การแปลงกระแสงานยอวล์เป็นโครงร่างบีเพล

นายสิทธิพงศ์ พรอุดมทรัพย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

TRANSFORMING YAWL WORKFLOW TO BPEL SKELETON

Mr.Sittiphong Pornudomthap

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2011

Copyright of Chulalongkorn University

สิทธิพงศ์ พรอุดมทรัพย์ : การแปลงกระแสงานยอวล์เป็นโครงร่างบีเพล. (TRANSFORMING YAWL WORKFLOW TO BPEL SKELETON) อ. ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. วิวัฒน์ วัฒนาวุฒิ, 138 หน้า.

องค์กรภาคธุรกิจในปัจจุบันมีความนิยมนำเว็บเซอร์วิสมาใช้งานผ่านภาษาบีเพล และกระบวนการทางธุรกิจในปัจจุบันมีความซับซ้อนมากขึ้น การใช้เพียงภาษาบีเพลไม่สามารถอธิบายกระบวนการทางธุรกิจเหล่านั้นได้ครบถ้วน ดังนั้นจึงมีการนำภาษายอวล์มาใช้เพื่ออธิบายกระบวนการทางธุรกิจที่มีความซับซ้อนแทน แต่ในปัจจุบันภาษายอวล์ยังไม่ได้รับความนิยมกว้างขวางนักเพราะภาษายอวล์พัฒนาขึ้นได้ไม่นานและยังขาดเครื่องมือที่สนับสนุนการเรียกใช้บริการระหว่างกัน

วิทยานิพนธ์นี้มุ่งศึกษาและพัฒนาเครื่องมือในการเปลี่ยนแบบรูปจากภาษายอวล์เป็นโครงร่างภาษาบีเพล เพื่อให้สามารถสร้างกระบวนการทางธุรกิจที่มีความซับซ้อนเป็นบริการใหม่ได้สะดวกมากยิ่งขึ้น และเป็นการช่วยนักพัฒนาเปลี่ยนกระบวนการทางธุรกิจในรูปกระแสงานยอวล์ ให้อยู่ในรูปของโครงร่างภาษาบีเพลเพื่อใช้ในการพัฒนาระบบเว็บเซอร์วิสได้อย่างถูกต้องสมบูรณ์ รวมถึงการแก้ไขปัญหากระแสงานที่มีโครงสร้างไม่ดีโดยการปรับปรุงโครงสร้างกระแสงานใหม่

เครื่องมือในการเปลี่ยนรูปกระแสงานที่พัฒนาขึ้นโดยรองรับแบบรูปในภาษายอวล์ 19 แบบรูป ให้สามารถเปลี่ยนเป็นโครงร่างบีเพล ที่มีผลลำดับการทำงานเหมือนกัน และสามารถปรับกระแสงานยอวล์ที่มีโครงสร้างที่ไม่ดี เป็นโครงร่างบีเพลโดยใช้แนวคิดของอัลกอริทึมแฮมมอค

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2554

5170496621 : MAJOR SOFTWARE ENGINEERING

KEYWORDS : BPEL / PATTERN / WEB SERVICE / WORKFLOW / YAWL

SITTIPHONG PORNUDOMTHAP : TRANSFORMING YAWL WORKFLOW TO BPEL SKELETON. ADVISOR : ASSOC. PROF. WIWAT VATANAWOOD,Ph.D., 138 pp.

Currently it is common to build business process using web services and BPEL is the common process execution language. Sometimes, BPEL is not completely practical for the complex business processes. Therefore, YAWL is proposed recently as an alternative to the complex business process description language. However, YAWL is still new and not popular among developers. And the YAWL supporting tool is considered rare.

This thesis studied and developed a tool to transform YAWL into BPEL skeleton. This approach guides and eases the complete transformation of available complex business processes into corresponding services. We provide the developer with the transformation of business processes written in YAWL into BPEL skeleton. The transformation of non well structured patterns of business processes in YAWL are also proposed. They are expected to be restructured into the new well structured ones.

The YAWL2BPEL transformation tool is developed to support 19 business process patterns and the BPEL skeleton codes are correspondingly generated. The resulting BPEL skeleton codes are consistent to the original YAWL. Hammock algorithm is used to restructure the non well structured YAWL.

Department : Computer Engineering Student's Signature .....

Field of Study : Software Engineering Advisor's Signature .....

Academic Year : 2011

หัวข้อวิทยานิพนธ์	การแปลงกระแสงานยอวล์เป็นโครงร่างบีเพล
โดย	นายสิทธิพงศ์ พรอุดมทรัพย์
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒิ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร. บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร. ธาราทิพย์ สุวรรณศาสตร์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒิ)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. อาทิตย์ ทองทักษ์)

..... กรรมการภายนอกมหาวิทยาลัย
(ดร. เนื่องวงศ์ ทวยเจริญ)

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี เนื่องจากความกรุณาของผู้เกี่ยวข้องทุกฝ่าย ผู้เขียนขอกราบขอบพระคุณ รศ.ดร. วิวัฒน์ วัฒนาวุฒิ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่กรุณาเอาใจใส่ มอบความรู้ คำแนะนำต่างๆ ที่เป็นประโยชน์อย่างยิ่งต่อวิทยานิพนธ์ รวมทั้งตรวจแก้ไขข้อบกพร่องให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์ยิ่งขึ้น

ขอกราบขอบพระคุณ รศ.ดร. ธาราทิพย์ สุวรรณศาสตร์ ที่กรุณาสละเวลามาเป็นประธานกรรมการการสอบวิทยานิพนธ์ ขอกราบขอบพระคุณ ผศ.ดร. อาทิตย์ ทองทักษ์ ดร.เนื่องวงศ์ ทวยเจริญ ที่กรุณาสละเวลามาเป็นกรรมการการสอบวิทยานิพนธ์รวมทั้งให้คำแนะนำและแก้ไขข้อบกพร่องให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์ยิ่งขึ้น

ขอบคุณ คุณทวี ไทยส่งสุวรรณ สำหรับคำแนะนำการทำงานภาษาบีเพลและเพื่อนๆวิศวกรรมซอฟต์แวร์ทุกคน ที่คอยช่วยเหลือ แก้ไขปัญหา ให้คำปรึกษา และให้กำลังใจจนวิทยานิพนธ์เล่มนี้สำเร็จ

สุดท้ายขอกราบขอบพระคุณ คุณชำนาญและคุณปราณี พรอุดมทรัพย์ คุณพ่อคุณแม่ที่แสนดี และทุกคนในครอบครัว ที่ให้ทุกสิ่งทุกอย่าง ทั้งความรัก ความห่วงใย และกำลังใจมาตลอด

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

บทที่ 1 บทนำ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์..... 2

1.3 ขอบเขตการดำเนินงาน..... 2

1.4 ขั้นตอนการวิจัย..... 3

1.5 ลำดับขั้นตอนการเสนอผลการวิจัย..... 3

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์..... 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 5

2.1 ภาษาบีเพล..... 5

2.1.1 การใช้งานของภาษาบีเพล..... 5

2.1.2 คำสั่งของภาษาบีเพล..... 6

2.1.3 เซิร์ฟเวอร์บีเพล (BPEL Server)..... 7

2.2 กระแสงานยอวล์..... 7

2.2.1 ลักษณะพื้นฐานของกระแสงานยอวล์..... 7

2.2.2 องค์ประกอบของภาษายอวล์..... 10

2.3 แบบรูปของกระแสงานยอวล์ 13

2.3.1 แบบรูปกระแสงานควบคุมพื้นฐาน..... 13

2.3.1.1 แบบรูป Sequence..... 13

2.3.1.2 แบบรูป Parallel Split..... 14

2.3.1.3 แบบรูป Synchronization..... 14

2.3.1.4 แบบรูป Exclusive Choice..... 15

2.3.1.5 แบบรูป Simple Merge..... 15

2.3.2 แบบรูปการแยกและผสานกระแสงานขั้นสูง..... 16

บทที่ หน้า

2.3.2.1 แบบรูป Multi-Choice…….. 16
2.3.2.2 แบบรูป Structured Synchronizing Merge…… 17
2.3.2.3 แบบรูป Multi-Merge…… 17
2.3.2.4 แบบรูป Structured Discriminator…….. 18
2.3.3 แบบรูปกลุ่มโครงสร้าง…… 18
2.3.3.1 แบบรูป Arbitrary Cycles…….. 19
2.3.4 แบบรูปหลายอ็อบเจกต์……..…. 19
2.3.4.1 แบบรูป Multiple Instances without Synchronization 20
2.3.4.2 แบบรูป Multiple Instances with a priori Design-Time Knowledge …… 20
2.3.4.3 แบบรูป Multiple Instances with a priori Run-Time Knowledge …… 20
2.3.4.4 แบบรูป Multiple instances without a priori run-time knowledge ……..... 21
2.3.5 แบบรูปสถานะ…….. 22
2.3.5.1 แบบรูป Deferred Choice…….. 22
2.3.5.2 แบบรูป Interleaved Parallel Routing……. 23
2.3.5.3 แบบรูป Milestone…….. 23
2.3.6 แบบรูปการยกเลิก (Cancellation Patterns)…….. 24
2.3.6.1 แบบรูป Cancel Activity……. 24
2.3.6.2 แบบรูป Cancel Case……. 25
2.4 อัลกอริทึมแฮมมอคกราฟ…….. 25
2.4.1 หลักการการทำงานอัลกอริทึม…….…… 26
2.4.2 ขั้นตอนการทำงานของอัลกอริทึม……. 27
2.5 ลักษณะแบบรูปในกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล……... 28
2.6 งานวิจัยที่เกี่ยวข้อง 31
2.6.1 งานวิจัย Transformation of BPMN to YAWL ของ JianHong และคณะ…… 31
2.6.2 งานวิจัย From BPMN Process Models to BPEL Web

บทที่ หน้า

Services ของ Ouyang และคณะ..... 31
2.6.3 งานวิจัย Pattern-based Translation of BPMN Process Models to BPEL Web Services ของ Ouyang และคณะ..... 31
2.6.4 งานวิจัย Using Hammock Graphs to Structure Programs ของ Zhang และคณะ..... 32
2.6.5 งานวิจัย Formal Semantics of BPMN Process Models Using YAWL ของ JianHong และคณะ..... 32
บทที่ 3 การวิเคราะห์และออกแบบวิธีการแปลงรูปกระแสงานยอวล์..... 33
3.1 ภาพรวมแนวคิดของเครื่องมือ..... 33
3.1.1 แบบรูปโครงสร้างที่ดี 34
3.1.2 แบบรูปโครงสร้างที่ไม่ดี 42
3.2 อัลกอริทึมการทำงาน Fold function..... 43
3.3 โครงสร้างการทำงานของเครื่องมือ..... 44
3.4 การวิเคราะห์และออกแบบเครื่องมือ..... 45
3.4.1 แผนภาพยูสเคส..... 46
3.4.2 แผนภาพคลาส..... 50
3.4.3 แผนภาพลำดับ..... 56
บทที่ 4 การพัฒนาเครื่องมือ..... 63
4.1 สภาพแวดล้อมในการพัฒนาระบบ..... 63
4.1.1 สภาพแวดล้อมในการพัฒนาเครื่องมือฮาร์ดแวร์..... 63
4.1.2 สภาพแวดล้อมในการพัฒนาเครื่องมือด้านซอฟต์แวร์..... 63
4.2 โครงสร้างส่วนต่อประสานกับผู้ใช้ของเครื่องมือ..... 63
บทที่ 5 การทดสอบ..... 68
5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ..... 68
5.2 ขั้นตอนการทดสอบเครื่องมือ..... 68
5.3 กระแสงานที่ใช้ในการทดสอบ..... 68
5.3.1 ระบบเช่ายืมหนังสือ..... 68
5.3.2 ระบบกู้ชำระเงิน..... 65
5.3.3 ระบบการลงทะเบียนเรียน 78

บทที่ หน้า

5.3.4 ระบบการสอบ..... 82
5.3.5 ระบบรับพนักงานใหม่ 84
5.3.6 ระบบสั่งซื้อหนังสือ..... 87
5.4 สรุปผลการทดสอบ..... 90
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ..... 91
6.1 สรุปผลการวิจัย..... 91
6.2 ข้อจำกัดของงานวิจัย..... 91
6.3 ข้อเสนอแนะ..... 91
6.4 ประโยชน์ที่ได้รับ..... 92
รายการอ้างอิง..... 93
ภาคผนวก..... 95
ประวัติผู้เขียนวิทยานิพนธ์..... 138

สารบัญตาราง

ตารางที่		หน้า
2.1	รายละเอียดสัญลักษณ์ในภาษายอวล์……..	11
2.2	แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล…….	28
3.1	ปรับปรุงโครงร่างบีเพลแบบรูป MI without Synchronization……..	40
3.2	ปรับปรุงโครงร่างบีเพลแบบรูป Deferred Choice……..	40
3.3	ปรับปรุงโครงร่างบีเพลแบบรูป Cancel Activity ……..	41
3.4	รายละเอียดยูสเคสนำเข้ากระแสงานยอวล์……..	47
3.5	รายละเอียดยูสเคสอ่านและตรวจสอบกระแสงานยอวล์……..	47
3.6	รายละเอียดยูสเคสแสดงกระแสงานในระบบ……...	47
3.7	รายละเอียดยูสเคสกระทำการเปลี่ยนกระแสงานเป็นบีเพล…….	48
3.8	รายละเอียดยูสเคสแปลงกระแสงานยอวล์เป็นบีเพล……	48
3.9	รายละเอียดยูสเคสสร้างบีเพล……..	49
3.10	รายละเอียดยูสเคสปรับโครงสร้างกระแสงานยอวล์…….	49
3.11	รายละเอียดยูสเคสสร้างไฟล์ยอวล์ใหม่.....	49
5.1	แบบรูปในระบบเช่ายืมคืนหนังสือ…….	59
5.2	กรณีทดสอบกระแสงานยอวล์ระบบเช่ายืมหนังสือ……..	72
5.3	แสดงแบบรูปในรูประบบกู้ชำระเงิน……..	74
5.4	กรณีทดสอบกระแสงานยอวล์ระบบกู้ชำระเงิน ……	77
5.5	แสดงแบบรูปในรูประบบลงทะเบียนเรียน……	79
5.6	กรณีทดสอบกระแสงานยอวล์ระบบลงทะเบียนเรียน……..	81
5.7	แสดงแบบรูปในรูประบบลงทะเบียนเรียน……..	83
5.8	กรณีทดสอบกระแสงานยอวล์ระบบการสอบ……	83
5.9	แสดงแบบรูปในรูประบบรับพนักงานใหม่…….	86
5.10	กรณีทดสอบกระแสงานยอวล์ระบบรับพนักงานใหม่.....	86
5.11	แสดงแบบรูปในรูประบบสั่งซื้อหนังสือ…….....	88
5.12	กรณีทดสอบกระแสงานยอวล์ระบบสั่งซื้อหนังสือ……	90
ก-1	แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล…….	94
ค-1	กรณีทดสอบกระแสงานยอวล์แบบรูป Multi Merge ……...	116

ตารางที่		หน้า
ค-2	กรณีทดสอบกระแสงานยอวล์แบบรูป Arbitrary Cycle ……...	121
ค-3	กรณีทดสอบกระแสงานยอวล์แบบรูป MI without a Priori Run-Time Knowledge……..	122
ค-4	กรณีทดสอบกระแสงานยอวล์แบบรูป Interleaved Parallel ……..	124
ค-5	กรณีทดสอบกระแสงานยอวล์แบบรูป Milestone……	126
ค-6	กรณีทดสอบกระแสงานยอวล์แบบรูป MI without Synchronization……..	129
ค-7	กรณีทดสอบกระแสงานยอวล์แบบรูป Deferred Choice……	131

สารบัญภาพ

ภาพที่		หน้า
2.1	องค์ประกอบของภาษาบีเพล.....	6
2.2	แสดงแบบรูปกระแสงาน ในยอวล์.....	9
2.3	ตัวอย่างกระแสงานยอวล์.....	12
2.4	ตัวอย่างการทำงานของภาษายอวล์.....	12
2.5	สภาพแวดล้อมของภาษายอวล์.....	13
2.6	แบบรูป Sequence.....	14
2.7	แบบรูป Parallel split.....	14
2.8	แบบรูป Synchronization.....	15
2.9	แบบรูป Exclusive Choice.....	15
2.10	แบบรูป Simple Merge.....	16
2.11	แบบรูป Multi-Choice.....	17
2.12	แบบรูป Synchronizing Merge.....	17
2.13	แบบรูป Multi-Merge.....	18
2.14	แบบรูป Structured Discriminator.....	18
2.15	ลักษณะแบบรูป Arbitrary cycle.....	19
2.16	แบบรูป Arbitrary cycle.....	19
2.17	แบบรูป Multiple Instances without Synchronization.....	20
2.18	แบบรูป Multiple Instances with a priori Design-Time Knowledge	21
2.19	แบบรูป Multiple Instances with a priori Run-Time Knowledge.....	21
2.20	แบบรูป Multiple instances without a priori run-time knowledge	22
2.21	แบบรูป Deferred Choice.....	23
2.22	แบบรูป Interleaved Parallel Routing.....	23
2.23	แบบรูป milestone.....	24
2.24	แบบรูป Cancel Activity.....	24
2.25	แบบรูป Cancel Case.....	25
2.26	การเปลี่ยนรูปโครงสร้างโปรแกรม.....	26
2.27	อัลกอริทึม Backward Copy	26

ภาพที่		หน้า
2.28	อัลกอริทึม Forward Copy	27
2.29	แสดงการขั้นตอนการปรับโครงสร้างของชุดคำสั่งภายในแฮมมอคกราฟ.....	28
3.1	ภาพร่วมการทำงาน.....	33
3.2	แบบรูป Multi Merge	34
3.3	โครงร่างบีเพลในแบบรูป Multi Merge	35
3.4	แบบรูป Multiple instances without a priori run-time knowledge	36
3.5	โครงร่างบีเพลในแบบรูป MI without a Priori Run-Time Knowledge	36
3.6	แบบรูป Interleaved Parallel Routing	37
3.7	โครงร่างบีเพลในแบบรูป Interleaved Parallel Routing	38
3.8	แบบรูป Milestone	38
3.9	โครงร่างบีเพลในแบบรูป Milestone	39
3.10	อัลกอริทึมในการปรับโครงสร้างแบบรูป Arbitrary Cycle	42
3.11	กระแสงานยอวล์แบบรูป Arbitrary Cycle ก่อนปรับโครงสร้าง.....	43
3.12	กระแสงานยอวล์แบบรูป Arbitrary Cycle หลังปรับโครงสร้าง.....	43
3.13	อัลกอริทึม Fold function	44
3.14	แผนภาพกิจกรรมการทำงานของระบบ.....	44
3.15	แผนภาพยูสเคสของเครื่องมือ.....	46
3.16	แผนภาพคลาสของเครื่องมือ.....	50
3.17	คลาส YawlUI	50
3.18	คลาส AboutFrame	51
3.19	คลาส HelpFrame	51
3.20	คลาส Run	51
3.21	คลาส ParserYawl	52
3.22	คลาส Queue	52
3.23	คลาส VertexYawl	53
3.24	คลาส CopyFile	53
3.25	คลาส FoldFunction	54
3.26	คลาส ReportBpel	55
3.27	คลาส Hammockgraph	55

ภาพที่		หน้า
3.28	คลาส CreateYAWL ……..……	55
3.29	คลาส JGraphAdapter ……..……	56
3.30	แผนภาพลำดับ Convert YAWL workflow(2)……..……	56
3.31	แผนภาพลำดับ Run fold function(2.1)……..……..	58
3.32	แผนภาพลำดับ Input yawl workflow(1)……	59
3.33	แผนภาพลำดับ Parser Yawl workflow(1.2)…… ….....	60
3.34	แผนภาพลำดับ Show workflow(1.3)…… ……..…	60
3.35	แผนภาพลำดับ Create BPEL skeleton(2.2)…... ……..……..	61
3.36	แผนภาพลำดับ Change Yawl workflow (3)…… ……..……	62
3.37	แผนภาพลำดับ Create BPEL skeleton(3.1)…... ……..……..	62
4.1	แผนภาพรายการต้นไม้เครื่องมือ……..	64
4.2	หน้าจอหลักของเครื่องมือ.....	64
4.3	หน้าจอรายละเอียดผู้จัดทำ……	65
4.4	หน้าจอแสดงวิธีใช้เครื่องมือ.....	65
4.5	หน้าจอเพิ่มกระแสงานยอวล์ใหม่……..	66
4.6	หน้าจอแสดงกระแสงานยอวล์……..	66
4.7	หน้าจอแสดงบีเพลที่สร้างขึ้น.....	67
5.1	กระแสงานยอวล์ทดสอบระบบเช่ายืมคืนหนังสือ……	69
5.2	โครงร่างบีเพลระบบเช่ายืมหนังสือ……	71
5.3	กระแสงานยอวล์ระบบกู้ชำระเงิน……..	74
5.4	โครงร่างบีเพลระบบกู้ชำระเงิน……	75
5.5	กระแสงานยอวล์ระบบลงทะเบียนเรียน.....……..	79
5.6	โครงร่างบีเพลระบบลงทะเบียนเรียน……	80
5.7	กระแสงานยอวล์ระบบการสอบ.....……	82
5.8	โครงร่างบีเพลระบบการสอบ……..	84
5.9	กระแสงานยอวล์ระบบรับพนักงานใหม่……..……	85
5.10	กระแสงานยอวล์ระบบรับพนักงานใหม่หลังปรับโครงสร้าง……..	85
5.11	โครงร่างบีเพลระบบรับพนักงานใหม่……..……..	87
5.12	กระแสงานยอวล์ระบบสั่งซื้อหนังสือ……	88

ภาพที่		หน้า
5.13	โครงร่างบีเพลระบบสั่งซื้อหนังสือ.....	89
ข-1	ระบบเช่ายืมหนังสือภาพรวม……...	102
ข-2	ระบบเช่ายืมหนังสือ ส่วน Flow 1……..	103
ข-3	ระบบเช่ายืมหนังสือ ส่วน Component 7 ……..……..	104
ข-4	โครงร่างบีเพลระบบกู้ชำระเงิน …….	105
ข-5	ระบบเช่ายืมหนังสือ ส่วน Component 9.....……	106
ข-6	ระบบเช่ายืมหนังสือ ส่วน Component 10 ……..……...	107
ข-7	โครงร่างบีเพลระบบลงทะเบียนเรียน……	108
ข-8	ระบบลงทะเบียนเรียน ส่วน Flow1……..….	109
ข-9	โครงร่างบีเพลระบบการสอบ…….....	110
ข-10	ระบบการสอบส่วน Component 6……..	111
ข-11	โครงร่างบีเพลระบบรับพนักงานใหม่…….	112
ข-12	ระบบรับพนักงานใหม่ ส่วน Component 6……..	113
ข-13	โครงร่างบีเพลระบบสั่งซื้อหนังสือ……..…	114
ข-14	ระบบรับสั่งซื้อหนังสือ ส่วนComponent 7…...……	115
ค-1	กระแสงานยอวล์แบบรูป Multi Merge …….	116
ค-2	บีเพลของแบบรูป Milestone……	118
ค-3	กระแสงานยอวล์แบบรูป Arbitrary Cycle ก่อนปรับโครงสร้าง……	119
ค-4	กระแสงานยอวล์แบบรูป Arbitrary Cycle หลังปรับโครงสร้าง ……..	119
ค-5	บีเพลของแบบรูป Arbitrary Cycle ……..……...	120
ค-6	กระแสงานยอวล์แบบรูป MI without a Priori Run-Time Knowledge ……..	121
ค-7	บีเพลของแบบรูป MI without a Priori Run-Time Knowledge……..	123
ค-8	กระแสงานยอวล์แบบรูป Interleaved Parallel ……..….	124
ค-9	บีเพลของแบบรูป Interleaved Parallel ……..……	125
ค-10	กระแสงานยอวล์แบบรูป Milestone ……	126
ค-11	บีเพลของแบบรูป Milestone ……..……..	128
ค-12	กระแสงานยอวล์แบบรูป MI without Synchronization ……..……...	129
ค-13	บีเพลของแบบรูป MI without Synchronization ……..…..	130
ค-14	กระแสงานยอวล์แบบรูป Deferred Choice ……..	131

ภาพที่ หน้า

ค-15 บีเพลของแบบรูป Deferred Choice …….. 132
ค-16 กระแสงานยอวล์แบบรูป Cancel Activity ……..……... 133
ค-17 บีเพลของแบบรูป Cancel Activity…….. 135

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันมีความนิยมนำเว็บเซอร์วิส (Web service) มาใช้ในองค์กรธุรกิจอย่างแพร่หลายทำให้เกิดการทำธุรกรรมระหว่างคู่ค้าทางธุรกิจ โดยในองค์กรต่างๆ มีแนวคิดที่จะนำเอาเซอร์วิส (Service) มาประกอบกันเพื่อให้เกิดเป็นกระบวนการทางธุรกิจใหม่ ส่วนใหญ่นิยมใช้ภาษาบีเพล (Bpel) ในการประกอบการทำงานจากหลายๆเซอร์วิส เข้าด้วยกันทำให้เกิดเซอร์วิสและกระบวนการทางธุรกิจใหม่ เพื่อรองรับกับความเปลี่ยนแปลงทางธุรกิจในปัจจุบันได้อย่างเหมาะสม [1]

อย่างไรก็ตามกระบวนการทางธุรกิจในปัจจุบันมีความซับซ้อนมากขึ้น การใช้ภาษาบีเพลในการประกอบการทำงานของกระบวนการทางธุรกิจที่มีความซับซ้อนนั้น ไม่สามารถอธิบายความหมายของกระบวนการทางธุรกิจนั้นได้อย่างเหมาะสม แต่มีภาษาที่ถูกออกแบบให้สามารถอธิบายกระบวนการทางธุรกิจที่มีความซับซ้อนได้ มีอยู่หลายภาษา เช่นภาษายอวล์ (YAWL) [2] โดยภาษายอวล์ได้ถูกออกแบบมาสำหรับ กำหนด (Define) วิเคราะห์ (Analysis) กระทำกระบวนการทางธุรกิจ และ รองรับกระบวนการทางธุรกิจที่ซับซ้อนได้ทำให้การนำเสนอกระบวนการทางธุรกิจโดยภาษายอวล์มีความสะดวกขึ้นกว่าใช้ภาษาบีเพล แต่ภาษายอวล์ไม่สนับสนุนกระบวนการประกอบกันเป็นเซอร์วิสใหม่ได้ และยังไม่ได้รับความนิยมอย่างแพร่หลาย

จากการศึกษางานวิจัย [3] ได้นำเสนอกระบวนการเปลี่ยนรูปกระแสงานจากแผนภาพบีพีเอ็มเอ็น (BPMN) เป็นภาษาบีเพล เพื่อสร้างเป็นเซอร์วิส แต่การนำเสนอกระบวนการทางธุรกิจด้วยแผนภาพบีพีเอ็มเอ็นมีข้อจำกัดเรื่องการอธิบายความหมาย เพื่อใช้ในการวิเคราะห์และตรวจสอบกระบวนการทางธุรกิจ จากงานวิจัย [4] ได้นำเสนอกระบวนการแปลงกระแสงานจากแผนภาพบีพีเอ็มเอ็นเป็นภาษายอวล์ การแปลงแผนภาพบีพีเอ็มเอ็น เป็นภาษายอวล์เพื่อช่วยในการวิเคราะห์กระบวนการทางธุรกิจ เพราะภาษายอวล์ถูกออกแบบมาเพื่อรองรับการนำเสนอกระบวนการธุรกิจที่มีความซับซ้อน และสามารถตรวจสอบได้ แต่ภาษายอวล์ไม่สามารถนำมาใช้ประกอบการทำงานเป็นเซอร์วิสใหม่ได้ โดยจากการศึกษาพบว่ายังไม่มีเครื่องมือในการเปลี่ยนรูปจากภาษายอวล์เป็นภาษาบีเพล เพื่อใช้ในการนำเสนอกระบวนการทางธุรกิจที่มีความซับซ้อนและสร้างกระบวนการทางธุรกิจนั้นเป็นเซอร์วิสใหม่ ในการเปลี่ยนรูปจากภาษายอวล์เป็นภาษาบีเพล ผู้วิจัยพบปัญหา คือ บางแบบรูปที่ภาษายอวล์รองรับ แต่แบบรูปนั้นภาษาบีเพลไม่รองรับ เช่น แบบรูป Multi Merge แบบรูป Interleaved Parallel Routing แบบรูป Milestone แบบรูป Arbitrary Cycle เป็นต้น

วิทยานิพนธ์นี้จึงนำเสนอกระบวนการเปลี่ยนแบบรูปในภาษายอวล์เป็นภาษาบีเพลโดยรองรับการเปลี่ยนแบบรูปที่ภาษาบีเพลไม่รองรับได้ โดยยังคงลักษณะการทำงานของแบบรูปในภาษายอวล์อยู่ และสามารถปรับแบบรูปโครงสร้างที่ไม่ดี โดยใช้แนวคิดอัลกอริทึมแฮมมอดกราฟ (Hammock graph) [5] แล้วจึงทำการเปลี่ยนรูปกระแสงานที่มีโครงสร้างที่ดีเป็นภาษาบีเพล และวิทยานิพนธ์นี้พัฒนาเครื่องมือการเปลี่ยนรูปจากภาษายอวล์เป็นภาษาบีเพลทำให้สามารถสร้างบริการจากเว็บเซอร์วิสได้ เพื่อลดช่องว่างของการนำเสนอกระบวนการทางธุรกิจและการประกอบกันของเว็บเซอร์วิสทำให้ง่ายขึ้น

1.2 วัตถุประสงค์

1) นำเสนอกระบวนการแปลงกระแสงานยอวล์ให้อยู่ในรูปโครงร่างบีเพล

2) พัฒนาเครื่องมือแปลงกระแสงานยอวล์ให้อยู่ในรูปโครงร่างบีเพล

1.3 ขอบเขตการดำเนินงาน

1) กำหนดข้อมูลนำเข้าจากกระแสงานยอวล์ที่อยู่ในรูปของภาษาเอ็กซ์เอ็มแอล (XML)

2) การตรวจสอบส่วนประกอบโครงสร้างที่ดี (Well structured component) ของกระแสงานยอวล์โดยประยุกต์ใช้แนวคิดของ Control flow graph หรือ Fold function [6]

3) ส่วนประกอบโครงสร้างที่ดีของกระแสงานยอวล์ที่นำมาใช้ในกระบวนการแปลงกระแสงานยอวล์ให้เป็นภาษาบีเพล ซึ่งกำหนดจาก [7] มีแบบรูปดังนี้

- แบบรูป Sequence
- แบบรูป Parallel Split
- แบบรูป Synchronization
- แบบรูป Exclusive Choice
- แบบรูป Simple Merge
- แบบรูป Multi Choice
- แบบรูป Synchroizing Merge
- แบบรูป Discriminator
- แบบรูป MI without Synchronization
- แบบรูป MI with a Priori Design-Time knowledge
- แบบรูป MI with a Prioir Run-Time knowledge
- แบบรูป Deferred Choice
- แบบรูป Cancellation Patterns

- แบบรูป Cancel Case

4) ทำการปรับปรุงส่วนประกอบโครงสร้างที่ไม่ดี (Non well structured component) ของกระแสงานยอวล์ต่อไปนี้ ให้เป็นส่วนประกอบโครงสร้างที่ดี

- แบบรูป Multi Merge
- แบบรูป MI without a Priori Run-Time Knowledge
- แบบรูป Interleaved Parallel Routing
- แบบรูป Milestone
- แบบรูป Arbitrary Cycle
- แบบรูป Implicit Termination

5) พัฒนาเครื่องมือในการเปลี่ยนรูปกระแสงานยอวล์ให้ใช้งานภายใต้ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ (Microsoft Windows)

6) ทดสอบความถูกต้องของผลลัพธ์ภาษาบีเพลที่ได้จากการเปลี่ยนรูปโดยใช้โปรแกรมตัวประมวลผลยอวล์ (YAWL engine) ในการประมวลผลกระแสงานยอวล์เปรียบเทียบกับใช้โปรแกรมตัวประมวลผลบีเพล (BPEL engine) ประมวลผลไฟล์บีเพลที่ได้ว่าให้ผลลัพธ์ตรงกันด้วยเปรียบเทียบอย่างน้อย 3 กระแสงานยอวล์

1.4 ขั้นตอนการวิจัย

1) ศึกษาทฤษฎีพื้นฐานของขั้นตอนกระแสงานยอวล์ภาษาบีเพล
2) ศึกษาวิธีการในการเปลี่ยนรูปจากกระแสงานไร้โครงสร้างเป็นกระแสงานมีโครงสร้าง
3) ออกแบบขั้นตอนวิธีเปลี่ยนรูปจากกระแสงานไร้โครงสร้างเป็นกระแสงานมีโครงสร้าง
4) ทดสอบวิธีการที่นำเสนอ
5) วิเคราะห์ผลการทดลอง
6) สรุปผลและเรียบเรียงวิทยานิพนธ์

1.5 ลำดับขั้นตอนการเสนอผลการวิจัย

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บทดังต่อไปนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึงความเป็นมาและความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย ขอบเขตของการวิจัย ประโยชน์ที่คาดว่าจะได้รับ วิธีการดำเนินการวิจัย ลำดับขั้นตอนการนำเสนอผลการวิจัย บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ บทที่ 3 กล่าวถึงแนวคิดวิธีการระบบเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพล ซึ่งกล่าวถึงการสร้างและออกแบบเครื่องมือ บทที่ 4 กล่าวถึงสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ และโครงสร้างเครื่องมือ บทที่ 5 กล่าวถึง

สภาพแวดล้อมที่ใช้ในการทดลองเครื่องมือ ขั้นตอนการทดสอบเครื่องมือ การสร้างกระแสงานยอวล์ที่ใช้ในการทดสอบ ผลการทดสอบ และบทที่ 6 เป็นข้อสรุปและข้อเสนอแนะจากการวิจัย

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ ได้รับการตีพิมพ์เป็นบทความทางวิชาการต่อไปนี้

1) บทความวิชาการเรื่อง "การแปลงกระแสงานยอวล์เป็นโครงร่างบีเพล" โดย สิทธิพงศ์ พรอุดมทรัพย์ และ รศ.ดร.วิวัฒน์ วัฒนาวุฒิ ในงานประชุมวิชาการ 14th National Computer Science and Engineering Conference (NCSEC 2010) ระหว่างวันที่ 17-19 พฤศจิกายน 2553 ณ จังหวัดเชียงใหม่ ประเทศไทย

2) บทความวิชาการเรื่อง “Transforming Yawl Workflow to BPEL Skeleton” โดย สิทธิพงศ์ พรอุดมทรัพย์ และ รศ.ดร.วิวัฒน์ วัฒนาวุฒิ ในงานประชุมวิชาการ The 2nd International Conference on Software Engineering and Service Science (ICSESS2011) ระหว่างวันที่ 14-18 กรกฎาคม 2554 ณ กรุงปักกิ่ง สาธารณรัฐประชาชนจีน

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องกับวิทยานิพนธ์นี้ประกอบด้วย ภาษาบีเพล กระแสงานยอวล์ แบบรูปกระแสงาน แนวคิดของอัลกอริทึมแฮมมอค ลักษณะแบบรูปในกระแสงานยอวล์ที่อยู่ในโครงสร้างที่ดีเป็นโครงร่างบีเพล และงานวิจัยที่เกี่ยวข้องโดยทั้ง 6 มีรายละเอียดดังนี้

2.1 ภาษาบีเพล

ภาษาบีเพล หรือเรียกว่า Web Services Business Process Execution Language [1, 8] เป็นภาษาเอ็กซ์เอ็มแอลที่ใช้ในการพัฒนากระบวนการทางธุรกิจจากเว็บเซอร์วิส โดยใช้ภาษาดับเบิลยูเอสดีแอล (WSDL) ในการอธิบายข้อมูลของเว็บเซอร์วิส ทั้งนี้ก่อนที่จะมีภาษาบีเพล แต่ละบริษัทผู้ผลิตกระบวนการทางธุรกิจต่างมีรูปแบบการเขียนกระบวนการทางธุรกิจที่แตกต่างกัน จึงมีการกำหนดภาษาบีเพลขึ้นเพื่อกำหนดเป็นภาษามาตรฐานกลางสำหรับการเขียนกระบวนการทางธุรกิจโดยใช้แพลตฟอร์มรูปแบบเว็บเซอร์วิส โดยภาษาบีเพลถูกพัฒนามาจากสองภาษาคือ ภาษา WSFL (Web Services Flow Language) ออกแบบโดยบริษัท IBM และภาษา XLANG ออกแบบโดยบริษัท Microsoft กำหนดเป็นเวอร์ชันแรกเมื่อเดือนสิงหาคมปีค.ศ. 2002 และได้ถูกปรับปรุงเป็นเวอร์ชัน 1.1 ในเดือนมีนาคมปี ค.ศ. 2003 ต่อทางองค์กร OASIS (The Organization for the Advancement of Structured Information Standards) ได้ประกาศให้ BPEL4WS 1.1 เป็นมาตรฐานสำหรับการพัฒนากระบวนการทางธุรกิจบนเว็บเซอร์วิสปัจจุบันเวอร์ชันล่าสุดของภาษาบีเพล คือ WS-BPEL2.0

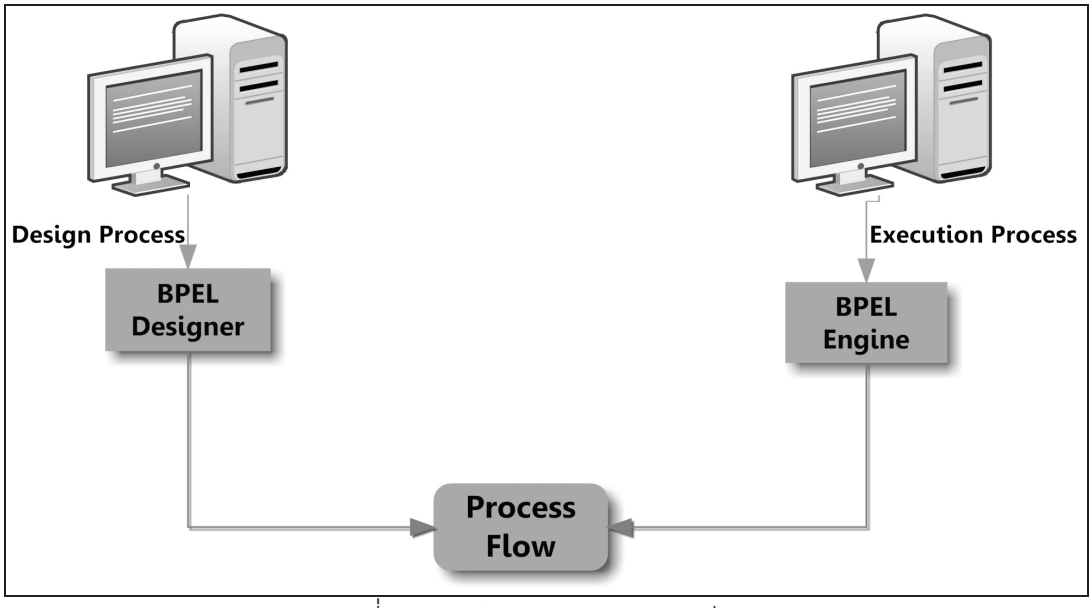
2.1.1 การใช้งานของภาษาบีเพล

การใช้งานของภาษาบีเพล ดังภาพที่ 2.1 ประกอบด้วย 3 ส่วนดังนี้

- ตัวออกแบบบีเพล (BPEL Designer) เป็นเครื่องมือในการพัฒนากระบวนการทางธุรกิจสามารถใช้จำลองกระบวนการทางธุรกิจโดยแสดงเป็นสัญลักษณ์กราฟิกผลลัพธ์ที่ได้อยู่ในรูปแบบไฟล์ Process Flow Template โดยทั่วไปเครื่องมือส่วนใหญ่ออกแบบตามมาตรฐานบีพีเอ็มเอ็น (BPMN) ในการเขียนสัญลักษณ์

- ตัวประมวลผลบีเพล (BPEL Engine) เป็นตัวประมวลผลแฟ้มแม่แบบการไหลของกระบวนการ (Process Flow Template) ตามมาตรฐานภาษาบีเพลโดยจะทำงานลักษณะต่างๆ เช่น การเรียกใช้เว็บเซอร์วิส การกำหนดเนื้อหาของข้อมูล การจัดการข้อผิดพลาด หรือ การควบคุมลำดับการทำงาน โดยทั่วไปตัวจักรภาษาบีเพล จะทำงานร่วมกับแอพพลิเคชัน เซิร์ฟเวอร์ (Application Server)

- แม่แบบการไหลของกระบวนการ (Process Flow Template) เป็นแฟ้มข้อมูลที่ระบุกระบวนการทางธุรกิจตามข้อกำหนดของภาษาบีเพลโดยจะเป็นไฟล์ที่ถูกสร้างมาจากตัวออกแบบบีเพล และใช้ตัวประมวลผลบีเพลในการประมวลผล



ภาพที่ 2.1 องค์ประกอบของภาษาบีเพล [1]

2.1.2 **คำสั่งของภาษาบีเพล**

ภาษาบีเพลได้รับการออกแบบให้มีโครงสร้างแบบภาษาเอ็กซ์เอ็มแอล และมีคำสั่งในการเรียกใช้บริการเว็บเซอร์วิส โดยชุดคำสั่งที่ประกอบกันนี้จะจำลองการทำงานของกระบวนการธุรกิจได้ สามารถเรียกใช้เหมือนเป็นเว็บเซอร์วิสโดยเรียกผ่านดับเบิลยูเอสดีแอล โดยภาษาบีเพลมีชุดของคำสั่งที่ระบุแอ็คติวิตีพื้นฐาน (Primitive activities) ที่ใช้กำหนดโครงสร้างพื้นฐานและใช้ในงานทั่วไปดังนี้

- แอ็คติวิตี<invoke> ใช้เพื่อกำหนดกระบวนการทางธุรกิจในการเรียกใช้โอเปอร์เรชัน (operation) ภายในแทค porttype ที่นิยามอยู่ในภาษาดับเบิลยูเอสดีแอลของเว็บเซอร์วิส

- แอ็คติวิตี <receive> ใช้เพื่อกำหนดกระบวนการทางธุรกิจหยุดรอข่าวสารที่จะมาถึง

- แอ็คติวิตี <reply> ใช้เพื่อกำหนดกระบวนการทางธุรกิจส่งข่าวสารเพื่อตอบกลับข่าวสารที่ได้รับมา

- แอ็คติวิตี <assign> ใช้เพื่อกำหนดคัดลอกข้อมูลจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง

- แอ็คติวิตี <throw> ใช้เพื่อระบุข้อผิดพลาดที่เกิดขึ้น

- แอ็คติวิตี <wait> ใช้เพื่อกำหนดกระบวนการทางธุรกิจหยุดรอตามระยะเวลาหนึ่ง

- แอ็คติวิตี <terminate> ใช้เพื่อยกเลิกกระบวนการทางธุรกิจทั้งหมด

นอกจากนี้ภาษาบีเพล ยังมีแอ็คติวิตีที่มีลักษณะกำหนดโครงสร้าง (Structured task)

ที่ใช้การรวมแอ็คติวิตีพื้นฐานเข้าด้วยกันเพื่อใช้ควบคุมลำดับการทำงานและสร้างกระบวนการทางธุรกิจที่ซับซ้อนขึ้น โดยมีคำสั่งต่างๆ ดังนี้

- แอ็คติวิตี <sequence> ใช้เพื่อกำหนดให้กลุ่มกิจกรรมบอกถึงลำดับในการทำงาน

- แอ็คติวิตี <flow> ใช้เพื่อระบุชุดคำสั่งทำงานแบบขนาน

- แอ็คติวิตี <switch> ใช้เพื่อกำหนดให้ชุดภารกิจทำงานแบบเลือกทำ (Case-switch) ตามเงื่อนไขตรรกะที่ระบุ

- แอ็คติวิตี <while> ใช้เพื่อกำหนดให้มีการทำงานซ้ำจนกว่าจะเป็นไปตามเงื่อนไขที่ระบุ

- แอ็คติวิตี <pick> ใช้เพื่อหยุดกระแสงานรอจนกว่าจะมีข่าวสารที่เหมาะสมมาถึง หรือหมดเวลาที่รอ ถ้าคำสั่งประเภทนี้เกิดการทำงานกิจกรรมที่เกี่ยวพันกันจะถูกกระทำและจะสิ้นสุดการเลือก (Pick) นอกจากนี้ภาษาบีเพลยังมีคำสั่งในการนิยามข้อมูลดังนี้

- แอ็คติวิตี <partnerLink> ใช้เพื่อกำหนด porttype ของเว็บเซอร์วิส (Partner) ที่จะเข้ามาร่วมในกระบวนการทางธุรกิจ

- แอ็คติวิตี <variable> ใช้เพื่อกำหนดค่าตัวแปรในกระบวนการทางธุรกิจ

การพัฒนาภาษาบีเพล และการทำงานของภาษาบีเพล จะต้องมีตัวออกแบบภาษาบีเพล และตัวประมวลผลภาษาบีเพล ซึ่งปัจจุบันเราสามารถที่จะใช้โปรแกรมเน็ตบีนส์ไอดีอี (NetBeans) 6.7.1 ที่ติดตั้งแบบ Enterprise Pack

2.1.3 **เซิร์ฟเวอร์บีเพล** (BPEL Server)

เซิร์ฟเวอร์บีเพลทำหน้าที่ ประมวลผลกระบวนการทางธุรกิจตามที่กำหนดในภาษาบีเพล ทั้งนี้เซิร์ฟเวอร์บีเพล ทำงานอยู่ภายใต้ Java EE หรือ .NET Application Server และเซิร์ฟเวอร์บีเพลทำหน้าที่ ด้านความปลอดภัย การเปลี่ยนรูป (transactions) การปรับขนาด (scalability) การเชื่อมต่อกับดาตาเบส EJB

วิทยานิพนธ์นี้นำเสนอการเปลี่ยนรูปกระแสงานยอวล์เป็นโครงร่างภาษาบีเพล โดยโครงร่างบีเพล หมายถึง ชุดคำสั่งบีเพลที่ระบุการเรียกใช้บริการที่ยังไม่มีอยู่จริงทั้งนี้ โครงร่างจะแสดงลำดับการทำงานเท่านั้น ยังขาดรายละเอียดของตัวการทำงานเช่น แอ็คติวิตี <receive> แอ็คติวิตี <assign> แอ็คติวิตี<reply> เป็นต้น โครงร่างบีเพลผลลัพธ์ช่วยให้ได้กรอบของลำดับการทำงานของขั้นตอนจริงได้และช่วยผู้พัฒนาในการพัฒนาเว็บเซอร์วิส

2.2 **กระแสงานยอวล์** [2]

2.2.1 **ลักษณะพื้นฐานของกระแสงานยอวล์**

ในปัจจุบันกระบวนการทางธุรกิจมีความซับซ้อนมากขึ้นจึงมีความต้องการภาษาใหม่ที่สามารถอธิบายกระบวนการทางธุรกิจที่มีความซับซ้อนได้ โดยภาษาที่สามารถอธิบายกระบวนการ

ทางธุรกิจที่มีความซับซ้อนมีหลายภาษาหนึ่งในภาษานั้นคือ ภาษายอวล์ โดยภาษายอวล์เป็นระบบจัดการวิเคราะห์กระแสงาน (Workflow Management System) เพื่อช่วยในการวิเคราะห์การจัดการกระแสงาน ใช้ในการระบุความแตกต่างระหว่างภาษา ประกอบโดยเซตสมบูรณ์ (Compete set) ของแบบรูปกระแสงานที่ สามารถตรวจสอบการขัดแย้งของกระแสงานควบคุมสายงาน (Control flow)ได้ โดยภาษายอวล์มีพื้นฐานจากภาษา Petri net คือมีโครงสร้างเป็น แบบรูปกระแสงานบนพื้นฐานสถานะ (State-based workflow pattern) แต่ภาษายอวล์พัฒนาให้รองรับแนวคิดกระแสงานที่มีความซับซ้อนมากขึ้น เช่นแบบรูปที่มีโครงสร้าง XOR, AND, OR-splits, joins, multiple instance, cancellation region ดังนั้นภาษายอวล์จึงถูกออกแบบขึ้นมาเพื่อรองรับแบบรูปกระแสงานที่มีความซับซ้อนต่างๆ และใช้กำหนด ความหมาย ของภาษาและใช้ตรวจสอบความถูกต้องของกระแสงานได้

กระแสงานยอวล์สามารถแบ่งออกเป็น 3 มุมมองคือ 1.) มุมมองการควบคุมสายงาน (Control-flow perspective) เป็นมุมมองเกี่ยวกับลำดับการทำงาน กับโครงสร้างที่แตกต่างกัน เช่น แบบรูป sequence แบบรูป choice parallelism และ แบบรูป join synchronization โดยแต่ละเซอร์วิส ประกอบกันเป็นลำดับของการทำงาน 2.) มุมมองข้อมูล(data perspective) เป็นมุมมองในระดับที่สูงกว่ามุมมองการควบคุมสายงาน เกี่ยวกับด้านข้อมูลในสายงาน ประกอบด้วย อ็อบเจกต์ระหว่างกระแสงาน ตัวแปรในกระแสงาน และคุณสมบัติเงื่อนไข(Condition) 3.) มุมมองทรัพยากร (resource perspective) เกี่ยวกับด้านโครงสร้างในองค์กร เป็นกระแสงานในรูปแบบของมนุษย์ และการตอบสนองของการทำงานของเซอร์วิส ด้วยในงานวิจัยนี้เราเน้นการทำงานด้านมุมมองการควบคุมสายงานของภาษายอวล์

ภาษายอวล์เป็นเครื่องมือจัดการกระแสงานที่พัฒนาขึ้นจากภาษา Petri nets ให้มีความง่ายในการใช้งานกว่าภาษา Petri net เดิมและ ถูกออกแบบให้รองรับแบบรูปที่มากกว่า จึงเป็นจุดเริ่มต้นในการพัฒนาภาษายอวล์ เริ่มพัฒนาขึ้นในปี ค.ศ. 2002 โดยเป็นความร่วมมือกันระหว่างมหาวิทยาลัย Eindhoven University of Technology และมหาวิทยาลัย Queensland University of Technology โดยพัฒนาจากภาษา Petri nets ให้รองรับโครงสร้างที่ซับซ้อนขึ้นเช่น แบบรูป multiple instances แบบรูป advanced synchronization และ แบบรูป cancellation ภาษายอวล์ถูกออกแบบให้ง่ายในการนำเสนอกระแสงานและรองรับแบบรูปที่ซับซ้อนกว่าภาษา Petri nets ด้วยให้รองรับ 19 แบบรูปกระแสงาน ใน 6 ประเภทดังภาพที่ 2.2

1) แบบรูปกระแสงานควบคุมพื้นฐาน (Basic Control-Flow Patterns) เป็นโครงสร้างพื้นฐานที่มีการทำงานพื้นฐาน เช่น แบบรูป sequential แบบรูป parallel

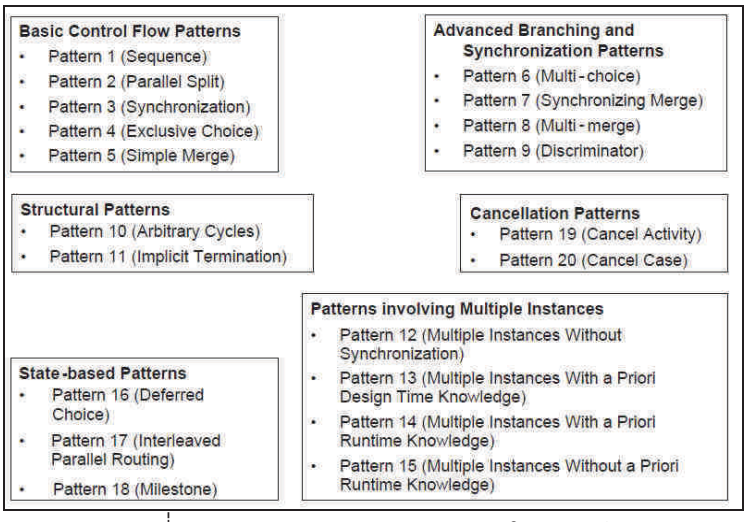
2) แบบรูปการแยกและผสานกระแสงานขั้นสูง (Advanced Branching and Synchronization Patterns) เป็นแบบรูปที่มีความซับซ้อนกว่าแบบรูปพื้นฐานโดยใช้โครงสร้าง spilt และ join ระดับสูง เช่น แบบรูป Synchronizing merge

3) แบบรูปกลุ่มโครงสร้าง (Structural Patterns) เป็นแบบรูปที่อนุญาตให้กำหนดโครงสร้างที่ไม่แน่นอนได้ เช่น แบบรูป Arbitrary Cycle

4) แบบรูปหลายอ็อบเจกต์ (Multiple Instance Patterns) เป็นแบบรูปที่เกิดการทำงานได้หลายอ็อบเจกต์ เช่น กระบวนการที่ต้องการ การทำงานหลายครั้ง

5) แบบรูปสถานะ (State-based Patterns) เป็นแบบรูปที่สนใจสถานะ การทำงานในกระแสงาน เพื่อกำหนดการทำงาน เช่น แบบรูป Milestone

6) แบบรูปการยกเลิก (Cancellation Patterns) เป็นเหตุการณ์ที่เกิดขึ้นในการยกเลิกกิจกรรมที่เกิดขึ้นแล้ว เช่น แบบรูป Cancel Activity



ภาพที่ 2.2 แสดงแบบรูปกระแสงาน ในยอวล์ [2]

ข้อกำหนดกระแสงาน ในภาษายอวล์คือเซตของ extended workflow nets (EWF-nets) โดยเรียกกระแสงานพื้นฐานที่มีงานเชิงเดียว(atomic tasks) และ งานประกอบ (composite tasks) คือ กระแสงานระดับสูง

โดยแต่ละกระแสงานประกอบด้วย งาน(tasks) และ เงื่อนไข(condition) มี place ขั้นกลางอยู่ แต่ละกระแสงานประกอบด้วย 1 ทางเข้า และ 1 ทางออก โดยแตกต่างจากภาษา Petri nets ที่จะติดต่อกับ transition กับ งานประกอบและ งานเชิงเดียว โดยตรงไม่ต้องมี place หรือ condition ขั้นกลาง

แต่ละงานสามารถมีลักษณะเป็นหลายอ็อบเจกต์ได้ โดยระบุตามภาพสัญลักษณ์ที่ใช้ในภาษายอวล์สามารถระบุจำนวนต่ำสุดและสูงสุดของข้อกำหนดกระแสงานดังนั้นเราสามารถระบุแบบสัญลักษณ์ของ EWF-net ได้ดังนี้

นิยาม 1 **EWF-net**: extended workflow net (EWF-net) ให้ N ประกอบด้วยส่วนสำคัญ $(C, i, o, T, F, split, join, rem, nofi)$

- C คือเซตของเงื่อนไข
- $i \in C$ คือ เงื่อนไขนำเข้า
- $o \in C$ คือ เงื่อนไขส่งออก
- T คือเซตของงาน
- $F \subseteq (C \setminus \{o\} \times T) \cup (T \times C \setminus \{i\}) \cup (T \times T)$ คือความสัมพันธ์ของกระแส
- ทุกโหนดในกราฟ $(C \cup T, F)$ คือเส้นทางจาก i ถึง o
- $split : T \to \{AND, XOR, OR\}$ ระบุ split แต่ละงาน
- $join : T \to \{AND, XOR, OR\}$ ระบุ join แต่ละงาน
- $rem : T! \to P(T \cup C \setminus \{i, o\})$ ระบุการเพิ่มงานโดยการถอดส่วนที่ว่างของกระแสงานออก
- $nofi : T! \to N \times N^{\inf} \times N^{\inf} \times \{dynamic, static\}$ ระบุจำนวนแต่ละงาน(ค่าต่ำสุด, ค่าสูงสุด,จุดเริ่มต้น, และ การสร้างกรณีตัวอย่าง แบบ พลวัต หรือ สถิต)

ต่อมาระบุข้อกำหนดกระแสงานคือการระบุส่วนประกอบของ EWF-nets โดยอยู่ในรูปของต้นไม้แบบลำดับชั้นดังนี้

นิยาม 2 workflow specification ให้ S ประกอบด้วยส่วนสำคัญ $(Q, top, T^{\diamond}, map)$

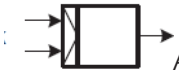
- Q คือ เซตของ EWF-nets
- $top \in Q$ คือกระแสงานบนสุด
- $T^{\diamond} = \cup_{N \in Q} T_N$ คือเซตของงานทั้งหมด
- $\forall_{N1, N2 \in Q} N1 \neq N2 \Rightarrow (C_{N1} \cup T_{N1}) \cap (C_{N1} \cup C_{N2}) = \phi$.คือไม่มีชื่อขัดแย้ง
- $map : T^{\diamond}! \to Q \setminus \{top\}$ คือ ฟังก์ชั่นแบบหนึ่งต่อหนึ่ง(Injective)หรือทั่วถึง (surjective) ที่ประกอบเป็นภารกิจเข้าไปใน EWF net
- ความสัมพันธ์ $\{(N1, N2) \in Q \times Q \mid \exists_{t \in dom(mapN1)} mapN1(t) = N2\}$ อยู่ในรูปของโครงสร้างต้นไม้

จะเห็นว่าภาษายอวล์สามารถนำไปปรับใช้ในงานเช่นเว็บเซอร์วิส และ แอพพลิเคชั่นขนาดใหญ่ที่ต้องการภาษาในการสื่อสาร

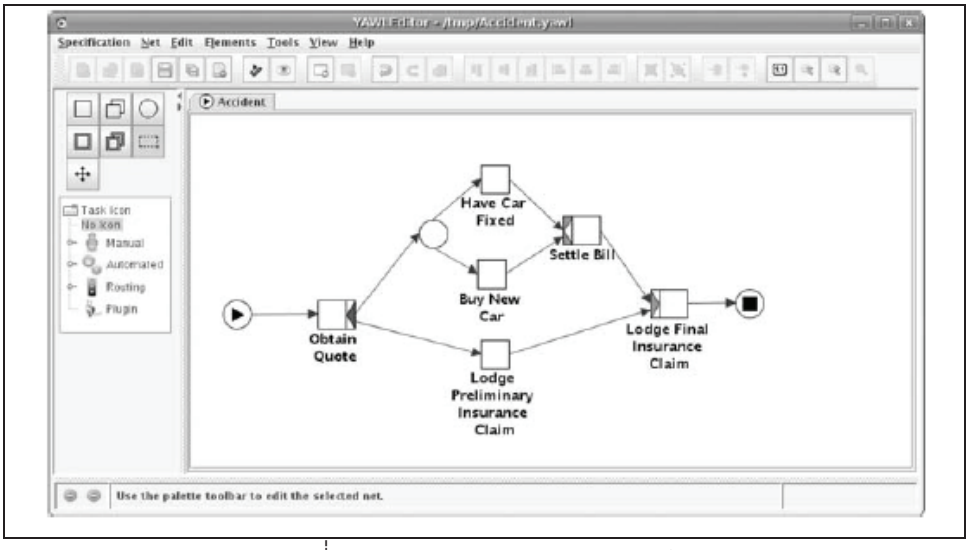
2.2.2 องค์ประกอบของภาษายอวล์

ในส่วนต่อไปแสดงสัญลักษณ์ที่มีในภาษายอวล์พร้อมอธิบายความหมายตามตารางที่ 2.1

ตารางที่ 2.1 รายละเอียดสัญลักษณ์ในภาษายอวล์

สัญลักษณ์	ความหมาย
Input Condition	ใช้แสดงจุดเริ่มต้นของกระแสงาน
Output Condition	ใช้แสดงจุดสิ้นสุดของกระแสงาน
Atomic Task	หมายถึงงานเดียวที่ใช้ดำเนินการโดยปกติผู้ใช้เป็นคน หรือโปรแกรมจากภายนอก
Composite Task	ใช้กำหนดกระแสงานย่อยของภาษายอวล์ ด้วยภายในใช้องค์ประกอบภาษายอวล์
Multiple Instance Atomic Task	อนุญาตให้เกิดการทำงานได้หลายอ็อบเจกต์ โดยเกิดขึ้นพร้อมกัน
Multiple Instance Composite Tasks	อนุญาตให้เกิดการทำงานได้หลายอ็อบเจกต์ ในกระแสงานย่อยโดยเกิดขึ้นพร้อมกัน
AND-Split	AND-Split ใช้ในการเริ่มต้นของงานขึ้นพร้อมกัน โดยที่เกิดขึ้นทุกกระแสงานขาออก
 AND-Join	AND-Join เกิดขึ้นเมื่อทุกกระแสงานเข้ามาในกิจกรรมที่มีโครงสร้าง AND-Join กระแสงานจึงเริ่มทำงาน
XOR-Split	XOR-Split เกิดกระแสงานออกเพียง 1 กระแสงานถูกใช้ในการเลือกกระทำการเส้นทางกระแสงาน โดยเกิดเพียง 1 กระแสงานเท่านั้น
XOR-Join	XOR-Join เกิดขึ้นเมื่อมี 1 กระแสงานเข้าสู่กิจกรรม ซึ่งมักใช้ในการร่วมกระแสงานที่เกิดขึ้นก่อนหน้า
OR-Split	OR-Split ใช้เมื่อเกิดกระแสงานออกไม่จำเป็นเกิดขึ้นทั้งหมด จำนวนการเกิดกระแสงานไม่ทราบจนถึงเวลาทำงานซึ่งเกิดการทำงานตามตรรกะที่เกิดขึ้น
OR-Join	OR-Join เกิดการทำงานขึ้นเมื่อมีกระแสงานทั้งหมดเกิดขึ้นหรือเกิดกระแสงานขึ้นอย่างใดอย่างนึ่ง
Condition	ใช้แสดงสถานะของ Net ในกระแสงาน

ส่วนต่อมาแสดงตัวอย่างการสร้างกระแสงานยอวล์ ตามภาพที่ 2.3 เป็นตัวอย่างกระแสงานเมื่อเกิดอุบัติเหตุขึ้นโดยสร้างจากโปรแกรม Yawl-Editor [2] เพื่อใช้กำหนดรายละเอียดของกระแสงานยอวล์และลำดับในการเรียกกิจกรรมที่เกิดขึ้น

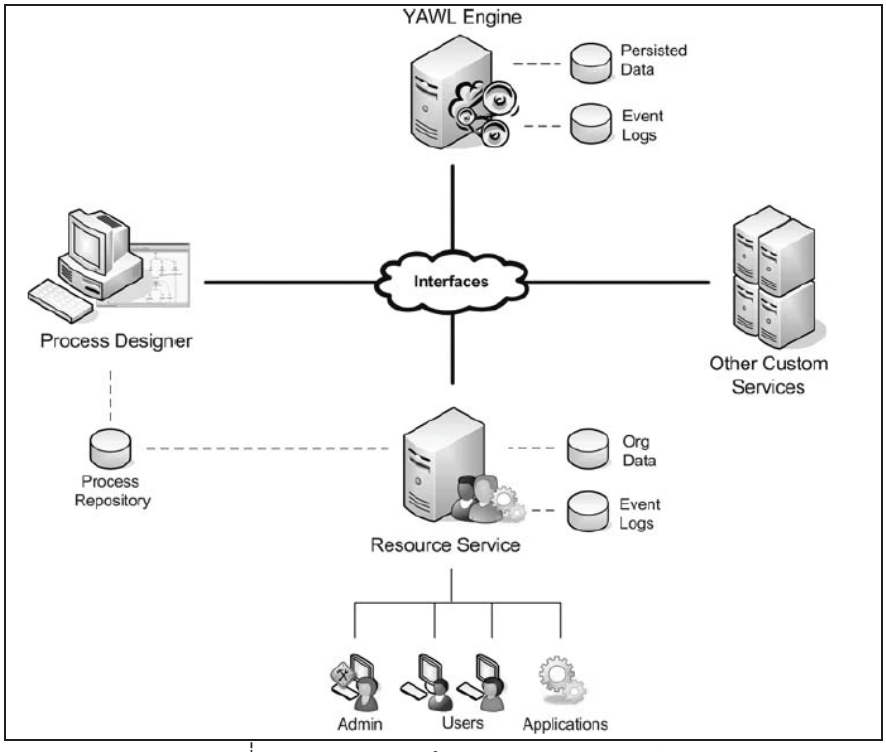


ภาพที่ 2.3 ตัวอย่างกระแสงานยอวล์ [2]

หลังจากกำหนดรายละเอียดของกระแสงานยอวล์ ต่อไปเป็นการแสดงตัวอย่างในการทำงานของตัวประมวลผลยอวล์ (Yawl Engine) ตามภาพที่ 2.4 เพื่อใช้งานกระแสงานยอวล์ที่ออกแบบ

ภาพที่ 2.4 ตัวอย่างการทำงานของภาษายอวล์ [2]

สภาพแวดล้อมในการทำงานของภาษายอวล์ตามภาพที่ 2.5 มีดังนี้ประกอบด้วย 1.) ส่วนออกแบบกระบวนการ (Process Designer) เป็นส่วนในการออกแบบกระแสงานยอวล์ 2.) ส่วนประมวลผลกระแสงานยอวล์เป็นส่วนในการทำงานกระแสงานยอวล์ที่ออกแบบขึ้น 3.) ส่วนบริการทรัพยากร (Resource Service) เป็นส่วนของการจัดการทรัพยากรบุคคลในสภาพแวดล้อมยอวล์ 4.) ส่วนบริการอื่นๆ (Other Custom Service) เป็นส่วนเสริมขยายการทำการทำงานของภาษายอวล์โดยผู้ใช้งานจากภายนอก



ภาพที่ 2.5 สภาพแวดล้อมของภาษายอวล์ [2]

2.3 แบบรูปของกระแสงานยอวล์ [9]

ในส่วนนี้นำเสนอลักษณะแบบรูปทั้ง 19 แบบรูปในกระแสงานยอวล์ที่รองรับ โดยแสดงรายละเอียดของแบบรูปต่างๆที่เกิดขึ้นโดยยกตัวอย่างสถานการณ์การดำเนินงานในกระบวนธุรกิจเพื่อแสดงถึงลักษณะต่างๆ เพื่อให้เกิดความเข้าใจชัดเจนไม่คลุมเครือ

2.3.1 แบบรูปกระแสงานควบคุมพื้นฐาน

แบบรูปกระแสงานควบคุมพื้นฐาน คือ กระบวนการกระแสงานมีการทำงานพื้นฐานโดยมีแบบรูปดังนี้

2.3.1.1 แบบรูป Sequence

- ลักษณะ

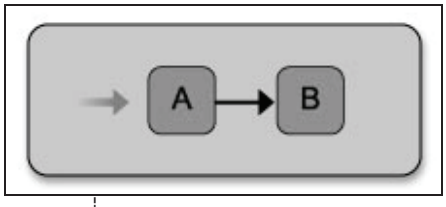
เป็นกิจกรรมในกระแสงาน ที่กิจกรรมเกิดการทำงานขึ้นหลังจากกิจกรรมก่อนหน้าทำงานเสร็จสิ้นและอยู่ในกระบวนการเดียวกัน

- ตัวอย่าง

- กิจกรรมการตรวจสอบบัญชีหลังจากกิจกรรมประมวลผลบัตรเครดิต

- ใบเสร็จรับเงินถูกพิมพ์หลังจากซื้อตั๋วรถไฟ

สามารถเขียนแบบรูป sequence ในภาษายอวล์ได้ดังภาพที่ 2.6



ภาพที่ 2.6 แบบรูป sequence [9]

2.3.1.2 แบบรูป Parallel Split

- ลักษณะ

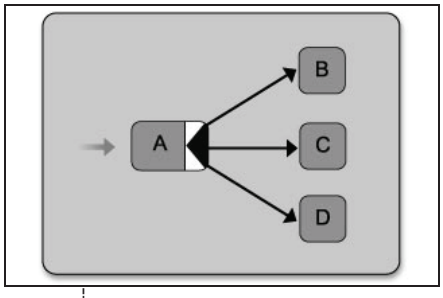
การแยกของกระแสงานควบคุม ตั้งแต่สองกระแสงานขึ้นไป ซึ่งแต่ละการทำงานของกระแสงานควบคุมทำงานขนานกัน

- ตัวอย่าง

-เมื่อลูกค้ามีการชำระเงินสินค้าจะเกิดกิจกรรมออกใบเสร็จรับเงินและกิจกรรมบรรจุของสำหรับส่งทำงานขนานกัน

-หลังจากกิจกรรมลงทะเบียนเสร็จสิ้น กิจกรรมแสดงรายละเอียดการลงทะเบียนจะเกิดขึ้น พร้อมกิจกรรมยืนยันการลงทะเบียน

สามารถเขียนแบบรูป parallel split ในภาษายอวล์ได้ดังภาพที่ 2.7



ภาพที่ 2.7 แบบรูป parallel split [9]

2.3.1.3 แบบรูป Synchronization

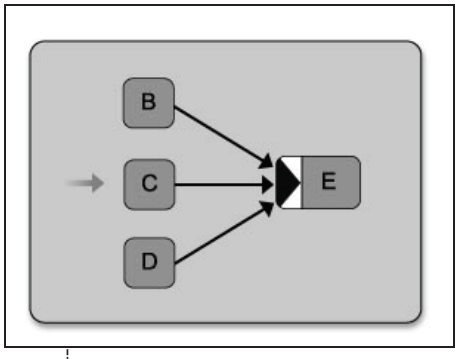
- ลักษณะ

การรวมกระแสงานควบคุม ตั้งแต่สองกระแสงานขึ้นไปเข้าด้วยกัน โดยกระแสงานถูกส่งไปที่กิจกรรมต่อไปเมื่อทุกกระแสงานควบคุมเกิดขึ้น

- ตัวอย่าง

กิจกรรมจัดส่งสินค้าเริ่มขึ้นทันทีหลังจากเกิดกิจกรรมตรวจสอบใบแจ้งหนี้และกิจกรรมออกใบแจ้งหนี้เสร็จสิ้น

สามารถเขียนแบบรูป Synchronization ในภาษายอวล์ได้ดังภาพที่ 2.8



ภาพที่ 2.8 แบบรูป Synchronization [9]

2.3.1.4 แบบรูป Exclusive Choice

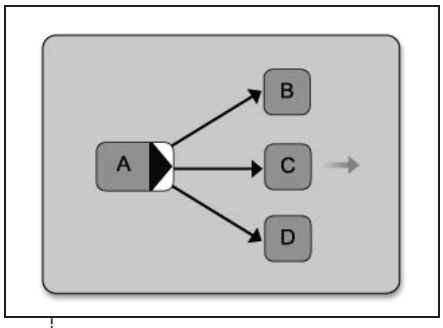
- ลักษณะ

แบบรูปนี้มีลักษณะการแยกการทำงานของกระแสงานควบคุมตั้งแต่สองกระแสงานขึ้นไป โดยกระแสงานควบคุมเข้ามาหนึ่งกระแสงาน และเกิดการเลือกการทำงานตามเงื่อนไขที่เกี่ยวข้องของกระแสงาน

- ตัวอย่าง

หลังจากกิจกรรมการเลือกตั้งเสร็จสิ้น กิจกรรมประกาศผลหรือกิจกรรมนับคะแนนจะเกิดการทำงาน

สามารถเขียนแบบรูป Exclusive Choice ในภาษายอวล์ได้ดังภาพที่ 2.9



ภาพที่ 2.9 แบบรูป Exclusive Choice [9]

2.3.1.5 แบบรูป Simple Merge

- ลักษณะ

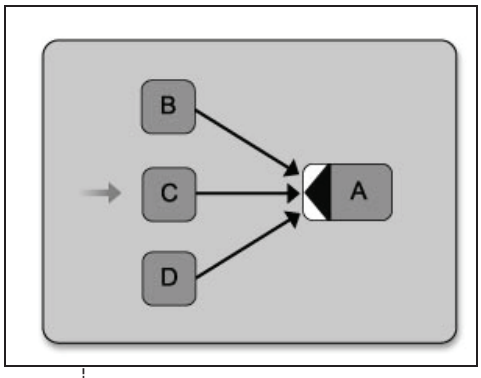
แบบรูปนี้มีลักษณะเป็นการรวมกระแสงานควบคุมตั้งแต่สองกระแสงานขึ้นไปเข้าด้วยกัน โดยการรวมกระแสงานเกิดขึ้นไม่พร้อมกัน แต่ละกระแสงานจะปรากฏครั้งเดียว

ในกระบวนการ

- ตัวอย่าง

กิจกรรมจ่ายเงินสดหรือจ่ายเงินผ่านบัตรเครดิต กิจกรรมทั้งสองเกิดขึ้นอย่างใดอย่างหนึ่งหรือทั้งสองกิจกรรมส่งต่อไปกิจกรรมชำระเงิน

สามารถเขียนแบบรูป Simple Merge ในภาษายอวล์ได้ดังภาพที่ 2.10



ภาพที่ 2.10 แบบรูป Simple Merge [9]

2.3.2 แบบรูปการแยกและผสานกระแสงานขั้นสูง

ในส่วนนี้นำเสนอแบบรูปที่มีความซับซ้อนมากขึ้นโดยมีลักษณะการแยกและการผสานกระแสงานที่มีความซับซ้อนขึ้น โดยเป็นแบบรูปที่เกิดขึ้นในกระบวนการทางธุรกิจจริง ถึงแม้แบบรูปกลุ่มนี้ค่อนข้างไม่ซับซ้อน แต่แบบรูปเหล่านี้มักไม่ได้รองรับในหลายภาษาที่ใช้นำเสนอกระบวนการธุรกิจต่างๆ แบบรูปในกลุ่มนี้มี 4 แบบรูปคือ แบบรูป Multi-Choice แบบรูป Synchronizing Merge แบบรูป Multi-Merge และแบบรูป Discriminator ด้วยอธิบายได้ดังนี้

2.3.2.1 แบบรูป Multi-Choice

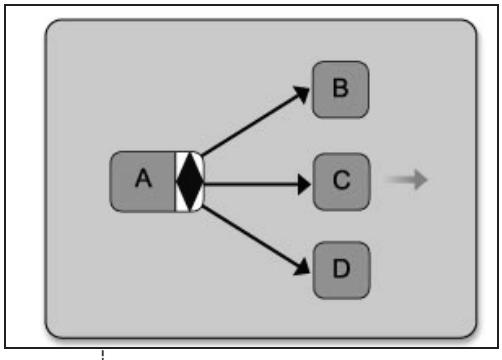
- ลักษณะ

เป็นแบบรูปที่ใช้ในการแยกกระแสงานควบคุมตั้งแต่สองกระแสงานขึ้นไป เมื่อมีกระแสงานควบคุมเข้ามาในกิจกรรมทำให้เกิดการทำงานตั้งแต่ 1 งานขึ้นไปโดยตามผลของนิพจน์ตรรกะที่เกี่ยวข้อง

- ตัวอย่าง

ในสถานการณ์ฉุกเฉิน อาจเกิดกิจกรรมเรียกตำรวจ เรียกพนักงานดับเพลิงหรือเรียกรถโรงพยาบาล กิจกรรมเหล่านี้สามารถเกิดขึ้นอย่างใดอย่างหนึ่งหรือเกิดขึ้นมากกว่าหนึ่งกิจกรรม

สามารถเขียนแบบรูป Multi-Choice ในภาษายอวล์ได้ดังภาพที่ 2.11



ภาพที่ 2.11 แบบรูป Multi-Choice [9]

2.3.2.2 แบบรูป Structured Synchronizing Merge

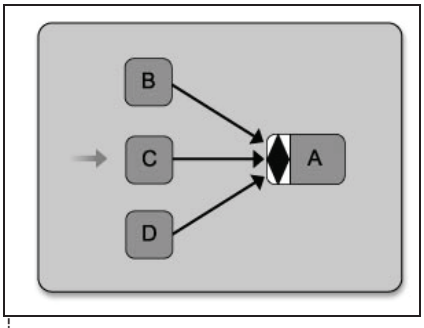
- ลักษณะ

เป็นแบบรูปในการร่วมกระแสงานตั้งแต่ 2 กระแสงานควบคุมขึ้นไปซึ่งก่อนหน้านี้เกิดการแยกออกในกระบวนการ โดยแต่ละกระแสงานจะเป็นกระแสงานที่ไม่ซ้ำกัน

- ตัวอย่าง

ในกรณีสถานการณ์ฉุกเฉินเกิดกิจกรรมขึ้นอย่างใดอย่างหนึ่ง คือเกิดกิจกรรมเรียกตำรวจและกิจกรรมเรียกรถพยาบาล ซึ่งกิจกรรมทั้ง 2 เริ่มพร้อมกัน และเมื่อทั้งสองกิจกรรมมาถึงที่เกิดอุบัติเหตุ กิจกรรมการขนส่งผู้บาดเจ็บจึงสามารถเริ่มขึ้น

สามารถเขียนแบบรูป Synchronizing Merge ในภาษายอวล์ได้ดังภาพที่ 2.12



ภาพที่ 2.12 แบบรูป Synchronizing Merge [9]

2.3.2.3 แบบรูป Multi-Merge

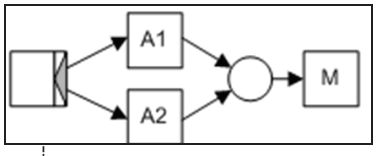
- ลักษณะ

การร่วมกระแสงานตั้งแต่ 2 กระแสงานขึ้นไป โดยแต่ละกระแสงานที่เกิดขึ้น จะถูกส่งไปยังกระแสงานที่ถัดมา

- ตัวอย่าง

กิจกรรมการก่อสร้าง สามารถเกิดกิจกรรมการสั่งซื้อ และกิจกรรมหาคนงาน กิจกรรมทั้ง 2 สามารถเกิดขึ้นพร้อมกันแบบคู่ขนาน หลังจากแต่ละกิจกรรมเสร็จสิ้น กิจกรรมตรวจสอบคุณภาพเกิดขึ้นเพื่อตรวจสอบคุณภาพของกิจกรรมทั้ง 2

สามารถเขียนแบบรูป Multi-Merge ในภาษายอวล์ได้ดังภาพที่ 2.13



ภาพที่ 2.13 แบบรูป Multi-Merge [9]

2.3.2.4 แบบรูป Discriminator

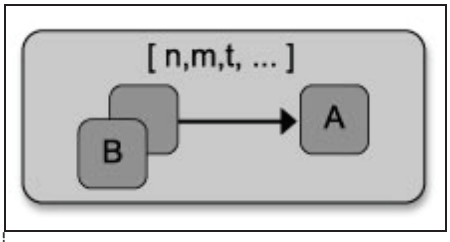
- ลักษณะ

เป็นแบบรูปที่อนุญาตให้เกิดการทำงานซ้ำตามจำนวน n ครั้งที่กำหนดในกระแสงานควบคุมโดยการทำงานของแบบรูปจะเสร็จสิ้นเมื่อการทำงานจำนวน m เสร็จสิ้น

- ตัวอย่าง

การจัดการกับภาวะหัวใจหยุดเต้น กระบวนการตรวจสอบการหายใจและการตรวจชีพจรเกิดขึ้นคู่กันเสมอ เมื่อสามารถตรวจเจอสิ่งใดสิ่งหนึ่งนั้น จะเกิดกิจกรรมคัดกรองขึ้น กิจกรรมการทำงานอื่นก่อนหน้าจะถูกยกเลิกและไม่เกิดผลลัพธ์

สามารถเขียนแบบรูป Discriminator ในภาษายอวล์ได้ดังภาพที่ 2.14



ภาพที่ 2.14 แบบรูป Structured Discriminator [9]

2.3.3 แบบรูปกลุ่มโครงสร้าง

แบบรูปกลุ่มโครงสร้างเป็นแบบรูปที่มีโครงสร้างแบบเฉพาะเจาะจง โดยมี 2 ส่วนที่สนใจในโครงสร้าง

1) รูปแบบของการวนซ้ำในขั้นตอนของแบบจำลอง

2) ความถูกต้องการสิ้นสุดของกระบวนการและ ความชัดเจนภายในโครงสร้าง การวนซ้ำถือเป็นโครงสร้างทั่วไปที่เกิดขึ้น ในระหว่างการออกแบบจำลองกระบวนการในสถานการณ์ หรือกลุ่มของกิจกรรมที่วนซ้ำ ในการเขียนโปรแกรมเหล่านี้การใช้คำสั่งวนซ้ำอยู่บนคำสั่ง go to

ซึ่งทำให้เกิดลักษณะการทำงานแบบไร้โครงสร้าง คือ เกิดการเข้าหรือออกในกระแสงานควบคุมมากกว่า 1 เส้นทาง

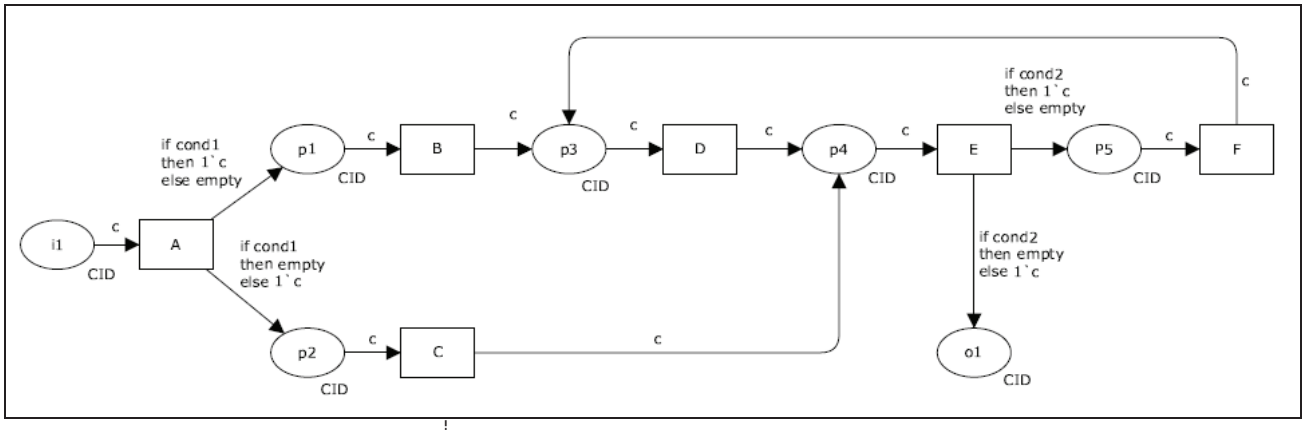
2.3.3.1 แบบรูป Arbitrary Cycle

- ลักษณะ

เกิดการวนซ้ำของกระบวนที่มีมากกว่า 1 ทางเข้าหรือ 1 ทางออก โดยทำให้เกิดการวนซ้ำแบบไร้โครงสร้าง

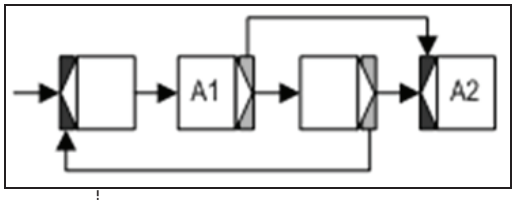
- ตัวอย่าง

ในภาพที่ 2.15 แสดงกระบวนการที่เป็นแบบรูป arbitrary cycle ที่เกิด 2 ทาง ในรอบการวนซ้ำ ที่ตำแหน่ง P3 และ P4



ภาพที่ 2.15 ลักษณะแบบรูป arbitrary cycle [9]

สามารถเขียนแบบรูป arbitrary cycle ในภาษายอวล์ได้ดังภาพที่ 2.16



ภาพที่ 2.16 แบบรูป arbitrary cycle [9]

2.3.4 แบบรูปหลายอ็อบเจกต์

แบบรูปหลายอ็อบเจกต์ สามารถเกิดการทำงานได้หลายหน่วยการทำงานย่อย (Thread) ของในกิจกรรมเดียวกัน สามารถแบ่งได้ 4 แบบรูป คือ

2.3.4.1 แบบรูป Multiple Instances without Synchronization

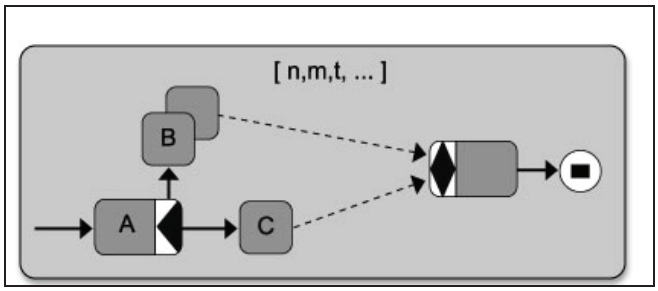
- ลักษณะ

เกิดการทำงานขึ้นภายในกิจกรรมที่สามารถเกิดหลายอ็อบเจกต์ได้ โดยแต่ละอ็อบเจกต์สามารถเกิดและทำงานควบคู่ได้อิสระและ ไม่เกิดทำงานพร้อมกันจนเสร็จสิ้น

- ตัวอย่าง

การวางไข่ของสัตว์ เกิดการวางไข่พร้อมกันแต่การฟักเป็นตัวไม่จำเป็นต้องเกิดพร้อมกัน

สามารถเขียนแบบรูป Multiple Instances without Synchronizationในภาษายอวล์ได้ดัง ภาพที่ 2.17



ภาพที่ 2.17 แบบรูป Multiple Instances without Synchronization [9]

2.3.4.2 แบบรูป Multiple Instances with a priori Design-Time Knowledge

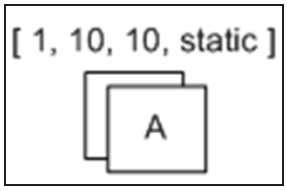
- ลักษณะ

เกิดการทำงานขึ้นภายในกระบวนการที่สามารถเกิดได้หลายอ็อบเจกต์ โดยจำนวนเกิดอ็อบเจกต์ทราบที่ขั้นตอนออกแบบ โดยแต่ละกรณีของอ็อบเจกต์เป็นอิสระต่อกัน และแต่ละกิจกรรมของอ็อบเจกต์จะต้องเสร็จพร้อมกันทั้งหมด

- ตัวอย่าง

รายงานประจำปีต้องลงนามของคณะกรรมการทั้งหมดก่อน จึงจะสามารถส่งรายงานการประชุมได้

สามารถเขียนแบบรูป Multiple Instances with a priori Design-Time Knowledge ในภาษายอวล์ได้ดัง ภาพที่ 2.18



ภาพที่ 2.18 แบบรูป Multiple Instances with a priori Design-Time Knowledge [9]

2.3.4.3 แบบรูป Multiple Instances with a priori Run-Time Knowledge

- ลักษณะ

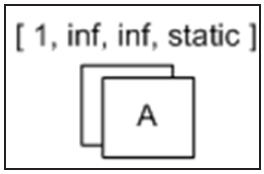
เกิดการทำงานขึ้นภายในกระบวนการที่สามารถเกิดได้หลายอ็อบเจกต์ โดยจำนวนอ็อบเจกต์ที่เกิดขึ้นทราบตอนปฏิบัติงานก่อนสร้างอ็อบเจกต์ และกิจกรรมของอ็อบเจกต์ทั้งหมดจะต้องเสร็จสิ้นพร้อมกัน

- ตัวอย่าง

- ในกิจกรรมตรวจซ่อมเครื่องยนต์การตรวจสอบเซ็นเซอร์สามารถเกิดขึ้นได้หลายครั้งขึ้นอยู่กับจำนวนของข้อความแสดงข้อผิดพลาดที่เกิดขึ้น โดยสามารถระบุจำนวนกิจกรรมตรวจสอบที่เกิดขึ้นได้ที่ขั้นตอนทำงาน

- เมื่อมีการเตรียมยา ในกิจกรรมจ่ายยาหรือเขียนใบสั่งยาเกิดขึ้นจะต้องพิจารณา ปัจจัยทางด้านน้ำหนักตัวของผู้ป่วย เพื่อใช้ดูส่วนประกอบหรือปริมาณยาก่อนการเตรียมยา และจ่ายยาออกไป ในผู้ป่วยแต่ละราย

สามารถเขียนแบบรูป Multiple Instances with a priori Run-Time Knowledge ในภาษายอวล์ได้ดังภาพที่ 2.19



ภาพที่ 2.19 แบบรูป Multiple Instances with a priori Run-Time Knowledge [9]

2.3.4.4 แบบรูป Multiple instances without a priori run-time knowledge

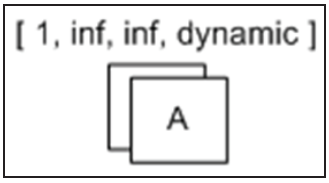
- ลักษณะ

เกิดการทำงานขึ้นภายในกิจกรรมที่สามารถเกิดขึ้นได้หลายอ็อบเจกต์ ซึ่งจำนวนอ็อบเจกต์สร้างขึ้นที่ตอนปฏิบัติงาน โดยจำนวนของอ็อบเจกต์สามารถเพิ่มขึ้นได้ ขึ้นอยู่กับปัจจัยด้านข้อมูล หรือการสื่อสารระหว่างกระบวนการ ดังนั้นการทำงานจะไม่เสร็จสิ้นจนกว่าอ็อบเจกต์สุดท้ายทำงานเสร็จสิ้น โดยแต่ละอ็อบเจกต์สามารถเกิดการทำงานควบคู่กันได้อย่างอิสระ และไม่เกิดการทำงานพร้อมกัน

- ตัวอย่าง

การขนส่งแทนขุดเจาะน้ำมันจากโรงงานไปยังที่ขุดเจาะน้ำมัน เกิดกิจกรรมขนส่งที่เกี่ยวข้องมากมาย กิจกรรมขนส่งอุปกรณ์ที่เกี่ยวข้องสามารถประมาณจำนวนการขนส่งที่จำเป็นได้ แต่เป็นไปได้ที่อาจต้องการอุปกรณ์เพิ่มเติม จึงจำเป็นต้องเพิ่มจำนวนขนส่งเพิ่ม เมื่อแท่นขุดเจาะน้ำมันทั้งหมดได้รับการขนส่งเสร็จสมบูรณ์ กิจกรรมถัดไปจึงสามารถเกิดขึ้นได้

สามารถเขียนแบบรูป Multiple instances without a priori run-time knowledge ในภาษายอวล์ได้ดังภาพที่ 2.20



ภาพที่ 2.20 แบบรูป Multiple instances without a priori run-time knowledge [9]

2.3.5 แบบรูปสถานะ

แบบรูปสถานะคือแบบรูปที่แก้ปัญหาของภาษากระบวนการให้รองรับแนวคิดสถานะของกระแสงาน โดยในแบบรูปกลุ่มนี้พิจารณาสถานะของกระแสงาน รวมถึงเกี่ยวกับข้อมูลสถานะของกิจกรรม แบบรูปในกลุ่มนี้สามารถแบ่งได้ 3 แบบรูปคือ

1) แบบรูป Deferred choice การตัดสินใจการทำงานของกิจกรรมโดยขึ้นอยู่กับสภาพแวดล้อมการทำงาน

2) แบบรูป Interleaved Parallel Routing คือกระแสงานตั้งแต่สองกระแสงานขึ้นไป โดยมี 1 กิจกรรมเท่านั้นที่สามารถดำเนินการได้

3) แบบรูป Milestone คือแบบรูปที่กิจกรรมเกิดการทำงานขึ้นเฉพาะกระบวนการที่อยู่ในสถานะเฉพาะเจาะจง

2.3.5.1 แบบรูป Deferred Choice

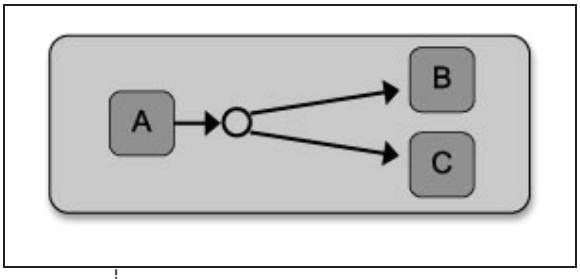
- ลักษณะ

แบบรูป Deferred Choice เป็นแบบรูปที่กิจกรรม จะเกิดการทำงานขึ้น โดยขึ้นอยู่กับปฏิสัมพันธ์กับสภาพแวดล้อมในการทำงาน ในการตัดสินใจดำเนินกิจกรรม

- ตัวอย่าง

เมื่อลูกค้าต้องการสั่งถุงลมนิรภัยสามารถเลือกวิธีการส่งอย่างใดอย่างหนึ่ง เช่น ไปรษณีย์ หรือการจัดส่งทางรถยนต์

สามารถเขียนแบบรูป Deferred Choice ในภาษายอวล์ได้ดังภาพที่ 2.21



ภาพที่ 2.21 แบบรูป Deferred Choice [9]

2.3.5.2 แบบรูป Interleaved Parallel Routing

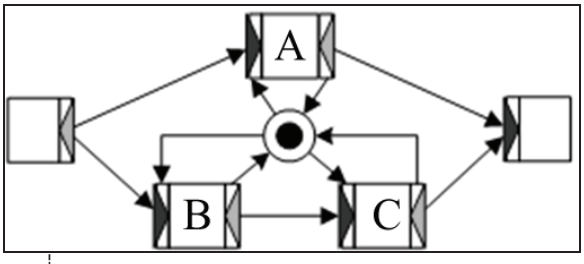
- ลักษณะ

เป็นแบบรูปของกิจกรรมที่ 2 กิจกรรมไม่สามารถดำเนินการได้พร้อมกันในเวลาเดียวกัน

- ตัวอย่าง

ในการซื้อสินค้ามีกิจกรรมที่เกี่ยวข้องคือ กิจกรรมจัดเตรียมใบแจ้งหนี้(A) กิจกรรมเลือกสินค้า(B) และกิจกรรมแพ็คสินค้า(C) เมื่อเกิดรายการสั่งซื้อสินค้าขึ้น กิจกรรมเลือกสินค้าจะต้องทำงานก่อนกิจกรรมแพ็คสินค้า ส่วนกิจกรรมจัดเตรียมใบแจ้งหนี้ สามารถเกิดขึ้นได้ในเวลาใด ๆ โดยไม่เกิดกิจกรรมทำงานพร้อมกัน

สามารถเขียนแบบรูป Interleaved Parallel Routing ในภาษายอวล์ได้ดังภาพที่ 2.22



ภาพที่ 2.22 แบบรูป Interleaved Parallel Routing [9]

2.3.5.3 แบบรูป Milestone

- ลักษณะ

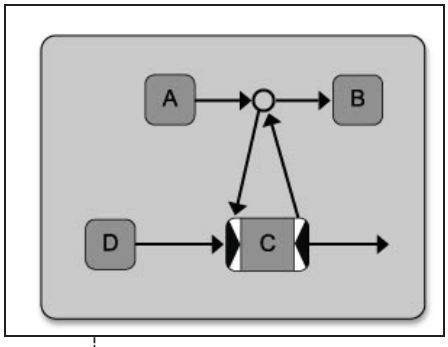
แบบรูปนี้เป็นกิจกรรมที่สามารถเกิดการทำงานขึ้น เมื่อกิจกรรมสามารถกำหนดสถานะที่เฉพาะเจาะจง โดยมีตัวดำเนินการชี้เฉพาะ (milestone) เมื่อกระแสงานดำเนินการถึงตัวดำเนินการชี้เฉพาะ(กิจกรรม C ในภาพที่2.23) สามารถเลือกเปิดใช้งานสถานะได้ ถ้ากระแสงานทำงานเกินกว่าสถานะที่กำหนดนี้แล้วกิจกรรมไม่สามารถเกิดการทำงานได้

- ตัวอย่าง

- ในระบบสายการบินอนุญาตให้ลูกค้าย้ายที่นั่งได้ก่อนออกบัตรผ่านขึ้นเครื่อง (boarding pass)

- กิจกรรมการลงทะเบียนของนักศึกษา(กิจกรรมB)สามารถลงทะเบียนเทอมใหม่ได้ตามช่วงที่กำหนดให้ลงทะเบียน และก่อนกิจกรรมปิดการลงทะเบียนจะเกิดขึ้น (กิจกรรมC)

สามารถเขียนแบบรูป milestone ในภาษายอวล์ได้ดังภาพที่ 2.23



ภาพที่ 2.23 แบบรูป milestone [9]

2.3.6 แบบรูปการยกเลิก (Cancellation Patterns)

เป็นแบบรูปที่มีแนวคิดของกิจกรรมยกเลิกการทำงาน คือมีส่วนที่ทำให้สามารถเกิดกิจกรรมขึ้นหรือยกเลิกกิจกรรมที่เกิดขึ้น ในส่วนนี้นำเสนอแนวคิดการยกเลิก ของกระบวนการอยู่บนพื้นฐาน exception handling มี 2 แบบรูปคือ แบบรูป Cancel Activity และ แบบรูป Cancel Case

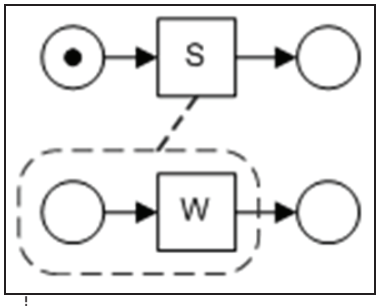
2.3.6.1 แบบรูป Cancel Activity

- ลักษณะ

เป็นกิจกรรมที่เกิดขึ้นเพื่อยกเลิกการทำงานของกิจกรรมก่อนหน้าที่จะเริ่มดำเนินการและในกรณีที่กิจกรรมทำงานอยู่จะหยุดการทำงาน

- ตัวอย่าง

ผู้ซื้อสามารถเปลี่ยนรายการซื้อ(กิจกรรมS)ได้ตลอดเวลาก่อนทำรายการจ่ายเงิน(กิจกรรมW)สามารถเขียนแบบรูป Cancel Activity ในภาษายอวล์ได้ดังภาพที่ 2.24



ภาพที่ 2.24 แบบรูป Cancel Activity [9]

2.3.6.2 แบบรูป Cancel Case

- ลักษณะ

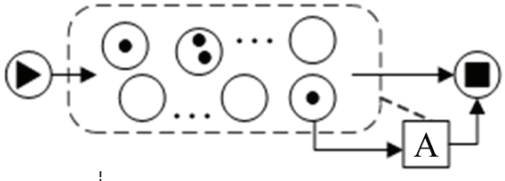
เป็นกระบวนการยกเลิกอย่างสมบูรณ์ ซึ่งอาจเป็นกิจกรรมที่ดำเนินอยู่ กิจกรรมเกิดขึ้นในอนาคต ทุกกิจกรรมในกระบวนการย่อย โดยทุกกระบวนจะถูกยกเลิกทั้งหมด

- ตัวอย่าง

- ในกิจกรรมการจำนองผู้ซื้อจะตัดสินใจไม่ดำเนินการซื้อบ้าน(A)และยกเลิกใบซื้อรายการซื้อบ้านได้ตลอดเวลา

- ในระหว่างขั้นตอนการเคลมประกัน เมื่อพบนโยบายเคลมประกันดังกล่าวได้หมดอายุและเป็นผลให้กิจกรรมทั้งหมดที่เกี่ยวกับกระบวนการถูกยกเลิก

สามารถเขียนแบบรูป Cancel Case ในภาษายอวล์ได้ดังภาพที่ 2.25



ภาพที่ 2.25 แบบรูป Cancel Case [9]

2.4 **อัลกอริทึมแฮมมอคกราฟ** [5]

อัลกอริทึมแฮมมอคกราฟ ใช้ในการปรับโปรแกรมให้มีโครงสร้าง โดยลักษณะโปรแกรมที่มีโครงสร้าง หมายถึงโปรแกรมที่มีเชิงบล็อก(Block oriented)โดยในแต่ละบล็อกของชุดคำสั่งโปรแกรมต้องมีลักษณะคือมีเพียง 1 ทางเข้า(single entry) และ 1 ทางออก(single exit) โดยโปรแกรมประกอบขึ้นจากโครงสร้างหลายบล็อกร่วมกัน การใช้ชุดคำสั่ง while และชุดคำสั่ง if-then-else ชุดคำสั่ง jump และชุดคำสั่ง go to ระหว่างบล็อกของโปรแกรมอาจทำให้โปรแกรมเกิดโครงสร้างแบบไร้โครงสร้างขึ้นคือ ไม่มีการทำงานที่แน่นอน(single locus) [11]

อัลกอริทึมแฮมมอคกราฟนี้จะทำการปรับชุดคำสั่งเงื่อนไขทั้งหมด โดยผลลัพธ์การปรับยังคงรักษาความยาวของโปรแกรม รักษาโครงสร้าง loop และโครงสร้าง if-test เดิม วิธีนี้ทำให้เกิดโครงสร้างแบบ 1 ทางเข้า ,1 ทางออก และโครงสร้างโปรแกรมอยู่ในโครงสร้างที่ดี

อัลกอริทึมแฮมมอคกราฟ ประกอบด้วยอัลกอริทึมพื้นฐาน 3 อัลกอริทึมคือ backward copy, forward copy และ cut operation ดูตัวอย่างการทำงานได้ในภาพที่ 2.26 ชุดคำสั่ง while ถูกสร้างขึ้นด้วยชุดคำสั่ง go to (บรรทัดที่ 100) ทำให้เกิดทางเข้า 2 ทางเข้า ทำให้เกิดโครงสร้างแบบไร้โครงสร้าง ต้องมีการปรับโครงสร้างโดยอัลกอริทึม backward copy ในรูป b และทำการปรับโครงสร้าง if ให้อยู่ในรูปโครงสร้างอย่างง่าย ได้ผลลัพธ์ตามรูป c มีความแตกต่างจากโครงสร้างโปรแกรม a คือมีความเหมาะสมไม่เกิดโครงสร้างแบบไร้โครงสร้าง สำหรับโครงสร้าง while loop อาจแปลงมาอยู่ในรูปโครงสร้าง parallel loop subject ขึ้นอยู่กับการวิเคราะห์

```
    if (x) goto 200          if(x) goto 200          if(.not. x) then
100 S1                   100 S1                        S1
200 S2                   200 S2                      endif
    if (y) goto 100          do while(y)             S2
                               S1                    do while(y)
                               S2                      S1
                             enddo                     S2
                                                     enddo
```

(a) (b) (c)

ภาพที่ 2.26 การเปลี่ยนรูปโครงสร้างโปรแกรม [5]

2.4.1 หลักการการทำงานอัลกอริทึม

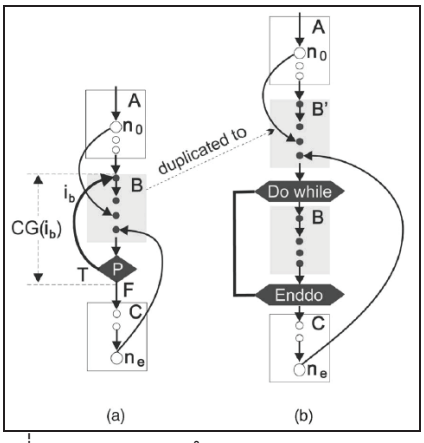
อัลกอริทึมแฮมมอคกราฟประกอบ โดยอัลกอริทึมย่อย 3 อัลกอริทึมคือ

1) อัลกอริทึม Cut Transformation

อัลกอริทึมนี้จะทำการแก้ปัญหาโครงสร้างโปรแกรมที่มีลักษณะ เกิดการเข้ามาในกระแสงานอื่น โดยอัลกอริทึมนี้ทำการเพิ่มตัวแปร b_{ri} เพื่อใช้ในการกำหนดเงื่อนไขที่เกิดขึ้นใน คำสั่ง loop และกำหนดส่วนการทำงานเป็น 2 ส่วนคือส่วนจบการทำงานของคำสั่งวนซ้ำด้วย คำสั่ง exit และส่วนการทำงานวนซ้ำ

2) อัลกอริทึม Backward Copy

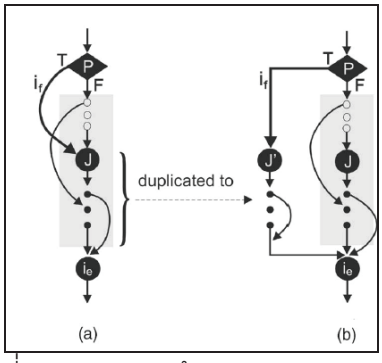
อัลกอริทึมทำการแก้ปัญหาเมื่อโปรแกรมมีโครงสร้างการทำงานย้อนกลับมาที่ชุดคำสั่งวนซ้ำทำให้เกิดการทำงานซ้อนทับกัน (Overlapping) สามารถดูภาพที่ 2.27 แสดงการวิธีการแก้ปัญหาโดยทำการกำหนด ne ในชุดคำสั่งวนซ้ำ i_b โดยที่ i_b ไม่สามารถเกิดการทำงานได้โดยตรงจึงจำเป็นกำหนดส่วน Duplicate code และสร้าง ชุดคำสั่งวนซ้ำด้วยโครงสร้าง While สามารถดูภาพที่ 2.27 ดูลักษณะการทำงานได้



ภาพที่ 2.27 อัลกอริทึม Backward Copy[5]

3) อัลกอริทึม Forward Copy

อัลกอริทึมจะทำงานหลังจากอัลกอริทึม Backward Copy ทำงานเสร็จแล้ว อัลกอริทึมนี้จะทำการกำหนดส่วนของคำสั่งเงื่อนไข ขึ้นมา 2 ส่วนคือ true_part ส่วนที่เกิดการทำงานเมื่อเงื่อนไขเป็นจริง และ false_part อัลกอริทึมจะทำการแก้ปัญหา Forward branch ทั้งหมด ด้วยทำการกำหนด *if* เพื่อทำการหาส่วน shared statement เพื่อทำการสร้างชุดคำสั่งโปรแกรมซ้ำ (Dupilcate code)สามารถดูภาพที่ 2.28 แสดงการทำงานได้



ภาพที่ 2.28 อัลกอริทึม Forward Copy [5]

2.4.2 ขั้นตอนการทำงานของอัลกอริทึม

แฮมมอคกราฟเป็นอัลกอริทึมในการปรับโครงสร้างโปรแกรมประกอบด้วย 5 ขั้นตอน อธิบายได้ดังนี้สามารถดูลักษณะการทำงานได้ในภาพที่ 2.29

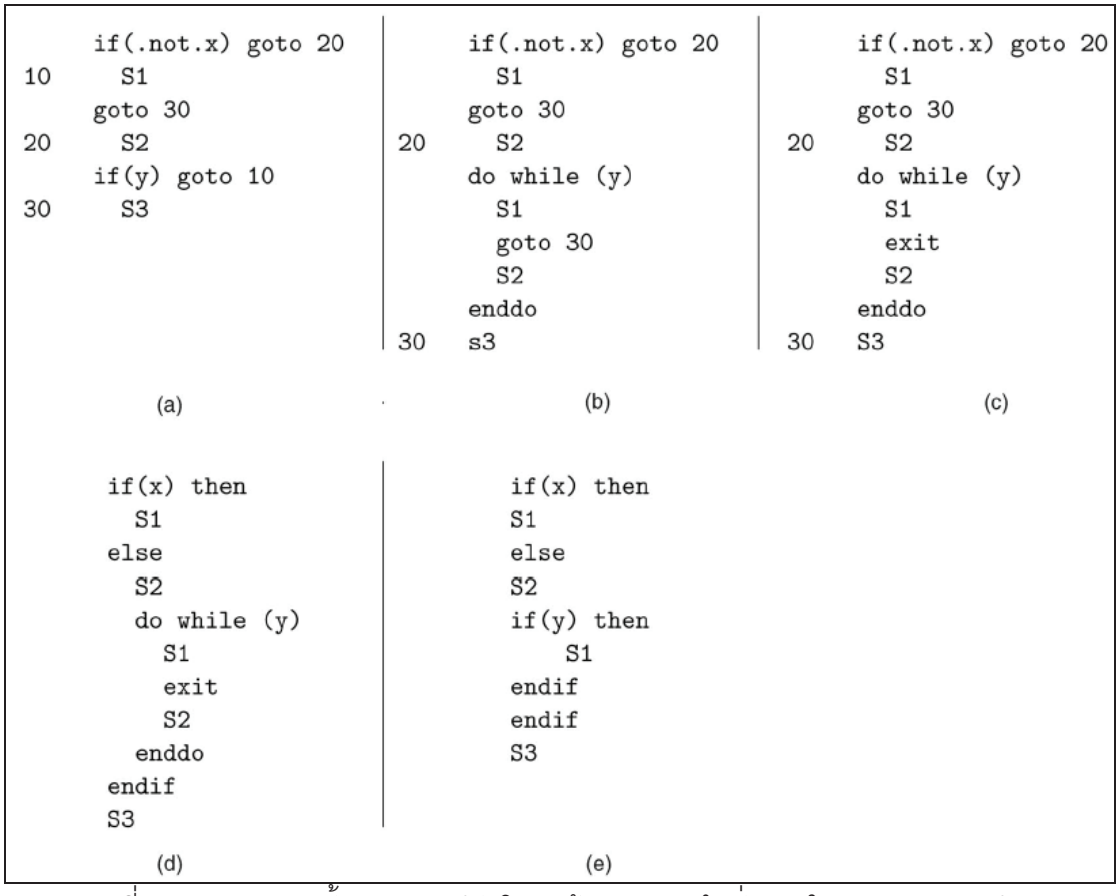
1) ขั้นตอน Preprocessing แทนชุดคำสั่ง if ระหว่าง branch โดยชุดคำสั่ง if-goto

2) ขั้นตอน Single branches แทน single branch โดยโครงสร้าง block-if หรือชุดคำสั่ง loop

3) ขั้นตอน Outgoing Branches แทน outgoing branch ใน loop โดยชุดคำสั่ง exit และใส่เงื่อนไข jump ไปที่เป้าหมายดังเดิมหลัง loop

4) ขั้นตอน Backward branches โดยแต่ละ backward branch จะถูกกำหนด minimal hammock graph รอบๆ ใน branch เริ่มต้นโดย backward branch แรก i_b ในกราฟจะแปลงเป็นเป็น loop และย้ายเป้าหมาย ของ incoming branch outside ใน loop body โดยตัดแต่ละ outgoing forward branch ใน loop ให้กลายเป็น exit jump followed ด้วยเงื่อนไข jump ไปยังเป้าหมายของ branch

5) ขั้นตอน Forward branch โดยแต่ละ forward branch จะถูกแทนโดย minimal hammock graph รอบ branch เริ่มต้นจาก forward branch แรกจะถูกแปลงเป็นโครงสร้างของชุดคำสั่ง if



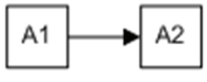
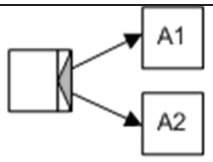
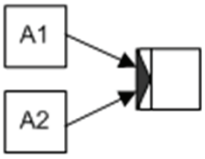
ภาพที่ 2.29 แสดงการขั้นตอนการปรับโครงสร้างของชุดคำสั่งภายในแฮมมอคกราฟ [5]

2.5 ลักษณะแบบรูปในกระแสงานยอวล์แบบรูปโครงสร้างที่ดี

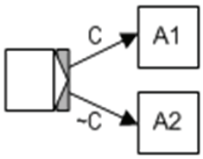
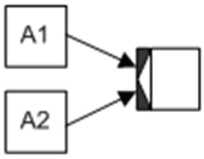
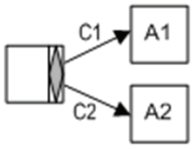
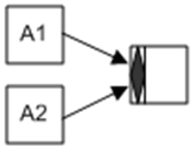
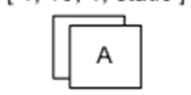
การตรวจสอบโครงสร้างในกระแสงานยอวล์ว่าเป็นแบบรูปโครงสร้างที่ดี (Well structured) ทำได้โดยการตรวจสอบว่ากระแสงานมีโครงสร้างตรงตามตารางที่ 2.2 จากแนวคิดของ C.OuYang [7]

ลำดับความสัมพันธ์ของแบบรูปจำเป็นต้องเริ่มและจบตามที่กำหนดในตารางที่ 2.2 เช่น เริ่มต้นจากแบบรูป Parallel Split จำเป็นต้องจบโดยแบบรูป Synchronization เป็นต้น

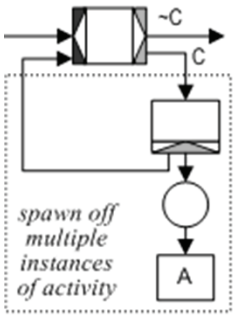
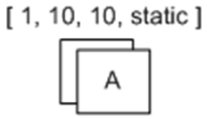
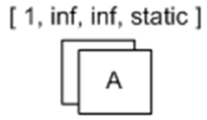
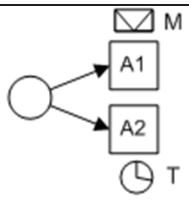
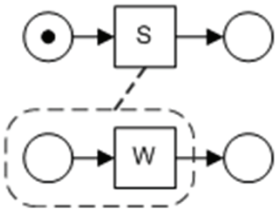
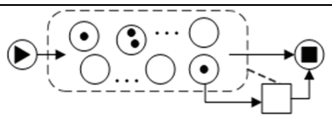
ตารางที่ 2.2 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล [7]

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
1	Sequence		<sequence> activity A1 activity A2 </sequence>
2	Parallel Split		<flow> activity A1 activity A2 </flow>
3	Synchronization		

ตารางที่ 2.2 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล [7] (ต่อ)

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
4	Exclusive Choice		<if> <condition>C</condition> activity A1 <else> activity A2 </else> </if>
5	Simple Merge		
6	Multi Choice		<flow> <links> <link name="SplitToA1"/> <link name="SplitToA2"/> <link name="A1ToMerge"/> <link name="A2ToMerge"/> </links> <empty name="Split"> <sources> <source linkName="SplitToA1"> <transitionCondition>C1</transitionConditio </source> <source linkName="SplitToA2"> <transitionCondition>C2</transitionCondition> </source> </sources> </empty> activity A1 <targets><target linkName="SplitToA1"/></targets> <sources><source linkName="A1ToMerge"/></sources> activity A2 <targets><target linkName="SplitToA2"/></targets> <sources><source linkName="A2ToMerge"/></sources> <empty name="Merge"> <targets> <joinCondition> $A1ToMerge OR $A2ToMerge </joinCondition> <target linkName="A1ToMerge"/> <target linkName="A2ToMerge"/> </targets> </empty> </flow>
7	Synchronizing Merge		
8	Discriminator		<forEach counterName="i" parallel="yes"> <startCouterValue>1</startCounterValue> <finalCouterValue>10</finalCounterValue> <CompletionCondition>1</completionConditi on> <scope> activity A </scope> </forEach>

ตารางที่ 2.2 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล [7] (ต่อ)

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
9	MI without Synchronization		<process name="Main"> ... <while> <condition>C</condition> <invoke process-MI ... /> </while> ... </process> <process> <receive process-Main ... creativieInstance="yes"/> activity A </process>
10	MI with a Priori Design-Time knowledge		<forEach counterName="i" parallel="yes"> <startCounterValue>1</startCounterValue> <finalCounterValue>10</finalCounterValue> <scope> activity A </scope> </forEach>
11	MI with a Priori Run-Time knowledge		<forEach counterName="i" parallel="yes"> <startCounterValue>1</startCounterValue> <finalCounterValue>x</finalCounterValue> <scope> activity A </scope> </forEach
12	Deferred Choice		<pick> <onMessage M> activity A1 </onMessage> <onAlarm> <until>T</until> activity A2 </onMessage> </pick>
13	Cancel Activity		<scope> <faultHandlers> <catch faultName="Wrong"> activity S </catch> </faultHandlers> <sequence> <throw faultName="Wrong"/> activity W </sequence> </process>
14	Cancel Case		<process> ... <exit/> ... </process>

2.6 งานวิจัยที่เกี่ยวข้อง

2.6.1 งานวิจัย Transformation of BPMN to YAWL ของ JianHong YE , ShiXin SUN, Lijie Wen และ Wen SONG [4]

งานวิจัยนี้ได้นำเสนอวิธีการเปลี่ยนรูปจากแผนภาพบีพีเอ็มเอ็น (BPMN) เป็นกระแสงานยอวล์ ผลจากการเปลี่ยนรูปทำให้เกิดความชัดเจนของอรรถศาสตร์ คือการวิเคราะห์กระบวนการทางธุรกิจทำได้ง่ายขึ้น โดยในงานวิจัยนี้ได้ทำการกำหนดกระบวนการหลักของแผนภาพบีพีเอ็มเอ็น (Core BPMN Process) ,แผนภาพบีพีเอ็มเอ็นแบบรูปที่ดี (Well formed core BPMN) ต่างๆ และนำเสนออัลกอริทึมในการเปลี่ยนรูปจากแผนภาพบีพีเอ็มเอ็นเป็นยอวล์ ผู้ทำวิทยานิพนธ์นำแนวคิดนี้มาเป็นแนวทางเทียบเคียงเพื่อแปลงกระบวนการทางธุรกิจที่อยู่ในรูปของกระแสงานยอวล์ให้เป็นโครงร่างบีเพลโดยแนวคิดอยู่ในงานวิทยานิพนธ์นี้

2.6.2 งานวิจัย From BPMN Process Models to BPEL Web Services ของ C. Ouyang, M. Dumas, A.H.M.T. Hofstede, และ W.M.P.V.D [3]

งานวิจัยนี้ทำการเปลี่ยนรูปจากแผนภาพบีพีเอ็มเอ็นเป็นภาษาบีเพลโดยได้กำหนดแบบรูปที่รู้จักดี (Well known pattern) ที่สามารถทำการเปลี่ยนรูปเป็นภาษาบีเพล เช่น แบบรูป sequence แบบรูป flow แบบรูป pick แบบรูป while เป็นต้น แต่ละแบบรูปจะถูกแทนเป็นส่วนประกอบ (Component) ที่ประกอบกันเป็นภาษาบีเพลได้ สำหรับส่วนที่ไม่ได้อยู่ในแบบรูปที่รู้จักดีจะได้รับการจัดการโดย Event action rule based translation โดยสร้างเป็นส่วนประกอบบีเพลอีกชุดหนึ่งขึ้นมาใหม่ แล้วนำไปอยู่ในส่วนของ event handlers ส่วนที่ไม่ได้อยู่ในแบบรูปที่รู้จักดีดังกล่าวนี้ผู้ทำวิทยานิพนธ์พบว่าไม่ใช่การแก้ปัญหาที่ได้ผลดี เพราะทำให้ภาษาบีเพลผลลัพธ์อ่านและเข้าใจยากเนื่องจากการมีส่วนไร้โครงสร้างอยู่ และการวิเคราะห์กระบวนการทางธุรกิจที่อยู่ในส่วนของแบบรูปที่รู้จักเป็นเรื่องยาก ผู้ทำวิทยานิพนธ์จึงนำแนวคิดเฉพาะการกำหนดแบบรูปมาประยุกต์ใช้ประโยชน์เท่านั้น

2.6.3 งานวิจัย Pattern-based Translation of BPMN Process Models to BPEL Web Services ของ C. Ouyang, M. Dumas, และ A.H.M.T. Hofstede [12]

งานวิจัยนี้ได้นำเสนอ กระบวนการเปลี่ยนแผนภาพบีพีเอ็มเอ็นเป็นภาษาบีเพลโดย งานวิจัยนี้นำเสนอเทคนิค Quasi-structured pattern-based คือการปรับเปลี่ยนแบบรูปที่มีลักษณะโครงสร้างแตกต่างกันเล็กน้อยจากแบบรูปที่รู้จักดี ให้มีลักษณะดีตามต้องการ ผู้ทำวิทยานิพนธ์จึงนำเสนอแนวคิดในการปรับแผนภาพในส่วนกระบวนการทางธุรกิจ ให้อยู่ในรูปของแบบรูปที่รู้จักก่อน

2.6.4 งานวิจัย Using Hammock Graphs to Structure Programs ของ F. Zhang และ E.H. D'Hollander [5]

งานวิจัยนี้ นำเสนออัลกอริทึมในการแปลงจากโปรแกรมไร้โครงสร้างให้อยู่ในรูปของโปรแกรมมีโครงสร้าง เนื่องจากโปรแกรมที่อยู่ในรูปแบบไร้โครงสร้างทำให้การวิเคราะห์ และการควบคุมสายงานทำได้ยาก การมีชุดคำสั่ง branch, jump , go to ทำให้การประมวลผลใช้เวลานาน

ผู้ทำวิทยานิพนธ์จึงใช้อัลกอริทึมแฮมมอคมาประยุกต์ใช้ในการเปลี่ยนกระแสงานยอวล์ที่มีโครงสร้างไม่ดี ให้มีโครงสร้างที่ดีก่อนแล้วจึงทำการเปลี่ยนรูปเป็นโครงร่างบีเพล

2.6.5 งานวิจัย Formal Semantics of BPMN Process Models Using YAWL ของ J. Ye, S. Sun, W. Song, และ L. Wen, [13]

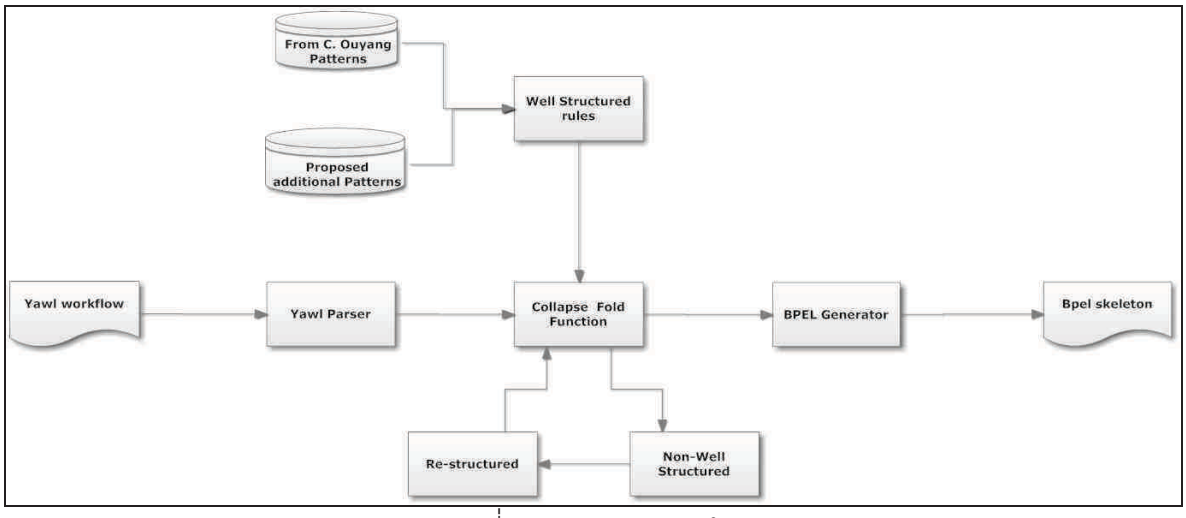
งานวิจัยนี้ได้นำเสนอ กระบวนการเปลี่ยนแผนภาพบีพีเอ็มเอ็นเป็นกระแสงานยอวล์โดยนำเสนอ Formal Semantics ในการเปลี่ยนรูปแผนภาพเพื่อช่วยในการกำหนดความหมายของแบบรูป ก่อนเปลี่ยนรูปเป็นภาษายอวล์ ผู้ทำวิทยานิพนธ์จึงนำแนวคิดในนำเสนอกระบวนการทางธุรกิจที่มีความชัดเจน Formal Semantics บนกระแสงานยอวล์

บทที่ 3
การวิเคราะห์และออกแบบวิธีการแปลงรูปกระแสงานยอวล์

ในบทนี้จะกล่าวถึงการวิเคราะห์และออกแบบเครื่องมือการแปลงรูปกระแสงานยอวล์เป็นโครงร่างบีเพลของวิทยานิพนธ์นี้ เริ่มต้นโดยแสดงภาพรวมแนวคิดของเครื่องมือ อัลกอริทึม Fold function โครงสร้างการทำงานของเครื่องมือ และแสดงการออกแบบเครื่องมือด้วยแผนภาพยูสเคส (Use case diagram) แผนภาพคลาส (Class diagram) แผนภาพลำดับ (Sequence diagram)

3.1 ภาพรวมแนวคิดของเครื่องมือ

วิทยานิพนธ์นี้กำหนดให้ กระแสงานยอวล์แบ่งตามประเภทแบบรูปที่รองรับได้ 2 ประเภท คือ แบบรูปโครงสร้างที่ดี (Well Structured) และ แบบรูปโครงสร้างที่ไม่ดี(Non well Structured)



ภาพที่ 3.1 ภาพรวมการทำงาน

โดยจากภาพที่ 3.1 ผู้ใช้นำกระแสงานยอวล์ (Yawl workflow) เข้าสู่ระบบ ระบบทำการอ่านกระแสงานยอวล์ (Yawl Parse) ให้อยู่ในรูปกราฟสายงานควบคุม ถัดไประบบทำการยุบกราฟสายงานควบคุม (Collapse Fold Function) โดยใช้กฎการยุบโครงสร้างที่ดี (Well structured rules) จากแนวคิดของ C.OuYang (From C.OuYang Patterns) [7] และจากแนวคิดของวิทยานิพนธ์ ถ้าไม่สามารถยุบกราฟสายงานควบคุมได้จะถือว่าสายงานควบคุมนั้น เป็นโครงสร้างไม่ดี (Non-well Structured) ซึ่งจะต้องผ่านการปรับโครงกระแสงานใหม่ (Re-structured) จากนั้นนำไปผ่านการยุบกราฟสายงานควบคุมอีกครั้ง ในท้ายที่สุดระบบจะสร้างไฟล์โครงร่างบีเพล (BPEL Generator) เพื่อส่งไฟล์ข้อมูลโครงร่างบีเพล(BPEL skeleton)ให้กับผู้ใช้

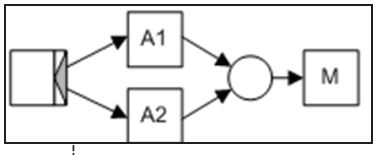
3.1.1 แบบรูปโครงสร้างที่ดี หมายถึงแบบรูปของกระแสงานยอวล์ที่ได้รับเปลี่ยนเป็นโครงร่างบีเพลได้โดยไม่จำเป็นต้องปรับเปลี่ยนโครงสร้างเพิ่มเติม ในวิทยานิพนธ์นี้กำหนดไว้ 18 แบบรูปซึ่งครอบคลุมดังนี้

1) แบบรูปจากแนวคิดของ C.OuYang (Form C.OuYang Pattern) [7] 11 แบบรูป คือ แบบรูป Sequence แบบรูป Parallel Split แบบรูป Synchronization แบบรูป Exclusive Choice แบบรูป Simple Merge แบบรูป Multi Choice แบบรูป Discriminator แบบรูป Synchronizing Merge แบบรูป MI with a Priori Design-Time knowledge แบบรูป MI with a Priori Run-Time knowledge และแบบรูป Cancel Case แสดงรายละเอียดในตาราง 2.2

2) แบบรูปที่วิทยานิพนธ์นี้ออกแบบเพิ่มเติม หมายถึงแบบรูปที่ [7] ไม่รองรับ เพราะแบบรูปนั้นมีลักษณะเป็น Non free choice คือ เป็นแบบรูปมีการขึ้นอยู่กันโดยนัย (implicit dependencies) จึงไม่จำเป็นใช้กลุ่มข้อมูลนำเข้าชุดเดียวกัน ทำให้เกิดโครงสร้างที่ไม่แน่นอนได้ [14]

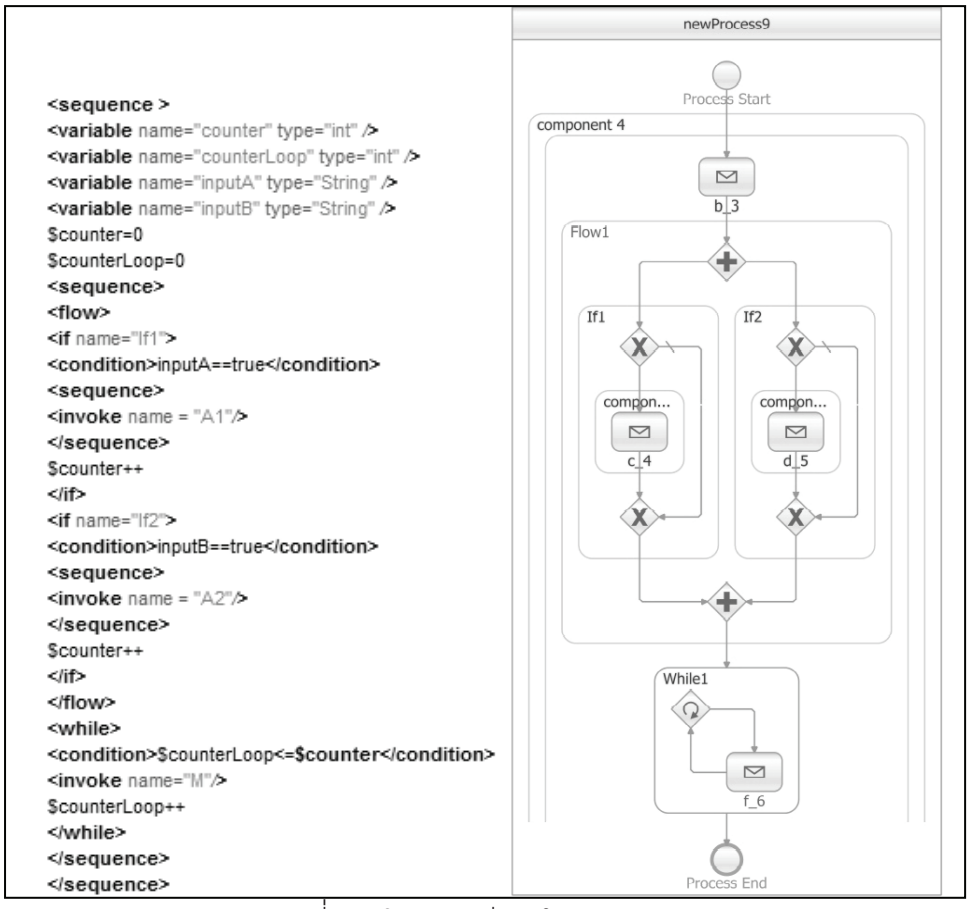
แบบรูปในกลุ่มนี้มี 4 แบบรูปคือแบบรูป Multi Merge แบบรูป MI without a Priori Run-Time Knowledge แบบรูป Interleaved Parallel Routing และแบบรูป Milestone เนื่องจากแบบรูปดังกล่าวได้รับการกำหนดไว้ใน [9] สามารถดูรายละเอียดเพิ่มเติมดังนี้

- แบบรูป Multi Merge คือ แบบรูปที่ร่วมกระแสงานตั้งแต่ 2 กระแสงานขึ้นไป โดยแต่ละกระแสงานที่เกิดขึ้น จะถูกส่งไปยังกระแสงานถัดมา [9] ในภาพที่ 3.2 แสดงลักษณะแบบรูป Multi merge ในภาษายอวล์



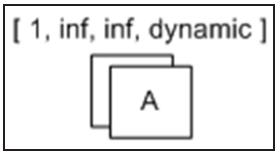
ภาพที่ 3.2 แบบรูป Multi Merge

โดยลำดับการทำงานของแบบรูปนี้สามารถเกิดขึ้นคือ <A1, A2, M, M> < A1, M> <A2, M> มีการทำงานซ้ำที่เซอร์วิส M เท่าจำนวนการทำงานที่ เซอร์วิส A1 และเซอร์วิส A2 เราสามารถเปลี่ยนแบบรูปนี้เป็นโครงร่างบีเพลได้ตามภาพที่ 3.3 โดยเครื่องมือสร้างตัวแปร counter และ counterLoop ชนิด Integer เพื่อนับจำนวนการทำงานที่เซอร์วิส A1 และเซอร์วิส A2 หลังจากเกิดการทำงานเซอร์วิส A1 และเซอร์วิส A2 จะเจอโครงสร้าง while เพื่อเกิดการทำงานซ้ำที่เซอร์วิส M เท่ากับจำนวนในตัวแปร counter โดยด้านขวาของรูปแสดงโครงร่างบีเพลที่เครื่องมือสร้างขึ้น และด้านซ้ายแสดงชุดคำสั่งบีเพลที่เครื่องมือสร้างขึ้น



ภาพที่ 3.3 โครงร่างบีเพลในแบบรูป Multi Merge

- แบบรูป MI without a Priori Run-Time Knowledge เป็นแบบรูปที่กิจกรรมสามารถเกิดการทำงานขึ้นได้หลายอ็อบเจกต์ จำนวนของอ็อบเจกต์สร้างขึ้นตอนปฏิบัติงาน โดยจำนวนของอ็อบเจกต์สามารถเพิ่มขึ้นได้ ขึ้นอยู่กับปัจจัยด้านข้อมูล หรือการสื่อสารระหว่างกระบวนการ ดังนั้นจะไม่จบการทำงานจนอ็อบเจกต์สุดท้ายทำงานเสร็จสิ้น โดยแต่ละอ็อบเจกต์สามารถเกิดการทำงานควบคู่ได้อิสระ และไม่เกิดการทำงานพร้อมกัน [9] ในภาพที่ 3.4 แสดงลักษณะแบบรูป MI without a Priori Run-Time Knowledge ในภาษายอวล์

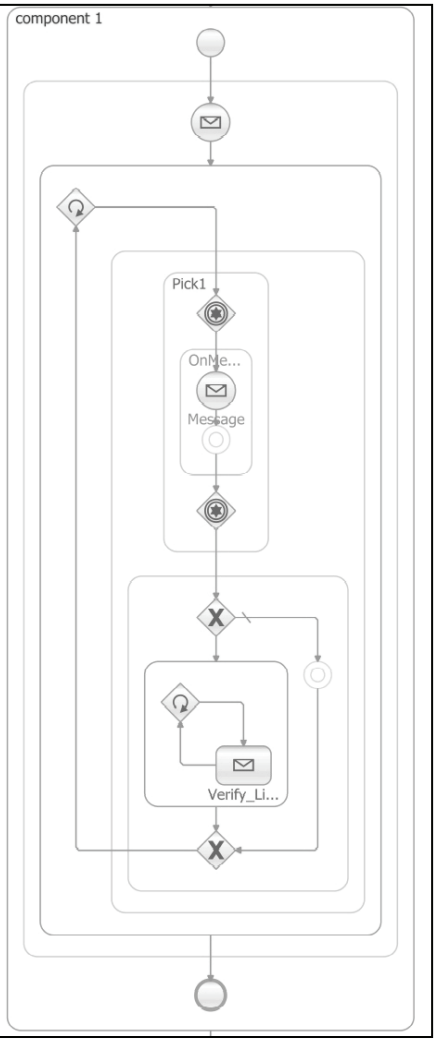


ภาพที่ 3.4 แบบรูป Multiple instances without a priori run-time knowledge [9]

เครื่องมือสามารถเปลี่ยนแบบรูปนี้เป็นโครงร่างบีเพลได้ตามภาพที่ 3.5 โดยเครื่องมือสร้างตัวแปร i ชนิด boolean ตัวแปร counter ชนิด integer มีโครงสร้างรับจำนวนการวนซ้ำ <receive

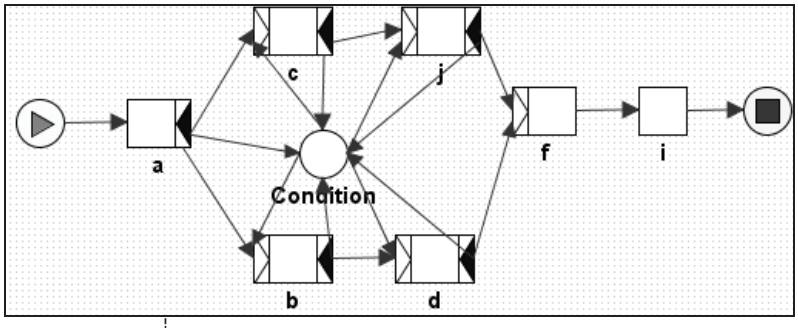
variable="quantity"/> ที่ตัวแปร quantity ใช้กำหนดจำนวนการทำงานซ้ำที่เซอร์วิส Verify_List1 ผ่านโครงสร้าง while โดยภายในมีโครงสร้าง <onMessage variable ="input"/> เพื่อเพิ่มจำนวนการวนซ้ำของเซอร์วิส Verfy_List1 และเมื่อทำงานครบตามเงื่อนไขที่ระบุใน while จะจบการทำงาน โดยด้านขวาของรูปแสดงโครงร่างบีเพลที่เครื่องมือสร้างขึ้น และด้านซ้ายแสดงชุดคำสั่งบีเพลที่เครื่องมือสร้างขึ้น

```
<scope>
<variable name="i" type="boolean" />
<variable name="counter" type="int" />
<sequence>
<receive  variable="quantity" />
$i=true
<while>
<condition> $i=true </condition>
      $counter=0
      <sequence>
<pick>
<onMessage variable="input"/>
</pick>
      <if>
<condition>input==true</condition>
            <while>
                  <condition> $counter<=quantity </condition>
<invoke name="Verify_List1_6"/>
                  $counter++
            </while>
      <else>
            $i=false
      </else>
      </if>
</sequence>
</while>
</sequence>
</scope>
```



ภาพที่ 3.5 โครงร่างบีเพลในแบบรูป MI without a Priori Run-Time Knowledge

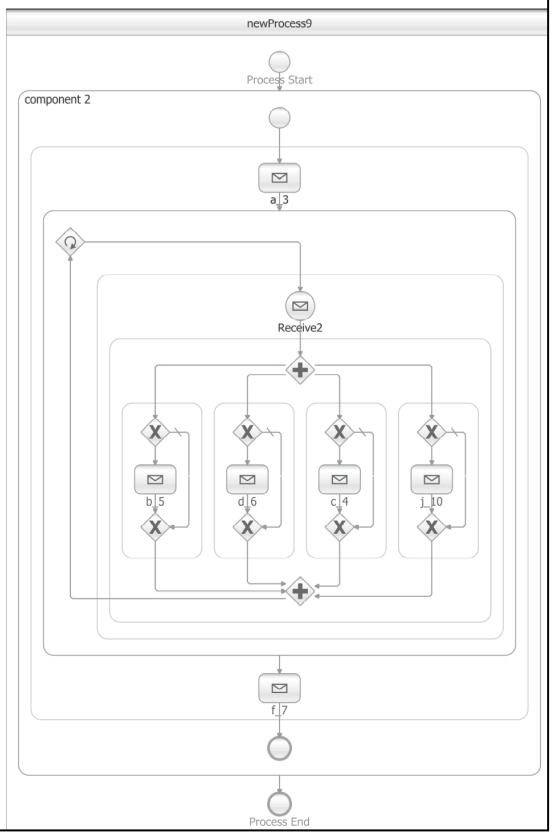
-แบบรูป Interleaved Parallel Routing คือแบบรูปของกลุ่มกิจกรรม ที่กลุ่มกิจกรรมไม่สามารถดำเนินการได้พร้อมกันและต้องเกิดการทำงานในกลุ่มกิจกรรมทั้งหมดจึงสามารถจบการทำงานของแบบรูป [9] ในภาพที่ 3.6 แสดงลักษณะแบบรูป Interleaved Parallel Routing ในภาษายอวล์



ภาพที่ 3.6 แบบรูป Interleaved Parallel Routing [9]

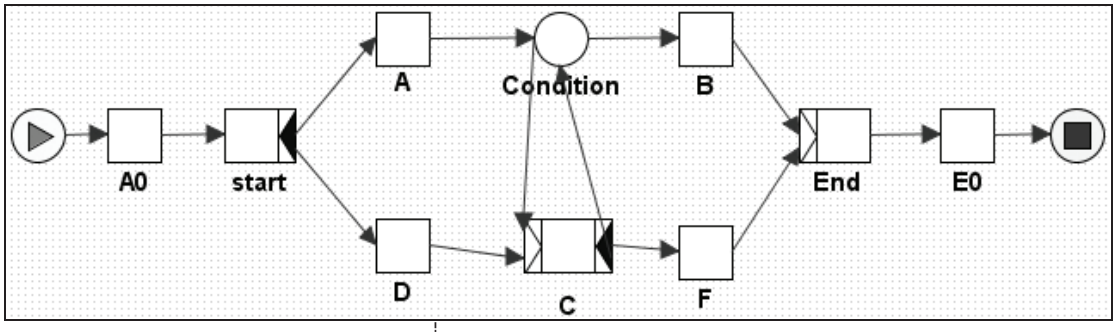
เครื่องมือสามารถเปลี่ยนแบบรูปนี้เป็นโครงร่างบีเพลได้ตามภาพที่ 3.7 โดยเครื่องมือสร้างตัวแปร valb,vald,valc,valj ขนิด boolean กำหนดให้ค่าตัวแปรเป็น true หลังจากนั้น มีโครงสร้าง while ทำการวนซ้ำภายในมีโครงสร้าง <receive variable="input">เพื่อรับลำดับการทำงานในการเรียกเซอร์วิส B,D,C,J ที่ขณะที่ทำงาน กรณีถ้าเซอร์วิสทำงานแล้ว ตัวแปร val ของเซอร์วิสนั้นจะถูกกำหนดให้เป็น false เพื่อไม่ให้เกิดการทำงานเซอร์วิสนั้นซ้ำ ระบบจะทำงานวนซ้ำจนเซอร์วิสทั้ง 4 ได้ทำงานทั้งหมด และจบการทำงาน โดยด้านขวาของรูปแสดงโครงร่างบีเพลที่เครื่องมือสร้างขึ้น และด้านซ้ายแสดงชุดคำสั่งบีเพลที่เครื่องมือสร้างขึ้น

```
<scope>
<sequence>
<variables>
<variable name= "$valb_5" type="xs:boolean"/>
<variable name= "$vald_6" type="xs:boolean"/>
<variable name= "$valc_4" type="xs:boolean"/>
<variable name= "$valj_10" type="xs:boolean"/>
</variables>
$valb_5=true
$vald_6=true
$valc_4=true
$valj_10=true
<invoke name = "a_3"/>
<while>
<condition>$count<4</condition>
 <sequence>
<receive name="Receive2" variable="input"/>
<flow>
<if>
<condition>$input=="b_5" && $valb_5==true </condition>
<invoke name="b_5"/>
$valb_5=false
$count++
</if>
<if>
<condition>$input=="d_6" && $vald_6==true && $valb_5==false</condition>
<invoke name="d_6"/>
$vald_6=false
$count++
</if>
<if>
<condition>$input=="j_10" && $valj_10==true && $valc_4==false</condition>
<invoke name="j_10"/>
$valj_10=false
$count++
</if>
</flow>
</sequence>
</while>
<invoke name="f_7"/>
</sequence>
</scope>
```



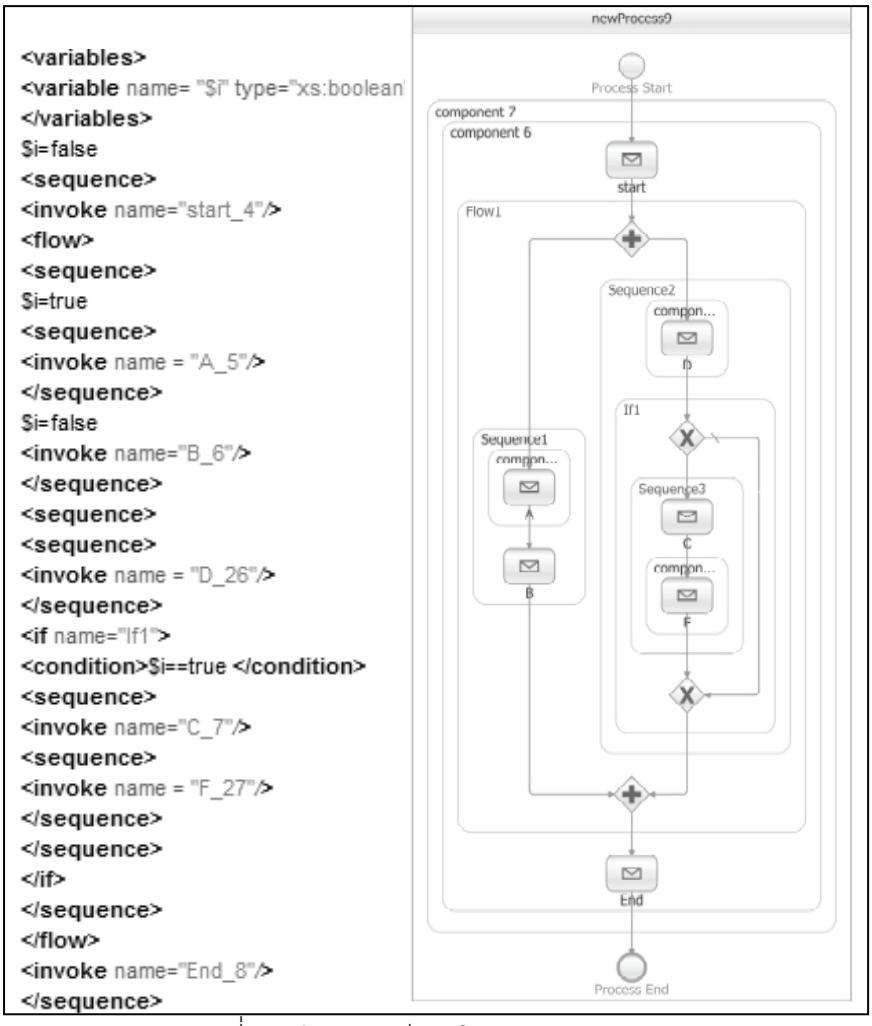
ภาพที่ 3.7 โครงร่างบีเพลในแบบรูป Interleaved Parallel Routing

-แบบรูป Milestone แบบรูปนี้มีลักษณะเป็นกิจกรรมที่สามารถเกิดการทำงานขึ้นเมื่อกระแสงาน สามารถกำหนดสถานะที่มีความเฉพาะเจาะจง โดยมีตัวดำเนินการชี้เฉพาะ เมื่อกระแสงานดำเนินการถึงตัวดำเนินการชี้เฉพาะ สามารถเลือกเปิดใช้งานสถานะได้ กรณีกระแสงานทำงานเกินกว่าสถานะที่กำหนดนี้แล้วกิจกรรมไม่สามารถเกิดการทำงานได้ [9] ในภาพที่ 3.8 แสดงลักษณะแบบรูป Milestone ในภาษายอวล์



ภาพที่ 3.8 แบบรูป Milestone [9]

ในภาพที่ 3.9 แสดงลักษณะแบบรูป Milestone โดยเครื่องมือสร้างตัวแปร j ชนิด boolean กำหนดให้ตัวแปรเป็น false หลังจากนั้นเกิดโครงสร้าง <flow> มีการทำงานพร้อมกัน 2 sequence โดยมีโครงสร้าง if ตรวจสอบสถานะ การทำงานของกระแสงาน ถ้าสถานะเซอร์วิส B ทำงานเสร็จก่อนเซอร์วิส F จะไม่เกิดทำงานขึ้น และจบการทำงาน โดยด้านขวาของรูปแสดงโครงร่างบีเพลที่เครื่องมือสร้างขึ้น และด้านซ้ายแสดงชุดคำสั่งบีเพลที่เครื่องมือสร้างขึ้น

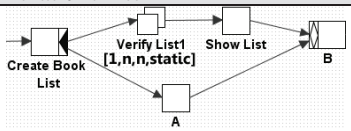


ภาพที่ 3.9 โครงร่างบีเพลในแบบรูป Milestone

ในส่วนต่อไปจะพูดถึงแบบรูป MI without Synchronization แบบรูป Deferred Choice และแบบรูป Cancel Activity ทั้ง 3 แบบรูปนี้เป็นแบบรูปที่ไม่สามารถเปลี่ยนเป็นโครงร่างบีเพลได้อย่างถูกต้องตามแนวคิดใน [7] ดังนั้นผู้ทำวิทยานิพนธ์จึงนำเสนอวิธีการเปลี่ยนเป็นโครงร่างบีเพลที่มีความเหมาะสมแทน [7]

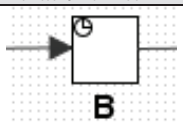
- แบบรูป MI without Synchronization เป็นแบบรูปนี้ เกิดการทำงานขึ้นภายในกิจกรรมที่สามารถเกิดการทำงานได้หลายอ็อบเจกต์ โดยแต่ละอ็อบเจกต์สามารถเกิดและทำงานควบคู่ได้อิสระและ ไม่เกิดทำงานพร้อมกันจนเสร็จสิ้น [9] โดยปรับปรุงโครงสร้างจากแบ่งแอ็คติวิตี <process> เป็น 2 <process> เพราะจากโครงสร้างดังกล่าวไม่ตรงโครงสร้างการทำงานใน [9] ผู้ทำวิทยานิพนธ์ปรับปรุงโครงสร้างโดยใช้ โครงสร้างแอ็คติวิตี <flow> เพื่อให้เกิดการทำงานควบคู่ได้อิสระ สามารถดูโครงร่างบีเพลที่ปรับปรุงได้ที่ตาราง 3.1 และสามารถตรวจสอบความถูกต้องการเปลี่ยนแบบรูปสามารถดูการทดสอบได้ที่ ภาคผนวก ค.

ตารางที่ 3.1 ปรับปรุงโครงร่างบีเพลแบบรูป MI without Synchronization

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
12	MI without Synchronization		<sequence> <invoke name = "Create_Book_List"/> <flow> <sequence > <invoke name = "A"/> </sequence> <sequence name= "component 2"> <forEach name = "component 1" counterName="i" parallel="yes"> <startCounterValue>1</startCounterValue> <finalCounterValue>10</finalCounterValue> <scope> <invoke name="Verify_List1"/> </scope> </forEach> <invoke name = "Show_List"/> </sequence> </flow> <invoke name = "B"/> </sequence>

- แบบรูป Deferred Choice คือ แบบรูปที่กิจกรรมจะเกิดการทำงานขึ้น โดยขึ้นกับปฏิสัมพันธ์กับสภาพแวดล้อมในการทำงาน เพื่อใช้ในการตัดสินใจดำเนินกิจกรรม [9] โดยปรับปรุงโครงสร้างจากแอ็คติวิตี <pick> แอ็คติวิตี <onAlarm> และแอ็คติวิตี <until> เพราะจากโครงสร้างดังกล่าวหมายความว่าเมื่อกระแสงานจะเกิดการทำงานตามเงื่อนไข แอ็คติวิตี <onAlarm> ซึ่งการออกแบบดังกล่าวไม่ตรงกับการทำงานในนิยาม [9] ผู้ทำวิทยานิพนธ์ปรับปรุงโครงสร้างโดยใช้ โครงสร้างแอ็คติวิตี <if> แอ็คติวิตี <condition> เงื่อนไขการทำงานจะเกิดขึ้นเมื่อเวลาการทำงานอยู่ในเงื่อนไขที่กำหนด สามารถดูโครงร่างบีเพลที่ปรับปรุงได้ที่ตาราง 3.2 และสามารถตรวจสอบความถูกต้องการเปลี่ยนแบบรูปสามารถดูการทดสอบได้ที่ ภาคผนวก ค.

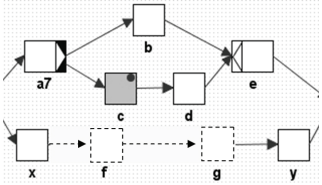
ตารางที่ 3.2 ปรับปรุงโครงร่างบีเพลแบบรูป Deferred Choice

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
16	Deferred Choice		..... <if> <condition>sxxf:dateTime-less-than(sxxf:current-dateTime(),'2011-10-02T14:58:51.99+07:00')</condition> <sequence> <invoke name="B"/> </sequence> </if> .....

- แบบรูป Cancel Activity คือ กิจกรรมที่เกิดการทำงานขึ้นเพื่อยกเลิกการทำงานของกิจกรรมก่อนหน้าที่จะเริ่มทำงาน และในกรณีที่กิจกรรมทำงานอยู่จะหยุดการทำงาน [9] โดยปรับปรุงโครงสร้างจากแอ็คติวิตี <faultHandlers> แอ็คติวิตี <catch> และ แอ็คติวิตี

<throw> เพราะจากโครงสร้างดังกล่าวหมายความว่าเมื่อกระแสงานเกิดการทำงานผิดพลาด (false) จะเกิดการทำงานที่อยู่ใน แอ็คติวีตี <catch> ขึ้น ซึ่งการออกแบบไม่ตรงกับการทำงานในนิยาม [9] ผู้ทำวิทยานิพนธ์ปรับปรุงโครงสร้างโดยใช้ โครงสร้างแอ็คติวิตี <if> และเพิ่มตัวแปร i และตัวแปร j สามารถดูโครงร่างบีเพลที่ปรับปรุงได้ที่ตาราง 3.3 เพื่อตรวจสอบความถูกต้องการเปลี่ยนแบบรูปสามารถดูการทดสอบได้ที่ ภาคผนวก ค.

ตารางที่ 3.3 ปรับปรุงโครงร่างบีเพลแบบรูป Cancel Activity

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
19	Cancel Activity		<variables> <variable name= "$i" type="xs:boolean"/> <variable name= "$j" type="xs:boolean"/> </variables> $i=true $j=true ..... <flow> <sequence> <invoke name="a"/> <if name> <invoke name="b"/> <else> <sequence> <invoke name="c"/> <if> <condition>$j==false</condition> $i=false </if> <invoke name="d"/> </sequence> </else> </if> <invoke name="e"/> </sequence> <sequence name="FlowSequence"> <invoke name="x"/> $j=false <if> <condition>$i==true</condition> <invoke name="f"/> </if> <if> <condition>$i==true</condition> <invoke name="g"/> </if> <if> <condition>$i==true</condition> <invoke name="y"/> </if> </sequence> </flow> .....

3.1.2 **แบบรูปโครงสร้างที่ไม่ดี** หมายถึงกระแสงานยอวล์อื่นๆที่ไม่มีโครงสร้างเชิงบล็อก กล่าวคือไม่มีจุดทางเข้า (Entry Point)และจุดทางออก (Exit point) ที่แน่นอน ทำให้ไม่สามารถใช้วิธีที่นำเสนอมาแปลงเป็นโครงร่างบีเพลได้ วิทยานิพนธ์นี้เลือกทำ Preprocessing เพื่อปรับ

โครงสร้างที่ไม่ดี นี้ให้กลับมาอยู่ในแบบรูปที่ดีก่อน จากนั้นผลลัพธ์ที่ได้จะสามารถใช้กระบวนการที่นำเสนอมาแปลงได้

การทำ Preprocessing กระทำโดยการปรับโครงสร้างของกระแสงานยอวล์ใหม่ โดยใช้แนวคิดอัลกอริทึมแฮมมอค ในรูป 3.11 แสดงกระแสงานยอวล์ที่มีโครงสร้างไม่ดี สามารถปรับแบบรูปให้อยู่ในแบบรูปโครงสร้างที่ดีโดยใช้อัลกอริทึมจากงานวิจัย [5]

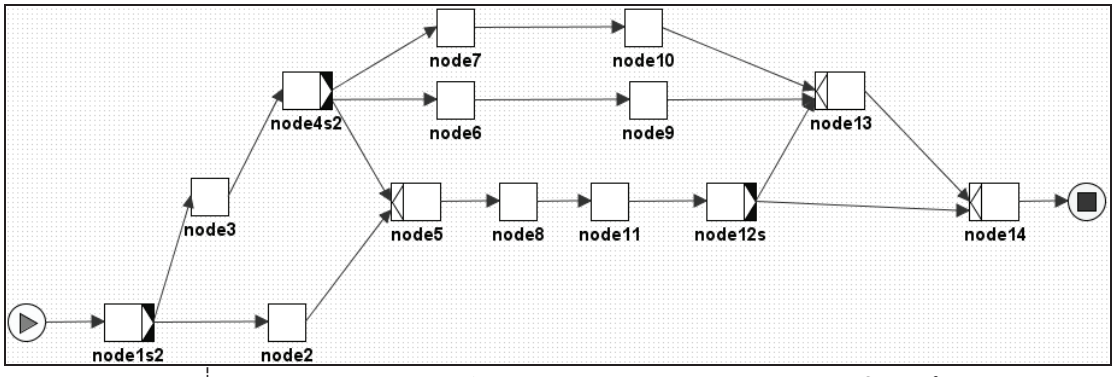
Algorithm 4 (elimination of forward branches)
input: CFG – a flowgraph with only forward branches;
output: CFG' – a flowgraph without branches.

```
Forward_copy(CFG) {
  { while (there exists an initial forward branch i_f)
    { Find the minimal hammock graph of i_f, MHG_if,
      with end node i_e;
      if (CG_if interacts with other forward branches)
      { /* a forward-copy transformation is applied */
        true_part = the shared statements =
          (MHG_if − CG_if − {i_e}) ∪ J;
        false_part = MHG_if − {i_f, i_e};
        MHG_if is replaced by:
          1. if (B) then true_part else false_part endif;
      } /* B is the Boolean expression of i_f.*/
      else /* a single forward branch */
        {true_part = MHG_if − {i_f, i_e};
          MHG_if is replaced by:
          1. if (¬B) then true_part endif; {i_e};
        }
      Eliminate_forward(true_part);
      Eliminate_forward(false_part);
    }
  }
```

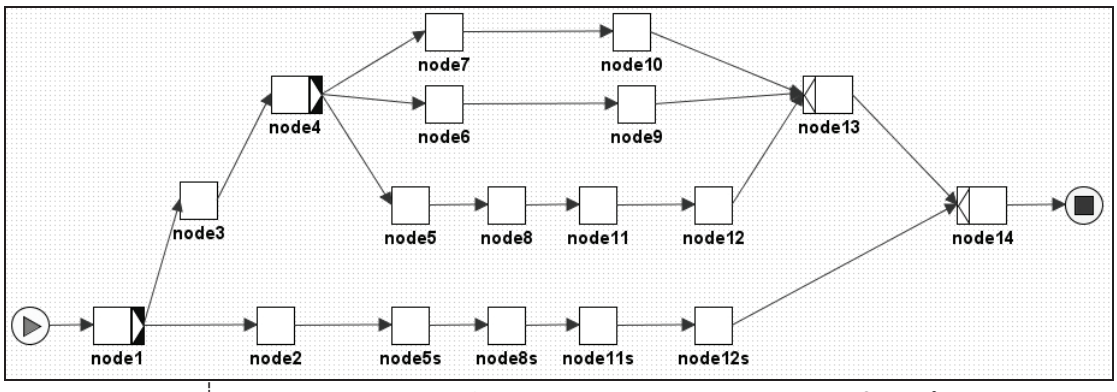
ภาพที่ 3.10 อัลกอริทึมในการปรับโครงสร้างแบบรูป Arbitrary Cycle [5]

โดยอัลกอริทึมนี้ทำค้นหาส่วนที่เกี่ยวข้องคือ CG_{if} เพื่อกำหนดจุดเริ่มต้นของโหนดการทำงานร่วมกัน (shared statement) กำหนดโหนด i_e เพื่อหาจุดสิ้นสุดของโหนดการทำงานร่วมกัน และกำหนด $true_part$, $false_part$ ของกระแสงานควบคุม CG_{if} เพื่อกำหนดกระแสงานควบคุมซ้ำ (Duplicated code) แล้วทำการปรับปรุงโครงสร้างของแบบรูป สามารถดูตัวอย่างการทำงานตามภาพที่ 3.11 จากในรูปโครงสร้างกระแสงานไม่สามารถเปลี่ยนเป็นโครงร่างบีเพลได้ ดังนั้นอัลกอริทึมทำการกำหนด CG_{if} คือโหนด node1s2 แล้วกำหนดโหนด i_e คือโหนด node12s และกำหนด $true_part$, $false_part$ โดยกำหนดจากเพรดิเคต (predicate) ของโหนด node1s2 และโหนด node12s เพื่อใช้สร้างกระแสงานควบคุมซ้ำ หลังจากนั้นระบบทำการปรับ

โครงสร้างกระแสงานใหม่จากภาพที่ 3.12 จะพบโครงสร้างกระแสงานใหม่มีโหนด node5s, โหนด node8s,โหนด node11s และโหนด node12s เพิ่มขึ้นมาโดยจากโครงสร้างใหม่ระบบสามารถเปลี่ยนเป็นโครงร่างบีเพลได้และทำงานได้อย่างถูกต้อง ดูตัวอย่างการทดสอบกระแสงานที่มีแบบรูป Arbitrary Cycle ได้ที่ภาคผนวก ค.



ภาพที่ 3.11 กระแสงานยอวล์แบบรูป Arbitrary Cycle ก่อนปรับโครงสร้าง



ภาพที่ 3.12 กระแสงานยอวล์แบบรูป Arbitrary Cycle หลังปรับโครงสร้าง

3.2 **อัลกอริทึมการทำงาน** Fold function

ในส่วนนี้จะอธิบายการทำงานอัลกอริทึม Fold function เป็นอัลกอริทึมใช้ในการลดโครงสร้างของกระแสงานยอวล์เพื่อเปลี่ยนเป็นโครงร่างบีเพล ด้วยทำการเปรียบเทียบโครงสร้างกระแสงานกับแบบรูปที่กำหนดและทำการวนซ้ำจนได้โครงสร้างสุดท้ายที่ต้องการ อัลกอริทึมนี้ประกอบด้วย 2 ส่วนการทำงาน คือการกำหนดแบบรูปที่ใช้วนซ้ำ และกำหนดการทำงานในการวนซ้ำ [6] รายละเอียดของอัลกอริทึมสามารถดูได้จากภาพที่ 3.13 โดยอัลกอริทึมจะทำการเปรียบเทียบกราฟสายงานควบคุม กับแบบรูปโครงสร้างที่ดี ถ้าโครงสร้างกระแสงานตรงกับแบบรูปที่กำหนดจะทำการลดกราฟสายงานควบคุมเข้าด้วยกัน และทำการวนซ้ำจนได้ส่วนประกอบเชิงเดียวแสดงว่ากราฟสายงานควบคุมนั้นมีโครงสร้างที่ดี

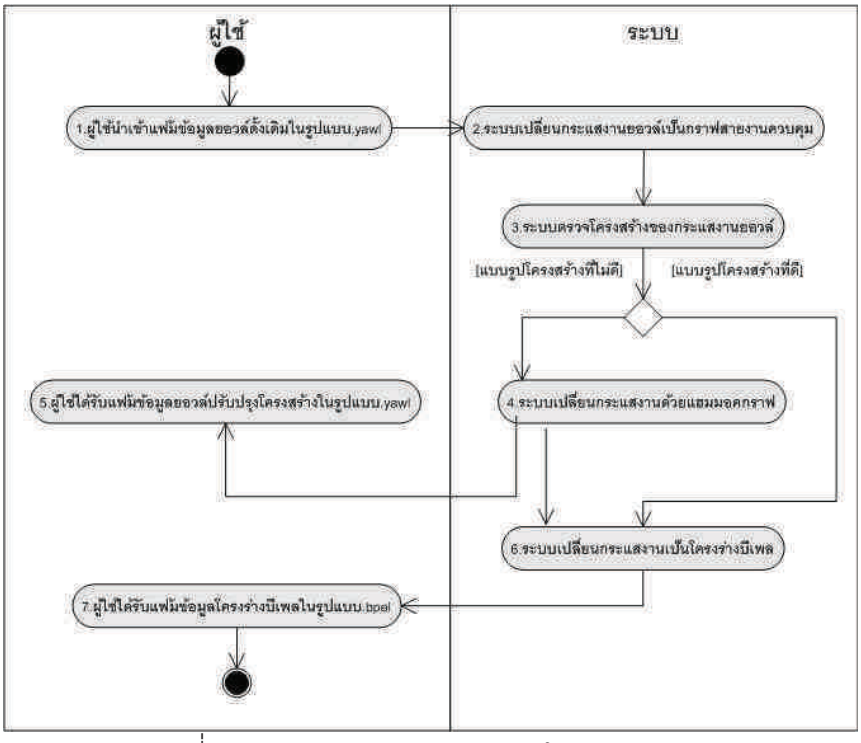
Fold function Algorithm
Input: Control flow graph(CFG) c
Output: BPEL skeleton b

Procedure fold function()
while(c is not Single component)
do
for each pi is well pattern table p
reduce all of the pi found in c
for each qi in well pattern q
reduce all of the qi found in c
end
p is well pattern
q is non well pattern

ภาพที่ 3.13 อัลกอริทึม Fold function

3.3 โครงสร้างการทำงานของเครื่องมือ

องค์ประกอบของกรอบการทำงานของเครื่องมือ คือระบบจะทำการตรวจสอบโครงสร้างในกระแสงานยอวล์ว่าเป็นแบบรูปโครงสร้างที่ดี ด้วยตรวจสอบจากแบบรูปในตารางที่ 2.2 ถ้าตรงกระแสงานยอวล์นั้นมีแบบรูปโครงสร้างที่ดี และ แบบรูปโครงสร้างที่ไม่ดี คือเป็นแบบรูปนั้นไม่สามารถเปลี่ยนเป็นโครงร่างบีเพลได้ โดยงานวิจัยนี้ใช้อัลกอริทึมแฮมมอคกราฟในการเปลี่ยนแบบรูปโครงสร้างที่ไม่ดีเป็นแบบรูปโครงสร้างที่ดีและทำการเปลี่ยนเป็นโครงร่างบีเพลโดยมีขั้นตอนดัง ภาพที่ 3.14



ภาพที่ 3.14 แผนภาพกิจกรรมการทำงานของระบบ

วิทยานิพนธ์ได้พัฒนาเครื่องมือในการเปลี่ยนภาษายอวล์เป็นโครงร่างบีเพลโดยมีลำดับของกิจกรรมการทำงานของระบบดังภาพที่ 3.14 มีรายละเอียดดังนี้

1) ผู้ใช้นำเข้าแฟ้มข้อมูลยอวล์ดั้งเดิมในรูปแบบ.yawl ผู้ใช้ทำการนำข้อมูลกระแสงานยอวล์ดั้งเดิมเข้าไปในระบบโดยเป็นแฟ้มข้อมูล.yawl

2) ระบบเปลี่ยนกระแสงานยอวล์เป็นกราฟสายงานควบคุม (Control flow graph) จากแผนภาพกระแสงานยอวล์ให้มีลักษณะเป็นกราฟสายงานควบคุมเพื่อใช้ในการตรวจโครงสร้าง

3) ระบบตรวจโครงสร้างที่ดีของกระแสงานยอวล์จากกราฟสายงานควบคุม ระบบใช้อัลกอริทึม Fold function โดยสามารถแยกแบบรูปโครงสร้างที่ดีกับแบบรูปโครงสร้างไม่ดีได้โดยเปรียบเทียบกราฟสายงานควบคุม กับแบบรูปโครงสร้างที่ดีและเปลี่ยนให้เป็นโครงร่างภาษาบีเพล ถ้าตรงกับที่กำหนดจะลดกราฟสายงานควบคุมเข้าด้วยกันและวนซ้ำจนเหลือส่วนประกอบเชิงเดียว (Single component)คือไม่สามารถลดรูปกราฟสายงานควบคุมได้อีกและเหลือโหนดเดียว

4) ระบบเปลี่ยนกระแสงานแบบรูปโครงสร้างที่ไม่ดี ด้วยอัลกอริทึมแฮมมอคกราฟ ถ้าเป็นแบบรูปโครงสร้างที่ไม่ดีจะทำการเปลี่ยนกราฟสายงานควบคุมนั้นใหม่ด้วยอัลกอริทึมแฮมมอคกราฟ ให้เป็นแบบรูปโครงสร้างที่ดี และสร้างกราฟสายงานควบคุม นั้นกลับเป็นกระแสงานยอวล์ใหม่ที่ปรับปรุงโครงสร้างแล้ว

5) ผู้ใช้ได้รับแฟ้มข้อมูลส่งออกยอวล์ที่ปรับปรุงโครงสร้างแล้วในรูปแบบไฟล์.yawl หลังจากระบบปรับกระแสงานยอวล์เป็นกระแสงานที่มีโครงสร้างที่ดี ระบบจะส่งไฟล์ยอวล์ที่ปรับปรุงโครงสร้างแล้วให้ผู้ใช้

6) ระบบเปลี่ยนกระแสงานเป็นโครงร่างบีเพล หลังจากได้กระแสงานยอวล์เป็นแบบรูปโครงสร้างที่ดีแล้วระบบจะสร้างโครงร่างบีเพลจากส่วนประกอบเชิงเดียวที่ได้จากอัลกอริทึม Fold function

7) ผู้ใช้ได้รับแฟ้มข้อมูลส่งออกบีเพลในรูปแบบไฟล์.bpel หลังจากระบบเปลี่ยนกระแสงานเป็นโครงร่างบีเพล ผู้ใช้จะได้ไฟล์โครงร่างบีเพลและส่งแฟ้มให้ผู้ใช้

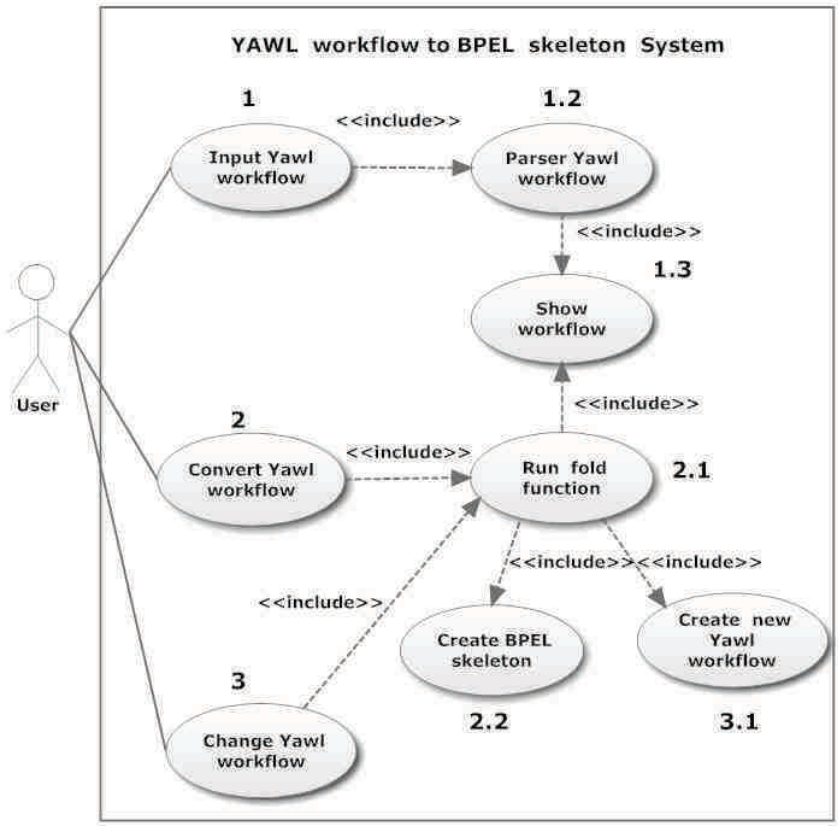
3.4 การวิเคราะห์และออกแบบเครื่องมือ

สำหรับการวิเคราะห์และออกแบบเครื่องมือนั้นอธิบายด้วยแผนภาพในภาษายูเอ็มแอล (UML:Unified Modeling Language) ซึ่งเป็นสัญลักษณ์มาตรฐานที่ใช้ในการวิเคราะห์และออกแบบเชิงวัตถุ (Object Oriented Analysis and Design) โดยแผนภาพที่เลือกใช้ได้แก่ แผนภาพยูสเคส (Use case diagram) แผนภาพคลาส (Class diagram) และแผนภาพลำดับ (Sequence diagram)

3.2.1 แผนภาพยูสเคส

แผนภาพยูสเคสใช้อธิบายขอบเขตของระบบและฟังก์ชั่นการทำงานพื้นฐานของระบบในมุมมองผู้ใช้ แผนภาพยูสเคสของเครื่องมือการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพลแสดงได้ดังภาพที่ 3.15

จากแผนภาพยูสเคสภาพที่ 3.15 เริ่มต้นโดยผู้ใช้นำเข้ากระแสงานยอวล์เข้าในระบบ (Input Yawl workflow) ซึ่งอยู่ในรูปไฟล์.yawl หลังจากนั้นระบบจะทำการอ่านและตรวจสอบกระแสงานยอวล์ (Parser Yawl workflow) แล้วระบบจะแสดงกระแสงานยอวล์ในรูปกราฟให้ผู้ใช้ (Show workflow) หลังจากนั้นผู้ใช้เลือกการแปลงกระแสงานยอวล์เป็นโครงร่างบีเพล (Convert Yawl workflow) ระบบจะทำการแปลงกระแสงานยอวล์เป็นโครงร่างบีเพล (Run fold function) หลังจากการแปลงเสร็จสิ้นระบบจะสร้างไฟล์โครงร่างบีเพลให้ผู้ใช้ (Create BPEL Skeleton) ส่วนกรณีที่กระแสงานยอวล์ที่มีลักษณะเป็นแบบรูปโครงสร้างไม่ดีระบบจะทำการปรับโครงสร้างกระแสงานยอวล์นั้นใหม่ (Change Yawl workflow) และระบบสร้างไฟล์กระแสงานยอวล์ที่ปรับโครงสร้างให้ผู้ใช้ (Create new Yawl workflow) ส่วนรายละเอียดของแต่ละยูสเคสแสดงในตารางที่ 3.4 - 3.11



ภาพที่ 3.15 แผนภาพยูสเคสของเครื่องมือ

ตารางที่ 3.4 รายละเอียดยูสเคสนำเข้ากระแสงานยอวล์

ยูสเคส	1 : Input Yawl workflow
แอคเตอร์	ผู้ใช้ (User)
เป้าหมาย	นำเข้ากระแสงานยอวล์เข้าสู่ระบบ
ยูสเคสที่สัมพันธ์	Include:Parser Yawl workflow
เงื่อนไขก่อนหน้า	กระแสงานยอวล์อยู่ในรูปของไฟล์.yawl
ขั้นตอน	1.ผู้ใช้งานเรียกใช้โปรแกรม 2.ระบบสร้างหน้าต่างสำหรับนำเข้ากระแสงานยอวล์ 3.ผู้ใช้เลือกไฟล์กระแสงานยอวล์
เงื่อนไขภายหลัง	

ตารางที่ 3.5 รายละเอียดยูสเคสอ่านและตรวจสอบกระแสงานยอวล์

ยูสเคส	1.2 : Parser Yawl workflow
แอคเตอร์	
เป้าหมาย	อ่านกระแสงานยอวล์และสร้างกระแสงานควบคุม
ยูสเคสที่สัมพันธ์	Include: Show workflow
เงื่อนไขก่อนหน้า	กระแสงานยอวล์อยู่ในรูปของไฟล์.yawl
ขั้นตอน	อ่านกระแสงานยอวล์เข้าในระบบและสร้างอ็อบเจกต์ yawlvertex
เงื่อนไขภายหลัง	

ตารางที่ 3.6 รายละเอียดยูสเคสแสดงกระแสงานในระบบ

ยูสเคส	1.3 : Show workflow
แอคเตอร์	
เป้าหมาย	อ่านกระแสงานยอวล์และสร้างกระแสงานควบคุม
ยูสเคสที่สัมพันธ์	
เงื่อนไขก่อนหน้า	กระแสงานอยู่ในรูปอ็อบเจกต์ yawlvertex
ขั้นตอน	ระบบแสดงหน้าต่างกระแสงานในระบบให้ผู้ใช้ด้วยระบบเรียกใช้ไลเบอรี่ jgraph แสดง
เงื่อนไขภายหลัง	

ตารางที่ 3.7 รายละเอียดยูสเคสกระทำการเปลี่ยนกระแสงานเป็นบีเพล

ยูสเคส	2 : Convert Yawl workflow
แอคเตอร์	ผู้ใช้ (User)
เป้าหมาย	เพื่อสั่งการทำงานการเปลี่ยนกระแสงาน
ยูสเคสที่สัมพันธ์	Include: Run fold function
เงื่อนไขก่อนหน้า	ผู้ใช้นำเข้ากระแสงานยอวล์สู่ระบบ
ขั้นตอน	1. ผู้ใช้กดปุ่มแปลงกระแสงานยอวล์เป็นบีเพล 2. ระบบเรียกใช้ยูสเคส Run fold function
เงื่อนไขภายหลัง	

ตารางที่ 3.8 รายละเอียดยูสเคสแปลงกระแสงานยอวล์เป็นบีเพล

ยูสเคส	2.1 : Run fold function
แอคเตอร์	
เป้าหมาย	เพื่อลดรูปกระแสงานยอวล์เป็นบีเพล
ยูสเคสที่สัมพันธ์	Include: Create BPEL skeleton
เงื่อนไขก่อนหน้า	โปรแกรมได้รับนำเข้ากระแสงานยอวล์เข้าระบบ
ขั้นตอน	1. ระบบสร้างหน้าต่างแสดงกราฟสายงานควบคุมที่นำเข้า 2. ระบบตรวจสอบอ็อบเจกต์ yawlvertex และเปรียบเทียบกราฟสายงานควบคุม กับส่วนประกอบโครงสร้างที่ดีที่กำหนด 3. ลดกราฟสายงานควบคุม เข้าด้วยกันและระบบจะแสดงกราฟสายงานควบคุมที่ผ่านลดรูป 4. และวนซ้ำจนได้ส่วนประกอบเชิงเดียว
เงื่อนไขภายหลัง	

ตารางที่ 3.9 รายละเอียดยูสเคสสร้างบีเพล

ยูสเคส	2.2 : Create BPEL skeleton
แอคเตอร์	
เป้าหมาย	เพื่อสร้างไฟล์ข้อมูลบีเพล
ยูสเคสที่สัมพันธ์	
เงื่อนไขก่อนหน้า	โปรแกรมลดกราฟสายงานควบคุมได้ส่วนประกอบเชิงเดียว
ขั้นตอน	ระบบสร้างไฟล์ข้อมูลบีเพล
เงื่อนไขภายหลัง	โปรแกรมได้รับไฟล์ข้อมูลบีเพล

ตารางที่ 3.10 รายละเอียดยูสเคสปรับโครงสร้างกระแสงานยอวล์

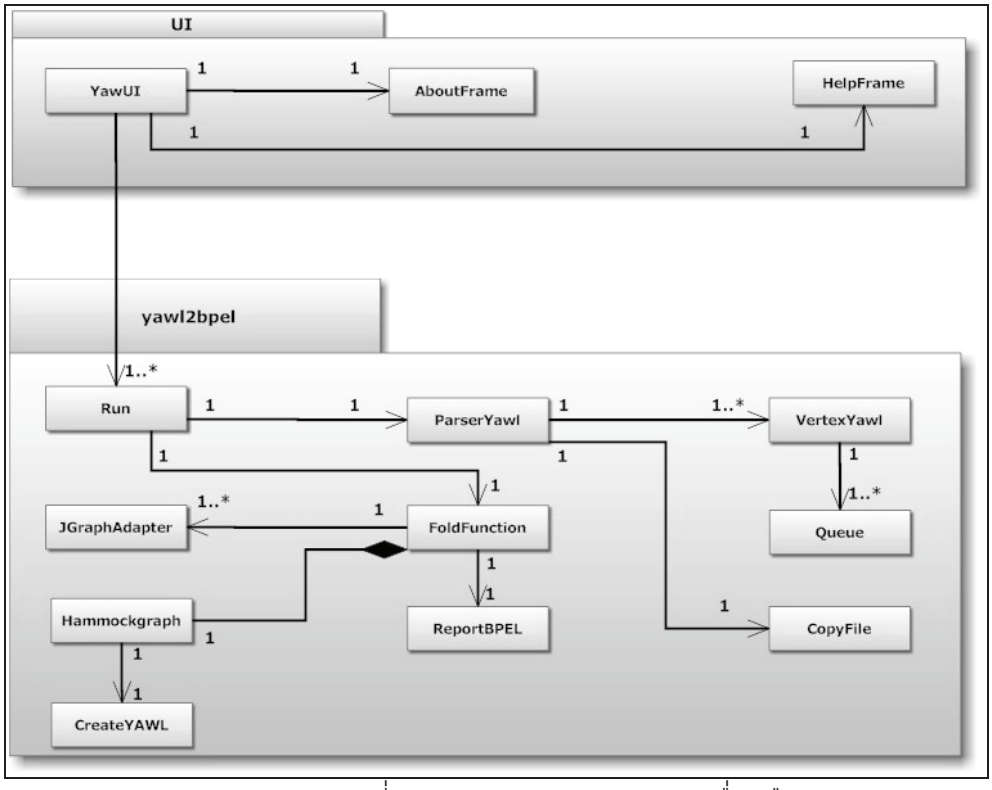
ยูสเคส	3 : Change Yawl workflow
แอคเตอร์	ผู้ใช้ (User)
เป้าหมาย	เพื่อปรับโครงสร้างกระแสงานยอวล์ใหม่
ยูสเคสที่สัมพันธ์	Include: Run fold function
เงื่อนไขก่อนหน้า	ยูสเคส Run fold function ไม่สามารถลดกราฟสายงานควบคุมจนได้ส่วนประกอบเชิงเดียว
ขั้นตอน	1. ผู้ใช้กดปุ่มปรับโครงสร้างกระแสงานยอวล์ 2. ระบบทำการปรับโครงสร้างกระแสงานยอวล์ใหม่ 3. ระบบเรียกใช้ยูสเคส Run fold function
เงื่อนไขภายหลัง	

ตารางที่ 3.11 รายละเอียดยูสเคสสร้างไฟล์ยอวล์ใหม่

ยูสเคส	3.1 : Create new Yawl workflow
แอคเตอร์	
เป้าหมาย	เพื่อสร้างไฟล์ข้อมูลยอวล์ใหม่
ยูสเคสที่สัมพันธ์	
เงื่อนไขก่อนหน้า	ยูสเคส Change Yawl workflow ถูกเรียกใช้งาน
ขั้นตอน	ระบบสร้างไฟล์ข้อมูลยอวล์
เงื่อนไขภายหลัง	โปรแกรมได้รับไฟล์ข้อมูลยอวล์ใหม่ที่ปรับโครงสร้างแล้ว

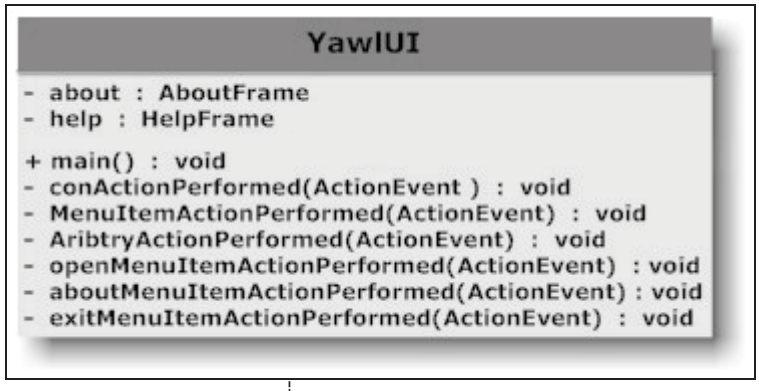
3.4.2 แผนภาพคลาส

แผนภาพคลาสใช้แสดงรายละเอียดคลาสและความสัมพันธ์ระหว่างคลาสต่างๆ เพื่อจำลองภาพการออกแบบส่วนที่เป็นโครงสร้างคงที่ของระบบ โดยในภาพที่ 3.16 เป็นแผนภาพคลาสของเครื่องมือสร้างการเปลี่ยนรูปกระแสงานยอวล์เป็นภาษาบีเพล ซึ่งแต่ละคลาสสามารถแบ่งได้ตามแพคเกจ (Package) และมีรายละเอียดดังต่อไปนี้



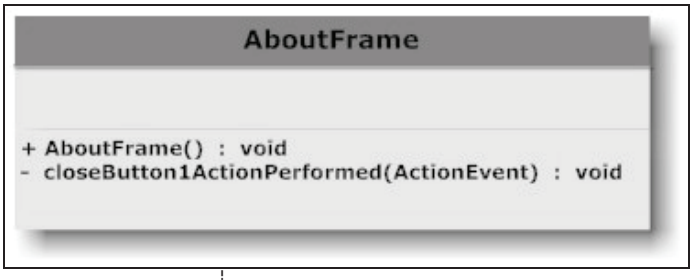
ภาพที่ 3.16 แผนภาพคลาสของเครื่องมือ

1) คลาส YawlUI คือ คลาสที่ทำหน้าที่ติดต่อกับผู้ใช้ในส่วนของการนำเข้ากระแสงานยอวล์ รายละเอียดของคลาสแสดงดังภาพที่ 3.17



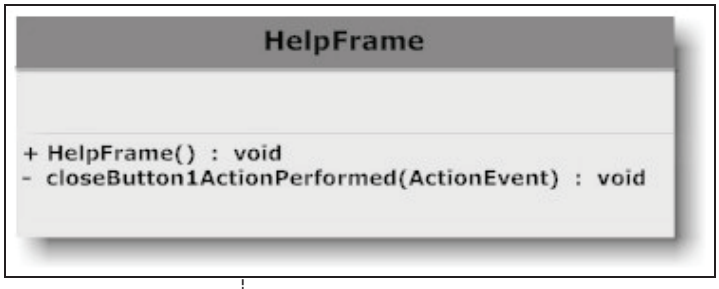
ภาพที่ 3.17 คลาส YawlUI

2) คลาส AboutFrame คลาสที่ทำหน้าที่ติดต่อกับผู้ใช้โดยแสดงรายละเอียดเกี่ยวกับระบบ รายละเอียดของคลาสแสดงดังภาพที่ 3.18



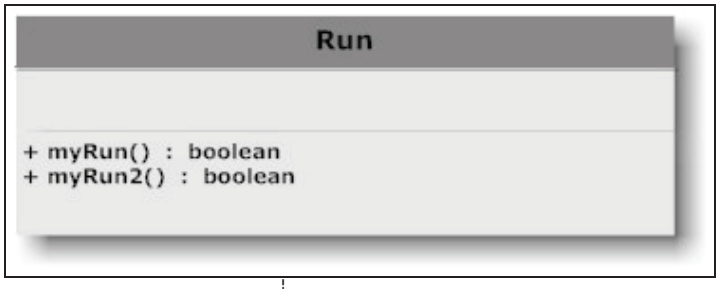
ภาพที่ 3.18 คลาส AboutFrame

3) คลาส HelpFrame คลาสที่ทำหน้าที่ติดต่อกับผู้ใช้โดยแสดงวิธีการใช้งานระบบให้ผู้ใช้ รายละเอียดของคลาสแสดงดังภาพที่ 3.19



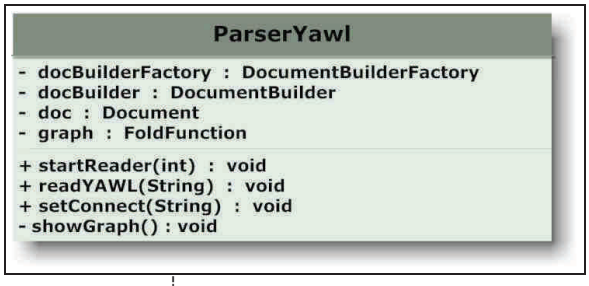
ภาพที่ 3.19 คลาส HelpFrame

4) คลาส Run คลาสที่ทำหน้าที่ดำเนินการเริ่มเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพล รายละเอียดของคลาสแสดงดังภาพที่ 3.20



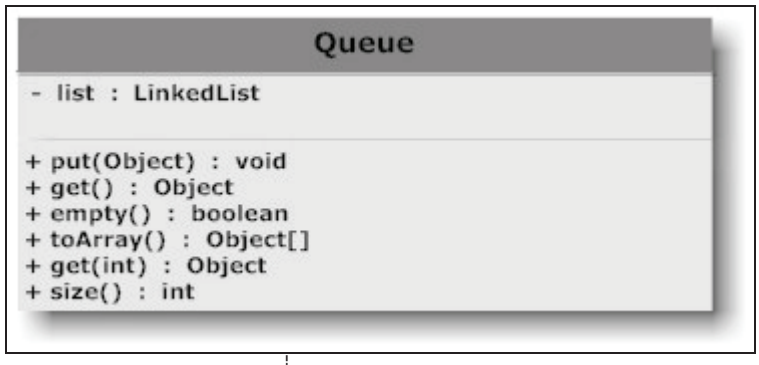
ภาพที่ 3.20 คลาส Run

5) คลาส ParserYawl คือ คลาสที่ทำหน้าดำเนินการอ่านไฟล์ข้อมูลนำเข้ากระแสงานยอวล์เข้าสู่ระบบ รายละเอียดของคลาสแสดงดังภาพที่ 3.21



ภาพที่ 3.21 คลาส ParserYawl

6) คลาส Queue คือ คลาสที่เก็บข้อมูลโครงสร้างข้อมูลแต่ละโหนดในยอวล์รูปของคิว รายละเอียดของคลาสแสดงดังภาพที่ 3.22

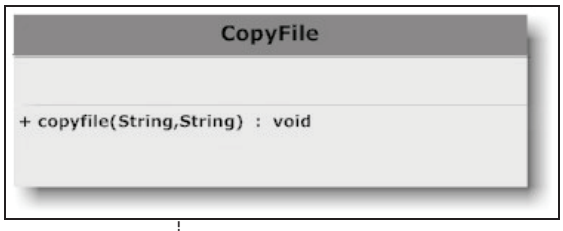


ภาพที่ 3.22 คลาส Queue

7) คลาส VertexYawl คือ คลาสที่ทำหน้าที่เก็บรายละเอียดข้อมูลแต่ละโหนดในกระแสงานยอวล์ประกอบไปด้วยข้อมูลลักษณะทั้งหมด เช่น โหนดถัดไป โหนดก่อนหน้า ชื่อของโหนด ตำแหน่งของโหนด รายละเอียดJoin-Split-Predicateของโหนด รายละเอียดออปเจ็กต์ของโหนดกรณีโหนดนั้นเป็น Multi Instance รายละเอียดTimerTrigger ของโหนด และรายละเอียด Cancel ของโหนด สามารถดูรายละเอียดของคลาสดังภาพที่ 3.23

ภาพที่ 3.23 คลาส VertexYawl

8) คลาส CopyFile คือคลาสที่ทำหน้าที่ในการเปลี่ยนข้อมูลนำเข้าไฟล์ yawl เป็นไฟล์xml รายละเอียดของคลาสแสดงดังภาพที่ 3.24

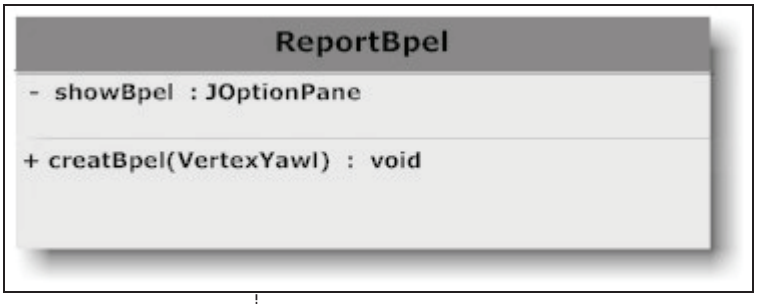


ภาพที่ 3.24 คลาส CopyFile

9) คลาส FoldFunction คือคลาสที่ทำหน้าที่เก็บโครงสร้างของกระแสงานยอวล์ และดำเนินการเปลี่ยนกระแสงานยอวล์เป็นส่วนประกอบเชิงเดียว โดยมีเมท็อดในการตรวจโครงสร้างของกระแสงานควบคุมกับแบบรูปในตารางที่ 2.2 และปรับโครงสร้างของกระแสงานควบคุมใหม่ รายละเอียดของคลาสแสดงดังภาพที่ 3.25

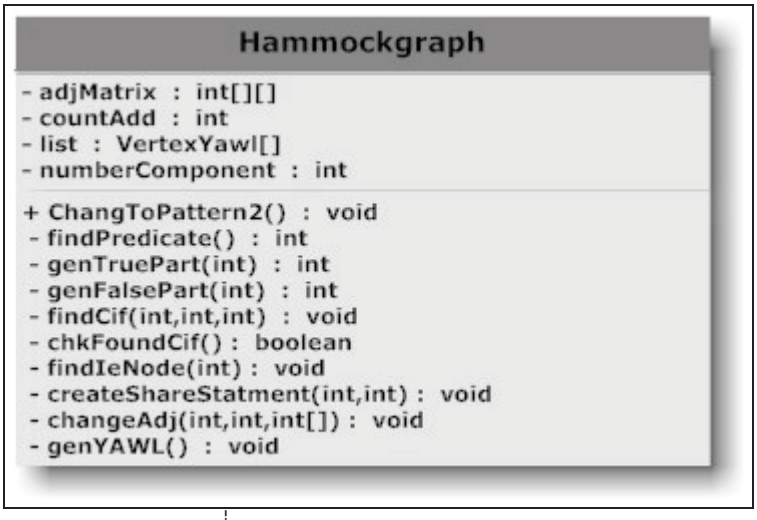
ภาพที่ 3.25 คลาส FoldFunction

10) คลาส ReportBpel คือคลาสที่ทำหน้าที่สร้างไฟล์บีเพลจากการเปลี่ยนรูปกระแสงานยอวล์รายละเอียดของคลาสแสดงดังภาพที่ 3.26



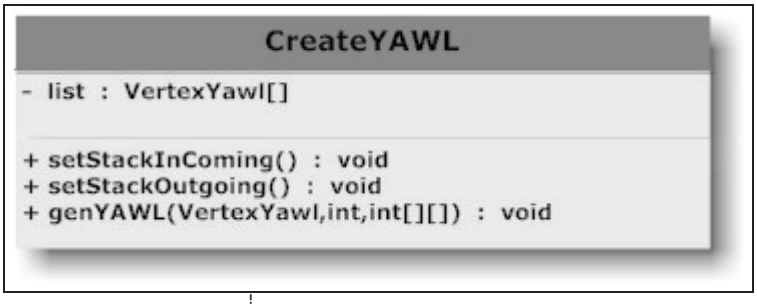
ภาพที่ 3.26 คลาส ReportBpel

11) คลาส Hammockgraph เป็นคลาสที่ทำหน้าที่ดำเนินการปรับโครงสร้างกระแสงานยอวล์ใหม่รายละเอียดของคลาสแสดงดังภาพที่ 3.27



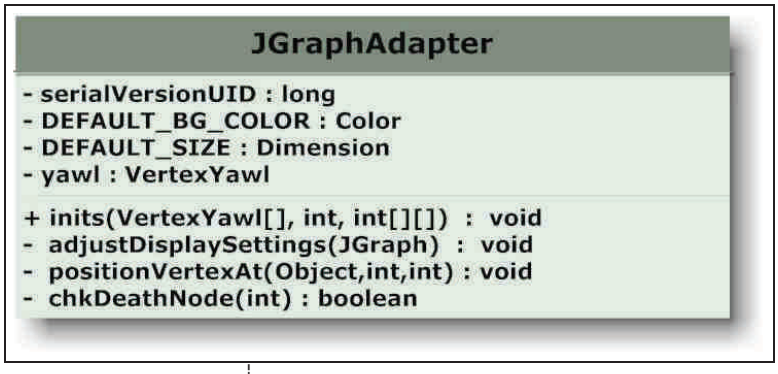
ภาพที่ 3.27 คลาส Hammockgraph

12) คลาส CreateYAWL คือคลาสที่ทำหน้าสร้างไฟล์กระแสงานยอวล์ใหม่หลังปรับโครงสร้าง รายละเอียดของคลาสแสดงดังภาพที่ 3.28



ภาพที่ 3.28 คลาส CreateYAWL

13) คลาส JGraphAdapter คือคลาสที่ทำหน้าแสดงรายละเอียดกระแสงานควบคุมรายละเอียดของคลาสแสดงดังภาพที่ 3.29

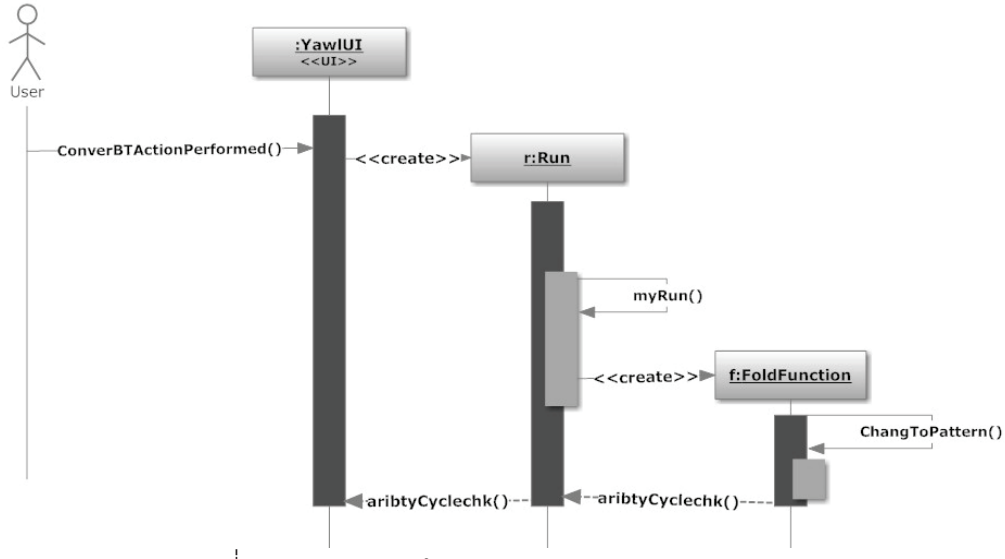


ภาพที่ 3.29 คลาส JGraphAdapter

3.4.3 แผนภาพลำดับ

แผนภาพลำดับเป็นแผนภาพที่แสดงปฏิสัมพันธ์ระหว่างวัตถุต่างๆที่อยู่ในระบบโดย แผนภาพลำดับของเครื่องมือมีดังต่อไปนี้

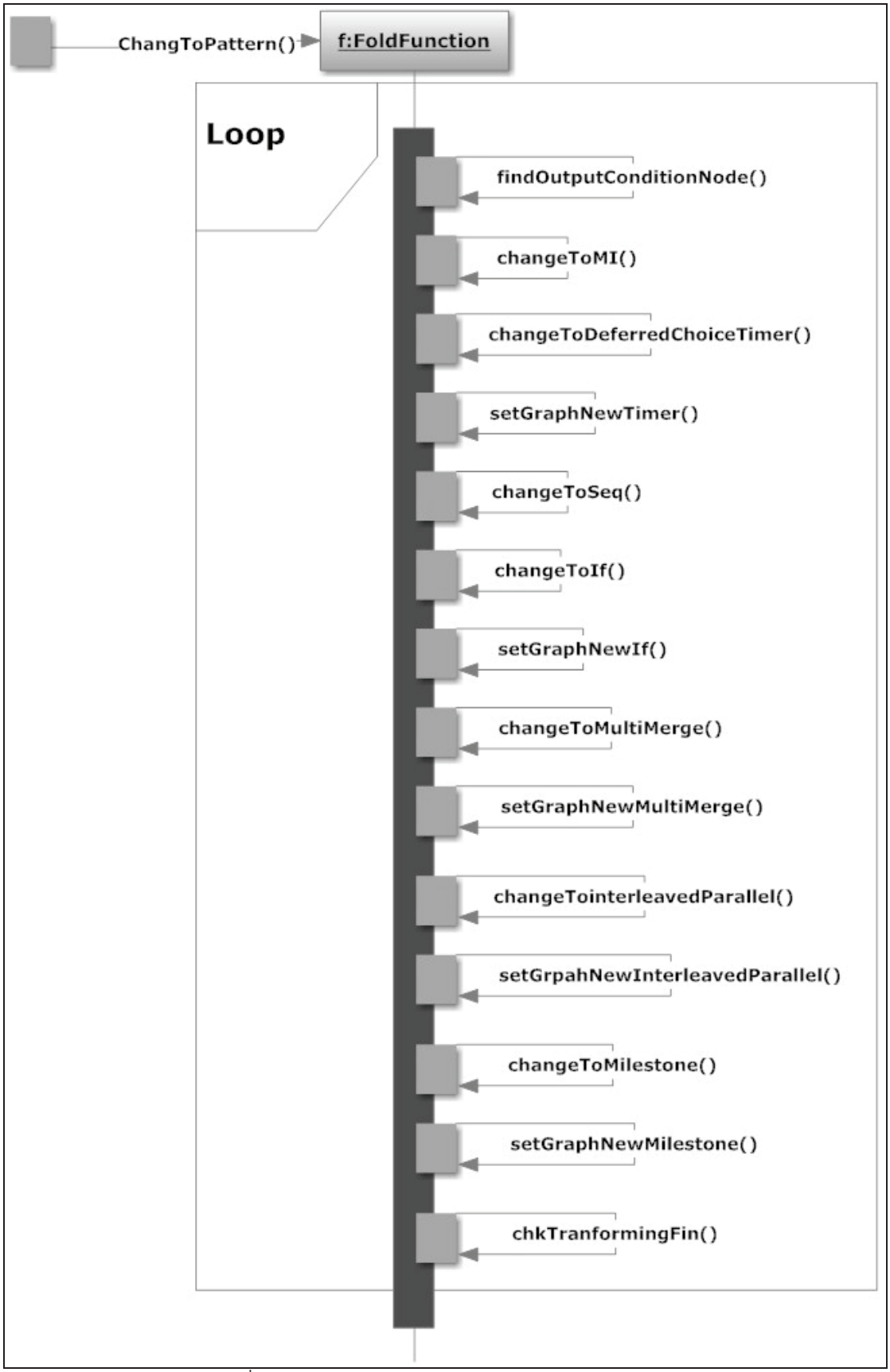
1) แผนภาพลำดับ Convert YAWL workflow ดังภาพที่ 3.30 เป็นการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพลเริ่มจากผู้ใช้นำกระแสงานยอวล์เข้าสู่ระบบ หลังจากนั้นผู้ใช้กดปุ่มคำสั่งเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพล ระบบทำการสร้างอ็อบเจกต์ Run หลังจากนั้นเรียกใช้เมท็อด myRun เพื่อทำการสร้างอ็อบเจกต์ Foldfunction และเรียกใช้เมท็อด ChangToPattern เพื่อใช้ตัวดำเนินการยุบรวมกระแสงานยอวล์ตามแบบรูปที่กำหนดและส่งอ็อบเจกต์ AribrtyCycleChk() กลับมา ซึ่งใช้แผนภาพกิจกรรมดังต่อไปนี้ในการแสดงลำดับการทำงานได้ดังนี้



ภาพที่ 3.30 แผนภาพลำดับ Convert YAWL workflow(2)

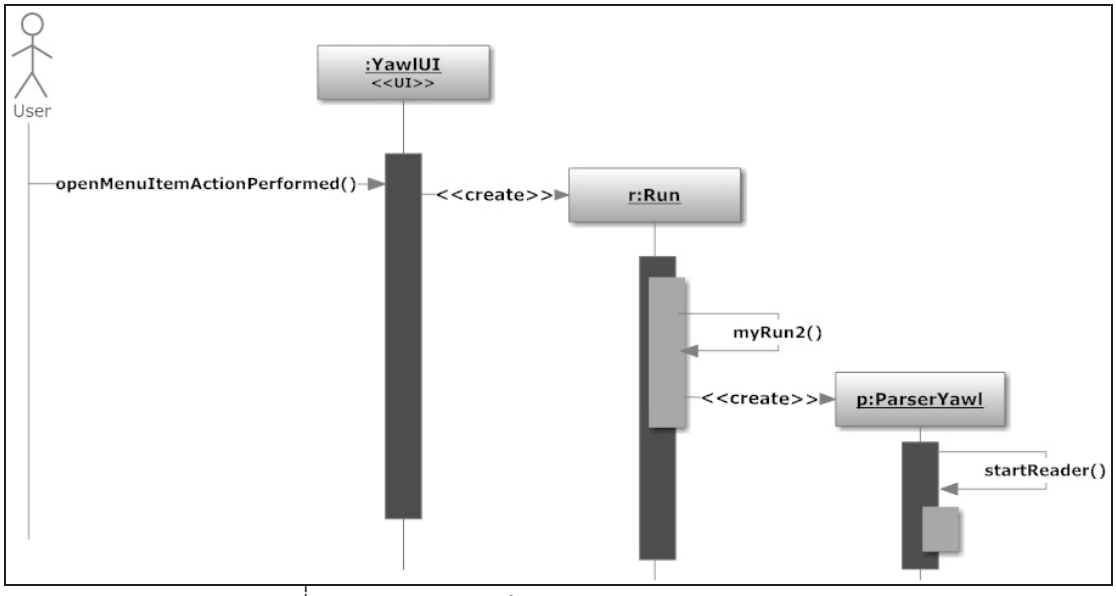
2) แผนภาพลำดับ Run fold function ดังภาพที่ 3.31 หลังจากผู้ใช้นำเข้ากระแสงานยอวล์และกดปุ่มคำสั่งเปลี่ยนกระแสงานยอวล์แล้ว ระบบทำการเรียกเมท็อด ChangeToPattern

ระบบจะทำการวิเคราะห์ และเปรียบเทียบกระแสงานยอวล์ กับแบบรูปในระบบโดยกระแสงานยอวล์ตรงกับแบบรูปในระบบ ระบบจะให้กระแสงานยอวล์นั้นเป็นแบบรูปโครงสร้างที่ดีสามารถรวมกระแสงานควบคุมเป็นโครงร่างบีเพลได้ โดยลำดับในการวิเคราะห์และเปรียบเทียบกระแสงานควบคุมยอวล์ ระบบจะใช้เรียกเมท็อด changeToSeq findOutputConditionNode changeToDeferredChoiceTimer changeToMilestone changeToIf changeToMultiMerge changeTointerleavedPatallel และทำการเก็บแต่ละโหนดในแต่ละแบบรูปที่ตรงกับแบบรูปที่ดีเข้าสู่คิวของระบบ และระบบเรียกใช้เมท็อดsetGraphNewTimer setGraphNewIf setGraphNewMultiMerge setGraphNewInterleavedParallel setGraphNewMilestone เพื่อทำการยุบกระแสงานควบคุมยอวล์ใหม่และเปลี่ยนเป็นโครงร่างบีเพล และระบบเรียกเมท็อด chkTranformingFin() เพื่อตรวจสอบการเปลี่ยนกระแสงานควบคุมยอวล์เสร็จสิ้น



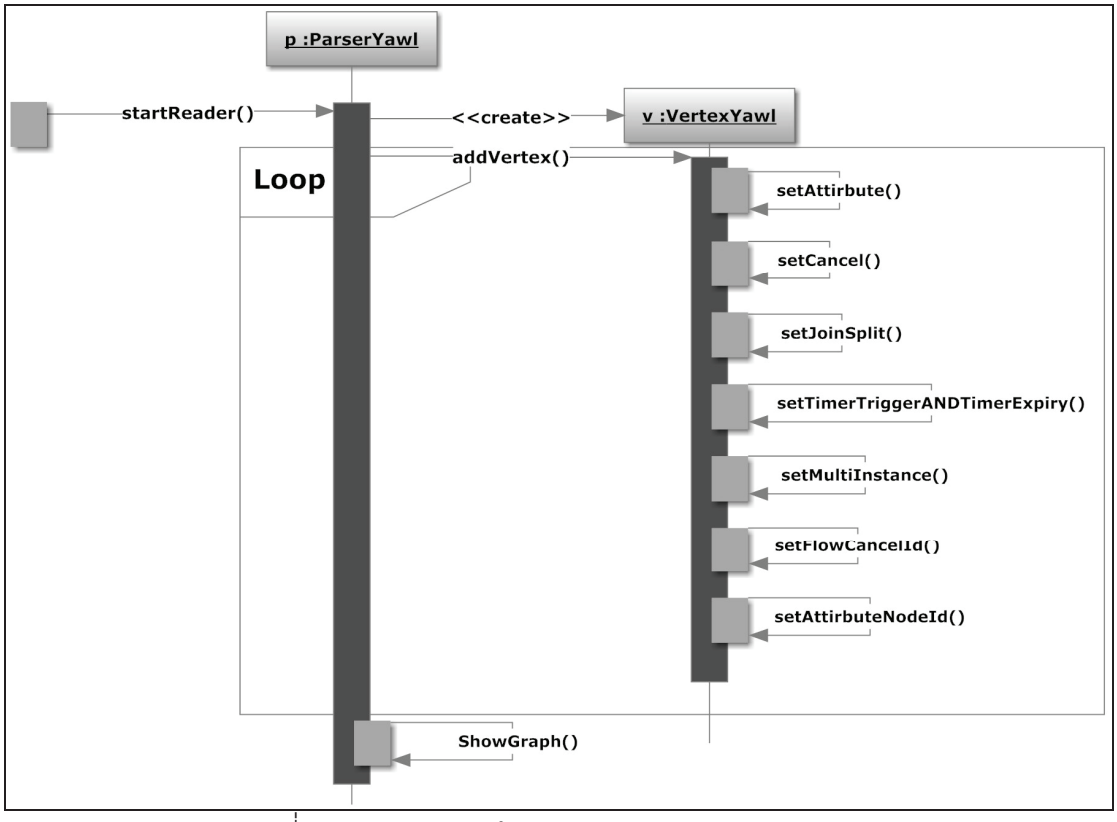
ภาพที่ 3.31 แผนภาพลำดับ Run fold function(2.1)

3) แผนภาพลำดับ Input yawl workflow ดังภาพที่ 3.32 การนำเข้ากระแสงานยอวล์เข้าสู่ระบบโดยผู้ใช้เรียกเมท็อด openMenuItemActionPerformed เพื่อเรียกอ็อบเจกต์ YawlUI โดยอ็อบเจกต์นี้จะสั่งให้อ็อบเจกต์ Run เรียก อ็อบเจกต์ ParserYawl แล้วจึงทำการอ่านกระแสงานยอวล์เข้าสู่ระบบผ่านเมท็อด startReader



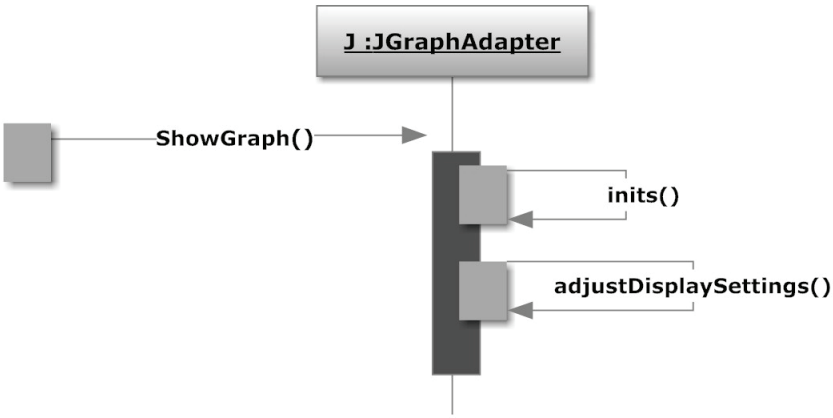
ภาพที่ 3.32 แผนภาพลำดับ Input yawl workflow(1)

4) แผนภาพลำดับ Parser Yawl workflow ดังภาพที่ 3.33 การอ่านกระแสงานยอวล์เข้าสู่ระบบสามารถทำได้ผ่านเมท็อด startReader โดยเมท็อดนี้จะสั่งให้อ็อบเจกต์ ParserYawl เรียกใช้เมท็อด addVertex เพื่ออ่านแต่ละโหนดในกระแสงานยอวล์แล้วเก็บลักษณะแต่ละโหนดอยู่ในอ็อบเจกต์ VertexYawl



ภาพที่ 3.33 แผนภาพลำดับ Parser Yawl workflow(1.2)

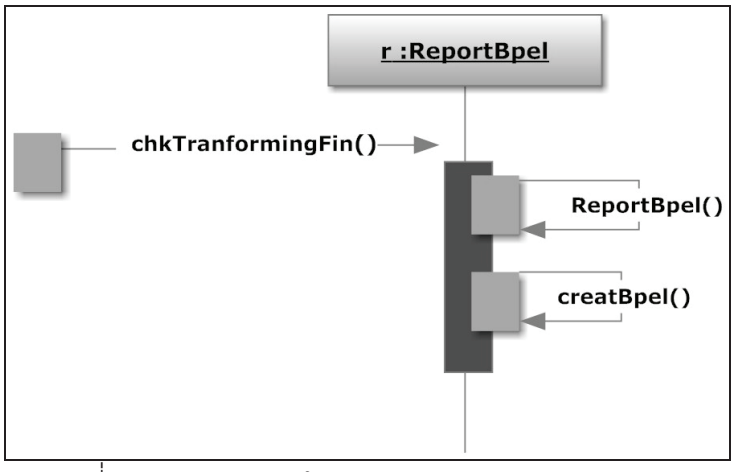
5) แผนภาพลำดับ Show workflow ดังภาพที่ 3.34 สร้างกราฟิกกระแสงานยอวล์ในระบบผ่านเมท็อด ShowGraph โดยเมท็อดนี้จะสั่งให้อ็อบเจกต์ JGraphAdapter เรียกใช้เมท็อด inits เพื่อกำหนดตำแหน่งแต่ละโหนดและแสดงใน Jpanel แล้วเรียกใช้เมท็อด adjustDisplaySettings เพื่อกำหนดลักษณะการแสดงผล



ภาพที่ 3.34 แผนภาพลำดับ Show workflow(1.3)

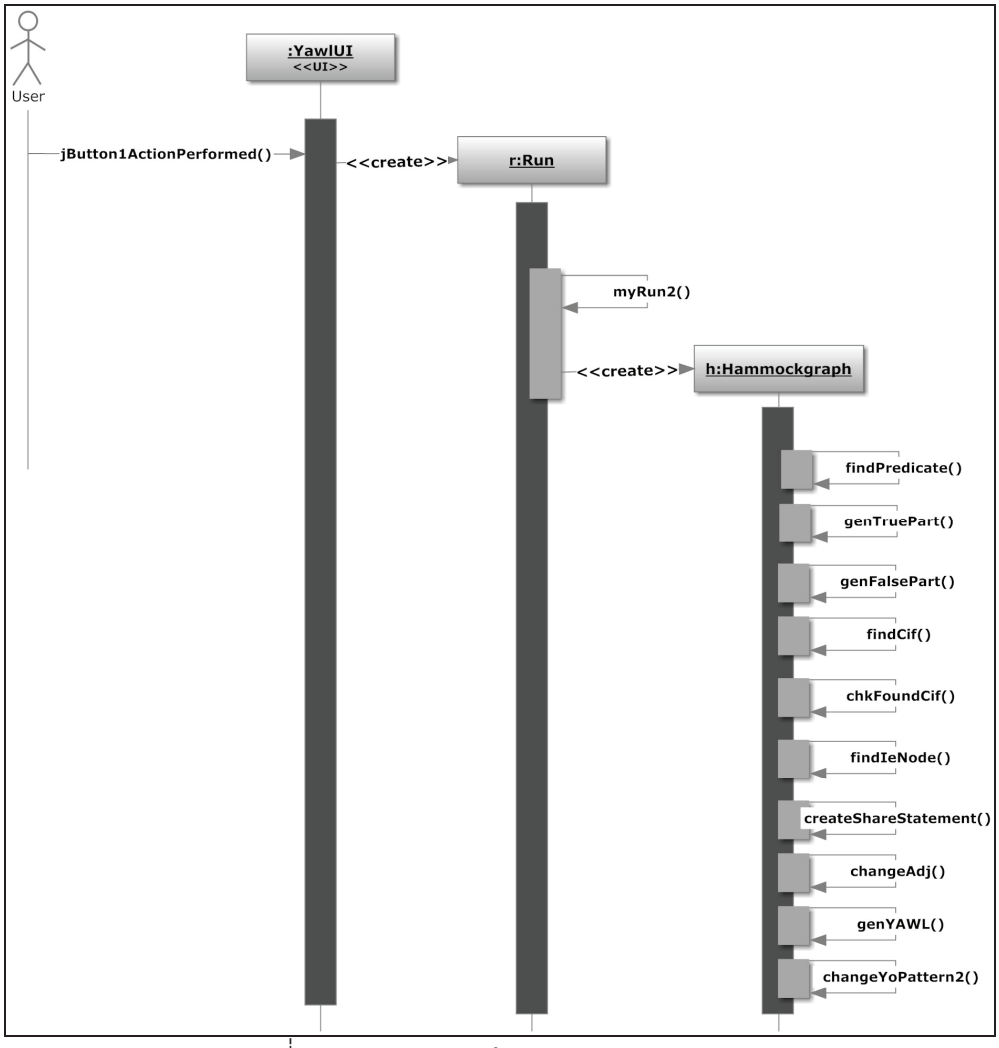
6) แผนภาพลำดับ Create BPEL skeleton ดังภาพที่ 3.35 ระบบสร้างไฟล์โครงร่างบีเพลผ่านเมท็อด chkTranformingFin โดยเมท็อดนี้จะสั่งให้อ็อบเจกต์ ReportBpel เรียกใช้เมท็อด

ReportBpel เพื่อกำหนดรายละเอียดโครงร่างบีเพล แล้วเรียกใช้เมท็อด creatBpel เพื่อสร้างไฟล์โครงบีเพลให้ผู้ใช้



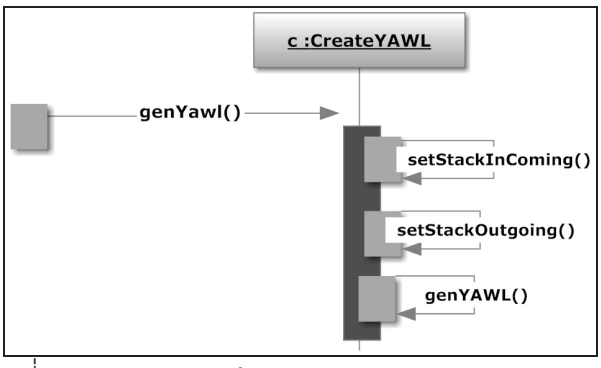
ภาพที่ 3.35 แผนภาพลำดับ Create BPEL skeleton(2.2)

7) แผนภาพลำดับ Change Yawl workflow ดังภาพที่ 3.36 เป็นการเปลี่ยนกระแสงานยอวล์ที่อยู่ในแบบรูปที่โครงสร้างไม่ดีเป็นกระแสงานยอวล์มีโครงสร้างที่ดี หลังจากนั้นผู้ใช้กดปุ่มคำสั่งปรับปรุงโครงสร้างกระแสงานยอวล์ ระบบทำการสร้างอ็อบเจกต์ Run หลังจากนั้นเรียกใช้เมท็อด myRun2 เพื่อทำการสร้างอ็อบเจกต์ Hammock และเรียกใช้เมท็อด findPredicate เมท็อด genTruePart เมท็อด genFalsePart เมท็อด findCif เมท็อด chkFoundCif เมท็อด findIeNode เมท็อด createShareStatement เมท็อด changeAdj เมท็อด genYAWL และเมท็อด changeToPattern2 เพื่อใช้ในการปรับปรุงโครงสร้างกระแสงานยอวล์และทำการยุบรวมกระแสงานยอวล์ตามแบบรูปที่กำหนด ซึ่งใช้แผนภาพกิจกรรมดังต่อไปนี้ในการแสดงลำดับการทำงานได้ดังนี้



ภาพที่ 3.36 แผนภาพลำดับ Change Yawl workflow (3)

8) แผนภาพลำดับ Create new YAWL workflow ดังภาพที่ 3.37 ระบบสร้างไฟล์กระแสงานยอวล์ใหม่หลังปรับปรุงโครงสร้างผ่านเมท็อด genYawl โดยเมท็อดนี้จะสั่งให้อ็อบเจกต์ CreateYAWL เรียกใช้เมท็อด setStackIncoming และเมท็อด setStackOutgoing เพื่อกำหนดรายละเอียดกระแสงานยอวล์ แล้วเรียกใช้เมท็อด genYAWL เพื่อสร้างไฟล์กระแสงานยอวล์ให้ผู้ใช้



ภาพที่ 3.37 แผนภาพลำดับ Create BPEL skeleton(3.1)

บทที่ 4
การพัฒนาเครื่องมือ

ในบทนี้จะกล่าวถึงสภาพแวดล้อม และโครงสร้างส่วนต่อประสานกับผู้ใช้ของเครื่องมือการเปลี่ยนรูปกระแสงานยอวล์เป็นภาษาบีเพลซึ่งมีรายละเอียดดังต่อไปนี้

4.1 สภาพแวดล้อมในการพัฒนาระบบ

สภาพแวดล้อมในการพัฒนาระบบจำแนกได้เป็น 2 ประเภท คือ ฮาร์ดแวร์(Hardware) และซอฟต์แวร์ (Software) โดยมีรายละเอียดดังนี้

4.1.1 สภาพแวดล้อมในการพัฒนาเครื่องมือฮาร์ดแวร์

เครื่องคอมพิวเตอร์ตั้งโต๊ะ 1 เครื่อง

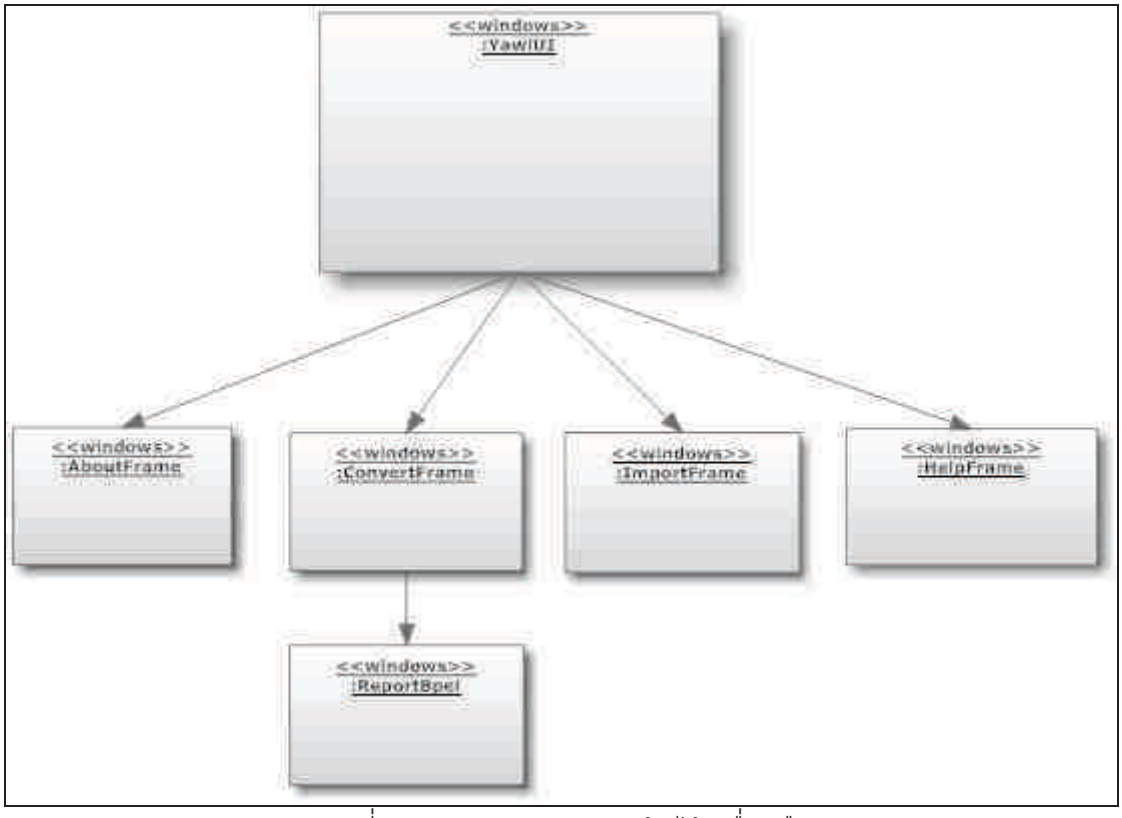
1) หน่วยประมวลผล Intel Core2 Duo Process P8400 ความเร็ว 2.26 กิกะเฮิร์ซ (GHz)
2) หน่วยความจำหลัก DDR2 ขนาด 2048 เมกกะไบต์ (MB)
3) ฮาร์ดดิสก์ความเร็ว 5,400 รอบ/วินาที ขนาด 250 กิกะไบต์ (GB)

4.1.2 สภาพแวดล้อมในการพัฒนาเครื่องมือด้านซอฟต์แวร์

1) ระบบปฏิบัติการวินโดส์เซเว่น เซอร์วิสแพ็ค 1 (Microsoft Windows 7 Service Pack 1) เป็นระบบปฏิบัติการของเครื่องที่ใช้ในการพัฒนา
2) เน็ตบีนส์ไอดีอี เวอร์ชัน 6.7.1 เป็นแกนหลักในการพัฒนาระบบ

4.2 โครงสร้างส่วนต่อประสานกับผู้ใช้ของเครื่องมือ

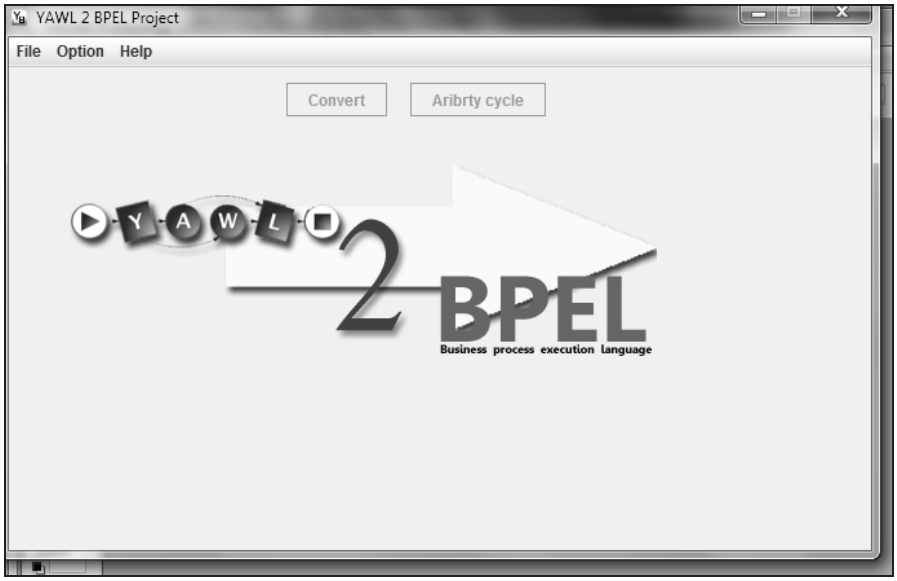
โครงสร้างในส่วนของส่วนต่อประสานกับผู้ใช้ของเครื่องมือการเปลี่ยนรูปกระแสงานยอวล์เป็นโครงร่างบีเพลได้รับการอธิบายด้วยแผนภาพรายการต้นไม้ (Menu tree Diagram) ซึ่งเป็นแผนภาพที่ใช้อธิบายส่วนประกอบต่างๆ ในระบบ โดยแผนภาพส่วนประกอบของส่วนต่อประสานกับผู้ใช้ของเครื่องมือแสดงดังภาพที่ 4.1



ภาพที่ 4. 1 แผนภาพรายการต้นไม้เครื่องมือ

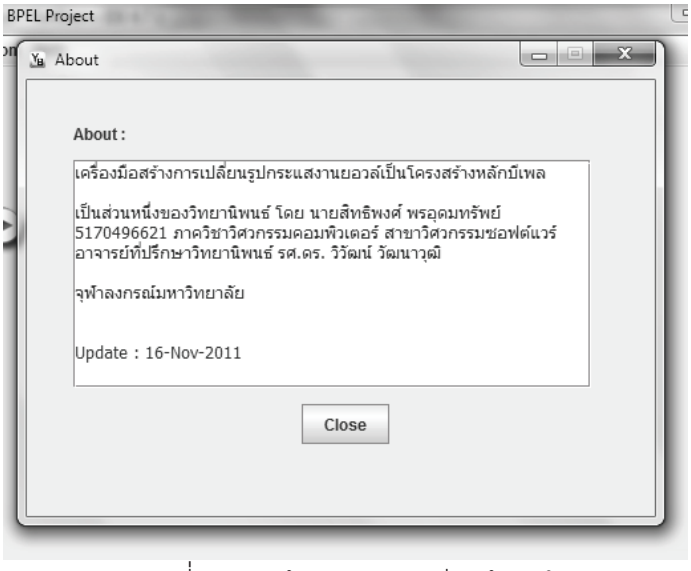
จากภาพที่ 4.1 กรอบ<<windows>>แสดงแทนแต่ละหน้าของส่วนต่อประสานกับผู้ใช้ ซึ่งมีรายละเอียดดังต่อไปนี้

1) หน้าจอ YawlUI เป็นหน้าจอหลักของเครื่องมือการเปลี่ยนรูปกระแสงานยอวล์เป็นโครงร่างบีเพลแสดงดังภาพที่ 4.2



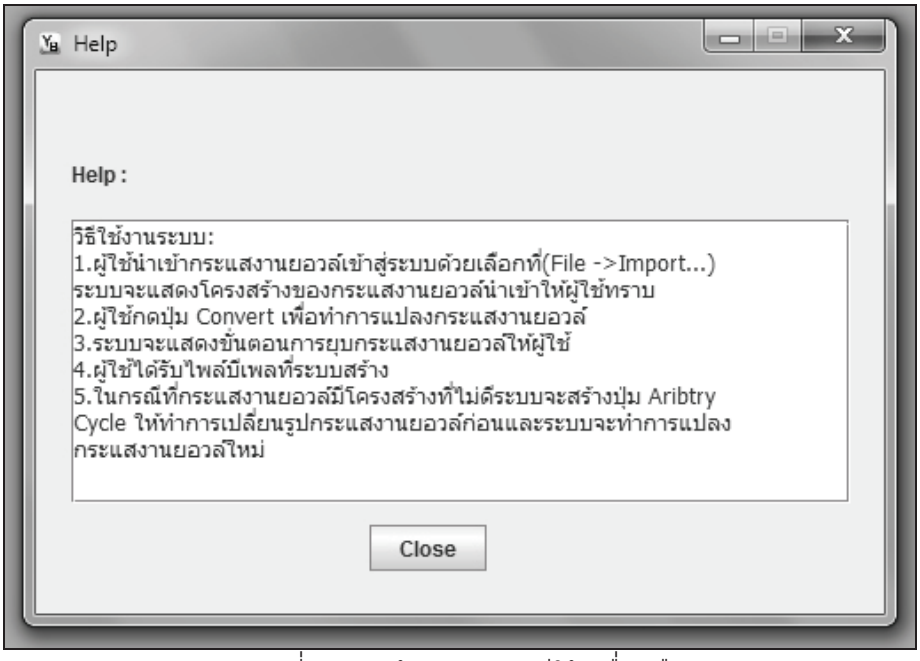
ภาพที่ 4.2 หน้าจอหลักของเครื่องมือ

2) หน้า AboutFrame แสดงดังภาพที่ 4.3 หน้าจอนี้มาจากการคลิกที่ปุ่ม Help-> About ของหน้าจอ YawlUI แสดงรายละเอียดของเครื่องมือสร้างการเปลี่ยนรูปกระแสงานยอวล์เป็นโครงร่างบีเพล



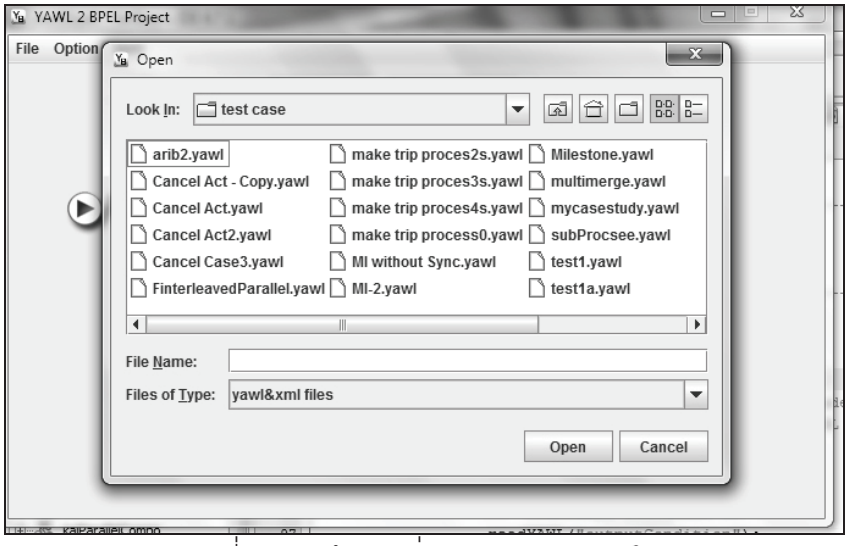
ภาพที่ 4.3 หน้าจอรายละเอียดผู้จัดทำ

3) หน้า HelpFrame แสดงดังภาพที่ 4.4 หน้าจอนี้มาจากการคลิกที่ปุ่ม Help->Help ของหน้าจอYawlUI โดยแสดงรายละเอียดขั้นตอนวิธีการใช้ระบบ



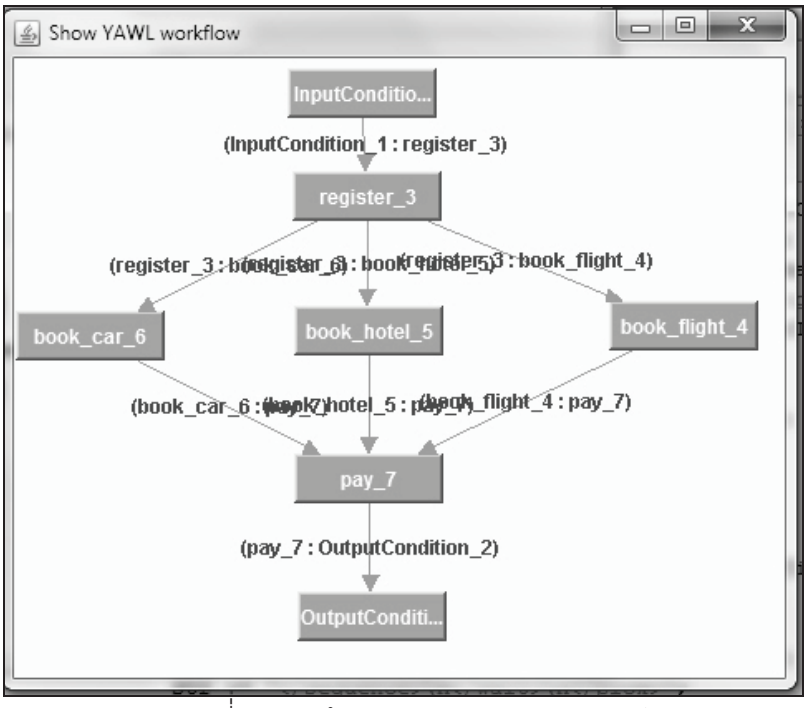
ภาพที่ 4.4 หน้าจอแสดงวิธีใช้เครื่องมือ

4) หน้า ImportFrame แสดงดังภาพที่ 4.5 ใช้ในการนำเข้าไฟล์กระแสงานยอวล์เข้าสู่ระบบที่อยู่ในรูปของไฟล์ยอวล์



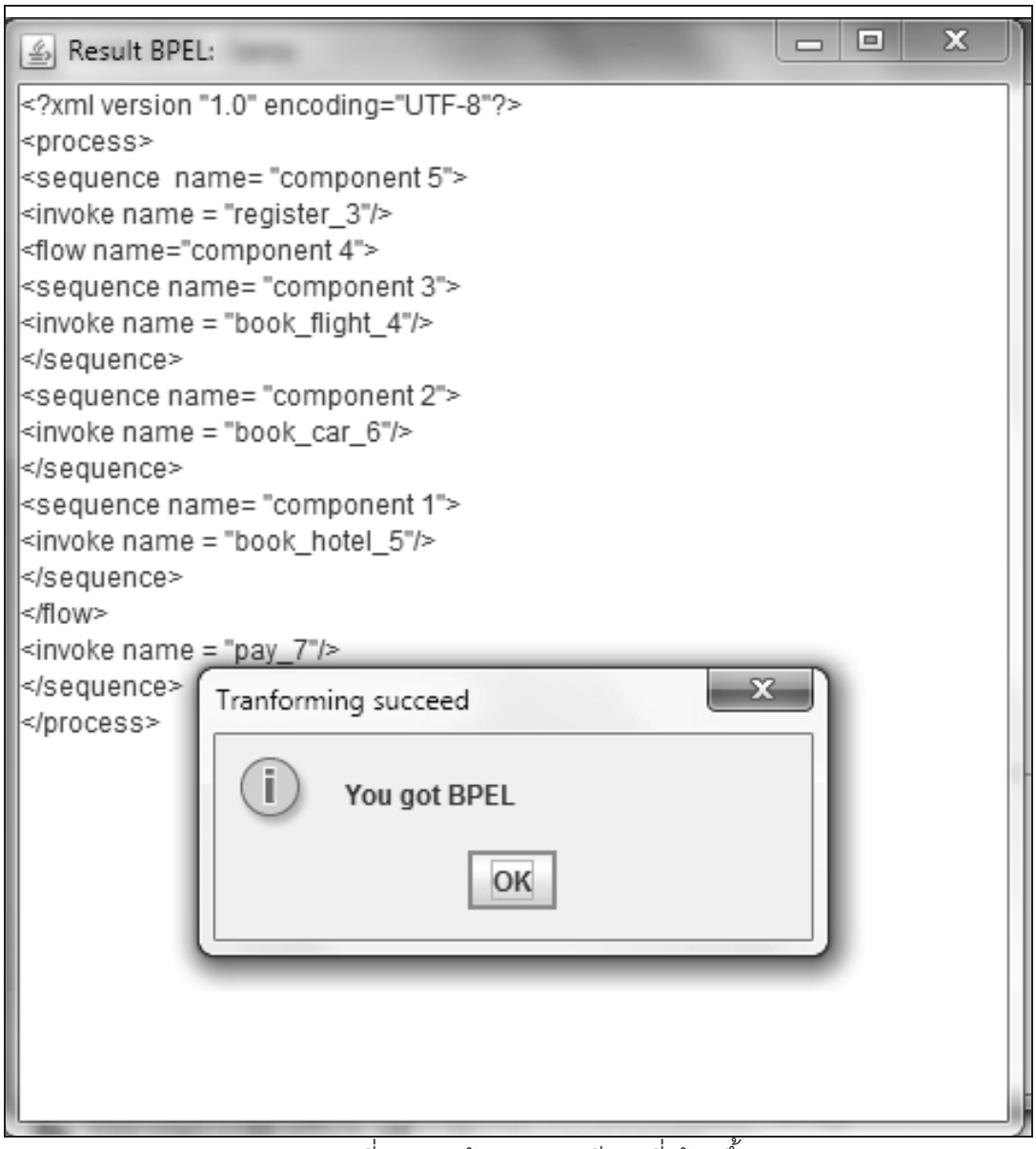
ภาพที่ 4.5 หน้าจอเพิ่มกระแสงานยอวล์ใหม่

5) หน้า JFrame แสดงดังรูป 4.6 มีไว้เพื่อแสดงกระแสงานยอวล์ที่นำเข้าในระบบจากหน้า ImportFrame ซึ่งจะแสดงรายละเอียดกราฟสายงานควบคุมในกระแสงาน และเมื่อผู้ใช้กดปุ่ม Convert ระบบจะทำแปลงกระแสงานเป็นบีเพลที่หน้าจอนี้จะแสดงลำดับในการลดกราฟสายงานควบคุมของระบบ



ภาพที่ 4.6 หน้าจอแสดงกระแสงานยอวล์

6) หน้า ReportBpel หน้าจอนี้มาจากผลการแปลงกระแสงานยอวล์เป็นโครงร่างบีเพล เสร็จเรียบร้อยเครื่องมือทำการแสดงรายงานผลของการแปลงดังภาพที่ 4.7



ภาพที่ 4.7 หน้าจอแสดงบีเพลที่สร้างขึ้น

บทที่ 5
การทดสอบ

การทดสอบเครื่องมือการเปลี่ยนรูปกระแสงานยอวล์เป็นโครงร่างบีเพลนั้นจะทำได้โดยการสร้างกระแสงานยอวล์ทดสอบประกอบไปด้วย 6 กระแสงานทดสอบ และการทดสอบครอบคลุมลักษณะแบบรูปทั้ง 19 แบบรูป

5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ

เป็นสภาพแวดล้อมเดียวกับที่ใช้ในการพัฒนาเครื่องมือในบทที่ 4

5.2 ขั้นตอนการทดสอบเครื่องมือ

1) สร้างกระแสงานยอวล์ทดสอบโดยสร้างจากโปรแกรม Yawl editor
2) นำกระแสงานยอวล์ทดสอบเปลี่ยนเป็นโครงร่างบีเพลโดยใช้เครื่องมือ
3) นำโครงร่างบีเพลที่เครื่องมือสร้างขึ้น ใส่รายละเอียดให้โครงร่างบีเพลสามารถทำงานได้ โดยเพิ่มส่วนติดต่อเว็บเซอร์วิส แอ็คติวิตี<partnerLink> แอ็คติวิตี<assign>และส่วนติดต่อภาษาดับบลิวเอสดีแอล
4) เริ่มทำการทดสอบโดยเปรียบเทียบลำดับการทำงานกระแสงานยอวล์กับโครงร่างบีเพลที่เพิ่มรายละเอียดแล้ว
5) รวบรวมและสรุปผลการทดสอบ

5.3 กระแสงานที่ใช้ในการทดสอบ

งานวิทยานิพนธ์นี้สร้างกระแสงานยอวล์ทดสอบเพื่อใช้ในการทดสอบเครื่องมือ 6 ระบบกระแสงาน คือระบบเช่ายืมหนังสือ ระบบกู้ชำระเงิน ระบบการลงทะเบียนเรียน ระบบการสอบ ระบบรับพนักงานใหม่ และระบบสั่งซื้อหนังสือ โดยกระแสงานเหล่านี้ถูกสร้างขึ้นจากโปรแกรม Yawl editor มีรายละเอียดดังนี้

5.3.1 ระบบเช่ายืมหนังสือ

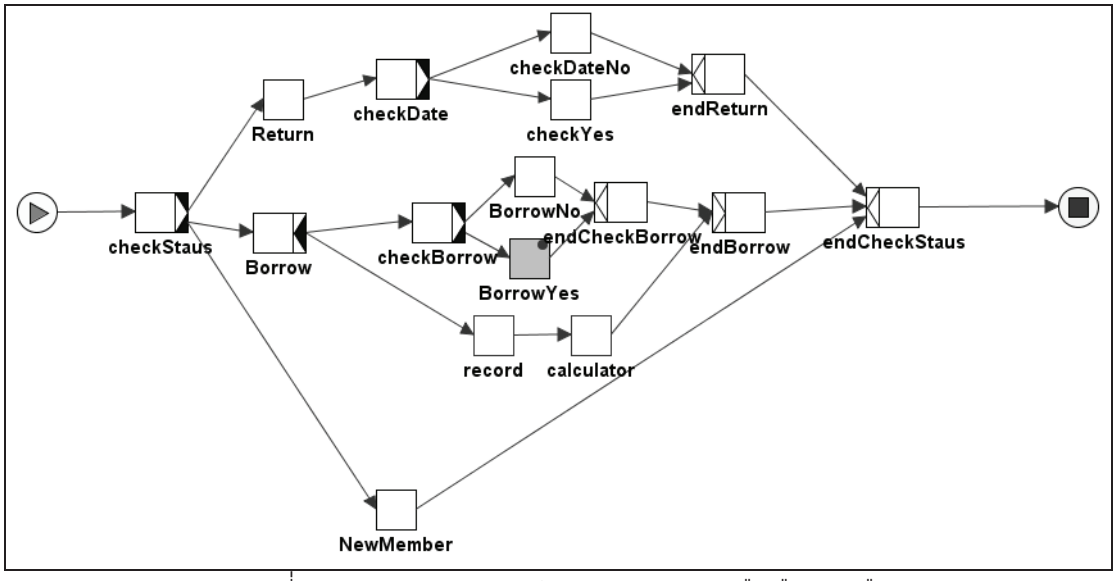
กระแสงานยอวล์ทดสอบระบบเช่ายืมคืนหนังสือ มีลักษณะการทำงานตามแผนภาพกระแสงานยอวล์ ดังภาพที่ 5.1 โดยกระแสงานเริ่มการทำงานจาก เซอร์วิส checkstaus เป็นเซอร์วิสรับการทำงานจากผู้ใช้ โดยแบ่งเป็น 3 กรณี

1) ในกรณีผู้ใช้เลือกเซอร์วิส NewMember เซอร์วิสทำการสมัครสมาชิกใหม่ และเรียกเซอร์วิส endCheckStaus และจบการทำงาน

2) ในกรณีผู้ใช้เลือกเซอร์วิส Return เพื่อทำการคืนหนังสือระบบทำการตรวจสอบการคืนหนังสือ โดยเรียกเซอร์วิส checkDate เพื่อตรวจสอบสถานการณ์ยืมเกินกำหนด โดยแบ่งเป็น 2 กรณี คือ 1. ยืมเกินกำหนดเซอร์วิส checkDateYes จะถูกทำงานเพื่อคิดเงินค่าปรับ 2. ยืมไม่เกินกำหนดเซอร์วิส checkDateNo จะถูกเรียกทำงาน แล้วเรียกเซอร์วิส endReturn เซอร์วิส endCheckstaus ทำงานและจบการทำงาน

3) ในกรณีผู้ใช้เลือกเซอร์วิส Borrow ยืมหนังสือ เซอร์วิสจะทำการเรียก เซอร์วิส record เพื่อทำการบันทึกข้อมูลการยืมสู่ระบบ แล้วเรียกเซอร์วิส calculator เพื่อคิดเงินค่ายืมหนังสือ และระบบจะเรียก เซอร์วิส checkBorrow โดยทำงานพร้อมกับเซอร์วิส record และ เซอร์วิส calculator เพื่อทำการตรวจสอบการยืมในระบบโดยถ้าพบมีการยืมหนังสือค้างในระบบ เซอร์วิส BorrowYes จะทำการยกเลิกการทำงานเซอร์วิส record และ เซอร์วิส calculator แล้วระบบเรียกเซอร์วิส endBorrow เซอร์วิส endCheckStaus จบการทำงาน

โดยกระแสงานทดสอบนี้ประกอบด้วย 6 แบบรูป แบบรูป Sequence แบบรูป Parallel Split รวมกับแบบรูป Synchronization แบบรูป Exclusive Choice รวมกับแบบรูป Simple Merge และแบบรูป Cancel Activity ตามตารางที่ 5.1 แสดงรายละเอียดแบบรูปของระบบเช่ายืมคืนหนังสือ



ภาพที่ 5.1 กระแสงานยอวล์ทดสอบระบบเช่ายืมคืนหนังสือ

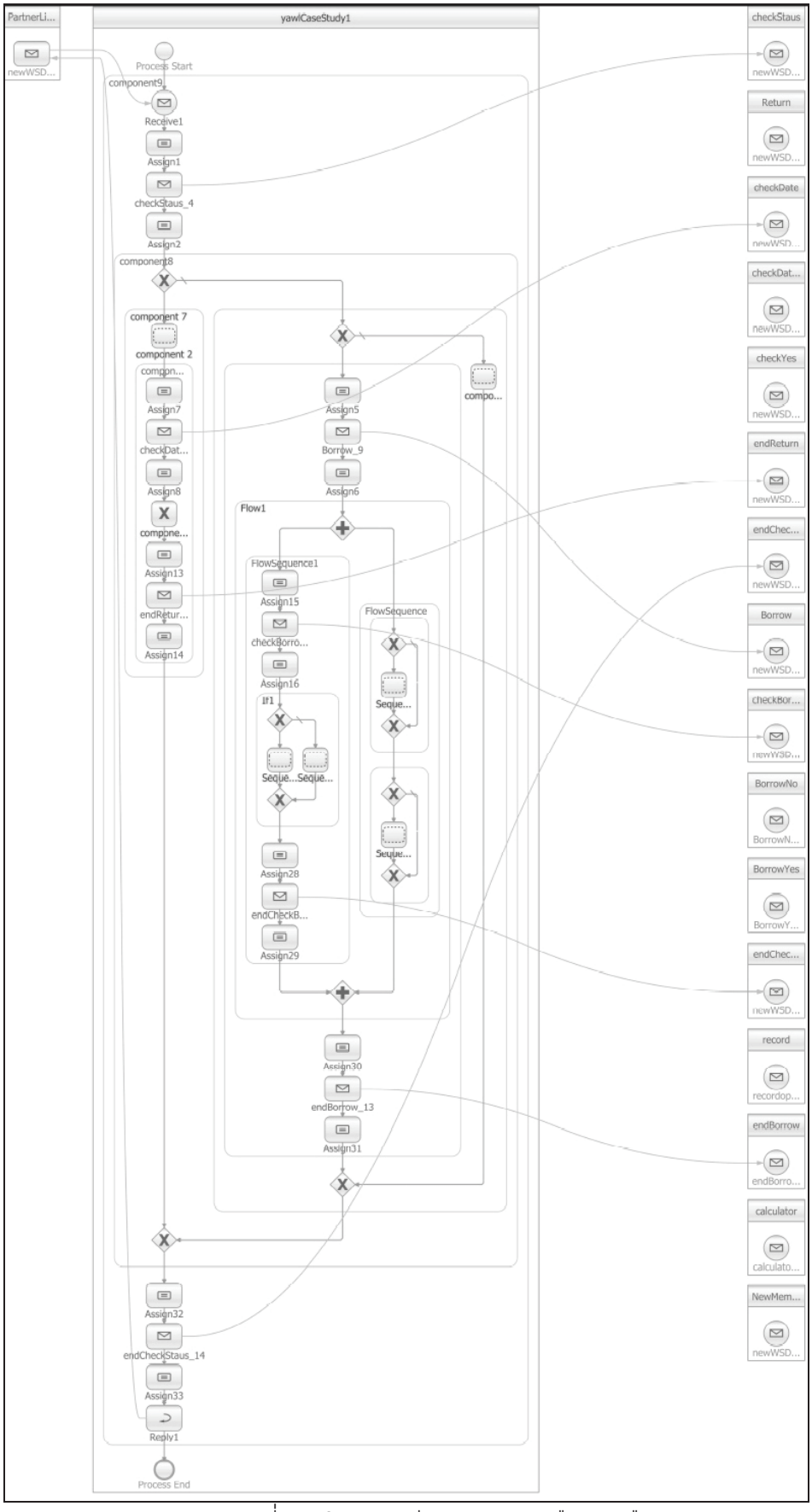
ตารางที่ 5.1 แบบรูปในระบบเช่ายืมคืนหนังสือ

	แบบรูป	**โครงสร้างแบบรูปในกระแสงานยอวล์**
1	แบบรูป Exclusive Choice รวมกับ แบบรูปSimple Merge	โครงสร้างประกอบด้วยเซอร์วิส checkStaus ร่วมกับเซอร์วิส endCheckStaus

ตารางที่ 5.1 แบบรูปในระบบเช่ายืมคืนหนังสือ(ต่อ)

	แบบรูป	**โครงสร้างแบบรูปในกระแสงานยอวล์**
2	แบบรูป Parallel Split รวมกับ แบบรูป Synchronization	โครงสร้างประกอบด้วยเซอร์วิส Borrow ร่วมกับเซอร์วิส endBorrow
3	แบบรูป Cancel Activity	โครงสร้างประกอบด้วยเซอร์วิส BorrowYes เป็น cancellation ประกอบเซอร์วิสCancellation set คือเซอร์วิส record และ เซอร์วิส calculator
4	แบบรูป Sequence	โครงสร้างประกอบด้วยเซอร์วิส record และเซอร์วิส calculator

จากโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติมให้โครงร่างบีเพลทำงานได้ตามภาพที่ 5.2



ภาพที่ 5.2 โครงร่างบีเพลระบบเช่ายืมหนังสือ

การทดสอบความถูกต้องของโครงร่างบีเพลที่เครื่องมือสร้างขึ้นมา ทำการสร้างกรณีทดสอบสำหรับ ระบบเช่ายืมหนังสือโดยเปรียบเทียบลำดับการทำงานในกระแสงานยอวล์ทดสอบกับโครงร่างบีเพลที่สร้างขึ้นตาม ตารางที่ 5.2

ตารางที่ 5.2 กรณีทดสอบกระแสงานยอวล์ระบบเช่ายืมหนังสือ

ลำดับ	กรณีทดสอบ	ผลลัพธ์
1	ลำดับการทำงานในกระแสงานคือ เซอร์วิส checkStaus เลือกเซอร์วิส NewMember ทำงาน เซอร์วิส endCheckStaus	✓
2	ลำดับการทำงานในกระแสงานคือ เซอร์วิส checkStaus เลือกเซอร์วิส Return ทำงาน เซอร์วิส checkDate ตรวจสอบการยืม กรณียืมไม่เกินกำหนด เรียกเซอร์วิส checkDateNo แล้วเรียกเซอร์วิส endReturn เซอร์วิส endCheckStaus	✓
3	ลำดับการทำงานในกระแสงานคือ เซอร์วิส checkStaus เลือกเซอร์วิส Return ทำงาน เซอร์วิส checkDate ตรวจสอบการยืม กรณีเกินกำหนดเรียกเซอร์วิส CheckDateYes แล้วเรียกเซอร์วิส endReturn เซอร์วิส endCheckStaus	✓
4	ลำดับการทำงานในกระแสงานคือ เซอร์วิส checkStaus เลือกการทำงานเซอร์วิส Borrow ทำการเรียกเซอร์วิส checkBorrow ทำการตรวจสอบการยืม และเรียกเซอร์วิส record ทำการบันทึกข้อมูลการยืมพร้อมกัน ถ้ามีการรายการยืมค้างในระบบเซอร์วิส BorrowYes ถูกทำงานจะยกเลิกการทำงานเซอร์วิส record เซอร์วิส calculator แล้วเรียกเซอร์วิส endBorrow เซอร์วิส endCheckStaus	✓
5	ลำดับการทำงานในกระแสงานคือ เซอร์วิส checkStaus เลือกการทำงานเซอร์วิส Borrow ทำการเรียกเซอร์วิส checkBorrow ทำการตรวจสอบการยืม และเรียกเซอร์วิส record ทำการบันทึกข้อมูลการยืมพร้อมกัน ถ้าไม่มีการรายการยืมค้างในระบบเซอร์วิส BorrowNo ถูกทำงาน แล้วเกิดการทำงานเซอร์วิส calculator เซอร์วิส endBorrow เซอร์วิส endCheckStaus	✓

5.3.2 ระบบกู้ชำระเงิน

กระแสงานยอวล์ระบบกู้ชำระเงินที่ใช้ในการทดสอบ มีลักษณะการทำงานตามแผนภาพกระแสงานยอวล์ดังภาพที่ 5.3 โดยกระแสงานเริ่มต้นการทำงานจาก เซอร์วิส StartLone เซอร์วิส chkDocument ทำการตรวจสอบเอกสารการกู้เงิน ในกรณีที่เอกสารไม่ครบระบบจะจบการทำงานของกระแสงาน

1) หลังจากเซอร์วิส chkDocument ตรวจสอบเอกสาร เซอร์วิส chkRisk ทำงานเพื่อประเมินความเสี่ยงด้วยการเรียก เซอร์วิส assets เพื่อตรวจสอบทรัพย์สิน เซอร์วิส income เพื่อตรวจสอบรายได้ เซอร์วิส credicard เพื่อตรวจสอบเครดิตการ์ด และเซอร์วิส debt เพื่อตรวจสอบหนี้สิน ลำดับการทำงานทั้ง 4 เซอร์วิสสามารถเกิดขึ้นก่อน หลังกันได้โดยไม่เกิดการทำงานพร้อมกัน และเมื่อทำงานครบ 4 เซอร์วิส จะเกิดการงานที่เซอร์วิส EndChkRisk

2) จากนั้นเซอร์วิส approval แสดงผลการอนุมัติการกู้เงิน ในกรณีที่ไม่อนุมัติการกู้เงิน เซอร์วิส approvalNo ทำงาน ในกรณีอนุมัติการกู้เงิน เซอร์วิส approvalYes ทำงาน แล้วเซอร์วิส repay ทำการเลือกชนิดการชำระเงินคืน

3) กรณีเลือกเซอร์วิส repayTypeA ทำงานที่เซอร์วิสนี้เป็นแบบ multiple task มีลักษณะเป็น [1,n,m,static] สามารถเกิดการทำงานเซอร์วิสวนซ้ำเท่ากับ n ครั้ง จำนวน completionCondition เท่ากับ m และ static คือจำนวนการวนซ้ำกำหนดที่ขั้นตอนออกแบบกระแสงาน

4) กรณีเลือกเซอร์วิส repayTypeB ทำงานที่เซอร์วิสนี้เป็นแบบ multiple task มีลักษณะเป็น [1,n,n,static] สามารถเกิดการทำงานเซอร์วิสวนซ้ำเท่ากับ n ครั้ง จำนวน completionCondition เท่ากับ n และ static คือจำนวนการวนซ้ำกำหนดที่ขั้นตอนการออกแบบ

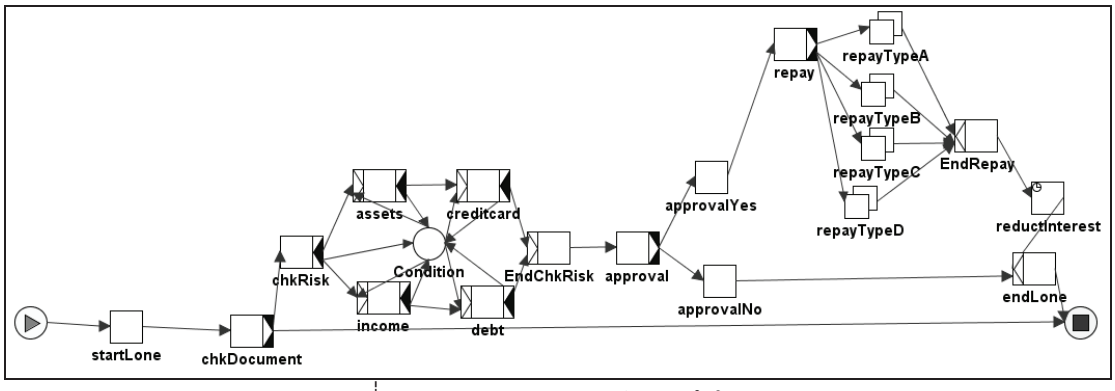
5) กรณีเลือกเซอร์วิส repayTypeC ทำงานที่เซอร์วิสนี้เป็นแบบ multiple task มีลักษณะเป็น [1,inf,inf,static] สามารถเกิดการทำงานเซอร์วิสวนซ้ำเท่ากับ จำนวนผู้ใช้กำหนดตอนทำงานของกระแสงาน

6) กรณีเลือกเซอร์วิส repayTypeD ทำงานที่เซอร์วิสนี้เป็นแบบ multiple task มีลักษณะเป็น [1,inf,inf,dynamic] สามารถเกิดการทำงานเซอร์วิสวนซ้ำเท่ากับจำนวน ความต้องการตอนทำงานของกระแสงานและสามารถเพิ่มจำนวนการวนซ้ำเซอร์วิสได้ขณะทำงาน

7) หลังจากนั้นเรียกเซอร์วิส EndRepay เซอร์วิส reductInterrest เพื่อลดดอกเบี้ยในกรณีชำระเงินครบตามเวลาที่กำหนด เรียกเซอร์วิส endLone และจบการทำงาน

โดยกระแสงานทดสอบนี้ประกอบด้วย 9 แบบรูป มีแบบรูป Cancel Case แบบรูป Interleaved Parallel Routing แบบรูป Discriminator แบบรูป MI with a Priori Design-Time knowledge แบบรูป MI with a Priori Run-Time knowledge แบบรูป MI without a Priori Run-

Time Knowledge แบบรูป Deferred Choice แบบรูป Exclusive Choice รวมกับแบบรูป Simple Merge ตารางที่ 5.3 แสดงรายละเอียดแบบรูปของระบบ

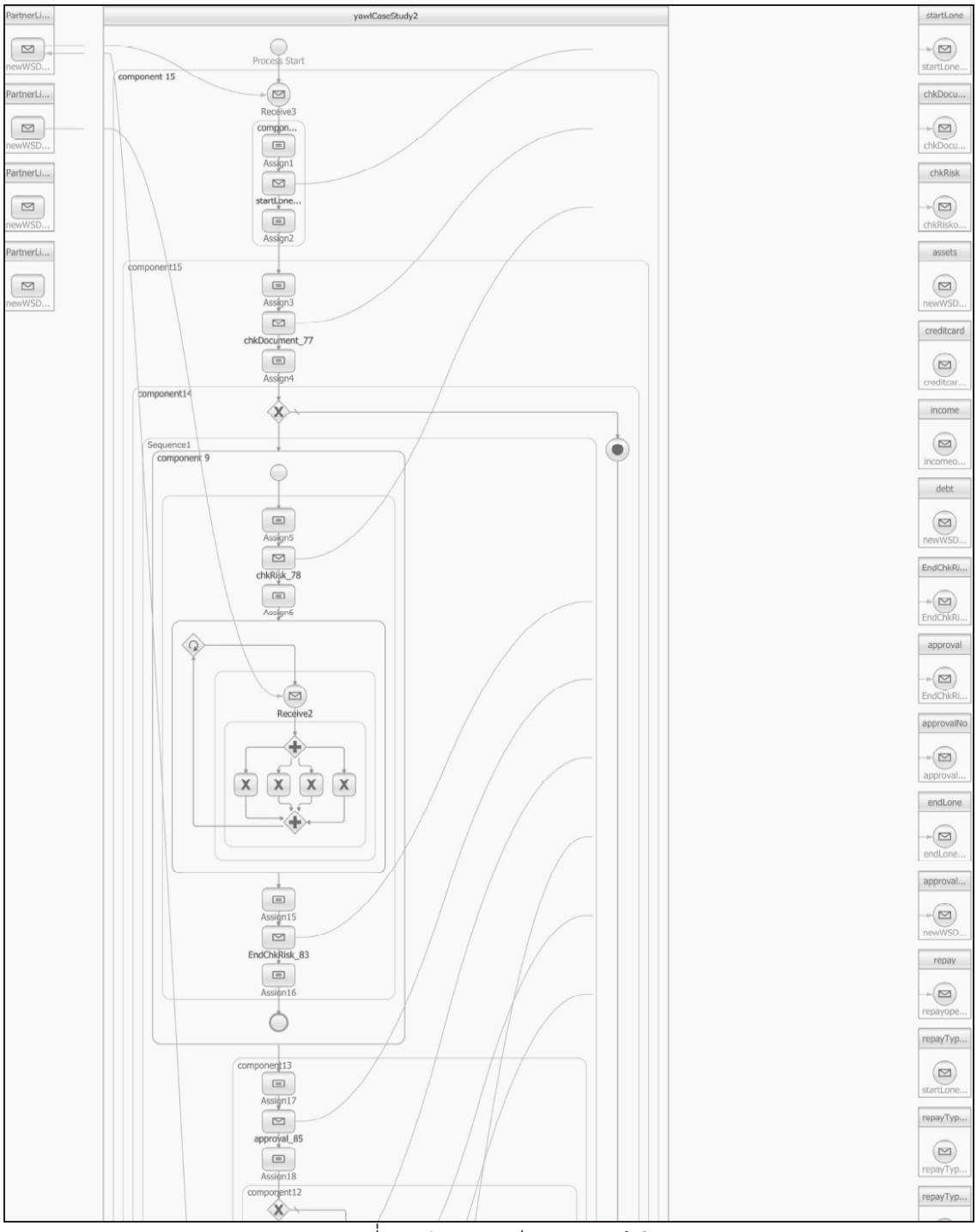


ภาพที่ 5.3 กระแสงานยอวล์ระบบกู้ชำระเงิน

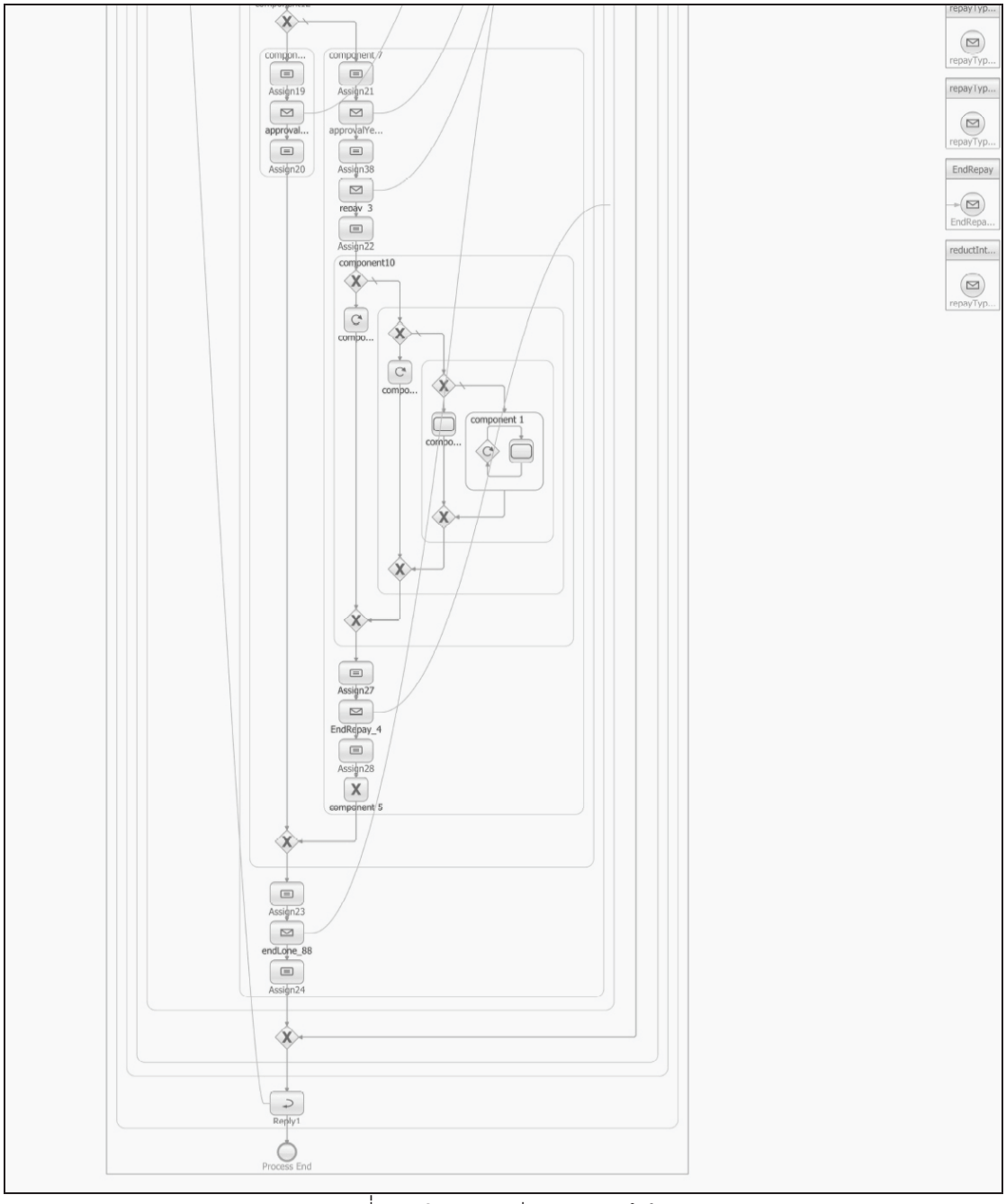
ตารางที่ 5.3 แสดงแบบรูปในรูประบบกู้ชำระเงิน

	แบบรูป	**โครงสร้างแบบรูปในกระแสงานยอวล์**
1	แบบรูป Cancel Case	โครงสร้างประกอบด้วยเซอร์วิส chkDocument กับ outputcondition เซอร์วิส
2	แบบรูป Interleaved Parallel Routing	โครงสร้างประกอบด้วยเซอร์วิส chkRisk เซอร์วิส assets เซอร์วิส creditcard เซอร์วิส income เซอร์วิส debt เซอร์วิส EndChkRisk และ Condition โหนด
3	แบบรูป Discriminator	โครงสร้างประกอบด้วยเซอร์วิส repayTypeA
4	แบบรูป MI with a Priori Design-Time knowledge	โครงสร้างประกอบด้วยเซอร์วิส repayTypeB
5	แบบรูป MI with a Priori Run-Time knowledge	โครงสร้างประกอบด้วยเซอร์วิส repayTypeC
6	แบบรูป MI without a Priori Run-Time Knowledge	โครงสร้างประกอบด้วยเซอร์วิส repayTypeD
7	แบบรูป Deferred Choice	โครงสร้างประกอบด้วยเซอร์วิส reductInterest
8	แบบรูป Exclusive Choice รวมกับ แบบรูป Simple Merge	โครงสร้างประกอบด้วยเซอร์วิส repay ร่วมกับเซอร์วิส EndRepay และเซอร์วิส approval ร่วมกับเซอร์วิส endLone

จากโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติมให้โครงร่างบีเพลทำงานได้ตามภาพที่ 5.4



ภาพที่ 5.4 โครงร่างบีเพลระบบกู้ชำระเงิน



ภาพที่ 5.4 โครงร่างบีเพลระบบกู้ชำระเงิน(ต่อ)

เพื่อที่จะทดสอบความถูกต้องของโครงร่างบีเพลของเครื่องมือที่สร้างขึ้นมา ดังนั้นจึงต้องสร้างกรณีทดสอบสำหรับ ระบบกู้ชำระเงินโดยเปรียบเทียบลำดับการทำงานในกระแสงานยอวล์ทดสอบกับโครงร่างบีเพลตาม ตารางที่ 5.4

ตารางที่ 5.4 กรณีทดสอบกระแสงานยอวล์ระบบกู้ชำระเงิน

รหัส	กรณีทดสอบ	ผลลัพธ์
1	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startLone เลือกเซอร์วิส chkDocument กรณีเอกสารไม่ครบระบบสิ้นสุดการทำงาน	✓
2	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startLone เลือกเซอร์วิส chkDocument กรณีเอกสารครบ เซอร์วิส chkRisk เลือกการทำงาน เซอร์วิส assets เซอร์วิส creditcard เซอร์วิส income เซอร์วิส debt ลำดับการทำงานทั้ง 4 เซอร์วิสสามารถเกิดขึ้นก่อน หลังกันได้โดยไม่เกิดการทำงานพร้อมกัน และเมื่อทำงานครบ 4 เซอร์วิส เรียกเซอร์วิส EndChkRisk เซอร์วิส approval เซอร์วิส approvalNo เซอร์วิส endLone	✓
3	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startLone เลือกเซอร์วิส chkDocument กรณีเอกสารครบ เซอร์วิส chkRisk เลือกการทำงาน เซอร์วิส assets เซอร์วิส creditcard เซอร์วิส income เซอร์วิส debt ลำดับการทำงานทั้ง 4 เซอร์วิสสามารถเกิดขึ้นก่อนหลังได้ โดยไม่เกิดการทำงานพร้อมกัน และเมื่อทำงานครบ 4 เซอร์วิส เรียกเซอร์วิส EndChkRisk เซอร์วิส approval เซอร์วิส approvalYes เซอร์วิส repay กรณีเลือกเซอร์วิส repayTypeA โดยเรียกเซอร์วิสซ้ำตามจำนวน<completionCondition> ตามที่กำหนดในกระแสงานยอวล์ เซอร์วิส EndRepay เกิดการทำงานเซอร์วิส reductinterest ในกรณีชำระเงินเสร็จก่อนที่กำหนด เซอร์วิส endLone	✓
4	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startLone เลือกเซอร์วิส chkDocument กรณีเอกสารครบ เซอร์วิส chkRisk เลือกการทำงาน เซอร์วิส assets เซอร์วิส creditcard เซอร์วิส income เซอร์วิส debt ลำดับการทำงานทั้ง 4 เซอร์วิสสามารถเกิดขึ้นก่อนหลังได้ โดยไม่เกิดการทำงานพร้อมกัน และเมื่อทำงานครบ 4 เซอร์วิส เรียกเซอร์วิส EndChkRisk เซอร์วิส approval เซอร์วิส approvalYes เซอร์วิส repay กรณีเลือกเซอร์วิส repayTypeB โดยเรียกเซอร์วิสซ้ำตามจำนวน instance ที่กำหนดในกระแสงาน เซอร์วิส EndRepay เซอร์วิส endLone	✓

ตารางที่ 5.4 กรณีทดสอบกระแสงานยอวล์ระบบกู้ชำระเงิน(ต่อ)

รหัส	กรณีทดสอบ	ผลลัพธ์
5	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startLone เลือกเซอร์วิส chkDocument กรณีเอกสารครบ เซอร์วิส chkRisk เลือกการทำงานเซอร์วิส assets เซอร์วิส creditcard เซอร์วิส income เซอร์วิส debt ลำดับการทำงานทั้ง 4 เซอร์วิสสามารถเกิดขึ้นก่อนหลังได้ โดยไม่เกิดการทำงานพร้อมกัน และเมื่อทำงานครบ 4 เซอร์วิส เรียกเซอร์วิส EndChkRisk เซอร์วิส approval เซอร์วิส approvalYes เซอร์วิส repay กรณีเลือกเซอร์วิส repayTypeC โดยเรียกเซอร์วิสซ้ำตามจำนวน instance ผู้ใช้กำหนดที่ขั่นตอนการทำงาน เรียกเซอร์วิส EndRepay เซอร์วิส endLone	✓
6	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startLone เลือกเซอร์วิส chkDocument กรณีเอกสารครบ เซอร์วิส chkRisk เลือกการทำงานเซอร์วิส assets เซอร์วิส creditcard เซอร์วิส income เซอร์วิส debt ลำดับการทำงานทั้ง 4 เซอร์วิสสามารถเกิดขึ้นก่อนหลังได้ โดยไม่เกิดการทำงานพร้อมกัน และเมื่อทำงานครบ 4 เซอร์วิส เรียกเซอร์วิส EndChkRisk เซอร์วิส approval เซอร์วิส approvalYes เซอร์วิส repay กรณีเลือกเซอร์วิส repayTypeDโดยผู้ใช้สามารถเพิ่มจำนวนเซอร์วิสซ้ำตามความต้องการตอนปฏิบัติการกระแสงาน แล้วเรียกเซอร์วิส EndRepay เซอร์วิส endLone	✓

5.3.3 ระบบการลงทะเบียนเรียน

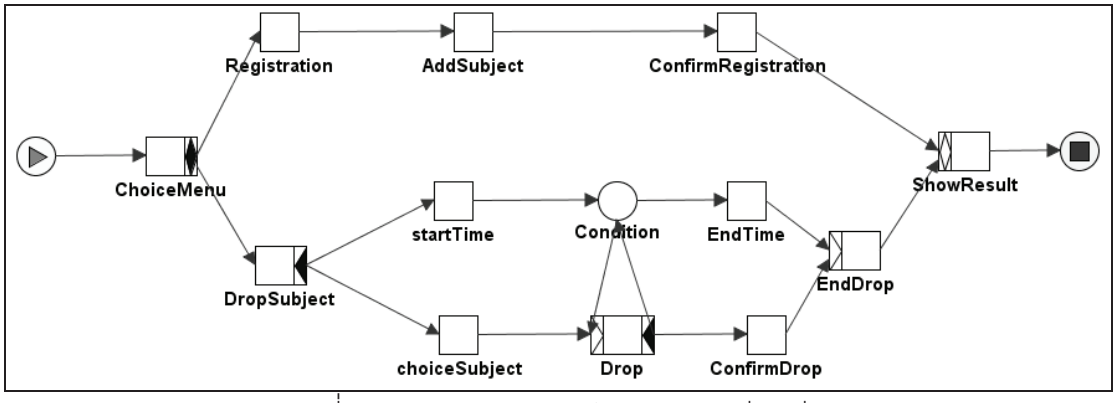
กระแสงานยอวล์ระบบการลงทะเบียนเรียน ระบบสามารถลงทะเบียนและถอนรายวิชาที่ลงทะเบียนเรียน มีลักษณะการทำงานตามแผนภาพกระแสงานยอวล์ ดังภาพที่ 5.5 โดยกระแสงานเริ่มการทำงานจาก เซอร์วิส ChoiceMenu เป็นเซอร์วิสรับการทำงานจากผู้ใช้ โดยแบ่งเป็น 2 กรณี

1) กรณีผู้ใช้ต้องการลงทะเบียนเรียนลำดับการทำงานดังต่อไปนี้ เซอร์วิส Registration ทำการลงทะเบียน เซอร์วิส AddSubject เพิ่มรายวิชาเรียน เซอร์วิส ConfirmRegistration ยืนยันการลงทะเบียนเรียน เซอร์วิส ShowResult และจบการทำงาน

2) กรณีผู้ใช้ต้องการถอนรายวิชาที่ลงทะเบียนเรียนลำดับการทำงานดังต่อไปนี้ เซอร์วิส DropSubject เกิดการทำงานพร้อมกันที่เซอร์วิส startTime และเซอร์วิส choiceSubject เพื่อเลือก

วิชาที่ต้องการถอน โดยเซอร์วิส Drop สามารถทำงานได้กรณีที่ เซอร์วิส EndTime ยังไม่เกิดการทำงาน แล้วเรียกเซอร์วิส ConfirmDrop เซอร์วิส EndDrop เซอร์วิส ShowResult และจบการทำงาน

โดยกระแสงานนี้ประกอบด้วย 4 แบบรูป มีแบบรูป Milestone แบบรูป Sequence และแบบรูป Multi Choice รวมกับแบบรูป Synchronizing Merge ตารางที่ 5.5 แสดงรายละเอียดแบบรูปของระบบลงทะเบียนเรียน

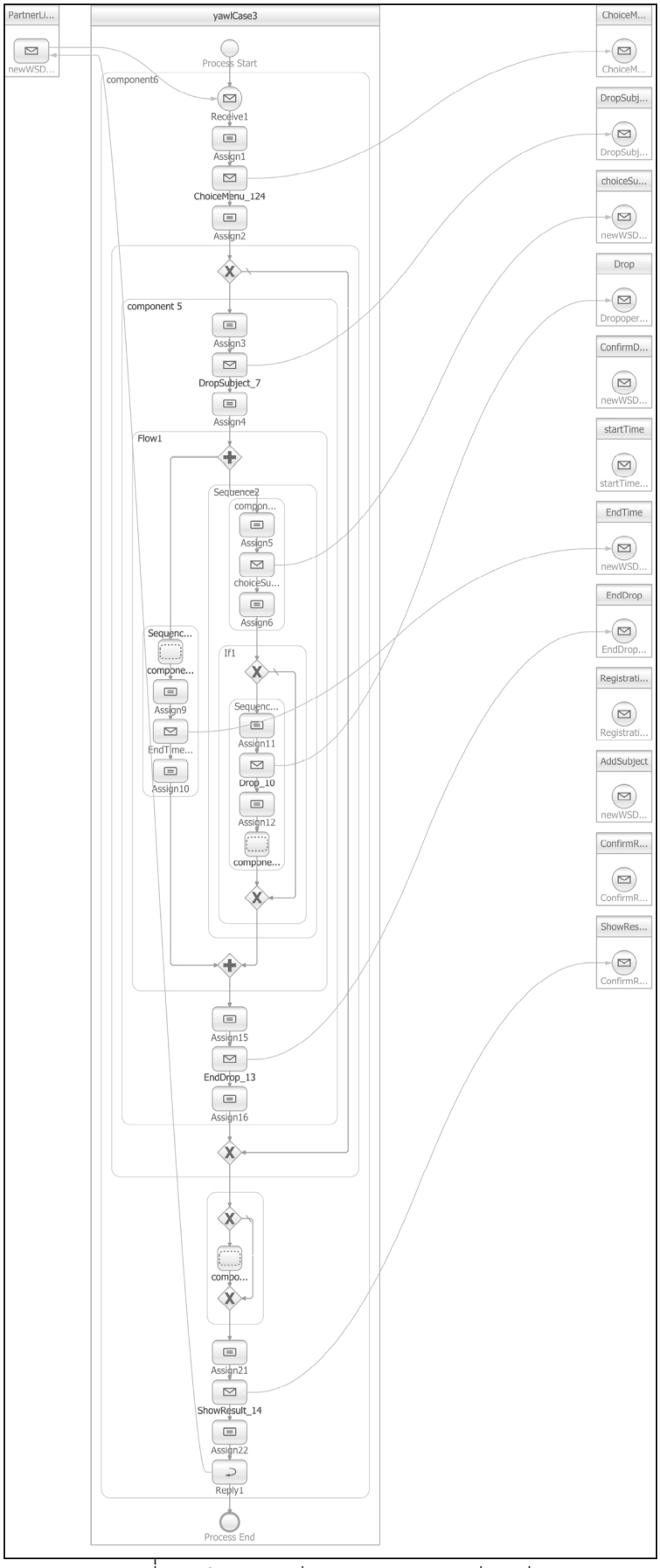


ภาพที่ 5.5 กระแสงานยอวล์ระบบลงทะเบียนเรียน

ตารางที่ 5.5 แสดงแบบรูปในรูประบบลงทะเบียนเรียน

	แบบรูป	**โครงสร้างแบบรูปในกระแสงานยอวล์**
1	แบบรูป Milestone	โครงสร้างประกอบด้วยเซอร์วิส startTime เซอร์วิส choiceSubject เซอร์วิส Drop เซอร์วิส EndTime เซอร์วิส ConfirmDrop และ Condition โหนด
2	แบบรูป Sequence	โครงสร้างประกอบด้วยเซอร์วิส Registration เซอร์วิส AddSubject เซอร์วิส ConfirmRegistration
3	แบบรูป Multi Choice รวมกับแบบรูป Synchronizing Merge	โครงสร้างประกอบด้วยเซอร์วิส ChoiceMenu กับเซอร์วิส ShowResult

จากโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติมให้โครงร่างบีเพลทำงานได้ตามภาพที่ 5.6



ภาพที่ 5.6 โครงร่างบีเพลระบบลงทะเบียนเรียน

เพื่อที่จะทดสอบความถูกต้องของโครงร่างบีเพลที่เครื่องมือสร้างขึ้น ดังนั้นจึงต้องสร้างกรณีทดสอบสำหรับระบบลงทะเบียนเรียน โดยเปรียบเทียบลำดับการทำงานในกระแสงานยอวล์ทดสอบกับโครงร่างบีเพลตาม ตารางที่ 5.6

ตารางที่ 5.6 กรณีทดสอบกระแสงานยอวล์ระบบลงทะเบียนเรียน

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับการทำงานในกระแสงานคือ เซอร์วิส ChoiceMenu เลือก เซอร์วิส Registration กรณีต้องการลงทะเบียนเรียน เซอร์วิส AddSubject เซอร์วิส ConfirmRegistration เซอร์วิส ShowResult จบการทำงาน	✓
2	ลำดับการทำงานในกระแสงานคือ เซอร์วิส ChoiceMenu เลือก เซอร์วิส DropSubject กรณีต้องการถอนรายวิชา เซอร์วิส startTime ทำงานพร้อมเซอร์วิส choiceSubject เกิดการทำงานที่เซอร์วิส Drop กรณีเซอร์วิส EndTime ยังไม่เกิดการทำงาน แล้วเกิดการทำงานที่ เซอร์วิส ConfirmDrop เซอร์วิส EndDrop เซอร์วิส ShowResult จบ การทำงาน	✓
3	ลำดับการทำงานในกระแสงานคือ เซอร์วิส ChoiceMenu เลือก เซอร์วิส DropSubject กรณีต้องการถอนรายวิชา เซอร์วิส startTime เกิดการทำพร้อมเซอร์วิส choiceSubject กรณีเซอร์วิส EndTime ทำงานก่อนเซอร์วิส Drop เซอร์วิส Drop ไม่เกิดการทำงาน แล้วเกิด การทำงานที่ เซอร์วิส EndDrop เซอร์วิส ShowResult จบการทำงาน	✓
4	ลำดับการทำงานในกระแสงานคือ เซอร์วิส ChoiceMenu เลือก เซอร์วิส DropSubject และเซอร์วิส Registration กรณีต้องการถอน รายวิชา เซอร์วิส startTime ทำงานพร้อมเซอร์วิส choiceSubject เกิดการทำงานที่เซอร์วิส Drop กรณีเซอร์วิส EndTime ยังไม่เกิดการ ทำงาน แล้วเกิดการทำงานที่เซอร์วิส ConfirmDrop เซอร์วิส EndDrop เซอร์วิส ShowResult จบการทำงาน	✓

ตารางที่ 5.6 กรณีทดสอบกระแสงานยอวล์ระบบลงทะเบียนเรียน(ต่อ)

5	ลำดับการทำงานในกระแสงานคือ เซอร์วิส ChoiceMenu เลือก เซอร์วิส DropSubject และเซอร์วิส Registration กรณีต้องการถอนรายวิชา เซอร์วิส startTime เกิดการทำพร้อมเซอร์วิส choiceSubject กรณีเซอร์วิส EndTimeทำงานก่อนเซอร์วิส Drop เซอร์วิส Drop ไม่เกิดการทำงาน แล้วเกิดการทำงานที่ เซอร์วิส EndDrop เซอร์วิส ShowResult จบการทำงาน	✓

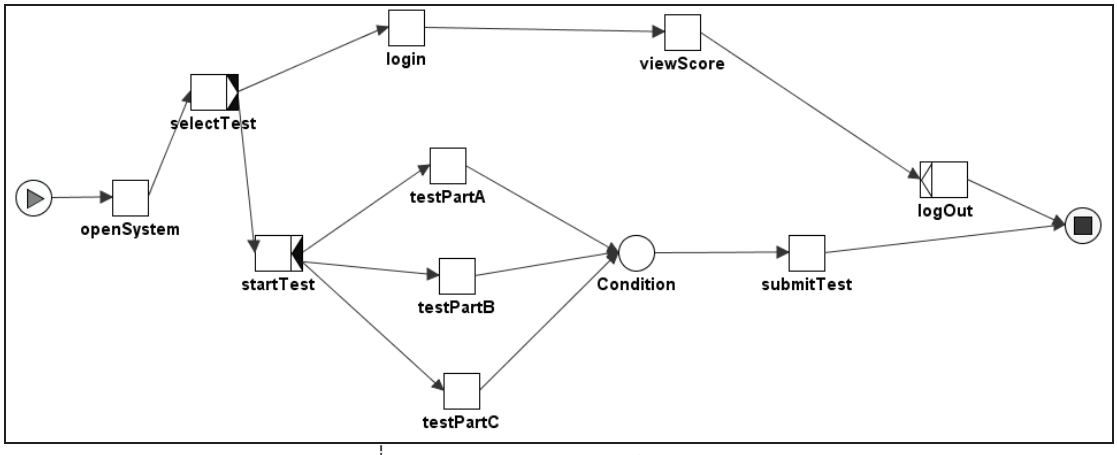
5.3.4 ระบบการสอบ

กระแสงานยอวล์ทดสอบระบบการสอบ ระบบสามารถทำการสอบ และสามารถดูคะแนนการสอบ มีลักษณะการทำงานตามแผนภาพกระแสงานยอวล์ ดังภาพที่ 5.7 โดยกระแสงานเริ่มการทำงานจาก เซอร์วิส openSystem เซอร์วิส selectTest เป็นเซอร์วิสรับการทำงานจากผู้ใช้ โดยแบ่งเป็น 2 กรณี

1) กรณีผู้ใช้ต้องการดูคะแนนการสอบ ต้องเรียกเซอร์วิส login ทำการเข้าสู่ระบบ ระบบแสดงคะแนนโดยเซอร์วิส viewScore เซอร์วิส logOut ออกจากระบบ และจบการทำงาน

2) กรณีผู้ใช้ต้องทำการสอบ เรียกเซอร์วิส startTest เพื่อทำการสอบ จะเกิดการทำงาน เซอร์วิส testPartA เซอร์วิส testPartB เซอร์วิส testPartC ทำงานพร้อมกัน เมื่อเซอร์วิสใดทำงานเสร็จสิ้นจะเรียกเซอร์วิส submitTest เพื่อทำการส่งการสอบ โดยการเรียกเซอร์วิส submit เกิดขึ้น 3 ครั้ง และจบการทำงาน

โดยกระแสงานนี้ประกอบด้วย 4 แบบรูป มีแบบรูป Multi Merge แบบรูป Sequence และแบบรูป Exclusive Choice รวมกับแบบรูป Simple Merge ตาราง 5.7 แสดงรายละเอียดแบบรูปของระบบการสอบ



ภาพที่ 5.7 กระแสงานยอวล์ระบบการสอบ

ตารางที่ 5.7 แสดงแบบรูปในรูประบบลงทะเบียนเรียน

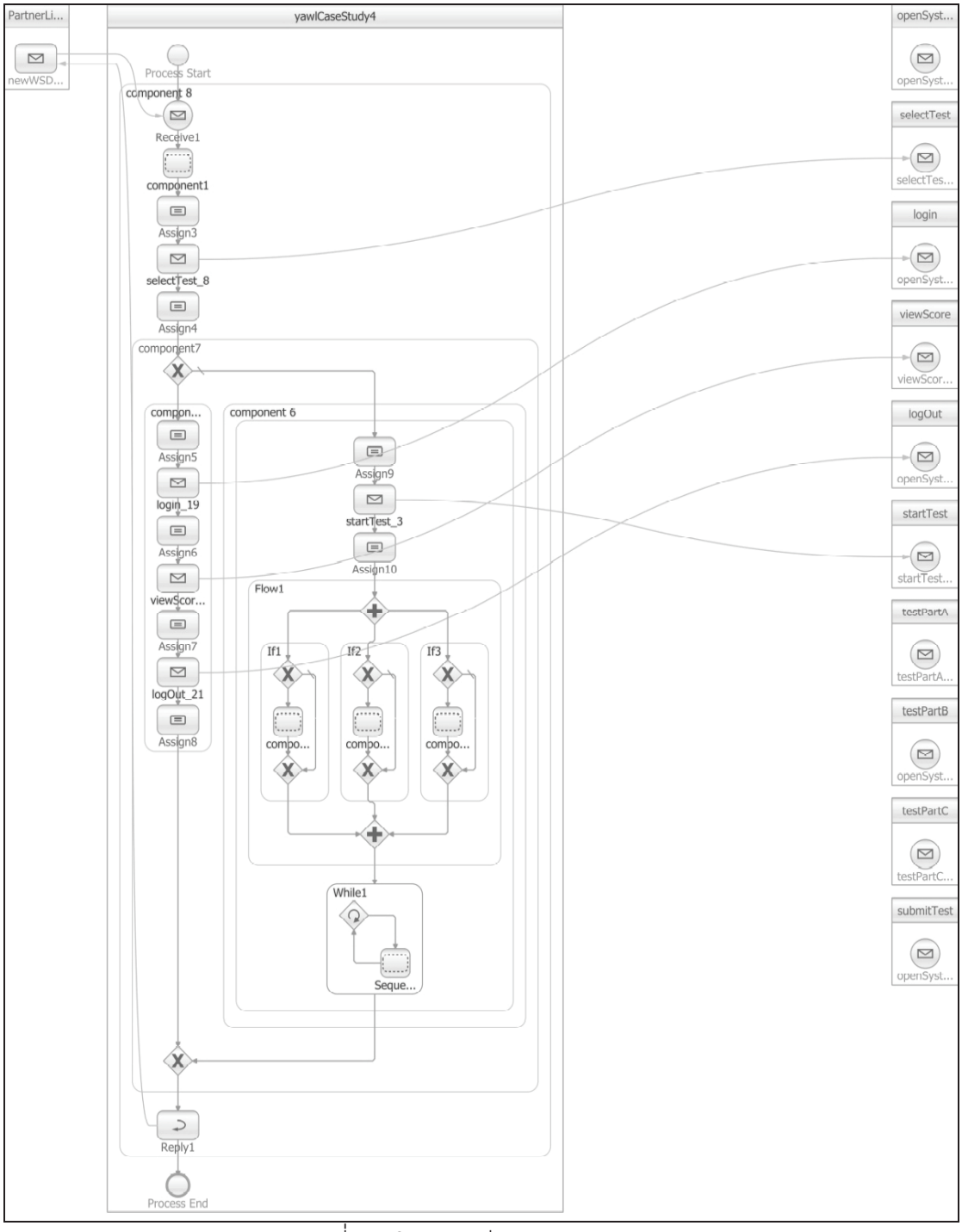
	แบบรูป	**โครงสร้างแบบรูปในกระแสงานยอวล์**
1	แบบรูป Multi Merge	โครงสร้างประกอบด้วยเซอร์วิส startTest เซอร์วิส testPartA เซอร์วิส testPartB เซอร์วิส testPartC เซอร์วิส submitTest และ Condition โหนด
2	แบบรูป Sequence	โครงสร้างประกอบด้วยเซอร์วิส login เซอร์วิส viewScore
3	แบบรูป Exclusive Choice รวมกับแบบรูป Simple Merge	โครงสร้างประกอบด้วยเซอร์วิส selectTest ร่วมกับเซอร์วิส logOut

จากโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติมให้โครงร่างบีเพลทำงานได้ตามภาพที่ 5.8

เพื่อที่จะทดสอบความถูกต้องของโครงร่างบีเพลที่เครื่องมือสร้างขึ้น ดังนั้นจึงต้องสร้างกรณีทดสอบสำหรับ ระบบการสอบโดยเปรียบเทียบลำดับการทำงานในกระแสงานยอวล์ทดสอบกับโครงร่างบีเพลตาม ตารางที่ 5.8

ตารางที่ 5.8 กรณีทดสอบกระแสงานยอวล์ระบบการสอบ

รหัส	**ผลลัพธ์ที่คาดหวัง**	**ผลลัพธ์จริง**
1	ลำดับการทำงานในกระแสงานคือ เซอร์วิส openSystem เซอร์วิส selectTest ในกรณีต้องการดูคะแนน เกิดการทำงานเซอร์วิส login เซอร์วิส viewScore เซอร์วิส logOut และจบการทำงาน	✓
2	ลำดับการทำงานในกระแสงานคือ เซอร์วิส openSystem เซอร์วิส selectTest ในกรณีทำข้อสอบเกิดการทำงานที่เซอร์วิส startTest เกิดการทำงานเซอร์วิส testPartA เซอร์วิส testPartB เซอร์วิส testPartC พร้อมกัน เมื่อเซอร์วิสใดทำงานเสร็จจะเกิดการทำงานที่เซอร์วิส submitTest เมื่อเซอร์วิส submit ทำงานครบ 3 ครั้ง จบการทำงาน	✓



ภาพที่ 5.8 โครงร่างบีเพลระบบการสอบ

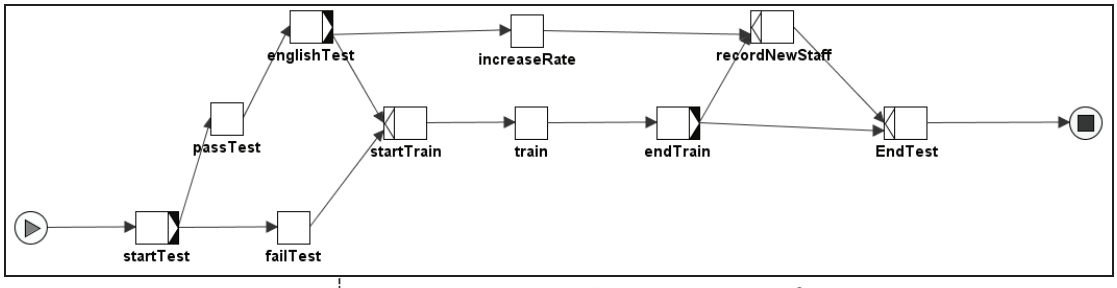
5.3.5 ระบบรับพนักงานใหม่

กระแสงานยอวล์ทดสอบของระบบรับพนักงานใหม่ มีลักษณะการทำงานตามแผนภาพกระแสงานยอวล์ ดังภาพที่ 5.9 โดยกระแสงานเริ่มการทำงานจาก เซอร์วิส startTest เพื่อทำการสอบผู้สมัครโดยแบ่งเป็น 3 กรณี

1) กรณีผู้สมัครผ่านการทดสอบ เกิดการทำงานที่เซอร์วิส passTest เซอร์วิส englishTest ทำการทดสอบความรู้ภาษาอังกฤษ ในกรณีผู้ทดสอบผ่านการทดสอบ เซอร์วิส increaseRate จะเกิดการทำงานเพื่อปรับอัตราเงินเดือนเริ่มต้น แล้วเซอร์วิส recordNewStaff บันทึกข้อมูลพนักงานใหม่ เซอร์วิส EndTest และจบการทำงาน

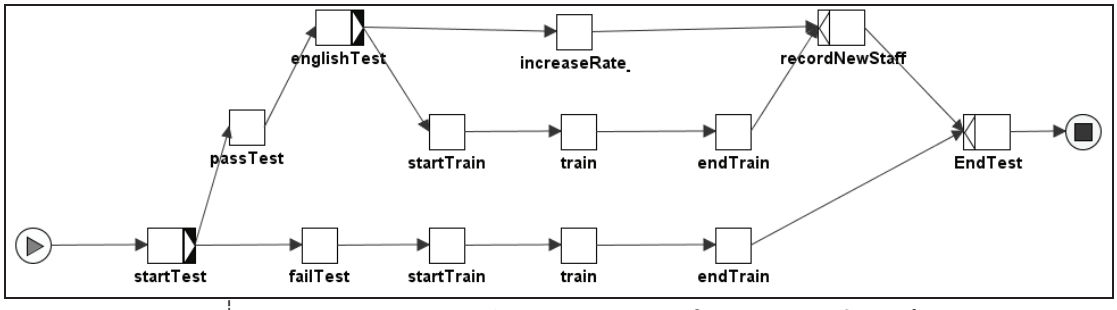
2) กรณีผู้สมัครผ่านการทดสอบ แต่ไม่ผ่านการทดสอบความรู้ภาษาอังกฤษ จะเกิดการทำงานที่เซอร์วิส startTrain เพื่อเริ่มการอบรม เซอร์วิส train ทำการอบรม เซอร์วิส endTrain สิ้นสุดการอบรม เซอร์วิส recordNewStaff เซอร์วิส EndTest และจบการทำงาน

3) กรณีผู้สมัครไม่ผ่านการทดสอบ เกิดการทำงานที่เซอร์วิส failTest เซอร์วิส startTrain เซอร์วิส train เซอร์วิส endTrain เซอร์วิส EndTest และจบการทำงาน



ภาพที่ 5.9 กระแสงานยอวล์ระบบรับพนักงานใหม่

สำหรับกระแสงานยอวล์ระบบรับพนักงานใหม่ที่ใช้ทดสอบเครื่องมือการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพลไม่สามารถเปลี่ยนได้ เพราะกระแสงานนี้มีลักษณะเป็นโครงสร้างที่ไม่ดี ระบบจึงทำการปรับโครงสร้างกระแสงานยอวล์ทดสอบใหม่โดยใช้อัลกอริทึมแฮมมอค โดยสามารถดูกระแสงานยอวล์หลังปรับโครงสร้างได้ตามภาพที่ 5.10



ภาพที่ 5.10 กระแสงานยอวล์ระบบรับพนักงานใหม่หลังปรับโครงสร้าง

โดยกระแสงานนี้ประกอบด้วย 4 แบบรูป มีแบบรูป Arbitrary Cycle แบบรูป Sequence และ แบบรูป Exclusive Choice รวมกับแบบรูป Simple Merge ตารางที่ 5.9 แสดงรายละเอียดแบบรูปของระบบรับพนักงานใหม่

ตารางที่ 5.9 แสดงแบบรูปในรูประบบรับพนักงานใหม่

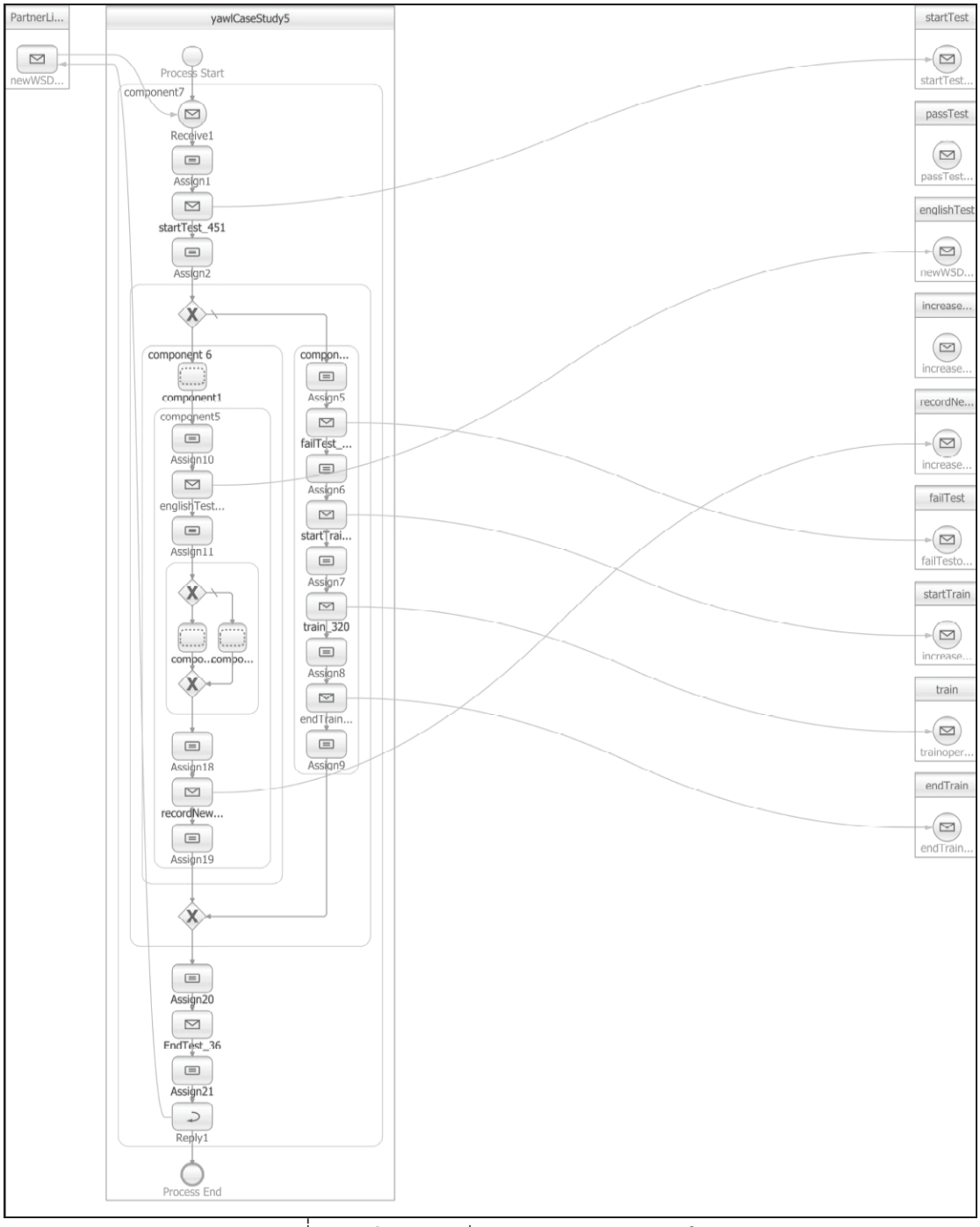
	แบบรูป	**โครงสร้างแบบรูปในกระแสงานยอวล์**
1	แบบรูป Arbitrary Cycle	โครงสร้างประกอบด้วยเซอร์วิส startTest เซอร์วิส testPartA เซอร์วิส testPartB เซอร์วิส testPartC เซอร์วิส submitTest และ Condition โหนด
2	แบบรูป Sequence	โครงสร้างประกอบด้วยเซอร์วิส login เซอร์วิส viewScore
3	แบบรูป Exclusive Choice รวมกับแบบรูป Simple Merge	โครงสร้างประกอบด้วยเซอร์วิส selectTest ร่วมกับเซอร์วิส logOut

จากโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติมให้โครงร่างบีเพลสามารถทำงานได้ตามภาพที่ 5.11

เพื่อที่จะทดสอบความถูกต้องของโครงร่างบีเพลที่เครื่องมือสร้างขึ้น ดังนั้นจึงต้องสร้างกรณีทดสอบสำหรับ ระบบรับพนักงานใหม่โดยเปรียบเทียบลำดับการทำงานในกระแสงานยอวล์ทดสอบกับโครงร่างบีเพลตาม ตารางที่ 5.10

ตารางที่ 5.10 กรณีทดสอบกระแสงานยอวล์ระบบรับพนักงานใหม่

รหัส	**ผลลัพธ์ที่คาดหวัง**	**ผลลัพธ์จริง**
1	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startTest กรณีไม่ผ่านการทดสอบ เกิดการทำงานเซอร์วิสเซอร์วิส failTest เซอร์วิส startTest เซอร์วิส train เซอร์วิส endTrain และจบการทำงาน	✓
2	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startTest กรณีผ่านการทดสอบ เกิดการทำงานเซอร์วิส passTest เซอร์วิส englishTest ในกรณีไม่ผ่านการทดสอบ เกิดการทำงานที่เซอร์วิส startTrain เซอร์วิส train เซอร์วิส endTrain เซอร์วิส recordNewStaff เซอร์วิส EndTest และจบการทำงาน	✓
3	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startTest กรณีผ่านการทดสอบ เกิดการทำงานเซอร์วิส passTest เซอร์วิส englishTest ในกรณีผ่านการทดสอบ เกิดการทำงานที่เซอร์วิส increaseRate เซอร์วิส recordNewStaff เซอร์วิส EndTest และจบการทำงาน	✓



ภาพที่ 5.11 โครงร่างบีเพลระบบรับพนักงานใหม่

5.3.6 ระบบสั่งซื้อหนังสือ

กระแสงานยอวล์ทดสอบของระบบสั่งหนังสือ เป็นระบบทำการสมัครสมาชิกหนังสือรายปี มีลักษณะการทำงานตามแผนภาพกระแสงานยอวล์ ดังภาพที่ 5.12 โดยกระแสงานเริ่มการทำงานจาก

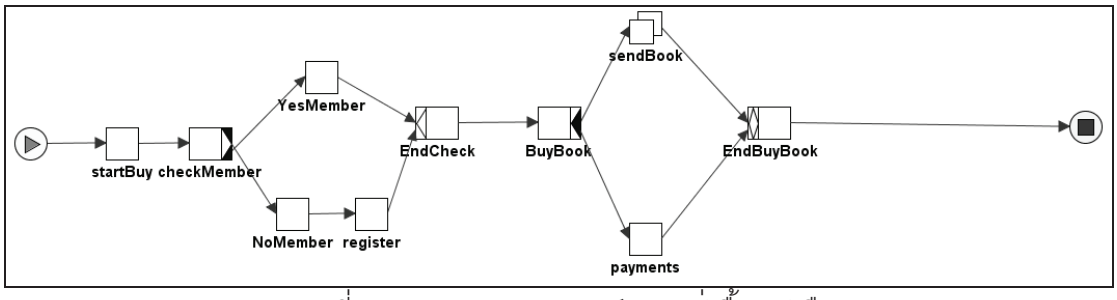
1) เซอร์วิส startBuy เซอร์วิส checkMember ทำการตรวจสอบการสมัคร

2) กรณีไม่เป็นสมาชิก เกิดการทำงานที่เซอร์วิส NoMember เซอร์วิส register เพื่อทำการสมัครสมาชิก แล้วเรียกเซอร์วิส EndCheck

3) กรณีเป็นสมาชิก เกิดการทำงานที่เซอร์วิส YesMember เซอร์วิส EndCheck

4) หลังจากนั้นเกิดการทำงานที่เซอร์วิส BuyBook การทำงานเซอร์วิส sendBook และเซอร์วิส payments ทำงานขนานกันแต่ไม่เกิดการทำงานพร้อมกัน โดยเซอร์วิส sendBook มีลักษณะเป็น [1,n,n,static] สามารถเกิดการทำงานเซอร์วิสซ้ำ จำนวนการวนซ้ำกำหนดที่ขั้นตอนการออกแบบ

5) เมื่อเซอร์วิส sendBook และเซอร์วิส payments ทำงานเสร็จ เซอร์วิส EndBuyBook จะเกิดการทำงานและกระแสจบการทำงาน



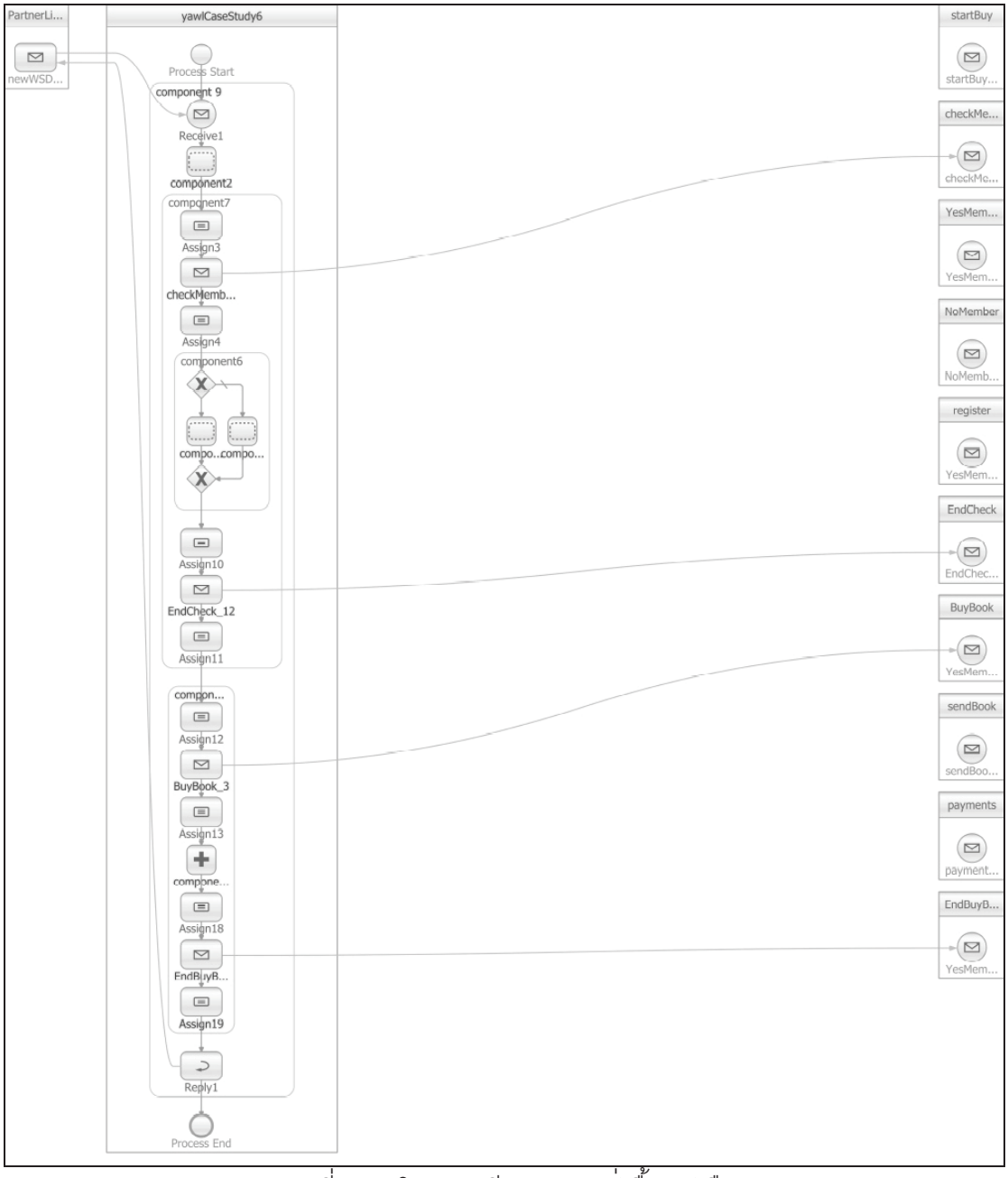
ภาพที่ 5.12 กระแสงานยอวล์ระบบสั่งซื้อหนังสือ

โดยกระแสงานนี้ประกอบด้วย 4 แบบรูป มีแบบรูป MI without Synchronization แบบรูป Sequence และ แบบรูป Exclusive Choice รวมกับแบบรูป Simple Merge ตารางที่ 5.11 แสดงรายละเอียดแบบรูปของระบบรับพนักงานใหม่

ตารางที่ 5.11 แสดงแบบรูปในรูประบบสั่งซื้อหนังสือ

	แบบรูป	**โครงสร้างแบบรูปในกระแสงานยอวล์**
1	แบบรูป MI without Synchronization	โครงสร้างประกอบด้วยเซอร์วิส BuyBook เซอร์วิส sendBook เซอร์วิส payments เซอร์วิส EndBuyBook
2	แบบรูป Sequence	โครงสร้างประกอบด้วยเซอร์วิส NoMember เซอร์วิส register
3	แบบรูป Exclusive Choice รวมกับแบบรูป Simple Merge	โครงสร้างประกอบด้วยเซอร์วิส checkMember ร่วมกับเซอร์วิส EndCheck

จากโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติมให้โครงร่างบีเพลสามารถทำงานได้ตามภาพที่ 5.13



ภาพที่ 5.13 โครงร่างบีเพลระบบสั่งซื้อหนังสือ

เพื่อที่จะทดสอบความถูกต้องของโครงร่างบีเพลที่เครื่องมือสร้างขึ้น ดังนั้นจึงต้องสร้างกรณีทดสอบสำหรับ ระบบรับพนักงานใหม่โดยเปรียบเทียบลำดับการทำงานในกระแสงานยอวล์ทดสอบกับโครงร่างบีเพลตาม ตารางที่ 5.12

ตารางที่ 5.12 กรณีทดสอบกระแสงานยอวล์ระบบสั่งซื้อหนังสือ

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startBuy เซอร์วิส checkMember กรณีเป็นสมาชิกเกิดการทำงานที่เซอร์วิส YesMember เซอร์วิส EndCheck เซอร์วิส BuyBook ที่เซอร์วิส sendBook เกิดการทำงานวนซ้ำ โดยทำงานขนานกับเซอร์วิส payments เซอร์วิส EndBuyBook และจบการทำงาน	✓
2	ลำดับการทำงานในกระแสงานคือ เซอร์วิส startBuy เซอร์วิส checkMember กรณีไม่เป็นสมาชิกเกิดการทำงานที่เซอร์วิส NoMember เซอร์วิส register เซอร์วิส EndCheck เซอร์วิส BuyBook ที่เซอร์วิส sendBook เกิดการทำงานวนซ้ำ โดยทำงานขนานกับเซอร์วิส payments เซอร์วิส EndBuyBook และจบการทำงาน	✓

หมายเหตุ ภาพผลลัพธ์ของการเปลี่ยนรูปกระแสงานยอวล์เป็นโครงร่างบีเพลทีเครื่องมือสร้างดูได้ที่ภาคผนวก ข

5.4 **สรุปผลการทดสอบ**

หลังจากทำการทดสอบเครื่องมือโดยใช้กระแสงานยอวล์ทดสอบ 6 ระบบกระแสงาน โดยให้เครื่องมือการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพล เปรียบเทียบลำดับการทำงานของกระแสงานยอวล์และโครงร่างบีเพลตามกรณีทดสอบในตารางที่ 5.2 5.4 5.6 5.8 5.10 5.12 ผลการทดสอบสามารถเกิดการทำงานได้ตรงตามกรณีทดสอบที่ได้ออกแบบทั้งหมด นอกจากนี้เครื่องมือยังสามารถปรับปรุงโครงสร้างกระแสงานยอวล์ กรณีกระแสงานที่มีแบบรูปโครงสร้างไม่ดีได้อย่างถูกต้อง

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ได้นำเสนอแนวคิดการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพลโดยแนวคิดในวิทยานิพนธ์นี้รองรับกระบวนการเปลี่ยนแบบรูปในภาษายอวล์ทั้ง 19 แบบรูป โดยแบ่งได้ 2 ประเภทคือ

1) แบบรูปโครงสร้างที่ดี คือ แบบรูปของกระแสงานยอวล์ที่สามารถเปลี่ยนเป็นโครงร่างบีเพลได้ โดยไม่จำเป็นต้องปรับโครงสร้างกระแสงานยอวล์เพิ่มเติม ในวิทยานิพนธ์นี้กำหนดไว้ 18 แบบรูป แบ่งได้ดังนี้

- แบบรูปจากแนวคิดของ C.OuYang [7] มี 11 แบบรูป คือแบบรูป Sequence แบบรูป Parallel Split แบบรูป Synchronization แบบรูป Exclusive Choice แบบรูป Simple Merge แบบรูป Multi Choice แบบรูป Discriminator แบบรูป Synchronizing Merge แบบรูป MI with a Priori Design-Time knowledge แบบรูป MI with a Priori Run-Time knowledge และแบบรูป Cancel Case แสดงรายละเอียดในตาราง 2.2

- แบบรูปที่วิทยานิพนธ์นี้ออกแบบเพิ่มเติม คือแบบรูปนอกเหนือจาก [7] นำเสนอ มี 4 แบบรูป คือแบบรูป Multi Merge แบบรูป MI without a Priori Run-Time Knowledge แบบรูป Interleaved Parallel Routing และแบบรูป Milestone

- แบบรูปที่วิทยานิพนธ์นำมาปรับปรุง มี 3 แบบรูปคือแบบรูป MI without Synchronization แบบรูป Deferred Choice และแบบรูป Cancel Activity ทั้ง 3 แบบรูปนี้เป็นแบบรูปที่ไม่สามารถเปลี่ยนเป็นโครงร่างบีเพลได้อย่างถูกต้องตามแนวคิดใน [7] ดังนั้นผู้ทำวิทยานิพนธ์จึงนำเสนอวิธีการเปลี่ยนเป็นโครงร่างบีเพลที่มีความเหมาะสมแทน [7]

2) แบบรูปโครงสร้างที่ไม่ดี คือ แบบของกระแสงานยอวล์ ไม่สามารถเปลี่ยนโครงร่างบีเพลได้จำเป็นต้องปรับโครงสร้าง มี 1 แบบรูปคือ แบบรูป Arbitrary Cycle จึงทำการปรับปรุงโครงสร้างให้เป็นกระแสงานโครงสร้างที่ดี โดยใช้แนวคิดอัลกอริทึมแฮมมอคแล้วจึงทำการเปลี่ยนรูปกระแสงานยอวล์เป็นโครงร่างบีเพล

เพื่อที่จะทดสอบเครื่องมือการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพลในวิทยานิพนธ์นี้ได้สร้างกระแสงานยอวล์ทดสอบประกอบไปด้วย 6 กระแสงานทดสอบ คือระบบเช่ายืมหนังสือ ระบบกู้ชำระเงิน ระบบการลงทะเบียนเรียน ระบบการสอบ ระบบรับพนักงานใหม่ และระบบสั่งซื้อหนังสือ โดยการทดสอบครอบคลุมลักษณะแบบรูปทั้ง 19 แบบรูป แล้วใช้เครื่องมือเปลี่ยนกระแสงานยอวล์ทดสอบเป็นโครงร่างบีเพล ใส่รายละเอียดให้โครงร่างบีเพลสามารถทำงานได้

แล้วเปรียบเทียบการทำงานระหว่างกระแสงานยอวล์ทดสอบ กับโครงร่างบีเพลที่เครื่องมือสร้างขึ้น พบว่าผลลัพธ์การทำงานได้อย่างถูกต้องตรงตามทุกกรณีทดสอบที่ออกแบบ

6.2 ข้อจำกัดของงานวิจัย

เครื่องมือการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพลที่สร้างขึ้นจากวิทยานิพนธ์นี้ โครงร่างบีเพลถูกออกแบบตามมาตรฐาน Web Services Business Process Execution Language Version 2.0 OASIS Standard และทำงานได้ถูกต้องบนเน็ตบีนส์ไอดีอี เวอร์ชั่น 6.7.1

6.3 ข้อเสนอแนะ

วิทยานิพนธ์นี้ออกแบบอัลกอริทึมในการเปลี่ยนรูป โดยตรวจสอบจากโครงสร้างของกระแสงานยอวล์ตามแบบรูปที่กำหนดเท่านั้น ดังนั้นควรปรับปรุงอัลกอริทึมให้สามารถตรวจสอบลักษณะการทำงานด้านอรรถศาสตร์เชิงกิจจะลักษณะ (formal semantics)ของกระแสงานยอวล์เพื่อความถูกต้องในการเปลี่ยนรูปมากขึ้น

6.3 ประโยชน์ที่ได้รับ

สามารถนำแนวคิดการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างภาษาบีเพลไปประยุกต์ใช้ในการเปลี่ยนกระแสงานอื่นๆได้ และเครื่องมือที่ได้ในวิทยานิพนธ์นี้ ทำให้ผู้ที่วิเคราะห์กระบวนการทางธุรกิจ สามารถใช้เป็นเครื่องมือในการนำเสนอกระบวนการ และช่วยวิเคราะห์ได้โดยการนำเสนอกระบวนการทางธุรกิจในรูปของกระแสงานยอวล์โดยไม่ต้องลงรายละเอียดในการพัฒนา และช่วยนักพัฒนา พัฒนาระบบเว็บเซอร์วิส จากกระบวนการทางธุรกิจในรูปกระแสงานยอวล์ ให้อยู่ในรูปโครงร่างภาษาบีเพล

รายการอ้างอิง

[1] T. Numnonda and T. Kruawaisayawan. Development Business Process for SOA. [Online]. 2009.Available form: http://www.thaijavadev.com/soa/articles/BPM.pdf [2009, July]

[2] W. van der Aalst and T. Hofstede. YAWL: Yet Another Workflow Language., Information Systems- vol. 30, pp. 245-275., 2005.

[3] C. Ouyang, M. Dumas, A.H.M.T. Hofstede, and W.M.P.V.D. Aalst. From BPMN Process Models to BPEL Web Services., Proceedings of the IEEE International Conference on Web Services, IEEE Computer Society, pp. 285-292., 2006.

[4] J. Ye, S. Sun, L. Wen, and W. Song. Transformation of BPMN to YAWL., Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Volume 02, IEEE Computer Society, pp. 354-359., 2008.

[5] F. Zhang and E.H. D'Hollander. Using Hammock Graphs to Structure Programs., IEEE Transactions on Software Engineering, vol. 30, pp. 231-245.,2004.

[6] Hutton, Graham. A tutorial on the universality and expressiveness of fold., Journal of Functional Programming (9 (4)), pp. 355–372., 1999.

[7] A. H.M. ter Hofstede, W.M. van der Aalst, M. Adams, and N. Russell. Modern Business Process Automation YAWL and its Support Environment. Springer, 2009.

[8] Jordan, D., and Alves, A. Web Services Business Process Execution Language Version 2.0. [Online].2007. Available from: http://www.citeulike.org/group/2540/article/1233272 [2007, July]

[9] W.M.P.V.D. Aalst, A.H.M.T. Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns., Distributed and Parallel Databases Volume 14, pp. 5-51., 2003.

[10] Arthur ter Hofstede. YAWL Leading the World in Process Innovation. [Online]. 2001. Avaliable from: http://www.yawlfoundation.org/pages/resources/patterns.html [2001]

[11] Oliver Kopp, Daniel Martin, Daniel Wutke, and Frank Leymann. On the Choice Between Graph-Based and Block-Structured Business Process Modeling Languages., MobIS, Vol. 141GI, pp. 59-72., 2008.

[12] Ouyang, Chun, Dumas, Marlon, ter Hofstede, Arthur H.M., and van der Aalst, Wil M.P. Pattern-based translation of BPMN process models to BPEL web services., International Journal of Web Services Research (JWSR), pp. 42-62.,2007.

[13] J. Ye, S. Sun, W. Song, and L. Wen. Formal Semantics of BPMN Process Models Using YAWL., Second International Symposium on Intelligent Information Technology Application, pp. 70-74., 2008.

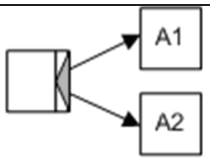
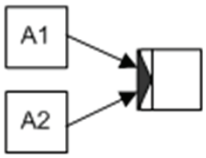
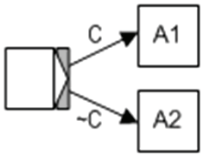
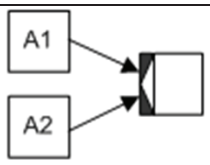
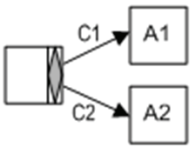
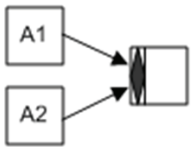
[14] Lijie Wen ,Wil M. P. van der Aalst, Jianmin Wang and Jiaguang Sun.Mining process models with non-free-choice constructs., Data Min. Knowl. Disco, , pp. 145-180., 2007.

ภาคผนวก

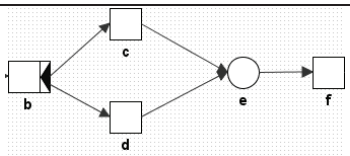
ภาคผนวก ก
สรุปแบบรูปในกระแสงานยอวล์เป็นโครงร่างบีเพล

แสดงลักษณะแบบรูปในกระแสงานยอวล์เปลี่ยนเป็นโครงร่างบีเพล ทั้ง 19 แบบรูป โดยผู้ทำวิทยานิพนธ์ได้ปรับปรุงการเปลี่ยนโครงร่างบีเพลจาก ตาราง 2.2 โดยปรับปรุงโครงร่างบีเพลตามลักษณะการทำงานของกระแสยอวล์ตามตารางที่ ก-1

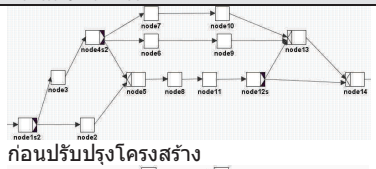
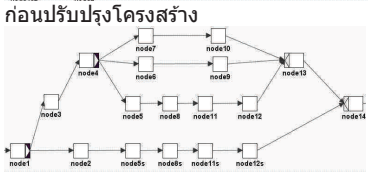
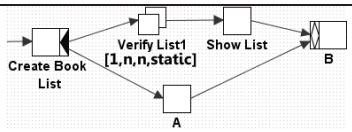
ตารางที่ ก-1 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
1	Sequence		<sequence> <invoke name = "A"/> <invoke name = "B"/> </sequence>
2	Parallel Split		<flow> <sequence> <invoke name = "A1"/> </sequence> <sequence> <invoke name = "A2"/> </sequence> </flow>
3	Synchronization		
4	Exclusive Choice		<if> <condition> C </condition> <sequence> <invoke name = "A1"/> </sequence> <else> <sequence> <invoke name = "A2"/> </sequence> </else> </if>
5	Simple Merge		
6	Multi Choice		<flow> <links> <link name="SplitToA1"/> <link name="SplitToA2"/> <link name="A1ToMerge"/> <link name="A2ToMerge"/> </links> <empty name="Split"> <sources> <source linkName="SplitToA1"> <transitionCondition>C1</transitionCondition> </source> <source linkName="SplitToA2"> <transitionCondition>C2</transitionCondition> </source> </sources> </empty> <sequence> <invoke name = "A1"/> </sequence> <targets><target linkName="SplitToA1"/></targets> <sources> <source linkName="A1ToMerge"/>
7	Synchronizing Merge		

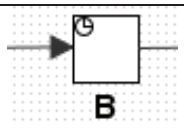
ตารางที่ ก-1 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล(ต่อ)

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
			</sources> <sequence> <invoke name = "A2"/> </sequence> <targets> <target linkName="SplitToA2"/> </targets> <sources> <source linkName="A2ToMerge"/> </sources> <empty name="Merge"> <targets> <joinCondition> $A1ToMerge OR $A2ToMerge </joinCondition> <target linkName="A1ToMerge"/> <target linkName="A2ToMerge"/> </targets> </empty> </flow>
8	Multi Merge		<sequence> <variable name="counter" type="int" /> <variable name="counterLoop" type="int" /> <variable name="inputA" type="String" /> <variable name="inputB" type="String" /> $counter=0 $counterLoop=0 <sequence> <invoke name="b"/> <flow> <if> <condition>inputA==true</condition> <sequence > <invoke name = "c"/> </sequence> $counter++ </if> <if> <condition>inputB==true</condition> <sequence> <invoke name = "d"/> </sequence> $counter++ </if> </flow> <while> <condition>$counterLoop<=$counter</condition> <invoke name="f"/> $counterLoop++ </while> </sequence>
9	Discriminator	[1, 10, 1, static] A	<forEach name = "component 1" counterName="i" parallel="yes"> <startCounterValue>1</startCounterValue> <finalCounterValue>10</finalCounterValue> <completionCondition> <branches>1</branches> </completionCondition> <scope> <invoke name="A"/> </scope> </forEach>

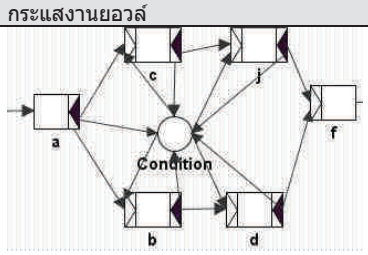
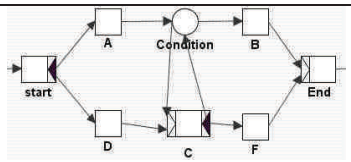
ตารางที่ ก-1 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล(ต่อ)

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
10	Arbitrary Cycle	 ก่อนปรับปรุงโครงสร้าง  หลังปรับปรุงโครงสร้าง	<sequence> <invoke name="node1"/> <if> <condition>C1</condition> <sequence> <sequence> <invoke name = "node3"/> </sequence> <sequence> <invoke name="node4"/> <if> <condition>C2</condition> <sequence> <invoke name = "node7"/> <invoke name = "node10"/> </sequence> <else> <if> <condition> C3 </condition> <sequence> <invoke name = "node6"/> <invoke name = "node9"/> </sequence> <else> <sequence> <invoke name = "node5"/> <invoke name = "node8"/> <invoke name = "node11"/> <invoke name = "node12"/> </sequence> </else> </if> </else> </if> <invoke name="node13"/> </sequence> </sequence> <else> <sequence> <invoke name = "node2"/> <invoke name = "node5s"/> <invoke name = "node8s"/> <invoke name = "node11s"/> <invoke name = "node12s"/> </sequence> </else> </if> <invoke name="node14"/> </sequence>
11	Implicit Termination	ไม่รองรับ	ไม่รองรับ
12	MI without Synchronization	 	<sequence> <invoke name = "Create_Book_List"/> <flow> <sequence > <invoke name = "A"/> </sequence> <sequence name= "component 2"> <forEach name = "component 1" counterName="i" parallel="yes"> <startCounterValue>1</startCounterValue> <finalCounterValue>10</finalCounterValue> <scope> <invoke name="Verify_List1"/> </scope> </forEach> <invoke name = "Show_List"/>

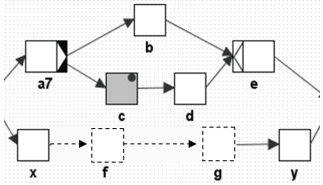
ตารางที่ ก-1 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล(ต่อ)

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
			</sequence> </flow> <invoke name = "B"/> </sequence>
13	MI with a Priori Design-Time knowledge	[1, 10, 10, static] 	<forEach counterName="i" parallel="yes"> <startCounterValue>1</startCounterValue> <finalCounterValue>10</finalCounterValue> <scope> <invoke name="A"/> </scope> </forEach>
14	MI with a Priori Run-Time knowledge	[1, inf, inf, static] 	<forEach counterName="i" parallel="yes"> <startCounterValue>1</startCounterValue> <finalCounterValue> $x </finalCounterValue> <scope> <invoke name="A"/> $x++ </scope> </forEach>
15	MI without a Priori Run-Time Knowledge	[1, inf, inf, dynamic] 	<scope> <variable name="i" type="boolean" /> <variable name="counter" type="int" /> <sequence> <receive variable="quantity" /> $i=true <while> <condition> $i=true </condition> $counter=0 <sequence> <pick> <onMessage variable="input"/> </pick> <if> <condition>input==true</condition> <while> <condition> $counter<=quantity </condition> <invoke name="A"/> $counter++ </while> <else> $i=false </else> 3</if> </sequence> </while> </sequence> </scope>
16	Deferred Choice		…… <if> <condition>sxxf:dateTime-less-than(sxxf:current-dateTime(),'2011-10-02T14:58:51.99+07:00')</condition> <sequence> <invoke name="B"/> </sequence> </if> ……..

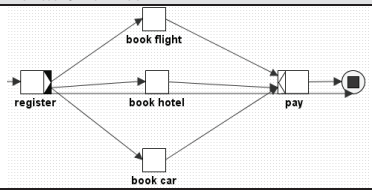
ตารางที่ ก-1 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล(ต่อ)

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
17	Interleaved Parallel Routing		<sequence> <scope> <sequence> <variables> <variable name= "$valb_5" type="xs:boolean"/> <variable name= "$vald_6" type="xs:boolean"/> <variable name= "$valc_4" type="xs:boolean"/> <variable name= "$valj_10" type="xs:boolean"/> </variables> $valb_5=true $vald_6=true $valc_4=true $valj_10=true <invoke name = "a"/> <while> <condition>$count<4</condition> <sequence> <receive name="Receive2" variable="input"/> <flow> <if> <condition>$input=="b_5" && $valb_5==true </condition> <invoke name="b"/> $valb_5=false $count++ </if> <if> <condition>$input=="d_6" && $vald_6==true && $valb_5==false</condition> <invoke name="d"/> $vald_6=false $count++ </if> <if> <condition>$input=="c_4" && $valc_4==true </condition> <invoke name="c"/> $valc_4=false $count++ </if> <if> <condition>$input=="j_10" && $valj_10==true && $valc_4==false</condition> <invoke name="j"/> $valj_10=false $count++ </if> </flow> </sequence> </while> <invoke name="f"/> </sequence> </scope> <sequence> <invoke name = "i"/> </sequence> </sequence>
18	Milestone		<variables> <variable name= "$i" type="xs:boolean"/> </variables> $i=false <sequence > <invoke name="start"/> <flow> <sequence> $i=true <sequence> <invoke name = "A"/>

ตารางที่ ก-1 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล(ต่อ)

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
			</sequence> $i=false <invoke name="B"/> </sequence> <sequence> <sequence> <invoke name = "D"/> </sequence> <if> <condition>$i==true </condition> <sequence> <invoke name="C"/> <sequence> <invoke name = "F"/> </sequence> </sequence> </if> </sequence> </flow> <invoke name="End"/> </sequence>
19	Cancel Activity		<variables> <variable name= "$i" type="xs:boolean"/> <variable name= "$j" type="xs:boolean"/> </variables> $i=true $j=true ..... <flow> <sequence> <invoke name="a"/> <if name> <invoke name="b"/> <else> <sequence> <invoke name="c"/> <if> <condition>$j==false</condition> $i=false </if> <invoke name="d"/> </sequence> </else> </if> <invoke name="e"/> </sequence> <sequence name="FlowSequence"> <invoke name="x"/> $j=false <if> <condition>$i==true</condition> <invoke name="f"/> </if> <if> <condition>$i==true</condition> <invoke name="g"/> </if> <if> <condition>$i==true</condition> <invoke name="y"/> </if> </sequence> </flow> .....

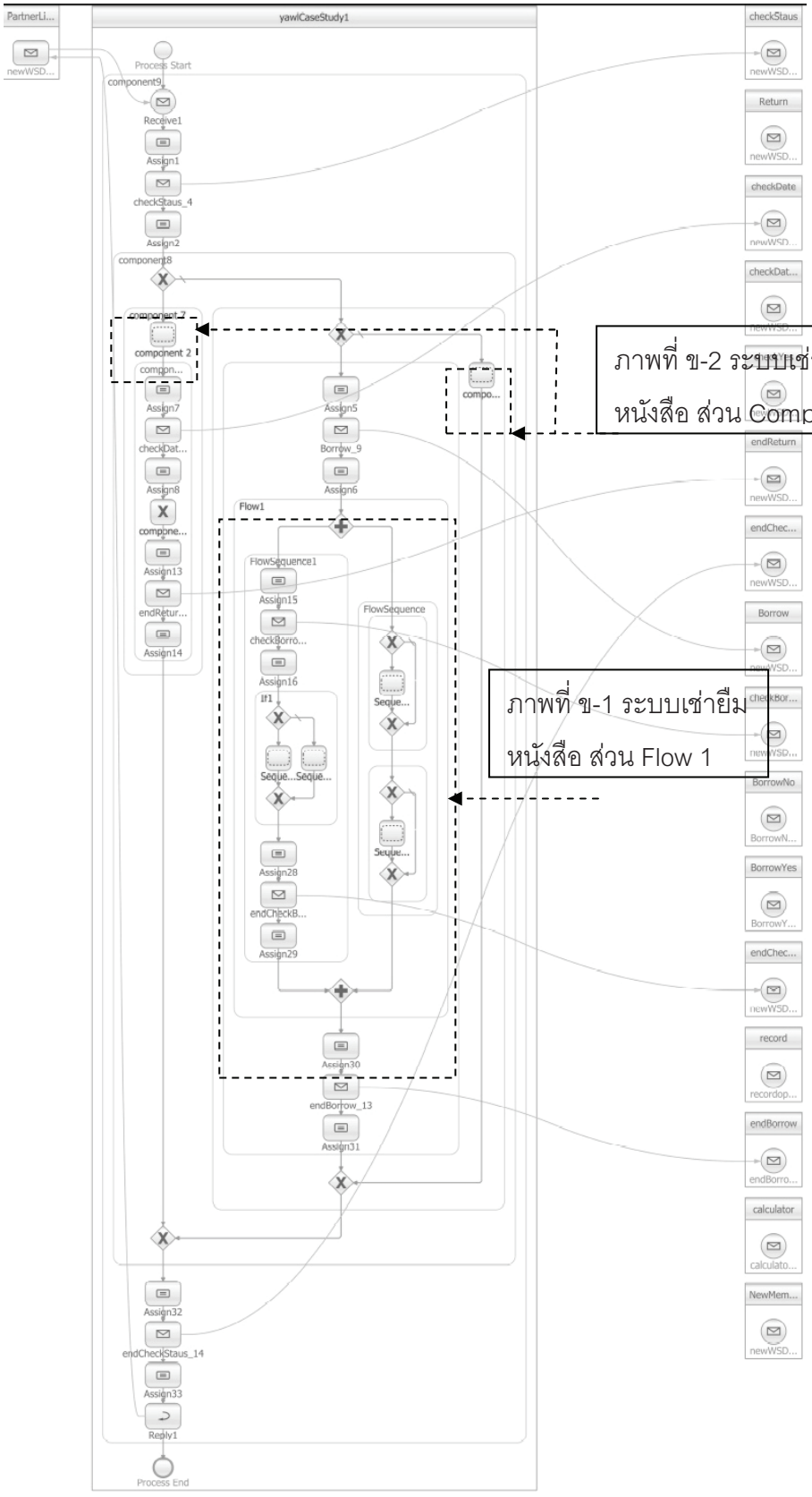
ตารางที่ ก-1 แบบรูปกระแสงานยอวล์โครงสร้างที่ดีเป็นโครงร่างบีเพล(ต่อ)

	แบบรูป	กระแสงานยอวล์	โครงร่างบีเพล
20	Cancel Case		<process> ... <exit/> ... </process>

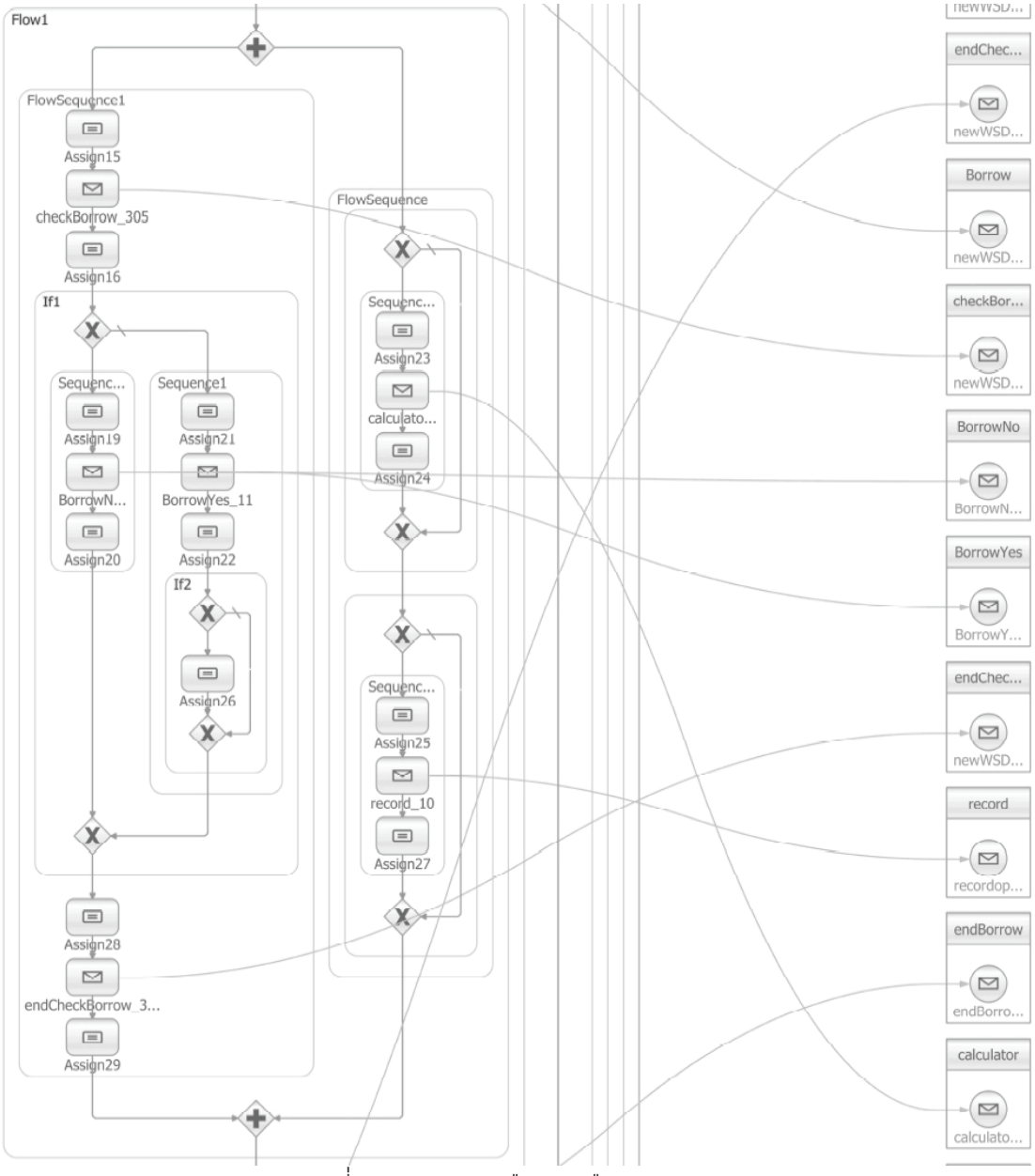
ภาคผนวก ข
ผลลัพธ์โครงร่างบีเพลของเครื่องมือ

ผลการทดสอบเครื่องมือการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพล ของ 6 กระแสงานทดสอบ ระบบเช่ายืมหนังสือ ระบบกู้ชำระเงิน ระบบการลงทะเบียนเรียน ระบบการสอบ ระบบรับพนักงานใหม่ ระบบสั่งซื้อหนังสือ

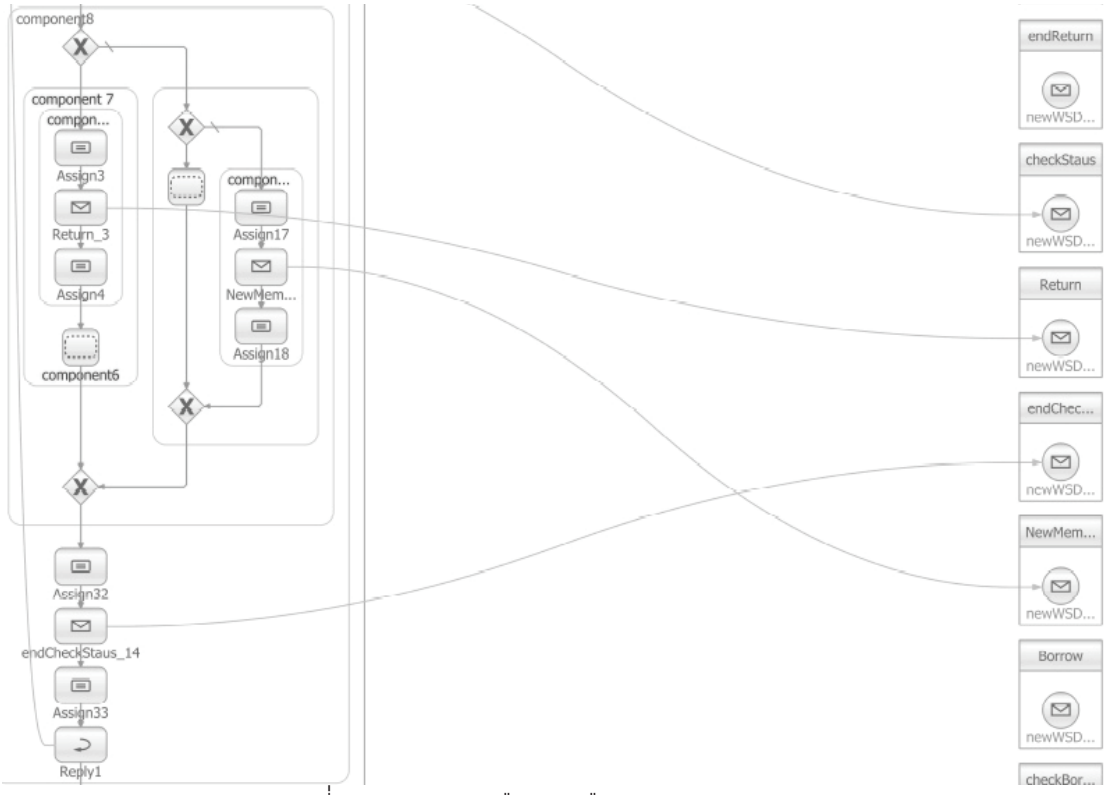
1. ระบบเช่ายืมหนังสือ



ภาพที่ ข-1 ระบบเช่ายืมหนังสือภาพรวม

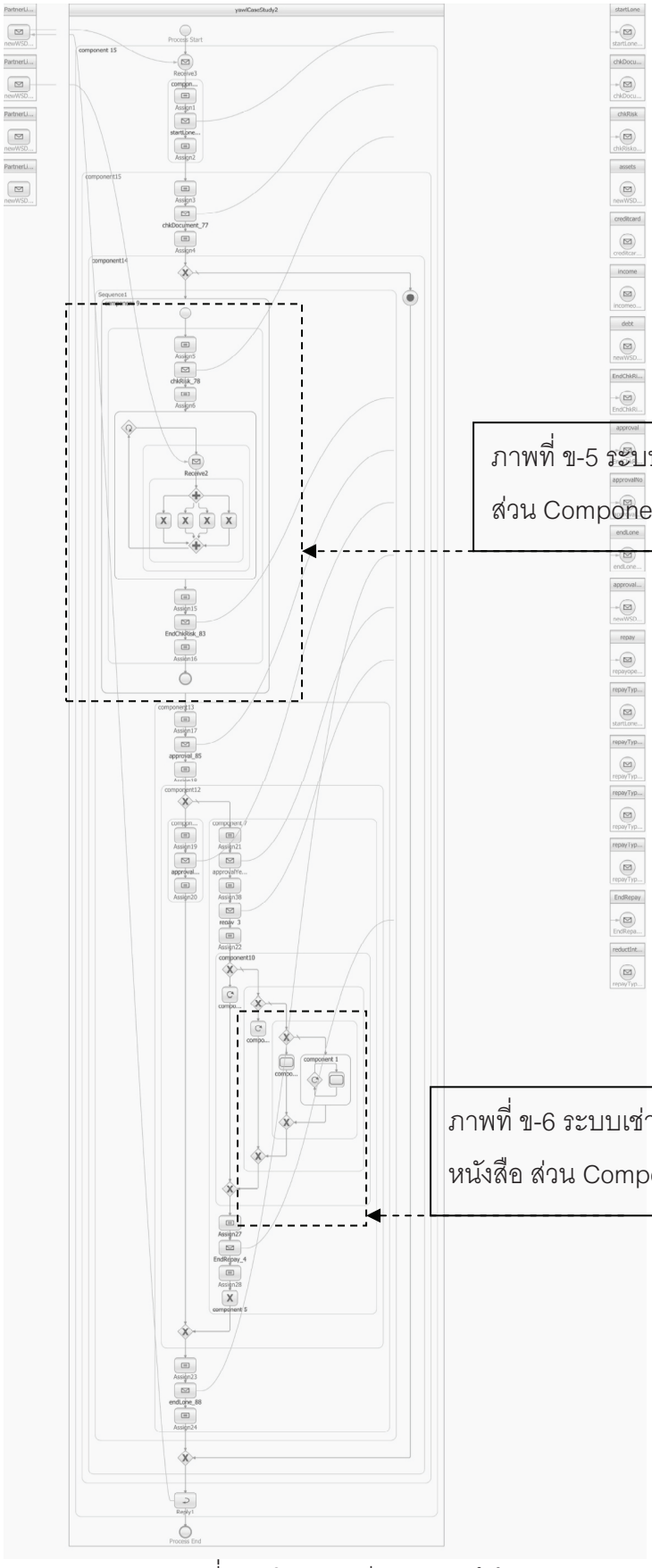


ภาพที่ ข-2 ระบบเช่ายืมหนังสือ ส่วน Flow 1

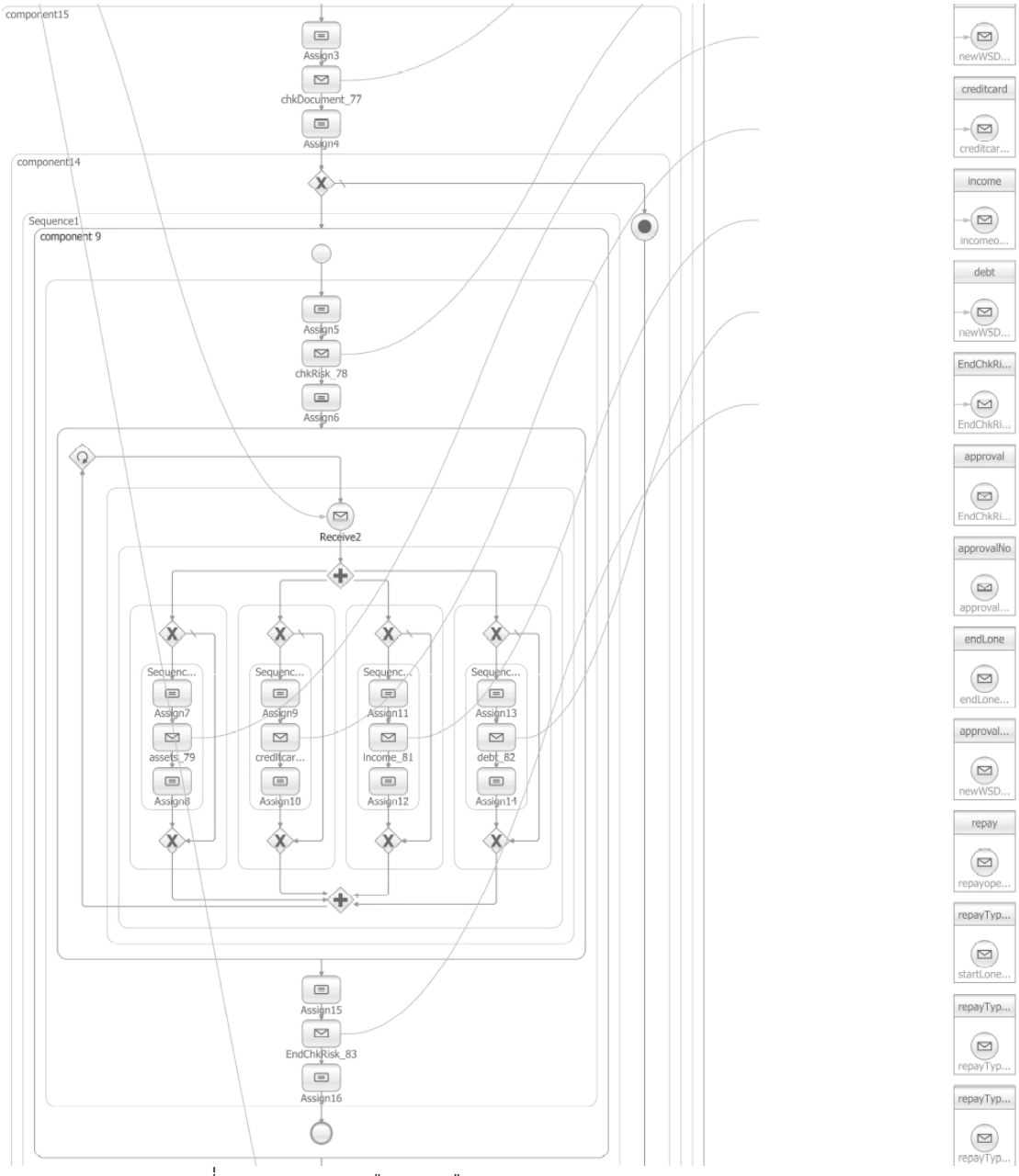


ภาพที่ ข-3 ระบบเช่ายืมหนังสือ ส่วน Component 7

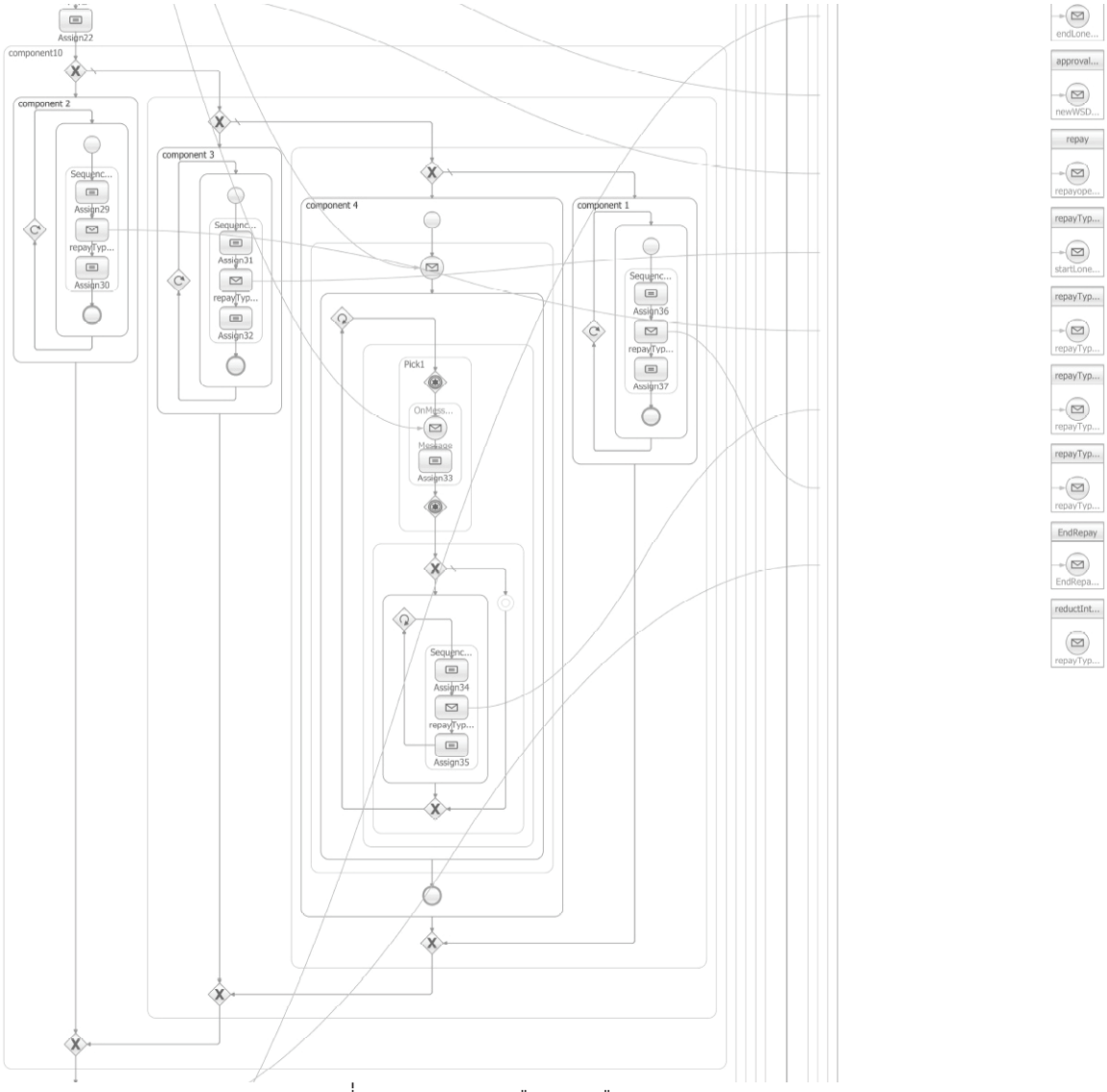
2. ระบบกู้ชำระเงิน



ภาพที่ ข-4 โครงร่างบีเพลระบบกู้ชำระเงิน

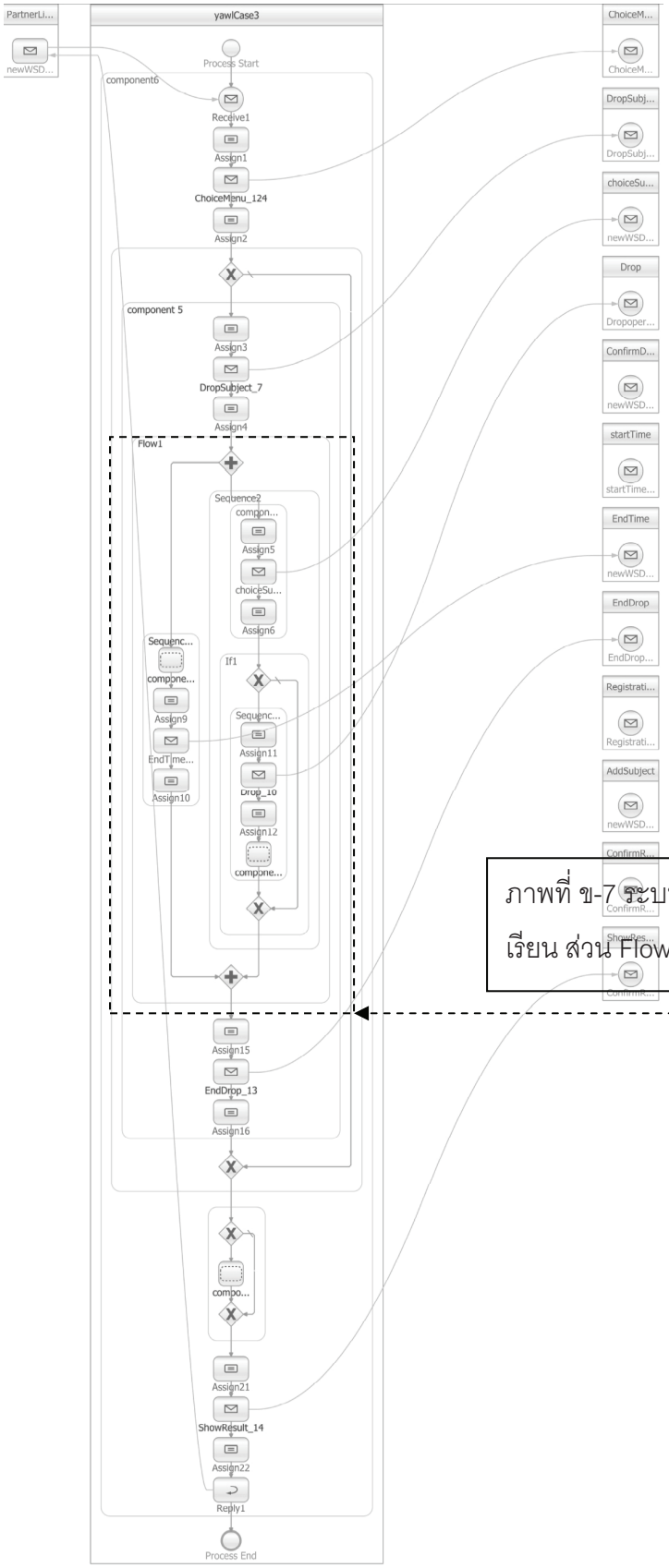


ภาพที่ ข-5 ระบบเช่ายืมหนังสือ ส่วน Component 9



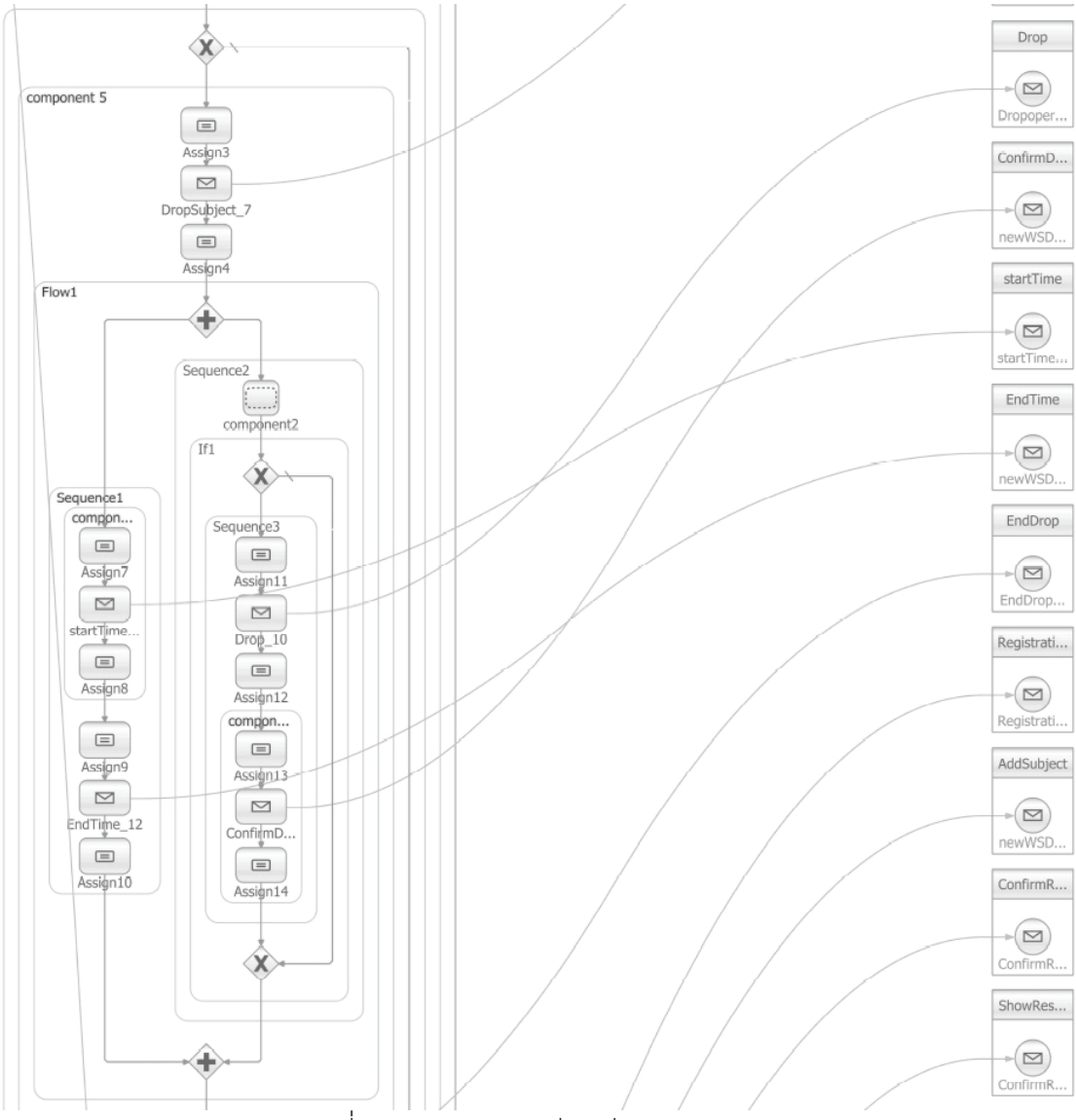
ภาพที่ ข-6 ระบบเซ่ายืมหนังสือ ส่วน Component 10

3. ระบบการลงทะเบียนเรียน



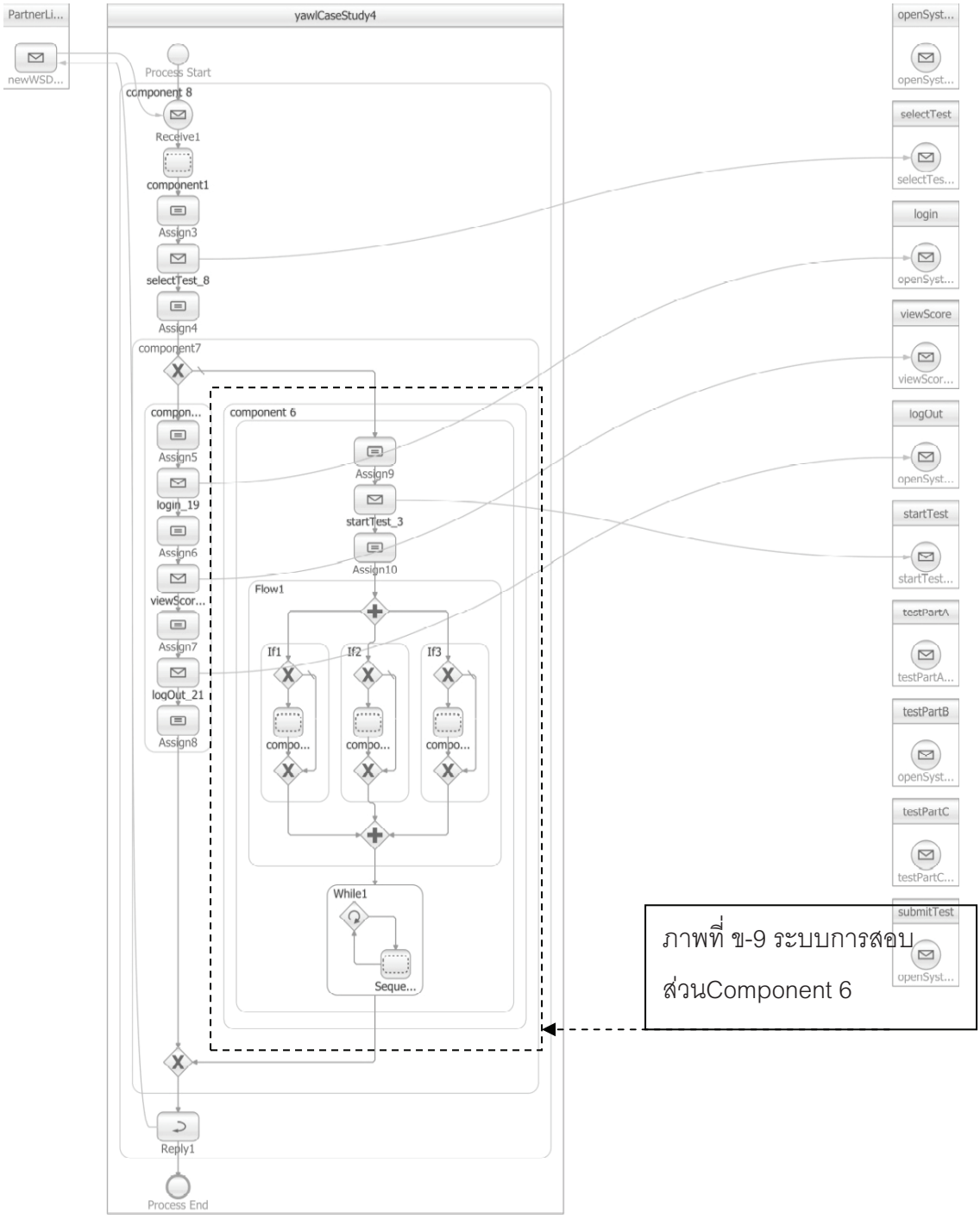
ภาพที่ ข-7 ระบบลงทะเบียนเรียน ส่วน Flow1

ภาพที่ ข-7 โครงร่างบีเพลระบบลงทะเบียนเรียน

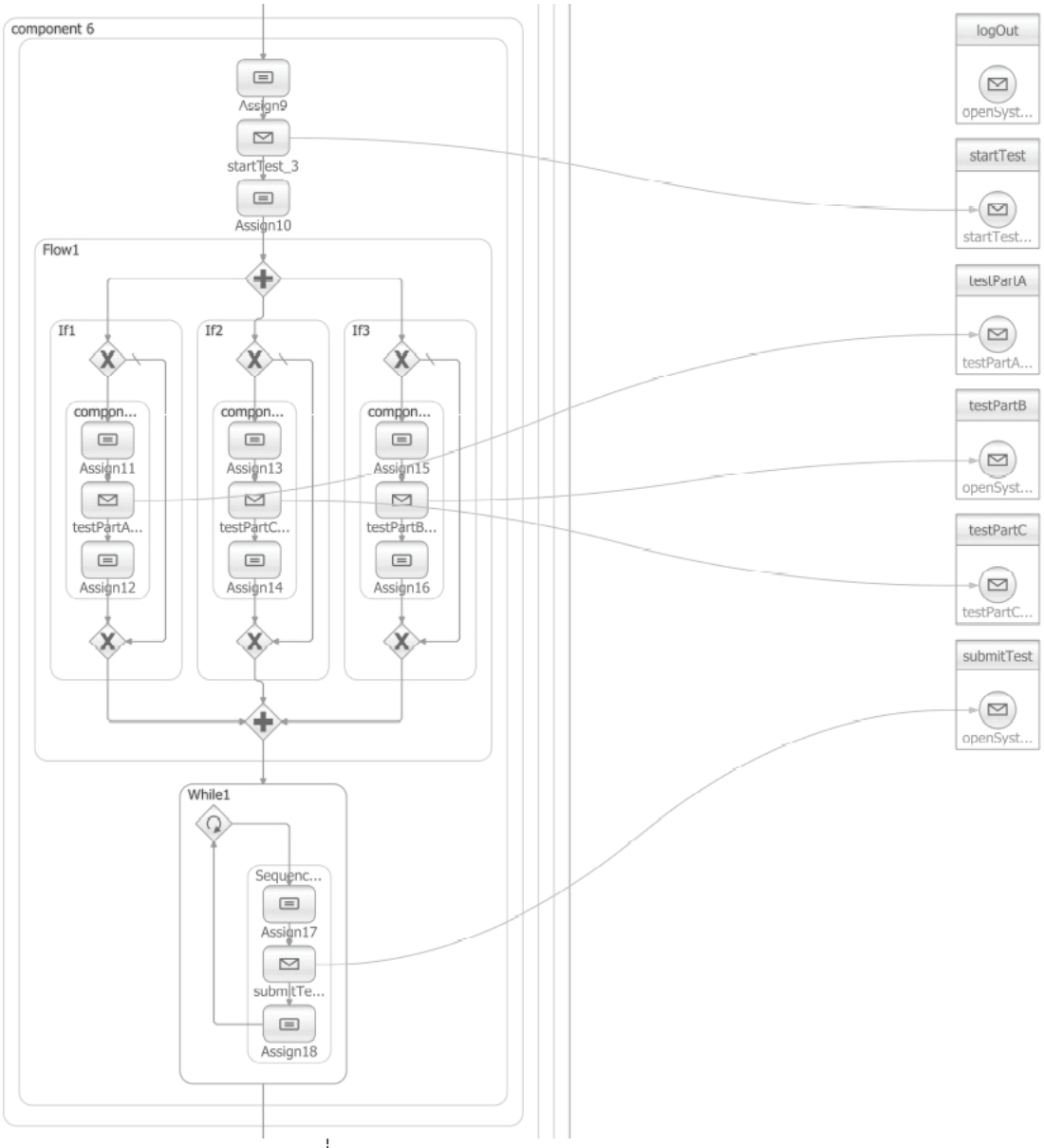


ภาพที่ ข-8 ระบบลงทะเบียนเรียน ส่วน Flow1

4. ระบบการสอบ

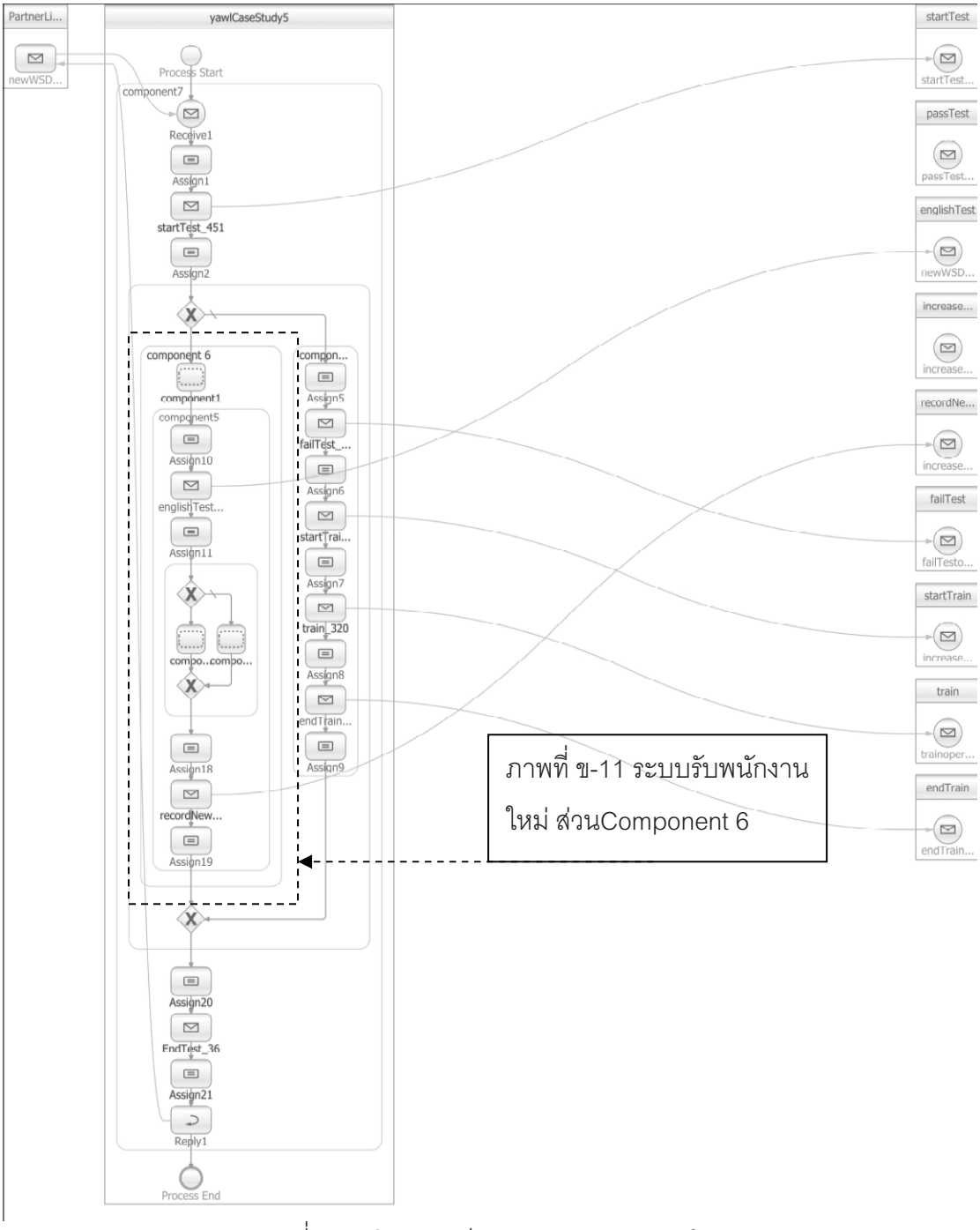


ภาพที่ ข-9 โครงร่างบีเพลระบบการสอบ

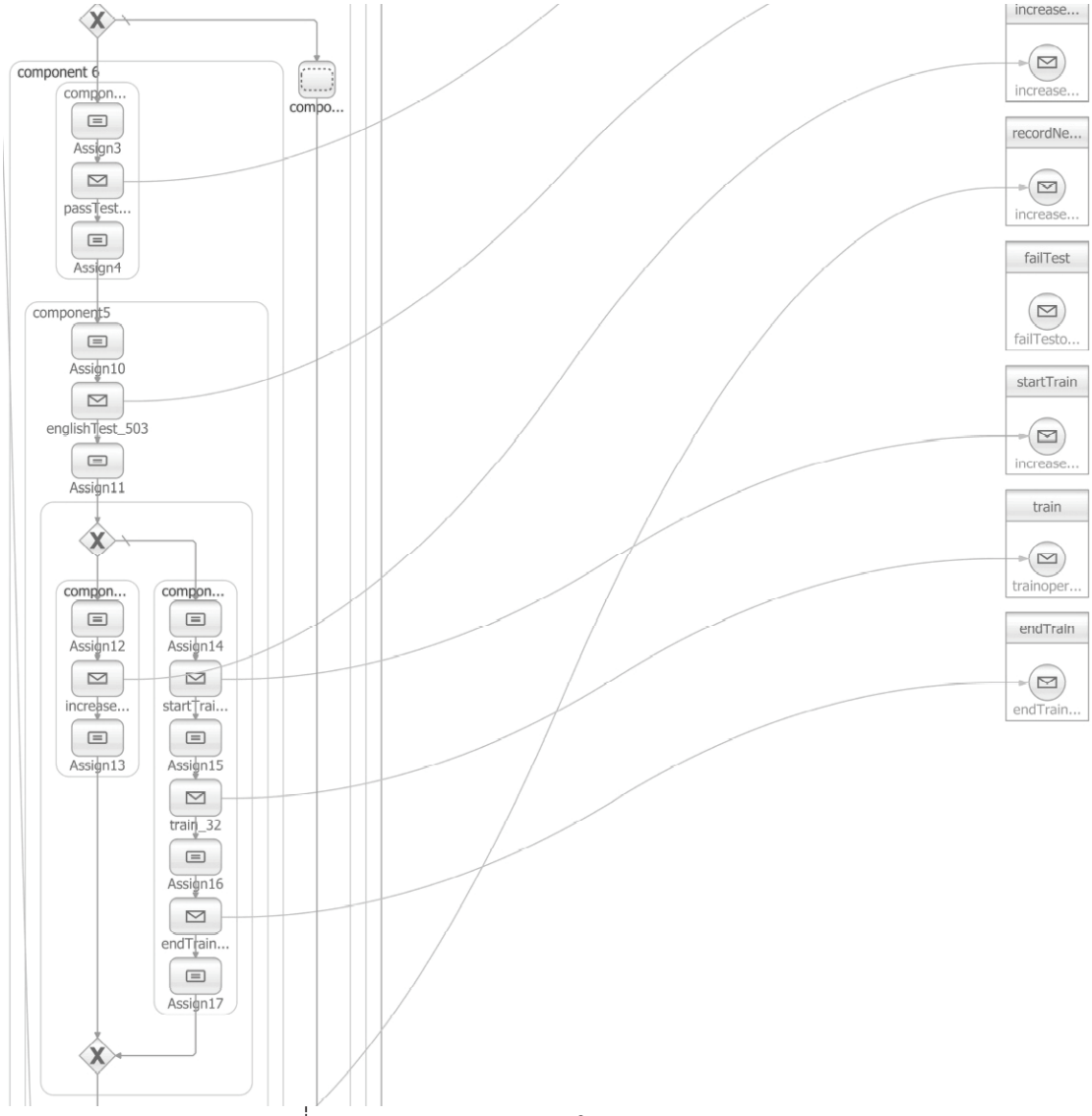


ภาพที่ ข-10 ระบบการสอบส่วน Component 6

5. ระบบรับพนักงานใหม่

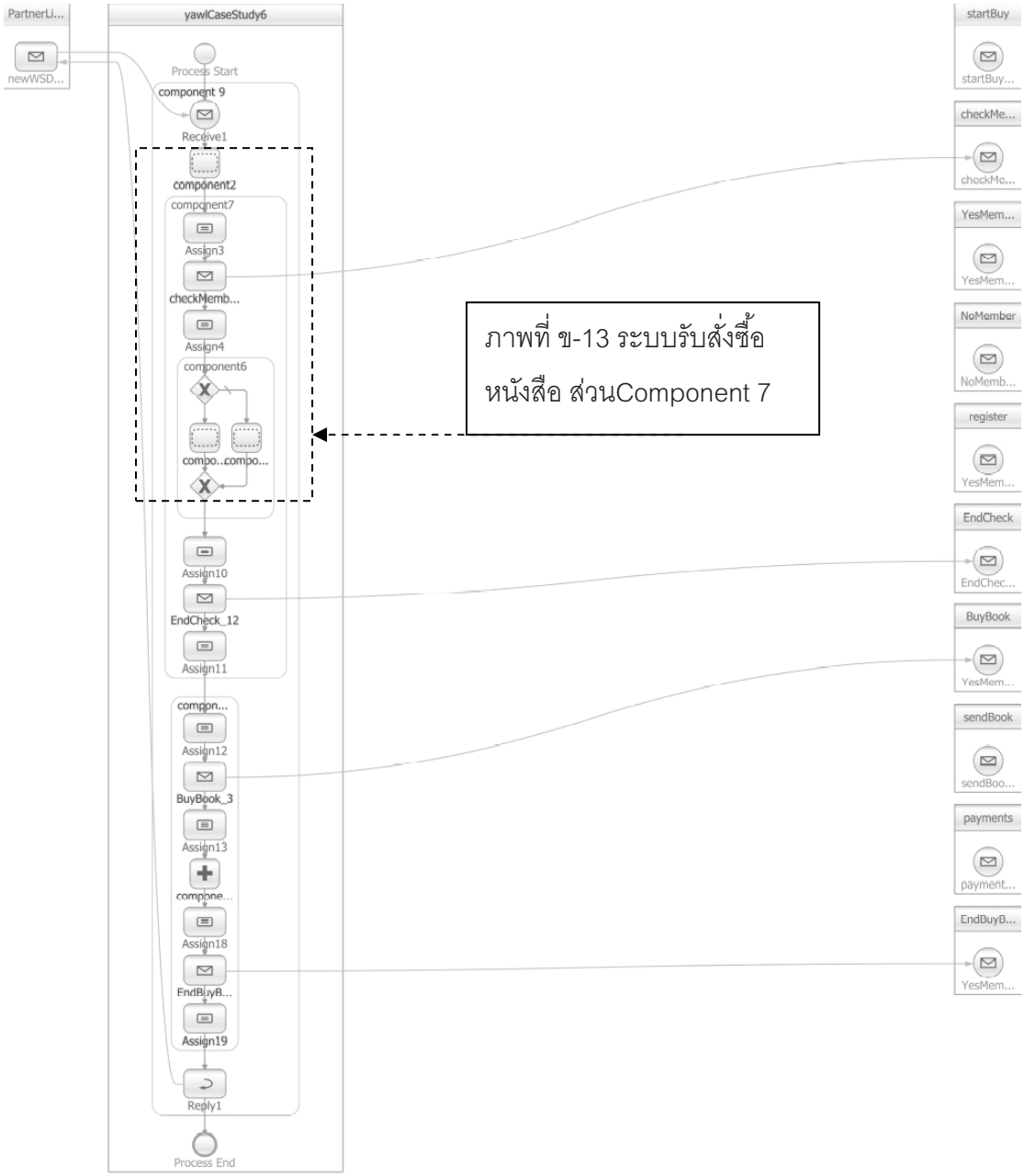


ภาพที่ ข-11 โครงร่างบีเพลระบบรับพนักงานใหม่

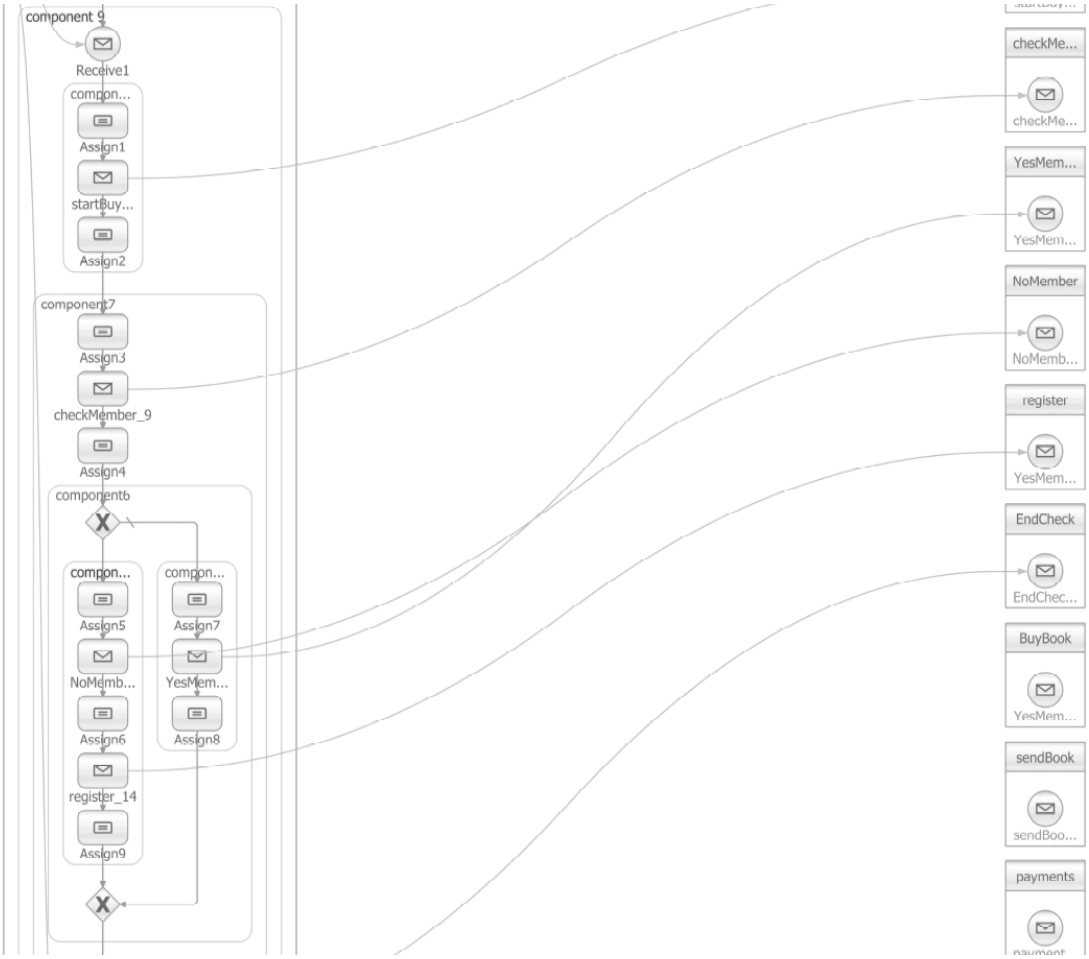


ภาพที่ ข-12 ระบบรับพนักงานใหม่ ส่วน Component 6

6. ระบบสั่งซื้อหนังสือ



ภาพที่ ข-13 โครงร่างบีเพลระบบสั่งซื้อหนังสือ



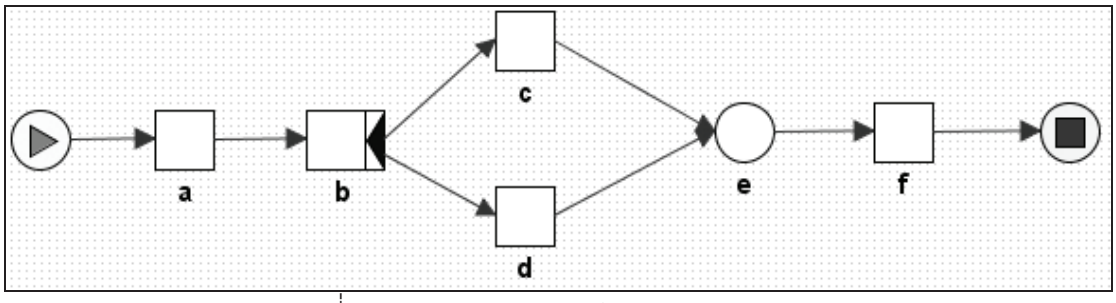
ภาพที่ ข-14 ระบบรับสั่งซื้อหนังสือ ส่วนComponent 7

ภาคผนวก ค
การทดสอบเครื่องมือตามแบบรูป

ในส่วนนี้ทำการทดสอบเครื่องมือ กรณีที่ผู้ทำวิทยานิพนธ์ออกแบบการเปลี่ยนรูป โดยทำการทำการทดสอบ 8 แบบรูปคือ แบบรูป Multi Merge, แบบรูป Arbitrary Cycle, แบบรูป MI without a Priori Run-Time Knowledge, แบบรูป Interleaved Parallel, แบบรูป Milestone แบบรูป MI without Synchronization แบบรูป Deferred Choice และแบบรูป Cancel Activity โดยการทดสอบแบบ 1 กระแสงานต่อ 1 แบบรูป

1 ทดสอบกระแสงานยอวล์แบบรูป Multi Merge

แผนภาพกระแสงานยอวล์ที่ใช้ทดสอบ ดังภาพที่ ค-1 โดยเป็นกระแสงานที่มีลักษณะตรงกับแบบรูป Multi Merge ลำดับการทำงานของกระแสงานเริ่มจากเซอร์วิส a เซอร์วิส b สามารถเลือกการทำงานของเซอร์วิส c หรือเซอร์วิส d และทำงานเซอร์วิส f โดยสามารถเกิดการทำงานเซอร์วิส f ซ้ำตามจำนวนที่เกิดเซอร์วิส c หรือเซอร์วิส d และจบการทำงาน ทำการทดสอบเครื่องมือโดยนำโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติม ดังภาพที่ ค-2 และทดสอบตามกรณีทดสอบตามตารางที่ ค-1



ภาพที่ ค-1 กระแสงานยอวล์แบบรูป Multi Merge

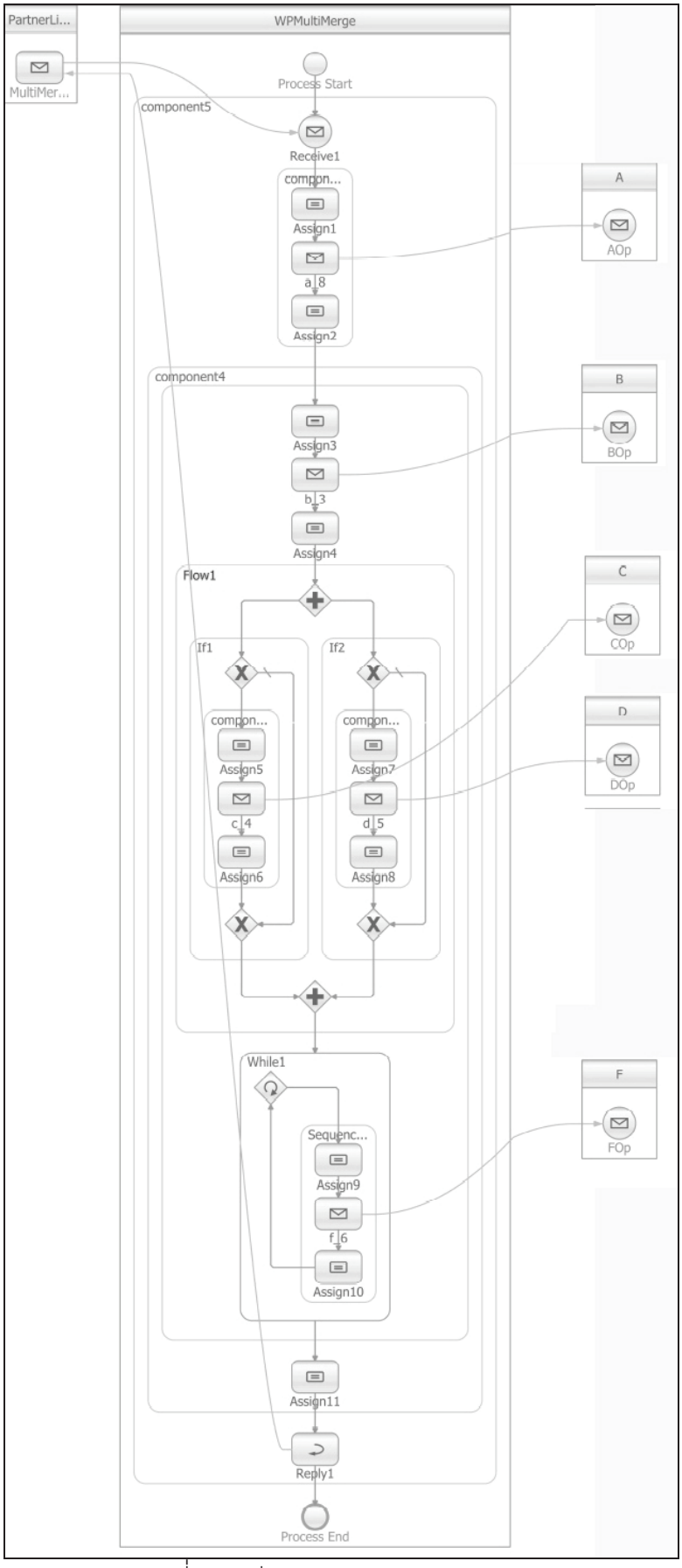
ตารางที่ ค-1 กรณีทดสอบกระแสงานยอวล์แบบรูป Multi Merge

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิส b ,เซอร์วิส c , เซอร์วิส e และเซอร์วิส f	✓
2	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิสb ,เซอร์วิส d , เซอร์วิส e และเซอร์วิสf	✓
3	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิส b ,เซอร์วิส c , เซอร์วิส d ,เซอร์วิส e ,เซอร์วิส e และเซอร์วิส f	✓

ตารางที่ ค-1 กรณีทดสอบกระแสงานยอวล์แบบรูป Multi Merge (ต่อ)

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
4	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิสb ,เซอร์วิส d , เซอร์วิส c ,เซอร์วิส e ,เซอร์วิส e และเซอร์วิส f	✓

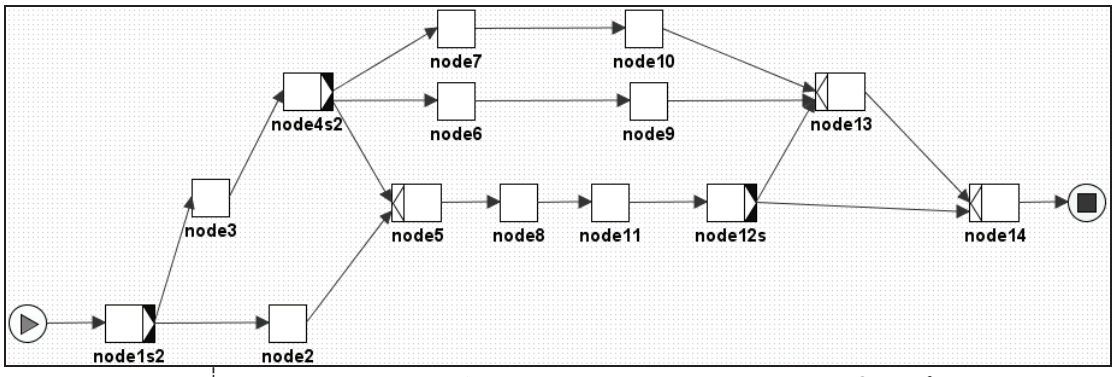
(หมายเหตุ เครื่องหมาย ✓หมายถึงได้ผลลัพธ์ตรงตามที่คาดหวัง)



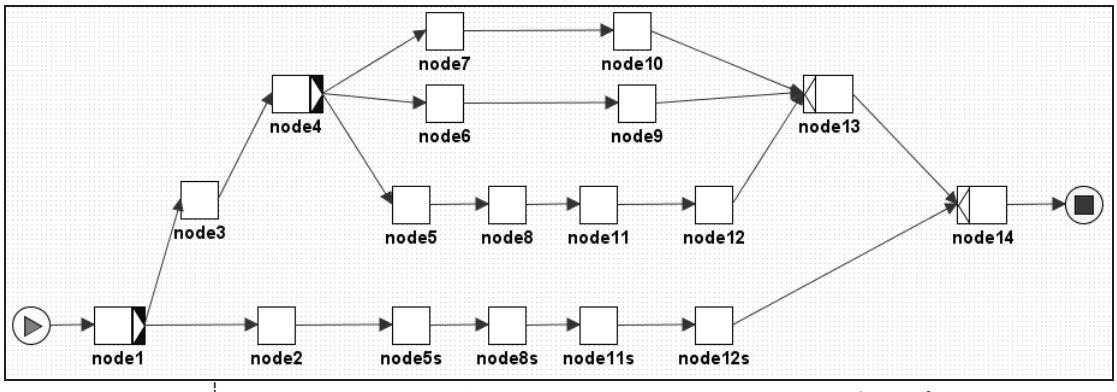
ภาพที่ ค-2 บีเพลของแบบรูป Multi Merge

2 ทดสอบกระแสงานยอวล์แบบรูป Arbitrary Cycle

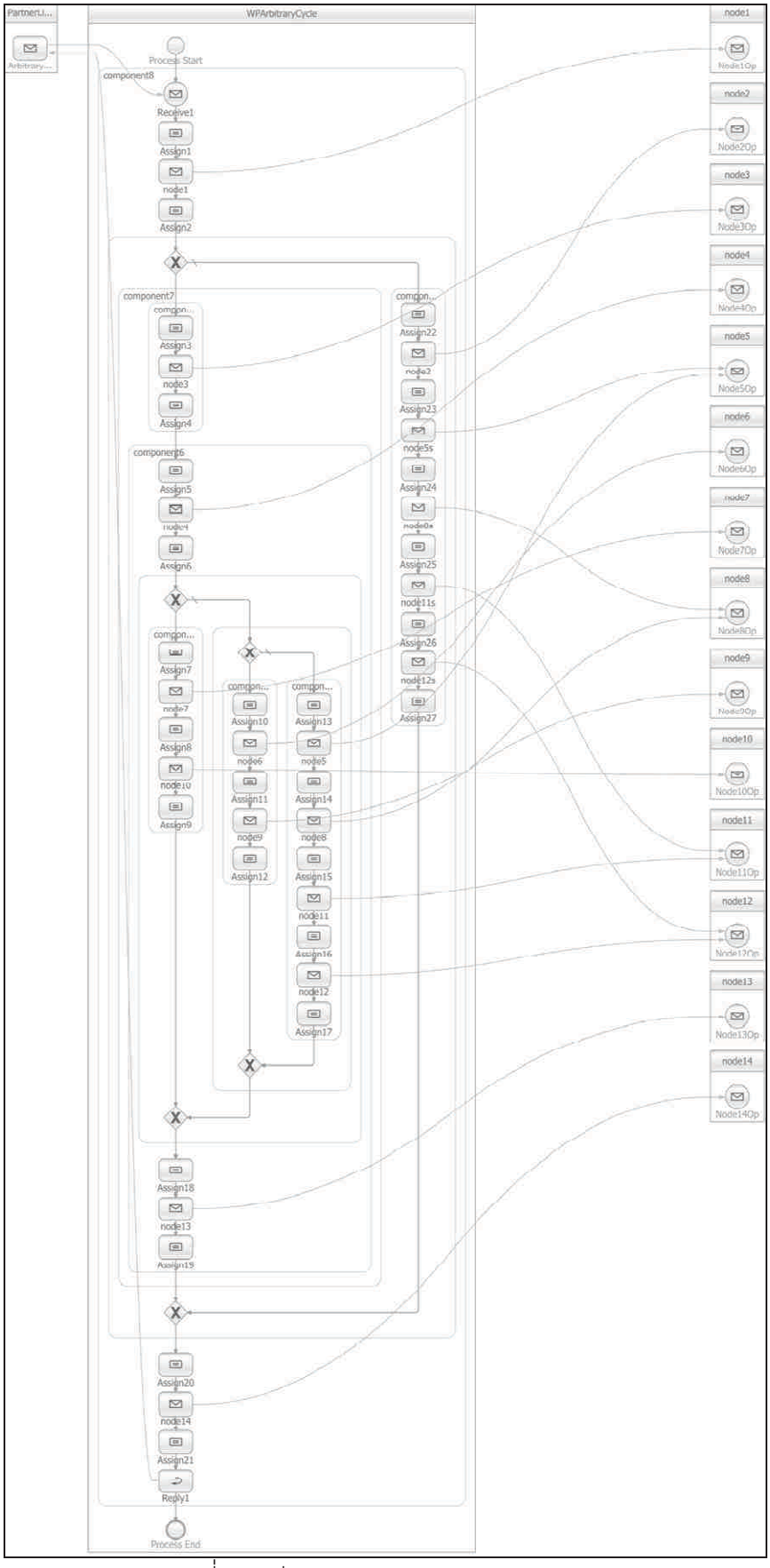
แผนภาพกระแสงานยอวล์ที่ใช้ทดสอบ ดังภาพที่ ค-3 โดยเป็นกระแสงานที่มีลักษณะตรงกับแบบรูป Arbitrary Cycle จากโครงสร้างกระแสงานไม่สามารถเปลี่ยนเป็นโครงร่างบีเพลได้ ระบบจึงทำการปรับโครงสร้างกระแสงานใหม่ตามรูป ค-4 แล้วทำการทดสอบเครื่องมือโดยนำโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติม ดังภาพที่ ค-5 และทดสอบตามกรณีทดสอบตามตารางที่ ค-2



ภาพที่ ค-3 กระแสงานยอวล์แบบรูป Arbitrary Cycle ก่อนปรับโครงสร้าง



ภาพที่ ค-4 กระแสงานยอวล์แบบรูป Arbitrary Cycle หลังปรับโครงสร้าง



ภาพที่ ค-5 บีเพลของแบบรูป Arbitrary Cycle

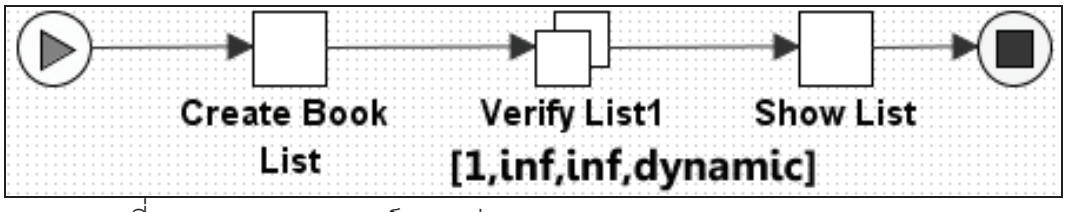
ตารางที่ ค-2 กรณีทดสอบกระแสงานยอวล์แบบรูป Arbitrary Cycle

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส Node15, เซอร์วิส Node3, เซอร์วิส Node4S2, เซอร์วิส Node7, เซอร์วิส Node10, เซอร์วิส Node13 และเซอร์วิส Node14	✓
2	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส Node15, เซอร์วิส Node3, เซอร์วิส Node4S2, เซอร์วิส Node7, เซอร์วิส Node10, เซอร์วิส Node13 และเซอร์วิส Node14	✓
3	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส Node15, เซอร์วิส Node3, เซอร์วิส Node4S2, เซอร์วิส Node7, เซอร์วิส Node10, เซอร์วิส Node13 และเซอร์วิส Node14	✓
4	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส Node15, เซอร์วิส Node3, เซอร์วิส Node4S2, เซอร์วิส Node7, เซอร์วิส Node10, เซอร์วิส Node13 และเซอร์วิส Node14	✓

(หมายเหตุ เครื่องหมาย ✓หมายถึงได้ผลลัพธ์ตรงตามที่คาดหวัง)

3 ทดสอบกระแสงานยอวล์แบบรูป MI without a Priori Run-Time Knowledge

แผนภาพกระแสงานยอวล์ที่ใช้ทดสอบ ดังภาพที่ ค-6 โดยเป็นกระแสงานที่มีลักษณะตรงกับแบบรูป MI without a Priori Run-Time Knowledge ลำดับการทำงานของกระแสงานเริ่มจากเซอร์วิส Create Book List แล้วเรียกเซอร์วิส Verify List1 โดยเป็นเซอร์วิสแบบ Multiple instance มีลักษณะเป็น [1,inf,inf,dynamic] คือ เกิดจำนวน instance เท่ากับจำนวน ความต้องการตอนทำงานของกระแสงานและสามารถเพิ่มจำนวน instance ได้ขณะเรียกเซอร์วิส VerifyList1 หลังจากนั้นเรียกเซอร์วิส Show List แล้วจบการทำงาน ทำการทดสอบเครื่องมือโดยนำโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติมดังภาพที่ ค-7 และทดสอบตามกรณีทดสอบตามตารางที่ ค-3

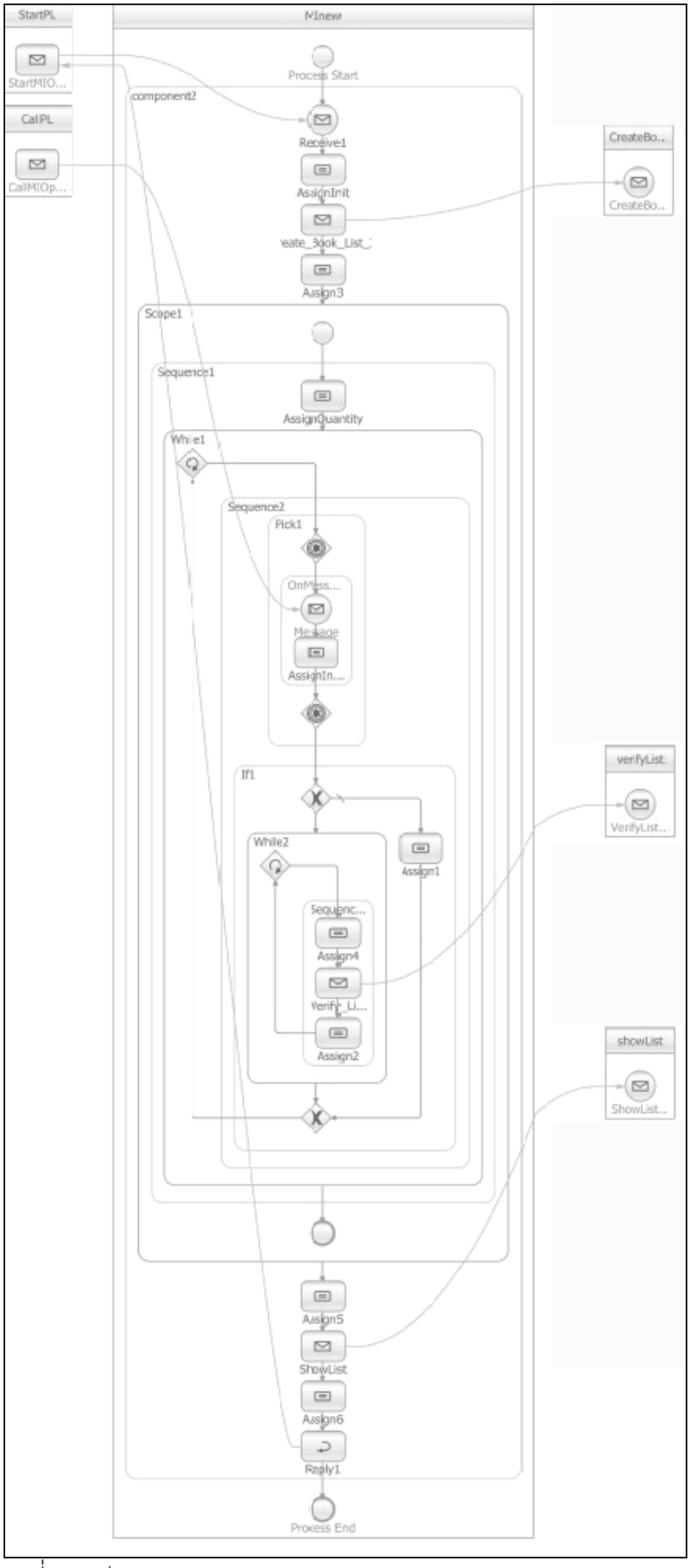


ภาพที่ ค-6 กระแสงานยอวล์แบบรูป MI without a Priori Run-Time Knowledge

ตารางที่ ค-3 กรณีทดสอบกระแสงานยอวล์แบบรูป MI without a Priori Run-Time Knowledge

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส Create Book List ,เซอร์วิส Verify List1 โดยผู้ใช้สามารถเพิ่มจำนวนเซอร์วิสซ้ำตามความต้องการตอนปฏิบัติการกระแสงาน แล้วเรียกเซอร์วิส Show List และจบการทำงาน	✓

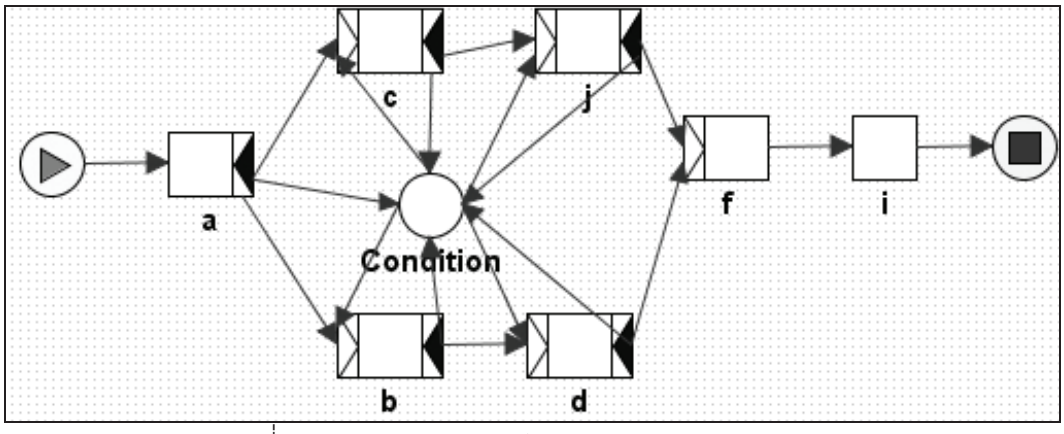
(หมายเหตุ เครื่องหมาย ✓หมายถึงได้ผลลัพธ์ตรงตามที่คาดหวัง)



ภาพที่ ค-7 บีเพลของแบบรูป MI without a Priori Run-Time Knowledge

4 ทดสอบกระแสงานยอวล์แบบรูป Interleaved Parallel

แผนภาพกระแสงานยอวล์ที่ใช้ทดสอบ ดังภาพที่ ค-8 โดยเป็นกระแสงานที่มีลักษณะตรงกับแบบรูป Interleaved Parallel ลำดับการทำงานของกระแสงานเริ่มจากเซอร์วิส a แล้วเรียกเซอร์วิส b หรือเซอร์วิส c ทำงานแล้วเซอร์วิส j หรือเซอร์วิส d ทำงานโดยเลือกทำงานเซอร์วิสได้ครั้งละ 1 เซอร์วิส หลังจากนั้นเรียกเซอร์วิส f เซอร์วิส i ทำงานแล้วจบการทำงาน ทำการทดสอบเครื่องมือโดยนำโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติม ตามภาพที่ ค-9 และทดสอบตามกรณีทดสอบตามตารางที่ ค-3

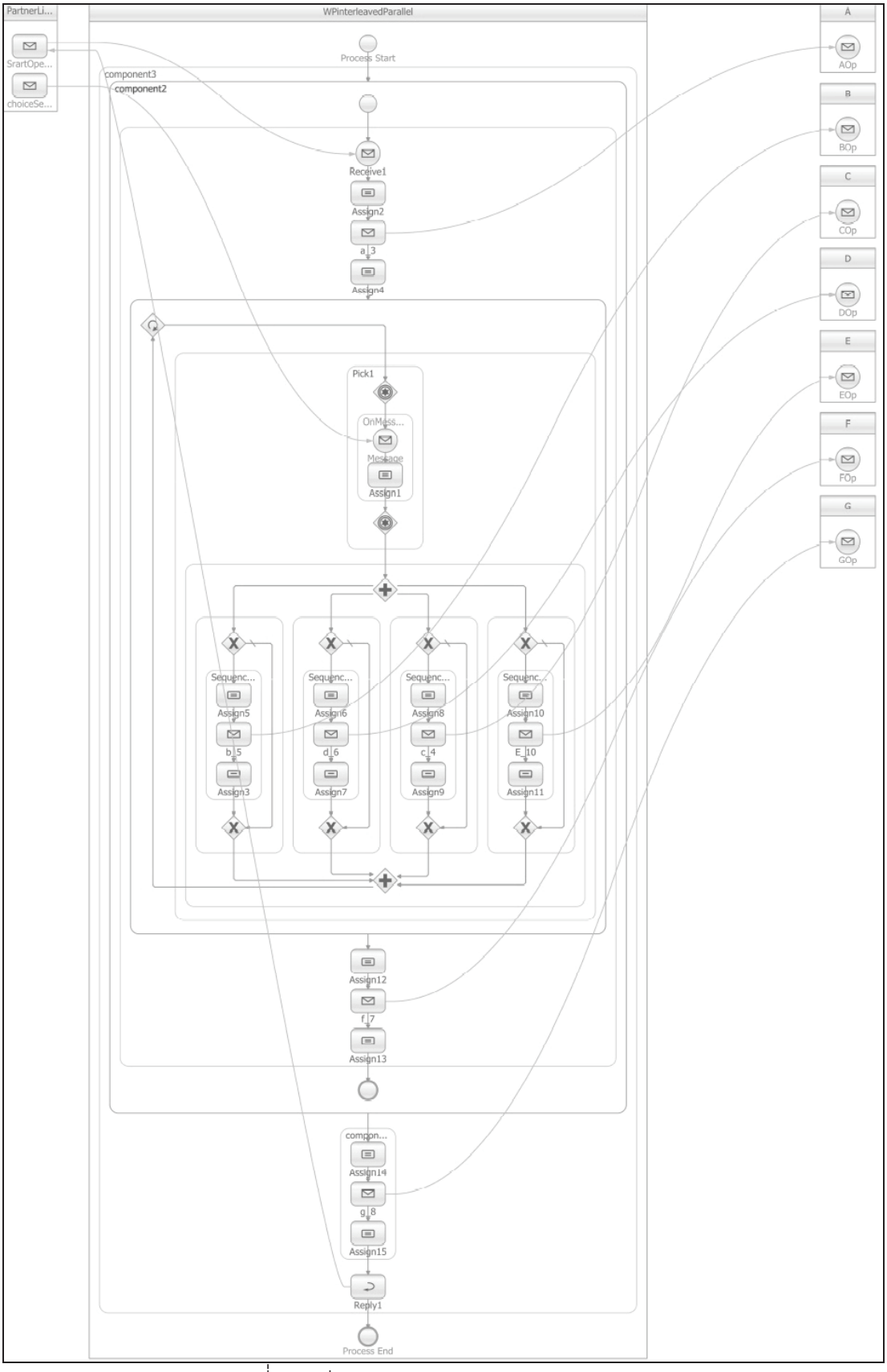


ภาพที่ ค-8 กระแสงานยอวล์แบบรูป Interleaved Parallel

ตารางที่ ค-4 กรณีทดสอบกระแสงานยอวล์แบบรูป Interleaved Parallel

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิส b ,เซอร์วิส d ,เซอร์วิส c ,เซอร์วิส j ,เซอร์วิส f ,เซอร์วิส i และจบการทำงาน	✓
2	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิส b ,เซอร์วิส c ,เซอร์วิส d ,เซอร์วิส j ,เซอร์วิส f ,เซอร์วิส i และจบการทำงาน	✓
3	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิส b ,เซอร์วิส c ,เซอร์วิส j ,เซอร์วิส d ,เซอร์วิส f ,เซอร์วิส i และจบการทำงาน	✓
4	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิส c ,เซอร์วิส j , เซอร์วิส b ,เซอร์วิส d ,เซอร์วิส f ,เซอร์วิส i และจบการทำงาน	✓
5	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิส c ,เซอร์วิส b ,เซอร์วิส d ,เซอร์วิส j ,เซอร์วิส f ,เซอร์วิส i และจบการทำงาน	✓
6	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส a ,เซอร์วิส c ,เซอร์วิส b ,เซอร์วิส j ,เซอร์วิส d ,เซอร์วิส f ,เซอร์วิส i และจบการทำงาน	✓

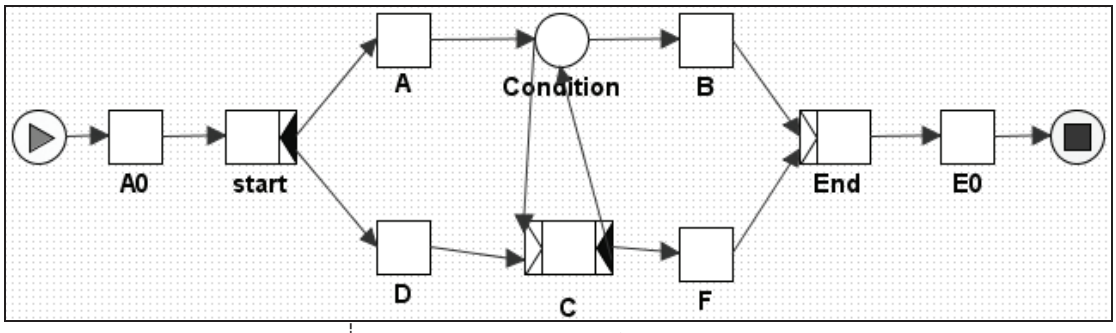
(หมายเหตุ เครื่องหมาย ✓ หมายถึงได้ผลลัพธ์ตรงตามที่คาดหวัง)



ภาพที่ ค-9 บีเพลของแบบรูป Interleaved Parallel

5 ทดสอบกระแสงานยอวล์แบบรูป Milestone

แผนภาพกระแสงานยอวล์ที่ใช้ทดสอบ ดังภาพที่ ค-10 โดยเป็นกระแสงานที่มีลักษณะตรงกับแบบรูป Milestone คือมีเซอร์วิสหนึ่งๆสามารถกำหนดให้เซอร์วิสอื่น ทำงานหรือไม่ทำงานได้ แล้วทำการทดสอบเครื่องมือการเปลี่ยนกระแสงานยอวล์เป็นโครงร่างบีเพลโดยนำโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติม ดังภาพที่ ค-11 จากตัวอย่างกระแสงานยอวล์ทดสอบสามารถแบ่งกรณีทดสอบได้ 10 กรณีตามตารางที่ ค-4



ภาพที่ ค-10 กระแสงานยอวล์แบบรูป Milestone

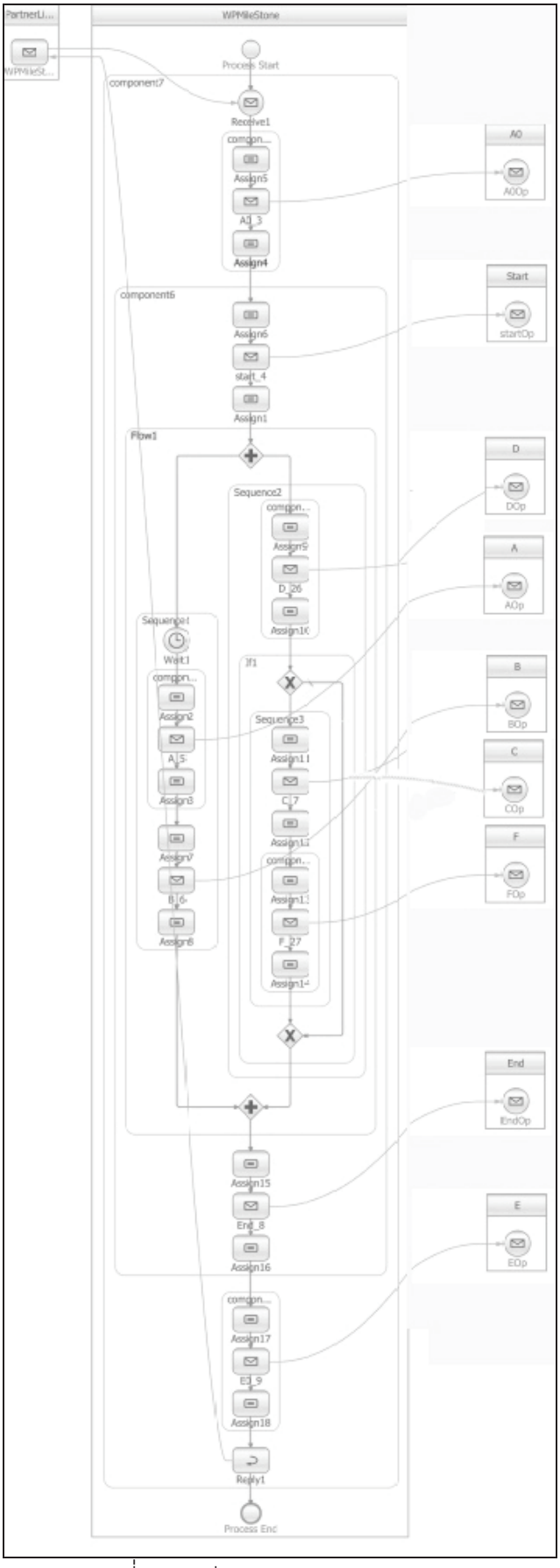
ตารางที่ ค-5 กรณีทดสอบกระแสงานยอวล์แบบรูป Milestone

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส B ,เซอร์วิส D ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓
2	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส Start , เซอร์วิส A ,เซอร์วิส D ,เซอร์วิส B ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓
3	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส start , เซอร์วิส A ,เซอร์วิส D ,เซอร์วิส C ,เซอร์วิส F ,เซอร์วิส B ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓
4	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส start , เซอร์วิส A ,เซอร์วิส D ,เซอร์วิส C ,เซอร์วิส B ,เซอร์วิส F ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓
5	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส start , เซอร์วิส D ,เซอร์วิส C ,เซอร์วิส F ,เซอร์วิส A ,เซอร์วิส B ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓

ตารางที่ ค-5กรณีทดสอบกระแสงานยอวล์แบบรูป Milestone (ต่อ)

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
6	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส start , เซอร์วิส D ,เซอร์วิส A ,เซอร์วิส C ,เซอร์วิส F ,เซอร์วิส B ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓
7	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส start , เซอร์วิส D ,เซอร์วิส C ,เซอร์วิส A ,เซอร์วิส B ,เซอร์วิส F ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓
8	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส start , เซอร์วิส D ,เซอร์วิส C ,เซอร์วิส A ,เซอร์วิส F ,เซอร์วิส B ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓
9	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส start , เซอร์วิส D ,เซอร์วิส A ,เซอร์วิส B ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓
10	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส A0 ,เซอร์วิส start , เซอร์วิส D ,เซอร์วิส A ,เซอร์วิส C ,เซอร์วิส B ,เซอร์วิส F ,เซอร์วิส End ,เซอร์วิส E0 และจบการทำงาน	✓

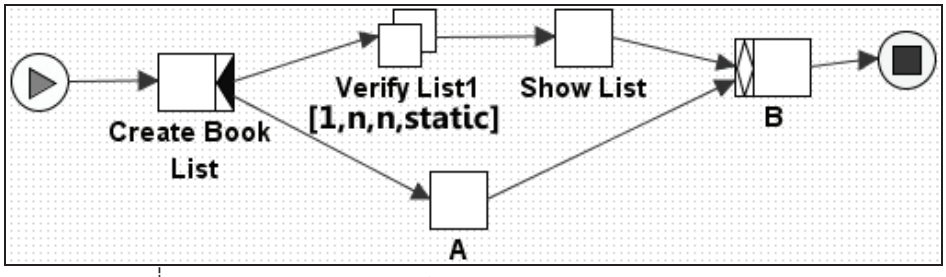
(หมายเหตุ เครื่องหมาย ✓หมายถึงได้ผลลัพธ์ที่คาดหวัง)



ภาพที่ ค-11 บีเพลของแบบรูป Milestone

6 ทดสอบกระแสงานยอวล์แบบรูป MI without Synchronization

แผนภาพกระแสงานยอวล์ที่ใช้ทดสอบ ดังภาพที่ ค-12 โดยเป็นกระแสงานที่มีลักษณะตรงกับแบบรูป MI without Synchronization ลำดับการทำงานของกระแสงานเริ่มจากเซอร์วิส Create Book List แล้วเรียกเซอร์วิส Verify List1 โดยเป็นเซอร์วิสแบบ multiple task มีลักษณะ [1,n,n,static] คือ เกิดจำนวน instance กับ completionCondition เท่ากับ n จำนวน และ static คือจำนวน instance ที่ขั้นตอนการออกแบบ หลังจากนั้นเรียกเซอร์วิส Show List โดยทำงานพร้อมกับเซอร์วิส A แล้วเรียกเซอร์วิส B และจบการทำงาน ทำการทดสอบเครื่องมือโดยนำโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติม ดังภาพที่ ค-13 และทดสอบตามกรณีทดสอบตามตารางที่ ค-6

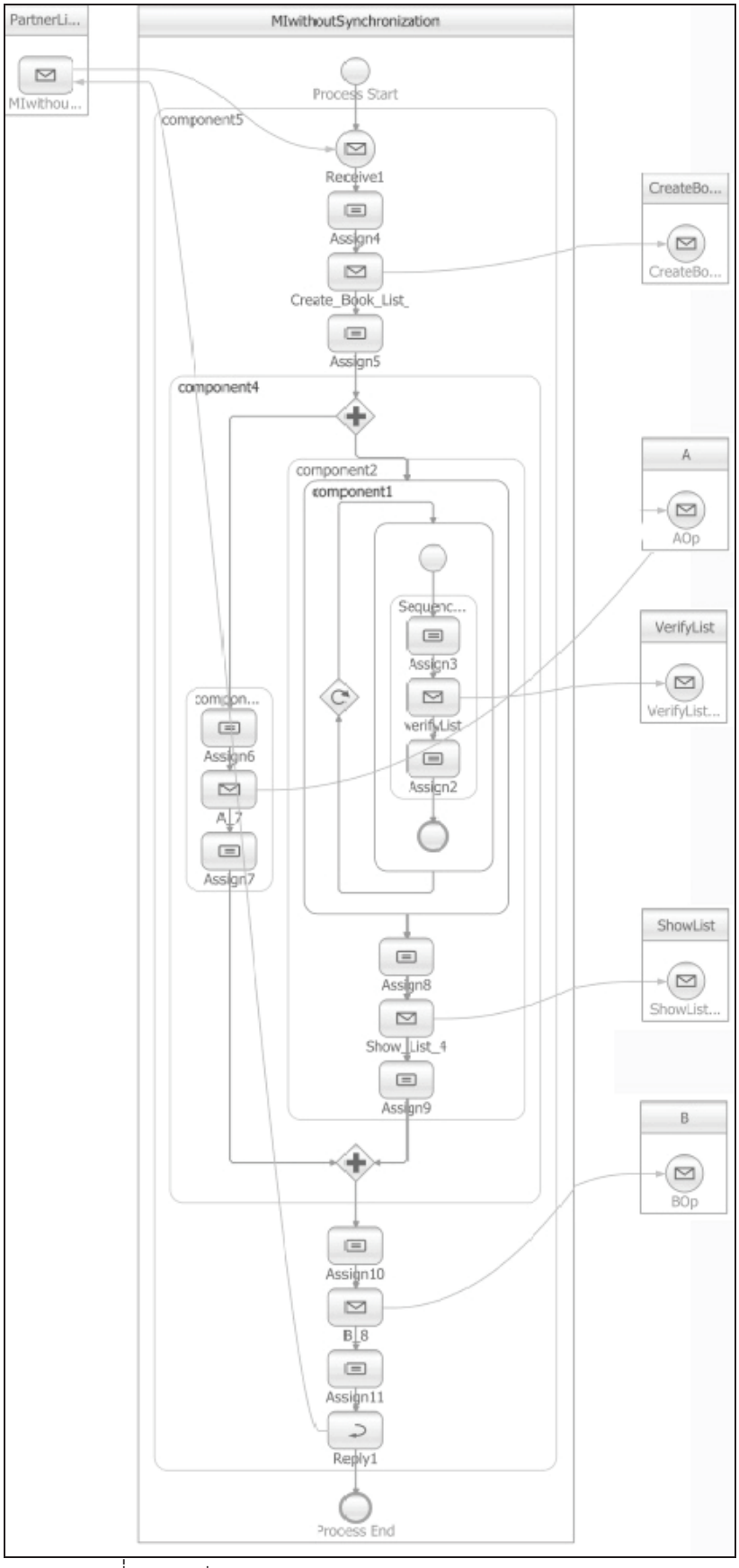


ภาพที่ ค-12 กระแสงานยอวล์แบบรูป MI without Synchronization

ตารางที่ ค-6 กรณีทดสอบกระแสงานยอวล์แบบรูป MI without Synchronization

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส Create Book List, เรียกเซอร์วิส Verify List1 โดยเรียกเซอร์วิสซ้ำตามจำนวนinstance ในบีเพลแล้วเรียกเซอร์วิส Show List โดยเซอร์วิส A ทำงานขนานกัน แล้วเรียกเซอร์วิส B และจบการทำงาน	✓

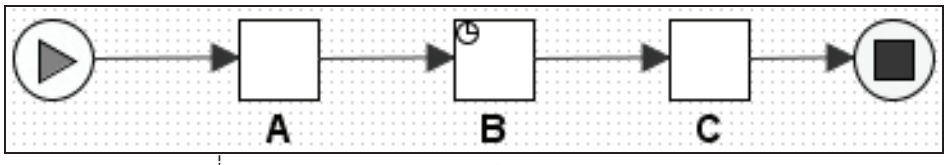
(หมายเหตุ เครื่องหมาย ✓หมายถึงได้ผลลัพธ์ตรงตามที่คาดหวัง)



ภาพที่ ค-13 บีเพลของแบบรูป MI without Synchronization

7 ทดสอบกระแสงานยอวล์แบบรูป Deferred Choice

แผนภาพกระแสงานยอวล์ที่ใช้ทดสอบ ดังภาพที่ ค-14 โดยเป็นกระแสงานที่มีลักษณะตรงกับแบบรูป Deferred Choice ลำดับการทำงานของกระแสงานเริ่มจากเซอร์วิส A แล้วเรียกเซอร์วิส B โดยเป็นเซอร์วิสแบบ Set task timer เซอร์วิสสามารถตั้งเวลาการทำงานได้ตามเวลาที่กำหนดถ้าเกินเวลาที่กำหนดเซอร์วิส B ไม่สามารถทำงาน หลังจากนั้นเรียกเซอร์วิส C และจบการทำงาน ทำการทดสอบเครื่องมือโดยนำโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติมดังภาพที่ ค-15 และทดสอบตามกรณีทดสอบตามตารางที่ ค-7

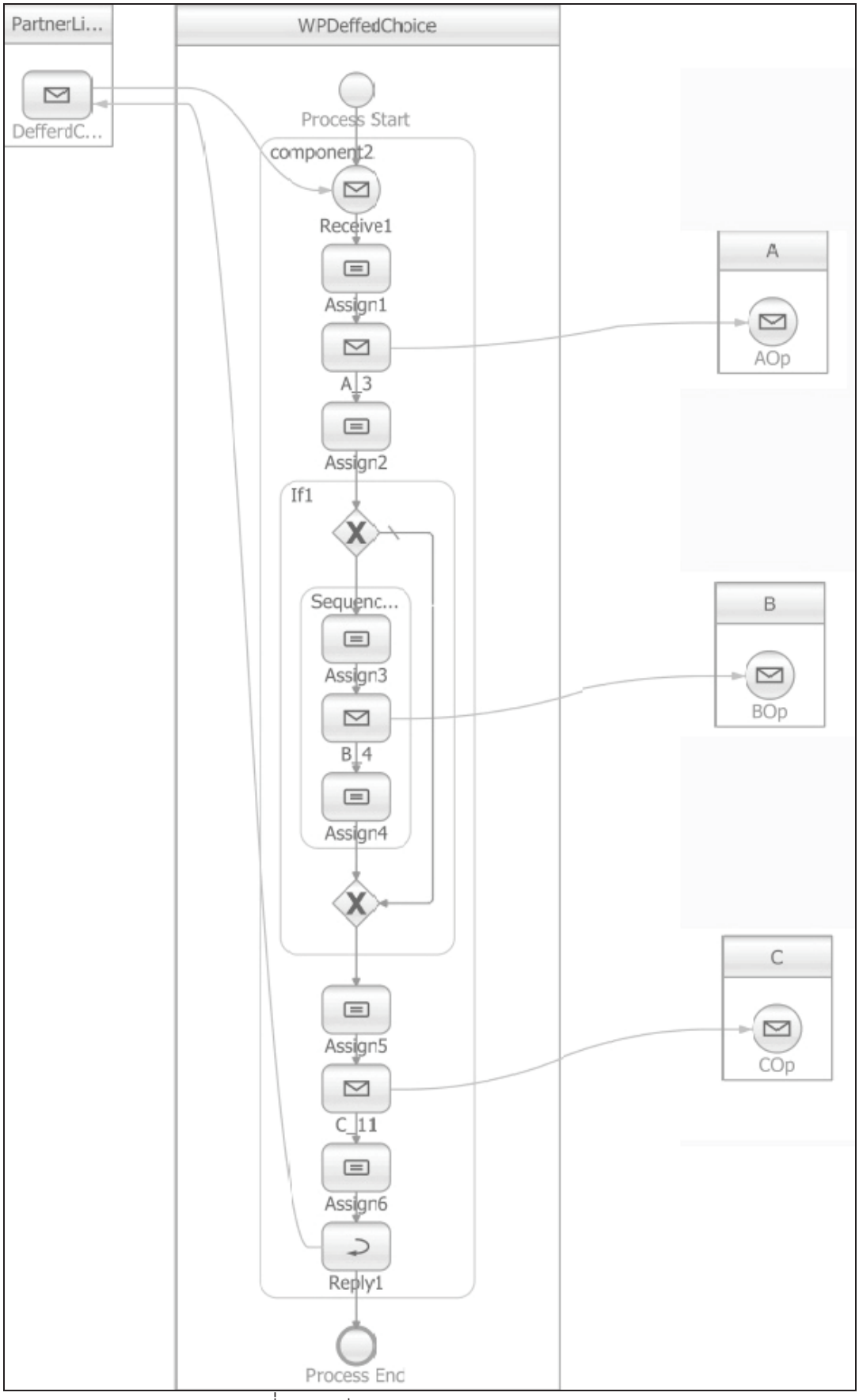


ภาพที่ ค-14 กระแสงานยอวล์แบบรูป Deferred Choice

ตารางที่ ค-7 กรณีทดสอบกระแสงานยอวล์แบบรูป Deferred Choice

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส A ,สามารถทำงานเซอร์วิส B โดยเวลาปฏิบัติการของกระแสงานน้อยกว่า เวลาที่กำหนดใน Set task timer แล้วเรียกเซอร์วิส C และจบการทำงาน	✓
2	ลำดับเรียกเซอร์วิสในกระแสงาน เซอร์วิส A ,ไม่สามารถทำงานเซอร์วิส B โดยเวลาปฏิบัติการของกระแสงานมากกว่า เวลาที่กำหนดใน Set task timer แล้วเรียกเซอร์วิส C และจบการทำงาน	✓

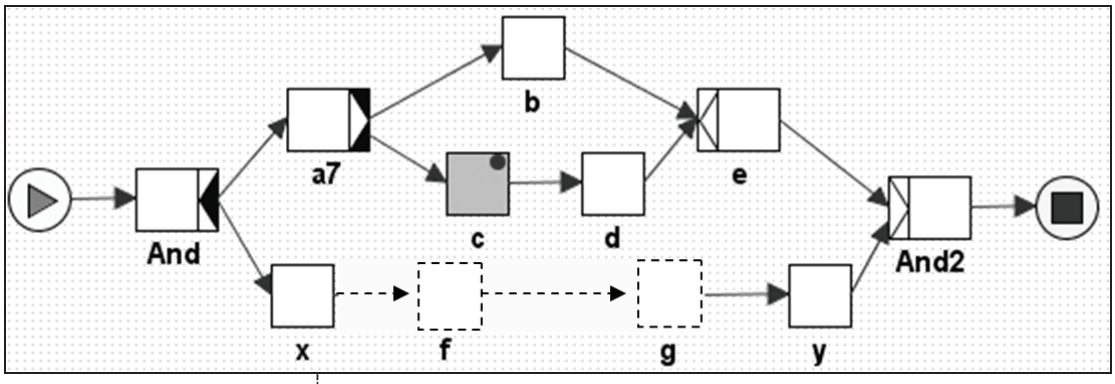
(หมายเหตุ เครื่องหมาย ✓หมายถึงได้ผลลัพธ์ตรงตามที่คาดหวัง)



ภาพที่ ค-15 บีเพลของแบบรูป Deferred Choice

8 ทดสอบกระแสงานยอวล์แบบรูป Cancel Activity

แผนภาพกระแสงานยอวล์ที่ใช้ทดสอบ ดังภาพที่ ค-16 โดยเป็นกระแสงานที่มีลักษณะตรงกับแบบรูป Cancel Activity คือมีเซอร์วิสที่เป็น Cancellation เมื่อเกิดการทำงานขึ้นสามารถยกเลิกการทำงานของเซอร์วิสที่เป็น cancellation set ของเซอร์วิส Cancellation ได้ โดยจากตัวอย่างกระแสงานยอวล์ทดสอบที่เซอร์วิส c เป็น Cancellation และ ที่เซอร์วิส f และ เซอร์วิส g เป็น Cancellation set แล้วทำการทดสอบเครื่องมือโดยนำโครงร่างบีเพลที่เครื่องมือสร้างขึ้นใส่รายละเอียดเพิ่มเติม ดังภาพที่ ค-16 และทดสอบตามกรณีทดสอบตามตารางที่ ค-8



ภาพที่ ค-16 กระแสงานยอวล์แบบรูป Cancel Activity

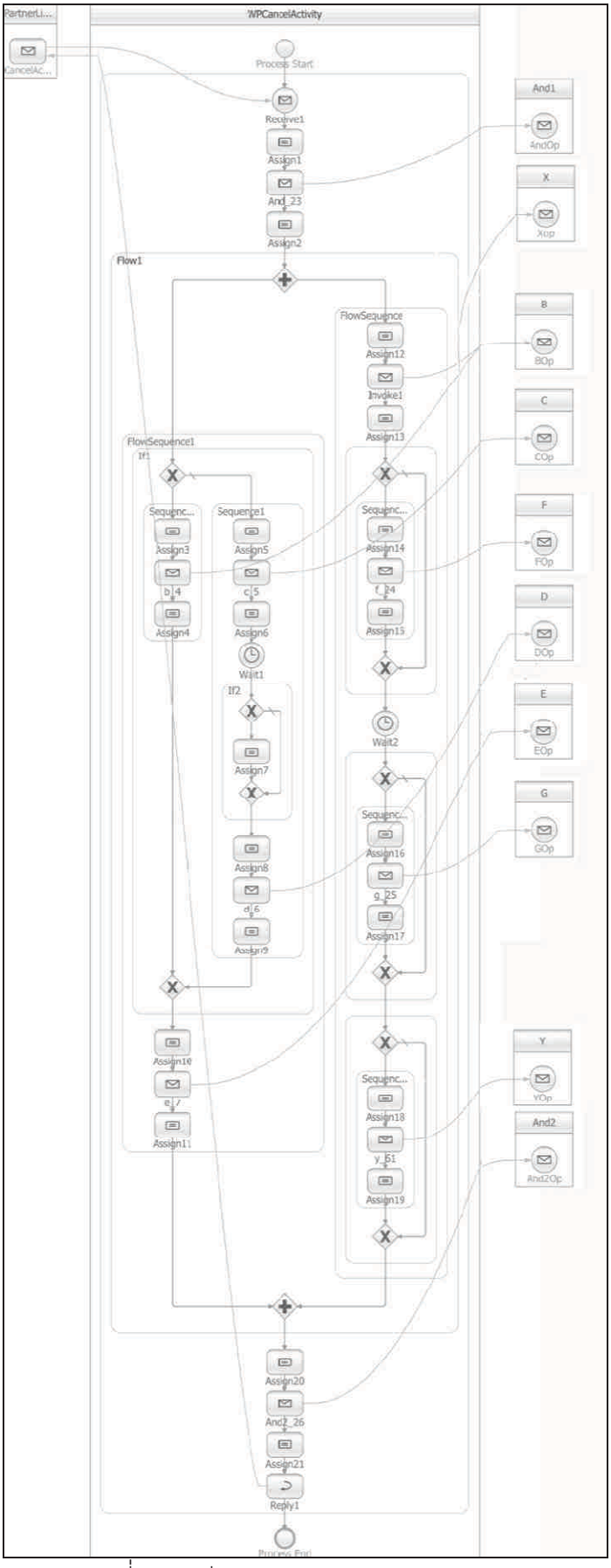
ตารางที่ ค-8 กรณีทดสอบกระแสงานยอวล์แบบรูป Cancel Activity

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
1	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส And ,เซอร์วิส a7 , เซอร์วิส b ,เซอร์วิส e ,เซอร์วิส x ,เซอร์วิส f ,เซอร์วิส g ,เซอร์วิส y , เซอร์วิส And2 และจบการทำงาน	✓
2	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส And ,เซอร์วิส x , เซอร์วิส a7 ,เซอร์วิส f ,เซอร์วิส c ,เซอร์วิส d ,เซอร์วิส y ,เซอร์วิส e , เซอร์วิส And2 และจบการทำงาน	✓
3	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส And ,เซอร์วิส x , เซอร์วิส a7 ,เซอร์วิส f ,เซอร์วิส g ,เซอร์วิส c ,เซอร์วิส d ,เซอร์วิส e , เซอร์วิส y ,เซอร์วิส And2 และจบการทำงาน	✓
4	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส And ,เซอร์วิส x , เซอร์วิส a7 ,เซอร์วิส c ,เซอร์วิส y ,เซอร์วิส e ,เซอร์วิส And2 และจบการทำงาน	✓

ตารางที่ ค-8 กรณีทดสอบกระแสงานยอวล์แบบรูป Cancel Activity(ต่อ)

รหัส	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์จริง
5	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส And ,เซอร์วิส a7 , เซอร์วิส x ,เซอร์วิส f ,เซอร์วิส c ,เซอร์วิส d ,เซอร์วิส y ,เซอร์วิส e , เซอร์วิส And2 และจบการทำงาน	✓
6	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส And ,เซอร์วิส a7 , เซอร์วิส x ,เซอร์วิส f ,เซอร์วิส g ,เซอร์วิส c ,เซอร์วิส d ,เซอร์วิส e , เซอร์วิส y,เซอร์วิส And2 และจบการทำงาน	✓
7	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส And ,เซอร์วิส a7 , เซอร์วิส x ,เซอร์วิส c ,เซอร์วิส y ,เซอร์วิส e ,เซอร์วิส And2 และจบ การทำงาน	✓
8	ลำดับการเรียกเซอร์วิสในกระแสงาน เซอร์วิส And ,เซอร์วิส x , เซอร์วิส a7 ,เซอร์วิส b ,เซอร์วิส e ,เซอร์วิส f ,เซอร์วิส g ,เซอร์วิส y , เซอร์วิส And2 และจบการทำงาน	✓

(หมายเหตุ เครื่องหมาย ✓หมายถึงได้ผลลัพธ์ตรงตามที่คาดหวัง)



ภาพที่ ค-17 บีเพลของแบบรูป Cancel Activity

ประวัติผู้เขียนวิทยานิพนธ์

นายสิทธิพงศ์ พรอุดมทรัพย์ เกิดเมื่อวันที่ 15 พฤศจิกายน พ.ศ. 2527 ที่อำเภอบางมูลนาก จังหวัดพิจิตร สำเร็จการศึกษาในหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร ในปีพ.ศ. 2551 เข้าศึกษาต่อในระดับปริญญามหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย