นดังนี้

1) ให้ผู้ใช้เลือกจุดตักและรถตัก เช่น ผู้ใช้ต้องการให้รถบรรทุกไปรับหินที่เขา AB โดยรถตักเป็น EX1100 ให้ดับเบิ้ลคลิ๊กที่หน่วย Create ชื่อ Truck2ArrAB_1100 ในช่อง Entities per Arrival: ใส่เลข 1 เพราะรถบรรทุกมาถึงจุกตักครั้งละ 1 คัน ใส่จำนวนรถบรรทุกตามต้องการในช่อง Max Arrivals: ซึ่งจำนวนรถบรรทุกค่านี้ จะใช้ไปตลอดช่วงเวลาที่ผู้ใช้กำหนด

2) ให้ดับเบิ้ลคลิ๊กที่หน่วย Submodel ชื่อ AssignTruck1-3 และดับเบิ้ลคลิ๊กที่หน่วย Assign ชื่อ AssignToTruck2 คลิ๊ก Add ช่อง Type: เลือก Variable ช่อง Variable Name: เลือก CR_TL และช่อง New Value: ใส่เลข 0 เพราะต้องการย่อย เครื่องย่อย TL

3) ให้ดับเบิ้ลคลิ๊กที่หน่วย Decide ชื่อ CR ? ช่อง Type: เลือก 2-way by Condition ช่อง If: เลือก

Variable ช่อง Named: เลือก CR_TL ช่อง Is: เลือก == และช่อง Value: เลือกเลข 0 ตามที่ได้กำหนดไว้ในข้อ 2)

4) คลิ๊ก Go เพื่อให้โปรแกรมเริ่มประมวลผล

ถ้าพิจารณาจำนวนรถบรรทุกที่เหมาะสมใน 1 กะ จะได้

- กรณีที่ 1 จุดตัก เขา AB รถตัก EX1800 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย TL ใช้รถบรรทุกเท่ากับ 4 และ 4 คัน ตามลำดับ
- กรณีที่ 2 จุดตัก เขา AB รถตัก EX1800 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย KW ใช้รถบรรทุกเท่ากับ 4 และ 4 คัน ตามลำดับ
- กรณีที่ 3 จุดตัก เขา AB รถตัก EX1800 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย TL ใช้รถบรรทุกเท่ากับ 4 และ 3 คัน ตามลำดับ
- กรณีที่ 4 จุดตัก เขา AB รถตัก EX1800 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย KW ใช้รถบรรทุกเท่ากับ 4 และ 4 คัน ตามลำดับ
- กรณีที่ 5 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก EX1800 ย่อยด้วยเครื่องย่อย TL ใช้รถบรรทุกเท่ากับ 3 และ 4 คัน ตามลำดับ
- กรณีที่ 6 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก EX1800 ย่อยด้วยเครื่องย่อย KW ใช้รถบรรทุกเท่ากับ 4 และ 4 คัน ตามลำดับ
- กรณีที่ 7 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย TL ใช้รถบรรทุกเท่ากับ 3 และ 3 คัน ตามลำดับ
- กรณีที่ 8 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย KW ใช้รถบรรทุกเท่ากับ 4 และ 3 คัน ตามลำดับ
- กรณีที่ 9 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย TL ใช้รถบรรทุกเท่ากับ 4 และ 3 คัน ตามลำดับ
- กรณีที่ 10 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย KW ใช้รถบรรทุกเท่ากับ 4 และ 4 คัน ตามลำดับ
- กรณีที่ 11 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก EX1800 ย่อยด้วยเครื่องย่อย TL ใช้รถบรรทุกเท่ากับ 4 และ 4 คัน ตามลำดับ

- กรณีที่ 12 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก EX1800 ย่อยด้วยเครื่องย่อย KW ใช้รถบรรทุกเท่ากับ 4 และ 4 คัน ตามลำดับ

- กรณีที่ 13 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย TL ใช้รถบรรทุกเท่ากับ 3 และ 4 คัน ตามลำดับ

- กรณีที่ 14 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย KW ใช้รถบรรทุกเท่ากับ 4 และ 4 คัน ตามลำดับ

- กรณีที่ 15 จุดตัก เขา AB รถตัก C AT992 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย TL ใช้รถบรรทุกเท่ากับ 4 และ 3 คัน ตามลำดับ

- กรณีที่ 16 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย KW ใช้รถบรรทุกเท่ากับ 4 และ 3 คัน ตามลำดับ ดังแสดงในตารางที่ 3.7

ตารางที่ 3.7 จำนวนรถบรรทุกที่เหมาะสมจากการจำลองด้วย Arena สำหรับท่าหลวง

จำนวนรถบรรทุกที่เหมาะสม (คัน)						
กรณีที่	จุดตัก					
	AB			A		
	EX1800	EX1100	CAT992	EX1800	EX1100	CAT992
1	4 ย่อย TL				4 ย่อย TL	
2	4 ย่อย KW				4 ย่อย KW	
3	4 ย่อย TL					3 ย่อย TL
4	4 ย่อย KW					4 ย่อย KW
5		3 ย่อย TL		4 ย่อย TL		
6		4 ย่อย KW		4 ย่อย KW		
7		3 ย่อย TL			3 ย่อย TL	
8		4 ย่อย KW			3 ย่อย KW	

จำนวนรถบรรทุกที่เหมาะสม (คัน)						
กรณีที่	จุดตัก					
	AB			A		
	EX1800	EX1100	CAT992	EX1800	EX1100	CAT992
9		4 ย่อย TL				3 ย่อย TL
10		4 ย่อย KW				4 ย่อย KW
11			4 ย่อย TL	4 ย่อย TL		
12			4 ย่อย KW	4 ย่อย KW		
13			3 ย่อย TL		4 ย่อย TL	
14			4 ย่อย KW		4 ย่อย KW	
15			4 ย่อย TL			3 ย่อย TL
16			4 ย่อย KW			3 ย่อย KW

เวลารอของรถบรรทุกที่รถตักและเครื่องย่อย เป็นดังนี้

- กรณีที่ 1 จุดตัก เขา AB รถตัก EX1800 จุดตัก เขา A รถตัก EX1100 และเครื่องย่อย TL รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.34, 0.50 และ 3.93 นาที ตามลำดับ

- กรณีที่ 2 จุดตัก เขา AB รถตัก EX1800 จุดตัก เขา A รถตัก EX1100 และเครื่องย่อย KW รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.51, 0.40 และ 3.17 นาที ตามลำดับ

- กรณีที่ 3 จุดตัก เขา AB รถตัก EX1800 จุดตัก เขา A รถตัก CAT992 และเครื่องย่อย TL รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.21, 0.00 และ 2.37 นาที ตามลำดับ

- กรณีที่ 4 จุดตัก เขา AB รถตัก EX1800 จุดตัก เขา A รถตัก CAT992 และเครื่องย่อย KW รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.33, 0.00 และ 2.97 นาที ตามลำดับ

- กรณีที่ 5 จุดตัก เขา AB รถตัก EX1100 จุดตัก เขา A รถตัก EX1800 และเครื่องย่อย TL รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.19, 0.43 และ 2.75 นาที ตามลำดับ

- กรณีที่ 6 จุดตัก เขา AB รถตัก EX1100 จุดตัก เขา A รถตัก EX1800 และเครื่องย่อย KW รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.52, 0.54 และ 3.05 นาที ตามลำดับ

- กรณีที่ 7 จุดตัก เขา AB รถตัก EX1100 จุดตัก เขา A รถตัก EX1100 และเครื่องย่อย TL รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.90, 0.90 และ 1.28 นาที ตามลำดับ

- กรณีที่ 8 จุดตัก เขา AB รถตัก EX1100 จุดตัก เขา A รถตัก EX1100 และเครื่องย่อย KW รถบรรทุกมารอต่อเที่ยวเท่ากับ 1.59, 1.19 และ 1.14 นาที ตามลำดับ

- กรณีที่ 9 จุดตัก เขา AB รถตัก EX1100 จุดตัก เขา A รถตัก CAT992 และเครื่องย่อย TL รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.66, 0.55 และ 2.35 นาที ตามลำดับ

- กรณีที่ 10 จุดตัก เขา AB รถตัก EX1100 จุดตัก เขา A รถตัก CAT992 และเครื่องย่อย KW รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.50, 0.34 และ 3.23 นาที ตามลำดับ

- กรณีที่ 11 จุดตัก เขา AB รถตัก CAT992 จุดตัก เขา A รถตัก EX1800 และเครื่องย่อย TL รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.65, 0.56 และ 3.74 นาที ตามลำดับ

- กรณีที่ 12 จุดตัก เขา AB รถตัก CAT992 จุดตัก เขา A รถตัก EX1800 และเครื่องย่อย KW รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.62, 0.53 และ 2.94 นาที ตามลำดับ

- กรณีที่ 13 จุดตัก เขา AB รถตัก CAT992 จุดตัก เขา A รถตัก EX1100 และเครื่องย่อย TL รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.04, 0.40 และ 2.29 นาที ตามลำดับ

- กรณีที่ 14 จุดตัก เขา AB รถตัก CAT992 จุดตัก เขา A รถตัก EX1100 และเครื่องย่อย KW รถบรรทุกมารอต่อเที่ยวเท่ากับ 0.56, 0.54 และ 2.97 นาที ตามลำดับ

- กรณีที่ 15 จุดตัก เขา AB รถตัก CAT992 จุดตัก เขา A รถตัก CAT992 และเครื่องย่อย TL รถบรรทุกมารอต่อเที่ยวเท่ากับ 1.32, 1.32 และ 1.78 นาที ตามลำดับ

- กรณีที่ 16 จุดตัก เขา AB รถตัก CAT992 จุดตัก เขา A รถตัก CAT992 และเครื่องย่อย KW รถบรรทุกมารอต่อเที่ยวเท่ากับ 1.31, 1.31 และ 1.00 นาที ตามลำดับ ตามตารางที่ 3.8

ตารางที่ 3.8 เวลารอของรถบรรทุกที่รถตักและเครื่องย่อยจากการจำลองด้วย Arena สำหรับท่าหลวง

เวลารอของรถบรรทุกที่รถตักและเครื่องย่อย (นาที)								
กรณีที่	รถตักที่จุดตัก						เครื่องย่อย	
	AB			A			TL	KW
	EX1800	EX1100	CAT992	EX1800	EX1100	CAT992		
1	0.34				0.50		3.93	
2	0.51				0.40			3.17
3	0.21					0.00	2.37	
4	0.33					0.00		2.97
5		0.19		0.43			2.75	
6		0.52		0.54				3.05
7		0.90			0.90		1.28	
8		1.59			1.19			1.14
9		0.66				0.55	2.35	
10		0.50				0.34		3.23
11			0.65	0.56			3.74	
12			0.62	0.53				2.94
13			0.04		0.40		2.29	
14			0.56		0.54			2.97
15			1.32			1.32	1.78	
16			1.31			1.31		1.00

สัดส่วนการทำงานของรถตักแต่ละกรณี เป็นดังนี้

- กรณีที่ 1 จุดตัก เขา AB รถตัก EX1800 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย TL สัดส่วนการทำงานของรถตักเท่ากับ 0.48 และ 0.52 ตามลำดับ

- กรณีที่ 2 จุดตัก เขา AB รถตัก EX1800 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย KW สัดส่วนการทำงานของรถตักเท่ากับ 0.48 และ 0.53 ตามลำดับ

- กรณีที่ 3 จุดตัก เขา AB รถตัก EX1800 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย TL สัดส่วนการทำงานของรถตักเท่ากับ 0.50 และ 0.21 ตามลำดับ

- กรณีที่ 4 จุดตัก เขา AB รถตัก EX1800 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย KW สัดส่วนการทำงานของรถตักเท่ากับ 0.48 และ 0.29 ตามลำดับ

- กรณีที่ 5 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก EX1800 ย่อยด้วยเครื่องย่อย TL สัดส่วนการทำงานของรถตักเท่ากับ 0.41 และ 0.54 ตามลำดับ

- กรณีที่ 6 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก EX1800 ย่อยด้วยเครื่องย่อย KW สัดส่วนการทำงานของรถตักเท่ากับ 0.52 และ 0.52 ตามลำดับ

- กรณีที่ 7 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย TL สัดส่วนการทำงานของรถตักเท่ากับ 0.45 และ 0.45 ตามลำดับ

- กรณีที่ 8 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย KW สัดส่วนการทำงานของรถตักเท่ากับ 0.47 และ 0.48 ตามลำดับ

- กรณีที่ 9 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย TL สัดส่วนการทำงานของรถตักเท่ากับ 0.52 และ 0.43 ตามลำดับ

- กรณีที่ 10 จุดตัก เขา AB รถตัก EX1100 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย KW สัดส่วนการทำงานของรถตักเท่ากับ 0.51 และ 0.59 ตามลำดับ

- กรณีที่ 11 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก EX1800 ย่อยด้วยเครื่องย่อย TL สัดส่วนการทำงานของรถตักเท่ากับ 0.52 และ 0.52 ตามลำดับ

- กรณีที่ 12 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก EX1800 ย่อยด้วยเครื่องย่อย KW สัดส่วนการทำงานของรถตักเท่ากับ 0.52 และ 0.54 ตามลำดับ

- กรณีที่ 13 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย

TL สัดส่วนการทำงานของรถตักเท่ากับ 0.41 และ 0.52 ตามลำดับ

- กรณีที่ 14 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก EX1100 ย่อยด้วยเครื่องย่อย KW สัดส่วนการทำงานของรถตักเท่ากับ 0.52 และ 0.53 ตามลำดับ

- กรณีที่ 15 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย TL สัดส่วนการทำงานของรถตักเท่ากับ 0.45 และ 0.45 ตามลำดับ

- กรณีที่ 16 จุดตัก เขา AB รถตัก CAT992 และจุดตัก เขา A รถตัก CAT992 ย่อยด้วยเครื่องย่อย KW สัดส่วนการทำงานของรถตักเท่ากับ 0.46 และ 0.46 ตามลำดับ ดังแสดงในตารางที่ 3.9

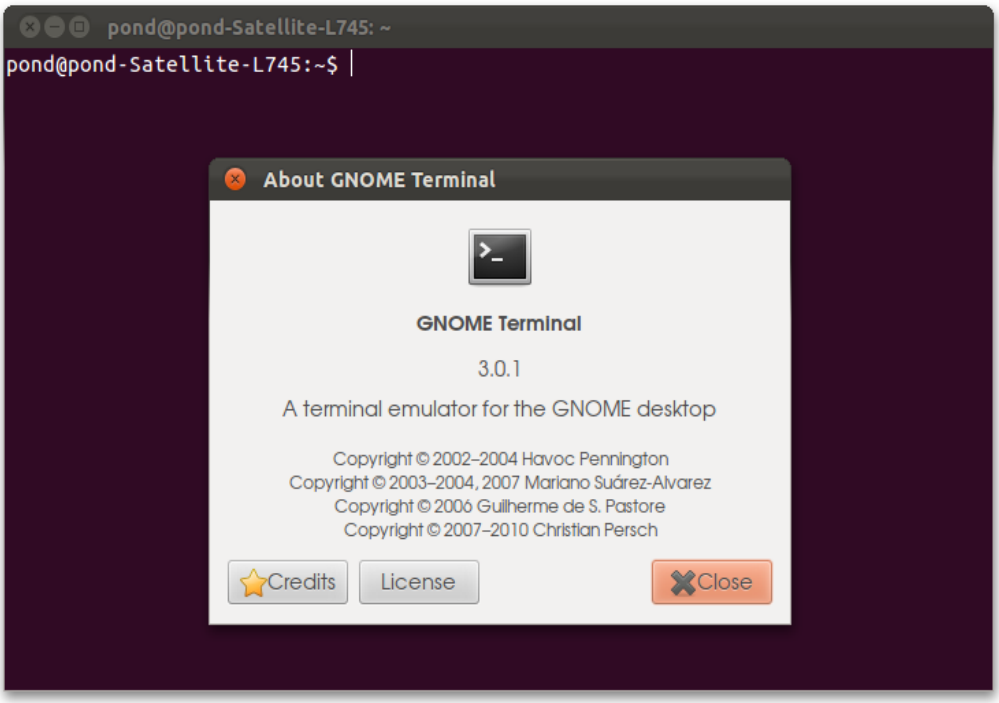
ตารางที่ 3.9 สัดส่วนการทำงานของรถตักจากการจำลองด้วย Arena สำหรับท่าหลวง

สัดส่วนการทำงานของรถตัก						
กรณีที่	จุดตัก					
	AB			A		
	EX1800	EX1100	CAT992	EX1800	EX1100	CAT992
1	0.48				0.52	
2	0.48				0.53	
3	0.50					0.21
4	0.48					0.29
5		0.41		0.54		
6		0.52		0.52		
7		0.45			0.45	
8		0.47			0.48	
9		0.52				0.43
10		0.51				0.59
11			0.52	0.52		

สัดส่วนการทำงานของรถตัก						
กรณีที่	จุดตัก					
	AB			A		
	EX1800	EX1100	CAT992	EX1800	EX1100	CAT992
12			0.52	0.54		
13			0.41		0.52	
14			0.52		0.53	
15			0.45			0.45
16			0.46			0.46

3.3 กรณีที่ 3 : จำลองระบบด้วยจาวา สำหรับทุ่งสง

ดังได้กล่าวไปแล้วในบทที่ 2 โปรแกรมจาวาที่เขียนขึ้นจะประมวลผลผ่านคอมมานด์ไลน์ ที่ชื่อว่าเทอร์มินอล (Terminal) ในระบบปฏิบัติการลีนุกซ์ ลักษณะของเทอร์มินอลเป็นดังรูปที่ 3.3



รูปที่ 3.3 โปรแกรมที่ใช้ประมวลผลผ่านคอมมานด์ไลน์ชื่อว่า "เทอร์มินอล"

ข้อมูลที่ผู้ใช้ต้องเตรียมให้พร้อมสำหรับใช้งานโปรแกรม ได้แก่

- จำนวนจุดตัก (Number of Loading areas) ที่ต้องการประมวลผล ไม่เกิน 3 จุด
- ชื่อรถบรรทุก (Truck name)
- จำนวนรถบรรทุก (Number of trucks) โดยรถบรรทุกต้องเป็นรุ่นเดียวกันทุกคัน
- ความจุของดัมพ์รถบรรทุก (Dump capacity) เป็นตัน
- ความเร็วเฉลี่ยบรรทุกหนัก (Average haul velocity) เป็นกิโลเมตรต่อชั่วโมง
- ความเร็วเฉลี่ยรถเปล่า (Average return velocity) เป็นกิโลเมตรต่อชั่วโมง
- เวลาเทของรถบรรทุกต่อดั้มพ์ (Dump time) ในรูปการกระจายข้อมูลแบบการกระจายปกติ เป็นนาทีโดยโปรแกรมแนะนำในการหาค่าการกระจายคือ Arena 13 ในส่วนของ Input Analyzer
- ชื่อรถตัก (Loader name)

- เวลาตักของรถตักตั้งแต่เริ่มตักจนเต็มดัมพ์รถบรรทุก (Load time) ในรูปการกระจายข้อมูลแบบการกระจายปกติ เป็นนาที

- จำนวนเครื่องย่อย (Number of crushers) ไม่เกิน 2 เครื่อง

- ชื่อเครื่องย่อย (Crusher name)

- กำลังการผลิตของเครื่องย่อย (Crusher capacity) เป็นตันต่อชั่วโมง

- ระยะทางระหว่างจุดเทและจุดตัก (Distance between dumping area and loading areas) เป็นเมตร

เมื่อ Program เริ่มทำงาน ผู้ใช้จะต้องป้อนจำนวนจุดตักไม่เกิน 3 จุด ดังได้กล่าวมาแล้วก่อนหน้า

```
How many loading area ? :
```

ถัดไปเป็นชื่อรถบรรทุก

```
Name the truck :
```

จำนวนรถบรรทุก

```
How many CAT777s ? :
```

ความจุของดัมพ์รถบรรทุก

```
Dump capacity of CAT777 (t) :
```

ความเร็วเฉลี่ยบรรทุกหนัก

```
Average haul velocity of CAT777 (km/h) :
```

ความเร็วเฉลี่ยรถเปล่า

```
Average return velocity of CAT777 (km/h) :
```

เวลาเทของรถบรรทุกในรูปการกระจายแบบปกติ

```
Dump time of CAT777 in term of Normal
Distribution :
```

ระบุค่าเฉลี่ย

```
Mean =
```

ระบุค่าเบี่ยงเบนมาตรฐาน

```
StdDev =
```

ชื่อรถตักจุดตักที่ 1 กรณีเลือกจำนวนจุดตักเป็น 3 จุด

```
Name the loader at 1st loading area :
```

เวลาตักของรถตักในรูปการกระจายแบบปกติ

```
Loading time of CAT992 in term of Normal
```

Distribution :

ชื่อรถตักจุดตักที่ 2

Name the loader at 2nd loading area :

เวลาตักของรถตักในรูปการกระจายแบบปกติ

Loading time of EX1100 in term of Normal Distribution :

ชื่อรถตักจุดตักที่ 3

Name the loader at 3rd loading area :

เวลาตักของรถตักในรูปการกระจายแบบปกติ

Loading time of EX1800 in term of Normal Distribution :

จำนวนเครื่องย่อย ไม่เกิน 2 เครื่อง

How many crushers ? :

ชื่อเครื่องย่อยที่ 1 กรณีเลิก 2 เครื่องย่อย

Name the 1st crusher :

กำลังการผลิตของเครื่องย่อย

Capacity of CR1 (t/h) :

ชื่อเครื่องย่อยที่ 2

Name the 2nd crusher :

กำลังการผลิตของเครื่องย่อย

Capacity of CR2 (t/h) :

ระยะทางระหว่างจุดเทและรถตัก ณ จุดตักที่ 1

Distance between dumping area and CAT992 (m.) :

ระยะทางระหว่างจุดเทและรถตัก ณ จุดตักที่ 2

Distance between dumping area and EX1100 (m.) :

ระยะทางระหว่างจุดเทและรถตัก ณ จุดตักที่ 3

Distance between dumping area and EX1800 (m.) :

ระบุว่าต้องการดูผลการทำงานที่จุดตักหรือจุดเท

Would you like to view the system at loading areas or dumping areas ?:

“DA” คือ ดูผลการทำงานที่จุดเทและ “LA” คือ ดูผลการทำงานที่จุดตักทุกจุด หลังจากนั้นโปรแกรมก็จะแสดงการคำนวณจำนวนรถบรรทุกที่พอดีสำหรับแต่ละจุดตัก แล้วแสดงว่าจะต้องใช้รถบรรทุกคันไหนบ้างจากรถบรรทุกที่มีอยู่ทั้งหมด จำนวนรถบรรทุกที่ผู้ใช้ระบุมาอาจจะเกิน พอดีหรือไม่พอก็ได้ แต่โปรแกรมจะระบุให้เห็นว่าจำนวนเท่าใดจึงจะพอดี แล้วจะนำค่านี้มาคำนวณในขั้นต่อไป จากนั้นโปรแกรมก็จะแสดงสถานะของรถบรรทุกที่จุดตักหรือจุดเทขึ้นอยู่กับผู้ใช้เลือก แสดงตั้งแต่เวลาปัจจุบันไปจนถึง 8 ชั่วโมง ของเที่ยวสุดท้ายที่มาถึงจุดตักหรือจุดเท ส่วนการประเมินของโปรแกรมจาวาสำหรับทุ่งสงจะเลือกกรณีที่จุดตักมากที่สุด คือ 3 จุดตัก ได้แก่ ชั้น 203, 221 และ 239 เป็นหลัก เนื่องจากเป็นระดับชั้นที่ต่ำที่สุดและปริมาณสำรองมีมากกว่าระดับชั้นบน และกำลังการผลิตของเครื่องย่อย 2 เครื่อง ยังสามารถรับที่จุดตัก 3 จุดตักได้ แล้วเลือกดูผลการประเมินที่จุดตัก ซึ่งการประเมินเป็นไปตามภาคผนวก ข

3.4 กรณีที่ 4 : จำลองระบบด้วยจาวา สำหรับท่าหลวง

การประเมินระบบลำเลียงหินด้วยโปรแกรมจาวาสำหรับท่าหลวงนั้น โดยปกติท่าหลวงจะย่อยแค่เครื่องย่อยเดียวและจุดตักจะเป็น 2 จุด ในที่นี้จะเลือกเครื่องย่อย TL และจุดตักเป็นเขา A และ เขา AB ซึ่งกำลังการผลิตของเครื่องย่อยยังคงรับหินจากจุดตักทั้ง 2 จุดตักได้อยู่ แล้วจะเลือกดูผลการประเมินที่จุดเท ซึ่งการประเมินเป็นดังภาคผนวก ข

บทที่ 4
วิจารณ์ผลการวิจัย

จากผลการวิจัย ในส่วนของโปรแกรมสำเร็จรูป Arena นั้น ดังที่ได้กล่าวมาแล้วในบทต้น ๆ ว่า รุ่นที่ผู้วิจัยใช้ในการประเมินนั้นเป็นรุ่นทดลองใช้ (Student version) ดังนั้นไม่ว่าจะเป็นความสามารถของตัวโปรแกรม ปริมาณงานที่โปรแกรมทำได้ จะด้อยกว่ารุ่นเต็ม (Full version) และโปรแกรมยังจำกัดข้อมูลนำเข้าบางส่วนด้วย ทั้งนี้ผู้วิจัยไม่มีนโยบายซื้อรุ่นเต็มมาใช้เพราะจะทำให้ค่าใช้จ่ายเพิ่มขึ้น

ถ้าพิจารณากรณีของทุ่งสง พื้นที่สำหรับกำหนดจุดตักอยู่ที่ระดับชั้นความสูงต่าง ๆ 5 ชั้น ได้แก่ 203, 221, 239, 257 และ 275 ระยะทางจากจุดตักถึงจุเทที่ระดับชั้นต่างกันจะแตกต่างกันไม่มาก ดังแสดงในรูปที่ 4.1 โดยการกำหนดจุดตักที่ระดับชั้นไม่ซ้ำกัน จำนวนรถบรรทุกที่เหมาะสมแต่ละจุดตักใน 1 กะ นั้น มีค่ามากที่สุดและน้อยที่สุดเท่ากับ 6 และ 4 คัน ตามลำดับ โดยคิดเฉพาะกรณีจุดตักมากที่สุดคือ 3 จุด เพราะจะได้รู้ว่าจะต้องใช้รถบรรทุกมากที่สุดกี่คัน และจำนวนรถบรรทุกรวมทั้งหมดที่ประเมินได้เท่ากับ 15 คัน ตามการประเมินผลด้วย Arena กรณีที่ 1, 2, 5, 8 และ 9 ดังตารางที่ 4.3 ทั้งนี้ยังไม่รวมรถสำรอง ข้อมูลในตารางเป็นข้อมูลที่ดีที่สุดจากการประเมินซ้ำ ๆ โดยมีเงื่อนไขว่า จำนวนรถบรรทุกที่มารอ ณ เครื่องจักรให้บริการใน 1 เที่ยว ต้องไม่เกิน 1 คัน เพื่อให้เวลารอของรถบรรทุกน้อยที่สุดในทุกกรณี นั่นหมายความว่าสัดส่วนการทำงานของรถบรรทุกไกล้เคียง 100 เปอร์เซ็นต์ เวลารอที่เห็นชัดเจนจะมีที่เครื่องจักรให้บริการคือรถตักและเครื่องย่อยเท่านั้น โดยที่สัดส่วนการทำงานของรถตักต้องไม่ต่ำกว่า 70 เปอร์เซ็นต์ กล่าวคือ ถ้าสัดส่วนการทำงานของรถตักเท่ากับ 100 เปอร์เซ็นต์ แสดงว่ารถตักไม่มีเวลารอในการทำงานนั่นเอง เวลารอของรถบรรทุกที่รถตักนั้นจะพิจารณารถตัก 3 คัน ในเวลาเดียวกัน และให้ระดับชั้นความสูงต่ำสุดเป็นรถตักปุ้งกี๋เสย EX1800 เนื่องจากเคลื่อนย้ายลำบาก ที่เหลืออีก 2 ชั้น เป็นรถตักล้อยาง CAT992 เวลารอของรถบรรทุกเฉลี่ยที่รถตักมีค่ามากที่สุดในกรณีที่ 1 ที่รถตักล้อยาง CAT992 เท่ากับ 3.52 นาทีต่อรอบ น้อยที่สุดกรณีที่ 10 ที่รถตักล้อยาง CAT992 เท่ากับ 1.07 นาทีต่อรอบ เวลารอของรถบรรทุกเฉลี่ยที่เครื่องย่อยมีค่ามากที่สุดในกรณีที่ 8 ที่เครื่องย่อย CR6 เท่ากับ 3.09 นาทีต่อรอบ น้อยที่สุดกรณีที่ 2 ที่เครื่องย่อย CR6 เท่ากับ 1.24 นาทีต่อรอบ ถ้าพิจารณาสัดส่วนการทำงานของรถตักจะพบว่า มีค่ามากที่สุดเท่ากับ 0.97 สำหรับรถตักล้อยาง CAT992 ในกรณี

ที่ 1 ที่ชั้น 221 และรถตักปุ้งกี๋เสย EX1800 ในกรณีที่ 5 ที่ชั้น 203 ส่วนค่าที่น้อยที่สุดเท่ากับ 0.72 สำหรับรถตักล้อยาง CAT992 ในกรณีที่ 6 ที่ชั้น 257

กรณีของท่าหลวง จำนวนจุดตักมี 2 จุด ตามการทำงานปกติของเหมือง พื้นที่สำหรับกำหนดจุดตักอยู่ที่ภูเขา A และภูเขา AB ซึ่งเรียกแทนให้ง่ายว่า เขา A และเขา AB ตามลำดับ ดังแสดงในรูปที่ 4.2 ซึ่งเขาทั้งสองอยู่ติดกัน เขา A กับจุดเทระยะทางจะใกล้ว่าเขา AB กับจุดเท ใน 1 กะ จะมีการย่อยแค่เครื่องย่อยเดียวเท่านั้น จำนวนรถบรรทุกที่เหมาะสมแต่ละจุดตักใน 1 กะ นั้น มีค่ามากที่สุดและน้อยที่สุดเท่ากับ 8 คัน และ 6 คัน ตามลำดับ โดยขึ้นอยู่กับว่าในกะดังกล่าวย่อยที่เครื่องย่อยใดด้วย ดังตารางที่ 4.7 เวลารอของรถบรรทุกเฉลี่ยที่รถตักมีค่ามากที่สุดในกรณีที่ 8 ที่รถตักปุ้งกี๋เสย EX1100 จุดตักเขา AB เท่ากับ 1.59 นาทีต่อรอบ น้อยที่สุดกรณีที่ 3 และ 4 ที่รถตักล้อยาง CAT992 จุดตักเขา A เท่ากับ 0.00 นาทีต่อรอบ เวลารอของรถบรรทุกเฉลี่ยที่เครื่องย่อยมีค่ามากที่สุดในกรณีที่ 1 ที่เครื่องย่อย TL เท่ากับ 3.93 นาทีต่อรอบ น้อยที่สุดกรณีที่ 16 ที่เครื่องย่อย KW เท่ากับ 1.00 นาทีต่อรอบ ถ้าพิจารณาสัดส่วนการทำงานของรถตักจะพบว่า มีค่ามากที่สุดเท่ากับ 0.59 สำหรับรถตักล้อยาง CAT992 ในกรณีที่ 10 ที่จุดตักเขา A ส่วนค่าที่น้อยที่สุดเท่ากับ 0.21 สำหรับรถตักล้อยาง CAT992 ในกรณีที่ 3 ที่จุดตักเขา A

รูปที่ 4.1 ภาพถ่ายทางอากาศเหมืองหินปูนบริษัท ปูนซิเมนต์ไทย (ทุ่งสง) จำกัด (earth.google.com)

รูปที่ 4.1 เป็นภาพถ่ายทางอากาศของเหมืองหินปูนทุ่งสง หมุดสีเหลืองคือจุดเทซึ่งประกอบด้วยเครื่องย่อย 2 เครื่องคือ CR5 และ CR6 จุดตักมีอยู่ด้วยกันทั้งหมด 5 ชั้น คือ ชั้น 203, 221, 239, 257 และ 275 ในความเป็นจริงระยะทางระหว่างจุดตักและจุดเทมีการเปลี่ยนแปลงไปตลอดเพราะต้องเปลี่ยนจุดตักไปเรื่อย ๆ ดังนั้นผลการวิจัยที่ได้จึงเป็นจริงเฉพาะช่วงเวลาที่ผู้วิจัยทำการเก็บข้อมูลจากภาคสนามเท่านั้น

รูปที่ 4.2 เป็นภาพถ่ายทางอากาศของเหมืองหินปูนท่าหลวง หมุดสีเหลืองคือจุดเทซึ่งประกอบด้วยเครื่องย่อย 2 เครื่องคือ TL และ KW จุดตักมีอยู่ด้วยกันทั้งหมด 2 จุดหลัก คือ เขา AB และ เขา A

รูปที่ 4.2 ภาพถ่ายทางอากาศเหมือหินปูนบริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด (earth.google.com)

ผลการวิจัยในส่วนของโปรแกรมภาษาจาวานั้น สำหรับทุ่งสงข้อมูลนำเข้าหรือข้อมูลที่ผู้ใช้ต้องระบุให้กับโปรแกรมได้แก่ จำนวนจุดตัก, ชื่อรถบรรทุก, จำนวนรถบรรทุก, ความจุดัมพ์รถบรรทุก, ความเร็วเฉลี่ยบรรทุกหนัก, ความเร็วเฉลี่ยรถเปล่า, เวลาเทของรถบรรทุก, ชื่อรถตัก ณ จุดตักที่ 1, เวลาตักของรถตัก, ชื่อรถตัก ณ จุดตักที่ 2, เวลาตักของรถตัก, ชื่อรถตัก ณ จุดตักที่ 3, เวลาตักของรถตัก,

จำนวนเครื่องย่อย, ชื่อเครื่องย่อยที่ 1, กำลังผลิตของเครื่องย่อย 1, ชื่อเครื่องย่อยที่ 2, กำลังผลิตของเครื่องย่อย 2 และผู้ใช้ต้องระบุว่าต้องการดูผลการทำงานที่จุดตักหรือจุดเท รายละเอียดตามตารางที่ 4.1

ตารางที่ 4.1 ข้อมูลนำเข้าสำหรับการประมวลผลด้วยโปรแกรมจาวาของทุ่งสง

ลำดับที่	รายการข้อมูล	หน่วย	ค่าข้อมูล
1	จำนวนจุดตัก	จุด	3
2	ชื่อรถบรรทุก	-	CAT777
3	จำนวนรถบรรทุก	คัน	12
4	ความจุดัมพ์รถบรรทุก	เมตริกตัน	85
5	ความเร็วเฉลี่ยบรรทุกหนัก	กิโลเมตร/ชั่วโมง	21
6	ความเร็วเฉลี่ยรถเปล่า	กิโลเมตร/ชั่วโมง	25
7	เวลาเทของรถบรรทุก	นาที	NORM(1, 0.2)
8	ชื่อรถตัก ณ จุดตักที่ 1	-	1800
9	เวลาตักของรถตัก	นาที	NORM(3.8, 0.2)
10	ชื่อรถตัก ณ จุดตักที่ 2	-	992#1
11	เวลาตักของรถตัก	นาที	NORM(4, 1)
12	ชื่อรถตัก ณ จุดตักที่ 3	-	992#1
13	เวลาตักของรถตัก	นาที	NORM(4, 1)
14	จำนวนเครื่องย่อย	เครื่อง	2
15	ชื่อเครื่องย่อยที่ 1	-	CR5
16	กำลังผลิตของเครื่องย่อย 1	เมตริกตัน/ชั่วโมง	1,300

ลำดับที่	รายการข้อมูล	หน่วย	ค่าข้อมูล
17	ชื่อเครื่องย่อยที่ 2	-	CR6
18	กำลังผลิตของเครื่องย่อย 2	เมตริกตัน/ชั่วโมง	1,700
19	ระยะทางระหว่างจุดเทกับจุดตักที่ 1	เมตร	1,000
20	ระยะทางระหว่างจุดเทกับจุดตักที่ 2	เมตร	1,100
21	ระยะทางระหว่างจุดเทกับจุดตักที่ 3	เมตร	1,200
22	ดูผลการทำงานที่ไหน	-	จุดตัก

ตามรูปที่ 3.2 ในบทที่ 3 โปรแกรมเริ่มทำงาน ณ เวลาปัจจุบัน (Current time) คือ 14.16 น. และโปรแกรมจะเริ่มทำงานที่เวลานี้ โดยปล่อยรถบรรทุกคันที่ 1, 2 และ 3 คือ CAT777#1, #2 และ #3 ตามลำดับ ออกจากจุดเท (Depart dumping area) เพราะจุดตักมีจำนวน 3 จุด จึงปล่อยรถบรรทุกพร้อมกัน 3 คัน เริ่มตักที่รถตักปุ้งกี๋เสีย EX1800, รถตักล้อยาง CAT992#1 และ CAT992#2 เวลา 14.18 น. ทั้ง 3 คัน (1800 start load, 992#1 start load และ 992#2 start load) ตักเสร็จเวลา 14.22 น., 14.21 น. และ 14.21 น. (1800 loaded, 992#1 loaded และ 992#2 loaded) เริ่มเทที่ CR5 เวลา 14.24 น. (Start dump at CR5) เทเสร็จเวลา 14.25 น. (Dumped at CR5) ทั้ง 3 คัน รถบรรทุกแต่ละคันถึงรถตักที่จุดตักแต่ละจุดอีกครั้ง เวลา 14.27 น. (Arrive 1800 again, Arrive 992#1 again, Arrive 992#2 again) พร้อมกัน ถึงเวลานี้โปรแกรมจะเริ่มปล่อยรถคันที่ 4, 5 และ 6 จากจุดเทไปยังจุดตักที่ 1, 2 และ 3 ตามลำดับ แต่ก่อนปล่อยโปรแกรมจะตรวจสอบก่อนว่าเวลาที่คันที่ 1, 2 และ 3 ถึงจุดตักอีกครั้งมากกว่าเวลาที่จะปล่อยคันที่ 4, 5 และ 6 หรือไม่ ถ้าใช่จึงปล่อย ถ้าไม่ใช่โปรแกรมจะไม่ปล่อยรถบรรทุกอีกและจะนับจำนวนรถบรรทุกสำหรับแต่ละจุดตัก พร้อมทั้งแสดงออกทางหน้าจอว่าที่จุดตักไหนใช้รถบรรทุกคันใด จากรูป จุดตักที่ 1 ใช้รถบรรทุก 4 คัน ได้แก่ CAT777#1, #4, #7 และ #10 จุดตักที่ 2 ใช้รถบรรทุก 3 คัน ได้แก่ CAT777#2, #5 และ #8 และจุดตักที่ 3 ใช้รถบรรทุก 3 คัน ได้แก่ CAT777#3, #6 และ #9 หลังจากนี้ โปรแกรมจะมวลผลต่อไปจนครบ 8 ชั่วโมง โดยเริ่มนับตั้งแต่เวลาปัจจุบัน คือ 14.16 น. และระหว่างนี้ผู้ใช้จะเห็นผลทางหน้าจอถึงสถานะของรถบรรทุกแต่ละคันซึ่งแสดงเฉพาะที่จุดตักเท่านั้น เนื่องจากผู้ใช้เลือกดูการ

ประมวลผลที่จุดตักตามตารางที่ 4.1 ลำดับที่ 22 สิ่งที่ผู้ใช้ได้จากโปรแกรมนี้ก็คือ จำนวนรถบรรทุกที่เหมาะสมในแต่ละจุดตักที่ทำให้เวลารอของรถบรรทุกที่รถตักและเครื่องย่อยน้อยที่สุด

ส่วนผลการวิจัยโปรแกรมภาษาจาวาสำหรับทุ่งสง ข้อมูลนำเข้า เป็นไปตามตารางที่ 4.2

ตารางที่ 4.2 ข้อมูลนำเข้าสำหรับการประมวลผลด้วยโปรแกรมจาวาของท่าหลวง

ลำดับที่	รายการข้อมูล	หน่วย	ค่าข้อมูล
1	จำนวนจุดตัก	จุด	2
2	ชื่อรถบรรทุก	-	CAT777
3	จำนวนรถบรรทุก	คัน	12
4	ความจุดัมพ์รถบรรทุก	เมตริกตัน	85
5	ความเร็วเฉลี่ยบรรทุกหนัก	กิโลเมตร/ชั่วโมง	21
6	ความเร็วเฉลี่ยรถเปล่า	กิโลเมตร/ชั่วโมง	25
7	เวลาเทของรถบรรทุก	นาที	NORM(1, 0.2)
8	ชื่อรถตัก ณ จุดตักที่ 1	-	1800
9	เวลาตักของรถตัก	นาที	NORM(3.8, 0.2)
10	ชื่อรถตัก ณ จุดตักที่ 2	-	CAT992
11	เวลาตักของรถตัก	นาที	NORM(4, 0.2)
12	จำนวนเครื่องย่อย	เครื่อง	1
13	ชื่อเครื่องย่อยที่	-	TL
14	กำลังผลิตของเครื่องย่อย	เมตริกตัน/ชั่วโมง	1,800
15	ระยะทางระหว่างจุดเทกับจุดตักที่ 1	เมตร	2,000

ลำดับที่	รายการข้อมูล	หน่วย	ค่าข้อมูล
16	ระยะทางระหว่างจุดเทกับจุดตักที่ 2	เมตร	1,800
17	ดูผลการทำงานที่ไหน	-	จุดเท

ตามรูปที่ 3.3 ในบทที่ 3 โปรแกรมเริ่มทำงานที่เวลาปัจจุบัน คือ 13.55 น. และโปรแกรมจะเริ่มทำงานที่เวลานี้ โดยปล่อยรถบรรทุกคันที่ 1 และ 2 คือ CAT777#1 และ #2 ตามลำดับ ออกจากจุดเท (Depart dumping area) เพราะจุดตักมีจำนวน 2 จุด จึงปล่อยรถบรรทุกพร้อมกัน 2 คัน เริ่มตักที่รถตักปุ้งกี๋เสย EX1800 และ CAT992 เวลา 13.59 น. พร้อมกันทั้ง 2 คัน (EX1800 start load และ CAT992 start load) ตักเสร็จเวลา 14.02 น. และ 14.03 น. ตามลำดับ (EX1800 loaded และ CAT992 loaded) เริ่มเทที่ TL เวลา 14.07 น. และ 14.08 น. ตามลำดับ (start dump at TL) เทเสร็จเวลา 14.07 น. และ 14.09 น. ตามลำดับ (dumped at TL) รถบรรทุกแต่ละคันถึงรถตักที่จุดตักแต่ละจุดอีกครั้งเวลา 14.11 น. และ 14.13 น. ตามลำดับ (Arrive EX1800 again และ Arrive CAT992 again) ถึงเวลานี้โปรแกรมจะเริ่มปล่อยรถคันที่ 3 และ 4 จากจุดเทไปยังจุดตักที่ 1 และ 2 ตามลำดับ แต่ก่อนปล่อยโปรแกรมจะตรวจสอบก่อนว่าเวลาที่คันที่ 1 และ 2 ถึงจุดตักอีกครั้งมากกว่าเวลาที่จะปล่อยคันที่ 3 และ 4 หรือไม่ ถ้าใช่จึงปล่อย ถ้าไม่ใช่โปรแกรมจะไม่ปล่อยรถบรรทุกอีกและจะนับจำนวนรถบรรทุกสำหรับแต่ละจุดตัก พร้อมทั้งแสดงออกทางหน้าจอว่าที่จุดตักไหนใช้รถบรรทุกคันใด จากรูป จุดตักที่ 1 ใช้รถบรรทุก 4 คัน ได้แก่ CAT777#1, #3, #5 และ #7 จุดตักที่ 2 ใช้รถบรรทุก 4 คัน ได้แก่ CAT777#2, #4, #6 และ #8 หลังจากนี้ โปรแกรมจะมวลผลต่อไปจนครบ 8 ชั่วโมง โดยเริ่มนับตั้งแต่เวลาปัจจุบัน คือ 13.55 น. และระหว่างนี้ผู้ใช้จะเห็นผลทางหน้าจอถึงสถานะของรถบรรทุกแต่ละคันซึ่งแสดงเฉพาะที่จุดเทเท่านั้น เนื่องจากผู้ใช้เลือกดูการประมวลผลที่จุดเทตามตารางที่ 4.2 ลำดับที่ 17

บทที่ 5
สรุปผลการวิจัย

เครื่องมือที่ผู้วิจัยสร้างขึ้นมาทั้ง 2 ชิ้น ได้แก่ โปรแกรม Arena และโปรแกรมภาษาจาวานั้น สามารถนำไปใช้ประมวลผลระบบขนส่งหินจากหน้าเหมืองมายังจุดเท โดยเน้นที่การใช้รถบรรทุกในการขนถ่ายหินหรือวัตถุดิบโปรแกรม Arena ให้ข้อมูลสำคัญกับผู้ใช้ ได้แก่

1. จำนวนรถบรรทุกที่เหมาะสมในแต่ละจุดตัก โดยเวลารอของรถบรรทุกที่รถตักและเครื่อง ย่อยน้อยที่สุด
2. เวลารอเฉลี่ยของรถบรรทุกที่รถตักและเครื่องย่อย และ
3. สัดส่วนการทำงานของรถตักที่มากที่สุดที่ทำให้เวลารอของรถบรรทุกน้อยที่สุด ทั้งที่จุดตักและจุดเท

ส่วนโปรแกรมภาษาจาวาให้ข้อมูลกับผู้ใช้ ได้แก่

1. จำนวนรถบรรทุกที่เหมาะสมในแต่ละจุดตัก
2. สถานะของรถบรรทุก ณ เวลาต่าง ๆ
3. ผู้ใช้สามารถพยากรณ์การทำงานล่วงหน้าของระบบได้ เนื่องจากโปรแกรมอ้างอิงเวลาท้องถิ่น

ทั้งโปรแกรม Arena และโปรแกรมภาษาจาวามีส่วนที่ยังไม่สมบูรณ์ เนื่องมาจากการพัฒนางานวิจัยโดยใช้ซอฟต์แวร์และการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ต้องอาศัยทักษะและระยะเวลาที่ค่อนข้างนาน โปรแกรมทั้ง 2 ยังต้องการการพัฒนาต่อ ผู้ที่สนใจและมีความสามารถที่จะพัฒนาโปรแกรมให้ดีขึ้นในอีกระดับ สามารถนำแบบจำลองโปรแกรม Arena และรหัสต้นฉบับจาวาไปปรับปรุงเปลี่ยนแปลงให้เหมาะสมกับลักษณะงานได้ตามต้องการ

ในการดำเนินการวิจัยที่เกี่ยวกับการสร้างโมเดลหรือแบบจำลองใด ๆ ก็ตาม แบบจำลองจะไม่มีประโยชน์เลยถ้าข้อมูลที่นำมาสร้างเชื่อถือไม่ได้ ขั้นตอนที่สำคัญที่สุดในการสร้างแบบจำลองโดยทั่วไปมีอยู่ 2 ขั้นตอน คือ ขั้นตอนการเก็บข้อมูลจากภาคสนามและขั้นตอนการเขียนโปรแกรมหรือประมวลผลด้วยซอฟต์แวร์ ในการเก็บข้อมูลภาคสนามผู้วิจัยต้องมีเวลาสำหรับทำความเข้าใจระบบงานก่อน

ระยะหนึ่ง ซึ่งจะทำให้สามารถออกแบบเครื่องมือในการเก็บข้อมูลได้แม่นยำและรวดเร็วยิ่งขึ้น เครื่องมือในการเก็บข้อมูลดังกล่าว เช่น ตารางบันทึกเวลา, กล้องบันทึกวิดีโอ เป็นต้น ควรต้องใช้เวลาในขั้นตอนนี้อย่างน้อย 1 วัน หรือขึ้นอยู่กับความเหมาะสม เสร็จจากขั้นตอนนี้แล้วจึงค่อยลงมือเก็บข้อมูลจริง โดยผู้วิจัยต้องระลึกไว้เสมอว่าในการเก็บข้อมูลนั้นผู้วิจัยต้องไม่เข้าไปเป็นส่วนใดส่วนหนึ่งของระบบ ต้องปล่อยให้ระบบเดินไปตามปกติ เพราะไม่เช่นนั้นแล้วข้อมูลที่ได้มาก็จะไม่ใช่ข้อมูลที่แท้จริงเพราะไปใส่ตัวแปรเพิ่มให้กับระบบซึ่งก็คือตัวผู้วิจัยเอง ผลวิจัยออกมาก็ไม่สามารถนำไปใช้หรือพัฒนาต่อได้อีกเลย เพราะทำผิดตั้งแต่ขั้นตอนการเก็บข้อมูลแล้ว อีกสิ่งที่สำคัญไม่แพ้กันก็คือ ผู้วิจัยต้องเก็บข้อมูลด้วยตัวเอง ต้องลงภาคสนามด้วยตัวเอง จริงอยู่ที่ข้อมูลอาจจะมีอยู่แล้วเนื่องจากถ้าไปเก็บข้อมูลจากทางองค์กร อาจมีข้อมูลที่ถูกเก็บไว้เป็นประจำทุกวัน แต่ด้วยวัตถุประสงค์ของการสร้างแบบจำลองอาจจะเป็นข้อมูลที่แตกต่างกัน กล่าวคือ พนักงานในองค์กรเก็บข้อมูลเพื่อหวังสร้างผลกำไรให้กับองค์กรแต่เพียงอย่างเดียว แต่ผู้วิจัยเก็บข้อมูลเพื่อสร้างเครื่องมือขึ้นมาใหม่หรือพัฒนาเครื่องมือเดิมให้ดียิ่งขึ้น ดังนั้นข้อมูลที่องค์กรเก็บอาจจะไม่ใช่ที่ผู้วิจัยต้องการ

สิ่งที่ทุกฝ่ายคาดหวังการการสร้างแบบจำลอง ทั้งฝ่ายผู้วิจัยเองและฝ่ายผู้ใช้ ก็คือ มันต้องเป็นเครื่องมือที่มีความสามารถในการพยากรณ์เหตุการณ์ได้ในทุกสถานการณ์ ไม่ว่าจะมีเงื่อนไขหรือตัวแปรใด ๆ ก็ตามก็ยังยังสามารถใช้แบบจำลองนี้ได้ งานวิจัย 2 ชิ้น ที่ผู้วิจัยสร้างขึ้นนี้ สามารถใช้พยากรณ์ระบบได้ดีในระดับหนึ่ง โดยผู้ใช้ต้องมีข้อมูลสำหรับป้อนสู่ระบบตามเงื่อนไขที่ผู้ใช้ต้องการและตามขอบเขตของโปรแกรมทั้ง 2 ดังได้กล่าวมาแล้วในบทก่อนหน้า ทั้งนี้มีตัวแปรสำคัญบางตัวที่ผู้วิจัยไม่ได้ใส่เข้าไปในตัวโปรแกรม ได้แก่ สภาพอากาศ, ความไม่แน่นอนเนื่องจากเครื่องจักรเสีย, ความพร้อมของผู้บังคับเครื่องจักร เป็นต้น

ในโอกาสต่อไปผู้วิจัยยังมีความหวังว่า จะได้พัฒนาโปรแกรมได้สมบูรณ์ยิ่งขึ้น พร้อมทั้งเพิ่มตัวแปรเข้าไปเพื่อความสมบูรณ์ ให้เป็นเครื่องมือที่องค์กรนำไปใช้แล้วสร้างผลกำไรให้แก่องค์กรอย่างแท้จริง

เอกสารอ้างอิง

[1] ธนิศา เครือไวศยวรรณ. 2548. ความรู้เบื้องต้นภาษาจาวา. กรุงเทพมหานคร.

[2] กรัณย์ กันพ้นภัย. 2547. การจัดรูปแบบการขนส่งหินปูนในกระบวนการผลิตปูนซีเมนต์โดยใช้การโปรแกรมเชิงเส้นตรงและระบบแถวคอย. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต, สาขาวิศวกรรมการจัดการอุตสาหกรรม คณะวิศวกรรมศาสรต์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

[3] สมชาย วาณิชวศิน. 2540. การวิเคราะห์ปัจจัยที่มีผลต่อการเพิ่มอัตราการขนส่งปูนซีเมนต์ผงทางเรือ ด้วยแบบจำลองปัญหา. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต, สาขาวิศวกรรมการจัดการอุตสาหกรรม คณะวิศวกรรมศาตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

[4] นิพนธ์ วงษ์พานิช. 2525. การศึกษาการประยุกต์รูปแบบการขนส่ง. การค้นคว้าแบบอิสระเชิงวิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต, สาขาการสอนคณิตศาสตร์ คณะวิทยาศาตร์ มหาวิทยาลัยเชียงใหม่

[5] Raymond L Lowrie. 2002. Mines and Mineral Resources Handbooks Manuals. 2nd printing. Colorado, United States of America : Society for Mining, Metallurgy and Exploration, Inc. (SME)

[6] Andre James Krause. 2006. Shovel-truck Cycle Simulation Methods in Surface Mining. The degree of Masters of the Science in Engineering, Faculty of Engineering and Buit Environment University of The Witwatersrand.

[7] Fioroni, M.M., Franzese, L.A.G, Bianchi, T.J., Ezawa, L., Pinto, L.R. and Miranda, G., 2008. Concurrent Simulation and Optimization Models for Mining Planning. Proceedings of the 2008 Winter Simulation Conference 6, no.8(December): 759-767.

[8] Qing-hua, G., Cai-wu, L., Fa-ben, L. and Chang-yong, W., 2008. Monitoring Dispatch Information System of Trucks and Shovels in An Open Pit Based on GIS/GPS/GPRS. Journal of China University of Mining and Technology 18, no.2(June): 288-292.

[9] Krause, A. and Musingwini C., 2007. Modeling Open Pit Shovel-truck Systems Using the Machine Repair Model. The Journal of The Southern African Institute of Mining and Metallurgy 107, no.8(August): 469-476.

[10] Kang, J.H., Ahn, S.M. and Nam, J.H., 2006. Productivity Assessment of Rock Transportation Trucks Using Simulation Technology. ISARC 23, 592-595.

[11] Cetin, N., Erarslan, I. C. and Okuducu, A., 2001. Computer Simulation of Truck/Shovel System at Tuncbilek Coal Mine Using GPSS/H. International Mining Congress and Exhibition of Turkey 17, 715-718.

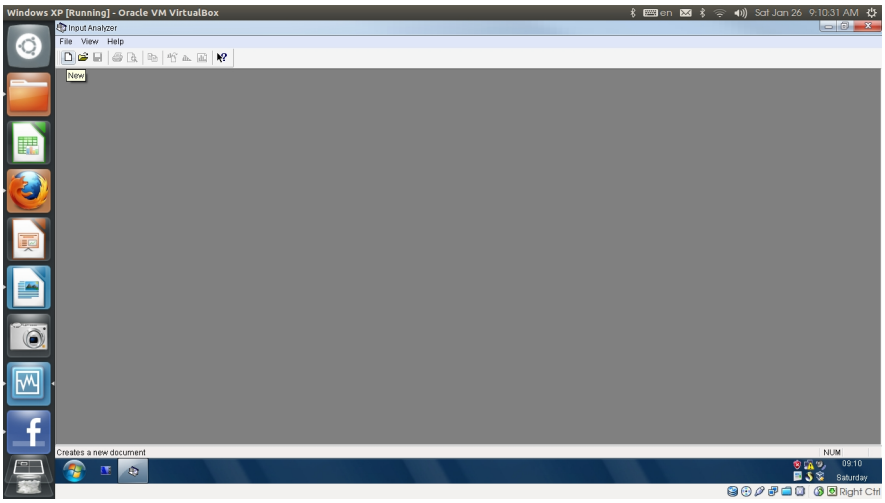
[12] ถาวร ศรีเสนพิลา. 2556. VDO Training, Java. http://www.javathailand.com

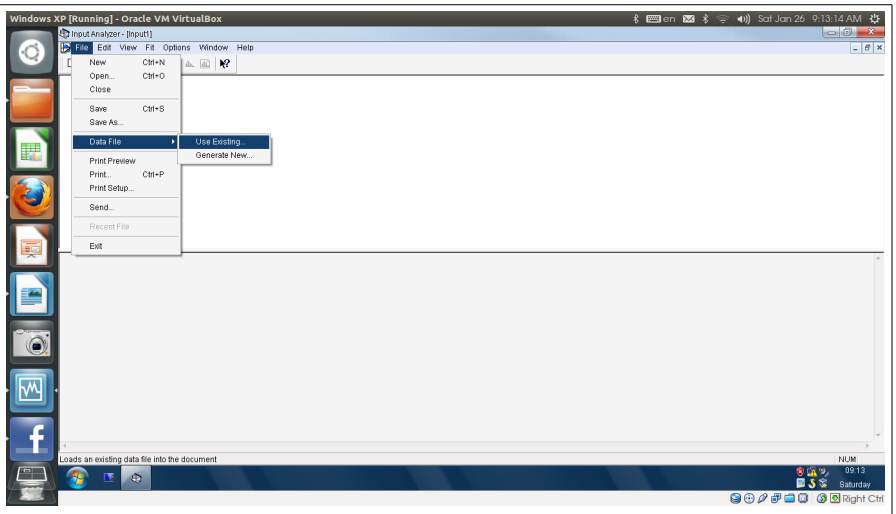
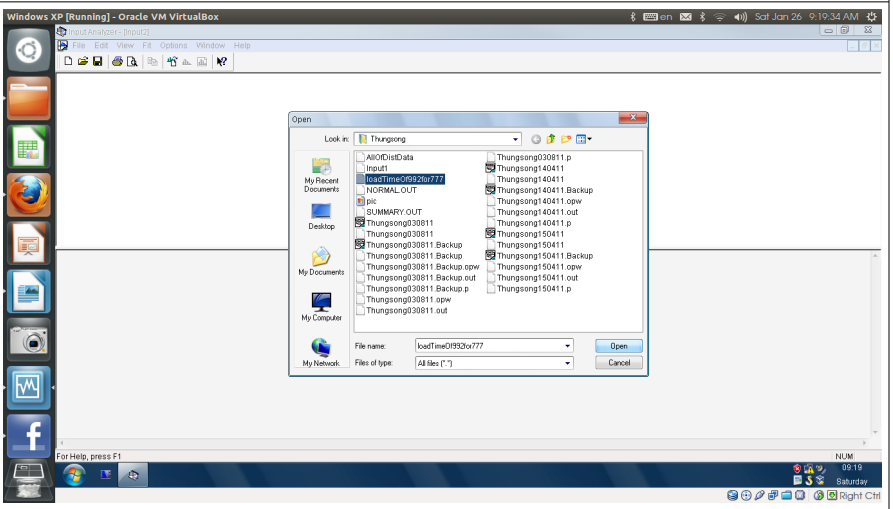
[13] Mark Shuttleworth. 2013. Download, Desktop. Canonical Ltd. http://www.ubuntu.com/download/desktop

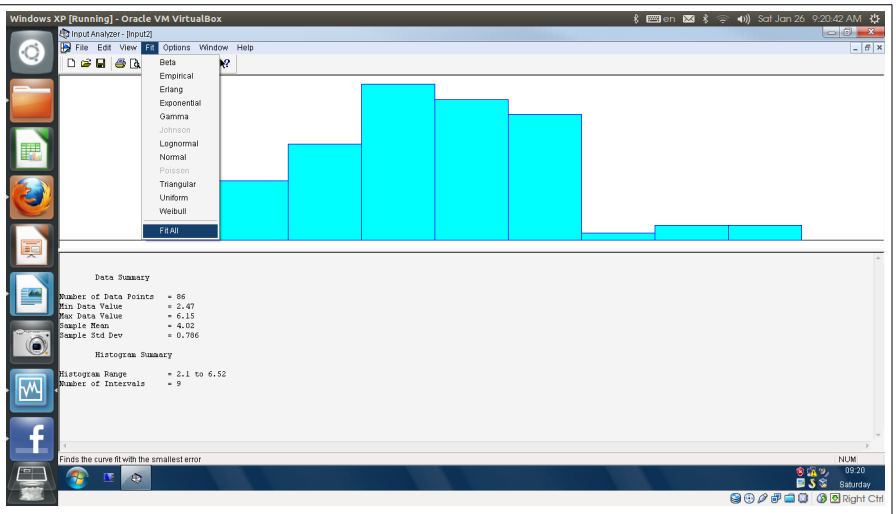
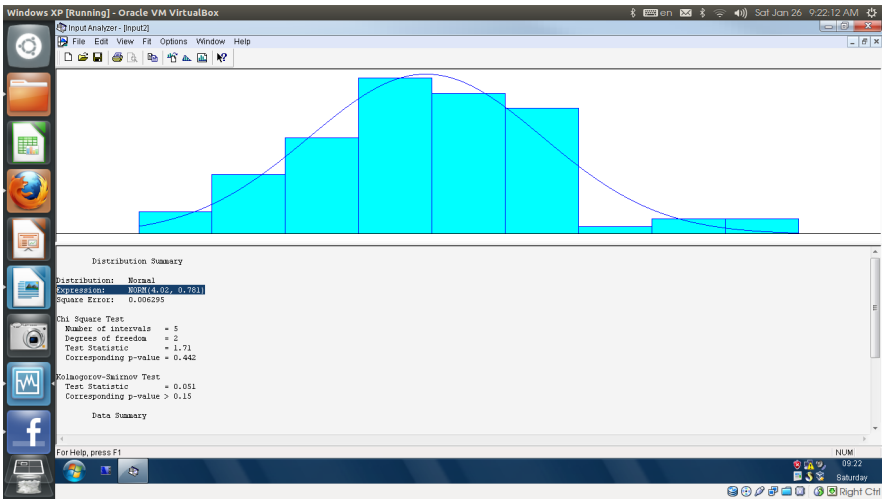
[14] Google, Inc. http://www.google.com/earth/download/ge/agree.html

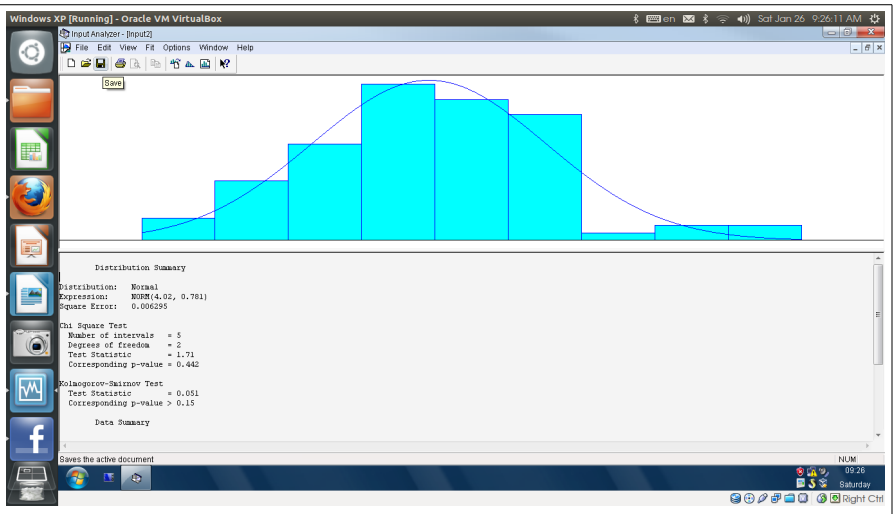
ภาคผนวก ก
โปรแกรม Arena

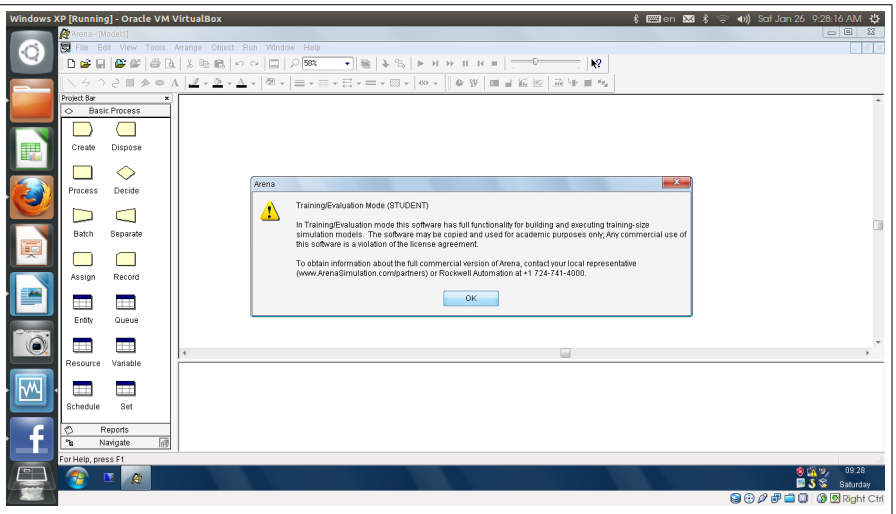
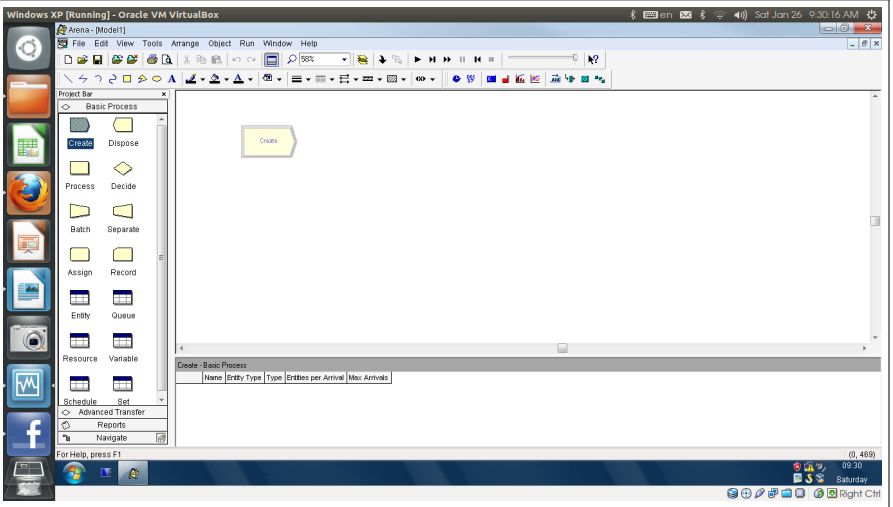
ขั้นตอนการสร้างแบบจำลองด้วยโปรแกรม Arena
สำหรับ บริษัท ปูนซิเมนต์ไทย (ทุ่งสง) จำกัด

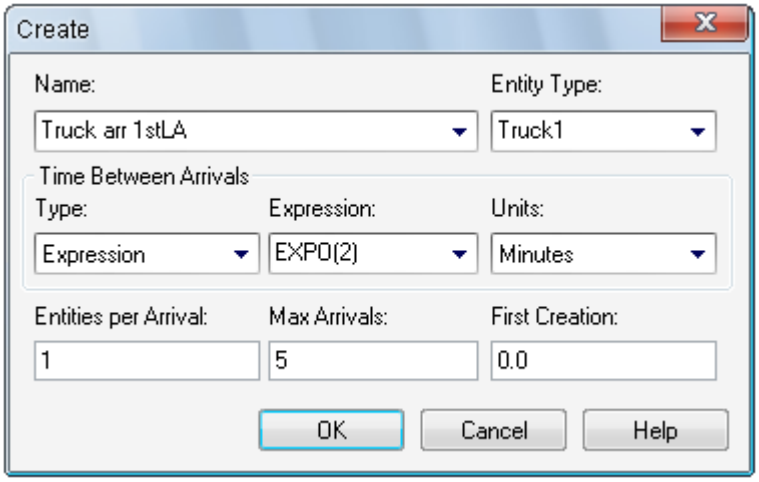
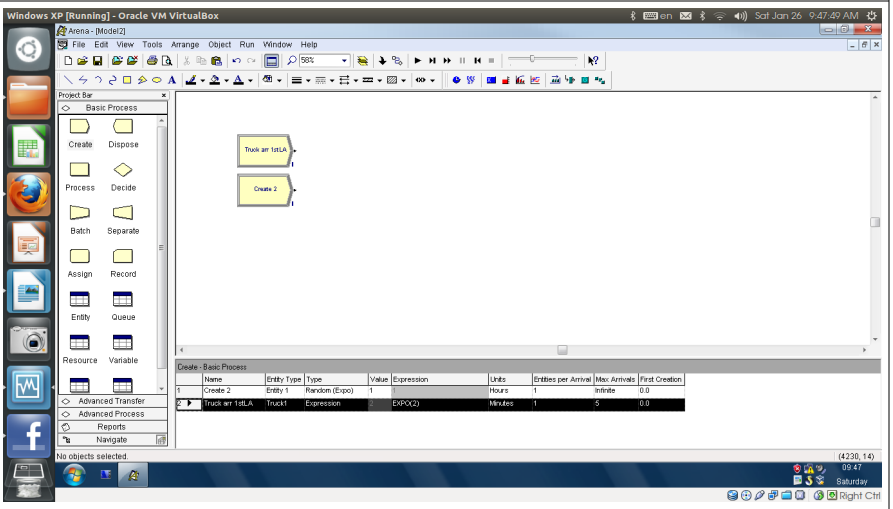
ลำดับขั้นตอน	ภาพประกอบ
1. เปิดโปรแกรม Input Analyzer	
2. เลือก New	

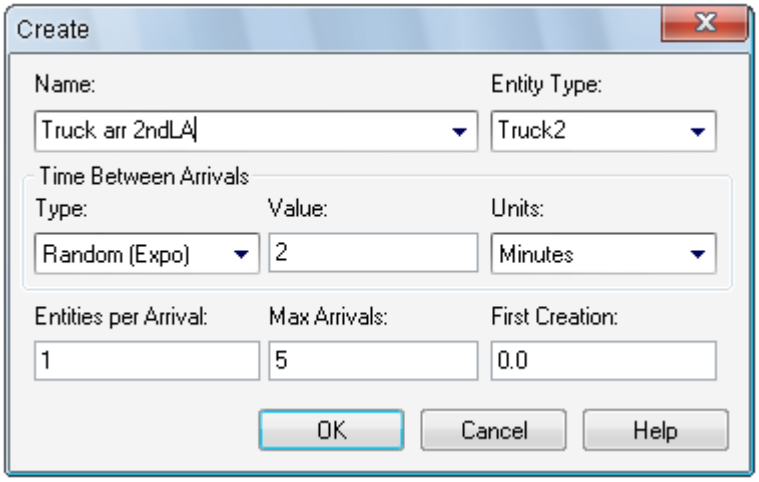
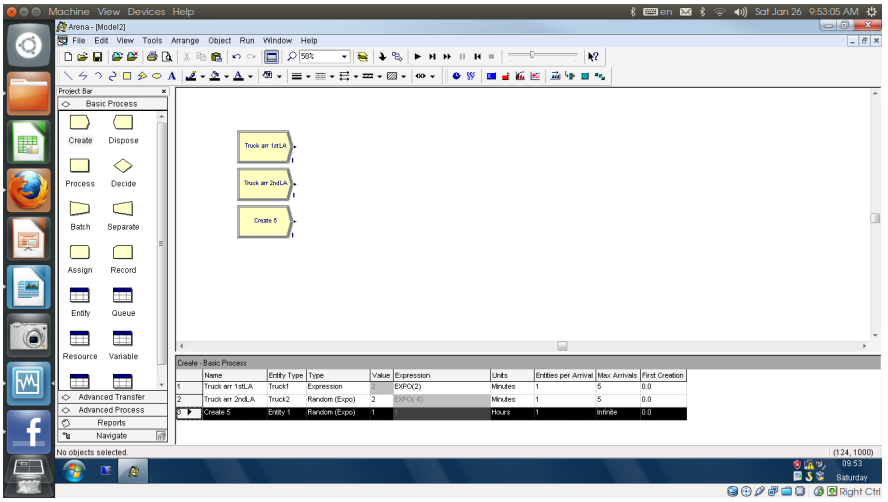
3. เลือก Use Exiting...	
4. เลือก file ข้อมูลนามสกุล .txt	

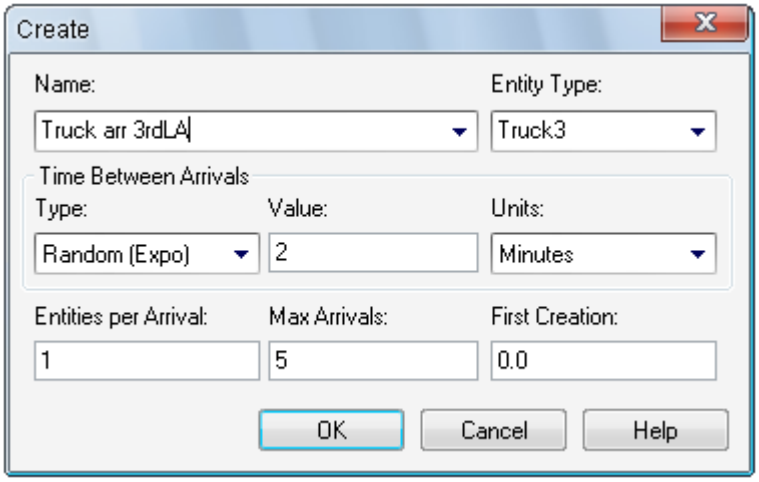
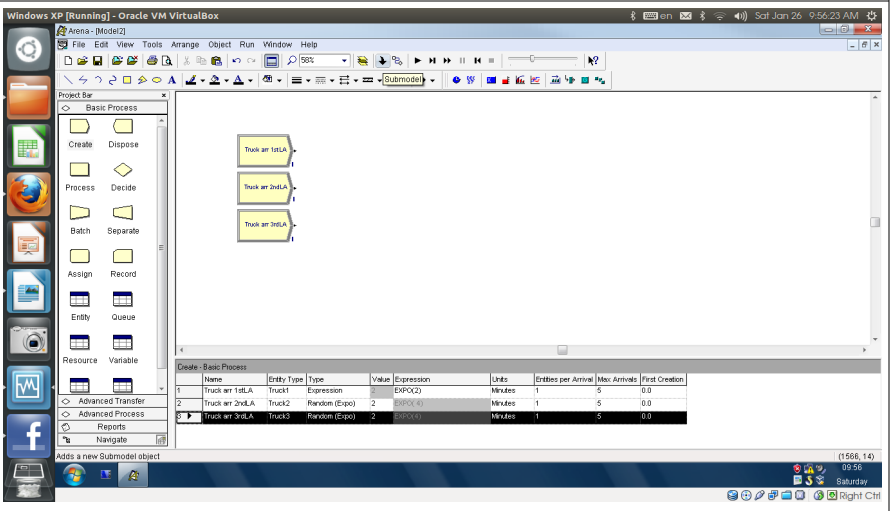
5. เลือก Fit All	
6. Distribution Summary ตรง Expression: คือข้อมูลที่ต้องนำไปใช้ในการสร้างแบบจำลอง ในที่นี้คือ NORM(4.02, 0.781)	

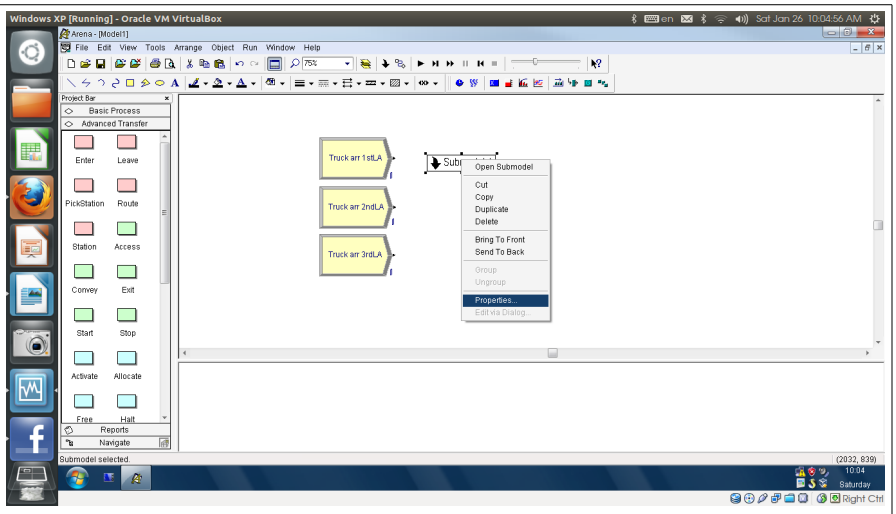
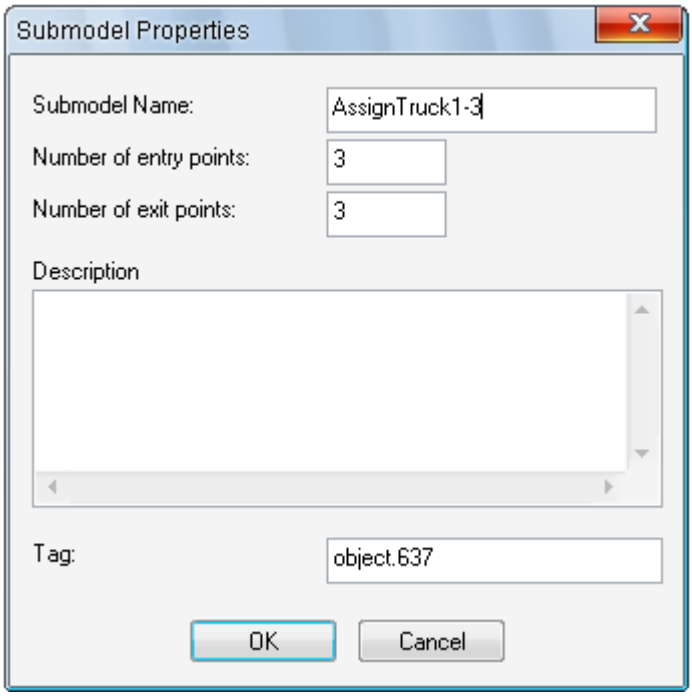
7. เลือก Save	
8. เปิดโปรแกรม Arena	

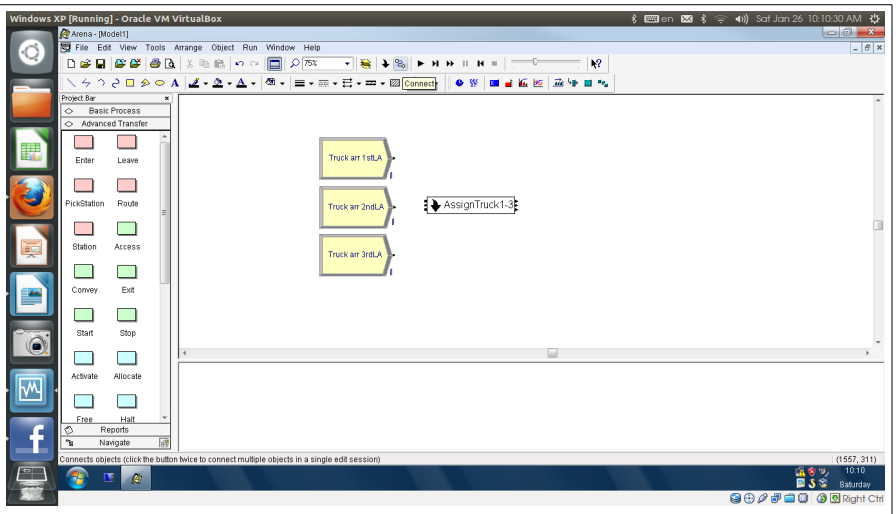
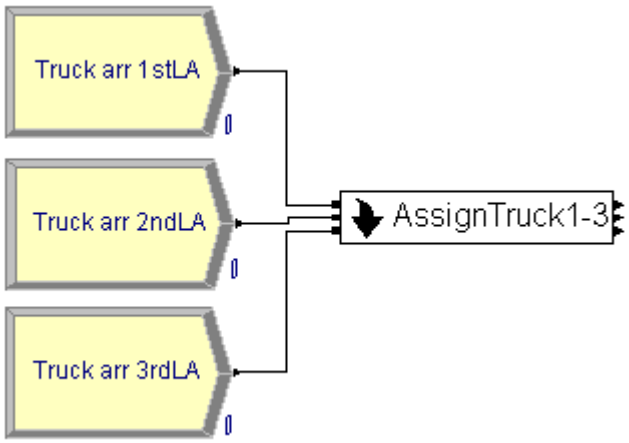
9. เลือก OK	
10. สร้าง Create Module ของรถบรรทุกสำหรับจุดตักที่ 1 โดยลาก Create Module ทางซ้ายมือเข้ามาใน Working area	

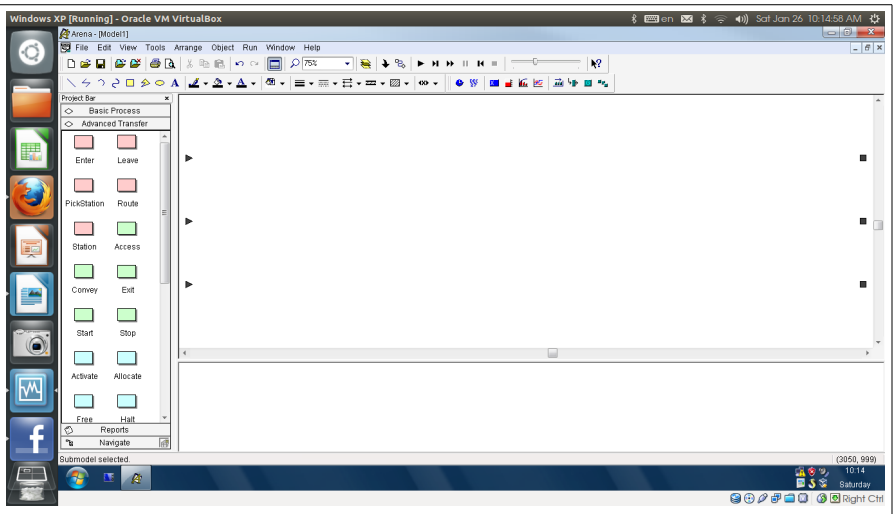
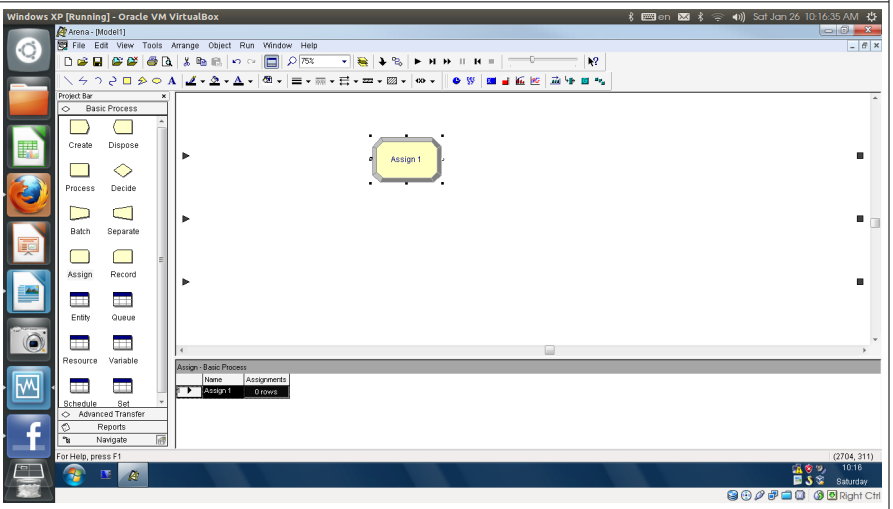
11. Double click แล้วใส่รายละเอียดตามรูป **ข้อมูลในช่อง Expression ได้มาจากการประเมินตามขั้นตอนที่ 6.**	
12. สร้าง Create Module ของรถบรรทุกสำหรับจุดตักที่ 2	

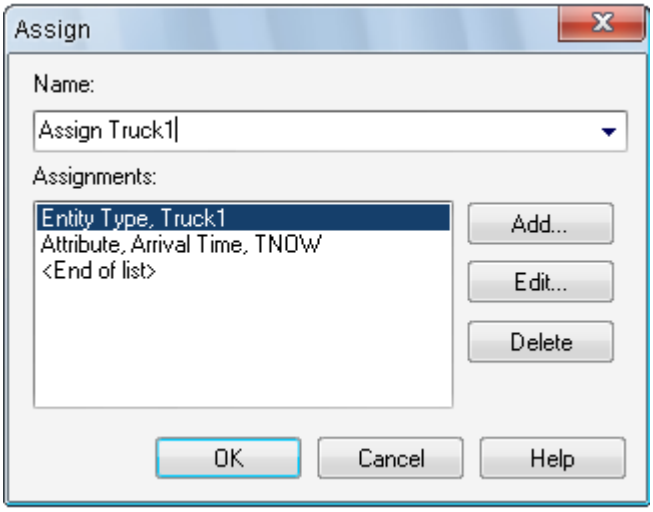
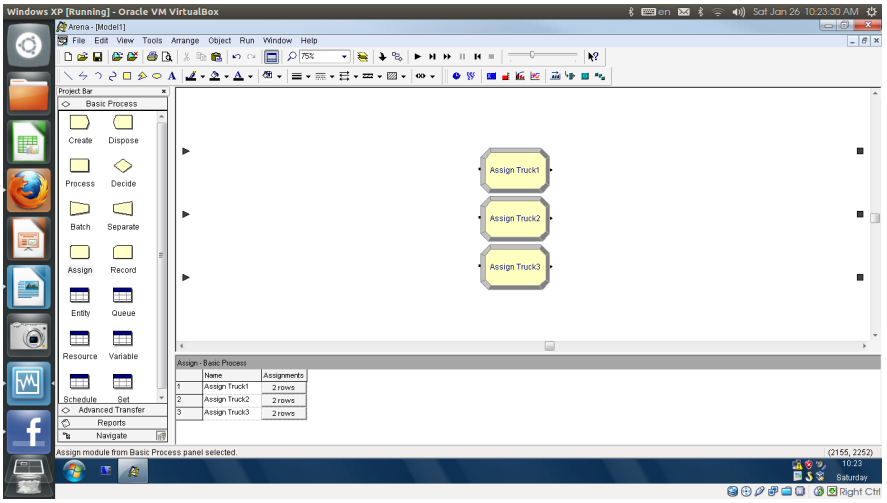
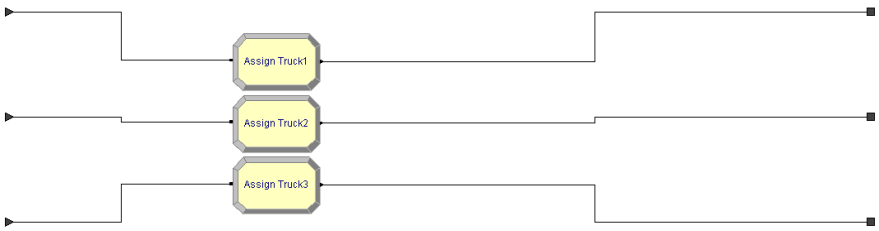
13. ใส่รายละเอียดตามรูป	
14. สร้าง Create Module ของรถบรรทุกสำหรับจุดตักที่ 3	

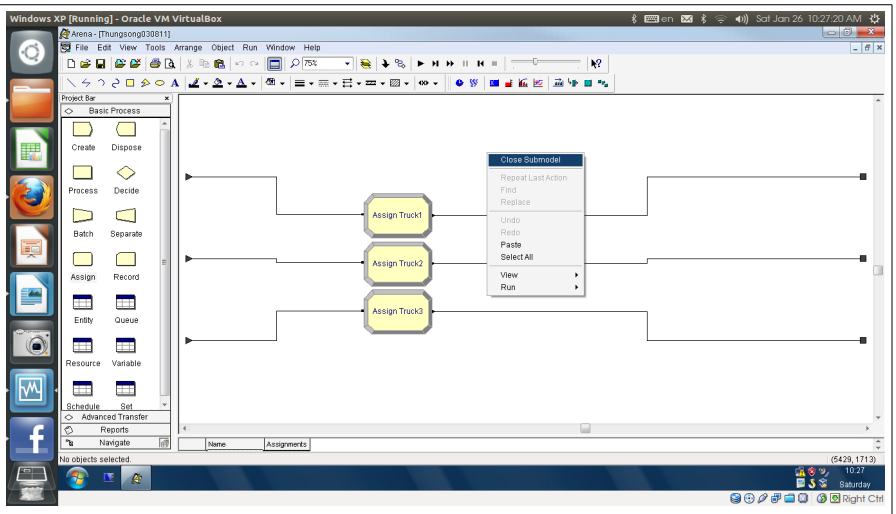
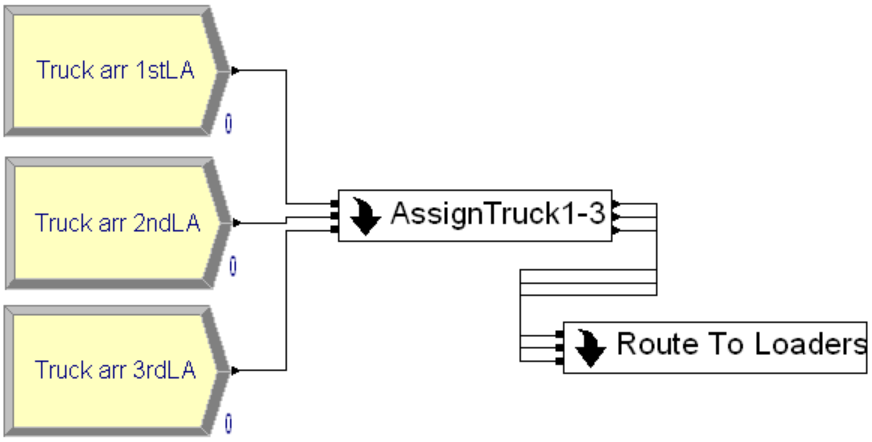
15. ใส่รายละเอียดตามรูป	
16. เลือก Submodel	

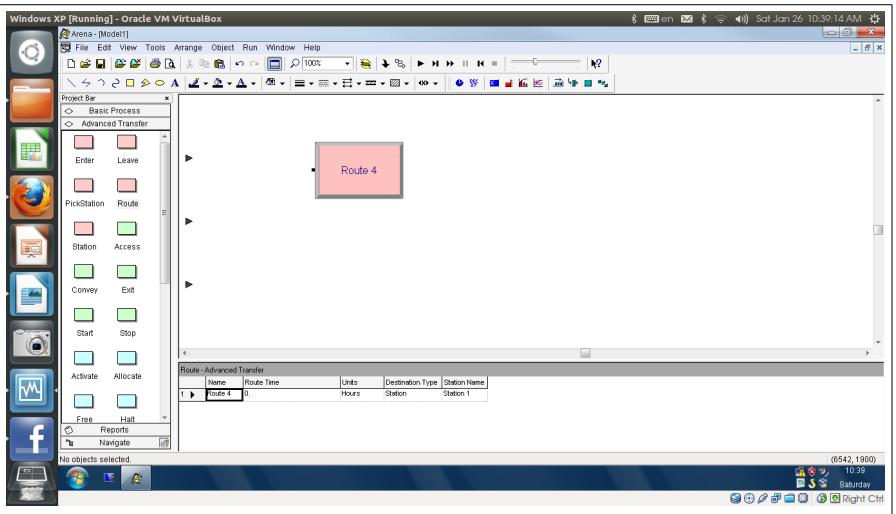
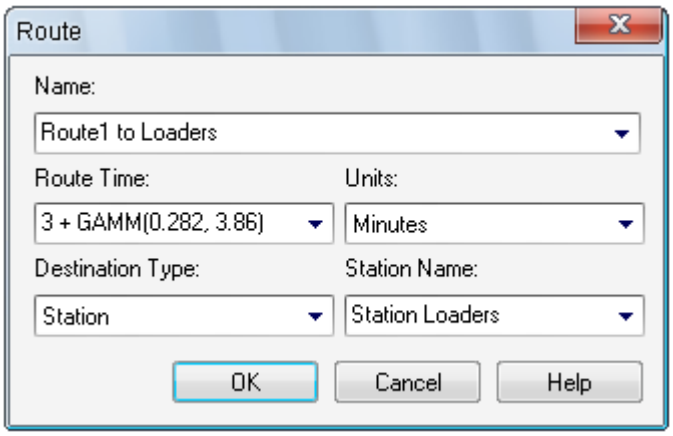
17. Click ขวาเลือก Properties...	
18. ใส่รายละเอียดตามรูป	

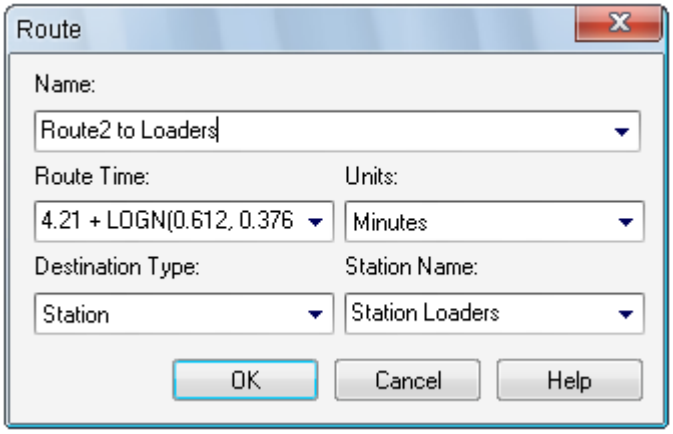
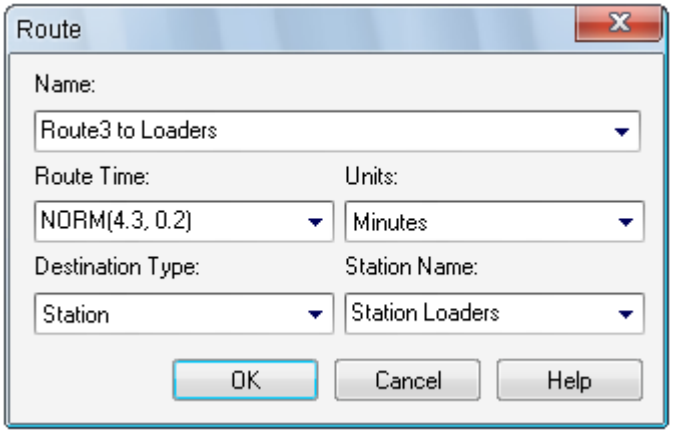
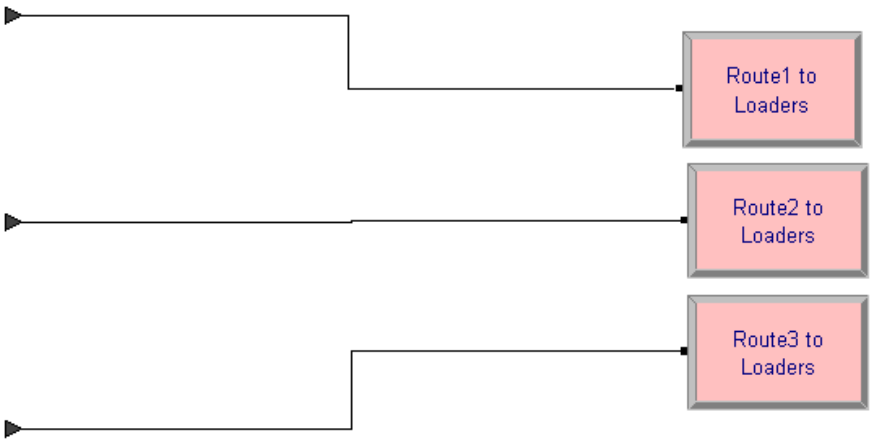
19. เลือก Connect	
20. เชื่อม Truck arr 1stLA, Truck arr 2ndLA และ Truck arr 3rdLA เข้ากับ Assigntruck1-3	

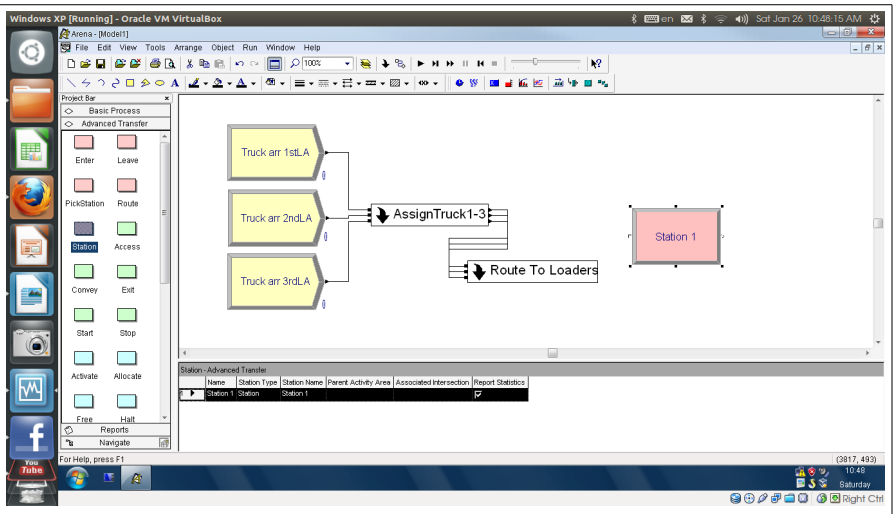
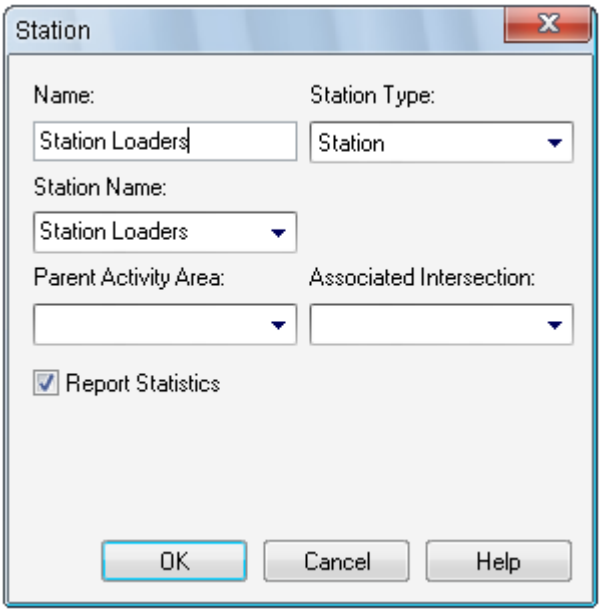
21. Double click ที่ Assigntruck1-3	
22. สร้าง Assign Module ของรถบรรทุกสำหรับจุดตักที่ 1	

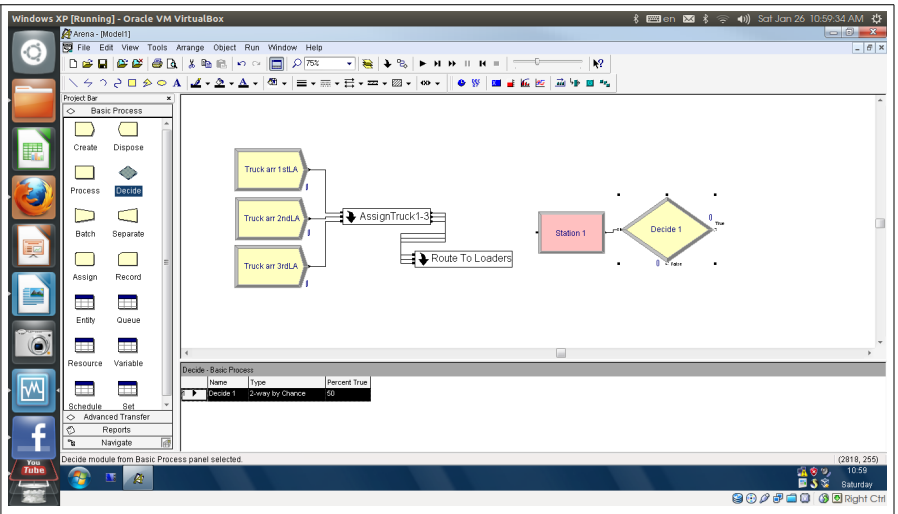
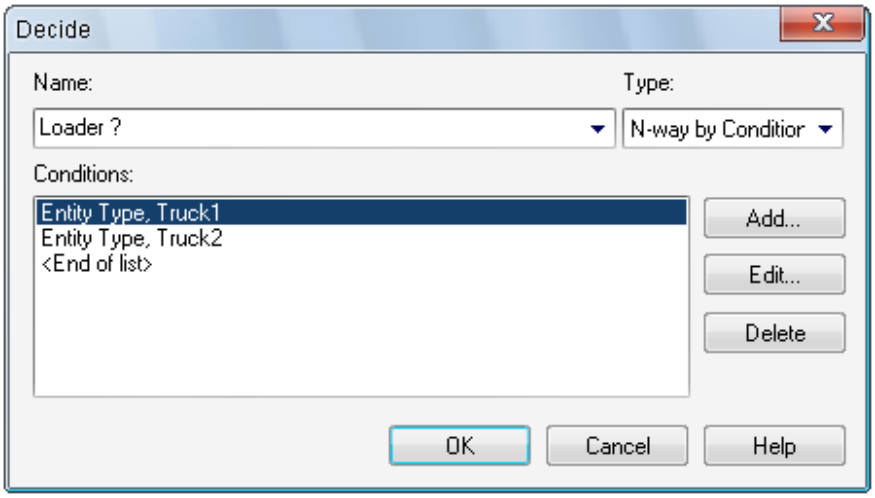
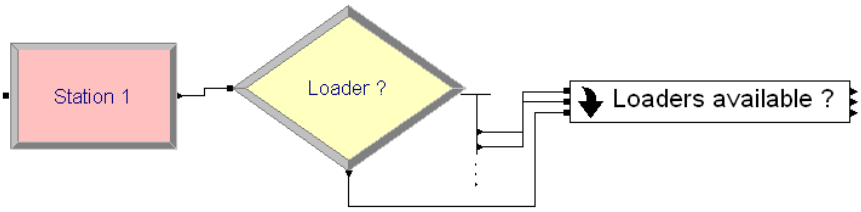
23. Double click แล้วใส่รายละเอียดตามรูป	
24. ทำเช่นเดียวกันสำหรับจุดตักที่ 2 และ 3	
25. สร้าง Connect ตามรูป	

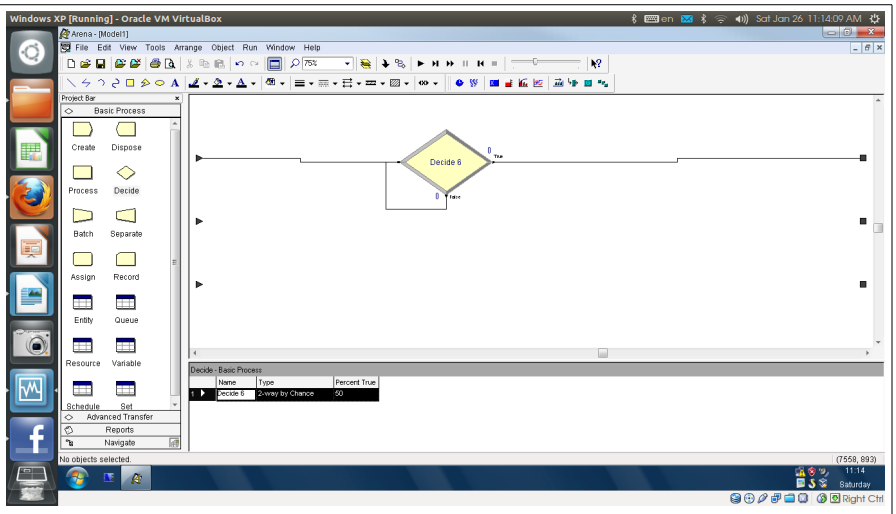
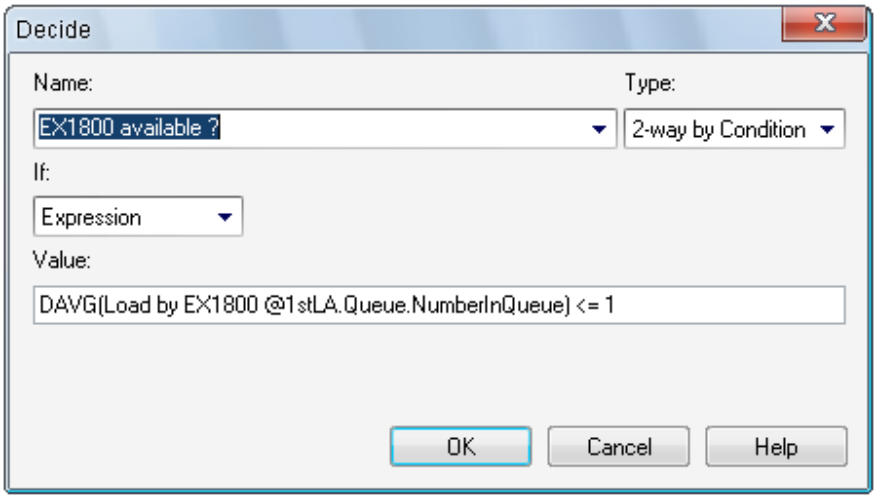
26. Click ขวาตรงพื้นที่ว่าง เลือก Close Submodel	
27. สร้าง Submodel ชื่อ Route To Loaders	

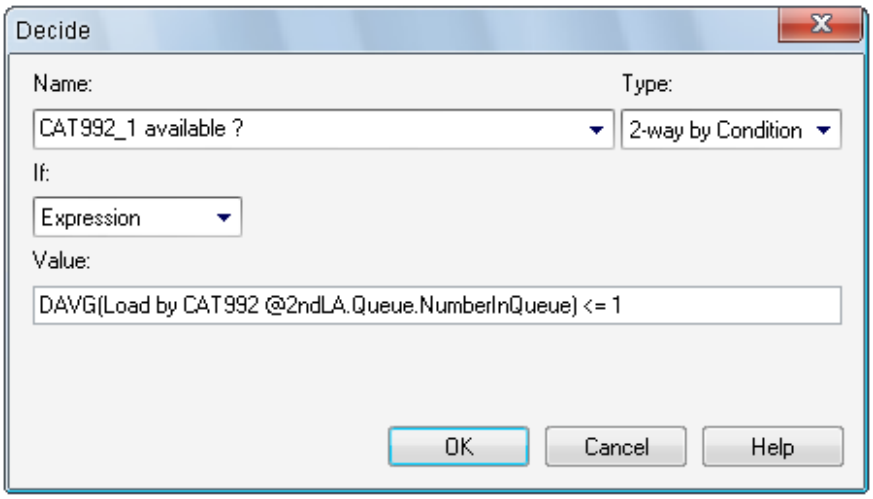
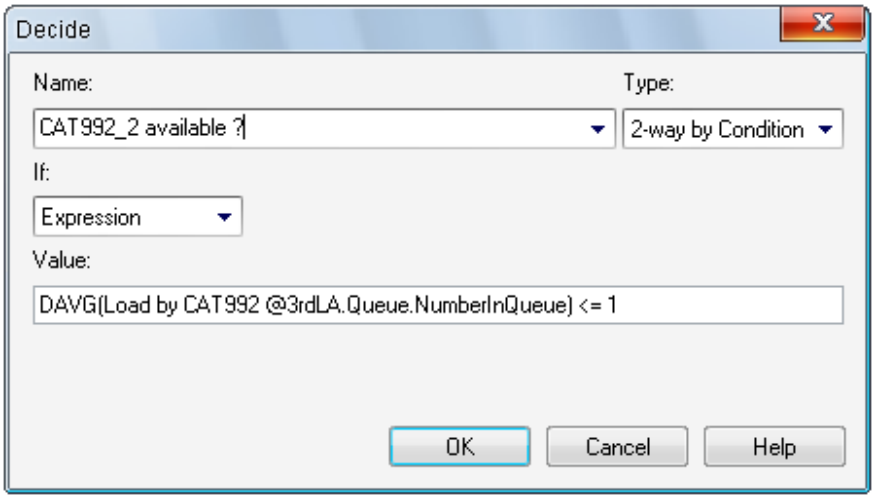
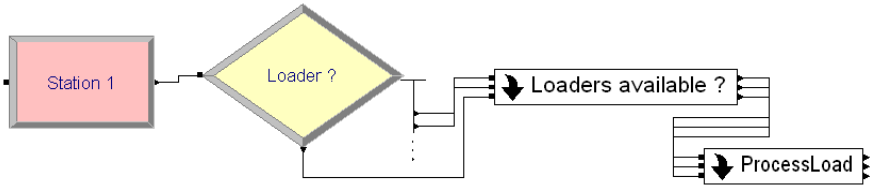
28. Doublc click ที่ Route To Loaders ที่ Project Bar เลือก Advanced Transfer แล้วสร้าง Route Module ของเส้นทางสำหรับจุดตักที่ 1	
29. Double click แล้วใส่รายละเอียดตามรูป สำหรับเส้นทางที่ 1	

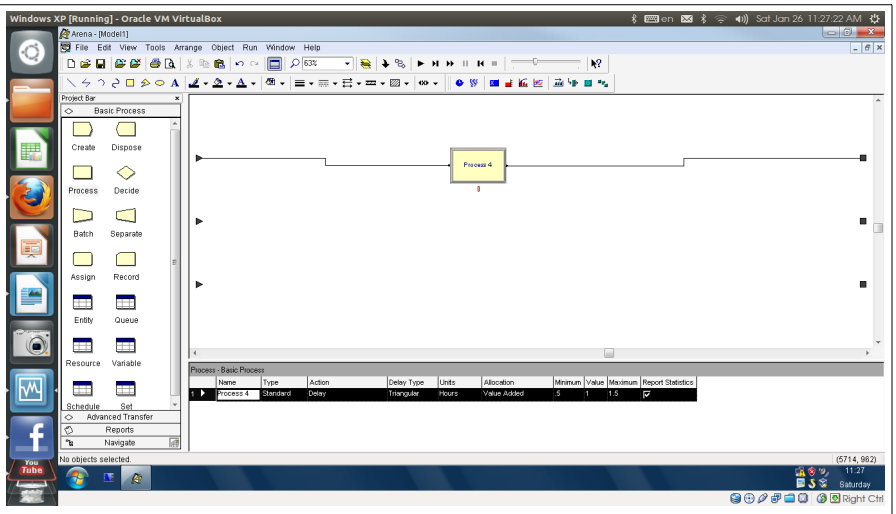
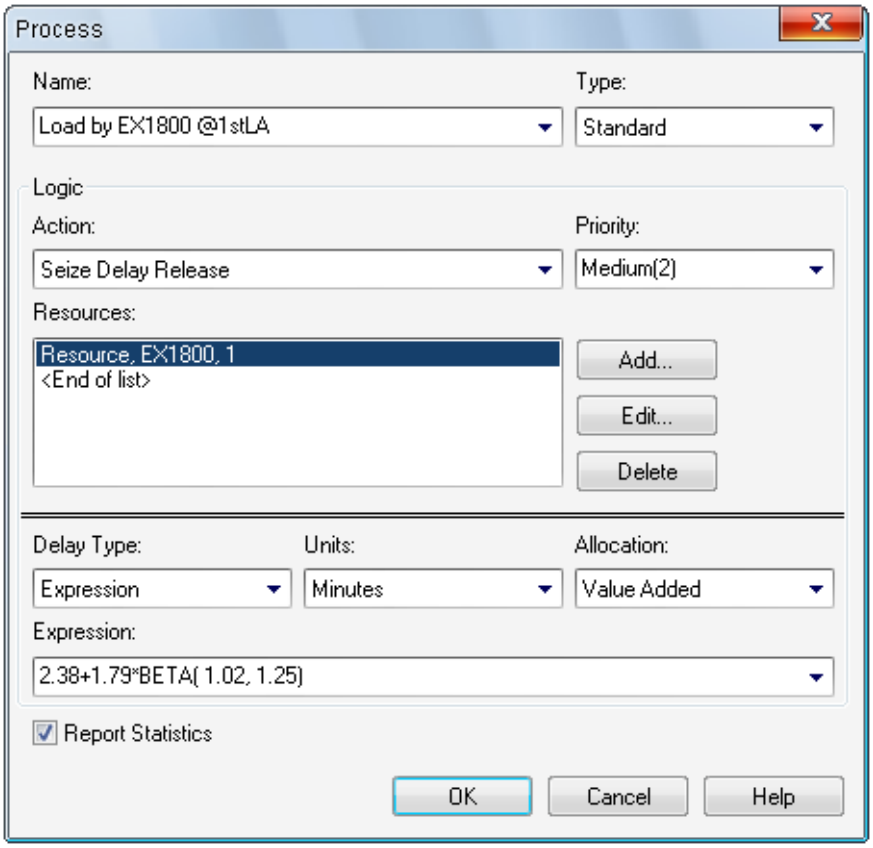
30. สำหรับเส้นทางที่ 2	
31. สำหรับเส้นทางที่ 3	
32. สร้าง Connect สำหรับเส้นทางทั้ง 3	

33. Click ขวาตรงพื้นที่ว่าง เลือก Close Submodel แล้วสร้าง Station Module	
34. ใส่รายละเอียดตามรูป	

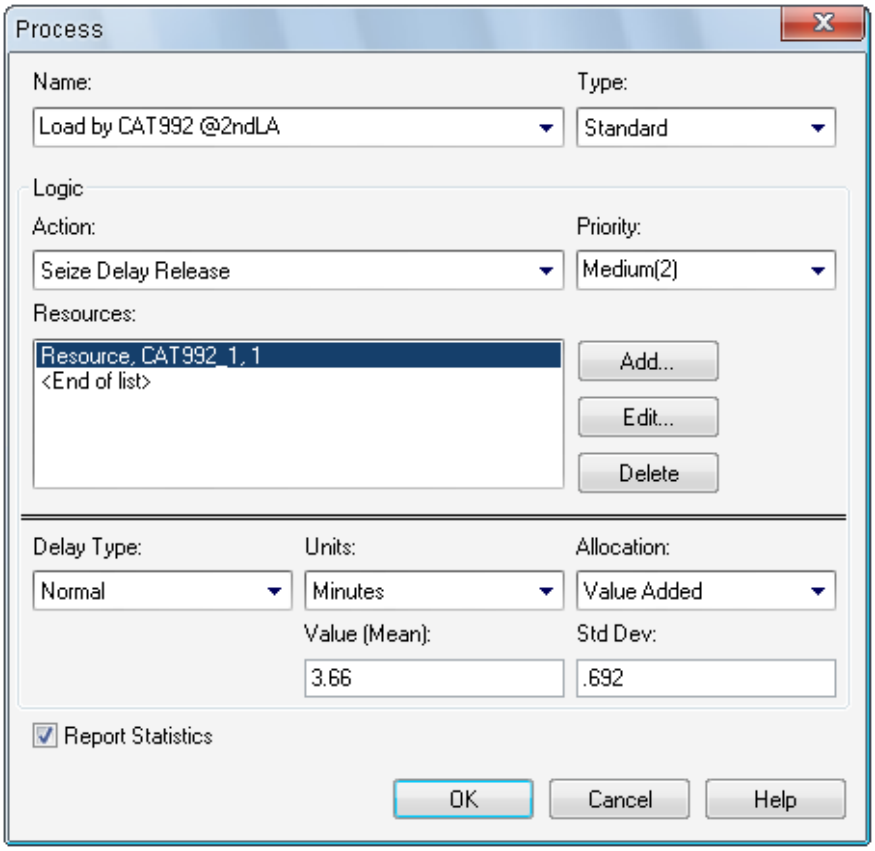
35. ที่ Project Bar เลือก Basic Process แล้วสร้าง Decide Module	
36. ใส่รายละเอียดตามรูป	
37. สร้าง Submodel ชื่อ Loaders available ?	

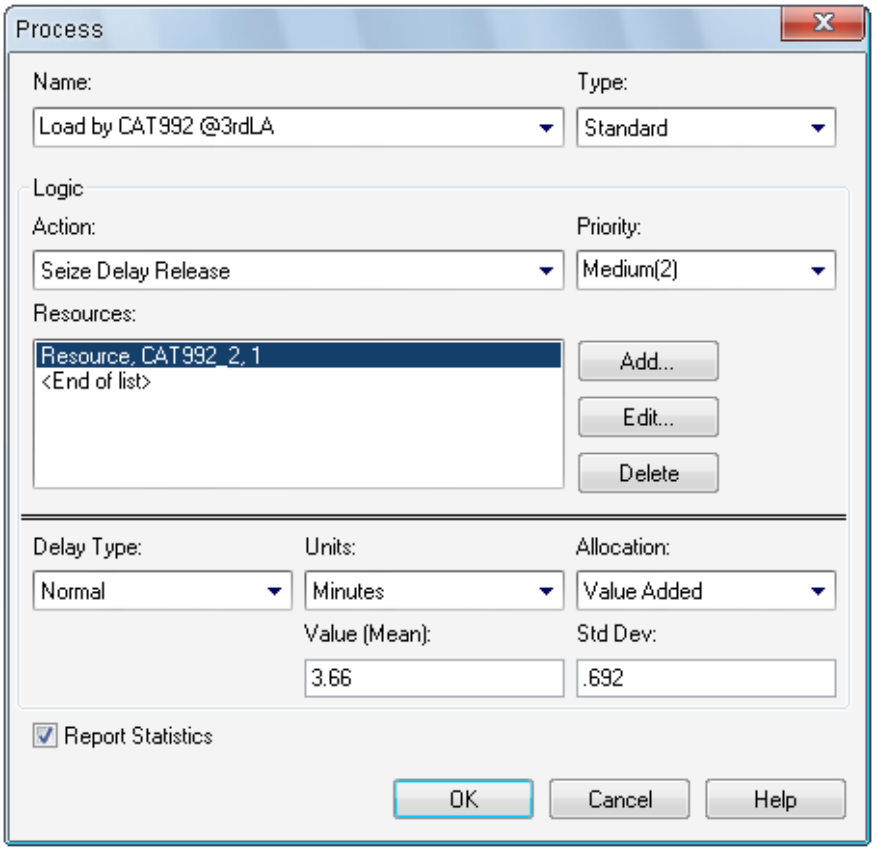
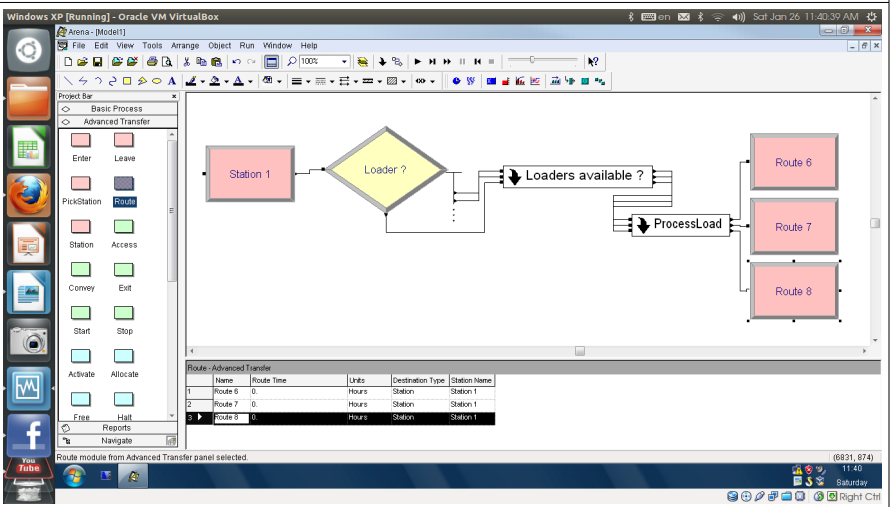
38. ใน Loaders available ? สร้าง Decide Module สำหรับรถตักคันที่ 1	
39. ใส่รายละเอียดสำหรับรถตักคันที่ 1 ตามรูป	

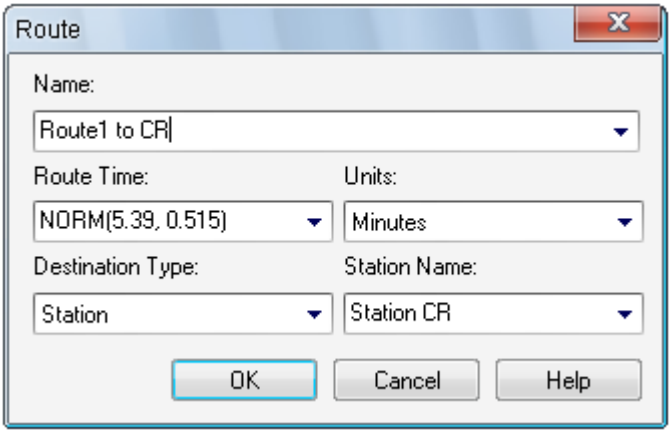
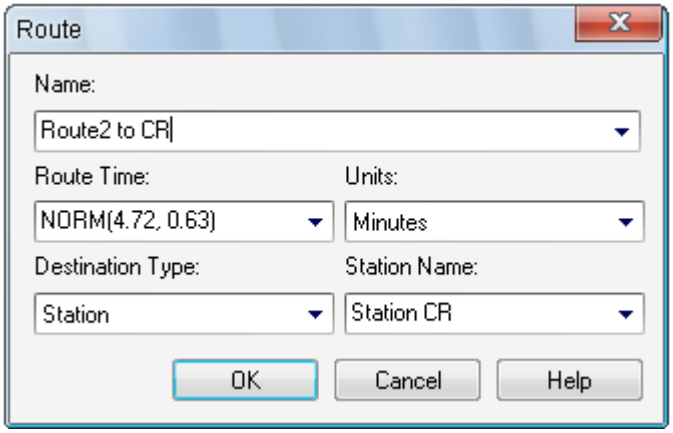
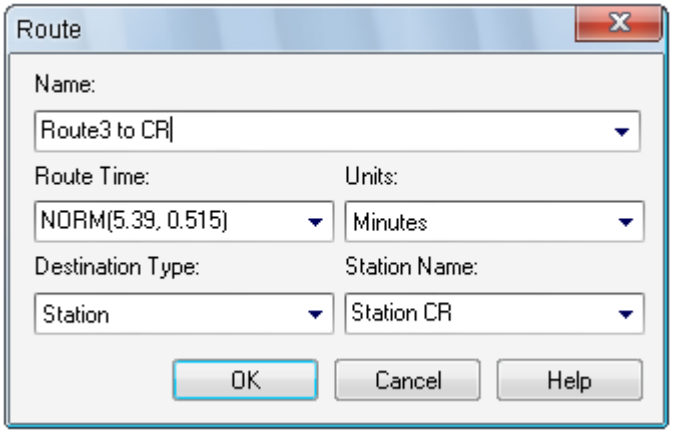
40. ใส่รายละเอียดสำหรับรถตักคันที่ 2 ตามรูป	
41. ใส่รายละเอียดสำหรับรถตักคันที่ 3 ตามรูป	
42. ออกจาก Submodel แล้วสร้าง Submodel ใหม่ชื่อ PrecessLoad	

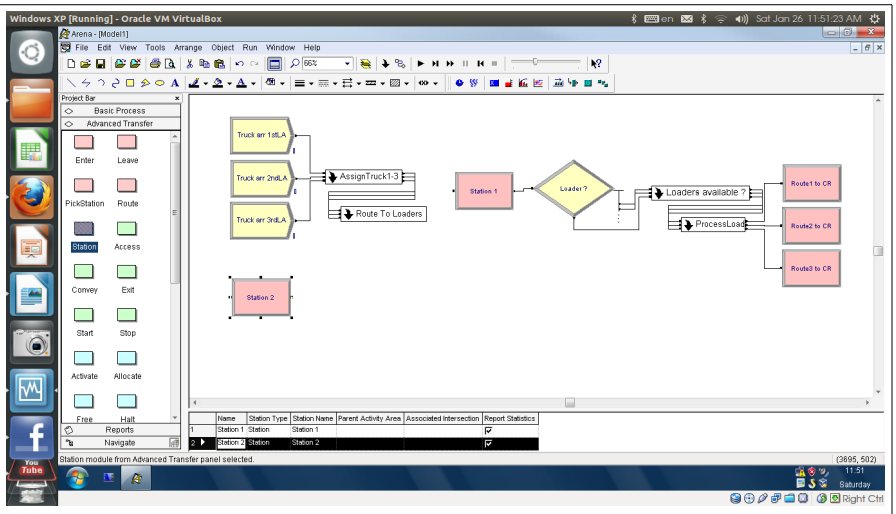
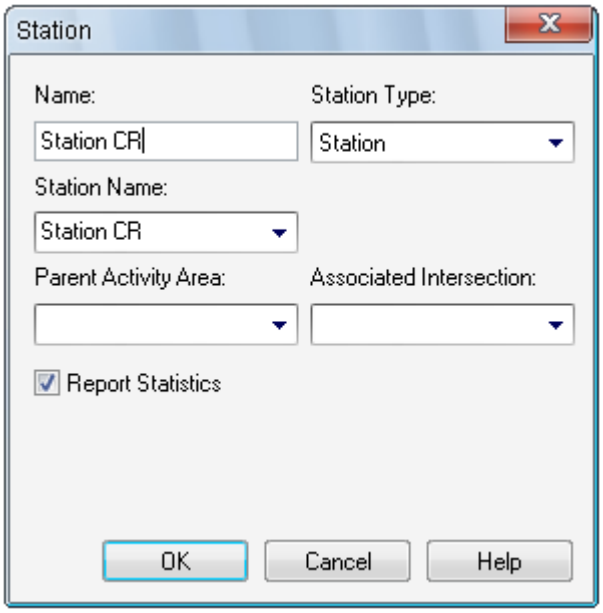
43. ใน ProcessLoad สร้าง Process Module สำหรับรถตักคันที่ 1	
44. ใส่รายละเอียดสำหรับรถตักคันที่ 1 ตามรูป	

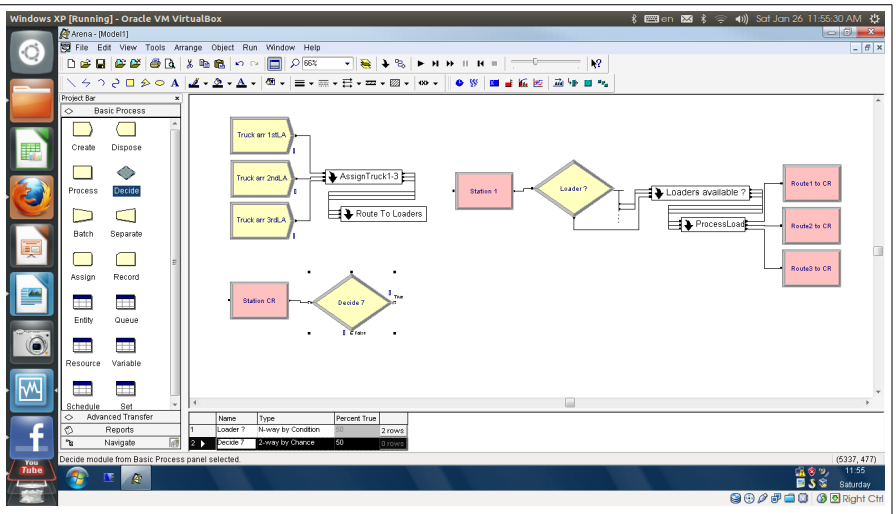
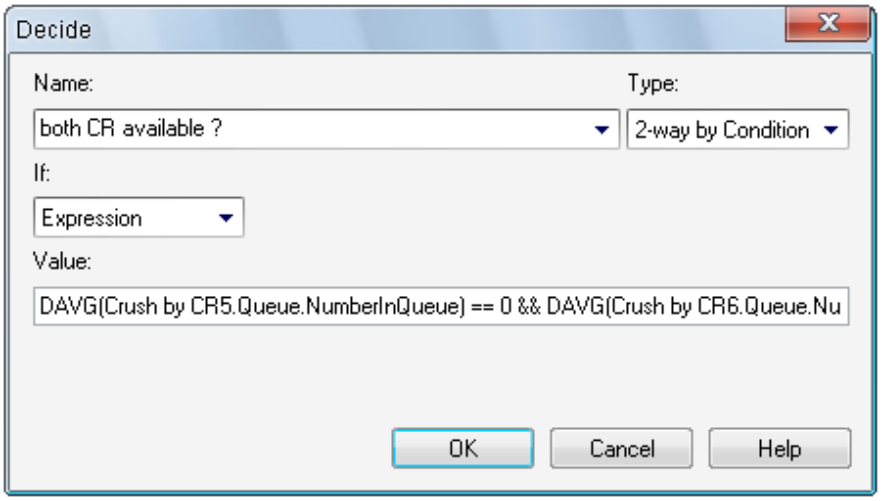
45. ใส่รายละเอียดสำหรับรถตักคันที่ 2 ตามรูป

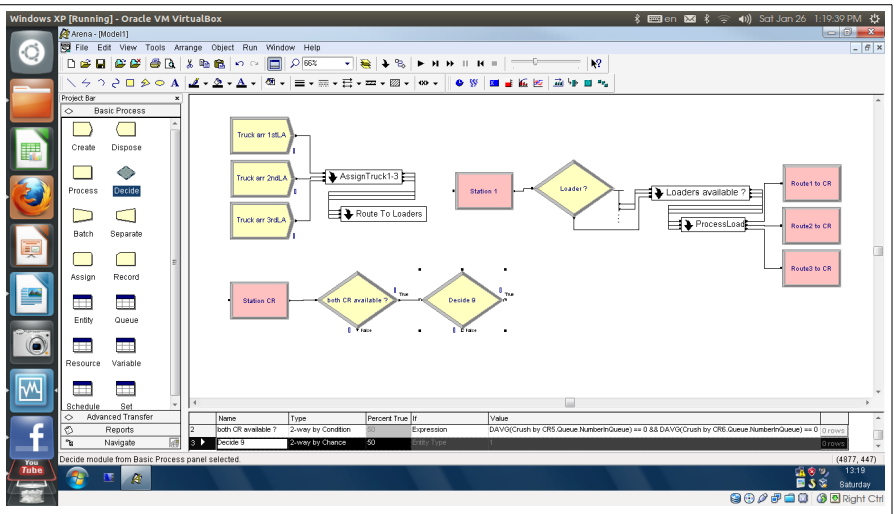
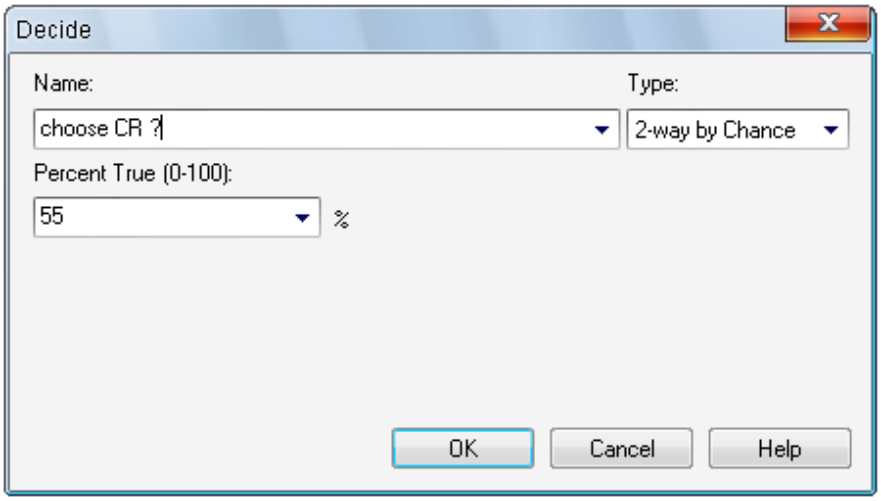


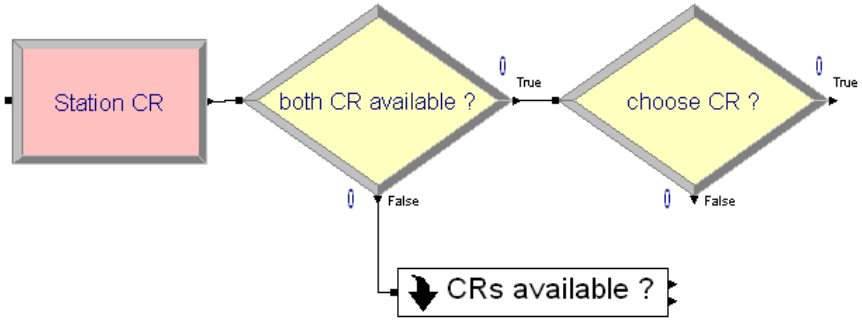
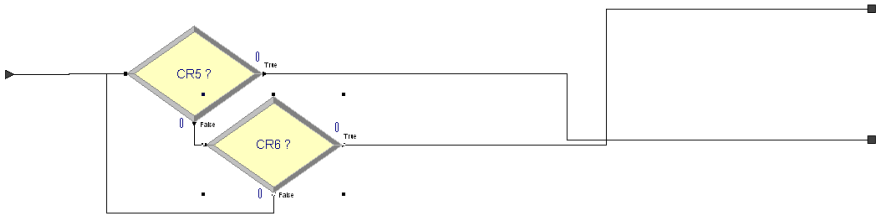
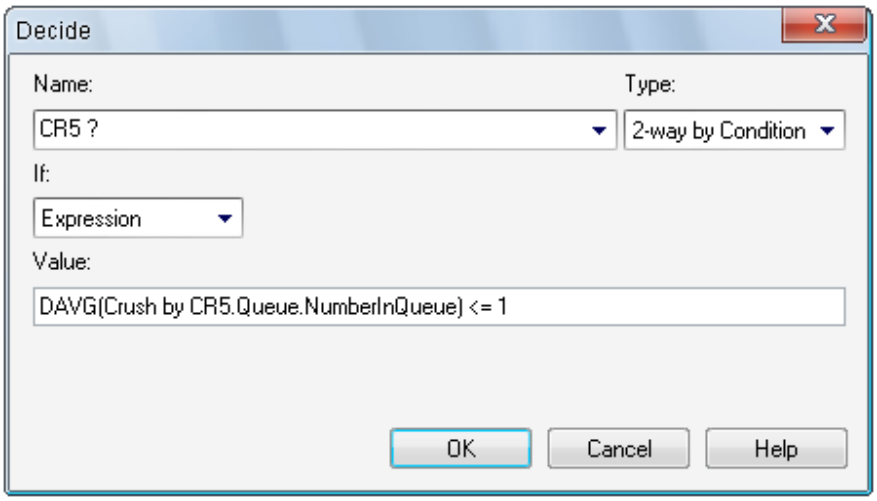
46. ใส่รายละเอียดสำหรับรถตักคันที่ 3 ตามรูป	
47. ออกจาก Submodel แล้วสร้าง Route Module สำหรับ 3 เส้นทาง	

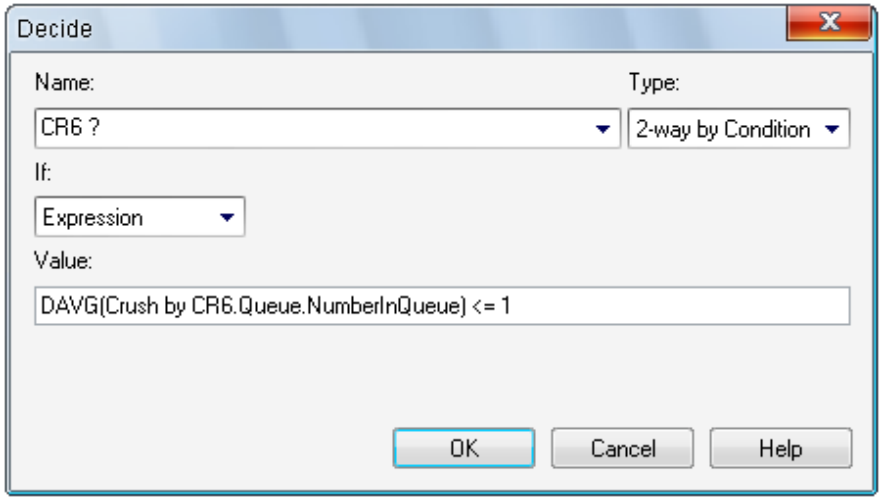
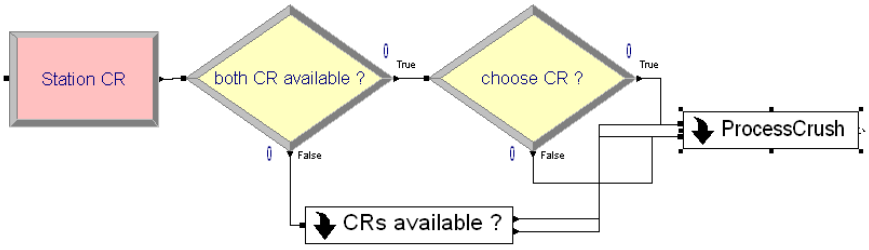
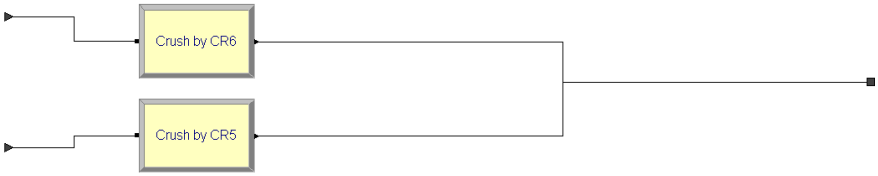
48. ใส่รายละเอียดสำหรับเส้นทางที่ 1 ตามรูป	
49. ใส่รายละเอียดสำหรับเส้นทางที่ 2 ตามรูป	
50. ใส่รายละเอียดสำหรับเส้นทางที่ 3 ตามรูป	

51. สร้าง Station Module	
52. ใส่รายละเอียดตามรูป	

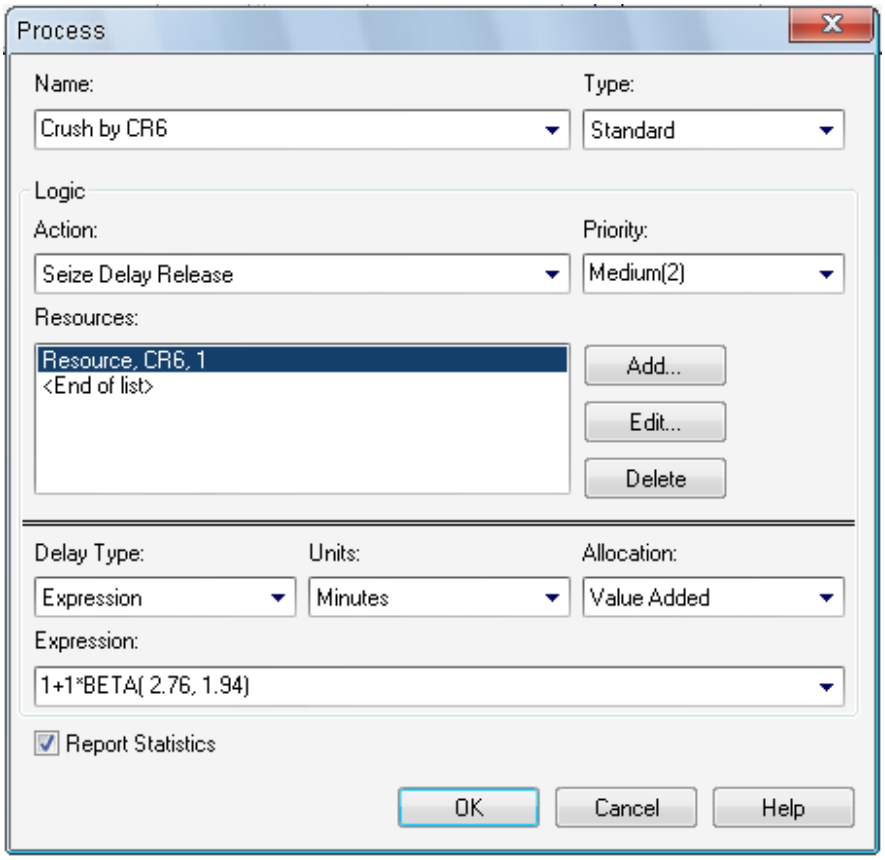
53. สร้าง Decide Module	
54. ใส่รายละเอียดตามรูป	

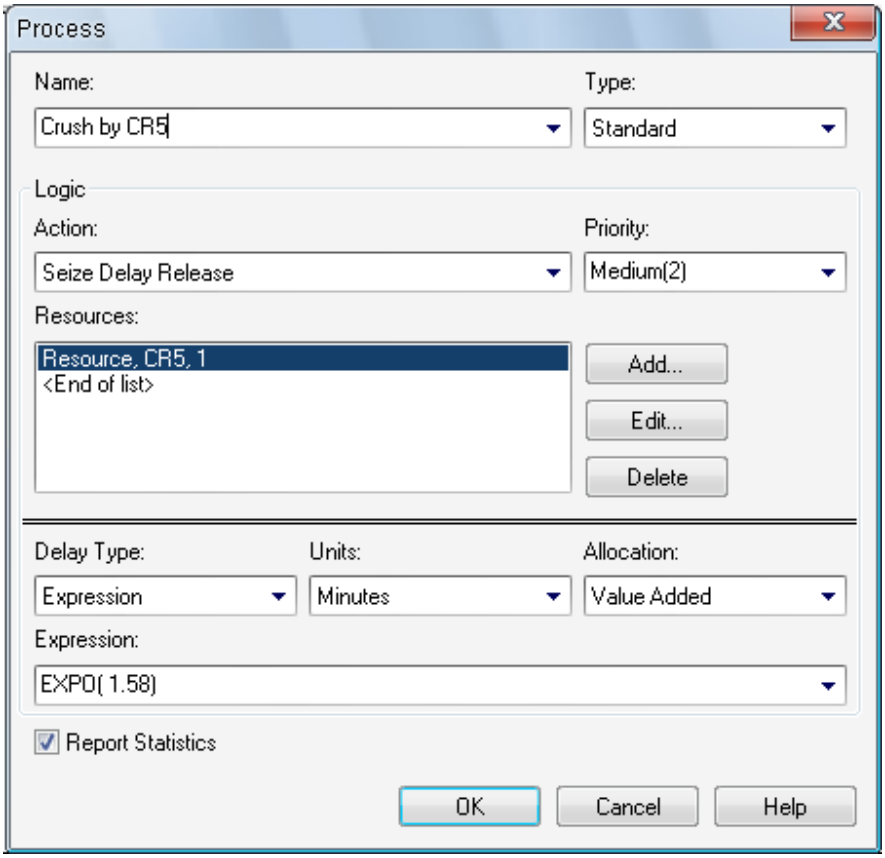
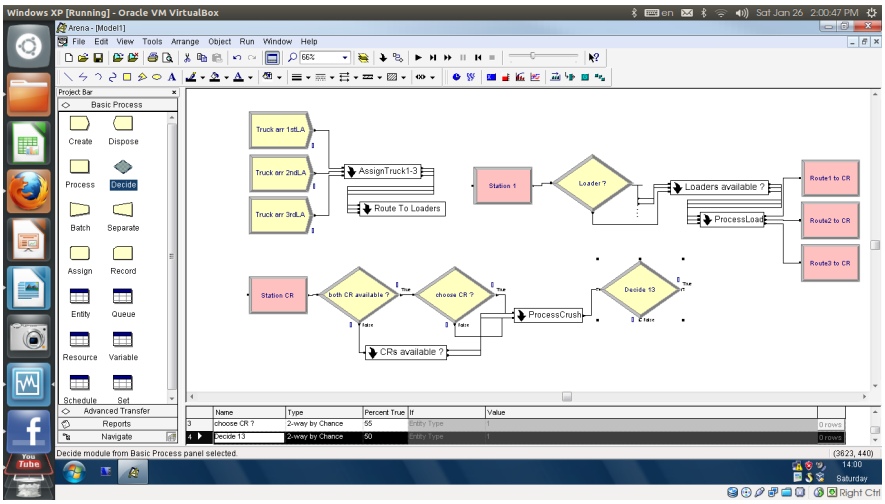
55. สร้าง Decide Module	
56. ใส่รายละเอียดตามรูป	

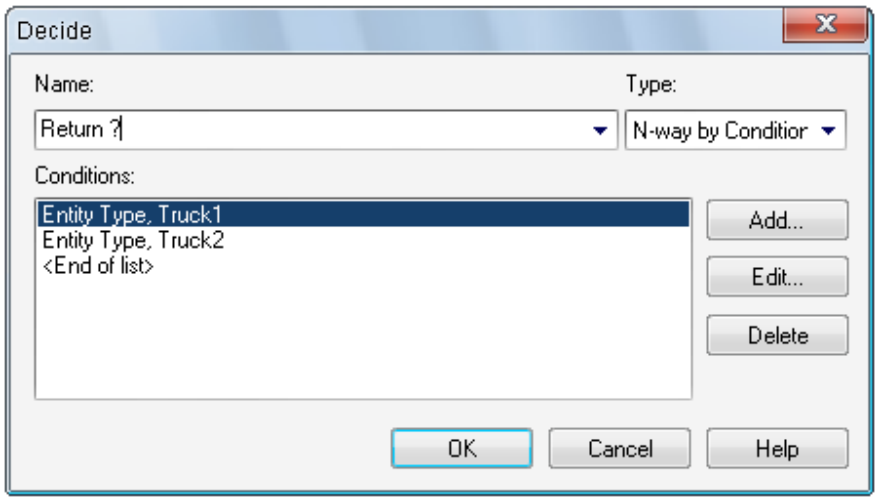
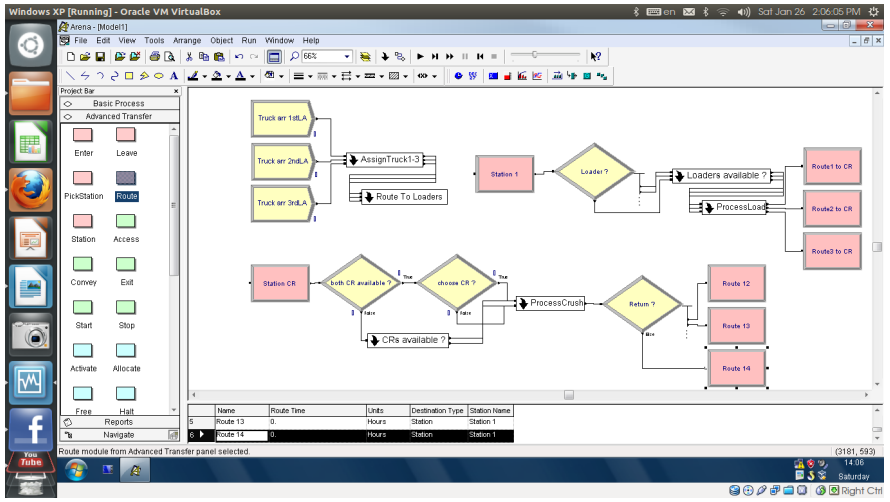
57. สร้าง Submodel ชื่อว่า CRs available ?	
58. ใน CRs available ? สร้าง Decide Module ตามรูป	
59. ใส่รายละเอียดของ CR5 ? ตามรูป	

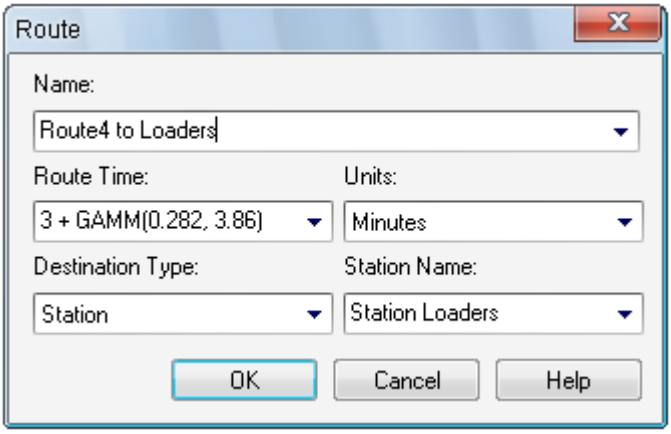
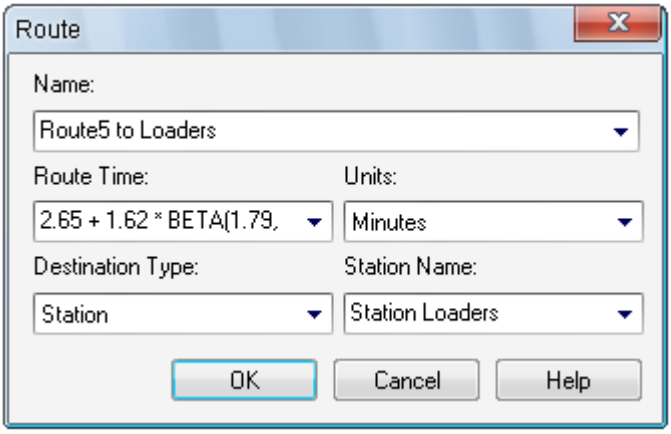
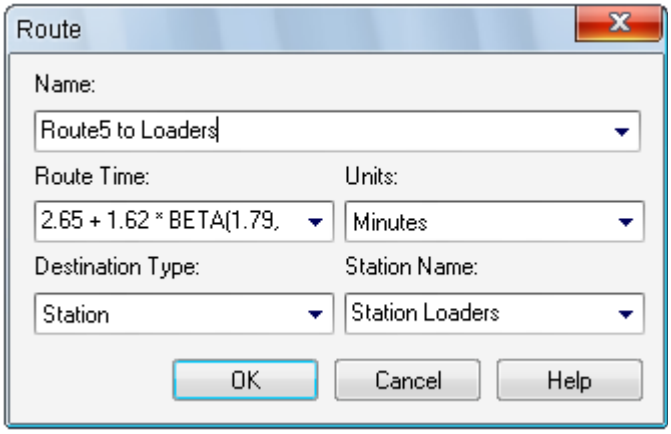
60. ใส่รายละเอียดของ CR6 ? ตามรูป	
61. สร้าง Submodel ชื่อ ProcessCrush	
62. ใน ProcessCrush สร้าง Process Module ตามรูป	

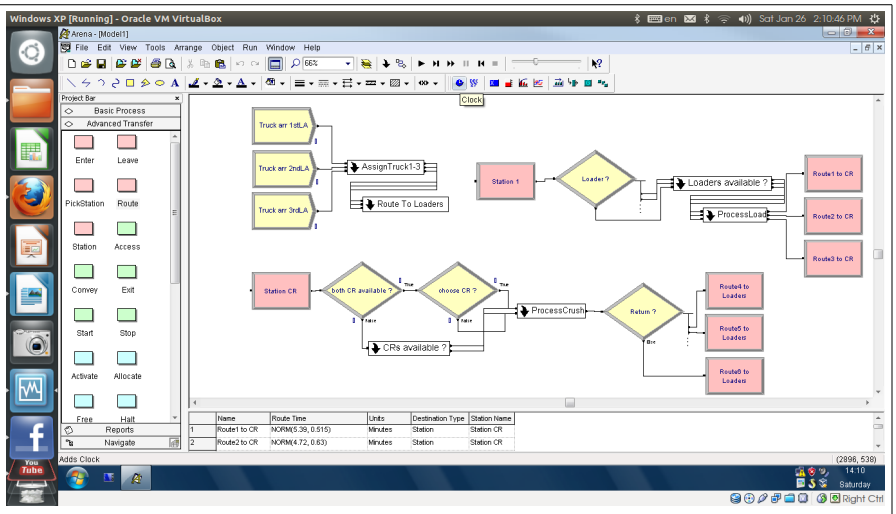
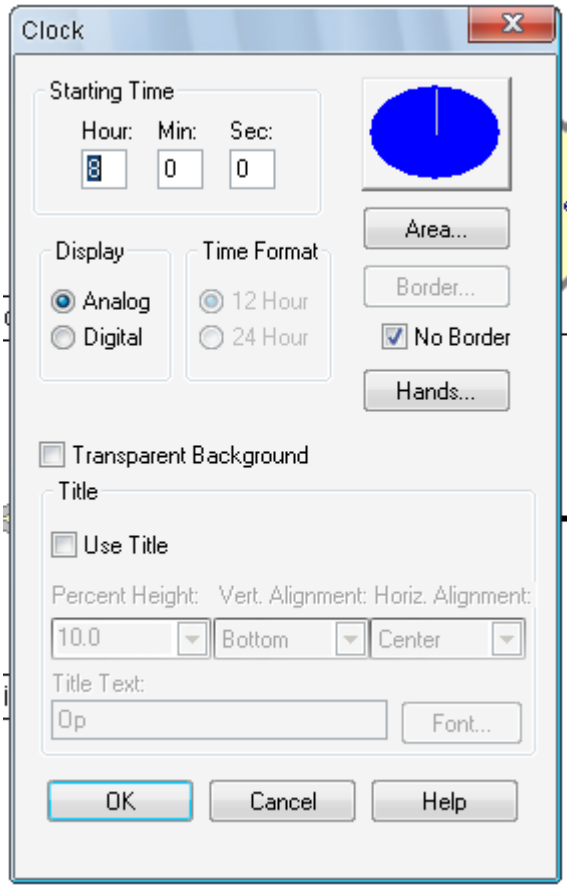
63. ใส่รายละเอียดของ Crush by CR6 ตามรูป

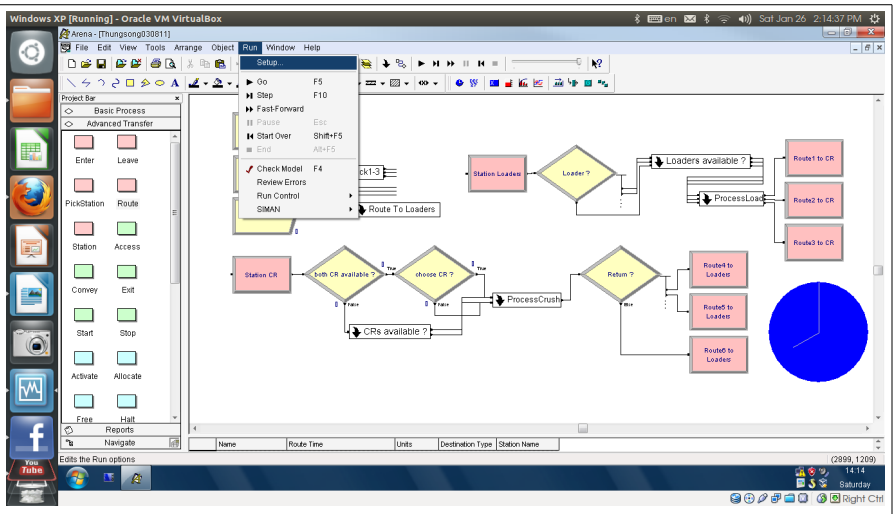


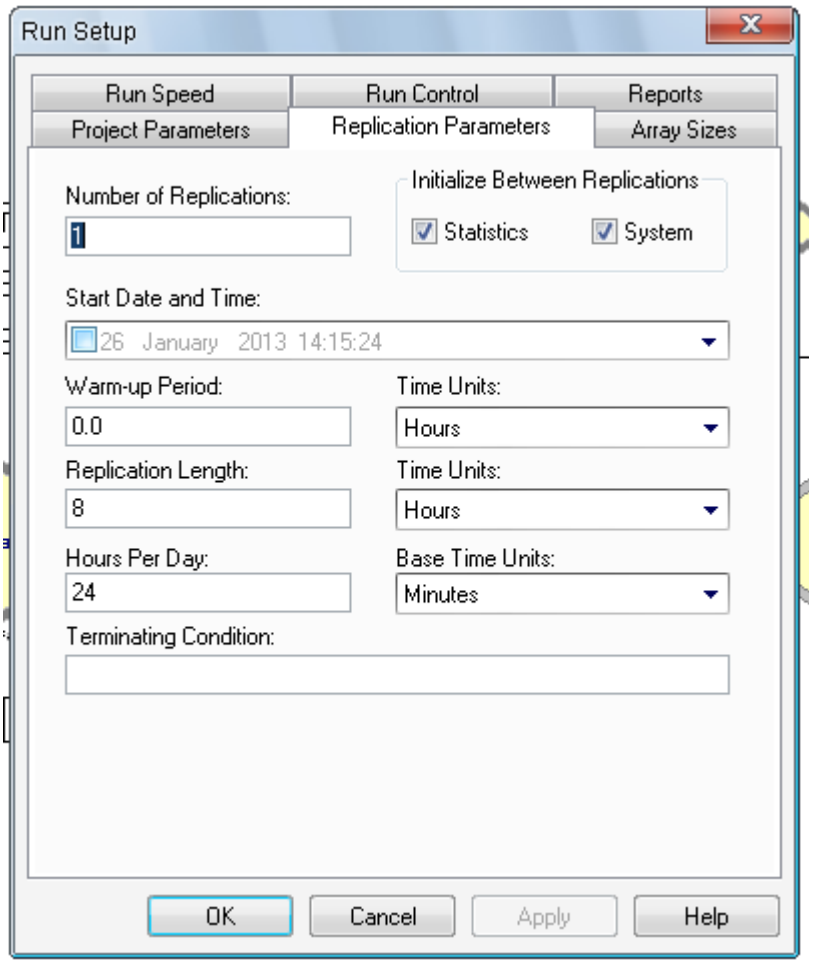
64. ใส่รายละเอียดของ Crush by CR5 ตามรูป	
65. สร้าง Decide Module	

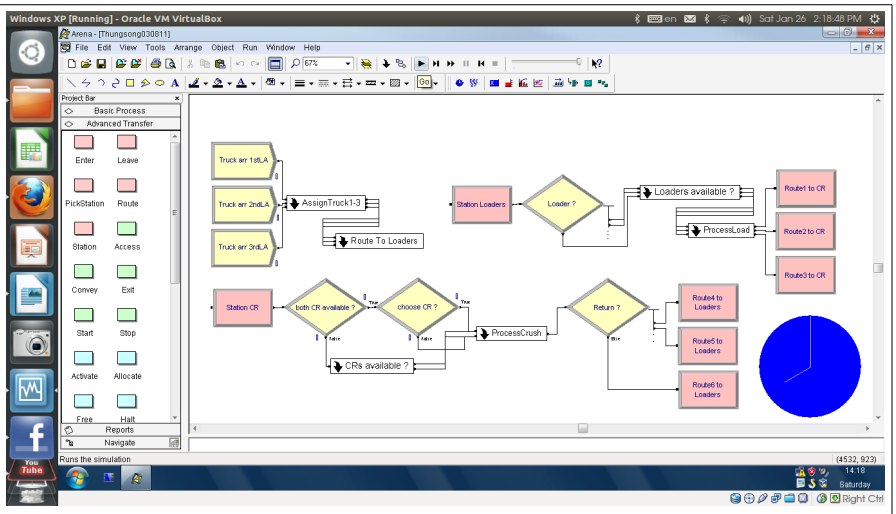
66. ใส่รายละเอียดตามรูป	
67. สร้าง Route Module สำหรับ 3 เส้นทาง	

68. ใส่รายละเอียดสำหรับเส้นทางที่ 1 ตามรูป	
69. ใส่รายละเอียดสำหรับเส้นทางที่ 2 ตามรูป	
70. ใส่รายละเอียดสำหรับเส้นทางที่ 3 ตามรูป	

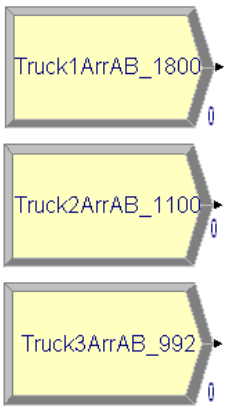
71. เลือก Clock ตามรูป	
72. ใส่รายละเอียดตามรูป	

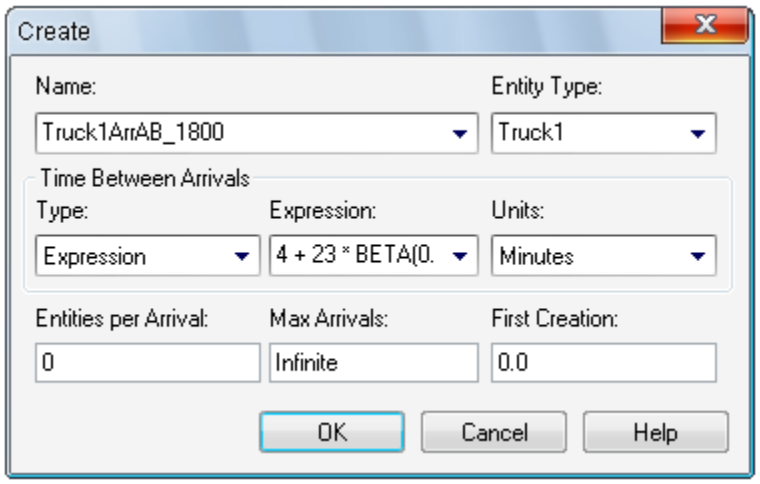
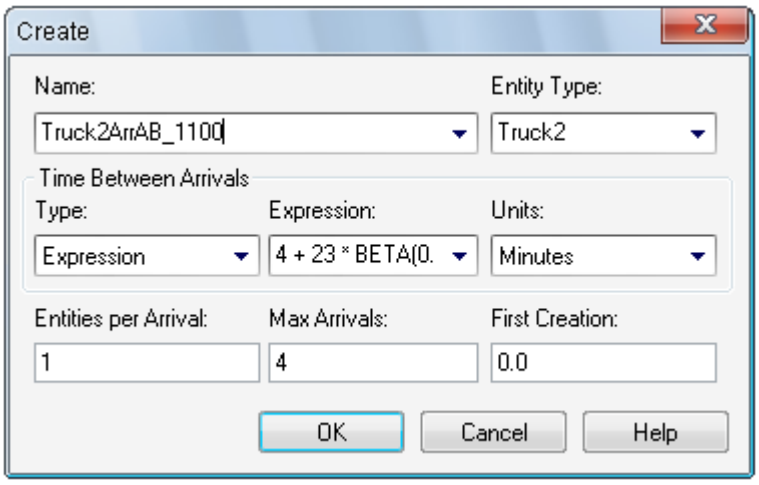
73. เลือก Setup...	

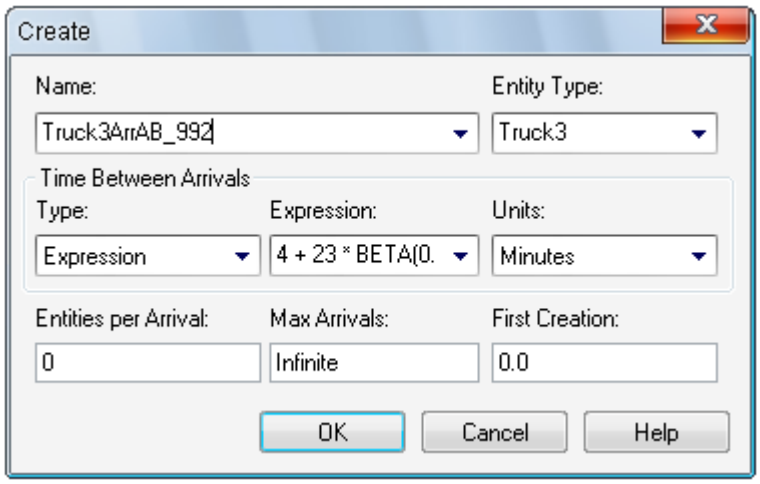
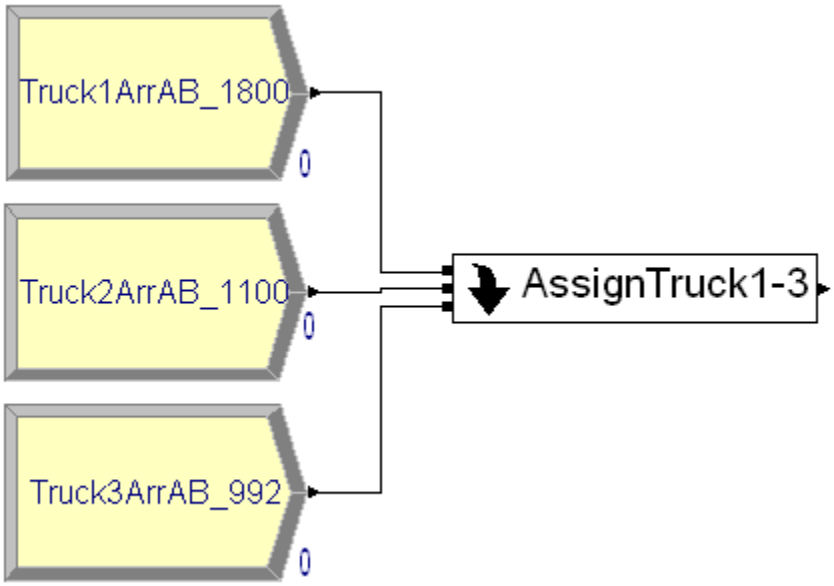
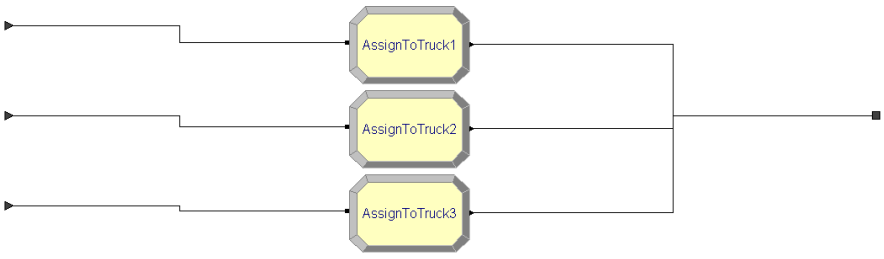
74. ใส่รายละเอียดตามรูป	

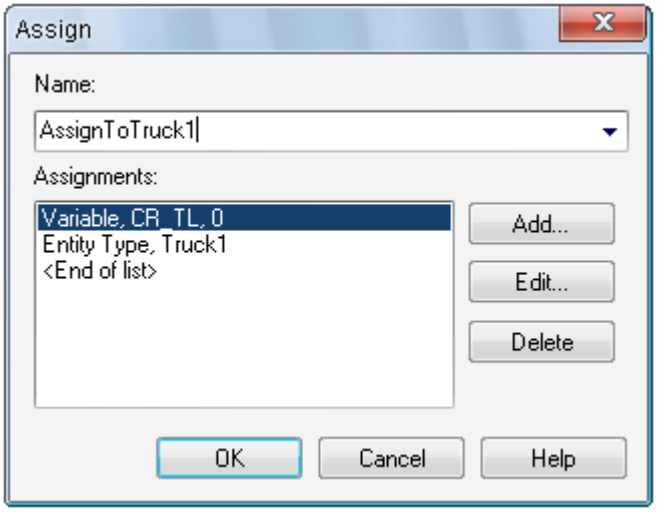
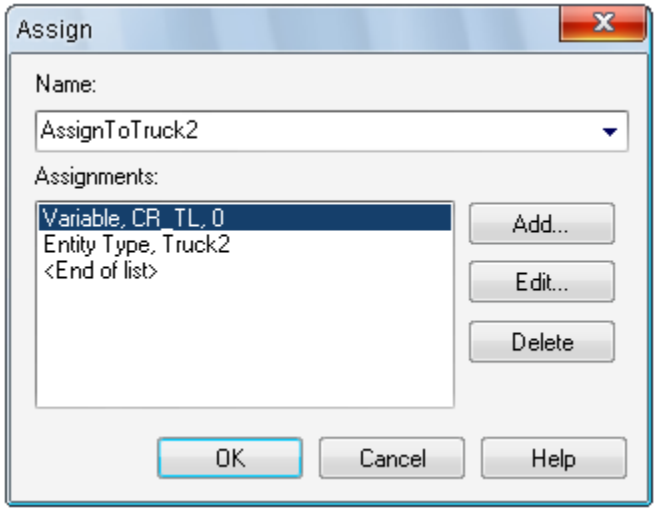
75. Click Go เพื่อรันโปรแกรม	

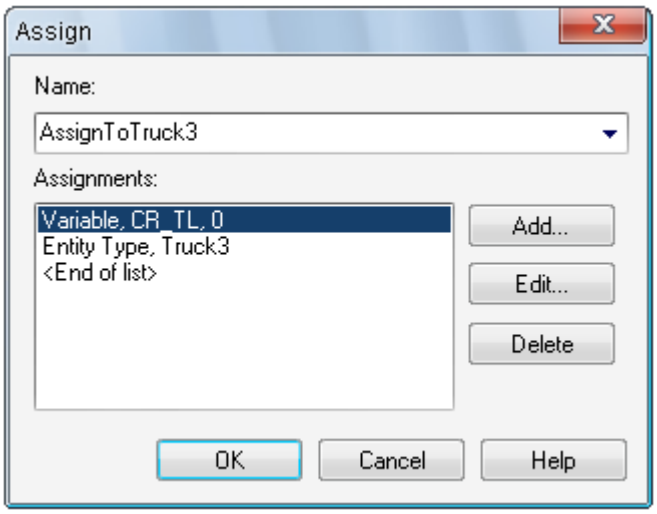
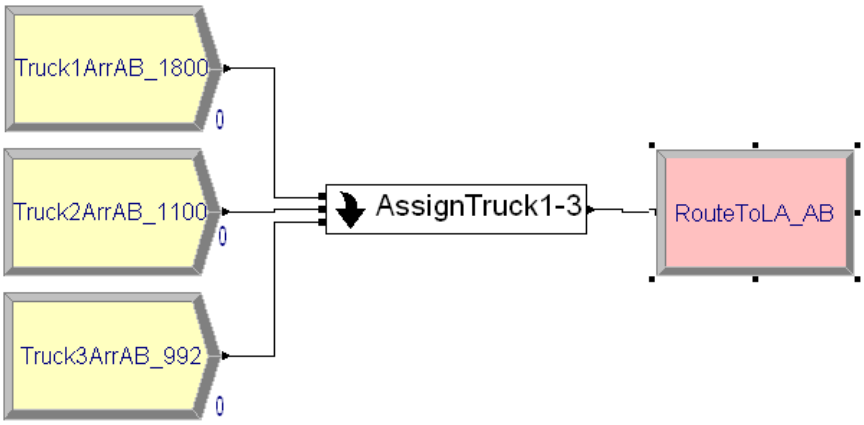
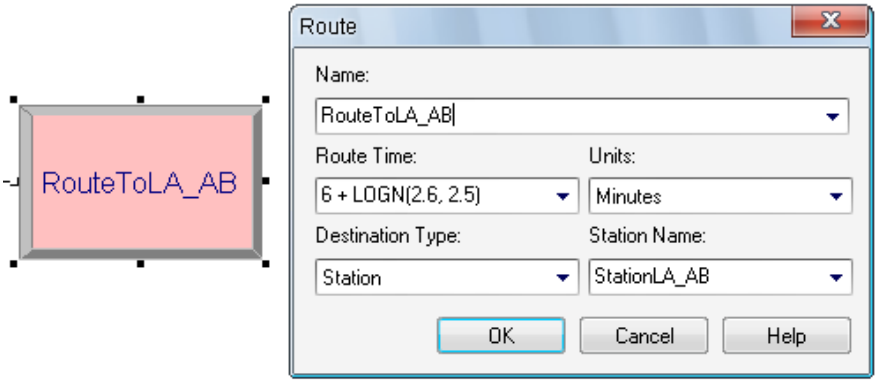
ขั้นตอนการสร้างแบบจำลองด้วยโปรแกรม Arena
สำหรับ บริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด

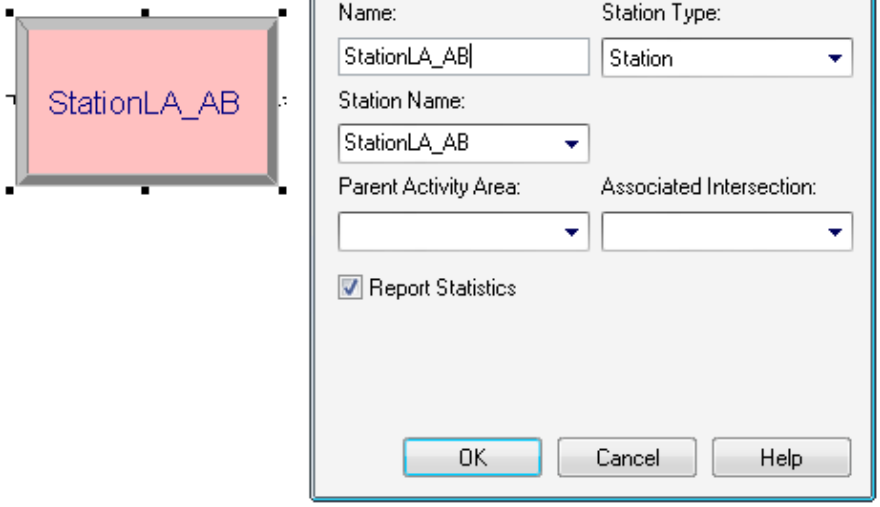
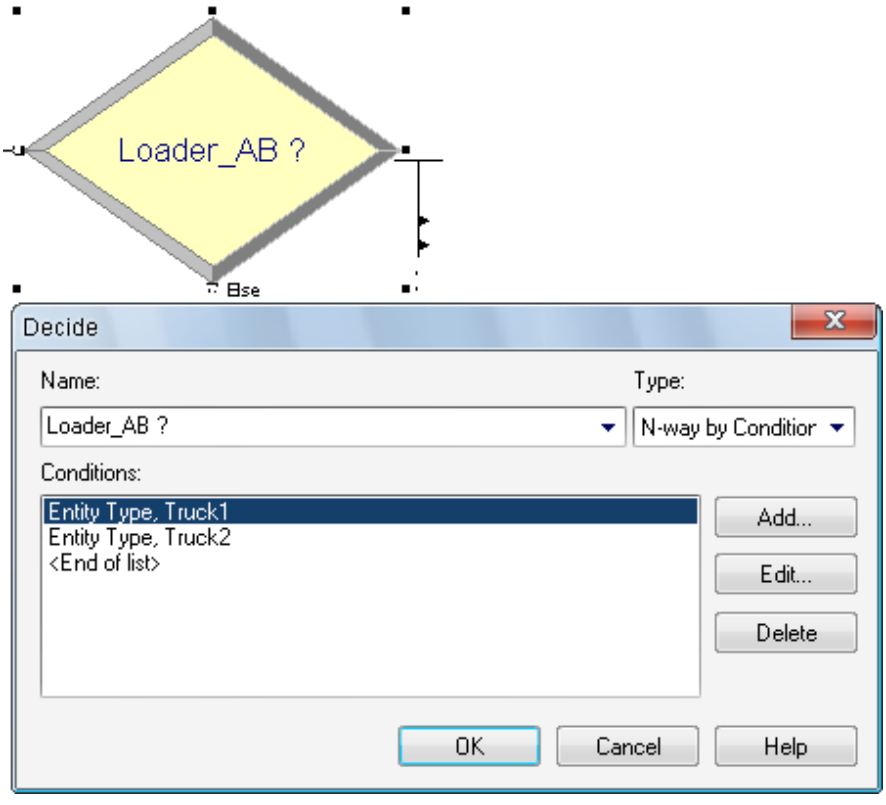
ลำดับขั้นตอน	ภาพประกอบ
1. สร้าง Create Module ของรถบรรทุกสำหรับรถตัก 3 ประเภทของเครื่องย่อยที่ 1	

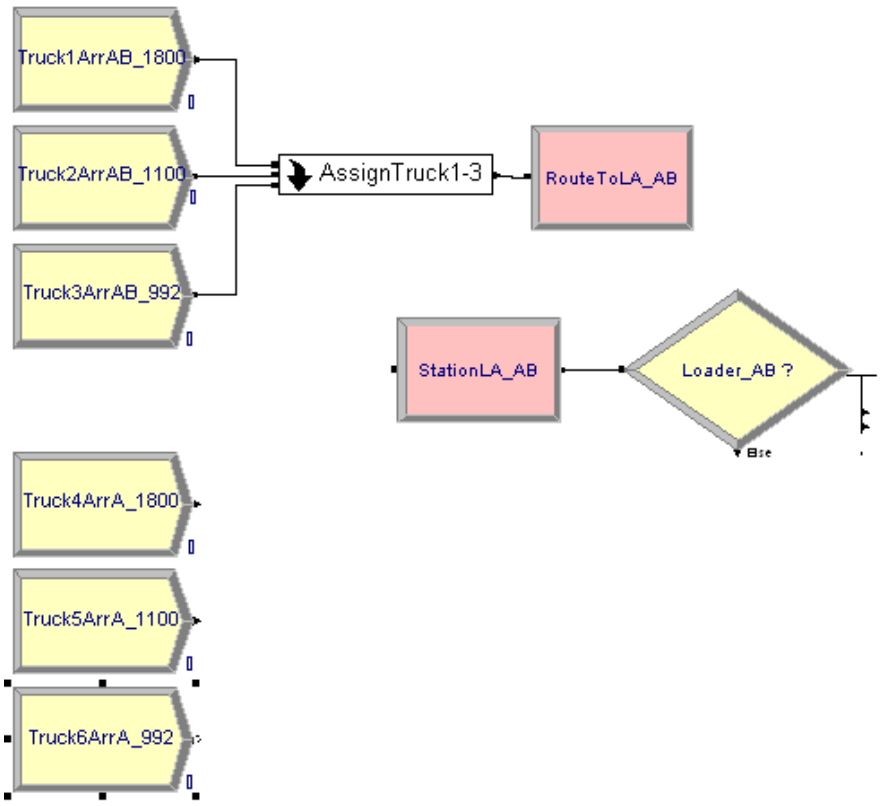
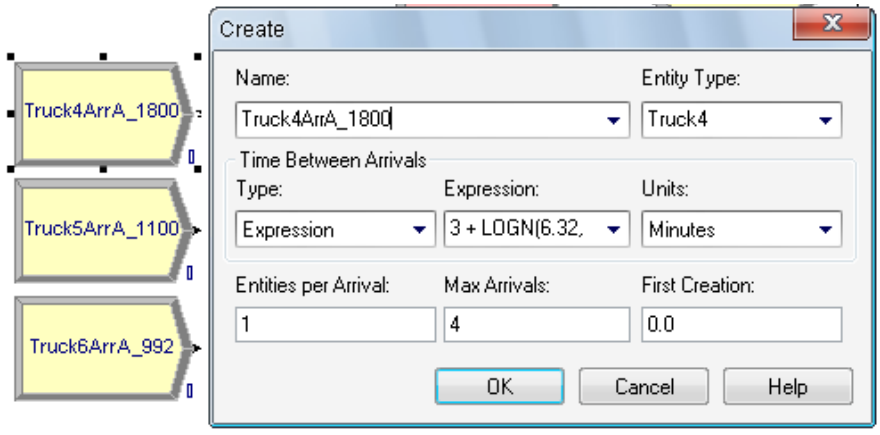
2. ใส่รายละเอียดของรถบรรทุกกลุ่มที่ 1	
3. ใส่รายละเอียดของรถบรรทุกกลุ่มที่ 2	

4. ใส่รายละเอียดของรถบรรทุกกลุ่มที่ 3	
5. สร้าง Submodel ชื่อ AssignTruck1-3	
6. ใน AssignTruck1-3 สร้าง Assign Module	

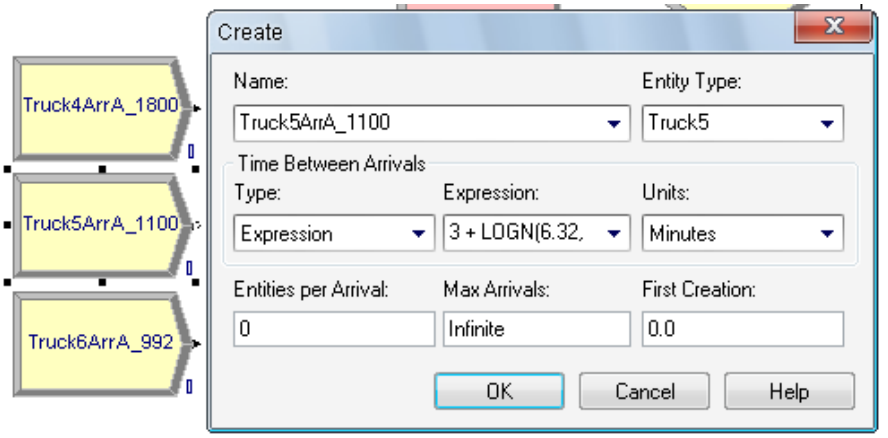
7. ใส่รายละเอียดของ AssignToTruck1	
8. ใส่รายละเอียดของ AssignToTruck2	

9. ใส่รายละเอียดของ AssignToTruck3	
10. สร้าง Route Module	
11. ใส่รายละเอียดตามรูป	

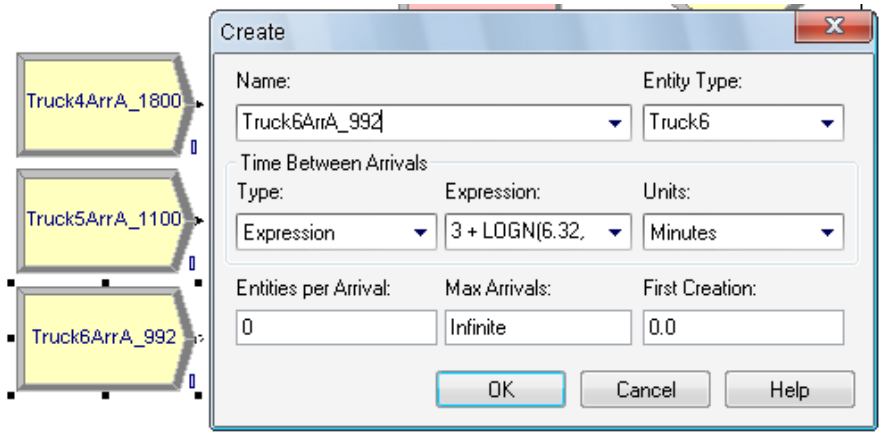
12. สร้าง Station Module	
13. สร้าง Decide Module	

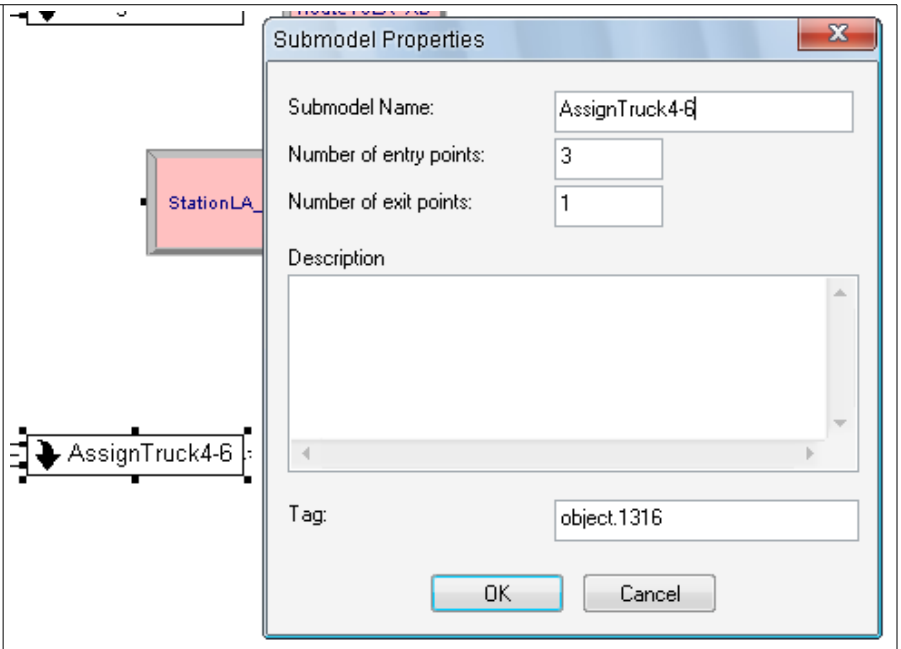
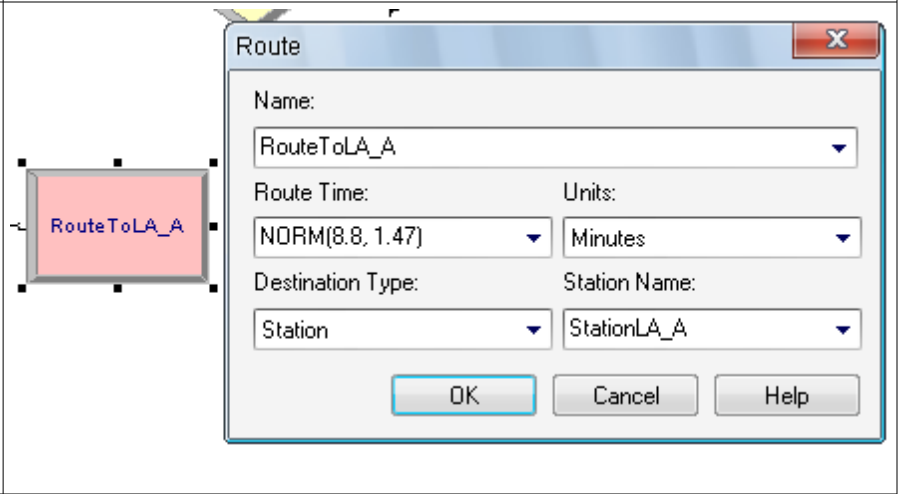
14. สร้าง Create Module ของรถบรรทุกสำหรับรถตัก 3 ประเภทของเครื่องย่อยที่ 1	
15. ใส่รายละเอียดของรถบรรทุกกลุ่มที่ 1	

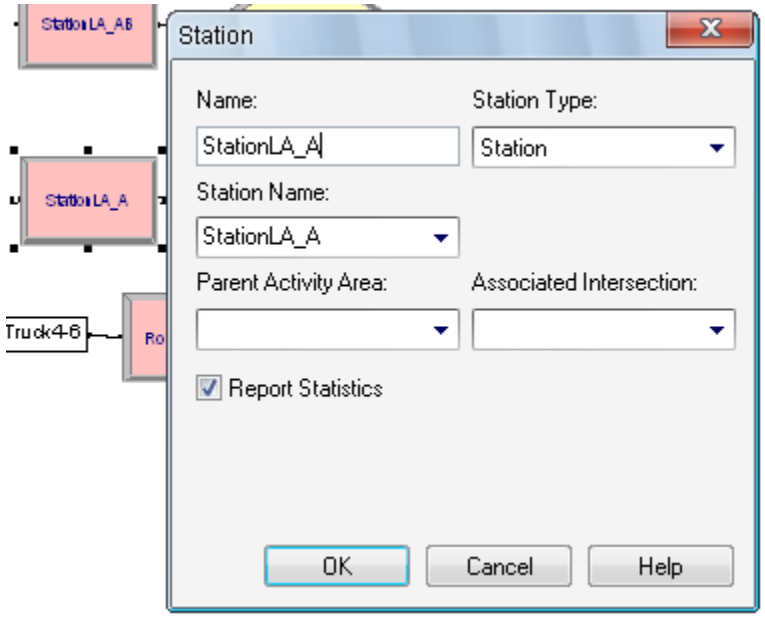
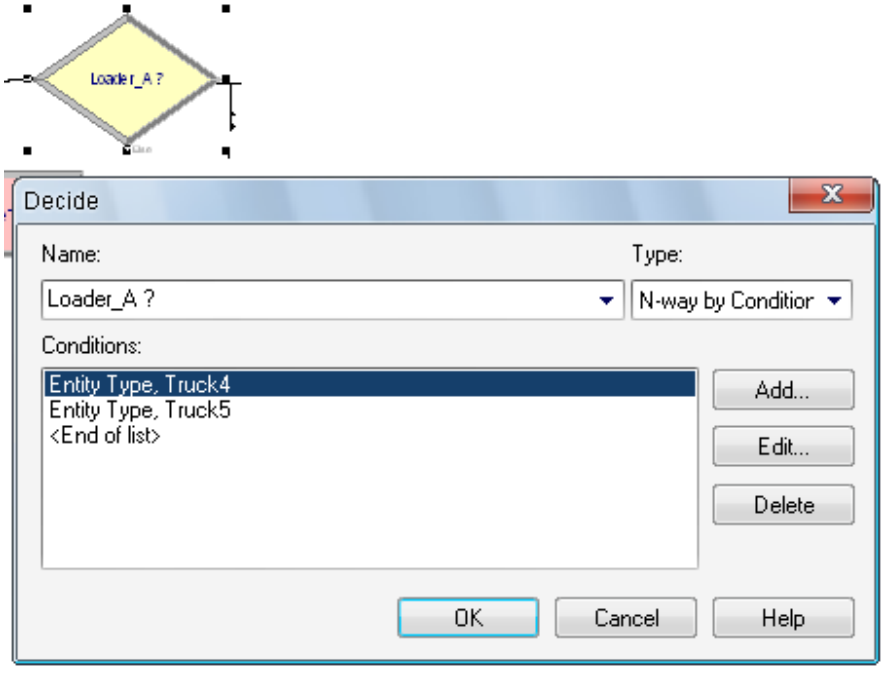
16. ใส่รายละเอียดของรถบรรทุกกลุ่มที่ 2

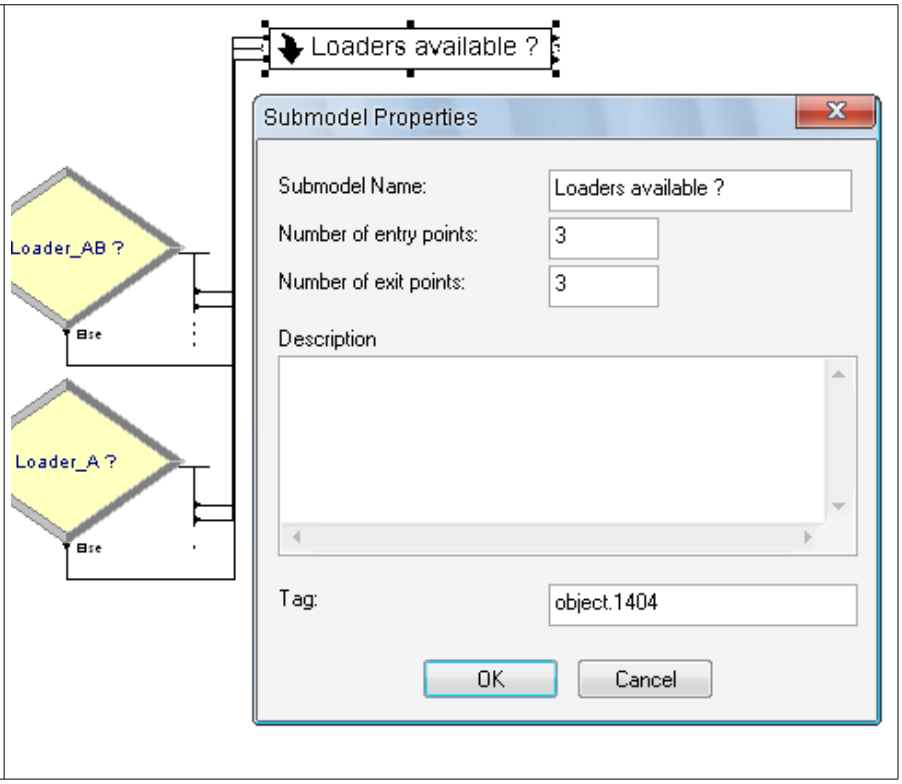


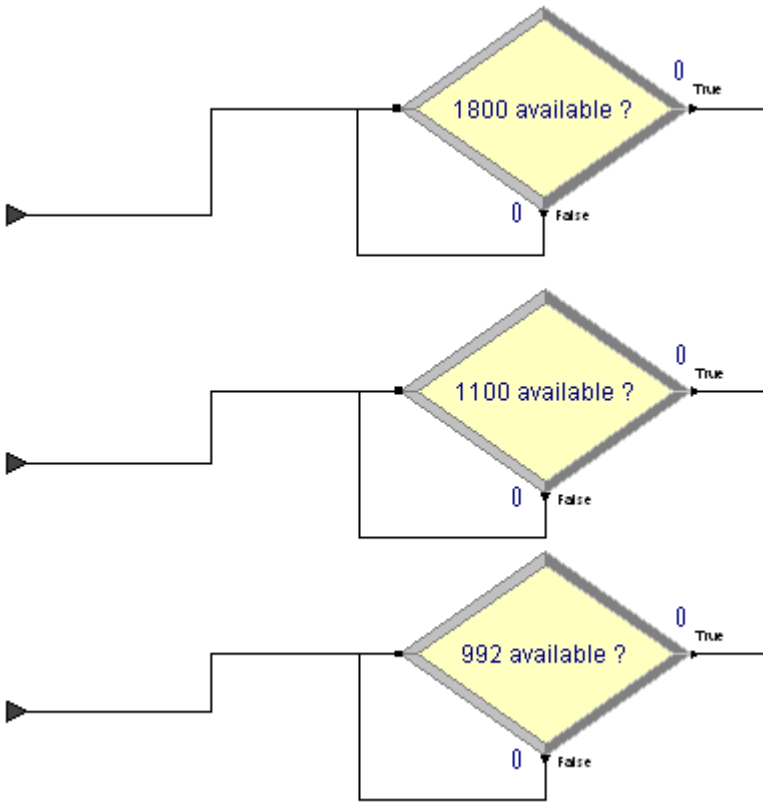
17. ใส่รายละเอียดของรถบรรทุกกลุ่มที่ 3

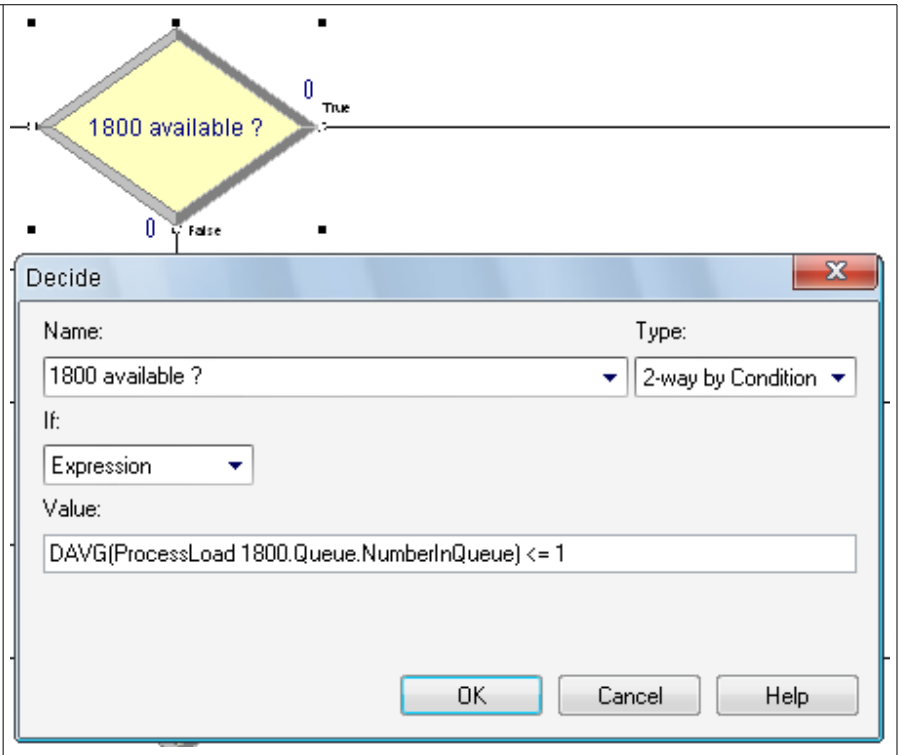


18. สร้าง Submodel	
19. สร้าง Route Module	

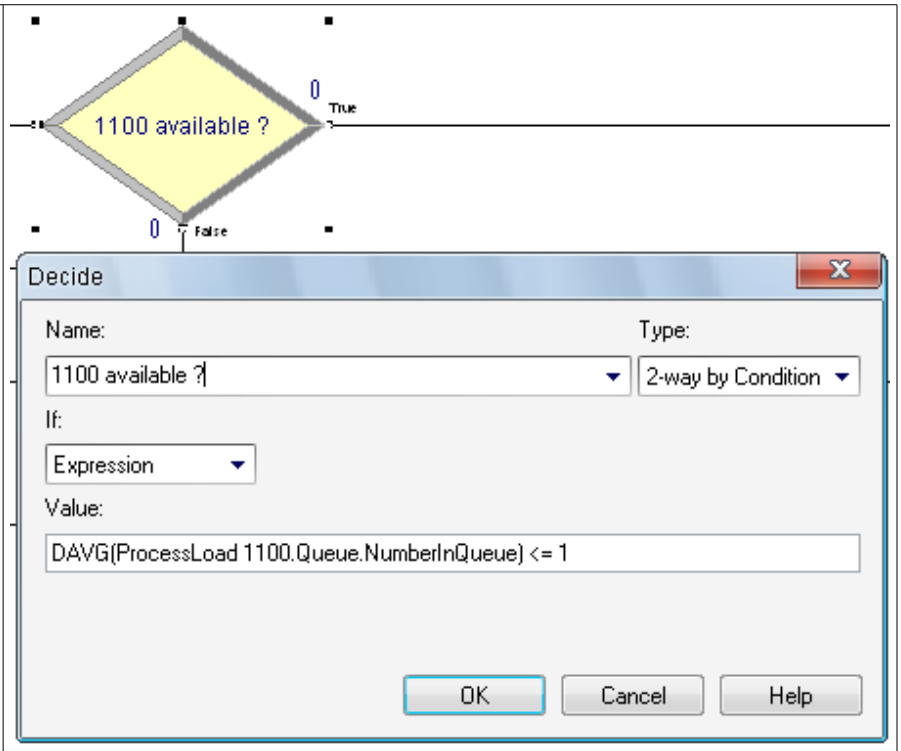
20. สร้าง Station Module	
21. สร้าง Decide Module	

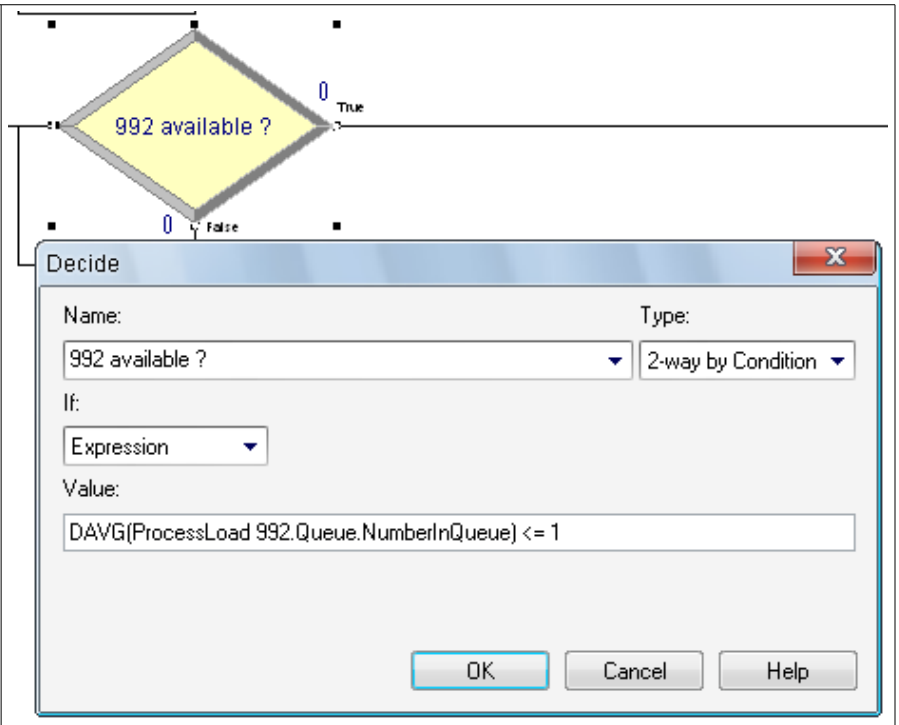
22. สร้าง Submodel	

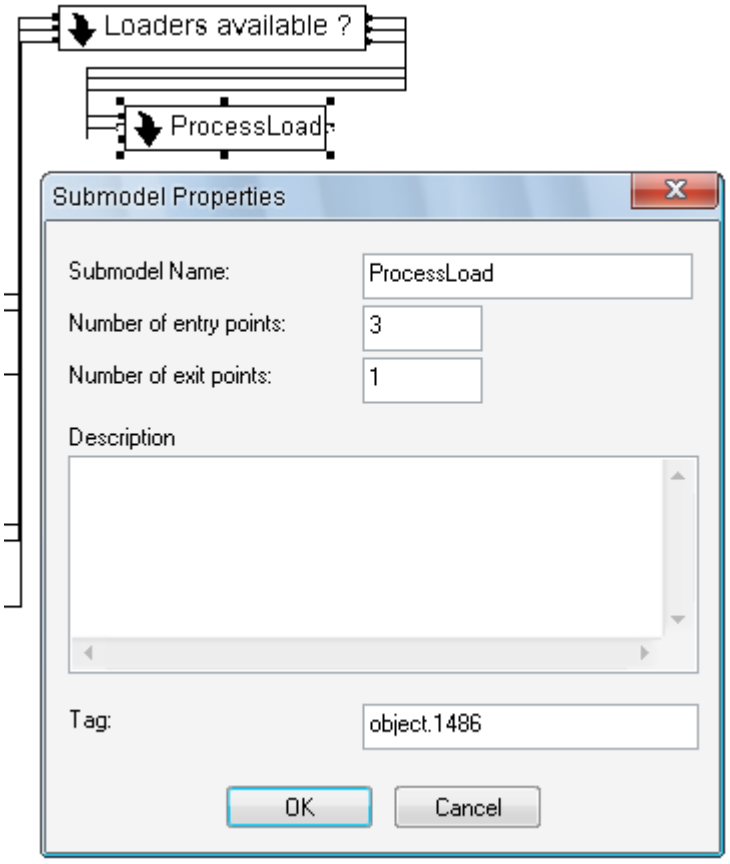
23. ใน Loaders available ? สร้าง Decide Module	

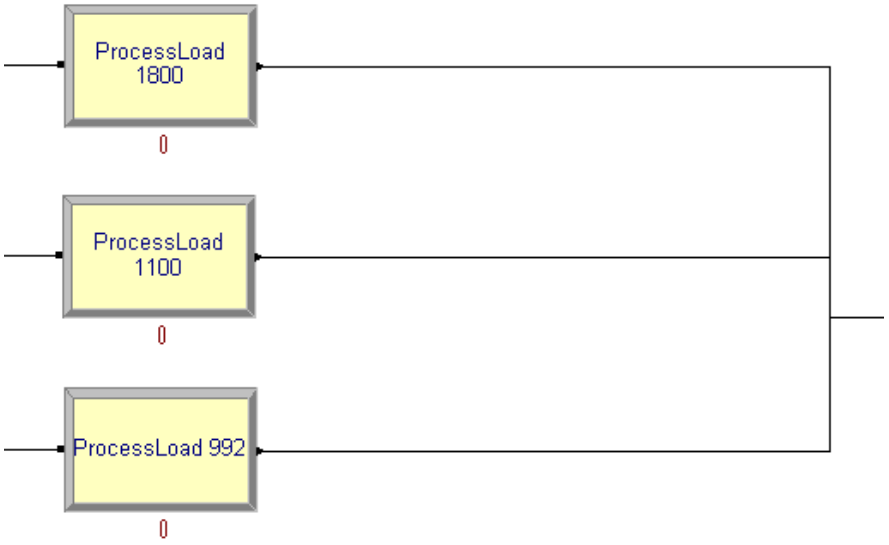
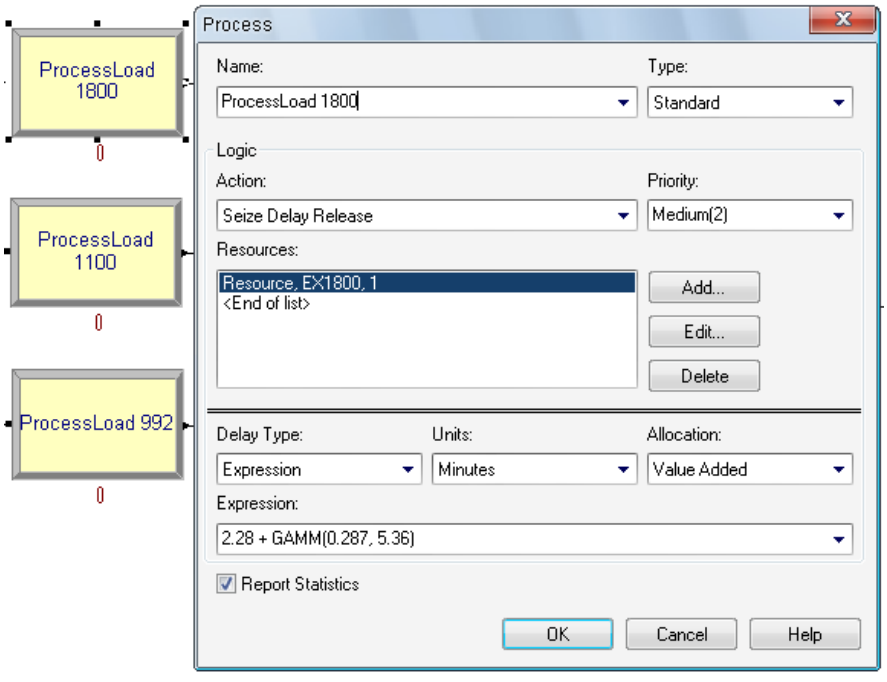
24. ใส่รายละเอียดของ 1800 available ?	

25. ใส่รายละเอียดของ 1100 available ?

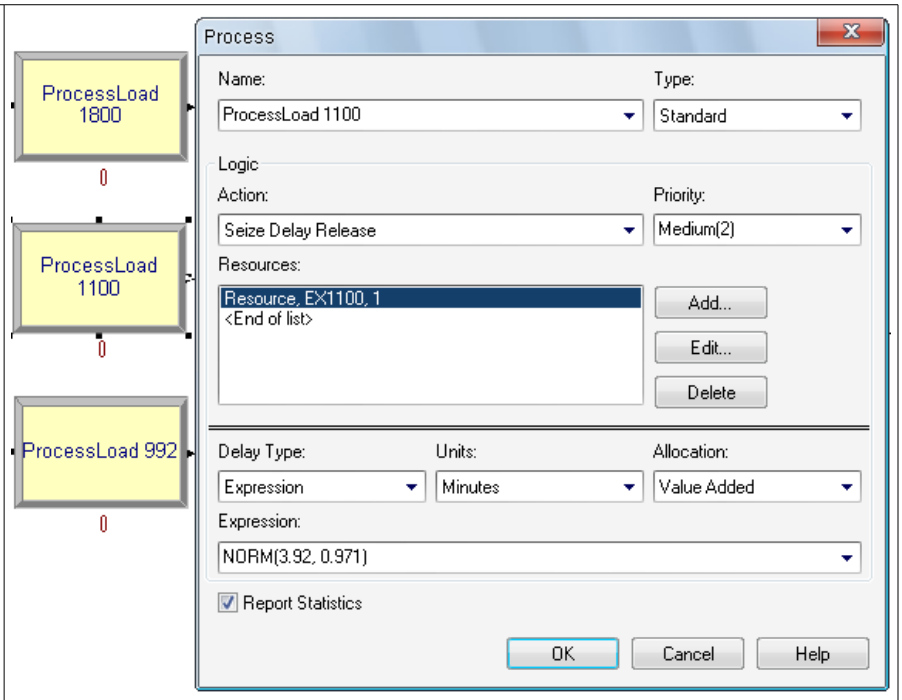


26. ใส่รายละเอียดของ 992 available ?	

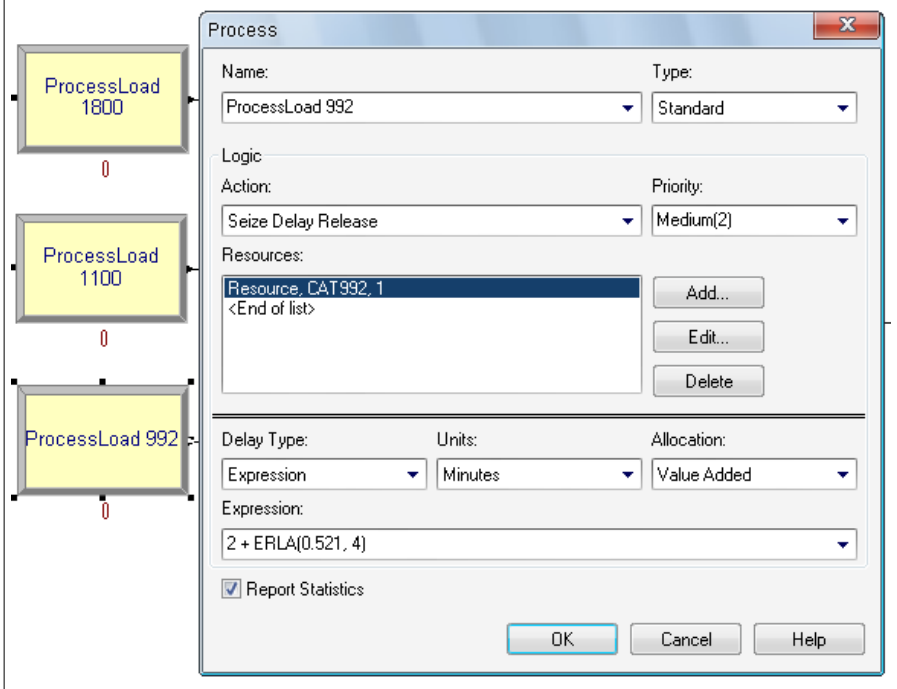
27. สร้าง Submodel	

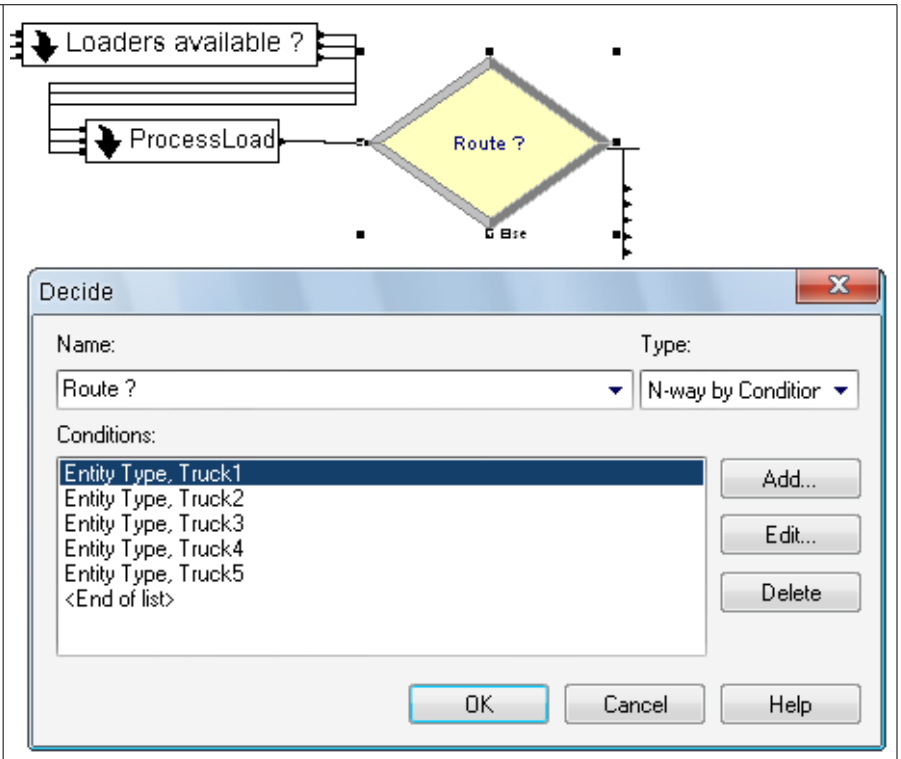
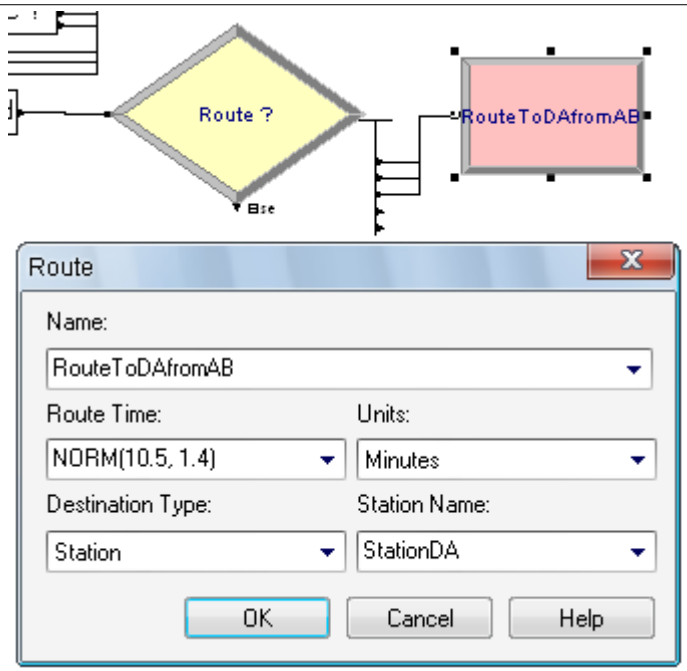
28. ใน ProcessLoad สร้าง Process Module	
29. ใส่รายละเอียดของ ProcessLoad 1800	

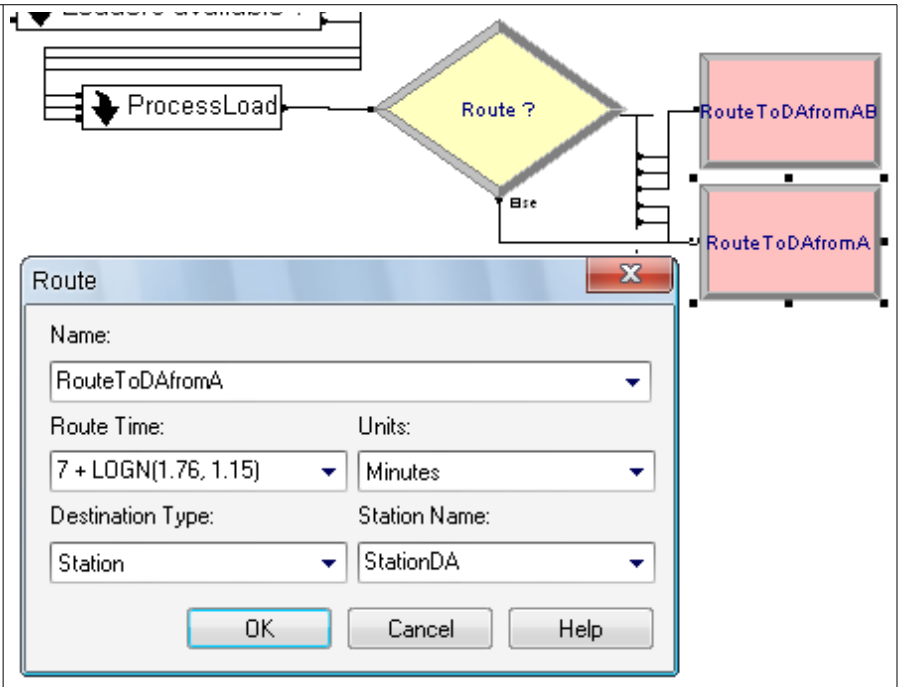
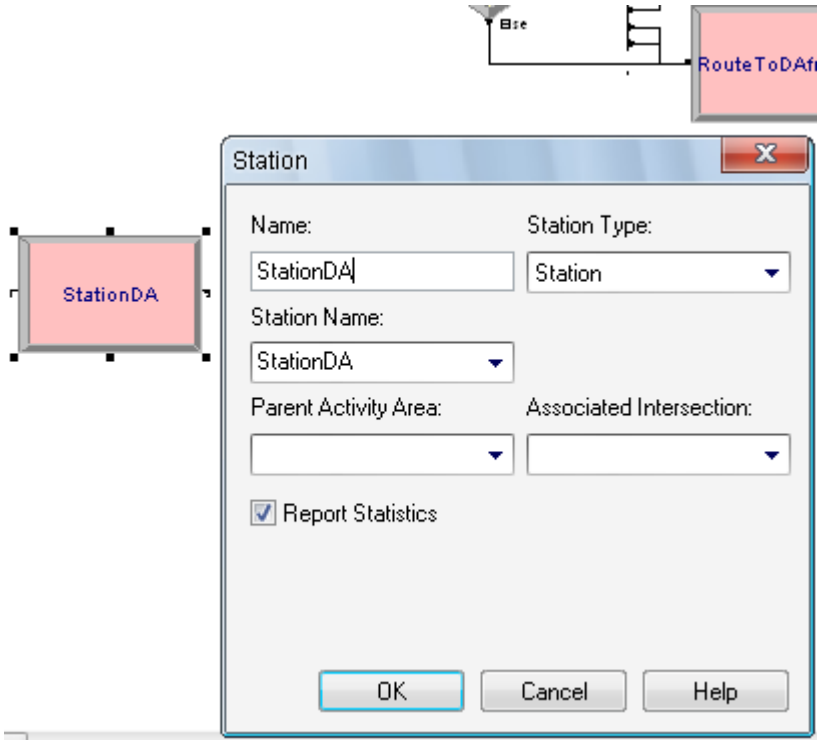
30. ใส่รายละเอียดของ ProcessLoad 1100

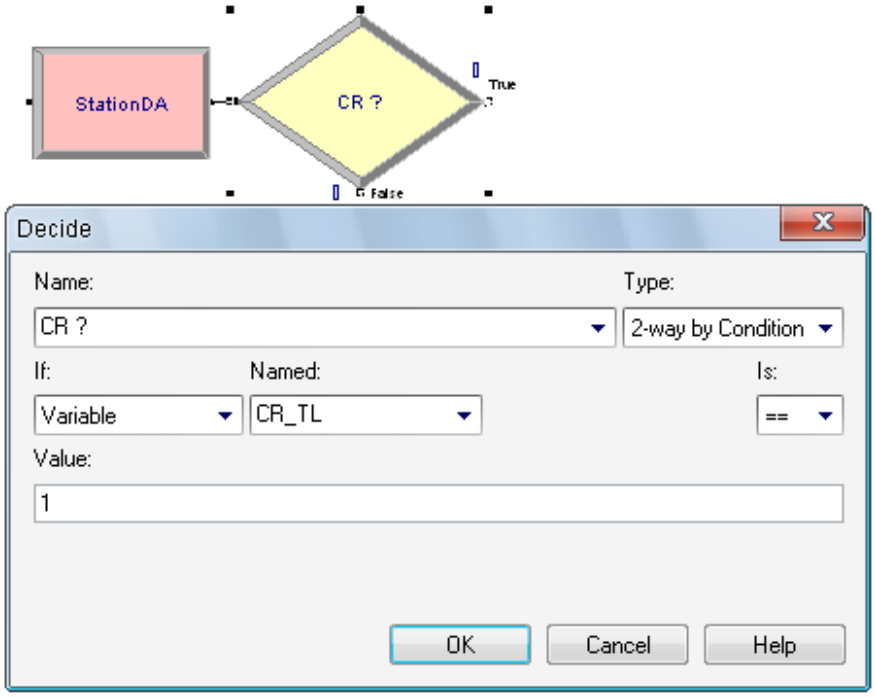


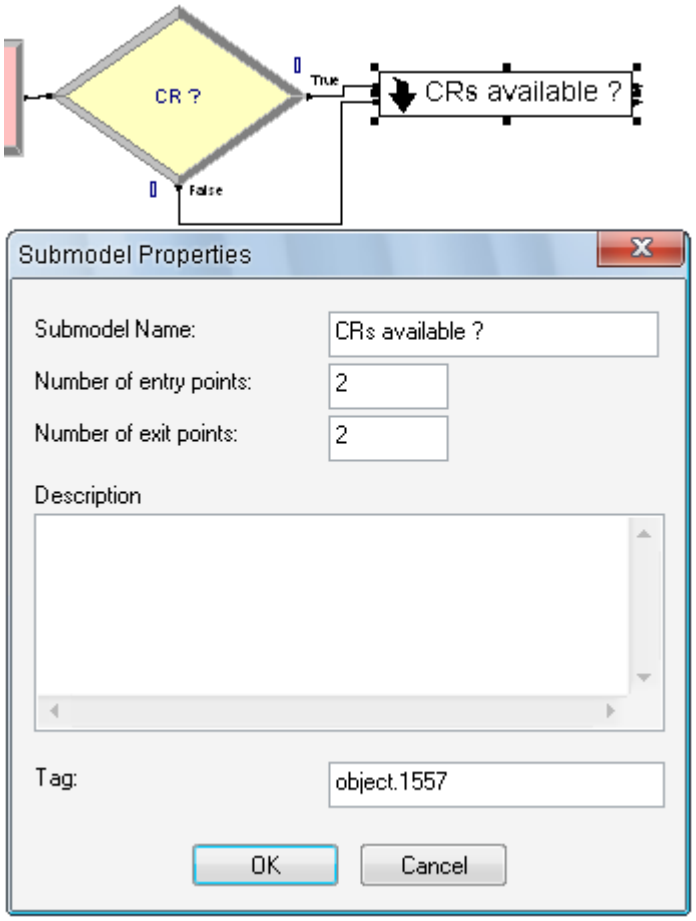
31. ใส่รายละเอียดของ ProcessLoad 992

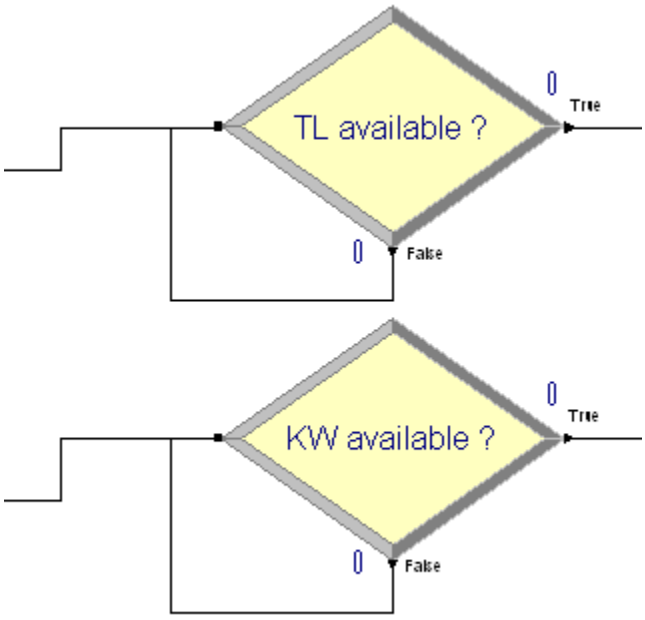


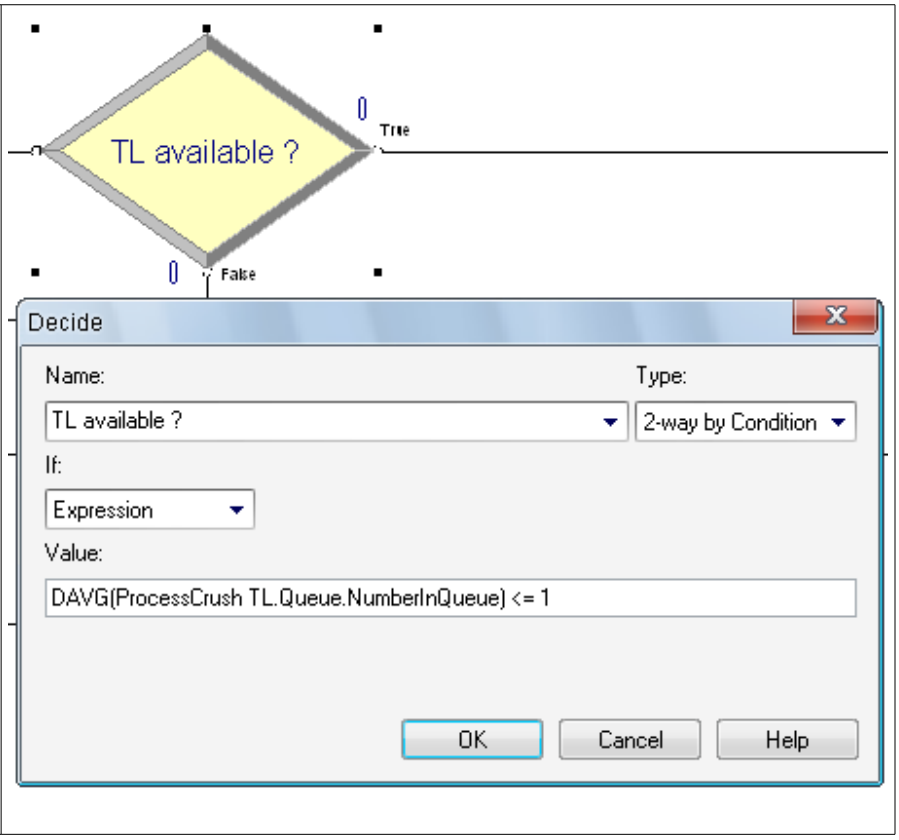
32. สร้าง Decide Module	
33. สร้าง Route Module สำหรับจุดตักที่ 1	

34. สร้าง Route Module สำหรับจุดตักที่ 2	
35. สร้าง Station Module	

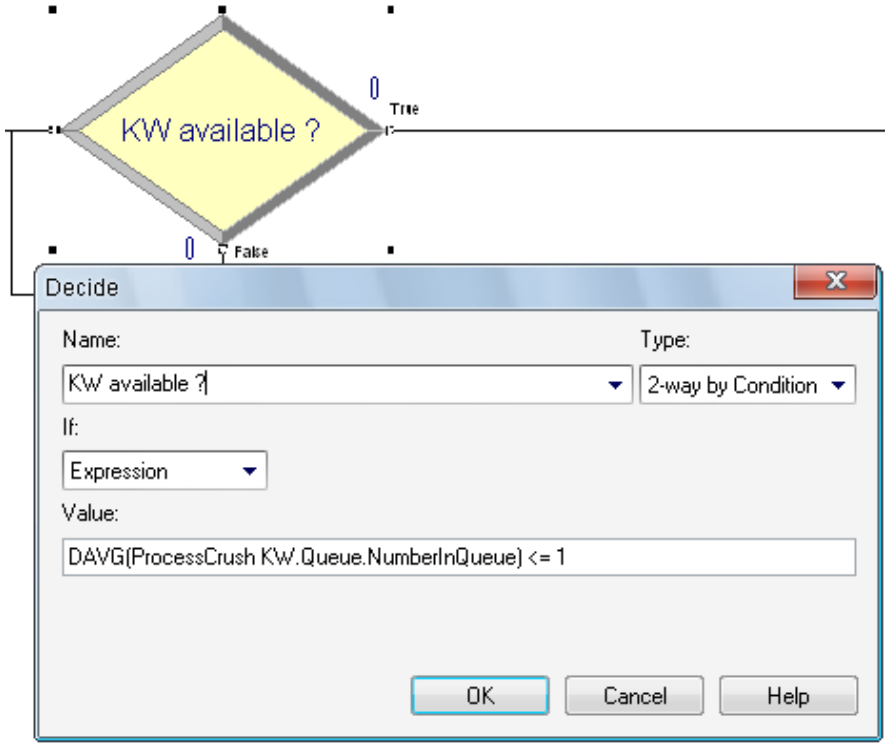
36. สร้าง Decide Module	

37. สร้าง Submodel	

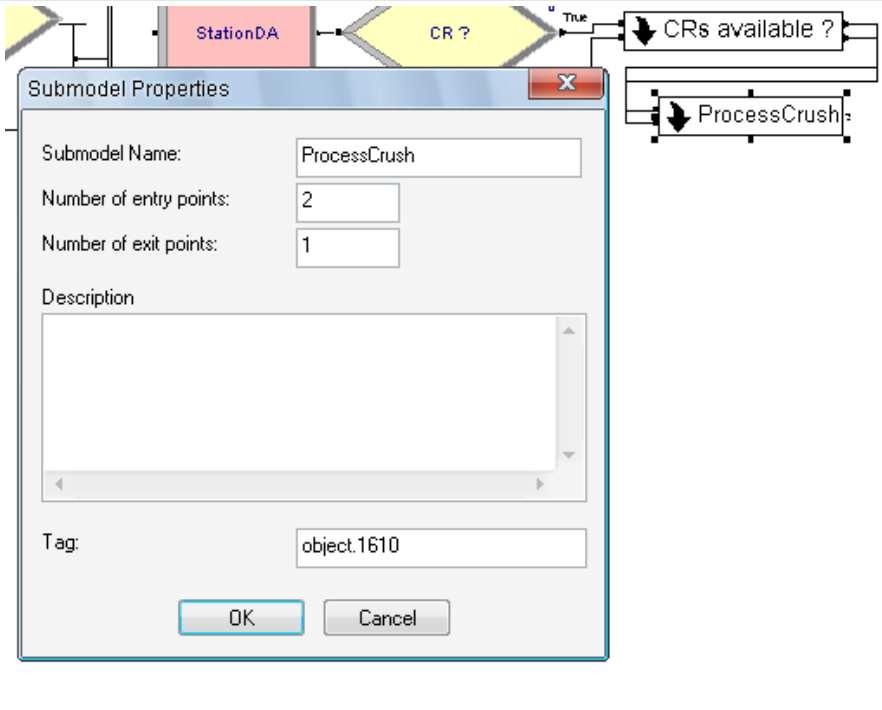
38. ใน CRs available ? สร้าง Decide Module	

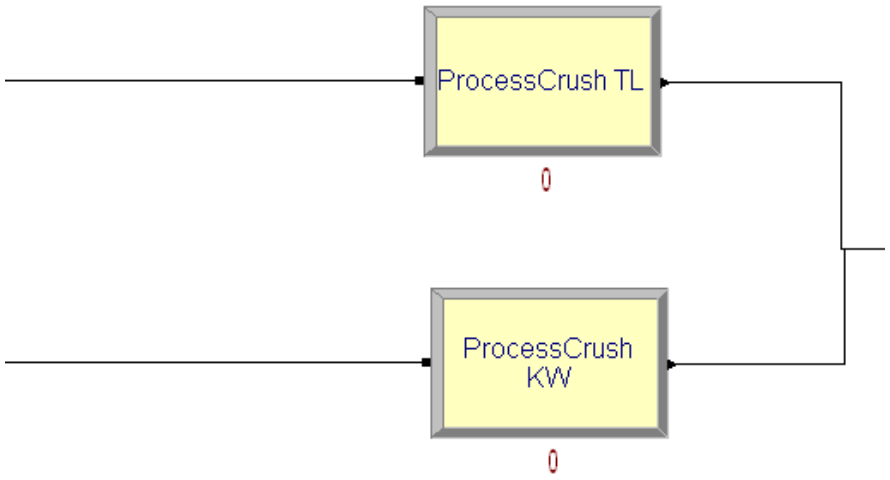
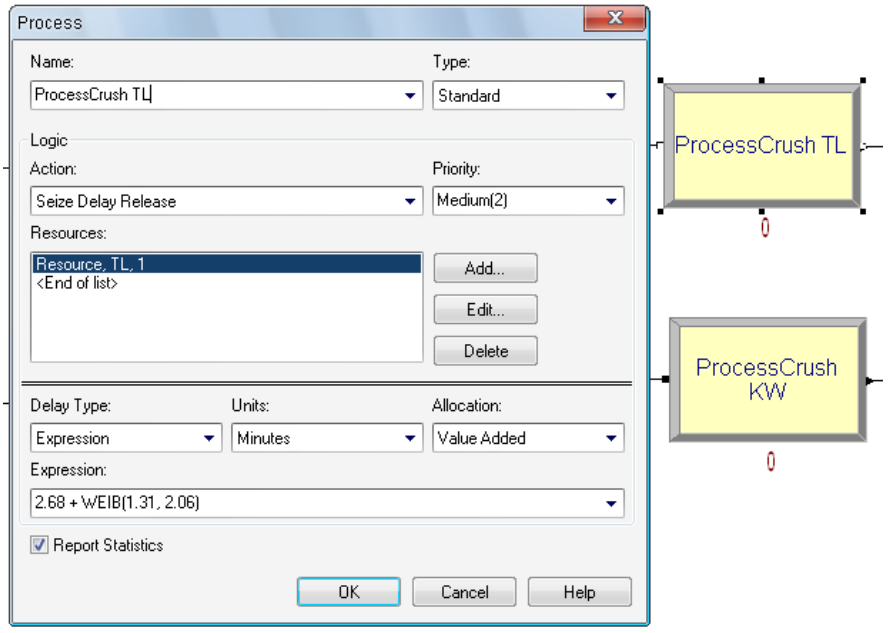
39. ใส่รายละเอียดของ TL available ?	

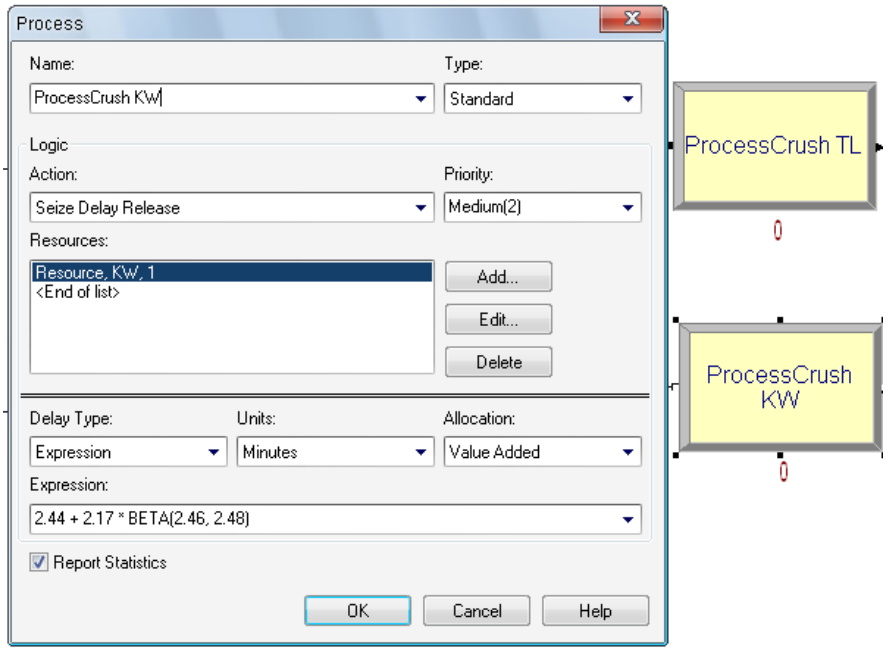
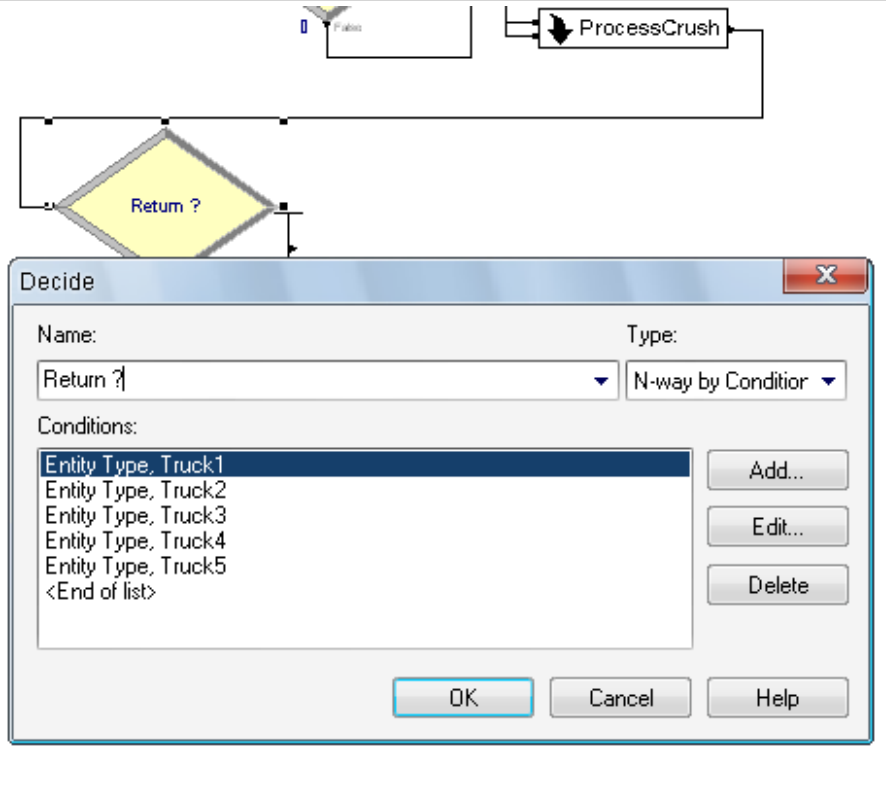
40. ใส่รายละเอียดของ KW available ?

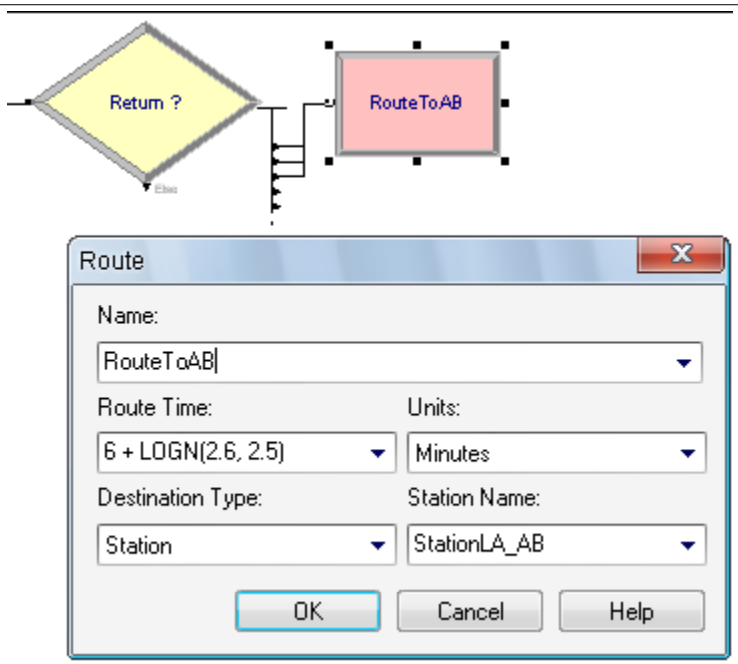
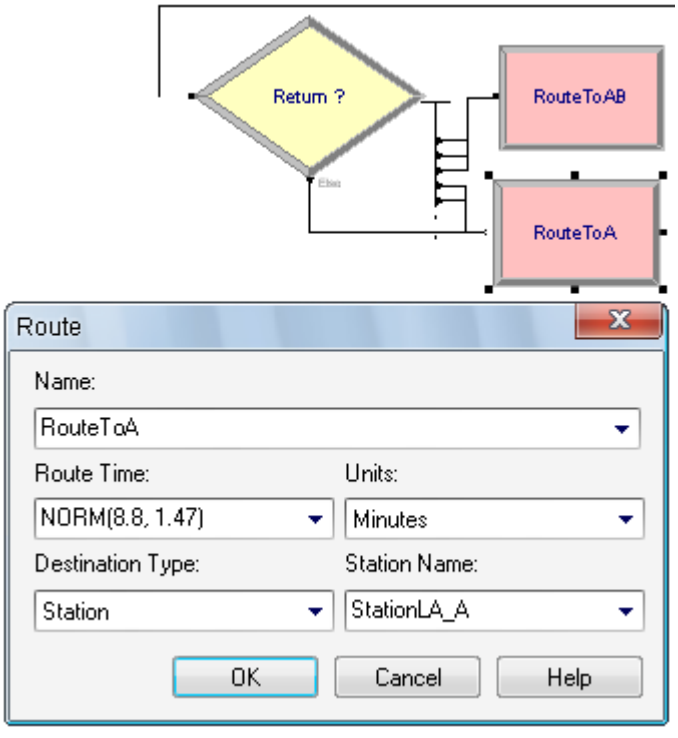


41. ออกจาก CRs available ? แล้วสร้าง Submodel



42. ใน ProcessCrush สร้าง Process Module	
43. ใส่รายละเอียดสำหรับ ProcessCrush TL	

44. ใส่รายละเอียดสำหรับ ProcessCrush KW	
45. ออกจาก ProcessCrush สร้าง Decide Module	

46. สร้าง Route Module สำหรับจุดตักที่ 1	
47. สร้าง Route Module สำหรับจุดตักที่ 2	

48. ทำตามขั้นตอนที่ 71. - 75. ของทุ่งสง

ภาคผนวก ข
โปรแกรมจาวา

รหัสต้นฉบับภาษาจาวา (Java source codes)

1. รหัสต้นฉบับรวมจุดตัก 3 จุด

```
import java.util.Scanner;
import static java.lang.System.out;

public class Assembly130611 {
	public static void main(String[] pond) {
		GetProcedures();
	}

	public static void GetProcedures() {
		Scanner T = new Scanner(System.in);
		out.println("\n\n\t\t ------------------ QUARRIES HAULAGE SYSTEM
SIMULATION ON UBUNTU
	------------------"+"\n\n\n\t-Loading Areas-");
		out.print("\t\tHow many loading areas ? : ");
		int noOfLA = T.nextInt();

		if (noOfLA == 1) {
			OneLA130611.GetProcedures();
		} else if (noOfLA == 2) {
			TwoLA130611.GetProcedures();
		} else if (noOfLA == 3) {
			ThreeLA130611.GetProcedures();
		} else {
			out.println("\n\t\tWarning ! Excessive number of loading
areas. Please, insert                          again.");
		}
	}
}
```

2. รหัสต้นฉบับจุดตัก 1 จุด

```
import java.util.*;
import java.text.*;
import static java.lang.System.out;

public class OneLA130611 {
	public static void main(String[] pond) {
		GetProcedures();
	}

	public static void GetProcedures() {
		Scanner T = new Scanner(System.in);

		int RVTruck1 = 0, HVTruck1 = 0, AmountTruck1 = 0, DumpCap = 0,
CapCR1_1 = 0, CapCR2_1 = 0;
```

```
            double StdDevLoadTimeLoader1_1LA = 0.0, MeanLoadTimeLoader1_1LA =
0.0, StdDevDumpTime1 =           0.0, MeanDumpTime1 = 0.0;
            String NameTruck1 = null, NameCR1_1 = null, NameCR2_1 = null;

            out.print("\n\t-Trucks-\n");
            out.print("\t\tName the truck : ");
            NameTruck1 = T.next();

            out.print("\t\tHow many "+NameTruck1+"s ? : ");
            AmountTruck1 = T.nextInt();

            out.print("\t\tDump capacity of "+NameTruck1+" (t) : ");
            DumpCap = T.nextInt();

            out.print("\t\tAverage haul velocity of "+NameTruck1+"(km/h) : ");
            HVTruck1 = T.nextInt();

            out.print("\t\tAverage return velocity of "+NameTruck1+"(km/h) :
");
            RVTruck1 = T.nextInt();

            out.println("\n\t\tDumping time of "+NameTruck1+" in term of
Normal Distribution.\n\t\t(If                    data not fit in Normal
Distribution, you must fit its first.\n\t\tSuggestion
      software : Arena 13, Input Analyzer)");

            out.print("\t\t\t1. Normal Distribution,
      insert \"N\"\n\t\t\t2. I don't know,
      insert \"-\"\n\t\t\t: ");
            String DistDumpTimeOfTruck1 = T.next();
if(DistDumpTimeOfTruck1.equalsIgnoreCase("n")){
                  out.print("\t\tNormal format -- NORM(Mean, StdDev)
--\n\t\t\tMean = ");
                  MeanDumpTime1 = T.nextDouble();
                  out.print("\t\t\tStdDev = ");
                  StdDevDumpTime1 = T.nextDouble();
            }else if(DistDumpTimeOfTruck1.equalsIgnoreCase("-")){
                  out.println("\t\tNormal default -- NORM(1.5, 0.5) --");
                  MeanDumpTime1 = 1.5;
                  StdDevDumpTime1 = .5;
            }

            out.print("\n\t-Loaders-\n\t\tName the loader : ");
            String NameLoader1_1LA = T.next();
            out.println("\t\tLoading Time of "+NameLoader1_1LA+" in term of
Normal Distribution.");
            out.print("\t\t\t1. Normal Distribution, insert \"N\"\n\t\t\t2. I
don't know,                                  insert \"-\"\n\t\t\t: ");
            String LoadTimeLoader1_1LA = T.next();

            if(LoadTimeLoader1_1LA.equalsIgnoreCase("n")){
                  out.print("\t\tNormal format -- NORM(Mean, StdDev)
--\n\t\t\tMean = ");
                  MeanLoadTimeLoader1_1LA = T.nextDouble();
                  out.print("\t\t\tStdDev = ");
                  StdDevLoadTimeLoader1_1LA = T.nextDouble();
            }else if(LoadTimeLoader1_1LA.equalsIgnoreCase("-")){
                  out.println("\t\tNormal default -- NORM(4.5, 0.8) --");
                  MeanLoadTimeLoader1_1LA = 4.5;
                  StdDevLoadTimeLoader1_1LA = .8;
            }
      out.print("\n\t-Crushers-\n");
            out.print("\t\tHow many crushers ? : ");
```

```
            int AmountCR1 = T.nextInt();

            if(AmountCR1 == 1){
                  out.print("\t\tName the crusher : ");
                  NameCR1_1 = T.next();
                  out.print("\t\tCapacity of "+NameCR1_1+" (t/h) : ");
                  CapCR1_1 = T.nextInt();
            }else if(AmountCR1 == 2){
                  out.print("\t\tName the 1st crusher : ");
                  NameCR1_1 = T.next();

                  out.print("\t\tCapacity of "+NameCR1_1+" (t/h) : ");
                  CapCR1_1 = T.nextInt();

                  out.print("\t\tName the 2nd crusher : ");
                  NameCR2_1 = T.next();

                  out.print("\t\tCapacity of "+NameCR2_1+" (t/h) : ");
                  CapCR2_1 = T.nextInt();

            }else{
                  out.println("\n\t\tWarning ! Excessive number of crushers.
      Please, insert again.");
            }

            out.print("\n\t\tDistance between dumping area and
"+NameLoader1_1LA+" (m.) : ");
            int DistanceDA_Loader1_1LA = T.nextInt();

            DateFormat StartTime_1LA1 = new SimpleDateFormat("HH:mm");
            Calendar startTime_1LA1 = Calendar.getInstance();
            DateFormat GoBack1For1stLA = new SimpleDateFormat("HH:mm");
            Calendar goBack1For1stLA = Calendar.getInstance();

            DateFormat StartRunIn8hrFor1stLA = new SimpleDateFormat("HH:mm");
            Calendar startRunIn8hrFor1stLA = Calendar.getInstance();

            Calendar finish8hr = Calendar.getInstance();
            finish8hr.add(Calendar.HOUR, 8);

            out.print("\n\t\tWould you like to view the system at loading
areas or dumping                                    areas ?\n\t\t\t1. Dumping
Areas, insert \"DA\"\n\t\t\t2. Loading Areas, insert
      \"LA\"\n\t\t\t: ");
            String ChooseView = T.next();

            if(ChooseView.equalsIgnoreCase("da")){
                  out.println("\n\n---------------------------- Start Time :
"+StartTime_1LA1.format(startTime_1LA1.getTime())
                        +"----------------------------\n");

                  out.println("OPERATION TIME\t\tTRUCK\t\t\tEVENT\n\n-->
Release 1st truck for each                                    loading
area");

                  if(AmountCR1 == 1){
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())
                        +"\t"+NameTruck1+"#1   \tDepart dumping area");

                        int returnTimeTo1stLA_1LA =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                        startTime_1LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_1LA);
```

```
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t
                        "+NameTruck1+"#1   \t"+NameLoader1_1LA+" start
load");

                        Random ran = new Random();
                        double LoadingTimeLoader1_1LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_1LA+
                              MeanLoadTimeLoader1_1LA);
                        startTime_1LA1.add(Calendar.MINUTE,     (int)
(LoadingTimeLoader1_1LA));
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t
                        "+NameTruck1+"#1   \t"+NameLoader1_1LA+" loaded");

                        int HaulingTimeLA1 = DistanceDA_Loader1_1LA*60/
      HVTruck1/1000;
                        startTime_1LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t
                        "+NameTruck1+"#1   \tstart dump at "+NameCR1_1);

                        double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_1LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t
                        "+NameTruck1+"#1   \tdumped at     "+NameCR1_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                        startTime_1LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t
                        "+NameTruck1+"#1   \tArrive "+NameLoader1_1LA+"
again");

                        out.println("\n--> To check, how many trucks
optimization ?");

                        int i;
                        for(i = 2; i <= AmountTruck1; i++){
                              goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_1LA);
                              out.println(""+GoBack1For1stLA.format(goBack1Fo
r1stLA.getTime())                                          +"\t   "+NameT
ruck1+"#"+i+"\tDepart dumping area");

                              int returnTimeLA1back =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                              goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);

                              out.println(""+GoBack1For1stLA.format(goBack1Fo
r1stLA.getTime())                                          +"\t
"+NameTruck1+"#"+i+"\t"+"Arrive "+NameLoader1_1LA);

                              if(goBack1For1stLA.after(startTime_1LA1)){
                                    if(goBack1For1stLA.after(startTime_1LA1)
){
                                          out.println("    --> The optimize
```

```
number of trucks = " + i);

                                                for (int j = 1; j <= i; j++) {
                                                      out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                                }

                                          } break;
                                    }
                              }

                              out.println("\n\n---------------------- Trucks status
in 8 hrs
      -----------------------\n\nOPERATION
                                    TIME\t\tTRUCK\t\t\tEVENT\n");

                              startRunIn8hrFor1stLA.add(Calendar.MINUTE,
                                                                              returnT
imeTo1stLA_1LA+int)LoadingTimeLoader1_1LA +
                        HaulingTimeLA1);

                              for(int k = 1;;k++){
                                    for(int l = 1; l <= i; l++){
                                          out.println("    "+
                                                                        StartRunIn8hrF
or1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                        "+NameTruck1+"#"+l+"   \tstart dump at "+NameCR1_1);
                                          DumpingTimeTruck1 =
                                                                                    (
ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E,
      (int)DumpingTimeTruck1);

                                          out.println("    "+
                                                                        StartRunIn8hrF
or1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                        "+NameTruck1+"#"+l+"   \tdumped at     "+NameCR1_1);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E, returnTimeTo1stLA_1LA+
      (int)LoadingTimeLoader1_1LA + HaulingTimeLA1);
                                    }
                                    if (startRunIn8hrFor1stLA.after(finish8hr))
{break;}
                              }
                        }else if(AmountCR1 == 2){
                              out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                              NameTruck1+"#1   \tDepart dumping area");

                              int returnTimeTo1stLA_1LA =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                              startTime_1LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_1LA);
                              out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                              NameTruck1+"#1   \t"+NameLoader1_1LA+" start load");

                              Random ran = new Random();
                              double LoadingTimeLoader1_1LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_1LA+
            MeanLoadTimeLoader1_1LA);
                              startTime_1LA1.add(Calendar.MINUTE,     (int)
(LoadingTimeLoader1_1LA));
```

```
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t
                    "+NameTruck1+"#1   \t"+NameLoader1_1LA+" loaded");

                        int HaulingTimeLA1 =
DistanceDA_Loader1_1LA*60/HVTruck1/1000;
                        startTime_1LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tstart dump at "+NameCR1_1);

                        double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_1LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tdumped at     "+NameCR1_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                        startTime_1LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t
       "+NameTruck1+"#1   \tArrive "+NameLoader1_1LA+" again");

                        out.println("\n--> To check, how many trucks
optimization ?");

                        int i;
                        for(i = 2; i <= AmountTruck1; i++){
                                goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_1LA);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                            "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA1back =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                                goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                            "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_1LA);

                                if(goBack1For1stLA.after(startTime_1LA1)){
                                       if(goBack1For1stLA.after(startTime_1LA1)
){
                                              out.println("    --> The optimize
number of trucks = "+i);

                                              for(int j = 1; j <= i; j++){
                                                     out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                              }

                                       }break;
                                }
                        }

                        out.println("\n\n---------------------- Trucks status
in 8 hrs
```

```
      ----------------------\n\nOPERATION    TIME\t\tTRUCK\t\t\tEVENT\n");

                           startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_1LA+
      (int)LoadingTimeLoader1_1LA+HaulingTimeLA1);

                           for(int k = 1;; k++){
                                 for(int l = 1; l <= i; l++){

                                       out.println("    "+
                                       StartRunIn8hrFor1stLA.format(startRunIn8
hrFor1stLA.getTime())+"\t
      "+NameTruck1+"#"+l+"   \tstart dump at "+NameCR1_1);
                                       DumpingTimeTruck1 =

      (ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                       startRunIn8hrFor1stLA.add(Calendar.MINUT
E,
                  (int)DumpingTimeTruck1);
                                       out.println("    "+
                                                              StartRunIn8hrF
or1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                  "+NameTruck1+"#"+l+"   \tdumped at     "+NameCR1_1);
                                       startRunIn8hrFor1stLA.add(Calendar.MINUT
E, returnTimeTo1stLA_1LA+
      (int)LoadingTimeLoader1_1LA + HaulingTimeLA1);
                                       double randForChooseCR =
                                                                              r
an.nextGaussian()*StdDevDumpTime1+MeanDumpTime1;

                                       if(randForChooseCR >= 1.5){
                                             out.println("    "+
                                                                    StartRu
nIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                 "+NameTruck1+"#"+l+"   \tstart dump at
"+NameCR2_1);
                                             DumpingTimeTruck1 =

      (ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                             startRunIn8hrFor1stLA.add(Calendar
.MINUTE,
                                 (int)DumpingTimeTruck1);
                                             out.println("    "+
                                                StartRunIn8hrF
or1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                           "+NameTruck1+"#"+l+"   \tdumped at     "+NameCR2_1);
                                             startRunIn8hrFor1stLA.add(Calendar
.MINUTE,
      returnTimeTo1stLA_1LA+(int)LoadingTimeLoader1_1LA +
                                 HaulingTimeLA1);
                                       }

                                 }
                                 if(startRunIn8hrFor1stLA.after(finish8hr))
{break;}
                           }
                     }
               }else if(ChooseView.equalsIgnoreCase("la")){
                     out.println("\n\n---------------------------- Start Time : "+
                                                                    StartTime_1LA1
.format(startTime_1LA1.getTime())+" ----------------------------\n");
                     out.println("OPERATION TIME\t\tTRUCK\t\t\tEVENT\n\n-->
Release 1st truck for each loading                                  area");
```

```
                    if(AmountCR1 == 1){
                            out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                            NameTruck1+"#1   \tDepart dumping area");

                                    int returnTimeTo1stLA_1LA =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                                    startTime_1LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_1LA);
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                            NameTruck1+"#1   \t"+NameLoader1_1LA+" start load");

                                    returnTimeTo1stLA_1LA =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                                    startTime_1LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_1LA);
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                            NameTruck1+"#2   \t"+NameLoader1_1LA+" start load");

                                    Random ran = new Random();
                                    double LoadingTimeLoader1_1LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_1LA+
             MeanLoadTimeLoader1_1LA);
                                    startTime_1LA1.add(Calendar.MINUTE,     (int)
(LoadingTimeLoader1_1LA));
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                            NameTruck1+"#1   \t"+NameLoader1_1LA+" loaded");

                                    int HaulingTimeLA1 =
DistanceDA_Loader1_1LA*60/HVTruck1/1000;
                                    startTime_1LA1.add(Calendar.MINUTE,
HaulingTimeLA1);
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                            NameTruck1+"#1   \tstart dump at "+NameCR1_1);

                                    double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                    startTime_1LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                            NameTruck1+"#1   \tdumped at     "+NameCR1_1);

                                    int ReturnTimeToLA1again =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                                    startTime_1LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                            NameTruck1+"#1   \tArrive "+NameLoader1_1LA+"
again");

                                    out.println("\n--> To check, how many trucks
optimization ?");
                                    int i;

                                    for(i = 2; i <= AmountTruck1; i++){

                                           goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_1LA);
```

```
                                        out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                          "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                        int returnTimeLA1back =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                                        goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                                        out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                          "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_1LA);

                                        if(goBack1For1stLA.after(startTime_1LA1)
){
                                               if(goBack1For1stLA.after(startTime
_1LA1)){
                                                      out.println("    --> The
optimize number of trucks = "+i);

                                                      for(int j = 1; j <= i; j++)
{
                          out.println("\t\t\t\t\t"+NameTruck1+"#"+j);
                                                                  }
                                               }break;
                                        }

                                  }

                                  out.println("\n\n---------------------- Trucks
status in 8 hrs
      ----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                                  startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_1LA);

                                  for(int k = 1;; k++){

                                        for(int l = 1; l <= i; l++){
                                               out.println("    "+

                    StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime()
)+"\t                                                  "+NameTruck1+"#"+l+"
\tstart load at "+NameLoader1_1LA);
                                               LoadingTimeLoader1_1LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_1LA+
                    MeanLoadTimeLoader1_1LA);
                                               startRunIn8hrFor1stLA.add(Calendar
.MINUTE,
              (int)LoadingTimeLoader1_1LA);
                                               out.println("    "+

                    StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime()
)+"\t                                                  "+NameTruck1+"#"+l+"
\tloaded at     "+NameLoader1_1LA);

                                               double LoopForLA_1stLA_Truck1 =

              (DistanceDA_Loader1_1LA*60/HVTruck1/1000)+
                                                      (ran.nextGaussian()*StdDevD
umpTime1+MeanDumpTime1)+
              (DistanceDA_Loader1_1LA*60/RVTruck1/1000);
                                               startRunIn8hrFor1stLA.add(Calendar
.MINUTE,(int)LoopForLA_1stLA_Truck1);                                    }
```

```
                                        if(startRunIn8hrFor1stLA.after(finish8hr
)){break;}
                                    }
                        }else if(AmountCR1 == 2){
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                                    NameTruck1+"#1   \tDepart dumping area");

                                    int returnTimeTo1stLA_1LA =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                                    startTime_1LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_1LA);
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                                    NameTruck1+"#1   \t"+NameLoader1_1LA+" start load");

                                    Random ran = new Random();
                                    double LoadingTimeLoader1_1LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_1LA+
                MeanLoadTimeLoader1_1LA);
                                    startTime_1LA1.add(Calendar.MINUTE,     (int)
(LoadingTimeLoader1_1LA));
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                                    NameTruck1+"#1   \t"+NameLoader1_1LA+" loaded");

                                    int HaulingTimeLA1 =
DistanceDA_Loader1_1LA*60/HVTruck1/1000;
                                    startTime_1LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                                    NameTruck1+"#1   \tstart dump at "+NameCR1_1);

                                    double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                    startTime_1LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                                    NameTruck1+"#1   \tdumped at     "+NameCR1_1);

                                    int ReturnTimeToLA1again =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
                                    startTime_1LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                                    out.println("
"+StartTime_1LA1.format(startTime_1LA1.getTime())+"\t      "+
                                    NameTruck1+"#1   \tArrive "+NameLoader1_1LA+"
again");

                                    out.println("\n--> To check, how many trucks
optimization ?");

                                    int i;
                                    for(i = 2; i <= AmountTruck1; i++){

                                        goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_1LA);
                                        out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                                    "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                        int returnTimeLA1back =
DistanceDA_Loader1_1LA*60/RVTruck1/1000;
```

```
                                    goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                                    out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                    "+NameTruck1+"#"+i+"   \t"+"Arrive "+NameLoader1_1LA);

                                    if(goBack1For1stLA.after(startTime_1LA1)){
                                          if(goBack1For1stLA.after(startTime_1LA1)
){
                                                out.println("    --> The optimize
number of trucks = "+i);

                                                for(int j = 1; j <= i; j++){

                    out.println("\t\t\t\t\t"+NameTruck1+"#"+j);
                                                      }
                                          }break;

                                          }
                              }

                              out.println("\n\n---------------------- Trucks status
in 8 hrs
      -----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                              startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_1LA);

                              for(int k = 1;; k++){

                                    for(int l = 1; l <= i; l++){
                                          out.println("    "+

             StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                                "+NameTruck1+"#"+l+"   \tstart
load at "+NameLoader1_1LA);
                                          LoadingTimeLoader1_1LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_1LA+
                    MeanLoadTimeLoader1_1LA);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E, (int)LoadingTimeLoader1_1LA);
                                          out.println("    "+

             StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                                "+NameTruck1+"#"+l+"   \tloaded at
"+NameLoader1_1LA);

                                          double LoopForLA_1stLA_Truck1 =
(DistanceDA_Loader1_1LA*60/HVTruck1/1000)+
             (ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1)+
                                                      (DistanceDA_Loader1_1LA*60/
RVTruck1/1000);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E,(int)LoopForLA_1stLA_Truck1);
                                    }

                                    if(startRunIn8hrFor1stLA.after(finish8hr))
{break;}
                              }
                    }else{
                              out.println("\n\t\tWarning ! Incorrect insert.
Please, insert again.");
                    }
             }
       }
```

```
}
```

3. รหัสต้นฉบับจุดตัก 2 จุด

```
import java.util.*;
import java.text.*;
import static java.lang.System.out;
import java.lang.String.*;

public class TwoLA130611{
	public static void main(String[] pond){
		GetProcedures();
	}
	public static void GetProcedures(){
		Scanner T = new Scanner(System.in);

		int RVTruck1 = 0, HVTruck1 = 0, AmountTruck1 = 0, DumpCapTruck1 =
0, CapCR2_1 = 0, CapCR2_2 = 0;
		double StdDevLoadTimeLoader1_2LA = 0.0, StdDevLoadTimeLoader2_2LA
= 0.0, MeanLoadTimeLoader1_2LA =                    0.0,
MeanLoadTimeLoader2_2LA = 0.0, StdDevDumpTime1 = 0.0, MeanDumpTime1 = 0.0;
		String NameTruck1 = null, NameCR2_1 = null, NameCR2_2 = null;

		out.print("\n\t-Trucks-\n");
		out.print("\t\tName the truck : ");
		NameTruck1 = T.next();
		out.print("\t\tHow many "+NameTruck1 + "s ? : ");
		AmountTruck1 = T.nextInt();
		out.print("\t\tDump capacity of "+NameTruck1+" (t) : ");
		DumpCapTruck1 = T.nextInt();
		out.print("\t\tAverage haul velocity of "+NameTruck1+" (km/h) :
");
		HVTruck1 = T.nextInt();
		out.print("\t\tAverage return velocity of "+NameTruck1+" (km/h) :
");
		RVTruck1 = T.nextInt();
		out.println("\n\t\tDumping time of "+NameTruck1+" in term of
Normal Distribution.\n\t\t(If data not               fit in Normal Distribution,
you must fit its first.\n\t\t Suggestion software : Arena 13, Input
	Analyzer)");
		out.print("\t\t\t1. Normal Distribution, insert \"N\"\n\t\t\t2. I
don't know, insert \"-\"\n\t\t\t:                          ");
		String DistDumpTimeOfTruck1 = T.next();

		if(DistDumpTimeOfTruck1.equalsIgnoreCase("n")){
			out.print("\t\tNormal format -- NORM(Mean, StdDev)
--\n\t\t\tMean   = ");
			MeanDumpTime1 = T.nextDouble();
			out.print("\t\t\tStdDev = ");
			StdDevDumpTime1 = T.nextDouble();
		}else if(DistDumpTimeOfTruck1.equalsIgnoreCase("-")){
			out.println("\t\tNormal default -- NORM(1.5, 0.5) --");
			MeanDumpTime1 = 1.5;
			StdDevDumpTime1 = .5;
		}

		out.print("\n\t-Loaders-\n\t\tName the loader at 1st loading
area : ");
		String NameLoader1_2LA = T.next();
```

```
            out.println("\t\tLoading Time of "+NameLoader1_2LA+" in term of
Normal Distribution.");
            out.print("\t\t\t1. Normal Distribution, insert \"N\"\n\t\t\t2. I
don't know, insert \"-\"\n\t\t\t:                          ");
            String LoadTimeLoader1_2LA = T.next();

            if(LoadTimeLoader1_2LA.equalsIgnoreCase("n")){
                  out.print("\t\tNormal format -- NORM(Mean, StdDev)
--\n\t\t\tMean   = ");
                  MeanLoadTimeLoader1_2LA = T.nextDouble();
                  out.print("\t\t\tStdDev = ");
                  StdDevLoadTimeLoader1_2LA = T.nextDouble();
            }else if(LoadTimeLoader1_2LA.equalsIgnoreCase("-")){
                  out.println("\t\tNormal default -- NORM(4.5, 0.8) --");
                  MeanLoadTimeLoader1_2LA = 4.5;
                  StdDevLoadTimeLoader1_2LA = .8;
            }

            out.print("\n\t\tName the loader at 2nd loading area : ");
            String NameLoader2_2LA = T.next();
            out.println("\t\tLoading Time of "+NameLoader2_2LA+" in term of
Normal Distribution.");
            out.print("\t\t\t1. Normal Distribution, insert \"N\"\n\t\t\t2. I
don't know, insert \"-\"\n\t\t\t:                          ");
            String LoadTimeLoader2_2LA = T.next();

            if(LoadTimeLoader2_2LA.equalsIgnoreCase("n")){
                  out.print("\t\tNormal format -- NORM(Mean, StdDev)
--\n\t\t\tMean   = ");
                  MeanLoadTimeLoader2_2LA = T.nextDouble();
                  out.print("\t\t\tStdDev = ");
                  StdDevLoadTimeLoader2_2LA = T.nextDouble();
            }else if(LoadTimeLoader2_2LA.equalsIgnoreCase("-")){
                  out.println("\t\tNormal default -- NORM(4.5, 0.8) --");
                  MeanLoadTimeLoader2_2LA = 4.5;
                  StdDevLoadTimeLoader2_2LA = .8;
            }

            out.print("\n\t-Crushers-\n");
            out.print("\t\tHow many crushers ? : ");
            int AmountCR2 = T.nextInt();

            if(AmountCR2 == 1){
                  out.print("\t\tName the crusher : ");
                  NameCR2_1 = T.next();
                  out.print("\t\tCapacity of "+NameCR2_1+" (t/h) : ");
                  CapCR2_1 = T.nextInt();
            }else if(AmountCR2 == 2){
                  out.print("\t\tName the 1st crusher : ");
                  NameCR2_1 = T.next();
                  out.print("\t\tCapacity of "+NameCR2_1+" (t/h) : ");
                  CapCR2_1 = T.nextInt();
                  out.print("\t\tName the 2nd crusher : ");
                  NameCR2_2 = T.next();
                  out.print("\t\tCapacity of "+NameCR2_2+" (t/h) : ");
                  CapCR2_2 = T.nextInt();
            }else{
                  out.println("\n\t\tWarning ! Excessive number of crushers.
Please, insert again.");
            }

            out.print("\n\t\tDistance between dumping area and
"+NameLoader1_2LA+" (m.) : ");
            int DistanceDA_Loader1_2LA = T.nextInt();
```

```
            out.print("\n\t\tDistance between dumping area and
"+NameLoader2_2LA+" (m.) : ");
            int DistanceDA_Loader2_2LA = T.nextInt();

            DateFormat StartTime_2LA1 = new SimpleDateFormat("HH:mm");
            Calendar startTime_2LA1 = Calendar.getInstance();
            DateFormat StartTime_2LA2 = new SimpleDateFormat("HH:mm");
            Calendar startTime_2LA2 = Calendar.getInstance();

            DateFormat GoBack1For1stLA = new SimpleDateFormat("HH:mm");
            Calendar goBack1For1stLA = Calendar.getInstance();

            DateFormat GoBack1For2ndLA = new SimpleDateFormat("HH:mm");
            Calendar goBack1For2ndLA = Calendar.getInstance();

            DateFormat StartRunIn8hrFor1stLA = new SimpleDateFormat("HH:mm");
            Calendar startRunIn8hrFor1stLA = Calendar.getInstance();
            DateFormat StartRunIn8hrFor2ndLA = new SimpleDateFormat("HH:mm");
            Calendar startRunIn8hrFor2ndLA = Calendar.getInstance();

            Calendar finish8hr = Calendar.getInstance();
            finish8hr.add(Calendar.HOUR, 8);

            out.print("\n\t\tWould you like to view the system at loading
areas or dumping areas ?\n\t\t\t1.                          Dumping Areas,
insert \"DA\"\n\t\t\t2. Loading Areas, insert \"LA\"\n\t\t\t: ");
            String ChooseView = T.next();

            if(ChooseView.equalsIgnoreCase("da")){
                  out.println("\n\n--------------------------- Start Time : "+
                        StartTime_2LA1.format(startTime_2LA1.getTime())+"
----------------------------\n");
                  out.println("OPERATION TIME\t\tTRUCK\t\t\tEVENT\n\n-->
Release 1st truck for each loading                                    area");

                  if(AmountCR2 == 1){
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tDepart dumping area");
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tDepart dumping area");

                        int returnTimeTo1stLA_2LA =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                        startTime_2LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_2LA);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                  +"#1   \t"+NameLoader1_2LA+" start load");

                        int returnTimeTo2ndLA_2LA =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                        startTime_2LA2.add(Calendar.MINUTE,
returnTimeTo2ndLA_2LA);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                  +"#2   \t"+NameLoader2_2LA+" start load");

                        Random ran = new Random();
                        double LoadingTimeLoader1_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_2LA+
            MeanLoadTimeLoader1_2LA);
                        startTime_2LA1.add(Calendar.MINUTE,     (int)
```

```
(LoadingTimeLoader1_2LA));
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_2LA+" loaded");

                        double LoadingTimeLoader2_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_2LA+
             MeanLoadTimeLoader2_2LA);
                        startTime_2LA2.add(Calendar.MINUTE,     (int)LoadingTi
meLoader2_2LA);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_2LA+" loaded");

                        int HaulingTimeLA1 =
DistanceDA_Loader1_2LA*60/HVTruck1/1000;
                        startTime_2LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tstart dump at "+NameCR2_1);

                        int HaulingTimeLA2 =
DistanceDA_Loader2_2LA*60/HVTruck1/1000;
                        startTime_2LA2.add(Calendar.MINUTE, HaulingTimeLA2);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tstart dump at "+NameCR2_1);

                        double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_2LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tdumped at     "+NameCR2_1);

                        double DumpingTimeTruck2 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_2LA2.add(Calendar.MINUTE,
(int)DumpingTimeTruck2);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tdumped at     "+NameCR2_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                        startTime_2LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \tArrive "+NameLoader1_2LA+" again");

                        int ReturnTimeToLA2again =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                        startTime_2LA2.add(Calendar.MINUTE,
ReturnTimeToLA2again);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \tArrive "+NameLoader2_2LA+" again");

                        out.println("\n--> To check, how many trucks
optimization ? \n    At 1st loading area");

                        for(int i = 3; i <= AmountTruck1; i+=2){
```

```
                                goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_2LA);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                          "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA1back =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                                goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                          "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_2LA);

                                if(goBack1For1stLA.after(startTime_2LA1)){
                                       if(goBack1For1stLA.after(startTime_2LA1)
){
                                              out.println("    --> The optimize
number of trucks = "+((i/2)+(i%2)));

                                              for(int j = 1; j <= i; j++){
                                                     if(j % 2 != 0){
                                                            out.println("\t\t\t\
t\t"+NameTruck1+"#"+j);
                                                     }

                                              }
                                       } break;
                                }
                          }

                          out.println("\n    At 2nd loading area");
                          int i;

                          for(i = 4; i <= AmountTruck1; i+=2){

                                goBack1For2ndLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader2_2LA);
                                out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                     "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA2back =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                                goBack1For2ndLA.add(Calendar.MINUTE,
returnTimeLA2back);
                                out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                          "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader2_2LA);

                                if(goBack1For2ndLA.after(startTime_2LA2)){
                                       if(goBack1For2ndLA.after(startTime_2LA2)
){
                                              out.println("    --> The optimize
number of trucks = "+((i/2)+(i%2)));

                                              for(int j = 1; j <= i; j++){
                                                     if(j % 2 == 0){
                                                            out.println("\t\t\t\
t\t"+NameTruck1+"#"+j);
                                                     }
```

```
            }
                                        }break;
                                  }
                            }

                            out.println("\n\n---------------------- Trucks status
in 8 hrs
      ----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                            startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_2LA+
      (int)LoadingTimeLoader1_2LA+HaulingTimeLA1);
                            startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
returnTimeTo2ndLA_2LA+
      (int)LoadingTimeLoader2_2LA + HaulingTimeLA2);

                            for(int k = 1;; k++){
                                  for(int l = 1; l <= i; l++){
                                        if(l%2 != 0){
                                              out.println("    "+
                                                    StartRunIn8hrFor1stLA.forma
t(startRunIn8hrFor1stLA.getTime())+"\t
            "+NameTruck1+"#"+l+"   \tstart dump at "+NameCR2_1);
                                              DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+
                            MeanDumpTime1);
                                              startRunIn8hrFor1stLA.add(Calendar
.MINUTE, (int)DumpingTimeTruck1);
                                              out.println("    "+

                  StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime()
)+"\t                                                 "+NameTruck1+"#"+l+"
\tdumped at     "+NameCR2_1);
                                              startRunIn8hrFor1stLA.add(Calendar
.MINUTE, returnTimeTo1stLA_2LA+
      (int)LoadingTimeLoader1_2LA + HaulingTimeLA1);
                                        }else{
                                              out.println("    "+

                  StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime()
)+"\t                                                 "+NameTruck1+"#"+l+"
\tstart dump at "+NameCR2_1);
                                              DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+
                            MeanDumpTime1);
                                              startRunIn8hrFor2ndLA.add(Calendar
.MINUTE, (int)DumpingTimeTruck1);
                                              out.println("    "+

                  StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime()
)+"\t                                                 "+NameTruck1+"#"+l+"
\tdumped at     "+NameCR2_1);

      startRunIn8hrFor2ndLA.add(Calendar.MINUTE, returnTimeTo2ndLA_2LA+
                                                      (int)LoadingTimeLoader2_2LA
+ HaulingTimeLA2);
                                        }

                                  }

                                  if(startRunIn8hrFor1stLA.after(finish8hr)||star
tRunIn8hrFor2ndLA.after(finish8hr))
      {break;}
                             }
```

```
                    }else if(AmountCR2 == 2){
                            out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                            NameTruck1+"#1   \tDepart dumping area");
                            out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                            NameTruck1+"#2   \tDepart dumping area");

                            int returnTimeTo1stLA_2LA =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                            startTime_2LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_2LA);
                            out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_2LA+" start load");

                            int returnTimeTo2ndLA_2LA =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                            startTime_2LA2.add(Calendar.MINUTE,
returnTimeTo2ndLA_2LA);
                            out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_2LA+" start load");

                            Random ran = new Random();
                            double LoadingTimeLoader1_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_2LA+
             MeanLoadTimeLoader1_2LA);
                            startTime_2LA1.add(Calendar.MINUTE,     (int)
(LoadingTimeLoader1_2LA));
                            out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_2LA+" loaded");

                            double LoadingTimeLoader2_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_2LA+
             MeanLoadTimeLoader2_2LA);
                            startTime_2LA2.add(Calendar.MINUTE,     (int)LoadingTi
meLoader2_2LA);
                            out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_2LA+" loaded");

                            int HaulingTimeLA1 =
DistanceDA_Loader1_2LA*60/HVTruck1/1000;
                            startTime_2LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                            out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                            NameTruck1+"#1   \tstart dump at "+NameCR2_1);

                            int HaulingTimeLA2 =
DistanceDA_Loader2_2LA*60/HVTruck1/1000;
                            startTime_2LA2.add(Calendar.MINUTE, HaulingTimeLA2);
                            out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                            NameTruck1+"#2   \tstart dump at "+NameCR2_1);

                            double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                            startTime_2LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                            out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
```

```
                        NameTruck1+"#1   \tdumped at     "+NameCR2_1);

                        double DumpingTimeTruck2 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_2LA2.add(Calendar.MINUTE,
(int)DumpingTimeTruck2);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tdumped at     "+NameCR2_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                        startTime_2LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \tArrive "+NameLoader1_2LA+" again");

                        int ReturnTimeToLA2again =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                        startTime_2LA2.add(Calendar.MINUTE,
ReturnTimeToLA2again);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \tArrive "+NameLoader2_2LA+" again");

                        out.println("\n--> To check, how many trucks
optimization ? \n    At 1st loading area");

                        for(int i = 3; i <= AmountTruck1; i+=2){
                              goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_2LA);
                              out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                              int returnTimeLA1back =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                              goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                              out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_2LA);

                              if(goBack1For1stLA.after(startTime_2LA1)){
                                    if(goBack1For1stLA.after(startTime_2LA1)
){
                                          out.println("    --> The optimize
number of trucks = "+((i/2)+(i%2)));

                                          for(int j = 1; j <= i; j++){
                                                if(j % 2 != 0){
                                                      out.println("\t\t\t\
t\t"+NameTruck1+"#"+j);
                                                }

            }
                                    }break;
                              }
                        }


                        out.println("\n    At 2nd loading area");
```

```
                        int i;
                        for(i = 4; i <= AmountTruck1; i+=2){
                              goBack1For2ndLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader2_2LA);
                              out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                              int returnTimeLA2back =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                              goBack1For2ndLA.add(Calendar.MINUTE,
returnTimeLA2back);
                              out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader2_2LA);

                              if(goBack1For2ndLA.after(startTime_2LA2)){
                                    if(goBack1For2ndLA.after(startTime_2LA2)
){
                                          out.println("    --> The optimize
number of trucks = "+((i/2)+(i%2)));

                                          for(int j = 1; j <= i; j++){
                                                if(j % 2 == 0){
                                                      out.println("\t\t\t\
t\t"+NameTruck1+"#"+j);
                                                }

                                          }

                                    }break;
                              }
                        }

                        out.println("\n\n---------------------- Trucks status
in 8 hrs
      -----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                        startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_2LA+
      (int)LoadingTimeLoader1_2LA+HaulingTimeLA1);
                        startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
returnTimeTo2ndLA_2LA+
      (int)LoadingTimeLoader2_2LA+HaulingTimeLA2);

                        for(int k = 1;; k++){
                              for(int l = 1; l <= i; l++){
                                    if(l%2 != 0){
                                          out.println("    "+

                  StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime()
)+"\t                                                   "+NameTruck1+"#"+l+"
\tstart dump at "+NameCR2_1);
                                          DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+
                        MeanDumpTime1);
                                          startRunIn8hrFor1stLA.add(Calendar
.MINUTE, (int)DumpingTimeTruck1);
                                          out.println("    "+

                  StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime()
)+"\t                                                   "+NameTruck1+"#"+l+"
\tdumped at     "+NameCR2_1);
                                          startRunIn8hrFor1stLA.add(Calendar
```

```
.MINUTE, returnTimeTo1stLA_2LA+
       (int)LoadingTimeLoader1_2LA + HaulingTimeLA1);

                                                double randForChooseCR =
ran.nextGaussian()*StdDevDumpTime1+
                     MeanDumpTime1;

                                                if(randForChooseCR >= 1.5){
                                                       out.println("    "+

                     StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime()
)+"\t
       "+NameTruck1+"#"+l+"   \tstart dump at "+NameCR2_2);
                                                       DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+
                            MeanDumpTime1);
                                                       startRunIn8hrFor1stLA.add(C
alendar.MINUTE,
                     (int)DumpingTimeTruck1);
                                                       out.println("    "+

                     StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime()
)+"\t
       "+NameTruck1+"#"+l+"   \tdumped at     "+NameCR2_2);
                                                       startRunIn8hrFor1stLA.add(C
alendar.MINUTE, returnTimeTo1stLA_2LA
              +(int)LoadingTimeLoader1_2LA+HaulingTimeLA1);
                                          }else{
                                                       out.println("    "+

              StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                                              "+NameTruck1+"#"+l+"
\tstart dump at "+NameCR2_1);
                                                       DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+
                            MeanDumpTime1);
                                                       startRunIn8hrFor2ndLA.add(Calendar
.MINUTE, (int)
              DumpingTimeTruck1);
                                                       out.println("    "+

              StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                                              "+NameTruck1+"#"+l+"
\tdumped at     "+NameCR2_1);
                                                       out.println("    "+

              StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                                              "+NameTruck1+"#"+l+"
\tdumped at     "+NameCR2_1);

                                                double randForChooseCR =
ran.nextGaussian()*StdDevDumpTime1+
                     MeanDumpTime1;

                                                if(randForChooseCR >= 1.5){
                                                       out.println("    "+

                     StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime()
)+"\t
       "+NameTruck1+"#"+l+"   \tstart dump at "+NameCR2_2);

                                                       DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+
                            MeanDumpTime1);
```

```
                                                        startRunIn8hrFor2ndLA.add(C
alendar.MINUTE, (int)
                                                              DumpingTimeTruck1);
                                                        out.println("    "+

                    StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime()
)+"\t
       "+NameTruck1+"#"+l+"   \tdumped at     "+NameCR2_2);
                                                        out.println("    "+

                    StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime()
)+"\t
       "+NameTruck1+"#"+l+"   \tdumped at     "+NameCR2_1);
                                                         }
                                                  }
                                     }

                                     if(startRunIn8hrFor1stLA.after(finish8hr)||star
tRunIn8hrFor2ndLA.after(finish8hr))                                              {
break;}
                              }
                        }
             }else if(ChooseView.equalsIgnoreCase("la")){
                    out.println("\n\n---------------------------- Start Time : "+
                                                                     StartTime_2LA1
.format(startTime_2LA1.getTime())+" -----------------------------\n");
                    out.println("OPERATION TIME\t\tTRUCK\t\t\tEVENT\n\n-->
Release 1st truck for each loading                                   area");

                    if(AmountCR2 == 1){
                           out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                           NameTruck1+"#1   \tDepart dumping area");
                           out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                           NameTruck1+"#2   \tDepart dumping area");

                           int returnTimeTo1stLA_2LA =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                           startTime_2LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_2LA);
                           out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_2LA+" start load");

                           int returnTimeTo2ndLA_2LA =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                           startTime_2LA2.add(Calendar.MINUTE,
returnTimeTo2ndLA_2LA);
                           out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_2LA+" start load");

                           Random ran = new Random();
                           double LoadingTimeLoader1_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_2LA+
             MeanLoadTimeLoader1_2LA);
                           startTime_2LA1.add(Calendar.MINUTE,     (int)
(LoadingTimeLoader1_2LA));
                           out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_2LA+" loaded");

                           double LoadingTimeLoader2_2LA =
```

```
(ran.nextGaussian()*StdDevLoadTimeLoader2_2LA+
            MeanLoadTimeLoader2_2LA);
                        startTime_2LA2.add(Calendar.MINUTE,     (int)LoadingTi
meLoader2_2LA);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_2LA+" loaded");

                        int HaulingTimeLA1 =
DistanceDA_Loader1_2LA*60/HVTruck1/1000;
                        startTime_2LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tstart dump at "+NameCR2_1);

                        int HaulingTimeLA2 =
DistanceDA_Loader2_2LA*60/HVTruck1/1000;
                        startTime_2LA2.add(Calendar.MINUTE, HaulingTimeLA2);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tstart dump at "+NameCR2_1);

                        double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_2LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tdumped at     "+NameCR2_1);

                        double DumpingTimeTruck2 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_2LA2.add(Calendar.MINUTE,
(int)DumpingTimeTruck2);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tdumped at     "+NameCR2_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                        startTime_2LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \tArrive "+NameLoader1_2LA+" again");

                        int ReturnTimeToLA2again =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                        startTime_2LA2.add(Calendar.MINUTE,
ReturnTimeToLA2again);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \tArrive "+NameLoader2_2LA+" again");

                              out.println("\n--> To check, how many trucks
optimization ? \n    At 1st loading area");

                              for(int i = 3; i <= AmountTruck1; i+=2){

                                    goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_2LA);
                                    out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");
```

```
                                int returnTimeLA1back =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                                goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                          "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_2LA);

                                if(goBack1For1stLA.after(startTime_2LA1)){
                                      if(goBack1For1stLA.after(startTime_2LA1)
){
                                            out.println("    --> The optimize
number of trucks = "+((i/2)+(i%2)));

                                            for(int j = 1; j <= i; j+=2){

                     out.println("\t\t\t\t\t"+NameTruck1+"#"+j);
                                                  }
                                      }break;
                                }
                          }

                          out.println("\n    At 2nd loading area");
                          int i;

                          for(i = 4; i <= AmountTruck1; i+=2){
                                goBack1For2ndLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader2_2LA);
                                out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                          "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA2back =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                                goBack1For2ndLA.add(Calendar.MINUTE,
returnTimeLA2back);
                                out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                          "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader2_2LA);

                                if(goBack1For2ndLA.after(startTime_2LA2)){
                                      if(goBack1For2ndLA.after(startTime_2LA2)
){
                                            out.println("    --> The optimize
number of trucks = "+((i/2)+(i%2)));

                                            for(int j = 2; j <= i; j+=2){

                     out.println("\t\t\t\t\t"+NameTruck1+"#"+j);
                                                  }
                                      }break;
                                }
                          }

                          out.println("\n\n---------------------- Trucks status
in 8 hrs
      ----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                          startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_2LA);
                          startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
returnTimeTo2ndLA_2LA);
```

```
                        for(int k = 1;; k++){

                              for(int l = 1, m = 2; l <= i || m <= i; l+=2,
m+=2){
                                    out.println("    "+

            StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                          "+NameTruck1+"#"+l+"   \tstart
load at "+NameLoader1_2LA);
                                          LoadingTimeLoader1_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_2LA
            +MeanLoadTimeLoader1_2LA);
                                    startRunIn8hrFor1stLA.add(Calendar.MINUT
E, (int)LoadingTimeLoader1_2LA);
                                    out.println("    "+

            StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                          "+NameTruck1+"#"+l+"   \tloaded at
"+NameLoader1_2LA);
                                    double LoopForLA_1stLA_Truck1 =
(DistanceDA_Loader1_2LA*60/HVTruck1/1000)+
            (ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1)+
                                                (DistanceDA_Loader1_2LA*60/
RVTruck1/1000);
                                    startRunIn8hrFor1stLA.add(Calendar.MINUT
E,(int)LoopForLA_1stLA_Truck1);

                                    out.println("    "+

            StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                          "+NameTruck1+"#"+m+"   \tstart
load at "+NameLoader2_2LA);
                                    LoadingTimeLoader2_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_2LA+
                  MeanLoadTimeLoader2_2LA);
                                    startRunIn8hrFor2ndLA.add(Calendar.MINUT
E, (int)LoadingTimeLoader2_2LA);
                                    out.println("    "+

            StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                          "+NameTruck1+"#"+m+"   \tloaded at
"+NameLoader2_2LA);
                                    double LoopForLA_2ndLA_Truck1 =
(DistanceDA_Loader2_2LA*60/HVTruck1/1000)+
            (ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1)+
                                                (DistanceDA_Loader2_2LA*60/
RVTruck1/1000);
                                    startRunIn8hrFor2ndLA.add(Calendar.MINUT
E,(int)LoopForLA_2ndLA_Truck1);                                    }


                              if(startRunIn8hrFor1stLA.after(finish8hr)||star
tRunIn8hrFor2ndLA.after(finish8hr))                                          {
break;}
                        }

                  }else if(AmountCR2 == 2){
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tDepart dumping area");
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tDepart dumping area");
```

```
                        int returnTimeTo1stLA_2LA =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                        startTime_2LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_2LA);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                  +"#1   \t"+NameLoader1_2LA+" start load");

                        int returnTimeTo2ndLA_2LA =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                        startTime_2LA2.add(Calendar.MINUTE,
returnTimeTo2ndLA_2LA);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                  +"#2   \t"+NameLoader2_2LA+" start load");

                        Random ran = new Random();
                        double LoadingTimeLoader1_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_2LA+
            MeanLoadTimeLoader1_2LA);
                        startTime_2LA1.add(Calendar.MINUTE,     (int)LoadingTi
meLoader1_2LA);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                  +"#1   \t"+NameLoader1_2LA+" loaded");

                        double LoadingTimeLoader2_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_2LA+
            MeanLoadTimeLoader2_2LA);
                        startTime_2LA2.add(Calendar.MINUTE,     (int)LoadingTi
meLoader2_2LA);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                  +"#2   \t"+NameLoader2_2LA+" loaded");

                        int HaulingTimeLA1 =
DistanceDA_Loader1_2LA*60/HVTruck1/1000;
                        startTime_2LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tstart dump at "+NameCR2_1);

                        int HaulingTimeLA2 =
DistanceDA_Loader2_2LA*60/HVTruck1/1000;
                        startTime_2LA2.add(Calendar.MINUTE, HaulingTimeLA2);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tstart dump at "+NameCR2_1);

                        double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_2LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tdumped at     "+NameCR2_1);

                        double DumpingTimeTruck2 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_2LA2.add(Calendar.MINUTE,
(int)DumpingTimeTruck2);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+
```

```
                        NameTruck1+"#2   \tdumped at     "+NameCR2_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                        startTime_2LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_2LA1.format(startTime_2LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \tArrive "+NameLoader1_2LA+" again");

                        int ReturnTimeToLA2again =
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                        startTime_2LA2.add(Calendar.MINUTE,
ReturnTimeToLA2again);
                        out.println("
"+StartTime_2LA2.format(startTime_2LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \tArrive "+NameLoader2_2LA+" again");

                        out.println("\n--> To check, how many trucks
optimization ? \n    At 1st loading area");

                        for(int i = 3; i <= AmountTruck1; i+=2){

                              goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_2LA);
                              out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                              int returnTimeLA1back =
DistanceDA_Loader1_2LA*60/RVTruck1/1000;
                              goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                              out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_2LA);

                              if(goBack1For1stLA.after(startTime_2LA1)){
                                    if(goBack1For1stLA.after(startTime_2LA1)
){
                                          out.println("    --> The optimize
number of trucks = "+((i/2)+(i%2)));

                                          for(int j = 1; j <= i; j+=2){

                    out.println("\t\t\t\t\t"+NameTruck1+"#"+j);
                                                }
                              }break;

                              }
                        }

                        out.println("\n    At 2nd loading area");
                        int i;

                        for(i = 4; i <= AmountTruck1; i+=2){
                              goBack1For2ndLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader2_2LA);
                              out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                              int returnTimeLA2back =
```

```
DistanceDA_Loader2_2LA*60/RVTruck1/1000;
                                    goBack1For2ndLA.add(Calendar.MINUTE,
returnTimeLA2back);
                                    out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                              "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader2_2LA);

                                    if(goBack1For2ndLA.after(startTime_2LA2)){
                                          if(goBack1For2ndLA.after(startTime_2LA2)
){
                                                out.println("    --> The optimize
number of trucks = "+((i/2)+(i%2)));

                                                for(int j = 2; j <= i; j+=2){
                                                      out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                                }
                                          }break;

                                    }
                              }

                              out.println("\n\n---------------------- Trucks status
in 8 hrs
      -----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                              startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_2LA);
                              startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
returnTimeTo2ndLA_2LA);

                              for(int k = 1;; k++){

                                    for(int l = 1, m = 2; l <= i || m <= i; l+=2,
m+=2){
                                          out.println("    "+

            StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                                "+NameTruck1+"#"+l+"   \tstart
load at "+NameLoader1_2LA);
                                          LoadingTimeLoader1_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_2LA+
                    MeanLoadTimeLoader1_2LA);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E, (int)LoadingTimeLoader1_2LA);
                                          out.println("    "+

            StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                                "+NameTruck1+"#"+l+"   \tloaded at
"+NameLoader1_2LA);

                                          double LoopForLA_1stLA_Truck1 =
(DistanceDA_Loader1_2LA*60/HVTruck1/1000)+
            (ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1)+
                                                      (DistanceDA_Loader1_2LA*60/
RVTruck1/1000);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E, (int)LoopForLA_1stLA_Truck1);

                                          out.println("    "+

            StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                                "+NameTruck1+"#"+m+"   \tstart
load at "+NameLoader2_2LA);
```

```
                                        LoadingTimeLoader2_2LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_2LA+
                  MeanLoadTimeLoader2_2LA);
                                        startRunIn8hrFor2ndLA.add(Calendar.MINUT
E, (int)LoadingTimeLoader2_2LA);
                                        out.println("    "+

            StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                              "+NameTruck1+"#"+m+"   \tloaded at
"+NameLoader2_2LA);

                                        double LoopForLA_2ndLA_Truck1 =
(DistanceDA_Loader2_2LA*60/HVTruck1/1000)+
            (ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1)+
                                                    (DistanceDA_Loader2_2LA*60/
RVTruck1/1000);
                                        startRunIn8hrFor2ndLA.add(Calendar.MINUT
E, (int)LoopForLA_2ndLA_Truck1);                                    }


                                  if(startRunIn8hrFor1stLA.after(finish8hr)||star
tRunIn8hrFor2ndLA.after(finish8hr))
      {break;}
                        }

                  }else{
                        out.println("\n\t\tWarning ! Incorrect insert.
Please, insert again.");
                  }
            }
      }
}
```

4. รหัสต้นฉบับจุดตัก 3 จุด

```
import java.util.*;
import java.text.*;
import static java.lang.System.out;
import java.lang.String.*;

public class ThreeLA130611{
      public static void main(String[] pond){
            GetProcedures();
      }
      public static void GetProcedures(){
            Scanner T = new Scanner(System.in);

            int RVTruck1 = 0, HVTruck1 = 0, AmountTruck1 = 0, DumpCapTruck1 =
0, CapCR3_1 = 0, CapCR3_2 = 0;
            double StdDevLoadTimeLoader1_3LA = 0.0, StdDevLoadTimeLoader2_3LA
= 0.0, StdDevLoadTimeLoader3_3LA =                  0.0,
MeanLoadTimeLoader1_3LA = 0.0, MeanLoadTimeLoader2_3LA = 0.0,
MeanLoadTimeLoader3_3LA =                     0.0, StdDevDumpTime1 = 0.0,
MeanDumpTime1 = 0.0;
            String NameTruck1 = null, NameCR3_1 = null, NameCR3_2 = null;

            out.print("\n\t-Trucks-\n");
            out.print("\t\tName the truck : ");
            NameTruck1 = T.next();
```

```
            out.print("\t\tHow many " + NameTruck1 + "s ? : ");
            AmountTruck1 = T.nextInt();
            out.print("\t\tDump capacity of "+NameTruck1+" (t) : ");
            DumpCapTruck1 = T.nextInt();
            out.print("\t\tAverage haul velocity of " + NameTruck1+ " (km/h) :
");
            HVTruck1 = T.nextInt();
            out.print("\t\tAverage return velocity of " + NameTruck1+ " (km/h)
: ");
            RVTruck1 = T.nextInt();

            out.println("\n\t\tDumping time of "+ NameTruck1+ " in term of
Normal Distribution.\n\t\t(If data not               fit in Normal Distribution,
you must fit its first.\n\t\t Suggestion software : Arena 13, Input
      Analyzer)");
            out.print("\t\t\t1. Normal Distribution, insert \"N\"\n\t\t\t2. I
don't know, insert \"-\"\n\t\t\t:                          ");
            String DistDumpTimeOfTruck1 = T.next();

            if(DistDumpTimeOfTruck1.equalsIgnoreCase("n")){
                  out.print("\t\tNormal format -- NORM(Mean, StdDev)
--\n\t\t\tMean   = ");
                  MeanDumpTime1 = T.nextDouble();
                  out.print("\t\t\tStdDev = ");
                  StdDevDumpTime1 = T.nextDouble();
            }else if(DistDumpTimeOfTruck1.equalsIgnoreCase("-")){
                  out.println("\t\tNormal default -- NORM(1.5, 0.5) --");
                  MeanDumpTime1 = 1.5;
                  StdDevDumpTime1 = .5;
            }

            out.print("\n\t-Loaders-\n\t\tName the loader at 1st loading
area : ");
            String NameLoader1_3LA = T.next();
            out.println("\t\tLoading Time of "+NameLoader1_3LA+" in term of
Normal Distribution.");
            out.print("\t\t\t1. Normal Distribution, insert \"N\"\n\t\t\t2. I
don't know, insert \"-\"\n\t\t\t:                          ");
            String LoadTimeLoader1_3LA = T.next();

            if(LoadTimeLoader1_3LA.equalsIgnoreCase("n")){
                  out.print("\t\tNormal format -- NORM(Mean, StdDev)
--\n\t\t\tMean   = ");
                  MeanLoadTimeLoader1_3LA = T.nextDouble();
                  out.print("\t\t\tStdDev = ");
                  StdDevLoadTimeLoader1_3LA = T.nextDouble();
            }else if(LoadTimeLoader1_3LA.equalsIgnoreCase("-")){
                  out.println("\t\tNormal default -- NORM(4.5, 0.8) --");
                  MeanLoadTimeLoader1_3LA = 4.5;
                  StdDevLoadTimeLoader1_3LA = .8;
            }

            out.print("\n\t\tName the loader at 2nd loading area : ");
            String NameLoader2_3LA = T.next();
            out.println("\t\tLoading Time of "+NameLoader2_3LA+" in term of
Normal Distribution.");
            out.print("\t\t\t1. Normal Distribution, insert \"N\"\n\t\t\t2. I
don't know, insert \"-\"\n\t\t\t:                          ");
            String LoadTimeLoader2_3LA = T.next();

            if(LoadTimeLoader2_3LA.equalsIgnoreCase("n")){
                  out.print("\t\tNormal format -- NORM(Mean, StdDev)
--\n\t\t\tMean   = ");
                  MeanLoadTimeLoader2_3LA = T.nextDouble();
```

```
                  out.print("\t\t\tStdDev = ");
                  StdDevLoadTimeLoader2_3LA = T.nextDouble();
            }else if(LoadTimeLoader2_3LA.equalsIgnoreCase("-")){
                  out.println("\t\tNormal default -- NORM(4.5, 0.8) --");
                  MeanLoadTimeLoader2_3LA = 4.5;
                  StdDevLoadTimeLoader2_3LA = .8;
            }

            out.print("\n\t\tName the loader at 3rd loading area : ");
            String NameLoader3_3LA = T.next();
            out.println("\t\tLoading Time of "+NameLoader3_3LA+" in term of
Normal Distribution.");
            out.print("\t\t\t1. Normal Distribution, insert \"N\"\n\t\t\t2. I
don't know,                                          insert \"-\"\n\t\t\t: ");
            String LoadTimeLoader3_3LA = T.next();

            if(LoadTimeLoader3_3LA.equalsIgnoreCase("n")){
                  out.print("\t\tNormal format -- NORM(Mean, StdDev)
--\n\t\t\tMean   = ");
                  MeanLoadTimeLoader3_3LA = T.nextDouble();
                  out.print("\t\t\tStdDev = ");
                  StdDevLoadTimeLoader3_3LA = T.nextDouble();
            }else if(LoadTimeLoader3_3LA.equalsIgnoreCase("-")){
                  out.println("\t\tNormal default -- NORM(4.5, 0.8) --");
                  MeanLoadTimeLoader3_3LA = 4.5;
                  StdDevLoadTimeLoader3_3LA = .8;
            }

            out.print("\n\t-Crushers-\n");
            out.print("\t\tHow many crushers ? : ");
            int AmountCR3 = T.nextInt();

            if(AmountCR3 == 1){
                  out.print("\t\tName the crusher : ");
                  NameCR3_1 = T.next();
                  out.print("\t\tCapacity of "+NameCR3_1+" (t/h) : ");
                  CapCR3_1 = T.nextInt();
            }else if(AmountCR3 == 2){
                  out.print("\t\tName the 1st crusher : ");
                  NameCR3_1 = T.next();
                  out.print("\t\tCapacity of "+NameCR3_1+" (t/h) : ");
                  CapCR3_1 = T.nextInt();
                  out.print("\t\tName the 2nd crusher : ");
                  NameCR3_2 = T.next();
                  out.print("\t\tCapacity of "+NameCR3_2+" (t/h) : ");
                  CapCR3_2 = T.nextInt();
            }else{
                  out.println("\n\t\tWarning ! Excessive number of crushers.
Please, insert again.");
            }

            out.print("\n\t\tDistance between dumping area and
"+NameLoader1_3LA+" (m.) : ");
            int DistanceDA_Loader1_3LA = T.nextInt();
            out.print("\n\t\tDistance between dumping area and
"+NameLoader2_3LA+" (m.) : ");
            int DistanceDA_Loader2_3LA = T.nextInt();
            out.print("\n\t\tDistance between dumping area and
"+NameLoader3_3LA+" (m.) : ");
            int DistanceDA_Loader3_3LA = T.nextInt();

            DateFormat StartTime_3LA1 = new SimpleDateFormat("HH:mm");
            Calendar startTime_3LA1 = Calendar.getInstance();
            DateFormat StartTime_3LA2 = new SimpleDateFormat("HH:mm");
```

```
            Calendar startTime_3LA2 = Calendar.getInstance();
            DateFormat StartTime_3LA3 = new SimpleDateFormat("HH:mm");
            Calendar startTime_3LA3 = Calendar.getInstance();

            DateFormat GoBack1For1stLA = new SimpleDateFormat("HH:mm");
            Calendar goBack1For1stLA = Calendar.getInstance();
            DateFormat GoBack1For2ndLA = new SimpleDateFormat("HH:mm");
            Calendar goBack1For2ndLA = Calendar.getInstance();
            DateFormat GoBack1For3rdLA = new SimpleDateFormat("HH:mm");
            Calendar goBack1For3rdLA = Calendar.getInstance();

            DateFormat StartRunIn8hrFor1stLA = new SimpleDateFormat("HH:mm");
            Calendar startRunIn8hrFor1stLA = Calendar.getInstance();
            DateFormat StartRunIn8hrFor2ndLA = new SimpleDateFormat("HH:mm");
            Calendar startRunIn8hrFor2ndLA = Calendar.getInstance();
            DateFormat StartRunIn8hrFor3rdLA = new SimpleDateFormat("HH:mm");
            Calendar startRunIn8hrFor3rdLA = Calendar.getInstance();

            Calendar finish8hr = Calendar.getInstance();
            finish8hr.add(Calendar.HOUR, 8);

            out.print("\n\t\tWould you like to view the system at loading
areas or dumping areas ?\n\t\t\t1.                          Dumping Areas,
insert \"DA\"\n\t\t\t2. Loading Areas, insert \"LA\"\n\t\t\t: ");
            String ChooseView = T.next();

            if(ChooseView.equalsIgnoreCase("da")){
                  out.println("\n\n---------------------------- Start Time : "+
                                                                StartTime_3LA1
.format(startTime_3LA1.getTime())+" ----------------------------\n");
                  out.println("OPERATION TIME\t\tTRUCK\t\t\tEVENT\n\n-->
Release 1st truck for each loading                                area");


                  if(AmountCR3 == 1){
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tDepart dumping area");
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tDepart dumping area");
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tDepart dumping area");

                        int returnTimeTo1stLA_3LA =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                        startTime_3LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_3LA);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                  +"#1   \t"+NameLoader1_3LA+" start load");

                        int returnTimeTo2ndLA_3LA =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                        startTime_3LA2.add(Calendar.MINUTE,
returnTimeTo2ndLA_3LA);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                  +"#2   \t"+NameLoader2_3LA+" start load");

                        int returnTimeTo3rdLA_3LA =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                        startTime_3LA3.add(Calendar.MINUTE,
```

```
returnTimeTo3rdLA_3LA);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \t"+NameLoader3_3LA+" start load");

                        Random ran = new Random();
                        double LoadingTimeLoader1_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_3LA+
            MeanLoadTimeLoader1_3LA);
                        startTime_3LA1.add(Calendar.MINUTE,     (int)LoadingTi
meLoader1_3LA);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_3LA+" loaded");

                        double LoadingTimeLoader2_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_3LA+
            MeanLoadTimeLoader2_3LA);
                        startTime_3LA2.add(Calendar.MINUTE,     (int)LoadingTi
meLoader2_3LA);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_3LA+" loaded");

                        double LoadingTimeLoader3_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader3_3LA+
            MeanLoadTimeLoader3_3LA);
                        startTime_3LA3.add(Calendar.MINUTE,     (int)LoadingTi
meLoader3_3LA);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \t"+NameLoader3_3LA+" loaded");

                        int HaulingTimeLA1 =
DistanceDA_Loader1_3LA*60/HVTruck1/1000;
                        startTime_3LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tstart dump at "+NameCR3_1);

                        int HaulingTimeLA2 =
DistanceDA_Loader2_3LA*60/HVTruck1/1000;
                        startTime_3LA2.add(Calendar.MINUTE, HaulingTimeLA2);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tstart dump at "+NameCR3_1);

                        int HaulingTimeLA3 =
DistanceDA_Loader3_3LA*60/HVTruck1/1000;
                        startTime_3LA3.add(Calendar.MINUTE, HaulingTimeLA3);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tstart dump at "+NameCR3_1);

                        double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_3LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tdumped at     "+NameCR3_1);

                        double DumpingTimeTruck2 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
```

```
                        startTime_3LA2.add(Calendar.MINUTE,
(int)DumpingTimeTruck2);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tdumped at     "+NameCR3_1);

                        double DumpingTimeTruck3 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_3LA3.add(Calendar.MINUTE,
(int)DumpingTimeTruck3);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tdumped at     "+NameCR3_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                        startTime_3LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \tArrive "+NameLoader1_3LA+" again");

                        int ReturnTimeToLA2again =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                        startTime_3LA2.add(Calendar.MINUTE,
ReturnTimeToLA2again);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \tArrive "+NameLoader2_3LA+" again");

                        int ReturnTimeToLA3again =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                        startTime_3LA3.add(Calendar.MINUTE,
ReturnTimeToLA3again);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \tArrive "+NameLoader3_3LA+" again");

                        out.println("\n--> To check, how many trucks
optimization ? \n    At 1st loading area");

                        for(int i = 4; i <= AmountTruck1; i+=3){
                                goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_3LA);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA1back =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                                goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_3LA);

                                if(goBack1For1stLA.after(startTime_3LA1)||i >
AmountTruck1){
                                       if(goBack1For1stLA.after(startTime_3LA1)
||i > AmountTruck1){
                                              out.println("    --> The optimize
number of trucks = "+(i-1)/2);
```

```
                                                for(int j = 1; j <= i; j+=3){
                                                      out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                                }
                                          } break;
                                    }
                              }

                              out.println("\n    At 2nd loading area");
                              int i;

                              for(i = 5; i <= AmountTruck1; i+=3){
                                    goBack1For2ndLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader2_3LA);
                                    out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                              "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                    int returnTimeLA2back =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                                    goBack1For2ndLA.add(Calendar.MINUTE,
returnTimeLA2back);
                                    out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                              "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader2_3LA);

                                    if(goBack1For2ndLA.after(startTime_3LA2)||i >
AmountTruck1){
                                          if(goBack1For2ndLA.after(startTime_3LA2)
||i > AmountTruck1){
                                                out.println("    --> The optimize
number of trucks = "+(i-2)/2);

                                                for(int j = 2; j <= i; j+=3){
                                                      out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                                }
                                          }break;
                                    }
                              }

                              out.println("\n    At 3rd loading area");

                              for(i = 6; i <= AmountTruck1; ){

                                    goBack1For3rdLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader3_3LA);
                                    out.println("
"+GoBack1For3rdLA.format(goBack1For3rdLA.getTime())+"\t
                              "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                    int returnTimeLA3back =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                                    goBack1For3rdLA.add(Calendar.MINUTE,
returnTimeLA3back);
                                    out.println("
"+GoBack1For3rdLA.format(goBack1For3rdLA.getTime())+"\t
                              "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader3_3LA);

                                    if(goBack1For3rdLA.after(startTime_3LA3)||i >=
AmountTruck1){
                                          if(goBack1For3rdLA.after(startTime_3LA3)
```

```
||i >= AmountTruck1){
                                                out.println("    --> The optimize
number of trucks = "+(i-2)/2);

                                                for(int j = 3; j <= i; j+=3){

                                                      out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
      }
                                          }break;
                                    }
                                    i+=3;
                              }

                              out.println("\n\n---------------------- Trucks status
in 8 hrs
      ----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                              startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_3LA+
      (int)LoadingTimeLoader1_3LA+HaulingTimeLA1);
                              startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
returnTimeTo2ndLA_3LA+
      (int)LoadingTimeLoader2_3LA+HaulingTimeLA2);
                              startRunIn8hrFor3rdLA.add(Calendar.MINUTE,
returnTimeTo3rdLA_3LA+
      (int)LoadingTimeLoader3_3LA+HaulingTimeLA3);

                              for(int k = 1;; k++){
                                    for(int l = 1, m = 2, n = 3; l <= i||m <= i||n
<= i; l+=3, m+=3, n+=3){
                                          out.println("    "+

            StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                                "+NameTruck1+"#"+l+"   \tstart
dump at "+NameCR3_1);
                                          DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+
                              MeanDumpTime1);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E, (int)DumpingTimeTruck1);
                                          out.println("    "+

            StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                                "+NameTruck1+"#"+l+"   \tdumped at
"+NameCR3_1);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E, returnTimeTo1stLA_3LA+
      (int)LoadingTimeLoader1_3LA+HaulingTimeLA1);

                                          out.println("    "+

            StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                                "+NameTruck1+"#"+m+"   \tstart
dump at "+NameCR3_1);
                                          DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                          startRunIn8hrFor2ndLA.add(Calendar.MINUT
E, (int)DumpingTimeTruck1);
                                          out.println("    "+

            StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                                "+NameTruck1+"#"+m+"   \tdumped at
"+NameCR3_1);
```

```
      startRunIn8hrFor2ndLA.add(Calendar.MINUTE, returnTimeTo2ndLA_3LA+
                                                    (int)LoadingTimeLoader2_3LA
+HaulingTimeLA2);

                                    out.println("    "+

            StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime())+"\t
                                          "+NameTruck1+"#"+n+"   \tstart
dump at "+NameCR3_1);
                                    DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                    startRunIn8hrFor3rdLA.add(Calendar.MINUT
E, (int)DumpingTimeTruck1);
                                    out.println("    "+

            StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime())+"\t
                                          "+NameTruck1+"#"+n+"   \tdumped at
"+NameCR3_1);

      startRunIn8hrFor3rdLA.add(Calendar.MINUTE, returnTimeTo3rdLA_3LA+
                                                    (int)LoadingTimeLoader3_3LA
+ HaulingTimeLA3);
                               }
                               if(startRunIn8hrFor1stLA.after(finish8hr)||star
tRunIn8hrFor2ndLA.after(finish8hr)
      ||startRunIn8hrFor3rdLA.after(finish8hr)){break;}

                           }

                  }else if(AmountCR3 == 2){

                           out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                           NameTruck1+"#1   \tDepart dumping area");
                           out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                           NameTruck1+"#2   \tDepart dumping area");
                           out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                           NameTruck1+"#3   \tDepart dumping area");

                           int returnTimeTo1stLA_3LA =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                           startTime_3LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_3LA);
                           out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_3LA+" start load");

                           int returnTimeTo2ndLA_3LA =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                           startTime_3LA2.add(Calendar.MINUTE,
returnTimeTo2ndLA_3LA);
                           out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_3LA+" start load");

                           int returnTimeTo3rdLA_3LA =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                           startTime_3LA3.add(Calendar.MINUTE,
returnTimeTo3rdLA_3LA);
                           out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \t"+NameLoader3_3LA+" start load");
```

```
                        Random ran = new Random();
                        double LoadingTimeLoader1_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_3LA+
            MeanLoadTimeLoader1_3LA);
                        startTime_3LA1.add(Calendar.MINUTE,     (int)
(LoadingTimeLoader1_3LA));
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_3LA+" loaded");

                        double LoadingTimeLoader2_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_3LA+
            MeanLoadTimeLoader2_3LA);
                        startTime_3LA2.add(Calendar.MINUTE,     (int)LoadingTi
meLoader2_3LA);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_3LA+" loaded");

                        double LoadingTimeLoader3_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader3_3LA+
            MeanLoadTimeLoader3_3LA);
                        startTime_3LA3.add(Calendar.MINUTE,     (int)LoadingTi
meLoader3_3LA);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \t"+NameLoader3_3LA+" loaded");

                        int HaulingTimeLA1 =
DistanceDA_Loader1_3LA*60/HVTruck1/1000;
                        startTime_3LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tstart dump at "+NameCR3_1);

                        int HaulingTimeLA2 =
DistanceDA_Loader2_3LA*60/HVTruck1/1000;
                        startTime_3LA2.add(Calendar.MINUTE, HaulingTimeLA2);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tstart dump at "+NameCR3_1);

                        int HaulingTimeLA3 =
DistanceDA_Loader3_3LA*60/HVTruck1/1000;
                        startTime_3LA3.add(Calendar.MINUTE, HaulingTimeLA3);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tstart dump at "+NameCR3_1);

                        double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_3LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tdumped at     "+NameCR3_1);

                        double DumpingTimeTruck2 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_3LA2.add(Calendar.MINUTE,
(int)DumpingTimeTruck2);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
```

```
                        NameTruck1+"#2   \tdumped at     "+NameCR3_1);

                        double DumpingTimeTruck3 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_3LA3.add(Calendar.MINUTE,
(int)DumpingTimeTruck3);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tdumped at     "+NameCR3_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                        startTime_3LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tArrive "+NameLoader1_3LA+"
again");

                        int ReturnTimeToLA2again =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                        startTime_3LA2.add(Calendar.MINUTE,
ReturnTimeToLA2again);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tArrive "+NameLoader2_3LA+"
again");

                        int ReturnTimeToLA3again =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                        startTime_3LA3.add(Calendar.MINUTE,
ReturnTimeToLA3again);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tArrive "+NameLoader3_3LA+"
again");

                        out.println("\n--> To check, how many trucks
optimization ? \n    At 1st loading area");

                        for(int i = 4; i <= AmountTruck1; i+=3){
                                goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_3LA);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA1back =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                                goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_3LA);

                                if(goBack1For1stLA.after(startTime_3LA1)||i >
AmountTruck1){
                                        if(goBack1For1stLA.after(startTime_3LA1)
|| i > AmountTruck1 ){
                                                out.println("    --> The optimize
number of trucks = "+(i-1)/2);

                                                for(int j = 1; j <= i; j+=3){
```

```
                                                        out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                                }
                                        }break;
                                }
                        }

                        out.println("\n    At 2nd loading area");
                        int i;

                        for(i = 5; i <= AmountTruck1; i+=3){
                                goBack1For2ndLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader2_3LA);
                                out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                    "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA2back =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                                goBack1For2ndLA.add(Calendar.MINUTE,
returnTimeLA2back);
                                out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                            "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader2_3LA);

                                if(goBack1For2ndLA.after(startTime_3LA2) || i >
AmountTruck1){
                                        if(goBack1For2ndLA.after(startTime_3LA2)
|| i > AmountTruck1){
                                                out.println("    --> The optimize
number of trucks = "+(i-2)/2);

                                                for(int j = 2; j <= i; j+=3){
                                                        out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                                }
                                        }break;
                                }
                        }

                        out.println("\n    At 3rd loading area");

                        for(i = 6; i <= AmountTruck1; ){

                                goBack1For3rdLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader3_3LA);
                                out.println("
"+GoBack1For3rdLA.format(goBack1For3rdLA.getTime())+"\t
                            "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA3back =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                                goBack1For3rdLA.add(Calendar.MINUTE,
returnTimeLA3back);
                                out.println("
"+GoBack1For3rdLA.format(goBack1For3rdLA.getTime())+"\t
                            "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader3_3LA);

                                if(goBack1For3rdLA.after(startTime_3LA3) || i
>= AmountTruck1){
                                        if(goBack1For3rdLA.after(startTime_3LA3)
|| i >= AmountTruck1){
```

```
                                                out.println("    --> The optimize
number of trucks = "+(i-2)/2);

                                                for(int j = 3; j <= i; j+=3){

                        out.println("\t\t\t\t\t"+NameTruck1+"#"+j);
                                                        }
                                        }break;
                                }
                                i+=3;
                        }

                        out.println("\n\n---------------------- Trucks status
in 8 hrs
      -----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                        startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_3LA+
      (int)LoadingTimeLoader1_3LA+HaulingTimeLA1);
                        startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
returnTimeTo2ndLA_3LA+
      (int)LoadingTimeLoader2_3LA+HaulingTimeLA2);
                        startRunIn8hrFor3rdLA.add(Calendar.MINUTE,
returnTimeTo3rdLA_3LA+
      (int)LoadingTimeLoader3_3LA+HaulingTimeLA3);

                        for(int k = 1;; k++){

                                for(int l = 1, m = 2, n = 3; l<=i || m<=i ||
n<=i; l+=3, m+=3, n+=3){
                                        out.println("    "+
                                                StartRunIn8hrFor1stLA.format(start
RunIn8hrFor1stLA.getTime())+"\t
      "+NameTruck1+"#"+l+"   \tstart dump at "+NameCR3_1);
                                        DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                        startRunIn8hrFor1stLA.add(Calendar.MINUT
E, (int)DumpingTimeTruck1);
                                        out.println("    "+

            StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                                "+NameTruck1+"#"+l+"   \tdumped at
"+NameCR3_1);
                                        startRunIn8hrFor1stLA.add(Calendar.MINUT
E, returnTimeTo1stLA_3LA+
      (int)LoadingTimeLoader1_3LA+HaulingTimeLA1);

                                        double randForChooseCR =
ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1;

                                        if(randForChooseCR >= 1.5){
                                                out.println("    "+

                  StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime()
)+"\t                                                   "+NameTruck1+"#"+l+"
\tstart dump at "+NameCR3_2);
                                                DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                                startRunIn8hrFor1stLA.add(Calendar
.MINUTE, (int)DumpingTimeTruck1);
                                                out.println("    "+

                  StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime()
)+"\t                                                   "+NameTruck1+"#"+l+"
\tdumped at     "+NameCR3_2);
```

```
                                                startRunIn8hrFor1stLA.add(Calendar
.MINUTE, returnTimeTo1stLA_3LA+
       (int)LoadingTimeLoader1_3LA+HaulingTimeLA1);
                                          }

                                          out.println("    "+

              StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                                "+NameTruck1+"#"+m+"   \tstart
dump at "+NameCR3_1);
                                          DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                          startRunIn8hrFor2ndLA.add(Calendar.MINUT
E, (int)DumpingTimeTruck1);
                                          out.println("    "+

              StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                                "+NameTruck1+"#"+m+"   \tdumped at
"+NameCR3_1);

       startRunIn8hrFor2ndLA.add(Calendar.MINUTE, returnTimeTo2ndLA_3LA+
                                                      (int)LoadingTimeLoader2_3LA
+HaulingTimeLA2);

                                          randForChooseCR =
ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1;

                                          if(randForChooseCR >= 1.5){
                                                out.println("    "+

                    StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime()
)+"\t                                                   "+NameTruck1+"#"+m+"
\tstart dump at "+NameCR3_2);
                                                DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+
                           MeanDumpTime1);
                                                startRunIn8hrFor2ndLA.add(Calendar
.MINUTE, (int)DumpingTimeTruck1);
                                                out.println("    "+

                    StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime()
)+"\t                                                   "+NameTruck1+"#"+m+"
\tdumped at     "+NameCR3_2);

       startRunIn8hrFor2ndLA.add(Calendar.MINUTE, returnTimeTo2ndLA_3LA+
                                                      (int)LoadingTimeLoader2_3LA
+ HaulingTimeLA2);
                                          }

                                          out.println("    "+

              StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime())+"\t
                                                "+NameTruck1+"#"+n+"   \tstart
dump at "+NameCR3_1);
                                          DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                          startRunIn8hrFor3rdLA.add(Calendar.MINUT
E, (int)DumpingTimeTruck1);
                                          out.println("    "+

              StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime())+"\t
                                                "+NameTruck1+"#"+n+"   \tdumped at
"+NameCR3_1);
```

```
      startRunIn8hrFor3rdLA.add(Calendar.MINUTE, returnTimeTo3rdLA_3LA+
                                                      (int)LoadingTimeLoader3_3LA
+HaulingTimeLA3);

                                        randForChooseCR =
ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1;

                                        if(randForChooseCR >= 1.5){
                                              out.println("    "+

                  StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime()
)+"\t                                                  "+NameTruck1+"#"+n+"
\tstart dump at "+NameCR3_2);
                                              DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                                              startRunIn8hrFor3rdLA.add(Calendar
.MINUTE, (int)DumpingTimeTruck1);
                                              out.println("    "+

                  StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime()
)+"\t                                                  "+NameTruck1+"#"+n+"
\tdumped at     "+NameCR3_2);

      startRunIn8hrFor3rdLA.add(Calendar.MINUTE, returnTimeTo3rdLA_3LA+
                                                      (int)LoadingTimeLoader3_3LA
+ HaulingTimeLA3);
                                        }
                                  }
                                  if(startRunIn8hrFor1stLA.after(finish8hr))
{break;}
                            }
                  }

            }else if(ChooseView.equalsIgnoreCase("la")){

                  out.println("\n\n---------------------------- Start Time : "+
                                                       StartTime_3LA1
.format(startTime_3LA1.getTime())+" -----------------------------\n");
                  out.println("OPERATION TIME\t\tTRUCK\t\t\tEVENT\n\n-->
Release 1st truck for each loading                          area");

                  if(AmountCR3 == 1){
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tDepart dumping area");
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tDepart dumping area");
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tDepart dumping area");

                        int returnTimeTo1stLA_3LA =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                        startTime_3LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_3LA);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                  +"#1   \t"+NameLoader1_3LA+" start load");

                        int returnTimeTo2ndLA_3LA =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                        startTime_3LA2.add(Calendar.MINUTE,
returnTimeTo2ndLA_3LA);
```

```
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_3LA+" start load");

                        int returnTimeTo3rdLA_3LA =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                        startTime_3LA3.add(Calendar.MINUTE,
returnTimeTo3rdLA_3LA);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \t"+NameLoader3_3LA+" start load");

                        Random ran = new Random();
                        double LoadingTimeLoader1_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_3LA+
            MeanLoadTimeLoader1_3LA);
                        startTime_3LA1.add(Calendar.MINUTE,     (int)LoadingTi
meLoader1_3LA);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_3LA+" loaded");

                        double LoadingTimeLoader2_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_3LA+
            MeanLoadTimeLoader2_3LA);
                        startTime_3LA2.add(Calendar.MINUTE,     (int)LoadingTi
meLoader2_3LA);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_3LA+" loaded");

                        double LoadingTimeLoader3_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader3_3LA+
            MeanLoadTimeLoader3_3LA);
                        startTime_3LA3.add(Calendar.MINUTE,     (int)LoadingTi
meLoader3_3LA);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \t"+NameLoader3_3LA+" loaded");

                        int HaulingTimeLA1 =
DistanceDA_Loader1_3LA*60/HVTruck1/1000;
                        startTime_3LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1+
                        "#1   \tstart dump at "+NameCR3_1);

                        int HaulingTimeLA2 =
DistanceDA_Loader2_3LA*60/HVTruck1/1000;
                        startTime_3LA2.add(Calendar.MINUTE, HaulingTimeLA2);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tstart dump at "+NameCR3_1);

                        int HaulingTimeLA3 =
DistanceDA_Loader3_3LA*60/HVTruck1/1000;
                        startTime_3LA3.add(Calendar.MINUTE, HaulingTimeLA3);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tstart dump at "+NameCR3_1);

                        double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_3LA1.add(Calendar.MINUTE,
```

```
(int)DumpingTimeTruck1);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                        NameTruck1+"#1   \tdumped at     "+NameCR3_1);

                        double DumpingTimeTruck2 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_3LA2.add(Calendar.MINUTE,
(int)DumpingTimeTruck2);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                        NameTruck1+"#2   \tdumped at     "+NameCR3_1);

                        double DumpingTimeTruck3 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_3LA3.add(Calendar.MINUTE,
(int)DumpingTimeTruck3);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tdumped at     "+NameCR3_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                        startTime_3LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \tArrive "+NameLoader1_3LA+" again");

                        int ReturnTimeToLA2again =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                        startTime_3LA2.add(Calendar.MINUTE,
ReturnTimeToLA2again);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \tArrive "+NameLoader2_3LA+" again");

                        int ReturnTimeToLA3again =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                        startTime_3LA3.add(Calendar.MINUTE,
ReturnTimeToLA3again);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \tArrive "+NameLoader3_3LA+" again");

                        out.println("\n--> To check, how many trucks
optimization ? \n    At 1st loading area");

                        for(int i = 4; i <= AmountTruck1; i+=3){
                                goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_3LA);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA1back =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                                goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                                out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                                        "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_3LA);
```

```
                                    if(goBack1For1stLA.after(startTime_3LA1) || i >
AmountTruck1){
                                          if(goBack1For1stLA.after(startTime_3LA1)
|| i > AmountTruck1 ){
                                                out.println("    --> The optimize
number of trucks = "+(i-1)/2);

                                                for(int j = 1; j <= i; j+=3){
                                                      out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                                }
                                          }break;
                                    }
                              }

                              out.println("\n    At 2nd loading area");
                              int i;

                              for(i = 5; i <= AmountTruck1; i+=3){
                                    goBack1For2ndLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader2_3LA);
                                    out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                              "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                    int returnTimeLA2back =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                                    goBack1For2ndLA.add(Calendar.MINUTE,
returnTimeLA2back);
                                    out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                              "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader2_3LA);

                                    if(goBack1For2ndLA.after(startTime_3LA2) || i >
AmountTruck1){
                                          if(goBack1For2ndLA.after(startTime_3LA2)
|| i > AmountTruck1){
                                                out.println("    --> The optimize
number of trucks = "+(i-2)/2);

                                                for(int j = 2; j <= i; j+=3){
                                                      out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                                }
                                          }break;
                                    }
                              }

                              out.println("\n    At 3rd loading area");

                              for(i = 6; i <= AmountTruck1; ){

                                    goBack1For3rdLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader3_3LA);
                                    out.println("
"+GoBack1For3rdLA.format(goBack1For3rdLA.getTime())+"\t
                              "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                    int returnTimeLA3back =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                                    goBack1For3rdLA.add(Calendar.MINUTE,
returnTimeLA3back);
                                    out.println("
```

```
"+GoBack1For3rdLA.format(goBack1For3rdLA.getTime())+"\t
                              "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader3_3LA);

                                    if(goBack1For3rdLA.after(startTime_3LA3) || i
>= AmountTruck1){
                                          if(goBack1For3rdLA.after(startTime_3LA3)
|| i >= AmountTruck1){
                                                out.println("    --> The optimize
number of trucks = "+(i-2)/2);

                                                for(int j = 3; j <= i; j+=3){

                        out.println("\t\t\t\t\t"+NameTruck1+"#"+j);
                                                      }
                                          }break;
                                    }
                                    i+=3;
                              }

                              out.println("\n\n---------------------- Trucks status
in 8 hrs
       ----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                              startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_3LA);
                              startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
returnTimeTo2ndLA_3LA);
                              startRunIn8hrFor3rdLA.add(Calendar.MINUTE,
returnTimeTo3rdLA_3LA);
                              for(int k = 1;; k++){
                                    for(int l = 1, m = 2, n = 3; l<=i || m<=i ||
n<=i; l+=3, m+=3, n+=3){
                                          out.println("    "+

             StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                                "+NameTruck1+"#"+l+"   \tstart
load at "+NameLoader1_3LA);
                                          LoadingTimeLoader1_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_3LA+
                        MeanLoadTimeLoader1_3LA);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E, (int)LoadingTimeLoader1_3LA);
                                          out.println("    "+

             StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())+"\t
                                                "+NameTruck1+"#"+l+"   \tloaded at
"+NameLoader1_3LA);
                                          startRunIn8hrFor1stLA.add(Calendar.MINUT
E, HaulingTimeLA1+
       (int)DumpingTimeTruck1+returnTimeTo1stLA_3LA);

                                          out.println("    "+

             StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
                                                "+NameTruck1+"#"+m+"   \tstart
load at "+NameLoader2_3LA);
                                          LoadingTimeLoader2_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_3LA+
                        MeanLoadTimeLoader2_3LA);
                                          startRunIn8hrFor2ndLA.add(Calendar.MINUT
E, (int)LoadingTimeLoader2_3LA);
                                          out.println("    "+

             StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())+"\t
```

```
                                                    "+NameTruck1+"#"+m+"   \tloaded at
"+NameLoader2_3LA);

       startRunIn8hrFor2ndLA.add(Calendar.MINUTE, HaulingTimeLA2+
                                                          (int)DumpingTimeTruck1 +
returnTimeTo2ndLA_3LA);

                                          out.println("    "+

             StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime())+"\t
                                                          "+NameTruck1+"#"+n+"
\tstart load at "+NameLoader3_3LA);
                                          LoadingTimeLoader3_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader3_3LA+
                    MeanLoadTimeLoader3_3LA);
                                          startRunIn8hrFor3rdLA.add(Calendar.MINUT
E, (int)LoadingTimeLoader3_3LA);
                                          out.println("    "+

             StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime())+"\t
                                                    "+NameTruck1+"#"+n+"   \tloaded at
"+NameLoader3_3LA);

       startRunIn8hrFor3rdLA.add(Calendar.MINUTE, returnTimeTo3rdLA_3LA+
                                                          (int)LoadingTimeLoader3_3LA
+ HaulingTimeLA3);
                                  }
                                  if(startRunIn8hrFor1stLA.after(finish8hr)||star
tRunIn8hrFor2ndLA.after(finish8hr)
       || startRunIn8hrFor3rdLA.after(finish8hr)){break;}
                                  }

                     }else if(AmountCR3 == 2){
                                  out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                                  NameTruck1+"#1   \tDepart dumping area");
                                  out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                                  NameTruck1+"#2   \tDepart dumping area");
                                  out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                                  NameTruck1+"#3   \tDepart dumping area");

                                  int returnTimeTo1stLA_3LA =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                                  startTime_3LA1.add(Calendar.MINUTE,
returnTimeTo1stLA_3LA);
                                  out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                     +"#1   \t"+NameLoader1_3LA+" start load");

                                  int returnTimeTo2ndLA_3LA =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                                  startTime_3LA2.add(Calendar.MINUTE,
returnTimeTo2ndLA_3LA);
                                  out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                     +"#2   \t"+NameLoader2_3LA+" start load");

                                  int returnTimeTo3rdLA_3LA =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                                  startTime_3LA3.add(Calendar.MINUTE,
returnTimeTo3rdLA_3LA);
                                  out.println("
```

```
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \t"+NameLoader3_3LA+" start load");

                           Random ran = new Random();
                           double LoadingTimeLoader1_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_3LA+
             MeanLoadTimeLoader1_3LA);
                           startTime_3LA1.add(Calendar.MINUTE,     (int)LoadingTi
meLoader1_3LA);
                           out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \t"+NameLoader1_3LA+" loaded");

                           double LoadingTimeLoader2_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_3LA+
             MeanLoadTimeLoader2_3LA);
                           startTime_3LA2.add(Calendar.MINUTE,     (int)LoadingTi
meLoader2_3LA);
                           out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \t"+NameLoader2_3LA+" loaded");

                           double LoadingTimeLoader3_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader3_3LA+
             MeanLoadTimeLoader3_3LA);
                           startTime_3LA3.add(Calendar.MINUTE,     (int)LoadingTi
meLoader3_3LA);
                           out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \t"+NameLoader3_3LA+" loaded");

                           int HaulingTimeLA1 =
DistanceDA_Loader1_3LA*60/HVTruck1/1000;
                           startTime_3LA1.add(Calendar.MINUTE, HaulingTimeLA1);
                           out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                           NameTruck1+"#1   \tstart dump at "+NameCR3_1);

                           int HaulingTimeLA2 =
DistanceDA_Loader2_3LA*60/HVTruck1/1000;
                           startTime_3LA2.add(Calendar.MINUTE, HaulingTimeLA2);
                           out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                           NameTruck1+"#2   \tstart dump at "+NameCR3_1);

                           int HaulingTimeLA3 =
DistanceDA_Loader3_3LA*60/HVTruck1/1000;
                           startTime_3LA3.add(Calendar.MINUTE, HaulingTimeLA3);
                           out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                           NameTruck1+"#3   \tstart dump at "+NameCR3_1);

                           double DumpingTimeTruck1 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                           startTime_3LA1.add(Calendar.MINUTE,
(int)DumpingTimeTruck1);
                           out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+
                                  NameTruck1+"#1   \tdumped at     "+NameCR3_1);

                           double DumpingTimeTruck2 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                           startTime_3LA2.add(Calendar.MINUTE,
(int)DumpingTimeTruck2);
```

```
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+
                    NameTruck1+"#2   \tdumped at     "+NameCR3_1);

                        double DumpingTimeTruck3 =
(ran.nextGaussian()*StdDevDumpTime1+MeanDumpTime1);
                        startTime_3LA3.add(Calendar.MINUTE,
(int)DumpingTimeTruck3);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+
                        NameTruck1+"#3   \tdumped at     "+NameCR3_1);

                        int ReturnTimeToLA1again =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                        startTime_3LA1.add(Calendar.MINUTE,
ReturnTimeToLA1again);
                        out.println("
"+StartTime_3LA1.format(startTime_3LA1.getTime())+"\t      "+NameTruck1
                    +"#1   \tArrive "+NameLoader1_3LA+" again");

                        int ReturnTimeToLA2again =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                        startTime_3LA2.add(Calendar.MINUTE,
ReturnTimeToLA2again);
                        out.println("
"+StartTime_3LA2.format(startTime_3LA2.getTime())+"\t      "+NameTruck1
                    +"#2   \tArrive "+NameLoader2_3LA+" again");

                        int ReturnTimeToLA3again =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                        startTime_3LA3.add(Calendar.MINUTE,
ReturnTimeToLA3again);
                        out.println("
"+StartTime_3LA3.format(startTime_3LA3.getTime())+"\t      "+NameTruck1
                    +"#3   \tArrive "+NameLoader3_3LA+" again");

                        out.println("\n--> To check, how many trucks
optimization ? \n    At 1st loading area");

                        for(int i = 4; i <= AmountTruck1; i+=3){
                            goBack1For1stLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader1_3LA);
                            out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                            int returnTimeLA1back =
DistanceDA_Loader1_3LA*60/RVTruck1/1000;
                            goBack1For1stLA.add(Calendar.MINUTE,
returnTimeLA1back);
                            out.println("
"+GoBack1For1stLA.format(goBack1For1stLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader1_3LA);

                            if(goBack1For1stLA.after(startTime_3LA1) || i >
AmountTruck1){
                                if(goBack1For1stLA.after(startTime_3LA1)
|| i > AmountTruck1 ){
                                    out.println("    --> The optimize
number of trucks = "+(i-1)/2);

                                    for(int j = 1; j <= i; j+=3){
                                        out.println("\t\t\t\t\t"+Na
```

```
meTruck1+"#"+j);
                                                }
                                        } break;
                                }
                        }

                        out.println("\n    At 2nd loading area");
                        int i;

                        for(i = 5; i <= AmountTruck1; i+=3){
                                goBack1For2ndLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader2_3LA);
                                out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA2back =
DistanceDA_Loader2_3LA*60/RVTruck1/1000;
                                goBack1For2ndLA.add(Calendar.MINUTE,
returnTimeLA2back);
                                out.println("
"+GoBack1For2ndLA.format(goBack1For2ndLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader2_3LA);

                                if(goBack1For2ndLA.after(startTime_3LA2) || i >
AmountTruck1){
                                        if(goBack1For2ndLA.after(startTime_3LA2)
|| i > AmountTruck1){
                                                out.println("    --> The optimize
number of trucks = "+(i-2)/2);

                                                for(int j = 2; j <= i; j+=3){
                                                        out.println("\t\t\t\t\t"+Na
meTruck1+"#"+j);
                                                }
                                        }break;
                                }
                        }

                        out.println("\n    At 3rd loading area");
                        for(i = 6; i <= AmountTruck1; ){

                                goBack1For3rdLA.add(Calendar.MINUTE,
(int)MeanLoadTimeLoader3_3LA);
                                out.println("
"+GoBack1For3rdLA.format(goBack1For3rdLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \tDepart dumping area");

                                int returnTimeLA3back =
DistanceDA_Loader3_3LA*60/RVTruck1/1000;
                                goBack1For3rdLA.add(Calendar.MINUTE,
returnTimeLA3back);
                                out.println("
"+GoBack1For3rdLA.format(goBack1For3rdLA.getTime())+"\t
                        "+NameTruck1+"#"+i+"   \t"+"Arrive
"+NameLoader3_3LA);

                                if(goBack1For3rdLA.after(startTime_3LA3) || i
>= AmountTruck1){
                                        if(goBack1For3rdLA.after(startTime_3LA3)
|| i >= AmountTruck1){
                                                out.println("    --> The optimize
number of trucks = "+(i-2)/2);
```

```
                                        for(int j = 3; j <= i; j+=3){

                                              out.println("\t\t\t\t\t"+NameTruck
1+"#"+j);                                                                    }
                                  }break;
                            }
                            i+=3;
                      }

                      out.println("\n\n---------------------- Trucks status in 8
hrs
      ----------------------\n\nOPERATION TIME\t\tTRUCK\t\t\tEVENT\n");
                      startRunIn8hrFor1stLA.add(Calendar.MINUTE,
returnTimeTo1stLA_3LA);
                      startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
returnTimeTo2ndLA_3LA);
                      startRunIn8hrFor3rdLA.add(Calendar.MINUTE,
returnTimeTo3rdLA_3LA);

                      for(int k = 1;; k++){
                            for(int l = 1, m = 2, n = 3; l<=i || m<=i || n<=i;
l+=3, m+=3, n+=3){
                                  out.println("
"+StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())
                                  +"\t      "+NameTruck1+"#"+l+"   \tstart load at
"+NameLoader1_3LA);
                                  LoadingTimeLoader1_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader1_3LA+
                      MeanLoadTimeLoader1_3LA);
                                  startRunIn8hrFor1stLA.add(Calendar.MINUTE,
(int)LoadingTimeLoader1_3LA);
                                  out.println("
"+StartRunIn8hrFor1stLA.format(startRunIn8hrFor1stLA.getTime())
                                  +"\t      "+NameTruck1+"#"+l+"   \tloaded at
"+NameLoader1_3LA);
                                  startRunIn8hrFor1stLA.add(Calendar.MINUTE,
HaulingTimeLA1+(int)DumpingTimeTruck1+
            returnTimeTo1stLA_3LA);

                                  out.println("
"+StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())
                                  +"\t      "+NameTruck1+"#"+m+"   \tstart load at
"+NameLoader2_3LA);
                                  LoadingTimeLoader2_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader2_3LA+
                      MeanLoadTimeLoader2_3LA);
                                  startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
(int)LoadingTimeLoader2_3LA);
                                  out.println("
"+StartRunIn8hrFor2ndLA.format(startRunIn8hrFor2ndLA.getTime())
                                  +"\t      "+NameTruck1+"#"+m+"   \tloaded at
"+NameLoader2_3LA);
                                  startRunIn8hrFor2ndLA.add(Calendar.MINUTE,
HaulingTimeLA2+(int)DumpingTimeTruck1+
            returnTimeTo2ndLA_3LA);

                                  out.println("
"+StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime())
                                  +"\t      "+NameTruck1+"#"+n+"   \tstart load at
"+NameLoader3_3LA);
                                  LoadingTimeLoader3_3LA =
(ran.nextGaussian()*StdDevLoadTimeLoader3_3LA+
                      MeanLoadTimeLoader3_3LA);
```

```
                                    startRunIn8hrFor3rdLA.add(Calendar.MINUTE,
(int)LoadingTimeLoader3_3LA);
                                    out.println("
"+StartRunIn8hrFor3rdLA.format(startRunIn8hrFor3rdLA.getTime())
                              +"\t      "+NameTruck1+"#"+n+"   \tloaded at
"+NameLoader3_3LA);
                                    startRunIn8hrFor3rdLA.add(Calendar.MINUTE,
returnTimeTo3rdLA_3LA+
       (int)LoadingTimeLoader3_3LA+HaulingTimeLA3);
                              }
                              if(startRunIn8hrFor1stLA.after(finish8hr)) {break;}
                        }
                  }else{
                        out.println("\n\t\tWarning ! Incorrect insert. Please,
insert again.");
                  }
                  }
       }
}
```

ผลการประเมินภาษาจาวาสำหรับเหมืองหินปูน บริษัท ปูนซิเมนต์ไทย (ทุ่งสง) จำกัด

```
------ QUARRIES HAULAGE SYSTEM SIMULATION ON UBUNTU -------


      -Loading Areas-
      How many loading areas ? : 3

      -Trucks-
            Name the truck : CAT777
            How many CAT777s ? : 12
            Dump capacity of CAT777 (t) : 85
            Average haul velocity of CAT777 (km/h) : 21
            Average return velocity of CAT777 (km/h) : 25

            Dumping time of CAT777 in term of Normal
            Distribution.(If data not fit in Normal
            Distribution, you must fit its first. Suggestion
            software : Arena 13, Input Analyzer)
                  1. Normal Distribution, insert "N"
                  2. I don't know,        insert "-"
                  : N
            Normal format -- NORM(Mean, StdDev) --
                  Mean   = 1
                  StdDev = 0.2

      -Loaders-
            Name the loader at 1st loading area : EX1800
```

```
     Loading Time of EX1800 in term of Normal
     Distribution.
          1. Normal Distribution, insert "N"
          2. I don't know,        insert "-"
          : N
     Normal format -- NORM(Mean, StdDev) --
          Mean   = 3.8
          StdDev = 0.2

     Name the loader at 2nd loading area : CAT992#1
     Loading Time of CAT992#1 in term of Normal
     Distribution.
          1. Normal Distribution, insert "N"
          2. I don't know,        insert "-"
          : N
     Normal format -- NORM(Mean, StdDev) --
          Mean   = 4
          StdDev = 1

     Name the loader at 3rd loading area : CAT992#2
     Loading Time of CAT992#2 in term of Normal
     Distribution.
          1. Normal Distribution, insert "N"
          2. I don't know,        insert "-"
          : N
     Normal format -- NORM(Mean, StdDev) --
          Mean   = 4
          StdDev = 1

-Crushers-
     How many crushers ? : 2
     Name the 1st crusher : CR5
     Capacity of CR5 (t/h) : 1300
     Name the 2nd crusher : CR6
     Capacity of CR6 (t/h) : 1700

     Distance between dumping area and EX1800 (m.) :
     1000
     Distance between dumping area and CAT992#1 (m.) :
     1100
     Distance between dumping area and CAT992#2 (m.) :
     1200

     Would you like to view the system at loading areas
     or dumping areas ?
          1. Dumping Areas, insert "DA"
          2. Loading Areas, insert "LA"
          : LA
```

```
------------------- Start Time : 12:58 --------------------

OPERATION TIME     TRUCK              EVENT

--> Release 1st truck for each loading area
    12:58       CAT777#1      Depart dumping area
    12:58       CAT777#2      Depart dumping area
    12:58       CAT777#3      Depart dumping area
    13:00       CAT777#1      EX1800 start load
    13:00       CAT777#2      CAT992#1 start load
    13:00       CAT777#3      CAT992#2 start load
    13:03       CAT777#1      EX1800 loaded
    13:04       CAT777#2      CAT992#1 loaded
    13:03       CAT777#3      CAT992#2 loaded
    13:05       CAT777#1      start dump at CR5
    13:07       CAT777#2      start dump at CR5
    13:06       CAT777#3      start dump at CR5
    13:05       CAT777#1      dumped at     CR5
    13:08       CAT777#2      dumped at     CR5
    13:07       CAT777#3      dumped at     CR5
    13:07       CAT777#1      Arrive EX1800 again
    13:10       CAT777#2      Arrive CAT992#1 again
    13:09       CAT777#3      Arrive CAT992#2 again

--> To check, how many trucks optimization ?
    At 1st loading area
    13:01       CAT777#4      Depart dumping area
    13:03       CAT777#4      Arrive EX1800
    13:06       CAT777#7      Depart dumping area
    13:08       CAT777#7      Arrive EX1800
    --> The optimize number of trucks = 3
                         CAT777#1
                         CAT777#4
                         CAT777#7

    At 2nd loading area
    13:02       CAT777#5      Depart dumping area
    13:04       CAT777#5      Arrive CAT992#1
    13:08       CAT777#8      Depart dumping area
    13:10       CAT777#8      Arrive CAT992#1
    13:14       CAT777#11     Depart dumping area
    13:16       CAT777#11     Arrive CAT992#1
    --> The optimize number of trucks = 4
                         CAT777#2
                         CAT777#5
```

CAT777#8
CAT777#11

At 3rd loading area

13:02	CAT777#6	Depart dumping area
13:04	CAT777#6	Arrive CAT992#2
13:08	CAT777#9	Depart dumping area
13:10	CAT777#9	Arrive CAT992#2

--> The optimize number of trucks = 3
CAT777#3
CAT777#6
CAT777#9

---------- Trucks status in 8 hrs ----------

OPERATION TIME	TRUCK	EVENT
13:00	CAT777#1	start load at EX1800
13:03	CAT777#1	loaded at EX1800
13:00	CAT777#2	start load at CAT992#1
13:04	CAT777#2	loaded at CAT992#1
13:00	CAT777#3	start load at CAT992#2
13:04	CAT777#3	loaded at CAT992#2
13:07	CAT777#4	start load at EX1800
13:10	CAT777#4	loaded at EX1800
13:09	CAT777#5	start load at CAT992#1
13:13	CAT777#5	loaded at CAT992#1
13:13	CAT777#6	start load at CAT992#2
13:18	CAT777#6	loaded at CAT992#2
13:14	CAT777#7	start load at EX1800
13:18	CAT777#7	loaded at EX1800
13:18	CAT777#8	start load at CAT992#1
13:21	CAT777#8	loaded at CAT992#1
13:28	CAT777#9	start load at CAT992#2
13:30	CAT777#9	loaded at CAT992#2
13:22	CAT777#1	start load at EX1800
13:25	CAT777#1	loaded at EX1800
13:26	CAT777#2	start load at CAT992#1
13:31	CAT777#2	loaded at CAT992#1
13:37	CAT777#3	start load at CAT992#2
13:40	CAT777#3	loaded at CAT992#2
13:29	CAT777#4	start load at EX1800
13:32	CAT777#4	loaded at EX1800
13:36	CAT777#5	start load at CAT992#1
13:38	CAT777#5	loaded at CAT992#1
13:48	CAT777#6	start load at CAT992#2
13:52	CAT777#6	loaded at CAT992#2

13:36	CAT777#7	start load at	EX1800
13:39	CAT777#7	loaded at	EX1800
13:43	CAT777#8	start load at	CAT992#1
13:45	CAT777#8	loaded at	CAT992#1
14:01	CAT777#9	start load at	CAT992#2
14:05	CAT777#9	loaded at	CAT992#2
13:43	CAT777#1	start load at	EX1800
13:47	CAT777#1	loaded at	EX1800
13:50	CAT777#2	start load at	CAT992#1
13:53	CAT777#2	loaded at	CAT992#1
14:14	CAT777#3	start load at	CAT992#2
14:19	CAT777#3	loaded at	CAT992#2
13:51	CAT777#4	start load at	EX1800
13:54	CAT777#4	loaded at	EX1800
13:58	CAT777#5	start load at	CAT992#1
14:01	CAT777#5	loaded at	CAT992#1
14:29	CAT777#6	start load at	CAT992#2
14:31	CAT777#6	loaded at	CAT992#2
13:58	CAT777#7	start load at	EX1800
14:01	CAT777#7	loaded at	EX1800
14:06	CAT777#8	start load at	CAT992#1
14:09	CAT777#8	loaded at	CAT992#1
14:38	CAT777#9	start load at	CAT992#2
14:42	CAT777#9	loaded at	CAT992#2
14:05	CAT777#1	start load at	EX1800
14:08	CAT777#1	loaded at	EX1800
14:14	CAT777#2	start load at	CAT992#1
14:19	CAT777#2	loaded at	CAT992#1
14:51	CAT777#3	start load at	CAT992#2
14:54	CAT777#3	loaded at	CAT992#2
14:12	CAT777#4	start load at	EX1800
14:15	CAT777#4	loaded at	EX1800
14:24	CAT777#5	start load at	CAT992#1
14:28	CAT777#5	loaded at	CAT992#1
15:02	CAT777#6	start load at	CAT992#2
15:07	CAT777#6	loaded at	CAT992#2
14:19	CAT777#7	start load at	EX1800
14:22	CAT777#7	loaded at	EX1800
14:33	CAT777#8	start load at	CAT992#1
14:35	CAT777#8	loaded at	CAT992#1
15:17	CAT777#9	start load at	CAT992#2
15:20	CAT777#9	loaded at	CAT992#2
14:26	CAT777#1	start load at	EX1800
14:29	CAT777#1	loaded at	EX1800
14:40	CAT777#2	start load at	CAT992#1
14:42	CAT777#2	loaded at	CAT992#1
15:28	CAT777#3	start load at	CAT992#2

15:31	CAT777#3	loaded at	CAT992#2
14:33	CAT777#4	start load at	EX1800
14:36	CAT777#4	loaded at	EX1800
14:47	CAT777#5	start load at	CAT992#1
14:51	CAT777#5	loaded at	CAT992#1
15:39	CAT777#6	start load at	CAT992#2
15:43	CAT777#6	loaded at	CAT992#2
14:40	CAT777#7	start load at	EX1800
14:44	CAT777#7	loaded at	EX1800
14:56	CAT777#8	start load at	CAT992#1
14:59	CAT777#8	loaded at	CAT992#1
15:52	CAT777#9	start load at	CAT992#2
15:57	CAT777#9	loaded at	CAT992#2
14:48	CAT777#1	start load at	EX1800
14:51	CAT777#1	loaded at	EX1800
15:04	CAT777#2	start load at	CAT992#1
15:07	CAT777#2	loaded at	CAT992#1
16:07	CAT777#3	start load at	CAT992#2
16:11	CAT777#3	loaded at	CAT992#2
14:55	CAT777#4	start load at	EX1800
14:58	CAT777#4	loaded at	EX1800
15:12	CAT777#5	start load at	CAT992#1
15:16	CAT777#5	loaded at	CAT992#1
16:20	CAT777#6	start load at	CAT992#2
16:23	CAT777#6	loaded at	CAT992#2
15:02	CAT777#7	start load at	EX1800
15:05	CAT777#7	loaded at	EX1800
15:21	CAT777#8	start load at	CAT992#1
15:26	CAT777#8	loaded at	CAT992#1
16:31	CAT777#9	start load at	CAT992#2
16:33	CAT777#9	loaded at	CAT992#2
15:09	CAT777#1	start load at	EX1800
15:12	CAT777#1	loaded at	EX1800
15:31	CAT777#2	start load at	CAT992#1
15:33	CAT777#2	loaded at	CAT992#1
16:40	CAT777#3	start load at	CAT992#2
16:41	CAT777#3	loaded at	CAT992#2
15:16	CAT777#4	start load at	EX1800
15:19	CAT777#4	loaded at	EX1800
15:38	CAT777#5	start load at	CAT992#1
15:41	CAT777#5	loaded at	CAT992#1
16:47	CAT777#6	start load at	CAT992#2
16:50	CAT777#6	loaded at	CAT992#2
15:23	CAT777#7	start load at	EX1800
15:26	CAT777#7	loaded at	EX1800
15:46	CAT777#8	start load at	CAT992#1
15:50	CAT777#8	loaded at	CAT992#1
16:58	CAT777#9	start load at	CAT992#2

17:01	CAT777#9	loaded at	CAT992#2
15:30	CAT777#1	start load at	EX1800
15:33	CAT777#1	loaded at	EX1800
15:55	CAT777#2	start load at	CAT992#1
15:59	CAT777#2	loaded at	CAT992#1
17:09	CAT777#3	start load at	CAT992#2
17:10	CAT777#3	loaded at	CAT992#2
15:37	CAT777#4	start load at	EX1800
15:40	CAT777#4	loaded at	EX1800
16:04	CAT777#5	start load at	CAT992#1
16:08	CAT777#5	loaded at	CAT992#1
17:16	CAT777#6	start load at	CAT992#2
17:22	CAT777#6	loaded at	CAT992#2
15:44	CAT777#7	start load at	EX1800
15:47	CAT777#7	loaded at	EX1800
16:13	CAT777#8	start load at	CAT992#1
16:17	CAT777#8	loaded at	CAT992#1
17:33	CAT777#9	start load at	CAT992#2
17:37	CAT777#9	loaded at	CAT992#2
15:51	CAT777#1	start load at	EX1800
15:54	CAT777#1	loaded at	EX1800
16:22	CAT777#2	start load at	CAT992#1
16:26	CAT777#2	loaded at	CAT992#1
17:46	CAT777#3	start load at	CAT992#2
17:50	CAT777#3	loaded at	CAT992#2
15:58	CAT777#4	start load at	EX1800
16:01	CAT777#4	loaded at	EX1800
16:31	CAT777#5	start load at	CAT992#1
16:36	CAT777#5	loaded at	CAT992#1
17:59	CAT777#6	start load at	CAT992#2
18:03	CAT777#6	loaded at	CAT992#2
16:05	CAT777#7	start load at	EX1800
16:08	CAT777#7	loaded at	EX1800
16:41	CAT777#8	start load at	CAT992#1
16:45	CAT777#8	loaded at	CAT992#1
18:12	CAT777#9	start load at	CAT992#2
18:14	CAT777#9	loaded at	CAT992#2
16:12	CAT777#1	start load at	EX1800
16:15	CAT777#1	loaded at	EX1800
16:50	CAT777#2	start load at	CAT992#1
16:53	CAT777#2	loaded at	CAT992#1
18:21	CAT777#3	start load at	CAT992#2
18:23	CAT777#3	loaded at	CAT992#2
16:19	CAT777#4	start load at	EX1800
16:23	CAT777#4	loaded at	EX1800
16:58	CAT777#5	start load at	CAT992#1
17:02	CAT777#5	loaded at	CAT992#1
18:30	CAT777#6	start load at	CAT992#2

18:34	CAT777#6	loaded at	CAT992#2
16:27	CAT777#7	start load at	EX1800
16:30	CAT777#7	loaded at	EX1800
17:07	CAT777#8	start load at	CAT992#1
17:09	CAT777#8	loaded at	CAT992#1
18:43	CAT777#9	start load at	CAT992#2
18:48	CAT777#9	loaded at	CAT992#2
16:34	CAT777#1	start load at	EX1800
16:38	CAT777#1	loaded at	EX1800
17:14	CAT777#2	start load at	CAT992#1
17:18	CAT777#2	loaded at	CAT992#1
18:58	CAT777#3	start load at	CAT992#2
19:02	CAT777#3	loaded at	CAT992#2
16:42	CAT777#4	start load at	EX1800
16:45	CAT777#4	loaded at	EX1800
17:23	CAT777#5	start load at	CAT992#1
17:27	CAT777#5	loaded at	CAT992#1
19:11	CAT777#6	start load at	CAT992#2
19:14	CAT777#6	loaded at	CAT992#2
16:49	CAT777#7	start load at	EX1800
16:52	CAT777#7	loaded at	EX1800
17:32	CAT777#8	start load at	CAT992#1
17:36	CAT777#8	loaded at	CAT992#1
19:22	CAT777#9	start load at	CAT992#2
19:25	CAT777#9	loaded at	CAT992#2
16:56	CAT777#1	start load at	EX1800
17:00	CAT777#1	loaded at	EX1800
17:41	CAT777#2	start load at	CAT992#1
17:44	CAT777#2	loaded at	CAT992#1
19:33	CAT777#3	start load at	CAT992#2
19:36	CAT777#3	loaded at	CAT992#2
17:04	CAT777#4	start load at	EX1800
17:07	CAT777#4	loaded at	EX1800
17:49	CAT777#5	start load at	CAT992#1
17:53	CAT777#5	loaded at	CAT992#1
19:44	CAT777#6	start load at	CAT992#2
19:49	CAT777#6	loaded at	CAT992#2
17:11	CAT777#7	start load at	EX1800
17:14	CAT777#7	loaded at	EX1800
17:58	CAT777#8	start load at	CAT992#1
18:02	CAT777#8	loaded at	CAT992#1
19:59	CAT777#9	start load at	CAT992#2
20:02	CAT777#9	loaded at	CAT992#2
17:18	CAT777#1	start load at	EX1800
17:21	CAT777#1	loaded at	EX1800
18:07	CAT777#2	start load at	CAT992#1
18:11	CAT777#2	loaded at	CAT992#1
20:10	CAT777#3	start load at	CAT992#2

20:13	CAT777#3	loaded at	CAT992#2
17:25	CAT777#4	start load at	EX1800
17:28	CAT777#4	loaded at	EX1800
18:16	CAT777#5	start load at	CAT992#1
18:20	CAT777#5	loaded at	CAT992#1
20:21	CAT777#6	start load at	CAT992#2
20:22	CAT777#6	loaded at	CAT992#2
17:32	CAT777#7	start load at	EX1800
17:35	CAT777#7	loaded at	EX1800
18:25	CAT777#8	start load at	CAT992#1
18:28	CAT777#8	loaded at	CAT992#1
20:28	CAT777#9	start load at	CAT992#2
20:32	CAT777#9	loaded at	CAT992#2
17:39	CAT777#1	start load at	EX1800
17:42	CAT777#1	loaded at	EX1800
18:33	CAT777#2	start load at	CAT992#1
18:35	CAT777#2	loaded at	CAT992#1
20:41	CAT777#3	start load at	CAT992#2
20:43	CAT777#3	loaded at	CAT992#2
17:46	CAT777#4	start load at	EX1800
17:49	CAT777#4	loaded at	EX1800
18:40	CAT777#5	start load at	CAT992#1
18:43	CAT777#5	loaded at	CAT992#1
20:50	CAT777#6	start load at	CAT992#2
20:55	CAT777#6	loaded at	CAT992#2
17:53	CAT777#7	start load at	EX1800
17:56	CAT777#7	loaded at	EX1800
18:48	CAT777#8	start load at	CAT992#1
18:52	CAT777#8	loaded at	CAT992#1
21:05	CAT777#9	start load at	CAT992#2
21:06	CAT777#9	loaded at	CAT992#2
18:00	CAT777#1	start load at	EX1800
18:03	CAT777#1	loaded at	EX1800
18:57	CAT777#2	start load at	CAT992#1
19:01	CAT777#2	loaded at	CAT992#1
21:12	CAT777#3	start load at	CAT992#2
21:15	CAT777#3	loaded at	CAT992#2
18:07	CAT777#4	start load at	EX1800
18:10	CAT777#4	loaded at	EX1800
19:06	CAT777#5	start load at	CAT992#1
19:09	CAT777#5	loaded at	CAT992#1
21:23	CAT777#6	start load at	CAT992#2
21:27	CAT777#6	loaded at	CAT992#2
18:14	CAT777#7	start load at	EX1800
18:17	CAT777#7	loaded at	EX1800
19:14	CAT777#8	start load at	CAT992#1
19:17	CAT777#8	loaded at	CAT992#1
21:36	CAT777#9	start load at	CAT992#2

21:40	CAT777#9	loaded at	CAT992#2
18:21	CAT777#1	start load at	EX1800
18:24	CAT777#1	loaded at	EX1800
19:22	CAT777#2	start load at	CAT992#1
19:26	CAT777#2	loaded at	CAT992#1
21:49	CAT777#3	start load at	CAT992#2
21:53	CAT777#3	loaded at	CAT992#2
18:28	CAT777#4	start load at	EX1800
18:31	CAT777#4	loaded at	EX1800
19:31	CAT777#5	start load at	CAT992#1
19:36	CAT777#5	loaded at	CAT992#1
22:02	CAT777#6	start load at	CAT992#2
22:06	CAT777#6	loaded at	CAT992#2
18:35	CAT777#7	start load at	EX1800
18:38	CAT777#7	loaded at	EX1800
19:41	CAT777#8	start load at	CAT992#1
19:45	CAT777#8	loaded at	CAT992#1
22:15	CAT777#9	start load at	CAT992#2
22:20	CAT777#9	loaded at	CAT992#2
18:42	CAT777#1	start load at	EX1800
18:46	CAT777#1	loaded at	EX1800
19:50	CAT777#2	start load at	CAT992#1
19:55	CAT777#2	loaded at	CAT992#1
22:30	CAT777#3	start load at	CAT992#2
22:34	CAT777#3	loaded at	CAT992#2
18:50	CAT777#4	start load at	EX1800
18:54	CAT777#4	loaded at	EX1800
20:00	CAT777#5	start load at	CAT992#1
20:04	CAT777#5	loaded at	CAT992#1
22:43	CAT777#6	start load at	CAT992#2
22:46	CAT777#6	loaded at	CAT992#2
18:58	CAT777#7	start load at	EX1800
19:01	CAT777#7	loaded at	EX1800
20:09	CAT777#8	start load at	CAT992#1
20:13	CAT777#8	loaded at	CAT992#1
22:54	CAT777#9	start load at	CAT992#2
22:57	CAT777#9	loaded at	CAT992#2
19:05	CAT777#1	start load at	EX1800
19:08	CAT777#1	loaded at	EX1800
20:18	CAT777#2	start load at	CAT992#1
20:22	CAT777#2	loaded at	CAT992#1
23:05	CAT777#3	start load at	CAT992#2
23:09	CAT777#3	loaded at	CAT992#2
19:12	CAT777#4	start load at	EX1800
19:15	CAT777#4	loaded at	EX1800
20:27	CAT777#5	start load at	CAT992#1
20:32	CAT777#5	loaded at	CAT992#1
23:18	CAT777#6	start load at	CAT992#2

23:21	CAT777#6	loaded at	CAT992#2
19:19	CAT777#7	start load at	EX1800
19:22	CAT777#7	loaded at	EX1800
20:37	CAT777#8	start load at	CAT992#1
20:40	CAT777#8	loaded at	CAT992#1
23:29	CAT777#9	start load at	CAT992#2
23:32	CAT777#9	loaded at	CAT992#2
19:26	CAT777#1	start load at	EX1800
19:29	CAT777#1	loaded at	EX1800
20:45	CAT777#2	start load at	CAT992#1
20:48	CAT777#2	loaded at	CAT992#1
23:40	CAT777#3	start load at	CAT992#2
23:43	CAT777#3	loaded at	CAT992#2
19:33	CAT777#4	start load at	EX1800
19:36	CAT777#4	loaded at	EX1800
20:53	CAT777#5	start load at	CAT992#1
20:58	CAT777#5	loaded at	CAT992#1
23:51	CAT777#6	start load at	CAT992#2
23:53	CAT777#6	loaded at	CAT992#2
19:40	CAT777#7	start load at	EX1800
19:43	CAT777#7	loaded at	EX1800
21:03	CAT777#8	start load at	CAT992#1
21:09	CAT777#8	loaded at	CAT992#1
00:00	CAT777#9	start load at	CAT992#2
00:03	CAT777#9	loaded at	CAT992#2
19:47	CAT777#1	start load at	EX1800
19:50	CAT777#1	loaded at	EX1800
21:14	CAT777#2	start load at	CAT992#1
21:18	CAT777#2	loaded at	CAT992#1
00:11	CAT777#3	start load at	CAT992#2
00:14	CAT777#3	loaded at	CAT992#2
19:54	CAT777#4	start load at	EX1800
19:57	CAT777#4	loaded at	EX1800
21:23	CAT777#5	start load at	CAT992#1
21:25	CAT777#5	loaded at	CAT992#1
00:22	CAT777#6	start load at	CAT992#2
00:26	CAT777#6	loaded at	CAT992#2
20:01	CAT777#7	start load at	EX1800
20:04	CAT777#7	loaded at	EX1800
21:30	CAT777#8	start load at	CAT992#1
21:33	CAT777#8	loaded at	CAT992#1
00:35	CAT777#9	start load at	CAT992#2
00:37	CAT777#9	loaded at	CAT992#2
20:08	CAT777#1	start load at	EX1800
20:12	CAT777#1	loaded at	EX1800
21:38	CAT777#2	start load at	CAT992#1
21:40	CAT777#2	loaded at	CAT992#1
00:44	CAT777#3	start load at	CAT992#2

00:47	CAT777#3	loaded at	CAT992#2
20:16	CAT777#4	start load at	EX1800
20:19	CAT777#4	loaded at	EX1800
21:45	CAT777#5	start load at	CAT992#1
21:49	CAT777#5	loaded at	CAT992#1
00:55	CAT777#6	start load at	CAT992#2
00:59	CAT777#6	loaded at	CAT992#2
20:23	CAT777#7	start load at	EX1800
20:27	CAT777#7	loaded at	EX1800
21:54	CAT777#8	start load at	CAT992#1
21:58	CAT777#8	loaded at	CAT992#1
01:08	CAT777#9	start load at	CAT992#2
01:11	CAT777#9	loaded at	CAT992#2
20:31	CAT777#1	start load at	EX1800
20:34	CAT777#1	loaded at	EX1800
22:03	CAT777#2	start load at	CAT992#1
22:06	CAT777#2	loaded at	CAT992#1
01:19	CAT777#3	start load at	CAT992#2
01:23	CAT777#3	loaded at	CAT992#2
20:38	CAT777#4	start load at	EX1800
20:41	CAT777#4	loaded at	EX1800
22:11	CAT777#5	start load at	CAT992#1
22:14	CAT777#5	loaded at	CAT992#1
01:32	CAT777#6	start load at	CAT992#2
01:36	CAT777#6	loaded at	CAT992#2
20:45	CAT777#7	start load at	EX1800
20:49	CAT777#7	loaded at	EX1800
22:19	CAT777#8	start load at	CAT992#1
22:22	CAT777#8	loaded at	CAT992#1
01:45	CAT777#9	start load at	CAT992#2
01:50	CAT777#9	loaded at	CAT992#2
20:53	CAT777#1	start load at	EX1800
20:56	CAT777#1	loaded at	EX1800
22:27	CAT777#2	start load at	CAT992#1
22:30	CAT777#2	loaded at	CAT992#1
02:00	CAT777#3	start load at	CAT992#2
02:05	CAT777#3	loaded at	CAT992#2
21:00	CAT777#4	start load at	EX1800
21:04	CAT777#4	loaded at	EX1800
22:35	CAT777#5	start load at	CAT992#1
22:40	CAT777#5	loaded at	CAT992#1
02:15	CAT777#6	start load at	CAT992#2
02:18	CAT777#6	loaded at	CAT992#2
21:08	CAT777#7	start load at	EX1800
21:11	CAT777#7	loaded at	EX1800
22:45	CAT777#8	start load at	CAT992#1
22:48	CAT777#8	loaded at	CAT992#1
02:26	CAT777#9	start load at	CAT992#2

```
02:29       CAT777#9      loaded at     CAT992#2
```

ผลการประมินภาษาจาวาสำหรับเหมืองหินปูน บริษัท ปูนซิเมนต์ไทย (ท่าหลวง) จำกัด

```
------ QUARRIES HAULAGE SYSTEM SIMULATION ON UBUNTU -------

     -Loading Areas-
          How many loading areas ? : 2

     -Trucks-
          Name the truck : CAT777
          How many CAT777s ? : 12
          Dump capacity of CAT777 (t) : 85
          Average haul velocity of CAT777 (km/h) : 21
          Average return velocity of CAT777 (km/h) : 25

          Dumping time of CAT777 in term of Normal
          Distribution.(If data not fit in Normal
          Distribution, you must fit its first. Suggestion
          software : Arena 13, Input Analyzer)
               1. Normal Distribution, insert "N"
               2. I don't know,        insert "-"
               : N
          Normal format -- NORM(Mean, StdDev) --
               Mean   = 1
               StdDev = 0.2

     -Loaders-
          Name the loader at 1st loading area : EX1800
          Loading Time of EX1800 in term of Normal
          Distribution.
               1. Normal Distribution, insert "N"
               2. I don't know,        insert "-"
               : N
          Normal format -- NORM(Mean, StdDev) --
               Mean   = 3.8
               StdDev = 0.2

          Name the loader at 2nd loading area : CAT992
          Loading Time of CAT992#1 in term of Normal
          Distribution.
               1. Normal Distribution, insert "N"
```

```
                2. I don't know,        insert "-"
                : N
           Normal format -- NORM(Mean, StdDev) --
                Mean   = 4
                StdDev = 0.2

     -Crushers-
           How many crushers ? : 1
           Name the 1st crusher : TL
           Capacity of CR5 (t/h) : 1800

           Distance between dumping area and EX1800 (m.) :
           2000
           Distance between dumping area and CAT992 (m.) :
           1800

           Would you like to view the system at loading areas
           or dumping areas ?
                1. Dumping Areas, insert "DA"
                2. Loading Areas, insert "LA"
                : DA

------------------- Start Time : 15:20 --------------------

OPERATION TIME     TRUCK               EVENT

--> Release 1st truck for each loading area
    15:20       CAT777#1      Depart dumping area
    15:20       CAT777#2      Depart dumping area
    15:24       CAT777#1      EX1800 start load
    15:24       CAT777#2      CAT992 start load
    15:27       CAT777#1      EX1800 loaded
    15:28       CAT777#2      CAT992 loaded
    15:32       CAT777#1      start dump at TL
    15:33       CAT777#2      start dump at TL
    15:32       CAT777#1      dumped at     TL
    15:33       CAT777#2      dumped at     TL
    15:36       CAT777#1      Arrive EX1800 again
    15:37       CAT777#2      Arrive CAT992 again

--> To check, how many trucks optimization ?
    At 1st loading area
    15:23       CAT777#3      Depart dumping area
    15:27       CAT777#3      Arrive EX1800
    15:30       CAT777#5      Depart dumping area
    15:34       CAT777#5      Arrive EX1800
    15:37       CAT777#7      Depart dumping area
```

```
    15:41       CAT777#7      Arrive EX1800
    --> The optimize number of trucks = 4
                          CAT777#1
                          CAT777#3
                          CAT777#5
                          CAT777#7

    At 2nd loading area
    15:24       CAT777#4      Depart dumping area
    15:28       CAT777#4      Arrive CAT992
    15:32       CAT777#6      Depart dumping area
    15:36       CAT777#6      Arrive CAT992
    15:40       CAT777#8      Depart dumping area
    15:44       CAT777#8      Arrive CAT992
    --> The optimize number of trucks = 4
                          CAT777#2
                          CAT777#4
                          CAT777#6
                          CAT777#8
------------------ Trucks status in 8 hrs -----------------

OPERATION TIME      TRUCK               EVENT

    15:32       CAT777#1      start dump at TL
    15:33       CAT777#1      dumped at     TL
    15:33       CAT777#2      start dump at TL
    15:34       CAT777#2      dumped at     TL
    15:45       CAT777#3      start dump at TL
    15:46       CAT777#3      dumped at     TL
    15:47       CAT777#4      start dump at TL
    15:47       CAT777#4      dumped at     TL
    15:58       CAT777#5      start dump at TL
    15:58       CAT777#5      dumped at     TL
    16:00       CAT777#6      start dump at TL
    16:01       CAT777#6      dumped at     TL
    16:10       CAT777#7      start dump at TL
    16:10       CAT777#7      dumped at     TL
    16:14       CAT777#8      start dump at TL
    16:15       CAT777#8      dumped at     TL
    16:22       CAT777#1      start dump at TL
    16:22       CAT777#1      dumped at     TL
    16:28       CAT777#2      start dump at TL
    16:29       CAT777#2      dumped at     TL
    16:34       CAT777#3      start dump at TL
    16:34       CAT777#3      dumped at     TL
    16:42       CAT777#4      start dump at TL
    16:43       CAT777#4      dumped at     TL
    16:46       CAT777#5      start dump at TL
```

16:47	CAT777#5	dumped at	TL
16:56	CAT777#6	start dump at	TL
16:56	CAT777#6	dumped at	TL
16:59	CAT777#7	start dump at	TL
16:59	CAT777#7	dumped at	TL
17:09	CAT777#8	start dump at	TL
17:10	CAT777#8	dumped at	TL
17:11	CAT777#1	start dump at	TL
17:12	CAT777#1	dumped at	TL
17:23	CAT777#2	start dump at	TL
17:24	CAT777#2	dumped at	TL
17:24	CAT777#3	start dump at	TL
17:25	CAT777#3	dumped at	TL
17:37	CAT777#4	start dump at	TL
17:37	CAT777#4	dumped at	TL
17:37	CAT777#5	start dump at	TL
17:37	CAT777#5	dumped at	TL
17:50	CAT777#6	start dump at	TL
17:51	CAT777#6	dumped at	TL
17:49	CAT777#7	start dump at	TL
17:50	CAT777#7	dumped at	TL
18:04	CAT777#8	start dump at	TL
18:04	CAT777#8	dumped at	TL
18:02	CAT777#1	start dump at	TL
18:03	CAT777#1	dumped at	TL
18:17	CAT777#2	start dump at	TL
18:18	CAT777#2	dumped at	TL
18:15	CAT777#3	start dump at	TL
18:16	CAT777#3	dumped at	TL
18:31	CAT777#4	start dump at	TL
18:32	CAT777#4	dumped at	TL
18:28	CAT777#5	start dump at	TL
18:28	CAT777#5	dumped at	TL
18:45	CAT777#6	start dump at	TL
18:46	CAT777#6	dumped at	TL
18:40	CAT777#7	start dump at	TL
18:40	CAT777#7	dumped at	TL
18:59	CAT777#8	start dump at	TL
19:00	CAT777#8	dumped at	TL
18:52	CAT777#1	start dump at	TL
18:53	CAT777#1	dumped at	TL
19:13	CAT777#2	start dump at	TL
19:13	CAT777#2	dumped at	TL
19:05	CAT777#3	start dump at	TL
19:06	CAT777#3	dumped at	TL
19:26	CAT777#4	start dump at	TL
19:26	CAT777#4	dumped at	TL
19:18	CAT777#5	start dump at	TL

19:18	CAT777#5	dumped at	TL
19:39	CAT777#6	start dump at	TL
19:39	CAT777#6	dumped at	TL
19:30	CAT777#7	start dump at	TL
19:31	CAT777#7	dumped at	TL
19:52	CAT777#8	start dump at	TL
19:53	CAT777#8	dumped at	TL
19:43	CAT777#1	start dump at	TL
19:44	CAT777#1	dumped at	TL
20:06	CAT777#2	start dump at	TL
20:07	CAT777#2	dumped at	TL
19:56	CAT777#3	start dump at	TL
19:56	CAT777#3	dumped at	TL
20:20	CAT777#4	start dump at	TL
20:20	CAT777#4	dumped at	TL
20:08	CAT777#5	start dump at	TL
20:08	CAT777#5	dumped at	TL
20:33	CAT777#6	start dump at	TL
20:33	CAT777#6	dumped at	TL
20:20	CAT777#7	start dump at	TL
20:20	CAT777#7	dumped at	TL
20:46	CAT777#8	start dump at	TL
20:47	CAT777#8	dumped at	TL
20:32	CAT777#1	start dump at	TL
20:32	CAT777#1	dumped at	TL
21:00	CAT777#2	start dump at	TL
21:00	CAT777#2	dumped at	TL
20:44	CAT777#3	start dump at	TL
20:44	CAT777#3	dumped at	TL
21:13	CAT777#4	start dump at	TL
21:14	CAT777#4	dumped at	TL
20:56	CAT777#5	start dump at	TL
20:56	CAT777#5	dumped at	TL
21:27	CAT777#6	start dump at	TL
21:28	CAT777#6	dumped at	TL
21:08	CAT777#7	start dump at	TL
21:08	CAT777#7	dumped at	TL
21:41	CAT777#8	start dump at	TL
21:41	CAT777#8	dumped at	TL
21:20	CAT777#1	start dump at	TL
21:20	CAT777#1	dumped at	TL
21:54	CAT777#2	start dump at	TL
21:55	CAT777#2	dumped at	TL
21:32	CAT777#3	start dump at	TL
21:32	CAT777#3	dumped at	TL
22:08	CAT777#4	start dump at	TL
22:08	CAT777#4	dumped at	TL
21:44	CAT777#5	start dump at	TL

21:44	CAT777#5	dumped at	TL
22:21	CAT777#6	start dump at	TL
22:22	CAT777#6	dumped at	TL
21:56	CAT777#7	start dump at	TL
21:56	CAT777#7	dumped at	TL
22:35	CAT777#8	start dump at	TL
22:36	CAT777#8	dumped at	TL
22:08	CAT777#1	start dump at	TL
22:09	CAT777#1	dumped at	TL
22:49	CAT777#2	start dump at	TL
22:50	CAT777#2	dumped at	TL
22:21	CAT777#3	start dump at	TL
22:21	CAT777#3	dumped at	TL
23:03	CAT777#4	start dump at	TL
23:03	CAT777#4	dumped at	TL
22:33	CAT777#5	start dump at	TL
22:33	CAT777#5	dumped at	TL
23:16	CAT777#6	start dump at	TL
23:17	CAT777#6	dumped at	TL
22:45	CAT777#7	start dump at	TL
22:45	CAT777#7	dumped at	TL
23:30	CAT777#8	start dump at	TL
23:30	CAT777#8	dumped at	TL