การขยายความสามารถเอดีแอลสำหรับการตรวจทานแผนภาพกิจกรรม

นายชินพัฒน์ แก้วชินพร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

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.

ENHANCEMENT OF ADL FOR ACTIVITY DIAGRAM REVIEW

Mr. Chinnapat Kaewchinporn

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 2012

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การขยายความสามารถเอดีแอลสำหรับการตรวจทานแผนภาพกิจกรรม
โดย	นายชินพัฒน์ แก้วชินพร
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

…… ประธานกรรมการ
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

…….. อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์)

…… กรรมการภายนอกมหาวิทยาลัย
(ดร.ภาสกร อภิรักษ์วรพินิต)

ชินพัฒน์ แก้วชินพร: การขยายความสามารถเอดีแอลสำหรับการตรวจทานแผนภาพกิจกรรม. (ENHANCEMENT OF ADL FOR ACTIVITY DIAGRAM REVIEW) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.ญาใจ ลิ่มปิยะกรณ์, 85 หน้า.

แผนภาพกิจกรรมเป็นการนำเสนอเชิงกราฟิก เพื่ออธิบายกระบวนการปฏิบัติงานและสิ่งที่เกี่ยวข้องที่ใช้ในแต่ละขั้นตอนภายในระบบ เพื่อให้เกิดความเข้าใจและการสื่อสารที่ถูกต้องตรงกัน มาตรฐานยูเอ็มแอลจึงเป็นสิ่งจำเป็นสำหรับกำหนดรูปแบบการใช้งานให้มีความสอดคล้องถูกต้องตรงกัน งานวิจัยนี้ได้นำเสนอแนวทางการตรวจทานแผนภาพกิจกรรมยูเอ็มแอลอัตโนมัติ บนพื้นฐานภาษาจำเพาะโดเมนชื่อว่า ภาษาอธิบายการกระทำ หรือเอดีแอล โดยมีวัตถุประสงค์เพื่อตรวจสอบการใช้งานที่ผิดและการใช้สัญกรณ์ที่ไม่ถูกต้อง ข้อมูลนำเข้าคือแผนภาพกิจกรรมที่อยู่ในรูปแบบเอ็กซ์เอ็มไอ เนื่องจากความหลากหลายของรูปแบบเอ็กซ์เอ็มไอ กระบวนการตรวจทานจึงเริ่มต้นโดยการแปลงไฟล์ต้นฉบับเอ็กซ์เอ็มไอให้เป็นรูปแบบเดียวกันตามที่ได้กำหนดขึ้นในงานวิจัยนี้ ขั้นตอนถัดไปคือ ดำเนินการสร้างเอดีแอลสคริปต์โดยใช้สารสนเทศที่สกัดได้จากไฟล์ข้อมูลนำเข้าเอ็กซ์เอ็มไอ สคริปต์เอดีแอลที่สร้างขึ้นจะถูกทวนสอบกับเงื่อนไขข้อจำกัดที่กำหนดไว้ในมาตรฐานยูเอ็มแอล 2.4.1 ซึ่งจะได้ผลลัพธ์ คือ รายงานการตรวจสอบ ในกรณีที่แผนภาพกิจกรรมถูกต้อง เอดีแอลสคริปต์นั้นจะถูกแจงส่วนเพื่อสร้างแบบจำลองความหมายของเอดีแอล ซึ่งเป็นผลลัพธ์สุดท้ายของระบบที่พัฒนาขึ้น การสาธิตผลการทำงานของวิธีการที่นำเสนอได้กระทำผ่าน 8 กรณีทดสอบ ปัจจุบัน ซอฟต์แวร์ต้นแบบที่พัฒนาขึ้นสามารถตรวจทานแผนภาพกิจกรรมที่สร้างขึ้นโดยใช้อัลโกยูเอ็มแอลและโมเดลลิโอเท่านั้น

ภาควิชา วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต .....
สาขาวิชา วิศวกรรมซอฟต์แวร์	ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....
ปีการศึกษา 2555	

5170266621 : MAJOR SOFTWARE ENGINEERING
KEYWORDS : DOMAIN SPECIFIC LANGUAGE / ACTIVITY DIAGRAM / VERIFICATION / PROCESS IMPROVEMENT

CHINNAPAT KAEWCHINPORN: ENHANCEMENT OF ADL FOR ACTIVITY DIAGRAM REVIEW: ASSOC. PROF. YACHAI LIMPIYAKORN, Ph.D. 85 pp.

The UML activity diagram is graphical presentation that describes the operational process and related causes used in each stage of the system. For understanding and accurate communication, the UML standard is required for determining the congruent and consistent format application. To detect the misconception and incorrect notation, this research presents an automation approach to reviewing UML activity diagrams based on a domain specific language, called Action Description Language (ADL). The input is the UML activity diagram in the XMI format. Due to the variations of XMI formats, the review process starts with the standardization of the XMI source file. Next, the ADL script will be created using the information extracted from the XMI file. The ADL script will then be verified against the UML constraints defined in the UML standard 2.4.1. The inspection result will be reported. In case of valid activity diagrams, the ADL scripts will be parsed to the ADL semantic model as the final output from the system. The demonstration of the proposed method was performed via eight cases. Currently, the implemented prototype can review only the activity diagrams created by ArgoUML and Modelio.

Department : Computer Engineering Student's Signature .....
Field of Study : Software Engineering Advisor's Signature .....
Academic Year : 2012 .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีจากความช่วยเหลือและสนับสนุนของบุคคลหลายท่าน ประกอบด้วย รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งเป็นผู้เสียสละเวลาในการแนะแนวทางพัฒนา ชี้ให้เห็นถึงปัญหา และคอยแก้ไขตรวจสอบความเรียบร้อยของงานมาโดยตลอด และคณะกรรมการสอบวิทยานิพนธ์ ประกอบด้วย ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุลและดร.ภาสกร อภิรักษ์วรพินิต ซึ่งเป็นผู้ให้คำแนะนำและชี้จุดบกพร่องที่ควรแก้ไข ข้าพเจ้าจึงขอกราบขอบพระคุณเป็นอย่างยิ่งในความกรุณาของทุกท่านไว้ ณ ที่นี้

ท้ายที่สุด ข้าพเจ้าขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และครอบครัว สำหรับกำลังใจที่มีค่ายิ่ง รวมถึงขอขอบพระคุณนิสิตรุ่นพี่สาขาวิศวกรรมซอฟต์แวร์ และมิตรสหายที่ให้กำลังใจ ให้การสนับสนุนและความช่วยเหลือในด้านต่างๆ และท่านอื่นๆ ที่มิได้กล่าวชื่อไว้ ณ ที่นี้ที่มีส่วนช่วยให้วิทยานิพนธ์ของข้าพเจ้าสำเร็จไปได้ด้วยดี

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ญ
สารบัญภาพ ..... ฎ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของการวิจัย 2
1.3 ขอบเขตของการวิจัย 2
1.4 ข้อตกลงเบื้องต้น 2
1.5 ข้อจำกัดของการวิจัย 3
1.6 ประโยชน์ที่คาดว่าจะได้รับ 3
1.7 วิธีดำเนินงานวิจัย 3
1.8 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์ 4
1.9 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5
2.1 ทฤษฎีที่เกี่ยวข้อง 5
2.2 งานวิจัยที่เกี่ยวข้อง 14
บทที่ 3 วิธีดำเนินการวิจัย 18
3.1 แนวคิดวิธีดำเนินการวิจัย 18
3.2 การแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน 18
3.3 การแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล 26
3.4 การตรวจทานแผนภาพกิจกรรมและสร้างรายงานการตรวจสอบ 29
3.5 การแจงส่วนเอกสารเอดีแอล 31

หน้า

บทที่ 4 การออกแบบและพัฒนาระบบ 33
4.1 สถาปัตยกรรมระบบ 33
4.2 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา 33
4.3 การพัฒนาระบบ 34
บทที่ 5 การประเมินและการวัดผล 48
5.1 แนวทางการประเมินผลงานวิจัย 48
5.2 ผลการเปรียบเทียบตัวอย่างที่ 1 48
5.3 ผลการเปรียบเทียบตัวอย่างที่ 2 51
5.4 ผลการเปรียบเทียบตัวอย่างที่ 3 54
5.5 ผลการเปรียบเทียบตัวอย่างที่ 4 59
5.6 ผลการเปรียบเทียบตัวอย่างที่ 5 63
5.7 ผลการเปรียบเทียบตัวอย่างที่ 6 66
5.8 ผลการเปรียบเทียบตัวอย่างที่ 7 71
5.9 ผลการเปรียบเทียบตัวอย่างที่ 8 73
บทที่ 6 สรุปผลการวิจัย และข้อเสนอแนะ 76
6.1 สรุปผลการวิจัย 76
6.2 ข้อจำกัด 76
6.3 แนวทางการวิจัยต่อ 76
รายการอ้างอิง 78
ภาคผนวก 79
ประวัติผู้เขียนวิทยานิพนธ์ 85

สารบัญตาราง

หน้า

ตารางที่ 1 ประเภทและสัญลักษณ์ที่ใช้ในแผนภาพกิจกรรม [4]..... 8

สารบัญภาพ

หน้า

ภาพที่ 1 การกำหนดพฤติกรรมของระบบด้วยบัพควบคุมต่างๆ 5
ภาพที่ 2 โครงสร้างความสัมพันธ์ของกิจกรรมแต่ละประเภท [1] 7
ภาพที่ 3 เมทาโมเดลของแผนภาพกิจกรรมระดับปานกลาง [4] 10
ภาพที่ 4 เมทาโมเดลของภาษาเอดีแอล [4] 11
ภาพที่ 5 ไวยากรณ์แอ็กชัน [4] 11
ภาพที่ 6 ไวยากรณ์ลำดับของแอ็กชันแบบแสดงวัตถุ [4] 12
ภาพที่ 7 ไวยากรณ์ลำดับของแอ็กชันแบบไม่แสดงวัตถุ [4] 12
ภาพที่ 8 ไวยากรณ์ลำดับของแอ็กชันแบบไม่แสดงวัตถุ [4] 13
ภาพที่ 9 ไวยากรณ์การตัดสินใจ [4] 13
ภาพที่ 10 แผนภาพกิจกรรมตัวอย่างที่ 4 [4] 14
ภาพที่ 11 เอดีแอลของแผนภาพกิจกรรมตัวอย่างที่ 4 [4] 15
ภาพที่ 12 การทำงานของระบบ [4] 15
ภาพที่ 13 สถาปัตยกรรมของชุดเครื่องมือ [15] 16
ภาพที่ 14 สถาปัตยกรรมการตรวจสอบความไม่สอดคล้องของแผนภาพยูเอ็มแอล [16] 17
ภาพที่ 15 ภาพรวมการทำงานระบบ 19
ภาพที่ 16 ตัวอย่าง UML:Model 20
ภาพที่ 17 ตัวอย่าง UML:CompositeState.subvertex 21
ภาพที่ 18 ตัวอย่าง UML:StateMachine.transitions 21
ภาพที่ 19 ตัวอย่าง packagedElement 24
ภาพที่ 20 ตัวอย่าง node 24
ภาพที่ 21 ตัวอย่าง edge 25
ภาพที่ 22 โครงสร้างแอ็กชันแบบไม่แสดงวัตถุ 26
ภาพที่ 23 โครงสร้างแอ็กชันแบบแสดงวัตถุ 27
ภาพที่ 24 การสร้างสร้างปลั๊กอินโพรเจกต์ 34
ภาพที่ 25 กำหนดชื่อของโพรเจกต์และรายละเอียดการสนับสนุนของอีคลิปส์ 35
ภาพที่ 26 กำหนดคุณสมบัติของปลั๊กอิน 36

หน้า

ภาพที่ 27 กำหนดเทมเพลตของปลั๊กอิน..... 36
ภาพที่ 28 หน้า Dependencies ของปลั๊กอินส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วย
โอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน..... 37
ภาพที่ 29 หน้า Extensions ของปลั๊กอินส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วย
โอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน..... 37
ภาพที่ 30 การทำงานของคลาสในปลั๊กอินส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วย
โอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน..... 38
ภาพที่ 31 หน้า Dependencies ของปลั๊กอินส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐาน
ให้อยู่ในรูปแบบเอดีแอล..... 39
ภาพที่ 32 หน้า Extensions ของปลั๊กอินส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้
อยู่ในรูปแบบเอดีแอล 39
ภาพที่ 33 การทำงานของคลาสในปลั๊กอินส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐาน
ให้อยู่ในรูปแบบเอดีแอล..... 39
ภาพที่ 34 หน้า Dependencies ของปลั๊กอินสร้างแบบจำลองความหมายของเอดีแอลและ
ตรวจทานแผนภาพกิจกรรม..... 40
ภาพที่ 35 หน้า Extensions ของปลั๊กอินสร้างแบบจำลองความหมายของเอดีแอลและ
ตรวจทานแผนภาพกิจกรรม..... 41
ภาพที่ 36 การทำงานของคลาสในปลั๊กอินสร้างส่วนสร้างแบบจำลองความหมายของเอดีแอล
และตรวจทานแผนภาพกิจกรรม 42
ภาพที่ 37 เริ่มต้นสร้างโพรเจกต์ฟีเจอร์..... 42
ภาพที่ 38 กำหนดรายละเอียดของฟีเจอร์..... 43
ภาพที่ 39 กำหนดปลั๊กอินอ้างอิง..... 44
ภาพที่ 40 สร้างไซต์อัพเดทโพรเจกต์..... 44
ภาพที่ 41 กำหนดชื่อของไซต์อัพเดทโพรเจกต์ 45
ภาพที่ 42 หน้าหลักของตัวสร้างไซต์อัพเดท..... 45
ภาพที่ 43 แพ็คเกจสำหรับติดตั้งในเว็บไซต์ 45
ภาพที่ 44 เมนูติดตั้งซอฟต์แวร์ใหม่..... 46
ภาพที่ 45 ส่วนติดตั้งปลั๊กอินเพิ่มเติม..... 46

หน้า

ภาพที่ 46 รายละเอียดปลั๊กอินทั้งหมดที่ติดตั้งในซอฟต์แวร์ 47
ภาพที่ 47 การใช้งานปลั๊กอินทั้งหมดที่พัฒนาขึ้น 47
ภาพที่ 48 แผนภาพกิจกรรมตัวอย่างที่ 1 [3] 49
ภาพที่ 49 แผนภาพกิจกรรมตัวอย่างที่ 1 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล 49
ภาพที่ 50 แผนภาพกิจกรรมตัวอย่างที่ 1 จากซอฟต์แวร์โมเดลลิโอ 49
ภาพที่ 51 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 1 50
ภาพที่ 52 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 1 51
ภาพที่ 53 แผนภาพกิจกรรมตัวอย่างที่ 2 [3] 51
ภาพที่ 54 แผนภาพกิจกรรมตัวอย่างที่ 2 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล 52
ภาพที่ 55 แผนภาพกิจกรรมตัวอย่างที่ 2 จากซอฟต์แวร์โมเดลลิโอ 52
ภาพที่ 56 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 2 53
ภาพที่ 57 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 2 53
ภาพที่ 58 แผนภาพกิจกรรมตัวอย่างที่ 3 [3] 54
ภาพที่ 59 แผนภาพกิจกรรมตัวอย่างที่ 3 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล 55
ภาพที่ 60 แผนภาพกิจกรรมตัวอย่างที่ 3 จากซอฟต์แวร์โมเดลลิโอ 56
ภาพที่ 61 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 3 58
ภาพที่ 62 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 3 59
ภาพที่ 63 แผนภาพกิจกรรมตัวอย่างที่ 4 [3] 59
ภาพที่ 64 แผนภาพกิจกรรมตัวอย่างที่ 4 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล 60
ภาพที่ 65 แผนภาพกิจกรรมตัวอย่างที่ 4 จากซอฟต์แวร์โมเดลลิโอ 60
ภาพที่ 66 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 4 62
ภาพที่ 67 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 4 62
ภาพที่ 68 แผนภาพกิจกรรมตัวอย่างที่ 5 [3] 63
ภาพที่ 69 แผนภาพกิจกรรมตัวอย่างที่ 5 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล 63
ภาพที่ 70 แผนภาพกิจกรรมตัวอย่างที่ 5 จากซอฟต์แวร์โมเดลลิโอ 64
ภาพที่ 71 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 5 65
ภาพที่ 72 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 5 65
ภาพที่ 73 แผนภาพกิจกรรมตัวอย่างที่ 6 [3] 66

หน้า

ภาพที่ 74 แผนภาพกิจกรรมตัวอย่างที่ 6 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล 67
ภาพที่ 75 แผนภาพกิจกรรมตัวอย่างที่ 6 จากซอฟต์แวร์โมเดลลิโอ 67
ภาพที่ 76 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 6 70
ภาพที่ 77 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 6..... 71
ภาพที่ 78 แผนภาพกิจกรรมตัวอย่างที่ 7 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล 72
ภาพที่ 79 แผนภาพกิจกรรมตัวอย่างที่ 7 จากซอฟต์แวร์โมเดลลิโอ 72
ภาพที่ 80 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 7..... 73
ภาพที่ 81 แผนภาพกิจกรรมตัวอย่างที่ 8 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล 74
ภาพที่ 82 แผนภาพกิจกรรมตัวอย่างที่ 8 จากซอฟต์แวร์โมเดลลิโอ 74
ภาพที่ 83 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 8..... 75
ภาพที่ 84 หน้า plugin.xml ของปลั๊กอินส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วย
โอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน..... 81
ภาพที่ 85 หน้า plugin.xml ของปลั๊กอินส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐาน
ให้อยู่ในรูปแบบเอดีแอล..... 83
ภาพที่ 86 หน้า plugin.xml ของปลั๊กอินส่วนสร้างแบบจำลองความหมายของเอดีแอลและ
ตรวจทานแผนภาพกิจกรรมมีดังนี้..... 84

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ยูเอ็มแอลเป็นเครื่องมือหนึ่งที่สำคัญที่ช่วยในการอธิบายพฤติกรรมของระบบ หน้าที่ของแผนภาพยูเอ็มแอล คือ ช่วยในการสื่อความหมายของส่วนประกอบต่างๆ ภายในระบบ อธิบายพฤติกรรมและฟังก์ชันการทำงานของวัตถุในระบบ หรือแสดงความสัมพันธ์ระหว่างวัตถุแต่ละวัตถุกับวัตถุอื่นๆ ภายในระบบและระบบอื่นๆ ปัจจุบันเพื่อให้การออกแบบแผนภาพยูเอ็มแอลมีคุณภาพและความถูกต้อง จึงต้องมีมาตรฐานเป็นสิ่งที่ในการควบคุมในการตีความและสัญกรณ์ โดยมาตรฐานสากลที่ได้รับการยอมรับในการออกแบบยูเอ็มแอลคือข้อกำหนดยูเอ็มแอลที่กำหนดโดยโอเอ็มจี [1], [2], [3]

จากงานวิจัยหลายๆ ฉบับชี้ให้เห็นว่าการออกแบบแผนภาพยูเอ็มแอลขององค์กรทั้งขนาดใหญ่ หรือขนาดเล็กต่างๆ นั้น ยังขาดความเอาใจใส่ในการนำเอาข้อกำหนดยูเอ็มแอลมาใช้งาน ทำให้แผนภาพที่ได้นั้นไม่มีความถูกต้องตามความหมายที่ต้องการ และข้อมูลหรือแผนภาพเกิดความไม่สอดคล้องกัน งานออกแบบที่ได้จึงพร่องคุณภาพ ส่งผลให้การทำงานในระยะถัดไปของโครงการมีโอกาสที่จะทวีความผิดพลาดมากขึ้น สุดท้ายแล้วอาจทำให้การดำเนินโครงการซอฟต์แวร์เกิดความล้มเหลวได้

จากการศึกษาข้อมูลเกี่ยวกับการตรวจสอบแผนภาพตามข้อกำหนดยูเอ็มแอลทำให้เห็นปัญหาสำคัญที่ทำให้นักออกแบบละเลยการนำข้อกำหนดยูเอ็มแอลเข้ามาใช้งานเนื่องจากประเด็นสำคัญ 3 ประการ ดังนี้

1) ข้อกำหนดมาตรฐานมีความเข้าใจยากและไม่ชัดเจนสำหรับนักออกแบบ
2) ไม่มีเครื่องมือที่ช่วยอำนวยความสะดวกในการตรวจสอบ
3) ในการพัฒนาแผนภาพยูเอ็มแอลซึ่งใช้สัจกรณ์ในการวาดแผนภาพ ทำให้ต่างคนต่างมีวิธีเขียนที่แตกต่างกันไป ส่งผลให้เกิดการตีความผิดพลาด และไม่สอดคล้องระหว่างแผนภาพเกิดขึ้น

ข้อกำหนดรูปนัยเป็นหนึ่งในวิธีที่นักวิจัยจำนวนมากนิยมนำมาใช้ในการแก้ไขปัญหาที่กล่าวมา เพราะสามารถกำหนดรายละเอียดการดำเนินงานได้มากกว่าการสร้างแบบจำลองด้วยสัญกรณ์กราฟิกเพียงอย่างเดียว แต่ปัญหาที่สำคัญของข้อกำหนดรูปนัยคือข้อกำหนดรูปนัยนั้น มักถูกกำหนดด้วยภาษาคณิตศาสตร์ที่นักออกแบบระบบไม่เข้าใจ ยากต่อการเรียนรู้ และการนำไปใช้งานจริง

เอดีแอลหรือข้อกำหนดรูปนัยในลักษณะของดีเอสแอล [4] จึงเป็นตัวเลือกหนึ่งที่น่าสนใจ สามารถช่วยการเกิดป้องกันมโนทัศน์ที่ผิดและความไม่ตรงกันของพฤติกรรมในแผนภาพกิจกรรม และมีความง่ายดายในการเข้าใจและใช้งาน แต่ปัญหาที่สำคัญของเอดีแอลคือ การไม่สามารถนำเอกสารยูเอ็มแอลที่อยู่ในองค์กรอยู่แล้วมาใช้ร่วมได้ การพัฒนาเอดีแอลนั้นจำเป็นต้องเริ่มต้นที่การออกแบบใหม่เท่านั้น และการประยุกต์ใช้งานเอดีแอลทำได้ยาก

งานวิจัยนี้นำเสนอวิธีการตรวจสอบแผนภาพกิจกรรมอัตโนมัติ ซึ่งพัฒนาบนภาษาจำเพาะโดเมน ชื่อว่าภาษาอธิบายการกระทำ หรือเอดีแอล เพื่อตรวจสอบความเข้าใจที่ไม่ตรงกันและสัญกรณ์ที่ไม่ถูกต้องของแผนภาพกิจกรรม ปัจจุบันซอฟต์แวร์ต้นแบบสามารถตรวจทานแผนภาพกิจกรรมที่พัฒนาขึ้น โดยอัลโก้ยูเอ็มแอล [5] และโมเดลลิโอ [6] เท่านั้น ผ่านมาตรฐานเอ็กซ์เอ็มไอ [7], [8] ผลลัพธ์สุดท้ายที่ได้คือแบบจำลองความหมายของเอดีแอล สำหรับการพัฒนานั้นจะได้ผลลัพธ์สุดท้าย คือ ปลั๊กอินอีคลิปส์ [9] และแบบแผนในการพัฒนาต่อยอดของเอดีแอล ซึ่งจะช่วยให้เอดีแอลมีบทบาทที่โดดเด่นและเหมาะสมกับธุรกิจยิ่งขึ้น และช่วยให้การพัฒนาต่อยอดทำได้อย่างสะดวกยิ่งขึ้น

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อขยายความสามารถเอดีแอลสำหรับการตรวจทานแผนภาพกิจกรรม
2. เพื่อเพิ่มศักยภาพของเอดีแอลให้มีความยืดหยุ่น และเหมาะสมในการใช้งานกับระบบงานทางธุรกิจที่โดยทั่วไปต้องวาดแผนภาพกิจกรรมในการปฏิบัติงาน

1.3 ขอบเขตของการวิจัย

1. ซอฟต์แวร์ที่พัฒนาขึ้นสามารถแปลงแผนภาพกิจกรรมให้อยู่ในรูปแบบเอดีแอลได้
2. ซอฟต์แวร์ที่พัฒนาขึ้นรองรับการต่อผสานสำหรับการพัฒนาเพิ่มเติมในอนาคต
3. ซอฟต์แวร์ที่พัฒนาขึ้นสามารถทวนสอบและตรวจสอบความสมเหตุสมผลของข้อมูลระหว่างแผนภาพกิจกรรม ดังนี้
    - ตรวจสอบไวยากรณ์ของแผนภาพกิจกรรมตามข้อกำหนดยูเอ็มแอล
    - ตรวจสอบความสมเหตุสมผลระหว่างวัตถุและกิจกรรมภายในแผนภาพกิจกรรม

1.4 ข้อตกลงเบื้องต้น

1. แผนภาพกิจกรรมเป็นไปตามข้อกำหนดของยูเอ็มแอลเวอร์ชัน 2.4.1

2. เอกสารเอ็กซ์เอ็มไอเป็นไปตามข้อกำหนดของเอ็กซ์เอ็มไอเวอร์ชัน 2.0
3. ซอฟต์แวร์สามารถรายงานข้อผิดพลาดหากพบว่าการร้อยกระบวนการจากข้อกำหนดรูปนัยมีการไม่ต้องกันของกระบวนการเกิดขึ้น หรือหากพบว่าเอกสารเอ็กซ์เอ็มไอนั้นไม่ตรงตามข้อจำกัดของการวิจัย

1.5 ข้อจำกัดของการวิจัย

1. ซอฟต์แวร์รองรับการแปลงแผนภาพกิจกรรมจากซอฟต์แวร์ออกแบบยูเอ็มแอลโอเพนซอร์สอัลโก้ยูเอ็มแอล 0.34 และโมเดลลิโอ 2.2.1 เท่านั้น
2. เอดีแอลรองรับเฉพาะแผนภาพกิจกรรมเชิงธุรกิจเท่านั้น กล่าวคือยังไม่รองรับแผนภาพกิจกรรมเชิงโครงสร้างหรือแผนภาพกิจกรรมที่ใช้สำหรับการเขียนหรือพัฒนาโปรแกรม
3. ข้อมูลในรูปแบบของเอ็กซ์เอ็มไอที่ได้จากระบบที่พัฒนาขึ้น ไม่สามารถใช้เป็นข้อมูลนำเข้าให้กับซอฟต์แวร์อื่นได้
4. ซอฟต์แวร์รองรับการแปลงแผนภาพกิจกรรมระดับปานกลางเท่านั้น

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เครื่องมือในการทวนสอบและตรวจสอบความสมเหตุสมผลของแผนภาพกิจกรรมที่พัฒนาบนซอฟต์แวร์ออกแบบยูเอ็มแอลโอเพนซอร์สกับข้อกำหนดยูเอ็มแอล
2. ได้เครื่องมือในการแปลงแผนภาพกิจกรรมให้อยู่ในรูปแบบภาษาเอดีแอล
3. ได้ระบบแนะนำการแก้ไขแผนภาพกิจกรรมเพื่อให้สอดคล้องตามข้อกำหนดยูเอ็มแอล

1.7 วิธีดำเนินงานวิจัย

1. ศึกษาและทำความเข้าใจการแปลงแผนภาพกิจกรรมให้อยู่ในรูปแบบภาษาธรรมชาติ
2. ศึกษาและทำความเข้าใจทฤษฎีที่เกี่ยวข้อง ได้แก่ ข้อกำหนดยูเอ็มแอล, ภาษาเอดีแอล, การแปลงแผนภาพ และการทวนสอบและการตรวจสอบความสมเหตุสมผลของข้อมูลระหว่างแผนภาพกิจกรรมและข้อกำหนดยูเอ็มแอล
3. ศึกษาและทำความเข้าใจงานวิจัยที่เกี่ยวข้อง
4. ศึกษาและทดลองใช้เครื่องมือในการสร้างแผนภาพกิจกรรมทั่วไป
5. วิเคราะห์และกำหนดภาพรวมของงานวิจัย
6. ออกแบบ ตั้งสมมติฐาน กำหนดเป้าหมาย และกำหนดตัวแปรที่เกี่ยวข้องของงานวิจัย
7. พัฒนาระบบการแปลงแผนภาพกิจกรรมให้อยู่ในรูปแบบเอดีแอล

8. ทดสอบและประเมินผลการวิจัยตามเป้าหมายที่กำหนด
9. สรุปผลการวิจัยและนำผลที่ได้ย้อนรอยไปปรับปรุงระบบเพื่อให้ได้เป้าหมายที่กำหนด
10. ตีพิมพ์ผลงานวิจัย
11. จัดทำวิทยานิพนธ์

1.8 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 6 บท ดังต่อไปนี้ บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการวิจัย ขอบเขตของการวิจัย ประโยชน์ที่คาดว่าจะได้รับและผลงานตีพิมพ์ บทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องของงานวิจัย บทที่ 3 กล่าวถึงวิธีดำเนินการวิจัย บทที่ 4 กล่าวถึง การออกแบบและพัฒนาระบบตามแนวทางการวิจัย บทที่ 5 กล่าวถึงวิธีการประเมินและวัดผลการทดลอง และบทที่ 6 สรุปผลการวิจัย ข้อเสนอแนะและแนวทางสำหรับการวิจัยต่อในอนาคต

1.9 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความวิชาการเรื่อง “Semantic Approach to Verifying Activity Diagrams with a Domain Specific Language”, Chinnapat Kaewchinporn and Yachai Limpiyakorn, in Proceedings of 2012 International Conference on Advanced Software Engineering & Its Applications, Jeju Island, Korea, Nov 28-30, 2012, pp. 466-473.

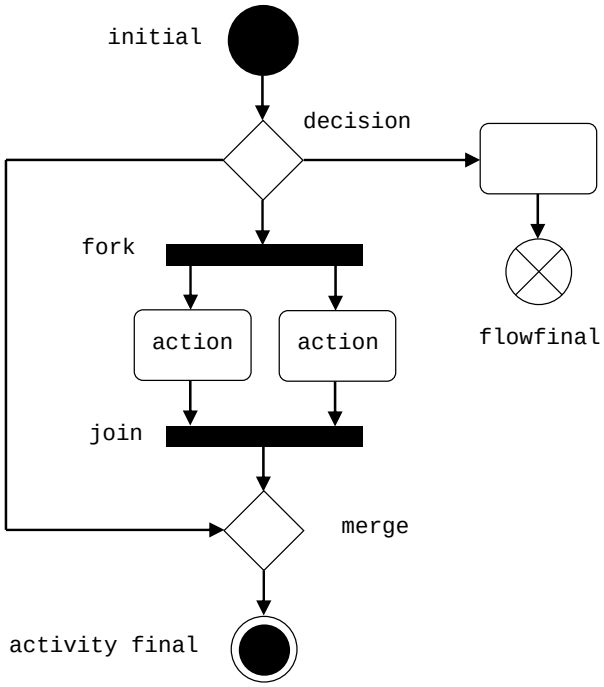
ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความวิชาการเรื่อง “Enhancement of Action Description Language for UML Activity Diagram Review”, Chinnapat Kaewchinporn and Yachai Limpiyakorn, in International Journal of Software Engineering and Its Application, Vol. 7, No. 2, March, 2013, pp. 255-272.

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 แผนภาพกิจกรรม

แผนภาพกิจกรรม คือ แผนภาพที่ใช้ในการอธิบายตรรกะเชิงกระบวนการ กระบวนการทางธุรกิจ และกระแสงาน (Workflow) ต่างๆ ทั้งกระบวนการที่ทำงานเป็นลำดับและทำงานแบบขนาน โดยแผนภาพกิจกรรมหนึ่งๆ จะประกอบด้วยลำดับและเงื่อนไขต่างๆ ที่ใช้ในการควบคุมพฤติกรรมของระบบ โดยมีสายงานควบคุม (Control flow) และสายงานวัตถุ (Object flow) อธิบายลำดับของแอ็กชัน และมีบัพควบคุม (Control node) อธิบายพฤติกรรมของระบบ [3], [4] แสดงได้ดังภาพที่ 1

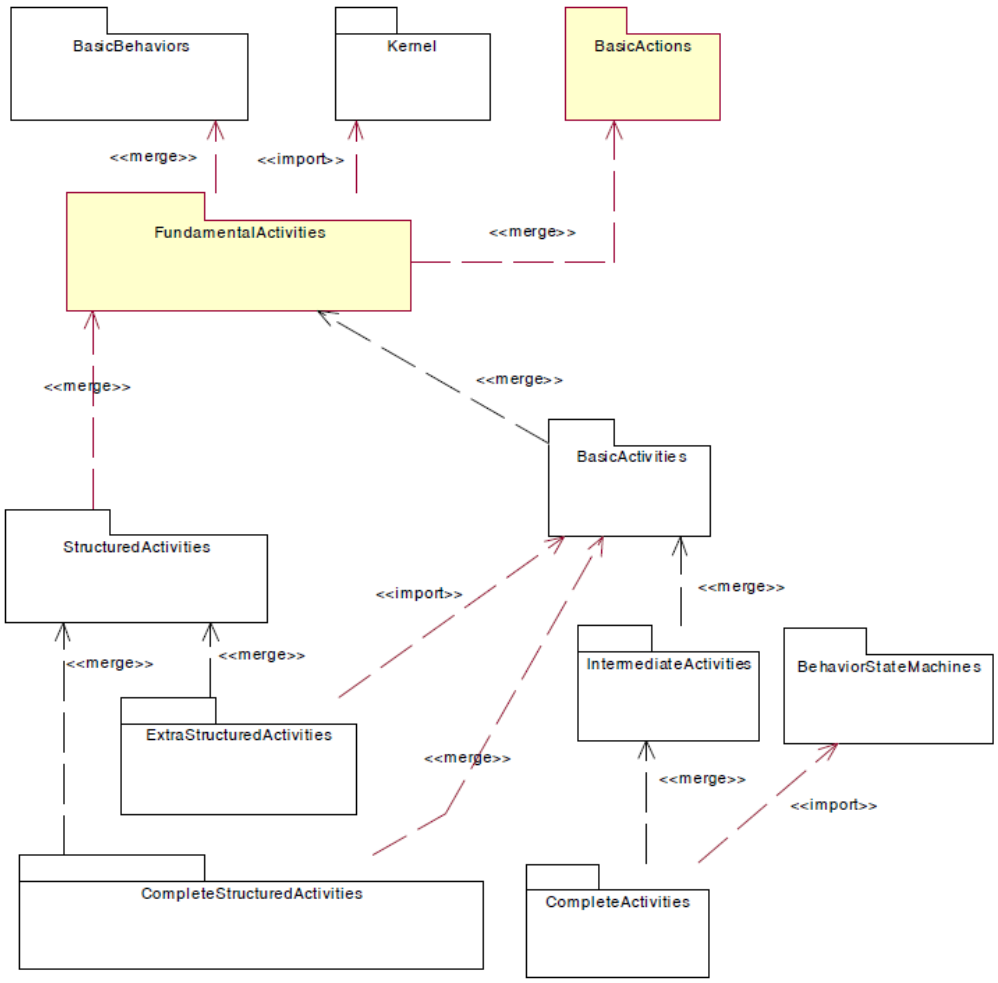


ภาพที่ 1 การกำหนดพฤติกรรมของระบบด้วยบัพควบคุมต่างๆ

แผนภาพกิจกรรมสามารถแบ่งกรอบงานได้ออกเป็น 7 ประเภทดังภาพที่ 2 โดยมีรายละเอียดดังต่อไปนี้ [4]

1. กิจกรรมระดับรากฐาน (Fundamental Activities) ใช้สำหรับกำหนดแอ็กชันหรือส่วนกระทำในกิจกรรมซึ่งอาจมีการจัดกลุ่มแอ็กชันหรือไม่ก็ได้

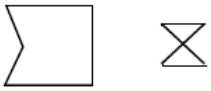
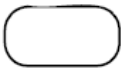
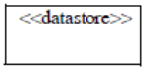
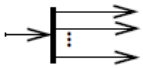
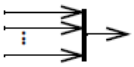
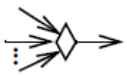
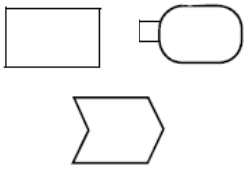
2. กิจกรรมระดับพื้นฐาน (Basic Activities) ใช้สำหรับกำหนดลำดับการทำงานของแอ็กชันในกิจกรรมระดับรากฐาน และมีการกำหนดจุดเริ่มต้นและจุดสิ้นสุดของสายงาน
3. กิจกรรมระดับปานกลาง (Intermediate Activities) ใช้สำหรับกำหนดรายละเอียดลำดับการทำงานที่ได้จากกิจกรรมระดับพื้นฐาน โดยอธิบายถึงรายละเอียดของพฤติกรรมที่เกี่ยวข้องกับระบบ เช่น การอธิบายการทำงานที่สามารถทำพร้อมกันได้ การทำงานที่มีการตัดสินใจหรือเงื่อนไขเข้ามาเกี่ยวข้อง เป็นต้น
4. กิจกรรมระดับสมบูรณ์ (Complete Activities) ใช้สำหรับกำหนดรายละเอียดเพิ่มเติมจากกิจกรรมระดับปานกลาง โดยจะเน้นการกำหนดค่าน้ำหนักบนสายงานข้อมูลต่าง ๆ เพื่อให้มีรายละเอียดการดำเนินงานที่ชัดเจนขึ้น
5. กิจกรรมเชิงโครงสร้าง (Structured Activities) ใช้สำหรับการออกแบบจำลองเพื่อการเขียนหรือพัฒนาโปรแกรมเป็นหลัก โดยจะมีการกำหนดลำดับการทำงาน การวนซ้ำ และเงื่อนไขที่เกี่ยวข้องกับลำดับการทำงานของแอ็กชันที่กำหนดขึ้นในกิจกรรมระดับรากฐาน
6. กิจกรรมเชิงโครงสร้างระดับสมบูรณ์ (Complete Structured Activities) ใช้สำหรับกำหนดรายละเอียดอินพุตและเอาต์พุตของแต่ละแอ็กชันในกิจกรรมเชิงโครงสร้าง
7. กิจกรรมเชิงโครงสร้างระดับพิเศษ (Extra Structured Activities) ใช้สำหรับกำหนดเงื่อนไขพิเศษต่าง ๆ ที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ ซึ่งประกอบด้วยการจัดการสิ่งผิดปกติและการเรียกทำงานแอ็กชันผ่านเงื่อนไขหรือตัวควบคุมเหตุการณ์ต่างๆ ที่กำหนดขึ้น

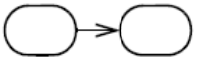
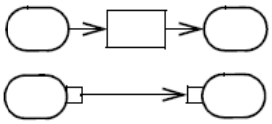
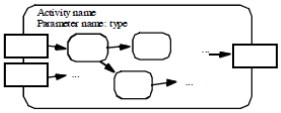
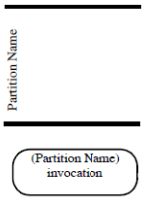
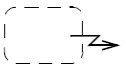
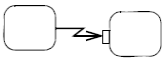
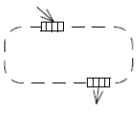
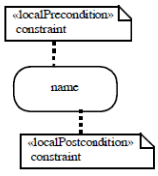


ภาพที่ 2 โครงสร้างความสัมพันธ์ของกิจกรรมแต่ละประเภท [1]

สัญกรณ์กราฟิกที่ใช้ในการออกแบบ ประเภทของสัญกรณ์กราฟิก และความสัมพันธ์ระหว่างสัญกรณ์กราฟิกกับระดับของกรอบงานแผนภาพกิจกรรมที่รองรับ [1], [4] แสดงดังตารางที่ 1

ตารางที่ 1 ประเภทและสัญลักษณ์ที่ใช้ในแผนภาพกิจกรรม [4]

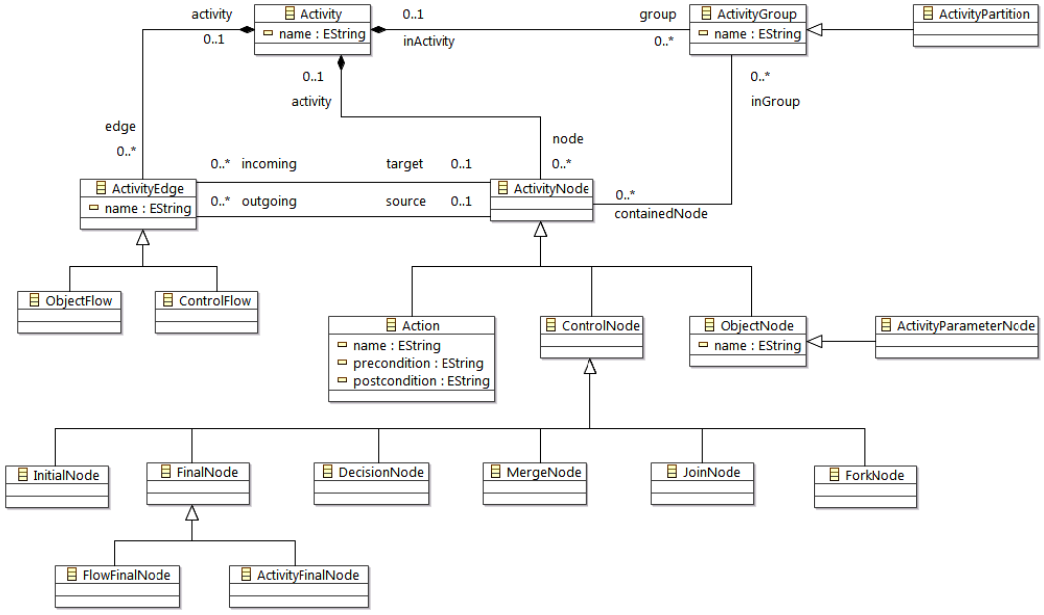
ประเภท	สัญลักษณ์	Fundamental	Basic	Intermediate	Complete	Structured	CompleteStructured	ExtraStructured
NODE								
AcceptEventAction					X			X
Action		X	X	X	X	X	X	X
ActivityFinal			X	X	X	X	X	X
DataStore					X			
DecisionNode				X	X	X	X	X
FlowFinal				X	X	X	X	X
ForkNode				X	X	X	X	X
InitialNode			X	X	X	X	X	X
JoinNode				X	X	X	X	X
MergeNode				X	X	X	X	X
ObjectNode			X	X	X			

ประเภท	สัญลักษณ์	Fundamental	Basic	Intermediate	Complete	Structured	CompleteStructured	ExtraStructured
SendSignalAction					X			X
PATH								
ControlFlow			X	X	X	X	X	X
ObjectFlow			X	X	X			
OTHER ELEMENTS								
Activity		X	X	X	X	X	X	X
ActivityPartition				X	X			
InterruptibleActivityRegion					X			X
ExceptionHandler								X
ExpansionRegion								X
LocalPrecondition, LocalPostcondition		X	X	X	X	X	X	X

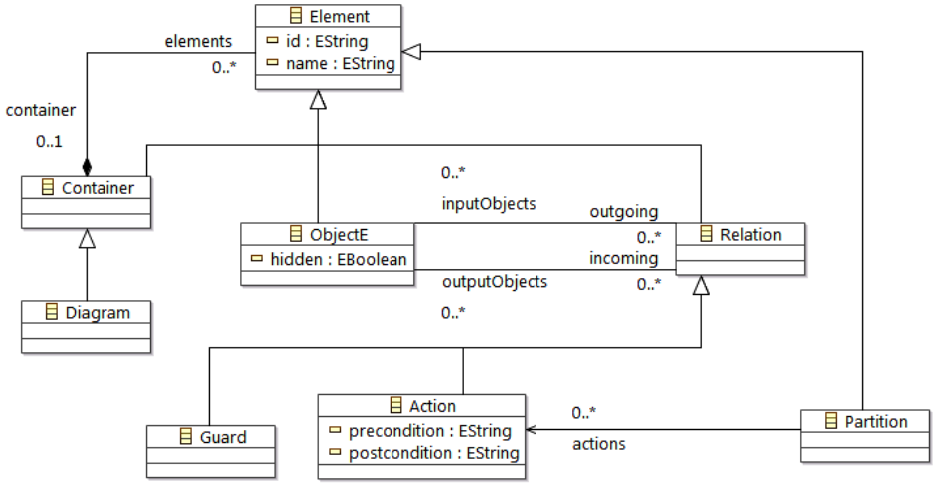
ประเภท	สัญลักษณ์	Fundamental	Basic	Intermediate	Complete	Structured	CompleteStructured	ExtraStructured
ParameterSet					X		X	X

2.1.2 เอดีแอล (Action Description Language–ADL)

เอดีแอลถูกพัฒนาขึ้นในงานวิจัย [4] จัดอยู่ในประเภทภาษาจำเพาะโดเมน (Domain Specific Language) ซึ่งเป็นภาษาข้อกำหนดที่อยู่ในรูปแบบของข้อความโดยเฉพาะเจาะจงสำหรับโดเมนปัญหาหนึ่งๆ วัตถุประสงค์ของภาษาเอดีแอลนั้นเพื่อป้องกันมโนทัศน์ (concept) ที่ผิดและความไม่ตรงกันของพฤติกรรมในแผนภาพกิจกรรม เมทาโมเดลของภาษาเอดีแอลที่เป็นตัวแบบต้นทางเพื่อใช้งานกับตัวแบบปลายทางหรือเมทาโมเดลของแผนภาพกิจกรรมในภาพที่ 3 แสดงได้ดังภาพที่ 4 [4]



ภาพที่ 3 เมทาโมเดลของแผนภาพกิจกรรมระดับปานกลาง [4]

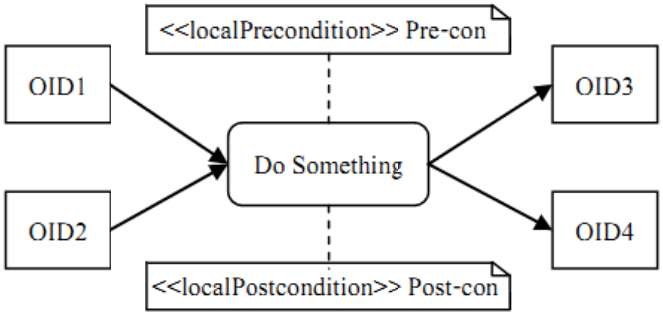


ภาพที่ 4 เมทาโมเดลของภาษาเอดีแอล [4]

ไวยากรณ์ที่สำคัญของเอดีแอลประกอบด้วย 3 ส่วน คือ ไวยากรณ์ของแอ็กชัน ไวยากรณ์ลำดับของแอ็กชัน และไวยากรณ์การตัดสินใจ

แอ็กชัน หมายถึง หน่วยของขั้นตอนหนึ่งๆที่แบ่งแยกไม่ได้ในกิจกรรม โดยที่แต่ละแอ็กชันจึงต้องมีอย่างน้อยหนึ่งเอาต์พุต (ข้อมูลนำออก) เพื่อเป็นหลักฐานการทำงานของแอ็กชัน โดยแอ็กชันอาจมีหรือไม่มีอินพุต (ข้อมลนำเข้า) นอกจากนี้อาจมีหรือไม่มีการกำหนดเงื่อนไขก่อน/หลังการกระทำไวยากรณ์ที่ใช้ในการกำหนดแอ็กชันแสดงได้ดังรูปที่ 5 อธิบายได้ด้วยรหัสต้นฉบับดังนี้:

```
action id
  [name 'string']
  [<- OID1[,OID2...[,OIDN]]] --input objects
  [-> OID3[,OID4...[,OIDN]]] --output objects
  [precondition 'string']
  [postcondition 'string']
End
```

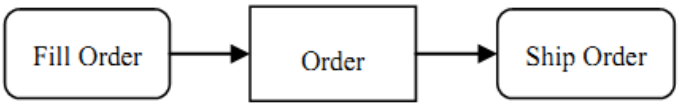


ภาพที่ 5 ไวยากรณ์แอ็กชัน [4]

ลำดับของแอ็กชัน (Sequence of actions) สามารถหาได้จากวัตถุที่ใช้ร่วมกันระหว่างสองแอ็กชัน ซึ่งลำดับของแอ็กชันที่สกัดออกมาได้นั้น จะเป็นการเชื่อมต่อแบบมีทิศทางประกอบด้วยแอ็กชันต้นทางและปลายทาง (ขึ้นอยู่กับว่าวัตถุนั้นเป็นเอาต์พุตหรืออินพุตของแอ็กชัน) สำหรับการกำหนดลำดับของแอ็กชันนั้น สามารถแบ่งออกได้ออกเป็น 2 กรณี ดังนี้

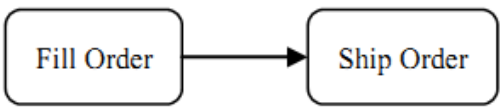
1. ใช้วัตถุเป็นตัวเชื่อมความสัมพันธ์ระหว่างสองแอ็กชัน โดยจะต้องมีการกำหนดวัตถุที่เป็นอินพุตหรือเอาต์พุตของแต่ละแอ็กชัน ดังตัวอย่างภาพที่ 6
2. ระบุความสัมพันธ์ของแอ็กชันโดยตรง ซึ่งถ้ามีการระบุความสัมพันธ์ของแอ็กชันโดยตรงแล้ว วัตถุแบบซ่อน (วัตถุโดยนัย) จะถูกสร้างขึ้นมา และถูกกำหนดเป็นเอาต์พุตของแอ็กชันต้นทางและเป็นอินพุตของแอ็กชันปลายทาง การกำหนดความสัมพันธ์แอ็กชันโดยตรง สามารถกำหนดได้ดังตัวอย่างภาพที่ 7

```
action fillOrder
 -> Order
end
action shipOrder
<- Order
end
```



ภาพที่ 6 ไวยากรณ์ลำดับของแอ็กชันแบบแสดงวัตถุ [4]

```
action fillOrder end
action shipOrder end

fillOrder->shipOrder
```



ภาพที่ 7 ไวยากรณ์ลำดับของแอ็กชันแบบไม่แสดงวัตถุ [4]

บัพตัดสินใจ คือ บัพควบคุมประเภทหนึ่งที่ทำหน้าที่ในการแยกย่อยสายงาน (flow) ออกเป็นหลายๆสายงาน โดยบัพตัดสินใจทำหน้าที่ในการรับโทเคนมาพิจารณาเพื่อเลือกสายงานที่

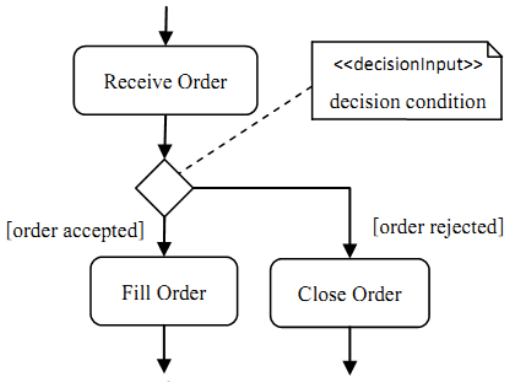
จะดำเนินการต่อไป (เพียงสายงานเดียว) เนื่องจากการตัดสินใจจำเป็นที่จะต้องมีเงื่อนไขอย่างน้อยหนึ่งเงื่อนไขในการตัดสินใจ และอาจมีวิธีการอธิบายขั้นตอนในการตัดสินใจ จึงกำหนดไวยากรณ์สำหรับการตัดสินใจดังภาพที่ 8

```
decision ['input']
  if 'condition1' then id1
  if 'condition2' then id2
  ...
end
```

ภาพที่ 8 ไวยากรณ์ลำดับของแอ็กชันแบบไม่แสดงวัตถุ [4]

การสร้างเงื่อนไขในตัวตัดสินใจ อินพุตที่เกี่ยวข้องในแต่ละแอ็กชันที่จะดำเนินการตามสายงานถัดไปจะต้องมีอินพุตที่เหมือนกัน โดยแอ็กชันที่จะทำงานหลังจากการตัดสินใจจะต้องมีวัตถุอินพุตที่เป็นวัตถุเดียวกันกับวัตถุเอาต์พุตของแอ็กชันก่อนการตัดสินใจ อย่างไรก็ตาม วัตถุที่กำหนดขึ้นมานั้นส่วนใหญ่จะไม่แสดงในแผนภาพกิจกรรม ดังนั้น จึงสามารถใช้วิธีกำหนดความสัมพันธ์แอ็กชันโดยตรงเพื่อเพิ่มความยืดหยุ่นและลดเวลาในการเขียนบทคำสั่ง ดังตัวอย่างภาพที่ 9

```
action receiveOrder end
action fillOrder end
action closeOrder end
receiveOrder->closeOrder
receiveOrder->fillOrder
decision 'decision condition'
  if 'order accepted' then fillOrder
  if 'order rejected' then closeOrder
end
```



ภาพที่ 9 ไวยากรณ์การตัดสินใจ [4]

2.1.3 **เอ็กซ์เอ็มไอ** (XML Metadata Interchange–XMI)

เอ็กซ์เอ็มไอ เป็นมาตรฐานที่ให้รายละเอียดการแลกเปลี่ยนข้อมูลเมทาดาตาสำหรับโปรแกรมเมอร์และผู้ใช้อื่นๆ วัตถุประสงค์ของเอ็กซ์เอ็มไอ คือ ช่วยในการแลกเปลี่ยนข้อมูลตัวแบบสำหรับโปรแกรมเมอร์เมื่อต้องใช้ภาษาหรือเครื่องมือที่ต่างกัน [7], [8]

เอ็กซ์เอ็มไอได้นิยามมุมมองที่เกี่ยวข้องกับวัตถุในเอ็กซ์เอ็มแอล [7] ดังนี้

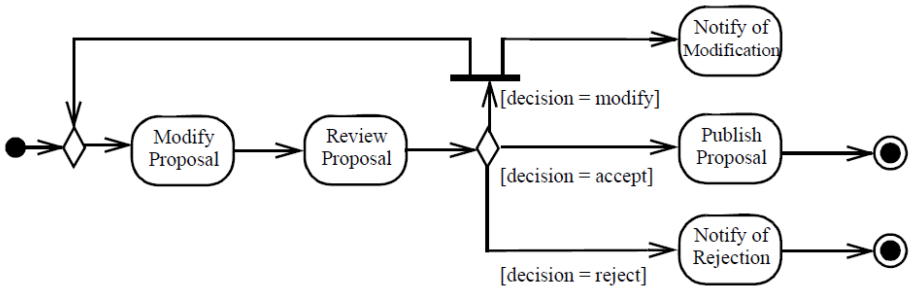
1. เป็นตัวแทนของของวัตถุในแง่ขององค์ประกอบของเอ็กซ์เอ็มแอลและแอทริบิวต์เป็นพื้นฐาน
2. เนื่องจากการเชื่อมต่อกันของวัตถุเอ็กซ์เอ็มไอจึงได้รวมมาตรฐานในการเชื่องโยงวัตถุภายในไฟล์เดียวกันหรือข้ามไฟล์
3. การระบุวัตถุสามารถอ้างอิงได้จากวัตถุอื่นในรูปแบบ ของ “IDs”และ “UUIDs”
4. การตรวจสอบเอ็กซ์เอ็มไอโดยใช้รูปแบบของเอ็กซ์เอ็มแอล

เอ็กซ์เอ็มไอสามารถตรวจสอบได้โดยการระบุ “EBNF product rule” [7] ในการสร้างเอกสารเอ็กซ์เอ็มแอลและรูปแบบซึ่งสามารถแลกเปลี่ยนวัตถุได้อย่างมีประสิทธิภาพ

2.2 **งานวิจัยที่เกี่ยวข้อง**

2.2.1 Formal Specification for Verifying Activity Diagram of Process Flow [4]

ในงานวิจัยนี้นำเสนอภาษาอธิบายการกระทำ หรือเอดีแอล ซึ่งจัดอยู่ในประเภทภาษาจำเพาะโดเมน เพื่อป้องกันมโนทัศน์ที่ผิดและความไม่ตรงกันของพฤติกรรมในแผนภาพกิจกรรม พัฒนาวิธีการอิงวัตถุสำหรับการสร้างแผนภาพกิจกรรมบนบทคำสั่งเอดีแอล และนำเสนอกฎตรวจสอบความสมเหตุสมผลสำหรับเอดีแอลเพื่อป้องกันความไม่ตรงกันของข้อมูลและพฤติกรรมในแผนภาพกิจกรรม รวมทั้งกฎการทวนสอบเพื่อให้แผนภาพถูกสร้างขึ้นอย่างถูกต้องตามข้อกำหนด ตัวอย่างของแผนภาพกิจกรรมและเอดีแอลที่พัฒนาตามแผนภาพกิจกรรมแสดงในภาพที่ 10 และ 11 ตามลำดับ

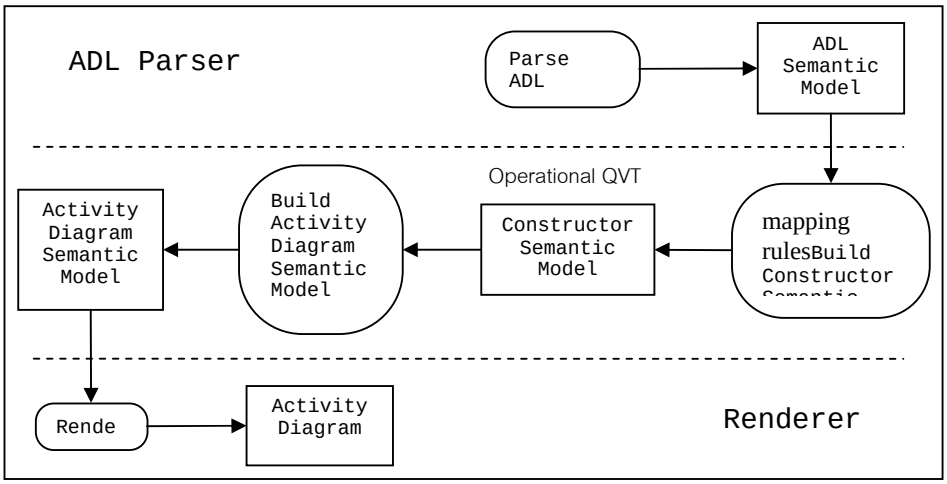


ภาพที่ 10 แผนภาพกิจกรรมตัวอย่างที่ 4 [4]

```
diagram 'ex4'
   modifyProposal->reviewProposal
   decision from reviewProposal
      if 'decision = modify' then notifyOfModification and modifyProposal
      else
             if 'decision = accept' then publishProposal
             else
                    if 'decision = reject' then notifyOfRejection
                    endif
             endif
      endif
    end
    notifyOfModification->break
end
```

ภาพที่ 11 เอดีแอลของแผนภาพกิจกรรมตัวอย่างที่ 4 [4]

ระบบที่พัฒนาขึ้นของงานวิจัยแบ่งออกเป็นสามมอดูลหลัก คือ มอดูลการแจงส่วนเอดีแอล มอดูลการแปลงแบบจำลอง และมอดูลการแสดงผลแผนภาพกิจกรรม โดยทั้งสามมอดูลมีความสัมพันธ์กันดังภาพที่ 12 การแจงส่วนเอดีแอลจะทำหน้าที่ในการแปลงบทคำสั่งที่เขียนขึ้นในรูปของเอดีแอลให้อยู่ในรูปของแบบจำลองความหมายของเอดีแอล หลังจากนั้นมอดูลการแปลงแบบจำลองจะทำหน้าที่แปลงแบบจำลองความหมายของเอดีแอลให้อยู่ในรูปของแบบจำลองความหมายของแผนภาพกิจกรรมต่าง ๆ เมื่อได้แบบจำลองความหมายของแผนภาพกิจกรรมแล้ว มอดูลการแสดงผลแผนภาพกิจกรรมจะหน้าที่ในการสร้างแผนภาพกิจกรรมในรูปแบบกราฟิวซเพื่อแปลงเป็นรูปภาพจากข้อมูลดังกล่าว



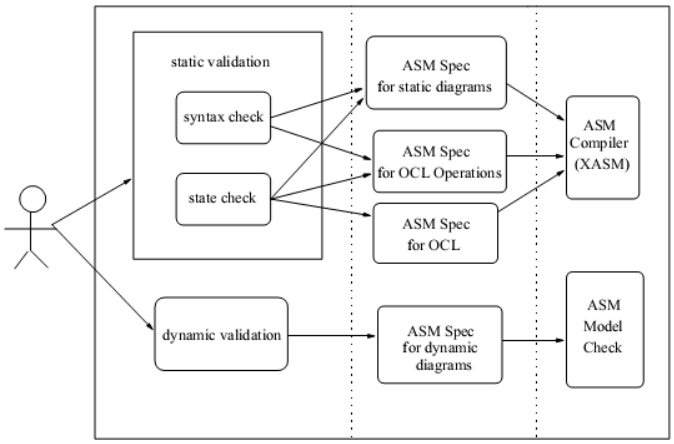
ภาพที่ 12 การทำงานของระบบ [4]

อย่างไรก็ตาม เอดีแอลมีปัญหาที่สำคัญ คือ การไม่สามารถนำเอกสารยูเอ็มแอลที่อยู่ในองค์กรอยู่แล้วมาใช้ร่วมได้ การพัฒนาเอดีแอลนั้นจำเป็นต้องเริ่มต้นที่การออกแบบใหม่เท่านั้น และการประยุกต์ใช้งานเอดีแอลนอกเหนือจากการสร้างแผนภาพกิจกรรมจะกระทำได้ยาก

2.2.2 A Toolset for Supporting UML Static and Dynamic Model Checking [11]

งานวิจัยนี้เสนอชุดเครื่องมือที่ช่วยในการตรวจสอบความสมเหตุสมผลระหว่างตัวแบบเชิงสถิตย์เช่น แผนภาพคลาส และตัวแบบเชิงพลวัต เช่น แผนภาพสถานะและแผนภาพกิจกรรม ซึ่งชุดเครื่องมือนี้ใช้แบบจำลองสื่อความหมายด้วยสถานะนามธรรมในการสื่อความหมายต่างๆ ของแต่ละแผนภาพยูเอ็มแอล

นอกจากนี้ ชุดเครื่องมือจะรองรับการตรวจสอบความสมเหตุสมผลด้วยเครื่องมือพัฒนายูเอ็มแอลอื่นๆ ถ้าเครื่องมือนั้นสามารถแปลงตัวแบบให้อยู่ในรูปแบบเอ็กซ์เอ็มไอได้



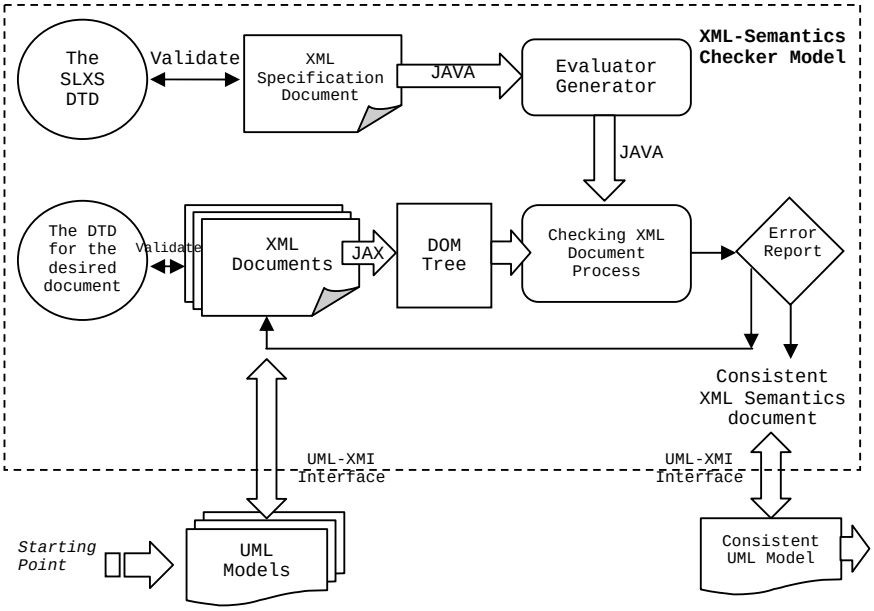
ภาพที่ 13 สถาปัตยกรรมของชุดเครื่องมือ [15]

จากภาพที่ 11 สถาปัตยกรรมประกอบด้วยกัน 3 ส่วนหลักๆ คือ ส่วนของการให้เหตุผล ซึ่งจะมีทั้งแบบสถิตย์และพลวัต รายละเอียดข้อมูลจำเพาะเอเอสเอ็ม และส่วนคอมไพเลอร์และส่วนตรวจสอบแบบจำลอง

อย่างไรก็ตาม เครื่องมือที่พัฒนาขึ้นยังไม่มีการทดสอบกับระบบงานที่มีขนาดใหญ่และใช้งานจริง และการพัฒนาต่อยอดกระทำได้ยาก เนื่องจากการการออกแบบนั้นไม่ได้ออกแบบไว้ สิ่งที่น่าสนใจจากงานวิจัยคือ แนวคิดในการประยุกต์ทำงานร่วมกับซอฟต์แวร์ออกแบบยูเอ็มแอลตัวอื่นบนภาษาเอ็กซ์เอ็มไอและการประยุกต์ตรวจสอบความสอดคล้องกันระหว่างแผนภาพสถิตย์และพลวัต

2.2.3 Consistency Checking of UML Model Diagrams Using the XML Semantics Approach [12]

งานวิจัยนี้ได้นำทฤษฎีการสื่อความหมายเอ็กซ์เอ็มแอลแบบใหม่ที่ผู้เขียนได้คิดค้นขึ้นในงานวิจัยก่อนหน้ามาใช้ในการตรวจสอบความไม่สอดคล้องในการสื่อความหมายของเอกสารเอ็กซ์เอ็มแอล โดยการใช้เทคนิคแกรมม่าแอททริบิวต์ เพื่อตรวจสอบความไม่สอดคล้องระหว่างแผนภาพยูเอ็มแอล แนวคิดหลักของงานวิจัยคือ แปลงแผนภาพยูเอ็มแอลให้อยู่ในรูปเอกสารเอ็กซ์เอ็มไอและตรวจสอบความไม่สอดคล้องบนเอกสารเอ็กซ์เอ็มไอ ซึ่งเป็นรูปแบบพิเศษของเอ็กซ์เอ็มแอล



ภาพที่ 14 สถาปัตยกรรมการตรวจสอบความไม่สอดคล้องของแผนภาพยูเอ็มแอล [16]

จากภาพที่ 14 ระบบแบ่งออกเป็น 3 ส่วน คือ

1. ชิ้นส่วนออกแบบตัวแบบยูเอ็มแอล (UML Model Design Part)
2. ชิ้นส่วนตรวจสอบตัวแบบยูเอ็มแอล (UML Model Checking Part)
3. ชิ้นส่วนเชื่อมต่อยูเอ็มแอล – เอ็กซ์เอ็มไอ (UML-XMI Interface Part)

อย่างไรก็ตาม งานวิจัยนี้ขาดการอธิบายการใช้เทคนิคแกรมม่าแอททริบิวต์อย่างชัดเจนว่ามีการประยุกต์ร่วมกันอย่างไร และผลลัพธ์ที่ได้หากมีและไม่มีจะเกิดผลอย่างไรขึ้น สิ่งที่น่าสนใจคือการตรวจสอบยูเอ็มแอลผ่านเอกสารเอ็กซ์เอ็มไอ โดยการตรวจนั้นใช้การตรวจด้วยการแปลงเอกสารเป็นดอมทรีและเปรียบเทียบกับเอกสารข้อกำหนดเอ็กซ์เอ็มแอลทีแปลงด้วยภาษาจาวาด้วยเทคนิคแกรมม่าแอททริบิวต์และได้ผลลัพธ์เป็นรายงานซึ่งบ่งบอกความไม่สอดคล้องของแบบจำลองออกมาได้

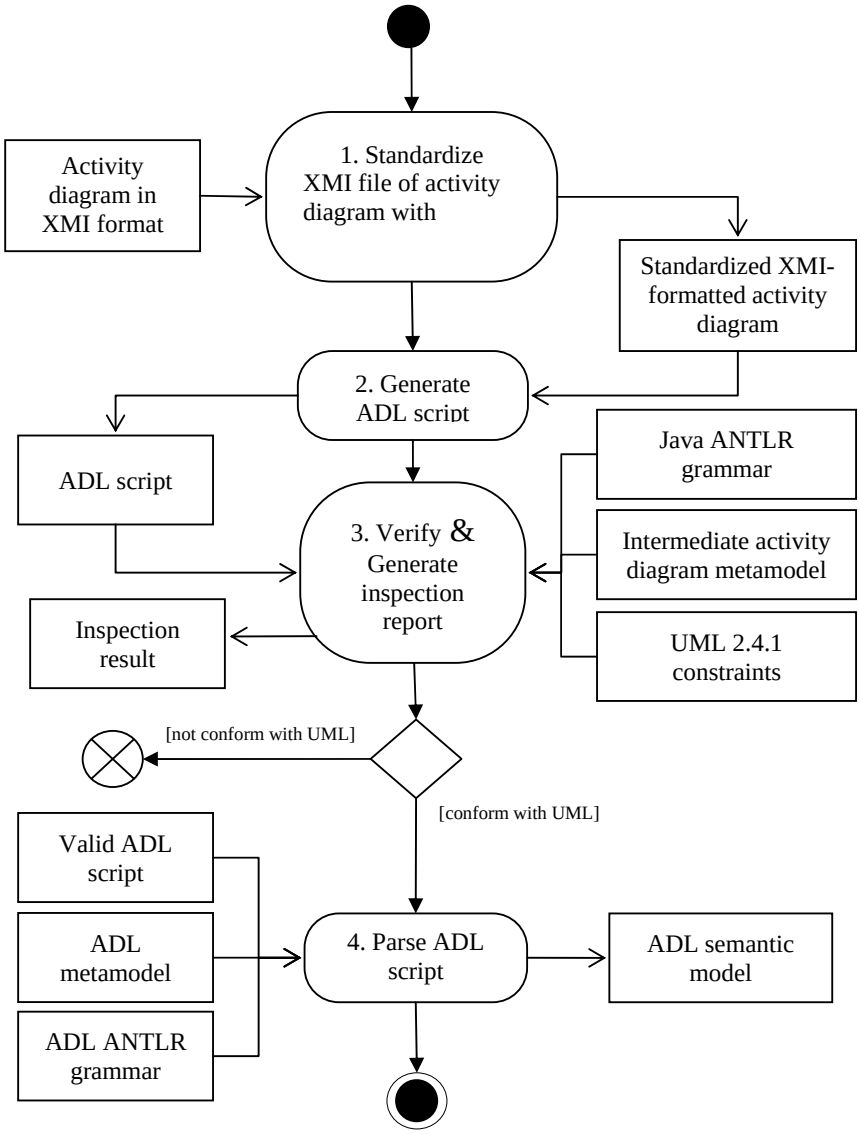
บทที่ 3
วิธีดำเนินการวิจัย

3.1 แนวคิดวิธีดำเนินการวิจัย

งานวิจัยนี้ได้ศึกษาและพัฒนาขยายความสามารถเอดีแอล เพื่อทำให้เอดีแอลเหมาะสมในการใช้งานในองค์กรและทางด้านธุรกิจมากขึ้น โดยการพัฒนาจะเริ่มต้นจากการวิเคราะห์หาจุดที่ต้องปรับปรุงของเอดีแอลและออกแบบซอฟต์แวร์ให้สามารถแก้ไขปัญหาที่มี ซึ่งจากการวิเคราะห์พบว่าการใช้เอดีแอลเป็นตัวเริ่มต้นในการออกแบบแผนภาพกิจกรรมนั้นไม่เหมาะสมกับลักษณะองค์กร เนื่องจากเอกสารต่างๆ ภายในองค์กรนั้นทั้งหมดถูกเก็บในลักษณะไฟล์ของซอฟต์แวร์ออกแบบยูเอ็มแอลหรือกระดาษ ดังนั้น หากต้องพัฒนาเอดีแอลตามแผนภาพที่มีจึงเป็นเรื่องที่เสียเวลาเป็นอย่างมาก ในการพัฒนานั้นจึงพยายามที่จะแก้ไขปัญหาดังกล่าวด้วยการแปลงแผนภาพที่ออกแบบด้วยซอฟต์แวร์อื่นเป็นรูปแบบมาตรฐานก่อนแล้วจึงแปลงรูปแบบมาตรฐานเป็นเอดีแอล ซึ่งในการพัฒนานั้นเลือกการพัฒนาซอฟต์แวร์ปลั๊กอินอีคลิปส์ เนื่องจากมีความเหมาะสมในการประยุกต์ใช้งานและพัฒนาต่อยอด รวมทั้งสะดวกในการให้นักวิจัยท่านอื่นได้นำมาติดตั้งใช้งานผ่านไซต์อัพเดท ในการพัฒนานั้นจะแยกฟังก์ชันหลักทั้งหมดออกเป็นอย่างละปลั๊กอินและรวมปลั๊กอินทั้งหมดเป็นหมวดหมู่ในการติดตั้งใช้งาน สำหรับรูปแบบมาตรฐานดังกล่าวได้เลือกเอกสารเอ็กซ์เอ็มไอเป็นตัวกลางในการสื่อสาร หลังจากที่แปลงให้เป็นเอกสารเอดีแอลแล้ว เอกสารเอดีแอลที่ได้จะถูกนำไปทวนสอบและตรวจสอบความสมเหตุสมผลของข้อมูลระหว่างแผนภาพกิจกรรม ผลลัพธ์ที่ได้คือรายงานการตรวจสอบว่าแผนภาพกิจกรรมนั้นว่าตรงตามมาตรฐานยูเอ็มแอลหรือไม่ ถ้าเกิดกรณีที่ไม่ตรงตามมาตรฐานขึ้น รายงานจะบ่งชี้ว่าแผนภาพนั้นขาดคุณสมบัติในข้อใด และสร้างแบบจำลองความหมายของเอดีแอลขึ้นมา เพื่อให้สามารถนำไปประยุกต์ใช้งานในงานอื่นๆ ต่อไปได้ ภาพรวมการทำงานระบบ แสดงดังภาพที่ 15

3.2 การแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน

การแปลงแผนภาพกิจกรรมเป็นเอดีแอลนั้นจะเริ่มด้วยไฟล์ต้นฉบับที่เป็นไฟล์เอ็กซ์เอ็มไอที่ได้มาจากซอฟต์แวร์โอเพนซอร์ส ไฟล์ต้นแบบจากซอฟต์แวร์ที่ต่างกันจะมีโครงสร้างที่ต่างกันเนื่องจากเวอร์ชันที่แตกต่างของเอ็กซ์เอ็มไอ และสัญกรณ์ที่ใช้ในโปรแกรมมีความไม่สอดคล้องกัน ในการแปลงนั้นจะทำการพิจารณาโครงสร้างของแต่ละซอฟต์แวร์ออกแบบยูเอ็มแอลและทำการแจงส่วนบนเอกสารเอ็กซ์เอ็มไอ



ภาพที่ 15 ภาพรวมการทำงานระบบ

3.2.1 แผนภาพที่ออกแบบด้วยอัลโก้ยูเอ็มแอล

อัลโก้ยูเอ็มแอลเป็นซอฟต์แวร์ออกแบบยูเอ็มแอล พัฒนาด้วยภาษาจาวาโดย Jason E. Robbins ปัจจุบันเป็นซอฟต์แวร์โอเพนซอร์สอยู่ภายใต้สัญญา Eclipse Publish License 1.0 [5]

เอกสารเอ็กซ์เอ็มไอที่ได้จากไฟล์ต้นฉบับอัลโก้ยูเอ็มแอลนั้นเป็นเอกสารเอ็กซ์เอ็มไอตามมาตรฐานเอ็กซ์เอ็มไอ 1.2 รองรับมาตรฐานยูเอ็มแอล 1.4 แท็กที่สำคัญของเอกสารเอ็กซ์เอ็มไอประกอบด้วยกัน 3 อย่างดังนี้

1. UML:Model เป็นแท็กที่ให้รายละเอียดของแผนภาพกิจกรรมได้แก่ ชื่อของแผนภาพกิจกรรม รหัสประจำตัวของแผนภาพกิจกรรม และคุณสมบัติต่างๆ ตัวอย่างแสดงได้ดังภาพที่ 16

```
<UML:Model xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000
   0000865'name = 'ex1' isSpecification = 'false' isRoot = 'false'
   isLeaf = 'false'isAbstract = 'false'>
```

ภาพที่ 16 ตัวอย่าง UML:Model

2. UML:CompositeState.subvertex เป็นแท็กที่รวบรวมวัตถุทั้งหมดของแผนภาพกิจกรรมได้แก่ แอ็กชัน ออปเจ็ก และบัพควบคุมต่างๆ ได้แก่ บัพแยก บัพรวม บัพตัดสินใจ บัพผสาน บัพเริ่มต้น บัพหยุดสายงาน และบัพหยุดกิจกรรม ตัวอย่างแสดงได้ดังภาพที่ 17 โดยการระบุว่าวัตถุแต่ละวัตถุเป็นประเภทใดนั้น จะใช้ชื่อแท็กแบ่งประเภท มีด้วยกัน 4 ค่า ดังนี้
   2.1. UML:ActionState เป็นแท็กซ์ที่ให้ข้อมูลแอ็กชัน
   2.2. UML:ObjectFlowState เป็นแท็กซ์ที่ให้ข้อมูลออปเจ็ก
   2.3. UML:Pseudostate เป็นแท็กซ์ที่ให้ข้อมูลบัพควบคุมต่างๆ ได้แก่ บัพแยก บัพรวม บัพตัดสินใจ บัพผสาน และบัพเริ่มต้น
   2.4. UML:FinalState เป็นแท็กซ์ที่ให้ข้อมูลบัพหยุดสายงาน

   ภายในแท็กวัตถุนั้นจะมีแท็กที่สำคัญด้วยกัน 2 อย่างคือ UML:StateVertex.outgoing และ UML:StateVertex.incoming เป็นแท็กที่ให้ข้อมูลเส้นเชื่อมออกและเส้นเชื่อมเข้าตามลำดับ โดยเส้นเชื่อมแต่ละเส้นเชื่อมจะใช้แท็ก UML:Transition เป็นตัวเชื่อมกันผ่านค่า xmi.idref
3. UML:StateMachine.transitions เป็นแท็กที่รวบรวมเส้นเชื่อมทั้งหมดของแผนภาพกิจกรรม ตัวอย่างแสดงได้ดังภาพที่ 18 โดยคุณสมบัติของเส้นเชื่อมคือเป็นความสัมพันธ์แบบ 1 ต่อ 1 ระหว่างวัตถุใดๆ กับอีกวัตถุดิบ แท็กที่ใช้ในการอ้างอิงต้นทางและปลายทางคือ UML:Transition.source และ UML:Transition.target

```
<UML:CompositeState.subvertex>
   <UML:ActionState xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000
       :0000000000000868'name = 'Unmarshall Order' isSpecification =
       'false' isDynamic = 'false'>
       <UML:StateVertex.outgoing>
           <UML:Transition xmi.idref = '-64--88-1-5-1cf81117:13aa30
               d6c5e:-8000:0000000000000874'/>
           <UML:Transition xmi.idref = '-64--88-1-5-1cf81117:13aa30
               d6c5e:-8000:0000000000000875'/>
           <UML:Transition xmi.idref = '-64--88-1-5-1cf81117:13aa30
               d6c5e:-8000:0000000000000876'/>
       </UML:StateVertex.outgoing>
       <UML:StateVertex.incoming>
           <UML:Transition xmi.idref = '-64--88-1-5-1cf81117:13aa30
               d6c5e:-8000:0000000000000873'/>
       </UML:StateVertex.incoming>
   </UML:ActionState>
</UML:CompositeState.subvertex>
```

ภาพที่ 17 ตัวอย่าง UML:CompositeState.subvertex

```
<UML:StateMachine.transitions>
    <UML:Transition xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000
        :0000000000000873'isSpecification = 'false'>
       <UML:Transition.source>
            <UML:ObjectFlowState xmi.idref = '-64--88-1-5-1cf81117:
                13aa30d6c5e:-8000:0000000000000869'/>
        </UML:Transition.source>
        <UML:Transition.target>
            <UML:ActionState xmi.idref = '-64--88-1-5-1cf81117:13aa
                30d6c5e:-8000:0000000000000868'/>
        </UML:Transition.target>
    </UML:Transition>
</UML:StateMachine.transitions>
```

ภาพที่ 18 ตัวอย่าง UML:StateMachine.transitions

3.2.1.1. กฎในการแปลงภาพที่ออกแบบด้วยอัลโก้ยูเอ็มแอล

กฎในการแปลงประกอบด้วยกัน 20 ข้อ ดังนี้

1. ถ้าแท็กเป็น ActivityGraph ให้ดึงข้อมูลเก็บชื่อของแผนภาพกิจกรรม
2. ถ้าแท็กเป็น StateMachine.transitions ให้เปลี่ยนสถานะพร้อมจะสร้างเส้นเชื่อม
3. ถ้าอยู่ในสถานะพร้อมจะสร้างเส้นเชื่อมและแท็กเป็น Transition ให้ทำการสร้างเส้นเชื่อมชั่วคราวขึ้นกำหนดรหัสตามลำดับที่เจอ กำหนดค่าอ้างอิงที่ได้รับจากซอฟต์แวร์อัลโก้ยูเอ็มแอล และจัดเก็บชื่อของเส้นเชื่อมถ้าหากมีการกำหนดมา เส้นเชื่อมจะมีชื่อในกรณีที่เป็นเส้นเชื่อมนั้นเป็นเส้นเชื่อมออกของบัพตัดสินใจ
4. ถ้าแท็กเป็น Transition.source ให้เปลี่ยนสถานะพร้อมจะบันทึกบัพต้นทาง
5. ถ้าแท็กเป็น Transition.target ให้เปลี่ยนสถานะพร้อมจะบันทึกบัพปลายทาง
6. ถ้าอยู่ในสถานะพร้อมจะบันทึกบัพต้นทางและแท็กเป็น ActionState, ObjectFlowState หรือ Pseudostate ให้กำหนดค่าบัพต้นทางกับเส้นเชื่อมชั่วคราว
7. ถ้าอยู่ในสถานะพร้อมจะบันทึกบัพปลายทางและแท็กเป็น ActionState, ObjectFlowState, Pseudostate หรือ FinalState ให้กำหนดค่าบัพปลายทางกับเส้นเชื่อมชั่วคราว
8. ถ้าแท็กเป็น CompositeState.subvertex ให้เปลี่ยนสถานะพร้อมจะสร้างบัพ
9. ถ้าอยู่ในสถานะพร้อมจะสร้างบัพและแท็กเป็น ActionState ให้ทำการสร้างบัพชั่วคราวขึ้น กำหนดรหัสตามลำดับที่เจอ กำหนดค่าอ้างอิงที่ได้รับจากซอฟต์แวร์อัลโก้ยูเอ็มแอล กำหนดชื่อ กำหนดประเภทของบัพเป็นแอ็กชัน และกำหนดสถานะกำลังพิจารณาบัพ
10. ถ้าอยู่ในสถานะพร้อมจะสร้างบัพและแท็กเป็น ObjectFlowState ให้ทำการสร้างบัพชั่วคราวขึ้น กำหนดรหัสตามลำดับที่เจอ กำหนดค่าอ้างอิงที่ได้รับจากซอฟต์แวร์อัลโก้ยูเอ็มแอล กำหนดชื่อ กำหนดประเภทของบัพเป็นวัตถุและกำหนดสถานะกำลังพิจารณาบัพ
11. ถ้าอยู่ในสถานะพร้อมจะสร้างบัพและแท็กเป็น Pseudostate และชนิดของแท็กมีค่า initial ให้ทำการสร้างบัพชั่วคราวขึ้น กำหนดรหัสตามลำดับที่เจอ

กำหนดค่าอ้างอิงที่ได้รับจากซอฟต์แวร์อัลโก้ยูเอ็มแอล กำหนดประเภทของบัพเป็นบัพเริ่มต้น กำหนดให้สถานะบัพที่กำลังพิจารณาเป็นบัพเริ่มต้น และกำหนดสถานะกำลังพิจารณาบัพ

12. ถ้าอยู่ในสถานะพร้อมจะสร้างบัพและแท็กเป็น Pseudostate ให้ทำการสร้างบัพชั่วคราวขึ้น กำหนดรหัสตามลำดับที่เจอ กำหนดค่าอ้างอิงที่ได้รับจากซอฟต์แวร์อัลโก้ยูเอ็มแอล กำหนดประเภทของบัพเป็นชนิดตามที่ระบุในแท็กและกำหนดสถานะกำลังพิจารณาบัพ
13. ถ้าอยู่ในสถานะพร้อมจะสร้างบัพและแท็กเป็น FinalState ให้ทำการสร้างบัพชั่วคราวขึ้น กำหนดรหัสตามลำดับที่เจอ กำหนดค่าอ้างอิงที่ได้รับจากซอฟต์แวร์อัลโก้ยูเอ็มแอล กำหนดประเภทของบัพเป็นบัพสิ้นสุดกิจกรรมและกำหนดสถานะกำลังพิจารณาบัพ
14. ถ้าอยู่ในสถานะกำลังพิจารณาบัพและแท็กเป็น StateVertex.outgoing ให้กำหนดสถานะกำลังพิจารณาเส้นเชื่อมออก
15. ถ้าอยู่ในสถานะกำลังพิจารณาบัพและแท็กเป็น StateVertex.incoming ให้กำหนดสถานะกำลังพิจารณาเส้นเชื่อมเข้า
16. ถ้าอยู่ในสถานะเส้นเชื่อมออกและแท็กเป็น Transition ให้บัพชั่วคราวเพิ่มข้อมูลเส้นเชื่อมออกด้วยค่าอ้างอิงของแท็กที่กำลังแจงส่วนข้อมูล
17. ถ้าอยู่ในสถานะเส้นเชื่อมเข้าและแท็กเป็น Transition ให้บัพชั่วคราวเพิ่มข้อมูลเส้นเชื่อมเข้าด้วยค่าอ้างอิงของแท็กที่กำลังแจงส่วนข้อมูล
18. ถ้าอยู่ในสถานะพร้อมจะสร้างเส้นเชื่อมและแท็กปิดเป็น Transition ให้เพิ่มเส้นเชื่อมชั่วคราวเข้าในชุดของเส้นเชื่อมแผนภาพกิจกรรม
19. ถ้าอยู่ในสถานะพร้อมจะสร้างบัพและแท็กปิดเป็น ActionState, ObjectFlow State หรือ Pseudostate ให้เพิ่มบัพชั่วคราวเข้าในชุดของบัพแผนภาพกิจกรรม
20. ถ้าเจอแท็กปิดของสถานะใดๆ ที่สร้างขึ้นมา ให้กำหนดยกเลิกสถานะนั้นๆ

3.2.2 แผนภาพที่ออกแบบด้วยโมเดลลิโอ

โมเดลลิโอเป็นซอฟต์แวร์ออกแบบยูเอ็มแอล พัฒนาโดยบริษัท ModelioSoft ปัจจุบันเป็นซอฟต์แวร์โอเพนซอร์สอยู่ภายใต้สัญญา GPLv3 และ Key APIs อยู่ภายใต้สัญญา Apache License 2.0 [6] ในเวอร์ชันปัจจุบัน 2.2.1 รองรับมาตรฐานยูเอ็มแอล 2.0 และ BPMN แท็กที่

สำคัญของเอกสารเอ็กซ์เอ็มไอประกอบด้วยกัน 3 อย่างคือ

1. packagedElement เป็นแท็กที่ให้รายละเอียดของแผนภาพกิจกรรมได้แก่ ประเภทของแผนภาพ รหัสประจำตัวของแผนภาพ และชื่อของแผนภาพ ตัวอย่างแสดงได้ดังภาพที่ 19

```
<packagedElement xmi:type="uml:Activity"
    xmi:id="_mhyeRiZCEeKSKePaylnoVA" name="ex1">
```

ภาพที่ 19 ตัวอย่าง packagedElement

2. node เป็นแท็กที่ให้รายละเอียดของบัพแต่ละบัพของแผนภาพกิจกรรมได้แก่ ประเภทของบัพ รหัสประจำตัวของบัพ ชื่อของบัพ เส้นเชื่อมเข้า และเส้นเชื่อมออก ตัวอย่างแสดงได้ดังภาพที่ 20 โดยภายในการเส้นเชื่อมเข้าและเส้นเชื่อมออกนั้นหากมีจำนวนหลายเส้นจะแบ่งด้วยช่องว่าง 1 ช่อง
3. edge เป็นแท็กซ์ที่ให้รายละเอียดของเส้นเชื่อมแต่ละเส้นเชื่อมของแผนภาพกิจกรรมได้แก่ ประเภทของเส้นเชื่อม รหัสประจำตัวของเส้นเชื่อม ชื่อของเส้นเชื่อม บัพต้นทางและบัพปลายทาง ตัวอย่างแสดงได้ดังภาพที่ 21

```
<node xmi:type="uml:OpaqueAction" xmi:id="_mhzFWSZCEeKSKePaylnoVA"
    name="Unmarshall Order" outgoing="_mhzFbiZCEeKSKePaylnoVA
    _mhzsYyZCEeKSKePaylnoVA _mhzsaCZCEeKSKePaylnoVA" incoming="_mhz
    sbSZCEeKSKePaylnoVA">
    <body/>
</node>
<node xmi:type="uml:CentralBufferNode" xmi:id="_mhzFXyZCEeKSKePay
    lnoVA" name="Name" incoming="_mhzFbiZCEeKSKePaylnoVA">
    <eAnnotations xmi:id="_mhzFYCZCEeKSKePaylnoVA" source="Objing">
        <contents xmi:type="uml:Property" xmi:id="_mhzFYSZCEeKSKe
            PaylnoVA" name="Type">
        </contents>
    </eAnnotations>
    <upperBound xmi:type="uml:LiteralString" xmi:id="_mhzFYyZCEeKSK
        ePaylnoVA" name="UpperBound" value="1"/>
</node>
```

ภาพที่ 20 ตัวอย่าง node

```
<edge xmi:type="uml:ObjectFlow" xmi:id="_mhzsbSZCEeKSKePaylnoVA"
    name="ObjectFlow" source="_mhzFWiZCEeKSKePaylnoVA" target="_mhz
    FWSZCEeKSKePaylnoVA">
    <eAnnotations xmi:id="_mhzsbiZCEeKSKePaylnoVA" source="Objing">
        <contents xmi:type="uml:Property" xmi:id="_mhzsbyZCEeKSKeP
            aylnoVA" name="Effect">
            <defaultValue xmi:type="uml:LiteralString" xmi:id="_mhz
                scCZCEeKSKePaylnoVA" value="READ_FLOW"/>
        </contents>
    </eAnnotations>
    <weight xmi:type="uml:LiteralInteger" xmi:id="_mhzscSZCEeKSKeP
        aylnoVA" value="1"/>
</edge>
```

ภาพที่ 21 ตัวอย่าง edge

3.2.2.1. **กฎในการแปลงภาพที่ออกแบบด้วยโมเดลลิโอ**

กฎในการแปลงประกอบด้วยกัน 3 ข้อ ดังนี้

1. ถ้าแท็กเป็น packagedElement ให้ดึงข้อมูลเก็บชื่อของแผนภาพกิจกรรม
2. ถ้าแท็กเป็น node ให้ดำเนินการต่อไปนี้
    - สร้างบัพชั่วคราวขึ้นมา
    - กำหนดรหัสตามลำดับที่เจอ
    - กำหนดรหัสอ้างอิงที่ได้รับจากซอฟต์แวร์โมเดลลิโอ
    - เพิ่มเส้นเชื่อมเข้า
    - เพิ่มเส้นเชื่อมออก
    - กำหนดประเภทของบัพ
    - พิจารณาบัพชั่วคราวที่สร้าง ถ้าประเภทของบัพเป็นบัพเริ่มต้นหรือบัพสิ้นสุดกิจกรรมให้เก็บค่าบัพชั่วคราวก่อนแล้วจึงค่อยเพิ่มบัพเข้าไปในรายการบัพหลังจากเสร็จสิ้นการแจงส่วนเอกสารเอ็กซ์เอ็มไอ
    - แต่ถ้าไม่ใช่ประเภทดังกล่าวให้เพิ่มบัพเข้าไปในรายการบัพของแผนภาพกิจกรรม

3. ถ้าแท็กเป็น edge ให้ดำเนินการต่อไปนี้
    - สร้างเส้นเชื่อมชั่วคราวขึ้นมา
    - กำหนดรหัสตามลำดับที่เจอ
    - กำหนดรหัสอ้างอิงที่ได้รับจากซอฟต์แวร์โมเดลลิโอ
    - กำหนดชื่อของเส้นเชื่อม (ถ้ามี)
    - กำหนดบัพต้นทาง
    - กำหนดบัพปลายทาง

ผลลัพธ์การแปลงเอกสารเอ็กซ์เอ็มไอที่ได้จากการแปลงนั้นจะเป็นเอกสารที่ตรงตามมาตรฐานเอ็กซ์เอ็มไอ 2.0 เนื่องจากเพื่อให้ที่ให้เอกสารนั้นนอกจากการนำไปใช้ในขั้นตอนต่อๆ ไปแล้วยังสามารถไปใช้กับคิววิทีเชิงปฏิบัติและกราฟวิชที่ใช้โครงสร้างตามการออกแบบของงานวิจัยข้อกำหนดรูปนัยเพื่อทวนสอบแผนภาพกิจกรรมของกระแสกระบวนการ

3.3 การแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล

การแปลงเอกสารเอ็มซ์เอ็มไอเป็นเอดีแอลนั้นจะใช้การแจงส่วนเอกสารเอ็มซ์เอ็มไอ ซึ่งจะได้ข้อมูลบัพและเส้นเชื่อมทั้งหมดเก็บเอาไว้ในตัวแปร ขั้นตอนการสร้างเอดีแอลก็จะดึงตัวแปรบัพและเส้นเชื่อมที่ได้มา แล้วค่อยๆ พิจารณาเริ่มต้นที่บัพแรกของข้อมูลบัพ ลำดับในการแปลงประกอบด้วยกัน 3 ขั้นตอนดังนี้

1. สร้างไวยากรณ์แอ็กชัน
2. สร้างไวยากรณ์การตัดสินใจ
3. สร้างไวยากรณ์ลำดับของแอ็กชัน

3.3.1 สร้างไวยากรณ์แอ็กชัน

การสร้างไวยากรณ์นั้นจะพิจารณาแอ็กชันเป็น 2 ประเภทคือ แอ็กชันแบบแสดงวัตถุ และแบบไม่แสดงวัตถุ ในกรณีที่ไม่แสดงวัตถุนั้น ซอฟต์แวร์จะสร้างไวยากรณ์ขึ้นมา โดยใช้รูปแบบโครงสร้างดังภาพที่ 22

```
action actionName end
```

ภาพที่ 22 โครงสร้างแอ็กชันแบบไม่แสดงวัตถุ

เหตุผลที่เลือกให้แสดงไวยากรณ์ แทนที่จะไม่แสดงเพื่อให้ผู้ใช้เห็นผลลัพธ์ที่ชัดเจนว่ามีแอ็กชันนี้ในแผนภาพกิจกรรม อีกทั้งเป็นการสร้างบรรทัดฐานขึ้นมาว่าการที่จะใช้งานแอ็กชันในเอดีแอลนั้นจำเป็นต้องมีการประกาศแอ็กชันขึ้นมาก่อน

กรณีที่แสดงวัตถุนั้น ซอฟต์แวร์จะพิจารณาว่าแอ็กชันดังกล่าวมีเส้นเชื่อมโยงเข้าและเส้นเชื่อมโยงออกหรือไม่ ไวยากรณ์ที่ได้จะมีรูปแบบโครงสร้างดังภาพที่ 23

```
action actionName
    <- incomingObject1, incomingObject2, …, incomingObjectN
    -> outgoingObject1, outgoingObject2, …, outgoingObjectN
End
```

ภาพที่ 23 โครงสร้างแอ็กชันแบบแสดงวัตถุ

โดยที่ incomingObjects คือชื่อของวัตถุที่เป็นข้อมูลนำเข้าสู่แอ็กชัน และ outgoing Objects เป็นชื่อของวัตถุที่เป็นข้อมูลส่งออกของแอ็กชัน การเขียนนั้นจะระบุวัตถุทีละตัวแล้วคั่นวัตถุแต่ละตัวด้วยเครื่องหมายจุลภาค

สำหรับกรณีที่แอ็กชันแสดงวัตถุนั้น บังคับให้ไวยากรณ์ต้องมีการแสดงผลเท่านั้น ไม่สามารถย่อหรือละไว้ในฐานที่เข้าใจได้

3.3.2 สร้างไวยากรณ์การตัดสินใจ

การสร้างไวยากรณ์นั้นจะใช้ฟังก์ชันเรียกตัวเองในการค้นหาบัพเป้าหมายของผลลัพธ์ของบัพตัดสินใจ หลักการสร้างไวยากรณ์ตัดสินใจนั้นมีดังนี้

1. การสร้างนั้นจะพิจารณาว่าบัพตัดสินใจจะมีเส้นเชื่อมออก 1 ถึง n เส้น
2. ผลลัพธ์ของเงื่อนไขจะอยู่ในรูปแบบชื่อของแอ็กชันหรือบัพแยกเท่านั้น
3. ในกรณีที่บัพตัดสินใจนั้น มีบัพเป้าหมายเส้นใดเส้นหนึ่งเป็นบัพตัดสินใจ ให้ทำการสร้างไวยากรณ์ซ้อนข้างในอีกชั้นหนึ่งแทนการสร้างไวยากรณ์ตัดสินใจอีกชุดหนึ่ง

3.3.2.1. ขั้นตอนการสร้างไวยากรณ์

1. ประกาศไวยากรณ์ decision from incomingNodeName เพื่อบอกว่าเป็นการตัดสินใจว่าบัพตัวใด โดยที่ incomingNodeName คือชื่อของแอ็กชันหรือบัพแยกก่อนที่จะเข้าสู่บัพตัดสินใจ ซึ่งแสดงถึงเงื่อนไขในการตัดสินใจ
2. พิจารณาเส้นเชื่อมออกของบัพตัดสินใจลำดับที่ i
3. กรณี i เท่ากับ 1 จะถือว่าเป็นกรณีแรกของการตัดสินใจให้ระบุคำสั่ง if เท่านั้น
4. กรณี i มากกว่า 1 จะถือว่าเป็นกรณีหลังของการตัดสินใจให้ระบบคำสั่ง else if เท่านั้น
5. ดึงข้อมูลเส้นเชื่อมลำดับที่ i ออกมา
6. ระบุไวยากรณ์ if edgeLabel than outgoingNodeName หรือ else if edgeLabel than outgoingNodeName โดยที่ edgeLabel คือชื่อของเส้น

เชื่อมออกที่ออกมาจากบัพตัดสินใจ และoutgoingNodeName คือชื่อของแอ็กชันหรือบัพแยกซึ่งจะได้ชื่อนั้นมาโดยการใช้ฟังก์ชันเรียกตัวเองที่มีการทำงานดังนี้

- เมื่อถูกเรียกทำงานจะมีพารามิเตอร์คือคลาสของบัพ และตำแหน่งของเส้นเชื่อมออกที่ต้องการ
- ดึงข้อมูลบัพปลายทางของเส้นเชื่อมออกมา
- ตรวจสอบประเภทของบัพที่ได้ว่าเป็นแอ็กชัน บัพแยก บัพเริ่มต้น บัพสิ้นสุดกิจกรรม หรือบัพหยุดสายงาน ถ้าใช่ให้คืนค่าบัพนั้นกลับไปสู่การทำงานหลัก
- ในกรณที่ไม่ใช่บัพประเภทที่กล่าวมาให้ทำการเรียกตัวเองด้วยบัพปลายทางที่ได้มา

7. กลับไปทำงานขั้นตอนที่ 2 ใหม่จนกว่าจะพิจารณาเส้นเชื่อมออกจนครบทุกเส้น
8. แสดงไวยากรณ์ end if เพื่อปิดไวยากรณ์การตัดสินใจ

3.3.3 **สร้างไวยากรณ์ลำดับของแอ็กชัน**

การสร้างไวยากรณ์นั้นจะใช้ฟังก์ชันเรียกตัวเองในการค้นหาลำดับของแอ็กชันที่มีความยาวที่สุด หลักการสร้างไวยากรณ์ตัดสินใจนั้นมีดังนี้

1. การสร้างลำดับของแอ็กชันจะเริ่มต้นด้วยบัพที่เป็นบัพแอ็กชันเท่านั้น
2. กรณีที่บัพต่อไปเป็นบัพสิ้นสุดกิจกรรม บัพตัดสินใจ หรือบัพแยกจะหยุดการค้นหาบัพต่อไป
3. กรณีที่บัพต่อไปเป็นบัพผสาน บัพรวม หรือบัพวัตถุจะข้ามบัพนี้ไป และค้นหาบัพต่อไป
4. กรณีที่บัพต่อไปเป็นบัพหยุดสายงาน จะคืนค่า break กลับไป
5. กรณีที่บัพต่อไปเป็นแอ็กชันจะทำการเรียกตัวเองอีกครั้งและส่งชื่อของบัพปัจจุบันกลับไปสร้างลำดับของแอ็กชัน
6. การสร้างลำดับของแอ็กชันของบัพแยกนั้นจะออกมาเป็นอีกกรณีหนึ่งซึ่งจะมีลำดับของแอ็กชันเพียงแค่ระหว่างแอ็กชันใดๆ กับบัพแยกเท่านั้น

3.4 การตรวจทานแผนภาพกิจกรรมและสร้างรายงานการตรวจสอบ

แผนภาพกิจกรรมในรูปแบบเอดีแอลนั้นจะถูกนำไปทวนสอบและตรวจสอบความสมเหตุสมผลระหว่างแผนภาพกิจกรรม ดังนี้

1. ตรวจสอบไวยากรณ์ของแผนภาพกิจกรรมตามข้อกำหนดยูเอ็มแอล
2. ตรวจสอบความสมเหตุสมผลระหว่างวัตถุและกิจกรรมภายในแผนภาพกิจกรรม

การตรวจสอบไวยากรณ์ และความสมเหตุสมผลนั้นจะเกิดขึ้นในระหว่างการแจงส่วนที่มีการแปลงบทคำสั่งที่เขียนด้วยเอดีแอลเป็นแบบจำลองความหมายของเอดีแอล ถ้าหากเกิดความผิดพลาดขึ้นมา ระบบจะแจ้งการแปลงผิดพลาดขึ้น และหยุดการสร้างแบบจำลองความหมายของเอดีแอล

การตรวจสอบแผนภาพกิจกรรมนั้นจะพิจารณาทุกๆ คลาสของเมทาโมเดลของแผนภาพกิจกรรมระดับปานกลางในภาพที่ 3 โดยพิจารณาว่าแผนภาพกิจกรรมนั้นสอดคล้องตามเงื่อนไขบังคับ [3] หรือไม่ องค์ประกอบของแผนภาพกิจกรรมระดับปานกลางที่จำเป็นต้องตรวจสอบประกอบไปด้วย กิจกรรม กลุ่มกิจกรรม ส่วนแบ่งกิจกรรม เส้นเชื่อมกิจกรรม บัพกิจกรรม สายงานวัตถุ สายงานควบคุม แอ็กชัน บัพตัวแปรกิจกรรม บัพเริ่มต้น บัพสิ้นสุด บัพตัดสินใจ บัพผสาน บัพรวม บัพแยก บัพสิ้นสุดสายงาน และบัพสิ้นสุดกิจกรรม เงื่อนไขบังคับแต่ละคลาส มีดังนี้

3.4.1 **เงื่อนไขบังคับของกิจกรรม** (Activity)

1. บัพกิจกรรมจะต้องมีบัพตัวแปรกิจกรรม [3] ต่อหนึ่งพารามิเตอร์เท่านั้น และไม่สามารถใช้บัพตัวแปรกิจกรรม [3] ร่วมกันได้
2. เมื่อมีการเรียกกิจกรรมแบบอิสระ (แบบทางตรง) แล้วจะไม่สามารถเรียกกิจกรรมด้วยคุณลักษณะการทำงานได้ (แบบทางอ้อม) กล่าวคือสามารถเรียกกิจกรรมได้แบบใดแบบหนึ่งเท่านั้น
3. จะต้องไม่มีกลุ่มกิจกรรมซ้อนกันเป็นลำดับชั้น

3.4.2 **เงื่อนไขบังคับของเส้นเชื่อมกิจกรรม** (ActivityEdge)

1. ต้นทางและปลายทางของเส้นเชื่อมจะต้องอยู่ภายในกิจกรรมเดียวกัน
2. เส้นเชื่อมกิจกรรมจะสามารถมีกิจกรรมหรือกลุ่มเป็นเจ้าของได้เท่านั้น

3.4.3 **เงื่อนไขบังคับของบัพกิจกรรม** (ActivityNode)

1. บัพกิจกรรมจะมีกิจกรรมหรือกลุ่มกิจกรรมเป็นเจ้าของได้เท่านั้น

3.4.4 **เงื่อนไขบังคับของสายงานวัตถุ** (ObjectFlow)

1. สายงานวัตถุไม่สามารถมีต้นทางและปลายทางเป็นส่วนแสดงทั้งสองทางได้
2. บัพวัตถุที่เชื่อมต่อกับสายงานวัตถุ บางกรณีอาจมีบัพควบคุมเข้ามาเกี่ยวข้อง ต้องเป็นชนิดที่สอดคล้องกัน กล่าวคือวัตถุปลายทางจะต้องเป็นวัตถุที่เป็นชนิดเดียวกันหรือสืบทอดมาจากวัตถุต้นทาง
3. บัพวัตถุที่เชื่อมต่อกับสายงานวัตถุ บางกรณีอาจมีบัพควบคุมเข้ามาเกี่ยวข้อง จะต้องมีขอบเขตบน (Upper Bound) เดียวกัน

3.4.5 **เงื่อนไขบังคับของสายงานควบคุม** (ControlFlow)

1. สายงานควบคุมไม่สามารถเชื่อมกับมีบัพวัตถุได้ ยกเว้นบัพวัตถุจะมีการกำหนดให้เป็นประเภทควบคุม

3.4.6 **เงื่อนไขบังคับของบัพวัตถุ** (ObjectNode)

1. เส้นเชื่อมเข้าและเส้นเชื่อมออกทั้งหมดของบัพวัตถุจะต้องเป็นสายงานวัตถุ

3.4.7 **เงื่อนไขบังคับของบัพเริ่มต้น** (InitialNode)

1. บัพเริ่มต้นไม่มีเส้นเชื่อมเข้า
2. เส้นเชื่อมที่ต่อกับบัพเริ่มต้น มีบัพเริ่มต้นเป็นต้นทาง จะต้องเป็นเส้นเชื่อมควบคุมเท่านั้น

3.4.8 **เงื่อนไขบังคับของ** FinalNode

1. บัพสิ้นสุดไม่มีเส้นเชื่อมออก

3.4.9 **เงื่อนไขบังคับของ** DecisionNode

1. บัพตัดสินใจสามารถมีได้เส้นเชื่อมเข้าได้เพียงหนึ่งหรือสองเส้นเท่านั้น และจะต้องมีเส้นเชื่อมออกอย่างน้อยหนึ่งเส้นเสมอ
2. เส้นเชื่อมเข้าหรือออกจากบัพตัดสินใจ นอกจากเหนือจากสายงานนำเข้าของการตัดสินใจ (ถ้ามี) จะต้องเป็นสายงานควบคุมทั้งหมดหรือสายงานวัตถุทั้งหมดอย่างใดอย่างหนึ่ง
3. สายงานนำเข้าของการตัดสินใจจะต้องเป็นเส้นเชื่อมเข้าของบัพตัดสินใจเท่านั้น
4. การตัดสินใจที่ต้องใช้ข้อมูลนำเข้าจะต้องไม่มีตัวแปรนำออก ตัวแปรไหลผ่าน หรือข้อมูลส่งกลับแต่อย่างใด

5. ถ้าบัพตัดสินใจไม่มีสายงานนำเข้าของการตัดสินใจ (Decision Input Flow) และสายงานควบคุมแล้ว ให้ถือว่าการตัดสินใจนี้ไม่มีตัวแปรนำเข้า
6. ถ้าบัพตัดสินใจไม่มีสายงานนำเข้าของการตัดสินใจและสายงานวัตถุแล้ว ให้ถือว่าการตัดสินใจมีเพียงหนึ่งตัวแปรเข้าคือโทเค็นวัตถุที่มาจากเส้นเชื่อมเข้า
7. ถ้าบัพตัดสินใจมีทั้งสายงานนำเข้าของการตัดสินใจและสายงานควบคุมแล้ว ให้ถือว่าการตัดสินใจมีเพียงหนึ่งตัวแปรเข้าคือโทเค็นวัตถุที่มาจากสายงานนำเข้าของการตัดสินใจ
8. ถ้าบัพตัดสินใจมีทั้งสายงานนำเข้าของการตัดสินใจและสายงานวัตถุแล้ว ให้ถือว่าการตัดสินใจมีสองตัวแปร โดยตัวแปรแรกคือโทเค็นวัตถุที่ไม่ได้มาจากสายงานนำเข้าของการตัดสินใจ ส่วนอีกตัวแปรคือโทเค็นวัตถุที่มาจากสายงานนำเข้าของการตัดสินใจ

3.4.10 **เงื่อนไขบังคับของ** MergeNode

1. บัพผสานจะมีเส้นเชื่อมออกได้เพียงหนึ่งเส้นเท่านั้น
2. เส้นเชื่อมเข้าและเส้นออกของบัพผสาน จะต้องเป็นสายงานวัตถุทั้งหมดหรือสายงานควบคุมทั้งหมดอย่างใดอย่างหนึ่ง

3.4.11 **เงื่อนไขบังคับของ** JoinNode

1. บัพรวมมีเส้นเชื่อมออกได้เพียงหนึ่งเส้นเท่านั้น
2. ถ้าบัพรวมมีเส้นเชื่อมเข้าเป็นสายงานวัตถุจะต้องมีเส้นเชื่อมออกเป็นสายงานวัตถุเช่นเดียวกันถ้าเส้นเชื่อมเข้าเป็นสายงานควบคุมจะต้องมีเส้นเชื่อมออกเป็นสายงานควบคุม

3.4.12 **เงื่อนไขบังคับของ** ForkNode

1. บัพแยกมีได้เพียงหนึ่งเส้นเชื่อมเข้าเท่านั้น
2. เส้นเชื่อมเข้าและออกจากบัพแยพ จะต้องเป็นสายงานวัตถุทั้งหมดหรือสายงานควบคุมทั้งหมดอย่างใดอย่างหนึ่ง

3.5 **การแจงส่วนเอกสารเอดีแอล**

ขั้นตอนสุดท้ายคือการสร้างแบบจำลองความหมายของเอดีแอลโดยการแจงส่วนนั้น มีข้อมูลนำเข้าคือ เอกสารเอดีแอลที่มีความถูกต้องจากขั้นตอนก่อนหน้า ไวยากรณ์แอนท์เลอร์เอดี

แอล และเมทาโมเดลของภาษาเอดีแอล ดังภาพที่ 4 เพื่อเป็นส่วนกำหนดกรอบผลลัพธ์ และเป็นส่วนให้ข้อมูลรายละเอียดในการแจงส่วน

แบบจำลองความหมายของเอดีแอลที่ได้นั้นประกอบไปด้วยรายละเอียดต่างๆ ได้แก่ บัพหลักฐานวัตถุ วัตถุเงื่อนไข และความสัมพันธ์ แบบจำลองความหมายที่ได้เป็นประโยชน์สำหรับการใช้งานต่อไป เช่น การสร้างกรณีทดสอบ การสร้างเอกสารออกแบบ การสร้างรหัสต้นแบบ

บทที่ 4
การออกแบบและพัฒนาระบบ

4.1 สถาปัตยกรรมระบบ

ระบบที่พัฒนาขึ้นแบ่งออกเป็น 3 ส่วนใหญ่ๆ คือ 1) ส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน 2) ส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล และ 3) ส่วนสร้างแบบจำลองความหมายของเอดีแอลและตรวจทานแผนภาพกิจกรรม แต่ละส่วนมีการพัฒนาแยกออกจากกันเป็นปลั๊กอินอีคลิปส์อย่างละตัว โดยทั้ง 3 ส่วนมีความสัมพันธ์กัน แสดงดังภาพที่ 15 ระบบจะเริ่มต้นด้วยการนำข้อมูลนำเข้าคือไฟล์ต้นฉบับจากซอฟต์แวร์โอเพนซอร์สมาผ่านกระบวนการแปลงแผนภาพกิจกรรมให้อยู่ในรูปแบบเอ็กซ์เอ็มไอมาตรฐาน จากนั้นนำเอกสารที่ได้ไปผ่านกระบวนการแปลงให้อยู่ในรูปแบบเอดีแอล สุดท้ายเอกสารเอดีแอลที่ได้จะถูกนำไปทวนสอบและตรวจสอบความสมเหตุสมผลระหว่างแผนภาพกิจกรรม ซึ่งได้รายงานแสดงผลลัพธ์การทวนสอบ แต่ถ้าหากทวนสอบแล้วพบว่าเกิดกรณีที่ไม่ตรงตามมาตรฐานขึ้น รายงานจะบ่งชี้ว่าแผนภาพนั้นขาดคุณสมบัติในข้อใดบ้างผ่านทางพื้นที่แสดงผลของซอฟต์แวร์ เมื่อผ่านขั้นตอนการทวนสอบแล้วก็จะนำเอกสารเอดีแอลที่ได้นั้นไปแจงส่วนเอดีแอล ซึ่งจะได้ผลลัพธ์คือแบบจำลองความหมายของเอดีแอลที่สามารถนำไปใช้ประยุกต์ใช้งานในด้านอื่นๆ สุดท้ายงานในทุกๆ ส่วนจะถูกรวมเป็นหมวดหมู่ของปลั๊กอินแล้วสร้างชุดของไซด์อัพเดทขึ้นมา แล้วจึงนำชุดของไซต์อัพเดทไปขึ้นสู่ระบบเว็บไซต์ เพื่อเปิดช่องทางให้นักวิจัยสามารถเข้าถึงได้สะดวกจากส่วนการติดตั้งปลั๊กอินเพิ่มเติมอัตโนมัติของอิคลิปส์

4.2 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา

สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒาระบบประกอบด้วยรายการฮาร์ดแวร์และซอฟต์แวร์ดังต่อไปนี้

4.2.1 สภาพแวดล้อม

1. คอมพิวเตอร์แมคบุ๊คแอร์ รุ่นปี 2012
2. หน่วยประมวลผลอินเทล คอร์ไอไฟว์รหัส 3427 ยู 1.80 กิกะเฮิร์ต
3. หน่วยความจำ 8 กิกะไบต์
4. ฮาร์ดดิสก์โซลิดสเตตไดรฟ์ความจุ 128 กิกะไบต์
5. ระบบปฏิบัติการไมโครซอฟต์วินโดว์ 7 แบบ 64 บิท และระบบปฏิบัติการแมคโอเอสเอ็กซ์ 10.8.2 แบบ 64 บิท

4.2.2 เครื่องมือที่ใช้ในการพัฒนา

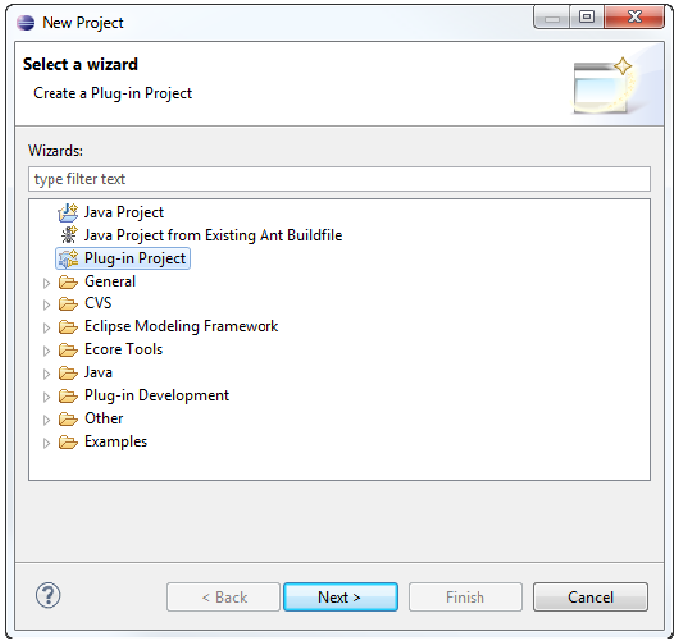
1. อีคลิปส์ 3.7.2
2. ชุดเครื่องมือพัฒนาจาวา 7
3. อีคลิปส์โมเดลลิ่งทูล 1.4.2
4. อัลโก้ยูเอ็มแอล 0.34
5. โมเดลลิโอ 2.2.1
6. ปลั๊กอินเอดีแอล 1.0.0.201

4.3 การพัฒนาระบบ

การพัฒนาระบบนั้น ส่วนต่างๆ ในระบบจะถูกพัฒนาโดยซอฟต์แวร์อีคลิปส์โมเดลลิ่งทูลเป็นเครื่องมือหลักและติดตั้งปลั๊กอินแอดีแอลเพื่อใช้งานร่วมในงานที่พัฒนา ระบบที่ได้จะอยู่ในลักษณะของปลั๊กอินอีคลิปส์ ที่แต่ละส่วนงานนั้นแยกออกจากกันอย่างชัดเจนตามแนวคิดของส่วนประกอบร่วม

4.3.1 การสร้างโพรเจกต์ปลั๊กอินอีคลิปส์

การสร้างโพรเจกต์ปลั๊กอินอีคลิปส์ เริ่มจากเราทำการเลือกสร้างโพรเจกต์ใหม่ แล้วเลือกสร้างปลั๊กอินโพรเจกต์ ดังภาพที่ 24



ภาพที่ 24 การสร้างสร้างปลั๊กอินโพรเจกต์

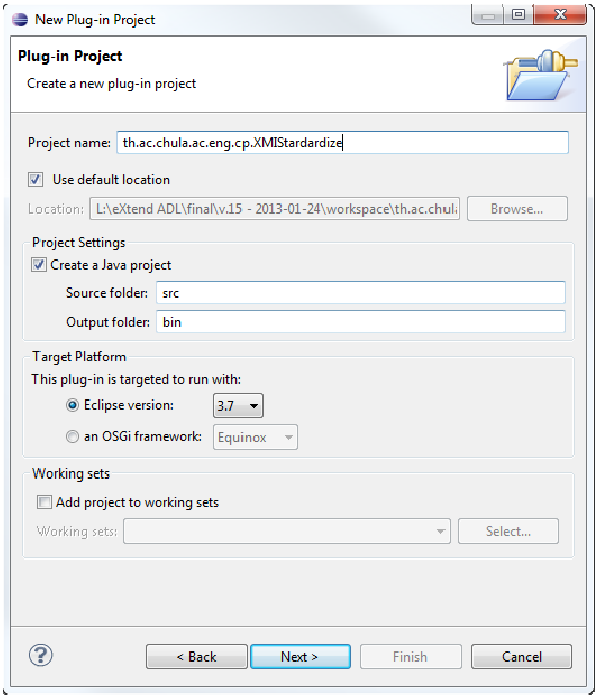
หลังจากนั้นจะเป็นการกำหนดรายละเอียดปลั๊กอินที่เราสร้างขึ้นมา ได้แก่ กำหนดชื่อของโพรเจกต์และรายละเอียดการสนับสนุนของอีคลิปส์ กำหนดคุณสมบัติของปลั๊กอินและกำหนดเทมเพลตของปลั๊กอิน ดังภาพที่ 25, 26 และ 27 ตามลำดับ

การกำหนดชื่อของโพรเจกต์และรายละเอียดการสนับสนุนของอีคลิปส์ ประกอบไปด้วยรายละเอียดที่สำคัญคือ ต้องการให้โพรเจกต์เป็นชนิดจาวาหรือไม่ และต้องการให้จัดเก็บไฟล์ต้นทางไว้ที่ใด ไฟล์ผลลัพธ์ไว้ที่ใด อีกทั้งระบุว่าต้องการให้ปลั๊กอินรองรับเวอร์ชันต่ำสุดที่เวอร์ชั่นใดในตัวอย่างจะให้รองรับที่เวอร์ชั่น 3.7 ซึ่งเป็นเวอร์ชั่นที่ใช้ในการพัฒนา

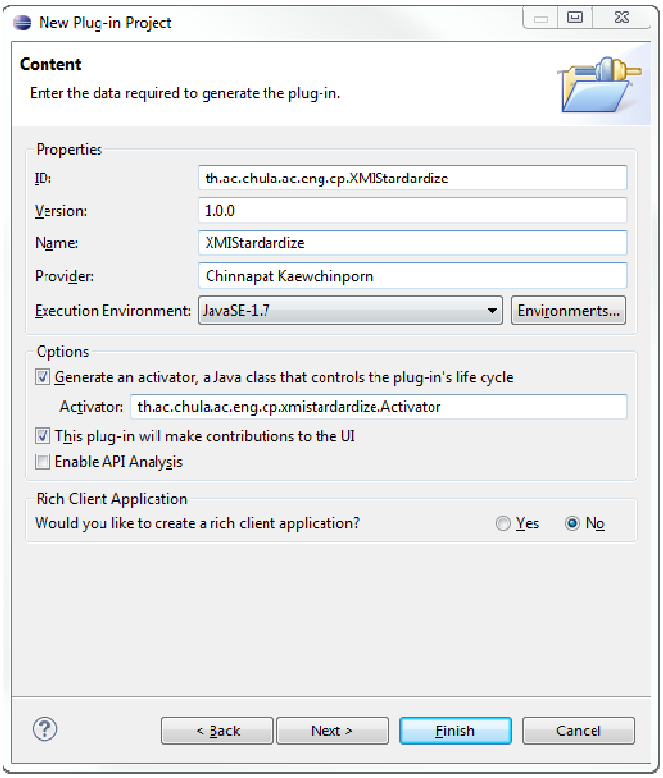
การกำหนดคุณสมบัติของปลั๊กอิน ประกอบไปด้วยรายละเอียดที่สำคัญคือ การกำหนดค่าต่างๆ ซึ่งได้แก่ รหัสระบุตัวตน เวอร์ชั่น ชื่อปลั๊กอิน ชื่อของนักพัฒนาและเวอร์ชั่นของจาวาที่ต้องการใช้ในการทำงานปลั๊กอิน

การกำหนดเทมเพลตของปลั๊กอิน จะเป็นรูปแบบสำเร็จรูปที่ทางอีคลิปส์เตรียมไว้ให้ แต่ในการพัฒนานี้นั้นเราจะไม่ใช้เทมเพลต

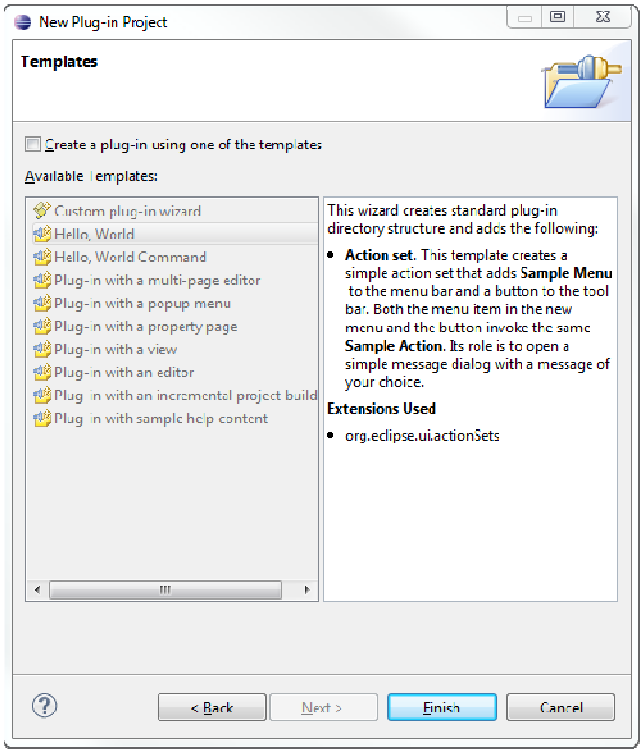
หลังจากที่ทำการสร้างโพรเจกต์ปลั๊กอินอีคลิปส์แล้ว ขั้นตอนต่อไปคือการกำหนดรายละเอียดไฟล์แมนนิเฟล็กของปลั๊กอิน การอัพโหลดไอคอนสำหรับเมนู และสร้างไฟล์คลาสต่างๆ ที่เกี่ยวข้องด้วยภาษาจาวา ซึ่งแต่ละปลั๊กอินก็จะมีรายละเอียดที่แตกต่างกันไป



ภาพที่ 25 กำหนดชื่อของโพรเจกต์และรายละเอียดการสนับสนุนของอีคลิปส์



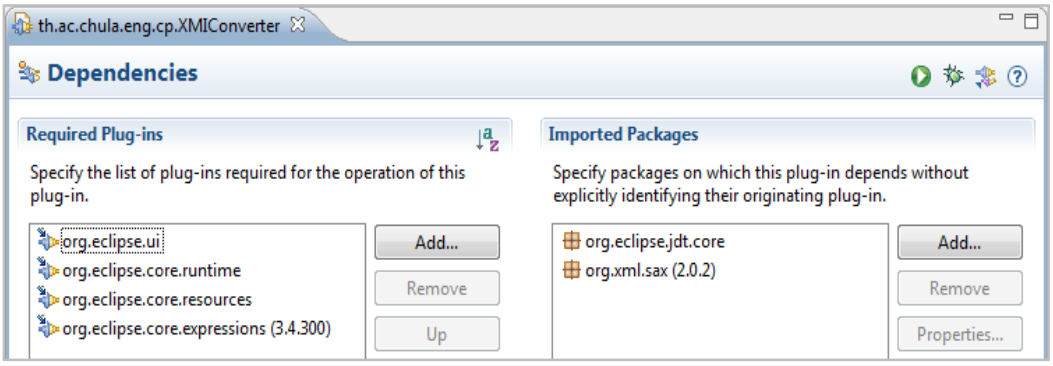
ภาพที่ 26 กำหนดคุณสมบัติของปลั๊กอิน



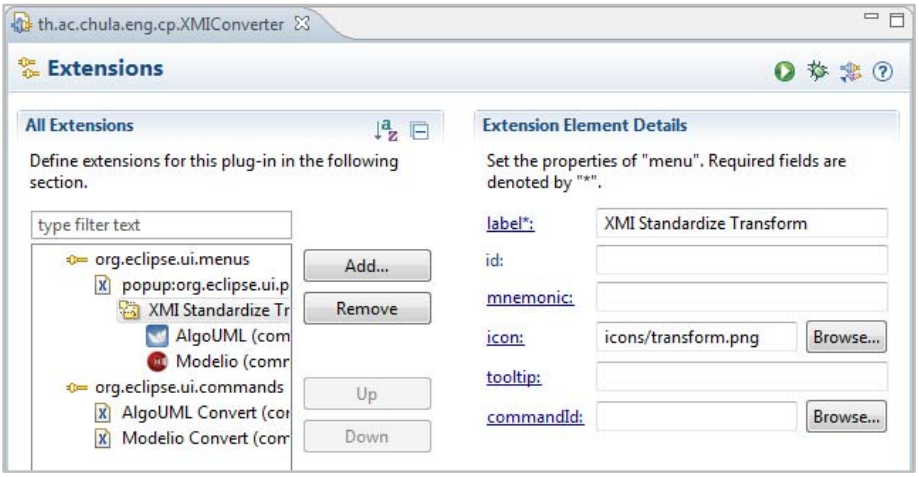
ภาพที่ 27 กำหนดเทมเพลตของปลั๊กอิน

4.3.2 การสร้างส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน

การพัฒนาส่วนการแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน เริ่มต้นด้วยการสร้างโพรเจกต์ปลั๊กอินอีคลิปส์ขึ้นมา จากนั้นทำการกำหนดรายละเอียดไฟล์แมนนิเฟล็กต่างๆ ที่สำคัญได้แก่ Dependencies และ Extensions การกำหนดรายละเอียดต่างๆ อ่านรายละเอียดเพิ่มเติมที่ภาคผนวก



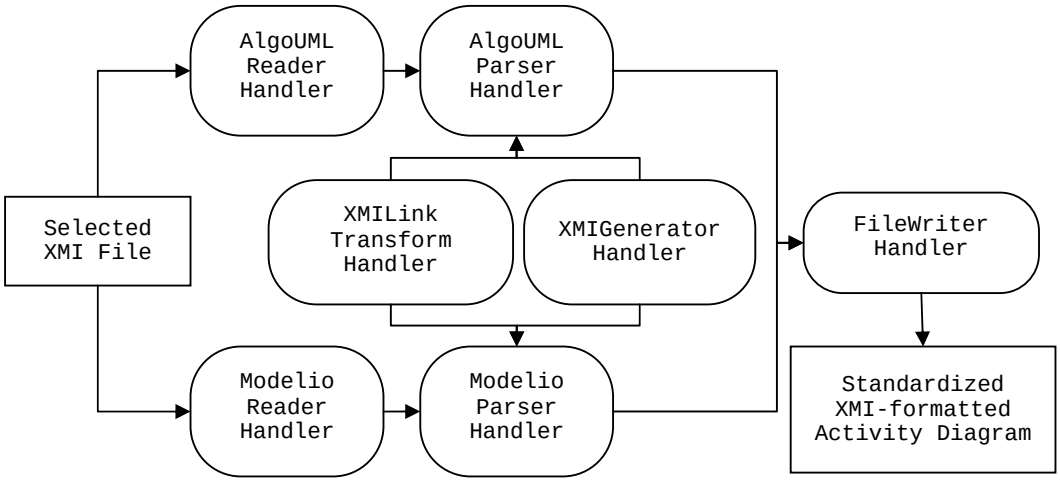
ภาพที่ 28 หน้า Dependencies ของปลั๊กอินส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน



ภาพที่ 29 หน้า Extensions ของปลั๊กอินส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน

หลังจากกำหนดค่าต่างๆ ในไฟล์แมนนิเฟล็กเรียบร้อย ขั้นตอนต่อไปจะเป็นการเขียนคลาสไฟล์ต่างๆ โดยความสัมพันธ์ของแต่ละคลาสแสดงได้ดังภาพที่ 30 ซึ่งแต่ละคลาสจะมีหน้าที่ต่างกันดังนี้

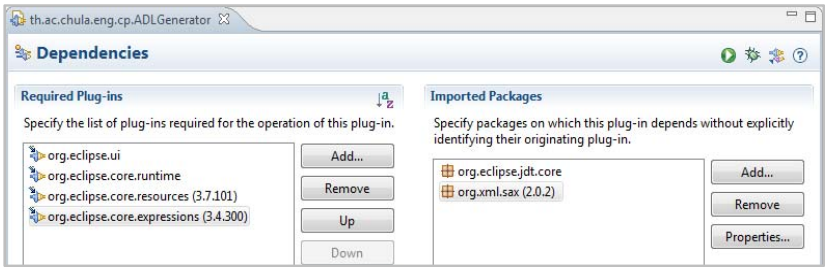
- AlgoUMLReaderHandler และ ModelioReaderHandler ทำหน้าที่อ่านไฟล์เอ็กซ์เอ็มไอที่เลือก และเรียกคลาสแจงส่วนข้อมูลขึ้นมาทำงาน
- AlgoUMLParserHandler และ ModelioParserHandler ทำหน้าที่เป็นตัวกลางในการสร้างเอกสารเอ็มเอ็มไอรูปแบบมาตรฐาน หน้าที่หลักคือแจงส่วนไฟล์เอ็กซ์เอ็มไอที่เลือกแล้วจัดเก็บข้อมูลที่แจงส่วนเรียบร้อยไว้ในอาเรย์ลิส หน้าที่รองคือผสานงานกับ XMLLinkTransformHandler และ XMIGeneratorHandler ผ่านอาเรย์ลิสและนำผลลัพธ์ที่ได้ที่จาก XMIGeneratorHandler ส่งไปให้คลาส FileWriterHandler ทำงานต่อไป
- XMLLinkTransformHandler ทำหน้าที่เป็นตัวแปลงรหัสที่ซอฟต์แวร์ออกแบบยูเอ็มแอลกำหนดมา ให้อยู่ในรูปแบบรหัสอ้างอิงที่ตัวแจงส่วนเข้าใจ สร้างบัพเริ่มต้นและบัพสิ้นสุดกิจกรรมขึ้นเมื่อตรวจสอบแล้วไม่พบบัพใดบัพหนึ่งในแอเรย์ลิส และสร้างบัพหยุดสายงานสำหรับเชื่อมต่อกับแอ็กชันที่ไม่มีเส้นเชื่อมออก
- XMIGeneratorHandler ทำหน้าที่สร้างเอ็กซ์เอ็มไอรูปแบบมาตรฐานขึ้นมาจากข้อมูลอาเรย์ลิส ใช้หลักการเช่นเดียวกับการท่องกราฟ แล้วสร้างคำสั่งขึ้นมารองรับในแต่ละบัพที่เจอด้วยวิธีการตามที่อธิบายในบทที่ 3
- FileWriterHandler ทำหน้าที่เขียนไฟล์เอ็กซ์เอ็มไอที่สร้างเรียบร้อยแล้ว ลงไปในตำแหน่งที่ผู้ใช้เป็นผู้กำหนดขึ้นมา ซึ่งไฟล์ที่ได้จะอยู่ในรูปแบบเอกสารเอ็กซ์เอ็มไอที่อยู่ในรูปแบบมาตรฐาน



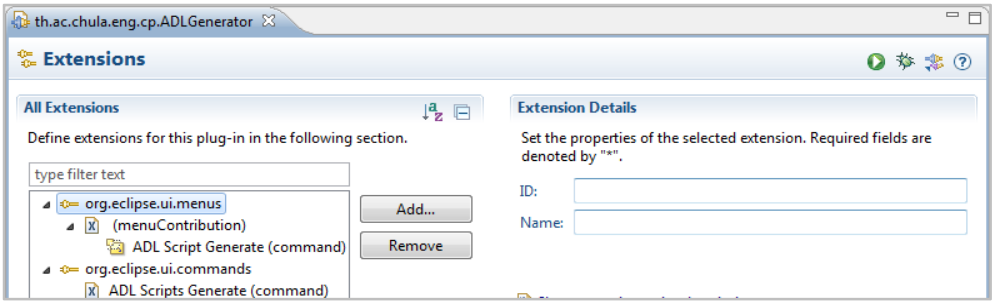
ภาพที่ 30 การทำงานของคลาสในปลั๊กอินส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน

4.3.3 การสร้างส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล

การพัฒนาส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล เริ่มต้นด้วยการสร้างโพรเจกต์ปลั๊กอินอีคลิปส์ขึ้นมา จากนั้นทำการกำหนดรายละเอียดไฟล์แมนนิเฟล็กต่างๆ ที่สำคัญได้แก่ Dependencies และ Extensions การกำหนดรายละเอียดต่างๆ อ่านรายละเอียดเพิ่มเติมที่ภาคผนวก

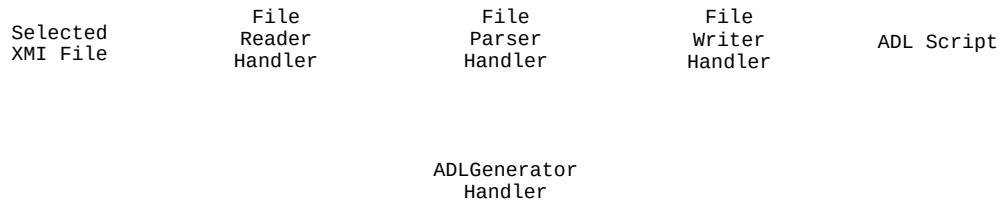


ภาพที่ 31 หน้า Dependencies ของปลั๊กอินส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล



ภาพที่ 32 หน้า Extensions ของปลั๊กอินส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล

หลังจากกำหนดค่าต่างๆ ในไฟล์แมนนิเฟล็กเรียบร้อย ขั้นตอนต่อไปจะเป็นการเขียนคลาสไฟล์ต่างๆ โดยความสัมพันธ์ของแต่ละคลาสแสดงได้ในภาพที่ 33



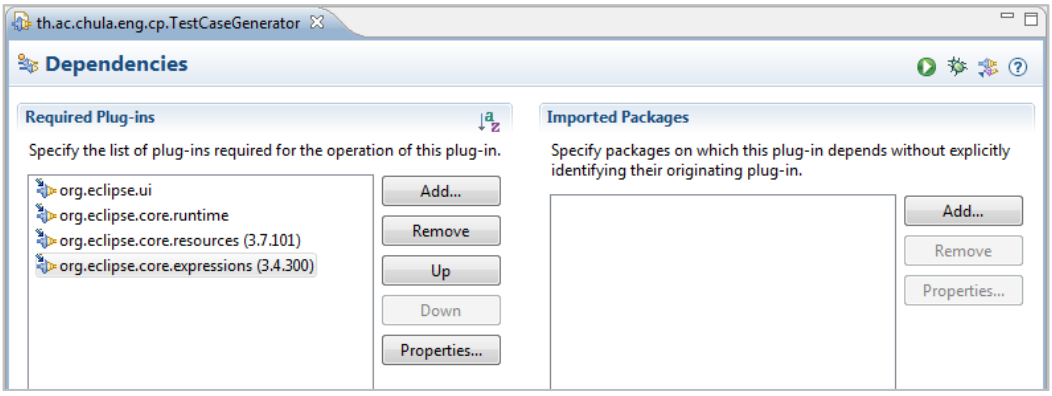
ภาพที่ 33 การทำงานของคลาสในปลั๊กอินส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล

แต่ละคลาสจะมีหน้าที่ต่างกัน ดังนี้

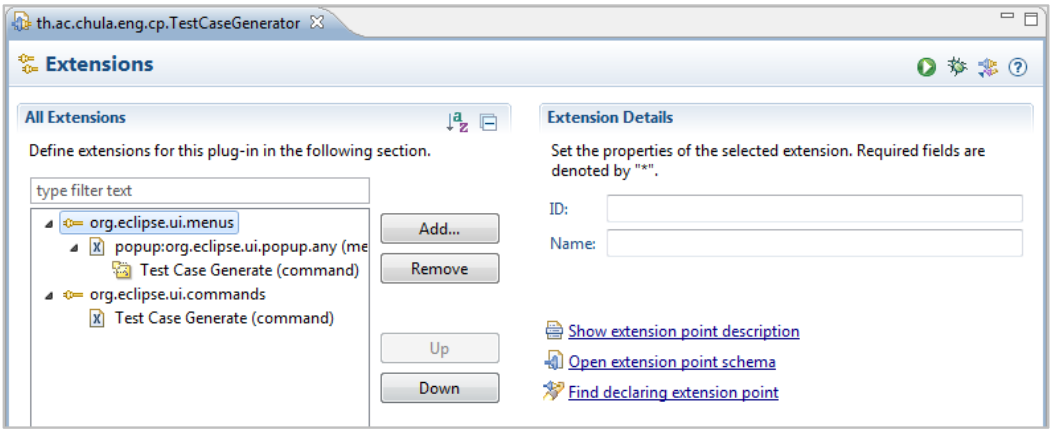
- FileReaderHandler ทำหน้าที่อ่านไฟล์เอ็กซ์เอ็มไอที่เลือก และเรียกคลาสแจงส่วนข้อมูลขึ้นมาทำงาน
- FileParserHandler ทำหน้าที่เป็นตัวกลางในการสร้างเอกสารเอดีแอล หน้าที่หลักคือแจงส่วนไฟล์เอ็กซ์เอ็มไอที่เลือกแล้วจัดเก็บข้อมูลที่แจงส่วนเรียบร้อยไว้ในอาเรย์ลิส หน้าที่รองคือผสานงานกับ ADLGeneratorHandler ผ่านอาเรย์ลิสและนำผลลัพธ์ที่ได้ที่จาก ADLGeneratorHandler ส่งไปให้คลาส FileWriterHandler ทำงานต่อไป
- ADLGeneratorHandler ทำหน้าที่สร้างเอดีแอลขึ้นมาจากข้อมูลอาเรย์ลิส ใช้หลักการเช่นเดียวกับการท่องกราฟ แล้วสร้างคำสั่งขึ้นมารองรับในแต่ละบัพที่เจอด้วยวิธีการตามที่อธิบายในบทที่ 3
- FileWriterHandler ทำหน้าที่เขียนไฟล์เอดีแอลที่สร้างเรียบร้อยแล้ว ลงไปในตำแหน่งที่ผู้ใช้เป็นผู้กำหนดขึ้นมา ซึ่งไฟล์ที่ได้จะอยู่ในรูปแบบเอดีแอล

4.3.4 การสร้างส่วนสร้างแบบจำลองความหมายของเอดีแอลและตรวจทานแผนภาพกิจกรรม

การพัฒนาส่วนสร้างแบบจำลองความหมายของเอดีแอลและตรวจทานแผนภาพกิจกรรม เริ่มต้นด้วยการสร้างโพรเจกต์ปลั๊กอินอีคลิปส์ขึ้นมา จากนั้นทำการกำหนดรายละเอียดไฟล์แมนนิเฟล็กต่างๆ ที่สำคัญ ได้แก่ Dependencies และ Extensions การกำหนดรายละเอียดต่างๆ อ่านรายละเอียดเพิ่มเติมที่ภาคผนวก



ภาพที่ 34 หน้า Dependencies ของปลั๊กอินสร้างแบบจำลองความหมายของเอดีแอลและตรวจทานแผนภาพกิจกรรม



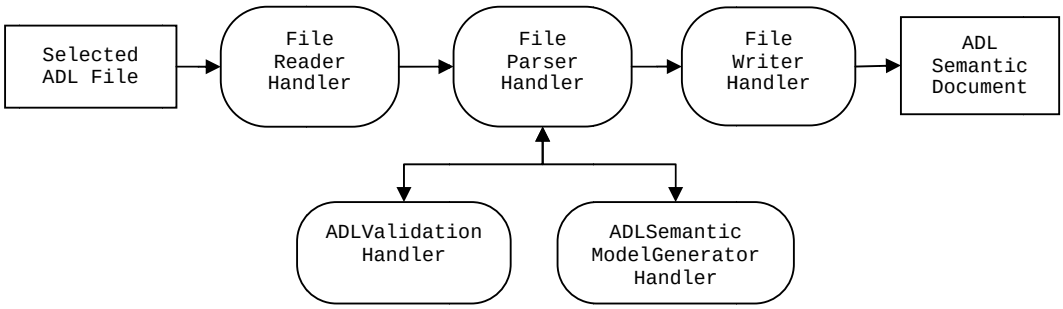
ภาพที่ 35 หน้า Extensions ของปลั๊กอินสร้างแบบจำลองความหมายของเอดีแอลและตรวจทานแผนภาพกิจกรรม

หลังจากกำหนดค่าต่างๆ ในไฟล์แมนนิเฟล็กเรียบร้อย ขั้นตอนต่อไปจะเป็นการเขียนคลาสไฟล์ต่างๆ โดยความสัมพันธ์ของแต่ละคลาสแสดงได้ดังภาพที่ 36

แต่ละคลาสจะมีหน้าที่ต่างกัน ดังนี้

- FileReaderHandler ทำหน้าที่อ่านไฟล์เอดีแอลที่เลือก และเรียกคลาสแจงส่วนข้อมูลขึ้นมาทำงาน
- FileParserHandler ทำหน้าที่เป็นตัวกลางในการสร้างรายงานและแบบจำลองความหมายของเอดีแอล หน้าที่หลักคือแจงส่วนไฟล์เอดีแอลที่เลือกแล้วจัดเก็บข้อมูลที่แจงส่วนเรียบร้อยไว้ในอาเรย์ลิส จากนั้นผสานงานกับ ADLValidationHandler และ ADLSemanticModelGeneratorHandler ผ่านอาเรย์ลิสและนำผลลัพธ์ที่ได้จาก ADLValidationHandler แสดงผลผ่านพื้นที่แสดงผลของซอฟต์แวร์ และนำผลลัพธ์จาก ADLSemanticModelGeneratorHandler ส่งไปให้คลาส FileWriterHandler ทำงานต่อไป
- ADLValidationHandler ทำหน้าที่ทวนสอบว่าแผนภาพกิจกรรมสอดคล้องตามมาตรฐานยูเอ็มแอล 2.4.1 [3] หรือไม่ โดยการตรวจสอบนั้น จะพิจารณาว่าแผนภาพกิจกรรมนั้นผ่านเงื่อนไขบังคับแต่ละคลาสครบทุกข้อหรือไม่ หากขาดเพียงข้อใดข้อหนึ่งจะถือว่าแผนภาพกิจกรรมนั้นไม่สอดคล้องทันที และบันทึกข้อความผิดพลาดไว้ว่าผิดเงื่อนไขบังคับในข้อใด เมื่อตรวจสอบครบทุกเงื่อนไขบังคับจะส่งผลลัพธ์ที่ได้กลับไปให้ FileParserHandler ทำการจัดการเรื่องการแสดงผลต่อไป

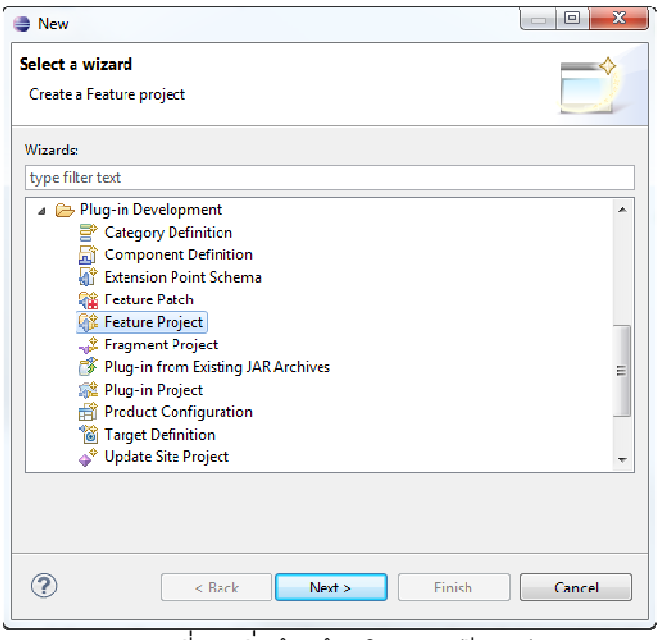
- ADLSemanticModelGeneratorHandler ทำหน้าที่สร้างแบบจำลองความหมายของเอดีแอลขึ้นมาโดยใช้ข้อมูลของอาเรย์ลิส ซึ่งสร้างผ่าน FileWriterHandler
- FileWriterHandler ทำหน้าที่เขียนไฟล์เอดีแอลที่สร้างเรียบร้อยแล้ว ลงไปในตำแหน่งที่ผู้ใช้เป็นผู้กำหนดขึ้นมา ผลลัพธ์ที่ได้จะอยู่ในรูปแบบเอกสารเอ็กซ์เอ็มไอ



ภาพที่ 36 การทำงานของคลาสในปลั๊กอินสร้างส่วนสร้างแบบจำลองความหมายของเอดีแอลและตรวจทานแผนภาพกิจกรรม

4.3.5 การสร้างส่วนไซต์อัพเดท ช่องทางสำหรับติดตั้งปลั๊กอิน

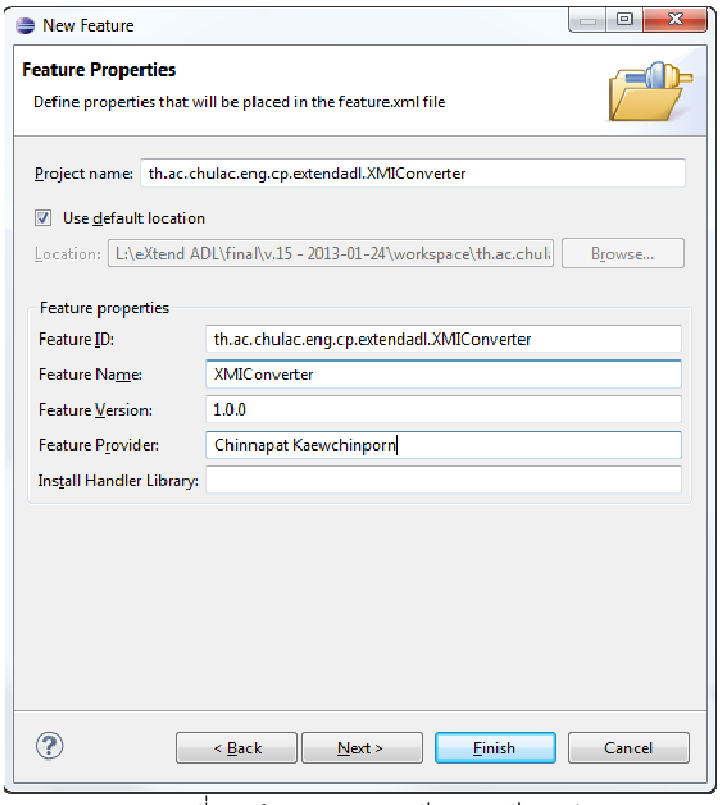
การพัฒนาสร้างส่วนไซต์อัพเดทนั้น จำเป็นต้องสร้างโพรเจกต์ฟีเจอร์ขึ้นมาเพิ่มตามจำนวนปลั๊กอินที่ต้องการสร้าง การสร้างโพรเจกต์ฟีเจอร์เริ่มจากเราทำการเลือกสร้างโพรเจกต์ใหม่แล้วเลือกสร้างโพรเจกต์ฟีเจอร์ ดังภาพที่ 37



ภาพที่ 37 เริ่มต้นสร้างโพรเจกต์ฟีเจอร์

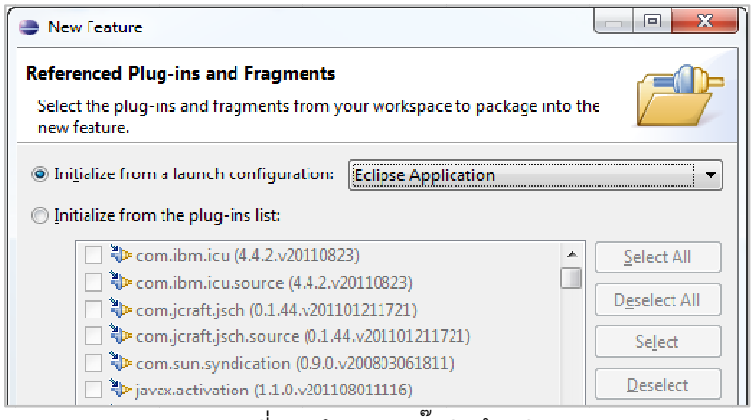
หลังจากนั้นจะเป็นการกำหนดรายละเอียดของฟีเจอร์ที่เราต้องการสร้างขึ้นมารายละเอียดที่ต้องการจะเป็นชุดข้อมูลเดียวกับรายละเอียดของปลั๊กอินโปรเจกดังภาพที่ 38

การกำหนดรายละเอียดของฟีเจอร์ ประกอบไปด้วยรายละเอียดที่สำคัญคือ การกำหนดค่าต่างๆ ซึ่งได้แก่ รหัสระบุตัวตน ชื่อฟีเจอร์ เวอร์ชั่น ชื่อของนักพัฒนาและไลบรารี่ และการกำหนดพื้นที่ในการติดตั้งฟีเจอร์ ซึ่งในตัวอย่างจะใช้ค่าโดยปริยายที่กำหนดโดยอีคลิปส์



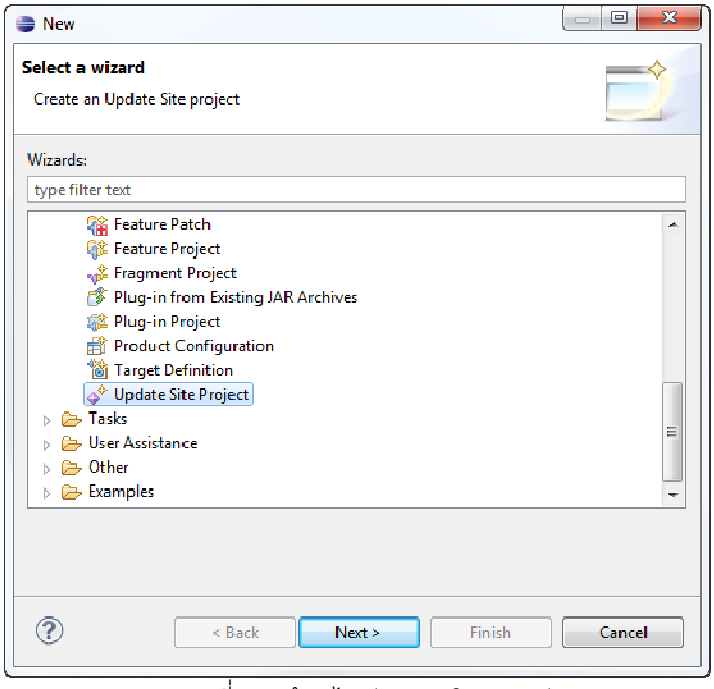
ภาพที่ 38 กำหนดรายละเอียดของฟีเจอร์

หลังจากนั้นจะเป็นการกำหนดปลั๊กอินอ้างอิงให้เลือก “Initialize from a launch configuration Eclipse Application” แล้วกดสิ้นสุดการสร้างดังภาพที่ 39



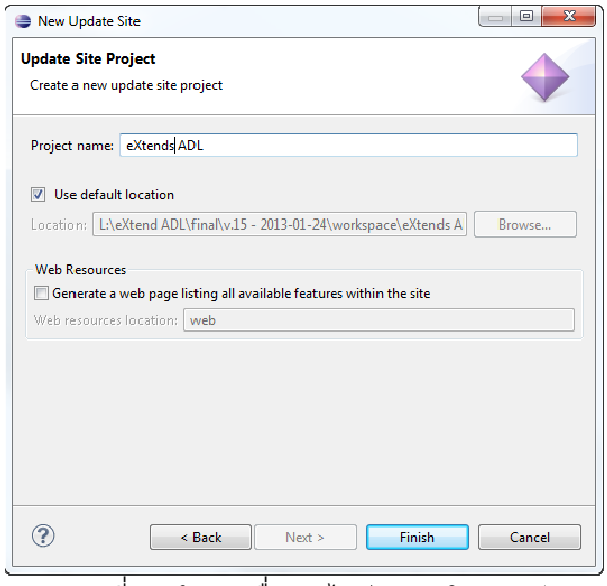
ภาพที่ 39 กำหนดปลั๊กอินอ้างอิง

จากนั้นเมื่อสร้างครบทั้งทุกฟีเจอร์แล้ว ให้เราสร้างไซต์อัพเดทโพรเจกต์ เริ่มจากเราทำการเลือกสร้างโพรเจกต์ใหม่ แล้วเลือกสร้างไซต์อัพเดทโพรเจกต์ ดังภาพที่ 40



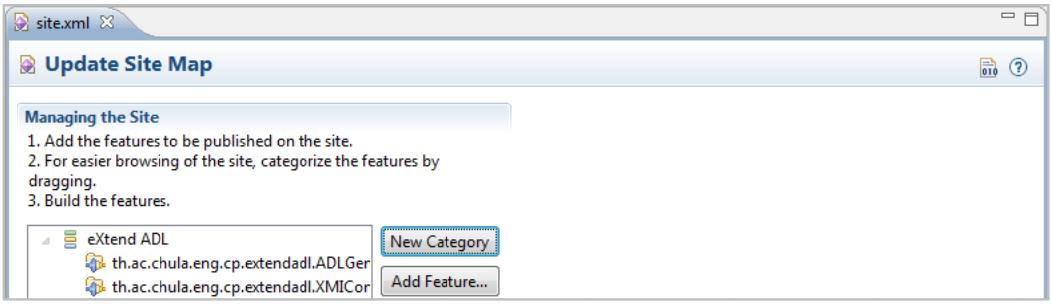
ภาพที่ 40 สร้างไซต์อัพเดทโพรเจกต์

หลังจากนั้นทำการกำหนดชื่อของโพรเจกต์ แล้วกดสิ้นสุดดังภาพที่ 41



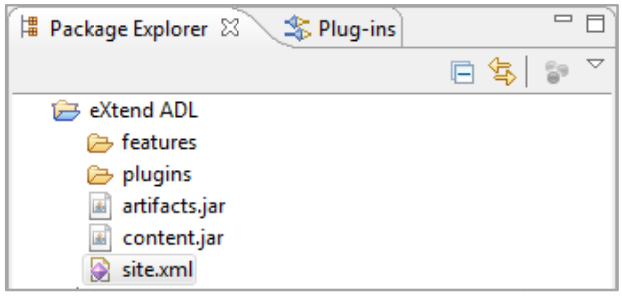
ภาพที่ 41 กำหนดชื่อของไซต์อัพเดทโพรเจกต์

ที่ไฟล์ site.xml นั้นจะเป็นตัวสร้างไซต์อัพเดทขึ้นมา โดยการสร้างนั้นจะสร้างหมวดหมู่ใหม่ขึ้นมาในที่นี้ชื่อว่า “eXtend ADL” จากนั้นกำหนดฟีเจอร์ของหมวดหมู่ โดยเลือกปุ่มเพิ่มฟีเจอร์แล้วเลือกฟีเจอร์ทุกสร้างที่สร้างไว้ ผลลัพธ์แสดงได้ดังภาพที่ 42



ภาพที่ 42 หน้าหลักของตัวสร้างไซต์อัพเดท

สุดท้ายเมื่อตรวจสอบแล้วพบว่าทุกอย่างถูกต้องให้กดที่ปุ่มสร้างทั้งหมด ซึ่งจะได้ผลลัพธ์เป็นแพ็คเกจสำหรับติดตั้งในเว็บไซต์ โดยผลลัพธ์นั้นจะไปเพิ่มอยู่ในไฟล์โพรเจกต์ดังภาพที่ 43

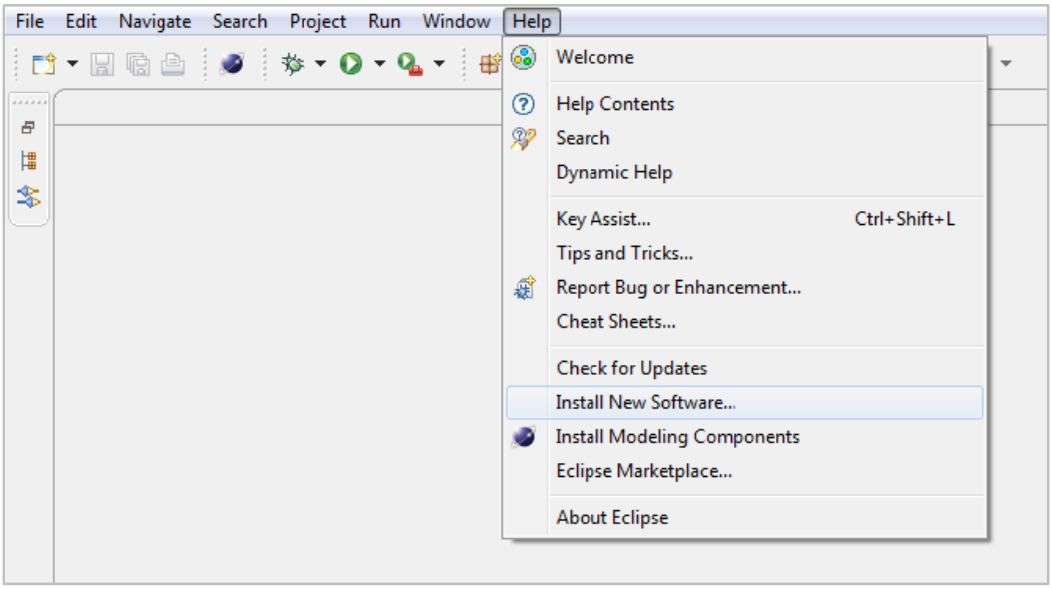


ภาพที่ 43 แพ็คเกจสำหรับติดตั้งในเว็บไซต์

4.3.6 การติดตั้งปลั๊กอินที่พัฒนา

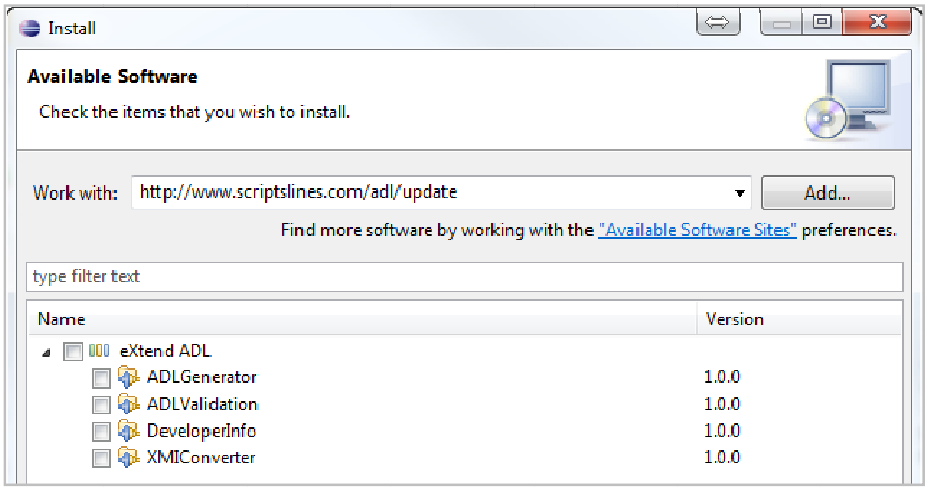
การติดตั้งปลั๊กอินที่พัฒนานั้นทำได้ 2 วิธี คือ การคัดลอกแล้วนำปลั๊กอินมาวางในโพลเดอร์ซอฟต์แวร์ที่พัฒนา และอีกวิธีคือการติดตั้งผ่านอินเตอร์เน็ตโดยการดาวน์โหลดปลั๊กอินจากไซต์อัพเดท วิธีการมีขั้นตอนดังนี้

เข้าไปที่เมนูช่วยเหลือ เลือกติดตั้งซอฟต์แวร์ใหม่ ดังภาพที่ 44



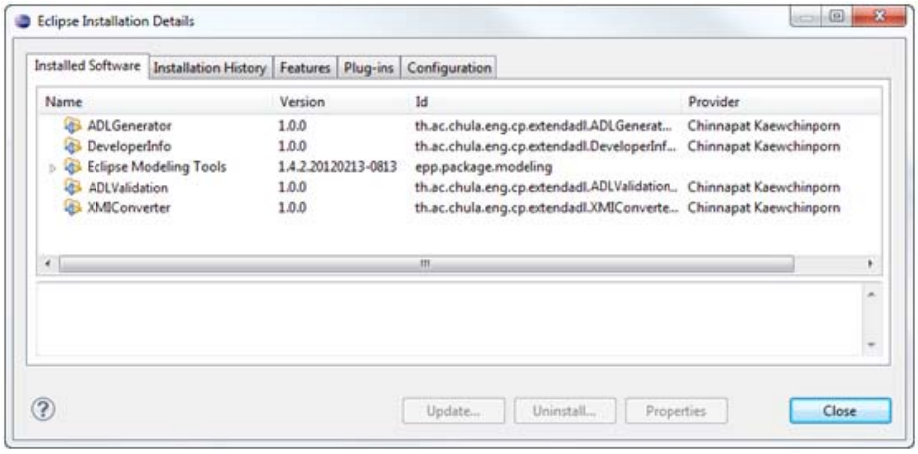
ภาพที่ 44 เมนูติดตั้งซอฟต์แวร์ใหม่

ที่ส่วนติดตั้งนั้นพิมพ์ตำแหน่งของไซต์อัพเดท ในที่นี้คือ http://www.scriptslines.com/adl/update จากนั้นกดเพิ่ม แล้วส่วนของปลั๊กอินจะแสดงให้เห็นด้านล่าง เราสามารถเลือกที่จะติดตั้งปลั๊กอินทั้งหมวดหมู่ หรือปลั๊กอินบางตัวก็ได้เช่นกัน ดังภาพที่ 45

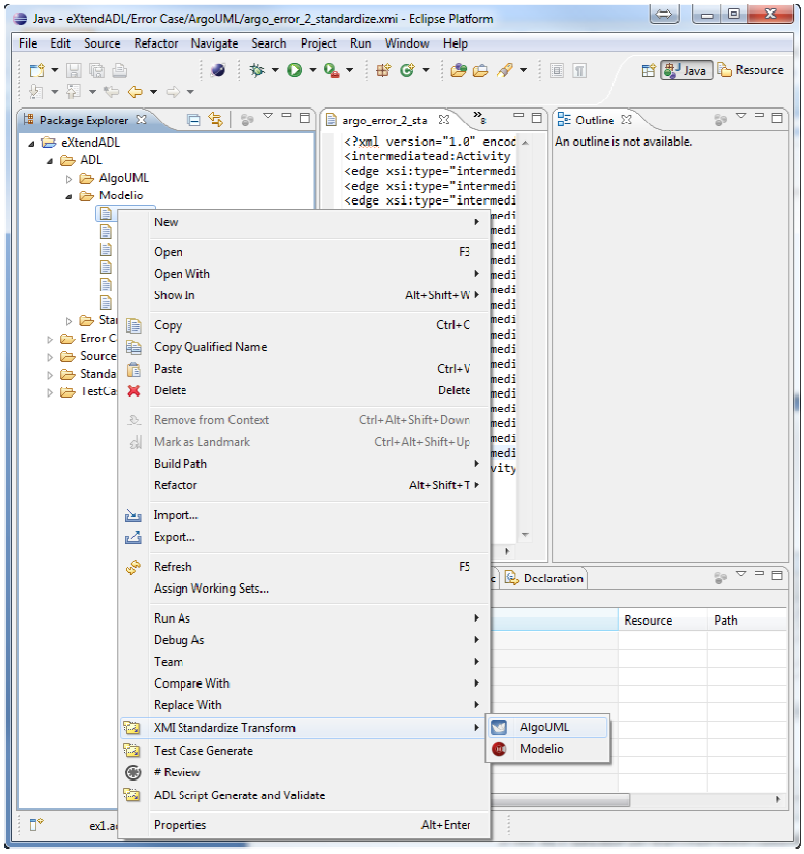


ภาพที่ 45 ส่วนติดตั้งปลั๊กอินเพิ่มเติม

หลังจากนั้นจะเป็นรายละเอียดใบอนุญาตต่างๆ และข้อตกลงในการใช้งาน เมื่อพิจารณาเรียบร้อยแล้วก็จะเริ่มติดตั้งและพร้อมใช้งานปลั๊กอิน เราสามารถตรวจสอบว่าติดตั้งปลั๊กอินทั้งหมดเรียบเรียบหรือไม่ โดยการเข้าไปตรวจสอบที่เมนูรายละเอียดการติดตั้งอีคลิปส์ดังภาพที่ 46 และสามารถใช้งานปลั๊กอินต่างๆ ได้ผ่านการคลิกขวาที่ไฟล์บนตัวจัดการแพ็กเกจดังภาพที่ 47



ภาพที่ 46 รายละเอียดปลั๊กอินทั้งหมดที่ติดตั้งในซอฟต์แวร์



ภาพที่ 47 การใช้งานปลั๊กอินทั้งหมดที่พัฒนาขึ้น

บทที่ 5
การประเมินและการวัดผล

5.1 แนวทางการประเมินผลงานวิจัย

แนวทางการประเมินจะใช้วิธีการเปรียบเทียบระหว่างประชากรตัวอย่างกับผลลัพธ์ที่ได้จากซอฟต์แวร์ของระบบ โดยข้อมูลตัวอย่างจะแบ่งออกเป็น 2 กลุ่ม คือกลุ่มที่หนึ่งเป็นข้อมูลแผนภาพกิจกรรมจากเอกสารยูเอ็มแอลของโอเอ็มจี [1] จำนวน 6 ตัวอย่าง และกลุ่มที่สองเป็นข้อมูลแผนภาพกิจกรรมที่ผิดตามเงื่อนไขข้อบังคับตามมาตรฐานยูเอ็มแอลของโอเอ็มจี 2.4.1 [3] จำนวน 2 ตัวอย่าง ทั้งนี้เพื่อให้มั่นใจได้ว่าข้อมูลตัวอย่างมีความถูกต้องตามมาตรฐานที่กำหนด สำหรับการเลือกข้อมูลตัวอย่างจะใช้เกณฑ์การเลือกโดยประชากรที่เลือกมาจะสอดคล้องและครอบคลุมกับแนวคิดในบทที่ 3 โดยแนวทางการประเมินประกอบด้วยรายละเอียดต่างๆ ดังต่อไปนี้

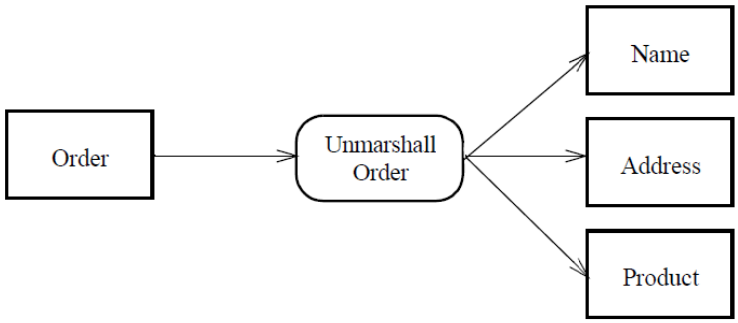
1. คำอธิบายแผนภาพกิจกรรม
2. ข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส
3. ข้อมูลอธิบายแผนภาพกิจกรรมด้วยเอดีแอลของงานวิจัย [4]
4. ผลลัพธ์เอดีแอลของระบบ
5. ผลลัพธ์แผนภาพกิจกรรมของระบบ
6. ผลลัพธ์การทวนสอบแผนภาพกิจกรรม
7. ข้อสังเกต

สำหรับการวัดผลนั้นจะใช้การเปรียบเทียบความแตกต่างของเอดีแอลที่ได้ของงานวิจัย [4] กับเอดีแอลที่ได้จากระบบ และความแตกต่างของแผนภาพกิจกรรมของระบบกับแผนภาพข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส

5.2 ผลการเปรียบเทียบตัวอย่างที่ 1

5.2.1 คำอธิบายแผนภาพกิจกรรม

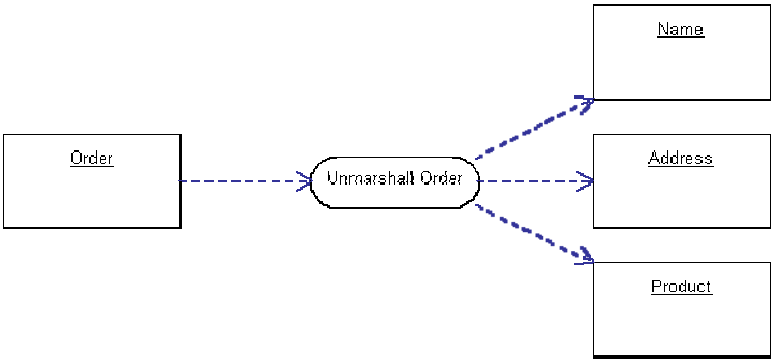
ตัวอย่างที่ 1 เป็นตัวอย่างของกลุ่มตัวอย่างที่หนึ่งซึ่งเป็นข้อมูลแผนภาพกิจกรรมจากเอกสารยูเอ็มแอลของโอเอ็มจี แสดงให้เห็นถึงความสามารถของแผนภาพกิจกรรมที่สามารถแสดงให้เห็นถึงการแจงรายละเอียดวัตถุได้ โดยการแสดงให้เห็นถึงอินพุตและเอาต์พุตของแอ็กชัน ดังภาพที่ 48



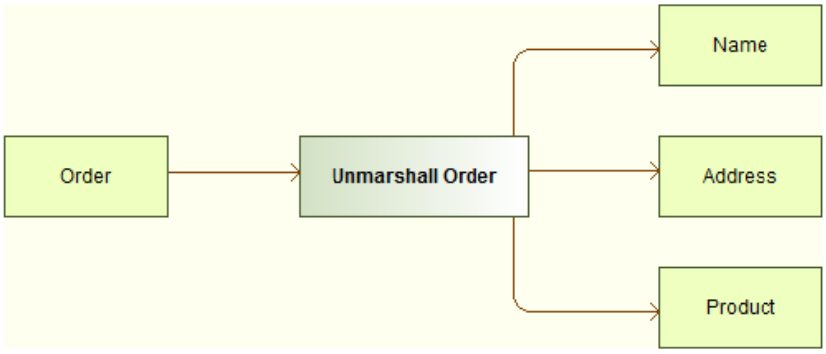
ภาพที่ 48 แผนภาพกิจกรรมตัวอย่างที่ 1 [3]

5.2.2 ข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส

ตัวอย่างที่ 1 สามารถพัฒนาด้วยซอฟต์แวร์โอเพนซอร์ส โดยผลลัพธ์จากซอฟต์แวร์อัลโก้ยูเอ็มแอลแสดงได้ดังภาพที่ 49 และผลลัพธ์จากซอฟต์แวร์โมเดลลิโอแสดงได้ดังภาพที่ 50



ภาพที่ 49 แผนภาพกิจกรรมตัวอย่างที่ 1 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล



ภาพที่ 50 แผนภาพกิจกรรมตัวอย่างที่ 1 จากซอฟต์แวร์โมเดลลิโอ

5.2.3 ข้อมูลอธิบายแผนภาพกิจกรรมด้วยเอดีแอลของงานวิจัย [4]

แผนภาพกิจกรรมตัวอย่างที่ 1 สามารถเขียนให้อยู่ในรูปบทคำสั่งเอดีแอลได้ดังต่อไปนี้

```
diagram 'ex1'
      action unmarshallOrder
            <- Order
            -> Name, Address, Product
      end
end
```

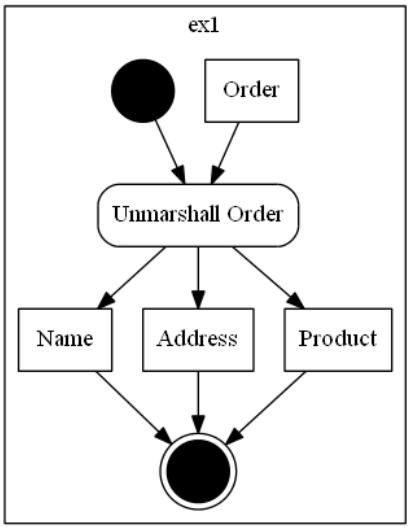
5.2.4 ผลลัพธ์เอดีแอลของระบบ

ผลลัพธ์ที่ได้จากการแปลงแผนภาพกิจกรรมตัวอย่างที่ 1 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังนี้

```
diagram 'ex1'
      action Unmarshall Order
            <- Order
            -> Name, Address, Product
      end
end
```

5.2.5 ผลลัพธ์แผนภาพกิจกรรมของระบบ

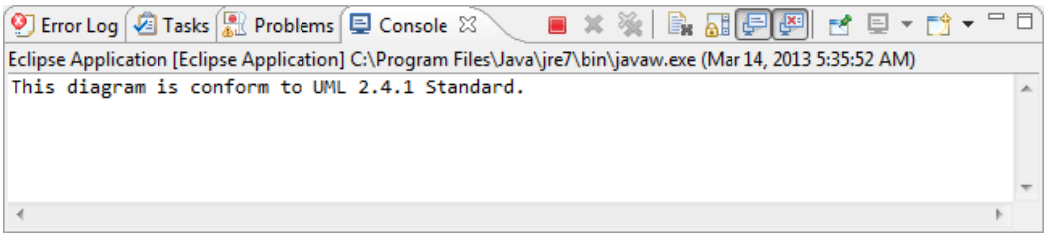
ผลลัพธ์ที่ได้จากการนำแบบจำลองความหมายของเอดีแอลตัวอย่างที่ 1 ไปสร้างแผนภาพกิจกรรมในรูปแบบกราฟวิซเพื่อแปลงเป็นรูปภาพ [4] แสดงได้ดังภาพที่ 51



ภาพที่ 51 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 1

5.2.6 ผลลัพธ์การทวนสอบแผนภาพกิจกรรม

ผลลัพธ์ที่ได้จากการทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 1 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังภาพที่ 52



ภาพที่ 52 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 1

5.2.7 ข้อสังเกต

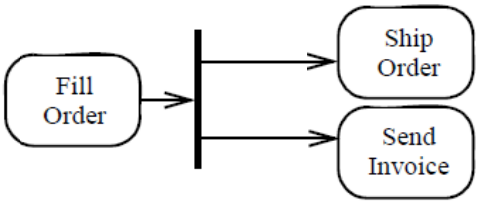
เอดีแอลที่ได้จากระบบ เมื่อเปรียบเทียบกับตัวอย่างข้อมูล พบว่าเหมือนกันทุกประการ และเมื่อเปรียบเทียบแผนภาพ แสดงให้เห็นว่าแผนภาพกิจกรรมที่ได้จากระบบมีการเพิ่มจุดเริ่มต้นและจุดสิ้นสุดของกิจกรรม ซึ่งจะแตกต่างจากตัวอย่างข้อมูล อย่างไรก็ตาม ผลลัพธ์ยังคงให้ความหมายเช่นเดียวกันกับแผนภาพตัวอย่าง

ผลลัพธ์การทวนสอบแผนภาพกิจกรรม แสดงให้เห็นว่าแผนภาพกิจกรรมสอดคล้องตามมาตรฐานยูเอ็มแอล 2.4.1

5.3 ผลการเปรียบเทียบตัวอย่างที่ 2

5.3.1 คำอธิบายแผนภาพกิจกรรม

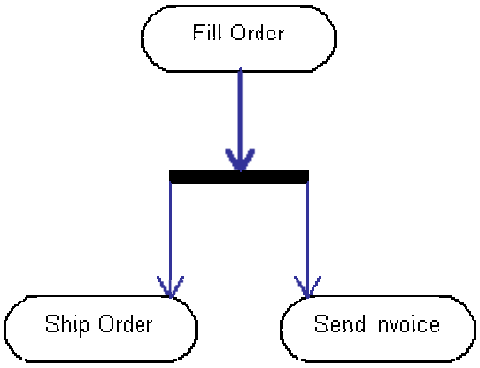
ตัวอย่างที่ 2 เป็นตัวอย่างของกลุ่มตัวอย่างที่หนึ่งซึ่งเป็นข้อมูลแผนภาพกิจกรรมจากเอกสารยูเอ็มแอลของโอเอ็มจี แสดงให้เห็นถึงความสามารถในการสร้างการดำเนินงานแบบคู่ขนานดังตัวอย่างภาพที่ 53 แสดงให้เห็นว่าจะมีการดำเนินงาน “Ship Order” และ “Send Invoice” ในการดำเนินงานลำดับถัดไปหลังจากแอ็กชัน “Fill Order” ทำงานเสร็จสิ้นแล้ว



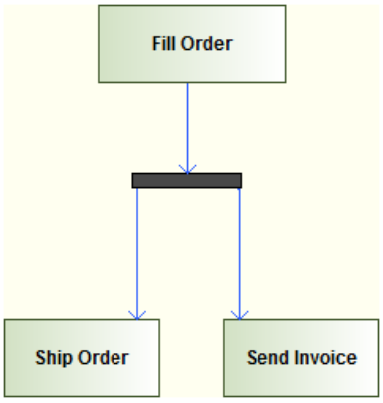
ภาพที่ 53 แผนภาพกิจกรรมตัวอย่างที่ 2 [3]

5.3.2 ข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส

ตัวอย่างที่ 2 สามารถพัฒนาด้วยซอฟต์แวร์โอเพนซอร์ส โดยผลลัพธ์จากซอฟต์แวร์อัลโก้ยูเอ็มแอลแสดงได้ดังภาพที่ 54 และผลลัพธ์จากซอฟต์แวร์โมเดลลิโอแสดงได้ดังภาพที่ 55



ภาพที่ 54 แผนภาพกิจกรรมตัวอย่างที่ 2 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล



ภาพที่ 55 แผนภาพกิจกรรมตัวอย่างที่ 2 จากซอฟต์แวร์โมเดลลิโอ

5.3.3 ข้อมูลอธิบายแผนภาพกิจกรรมด้วยเอดีแอลของงานวิจัย [4]

แผนภาพกิจกรรมตัวอย่างที่ 2 สามารถเขียนให้อยู่ในรูปบทคำสั่งเอดีแอลได้ดังต่อไปนี้

```
diagram 'ex2'
      fillOrder->shipOrder and sendInvoice
end
```

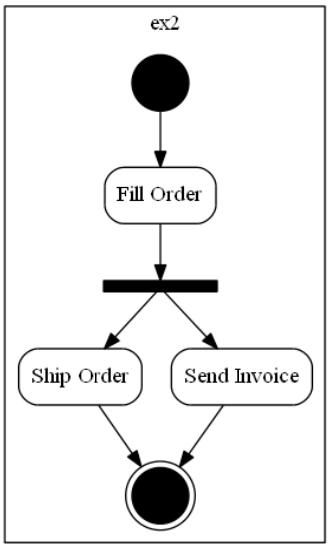
5.3.4 ผลลัพธ์เอดีแอลของระบบ

ผลลัพธ์ที่ได้จากการแปลงแผนภาพกิจกรรมตัวอย่างที่ 2 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล และโมเดลลิโอแสดงได้ดังนี้

```
diagram 'ex2'
      action Fill Order end
      action Ship Order end
      action Send Invoice end
      Fill Order->Ship Order and Send Invoice
end
```

5.3.5 ผลลัพธ์แผนภาพกิจกรรมของระบบ

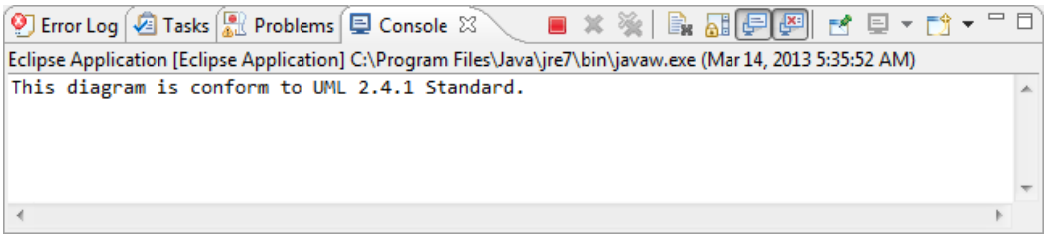
ผลลัพธ์ที่ได้จากการนำแบบจำลองความหมายของเอดีแอลตัวอย่างที่ 2 ไปสร้างแผนภาพกิจกรรมในรูปแบบกราฟวิซเพื่อแปลงเป็นรูปภาพ [4] แสดงได้ดังภาพที่ 56



ภาพที่ 56 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 2

5.3.6 ผลลัพธ์การทวนสอบแผนภาพกิจกรรม

ผลลัพธ์ที่ได้จากการทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 2 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังภาพที่ 57



ภาพที่ 57 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 2

5.3.7 ข้อสังเกต

เอดีแอลที่ได้จากระบบ เมื่อเปรียบเทียบตัวอย่างข้อมูล พบว่ามีการนิยามแอ็กชันเพิ่มเติมขึ้นมา นอกเหนือจากการใช้ลำดับของแอ็กชันอย่างเดียว และเมื่อเปรียบเทียบแผนภาพ แสดงให้เห็นว่าแผนภาพกิจกรรมที่ได้จากระบบมีการเพิ่มจุดเริ่มต้นและจุดสิ้นสุดของกิจกรรม โดยที่เริ่มต้นทำงานจาก “Fill Order” และสิ้นสุดหลังจากดำเนินการ “Ship Order” และ “Send Invoice” ซึ่งจะ

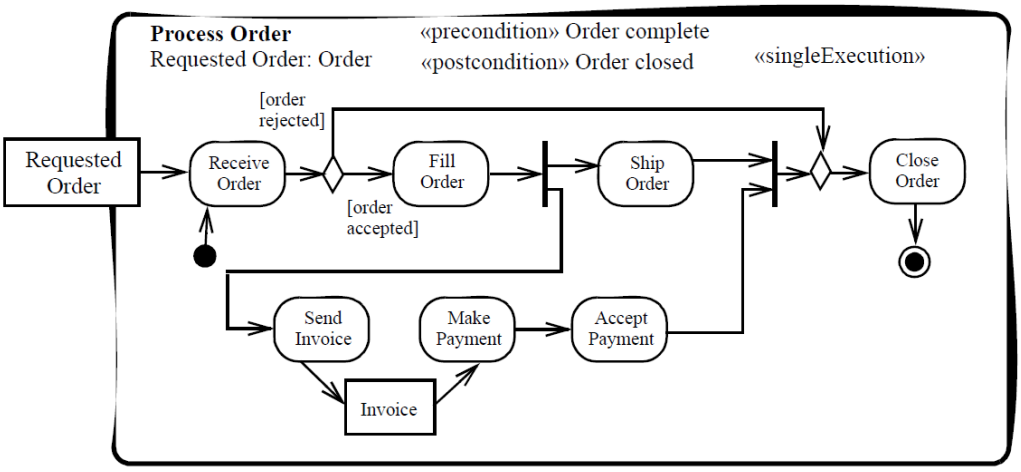
แตกต่างจากตัวอย่างข้อมูล อย่างไรก็ตามผลลัพธ์ยังคงให้ความหมายเช่นเดียวกันกับแผนภาพตัวอย่าง

ผลลัพธ์การทวนสอบแผนภาพกิจกรรม แสดงให้เห็นว่าแผนภาพกิจกรรมสอดคล้องตามมาตรฐานยูเอ็มแอล 2.4.1

5.4 ผลการเปรียบเทียบตัวอย่างที่ 3

5.4.1 คำอธิบายแผนภาพกิจกรรม

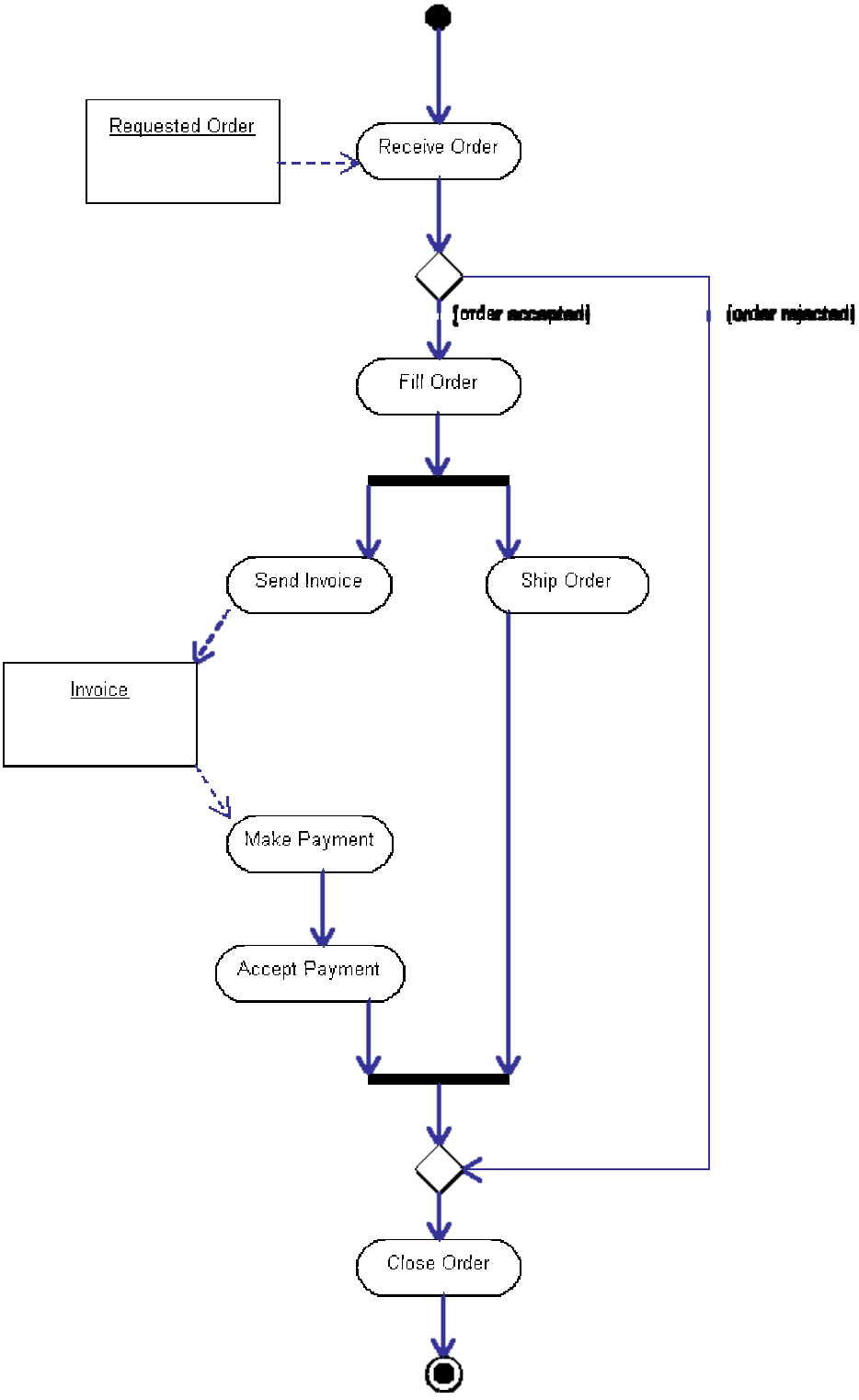
ตัวอย่างที่ 3 เป็นตัวอย่างของกลุ่มตัวอย่างที่หนึ่งซึ่งเป็นข้อมูลแผนภาพกิจกรรมจากเอกสารยูเอ็มแอลของโอเอ็มจี แสดงให้เห็นถึงการสร้างแผนภาพกิจกรรมทั่วๆ ไปซึ่งส่วนมากจะประกอบด้วยการสร้างลำดับการทำงาน การส่งผ่านวัตถุ การตัดสินใจ การทำงานแบบคู่ขนานและการรวมสายงานมากกว่าหนึ่งสายงานเข้าด้วยกัน



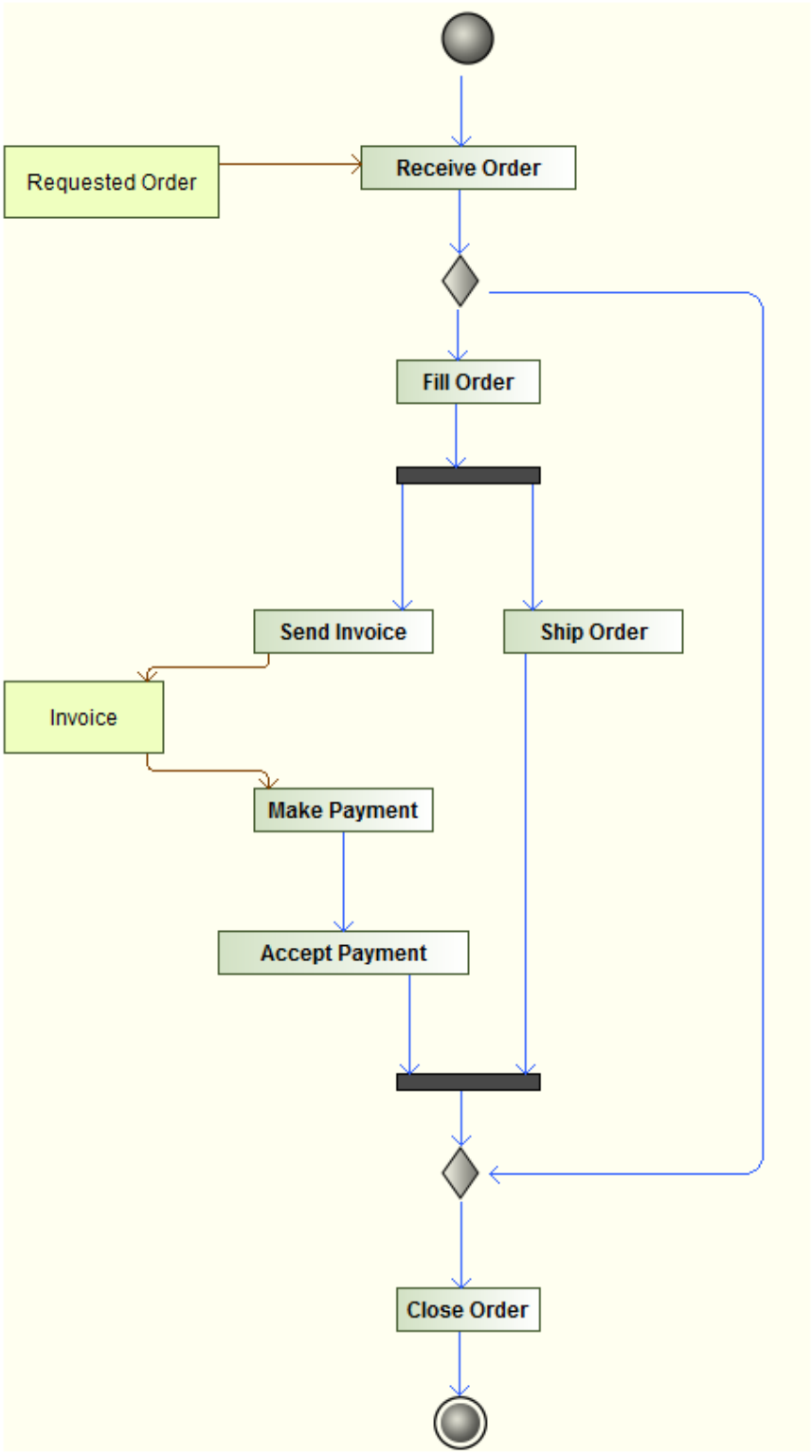
ภาพที่ 58 แผนภาพกิจกรรมตัวอย่างที่ 3 [3]

5.4.2 ข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส

ตัวอย่างที่ 3 สามารถพัฒนาด้วยซอฟต์แวร์โอเพนซอร์ส โดยผลลัพธ์จากซอฟต์แวร์อัลโก้ยูเอ็มแอลแสดงได้ดังภาพที่ 59 และผลลัพธ์จากซอฟต์แวร์โมเดลลิโอแสดงได้ดังภาพที่ 60



ภาพที่ 59 แผนภาพกิจกรรมตัวอย่างที่ 3 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล



ภาพที่ 60 แผนภาพกิจกรรมตัวอย่างที่ 3 จากซอฟต์แวร์โมเดลลิโอ

5.4.3 ข้อมูลอธิบายแผนภาพกิจกรรมด้วยเอดีแอลของงานวิจัย [4]

แผนภาพกิจกรรมตัวอย่างที่ 3 สามารถเขียนให้อยู่ในรูปบทคำสั่งเอดีแอลได้ดังต่อไปนี้

```
diagram 'process order'
       action receiveOrder
            <- RequestedOrder
       end
       action sendInvoice
            -> Invoice
       end

       decision from receiveOrder
            if 'order accepted' then fillOrder
            else
                  if 'order rejected' then closeOrder
                  endif
            endif
       end
       fillOrder->shipOrder and sendInvoice
       sendInvoice->makePayment->acceptPayment->closeOrder
       shipOrder->closeOrder
end
```

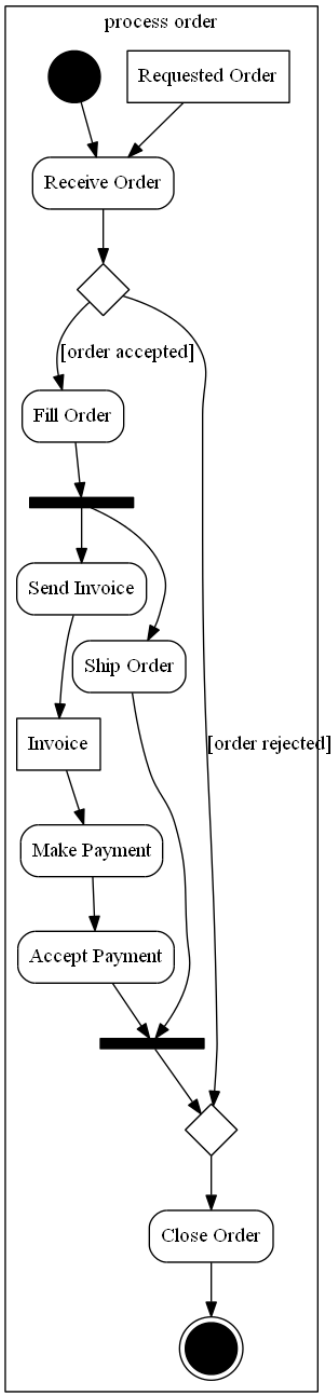
5.4.4 ผลลัพธ์เอดีแอลของระบบ

ผลลัพธ์ที่ได้จากการแปลงแผนภาพกิจกรรมตัวอย่างที่ 3 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังนี้

```
diagram 'process order'
      action Receive Order
            <- Requested Order,
      end
      action Send Invoice
            -> Invoice
      end
      action Fill Order end
      action Close Order end
      action Ship Order end
      action Make Payment
            <- Invoice
      end
      action Accept Payment end
      decision from Receive Order
            if 'order accepted' then Fill Order
            else
                  if 'order rejected' then Close Order
                  endif
            endif
      end
      Send Invoice->Make Payment->Accept Payment->Close Order
      Fill Order->Ship Order and Send Invoice
      Ship Order->Close Order
end
```

5.4.5 ผลลัพธ์แผนภาพกิจกรรมของระบบ

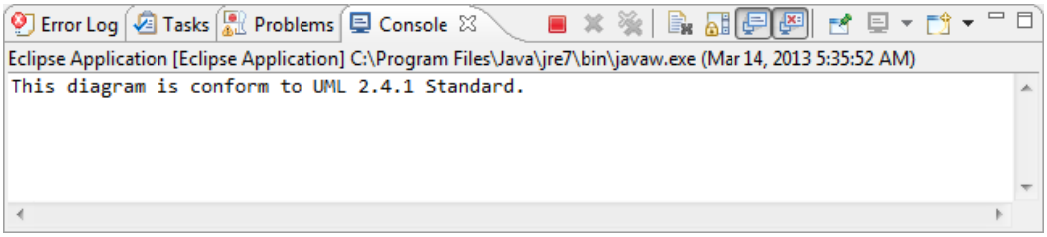
ผลลัพธ์ที่ได้จากการนำแบบจำลองความหมายของเอดีแอลตัวอย่างที่ 3 ไปสร้างแผนภาพกิจกรรมในรูปแบบกราฟวิซเพื่อแปลงเป็นรูปภาพ [4] แสดงได้ดังภาพที่ 61



ภาพที่ 61 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 3

5.4.6 ผลลัพธ์การทวนสอบแผนภาพกิจกรรม

ผลลัพธ์ที่ได้จากการทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 3 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังภาพที่ 62



ภาพที่ 62 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 3

5.4.7 ข้อสังเกต

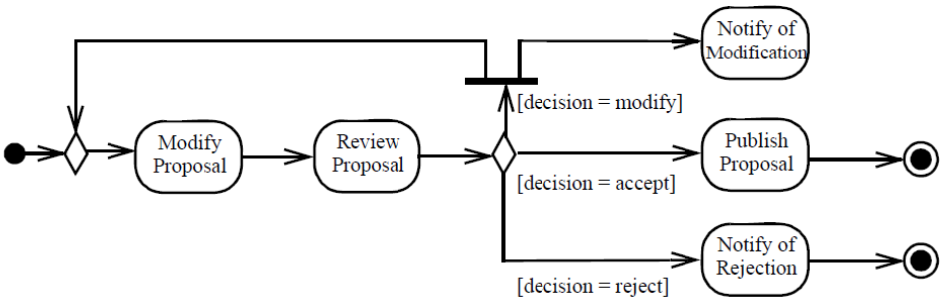
เอดีแอลที่ได้จากระบบ เมื่อเปรียบเทียบตัวอย่างข้อมูล พบว่ามีการนิยามแอ็กชันเพิ่มเติมขึ้นมาสำหรับแอ็กชันที่ไม่มีการเชื่อมโยงเข้าหรือออกกับบัฟวัตถุ สำหรับลำดับของแอ็กชันและการตัดสินใจเหมือนกันทุกประการ และเมื่อเปรียบเทียบแผนภาพ แสดงให้เห็นว่าแผนภาพกิจกรรมที่ได้จากระบบมีองค์ประกอบของแผนภาพกิจกรรมเหมือนกันทุกประการ

ผลลัพธ์การทวนสอบแผนภาพกิจกรรม แสดงให้เห็นว่าแผนภาพกิจกรรมสอดคล้องตามมาตรฐานยูเอ็มแอล 2.4.1

5.5 ผลการเปรียบเทียบตัวอย่างที่ 4

5.5.1 คำอธิบายแผนภาพกิจกรรม

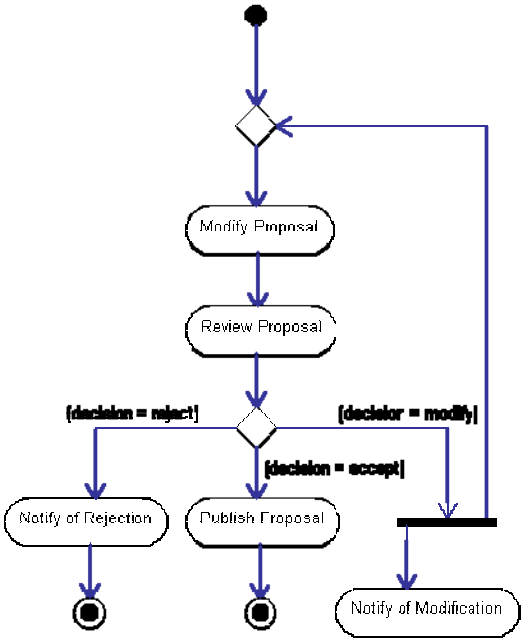
ตัวอย่างที่ 4 เป็นตัวอย่างของกลุ่มตัวอย่างที่หนึ่งซึ่งเป็นข้อมูลแผนภาพกิจกรรมจากเอกสารยูเอ็มแอลของโอเอ็มจี แสดงให้เห็นถึงการหยุดสายงานกิจกรรมและการวนซ้ำภายในกิจกรรม



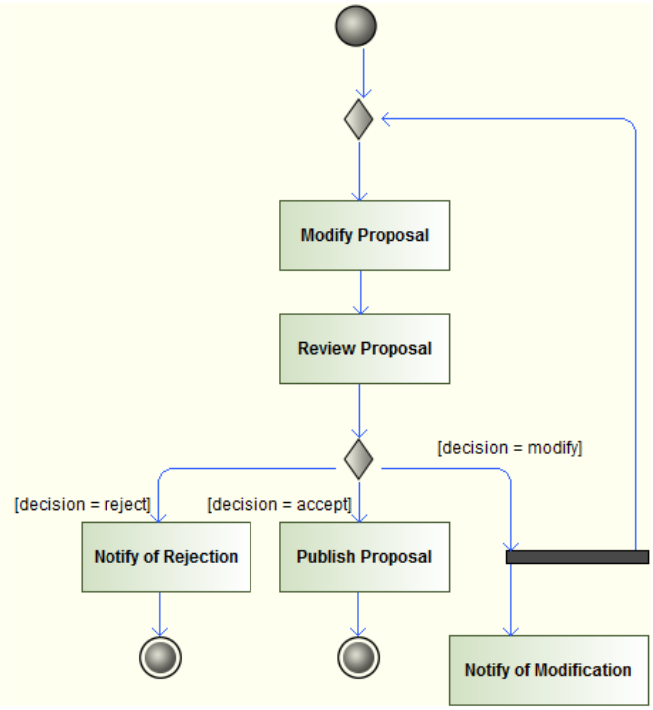
ภาพที่ 63 แผนภาพกิจกรรมตัวอย่างที่ 4 [3]

5.5.2 ข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส

ตัวอย่างที่ 4 สามารถพัฒนาด้วยซอฟต์แวร์โอเพนซอร์ส โดยผลลัพธ์จากซอฟต์แวร์อัลโก้ยูเอ็มแอลแสดงได้ดังภาพที่ 64 และผลลัพธ์จากซอฟต์แวร์โมเดลลิโอแสดงได้ดังภาพที่ 65



ภาพที่ 64 แผนภาพกิจกรรมตัวอย่างที่ 4 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล



ภาพที่ 65 แผนภาพกิจกรรมตัวอย่างที่ 4 จากซอฟต์แวร์โมเดลลิโอ

5.5.3 ข้อมูลอธิบายแผนภาพกิจกรรมด้วยเอดีแอลของงานวิจัย [4]

แผนภาพกิจกรรมตัวอย่างที่ 4 สามารถเขียนให้อยู่ในรูปบทคำสั่งเอดีแอลได้ดังต่อไปนี้

```
diagram 'ex4'
       modifyProposal->reviewProposal
       decision from reviewProposal
            if 'decision = modify'
            then notifyOfModification and modifyProposal
            else
                  if 'decision = accept' then publishProposal
                  else
                        if 'decision = reject'
                        then notifyOfRejection
                        endif
                  endif
            endif
       end
       notifyOfModification->break
end
```

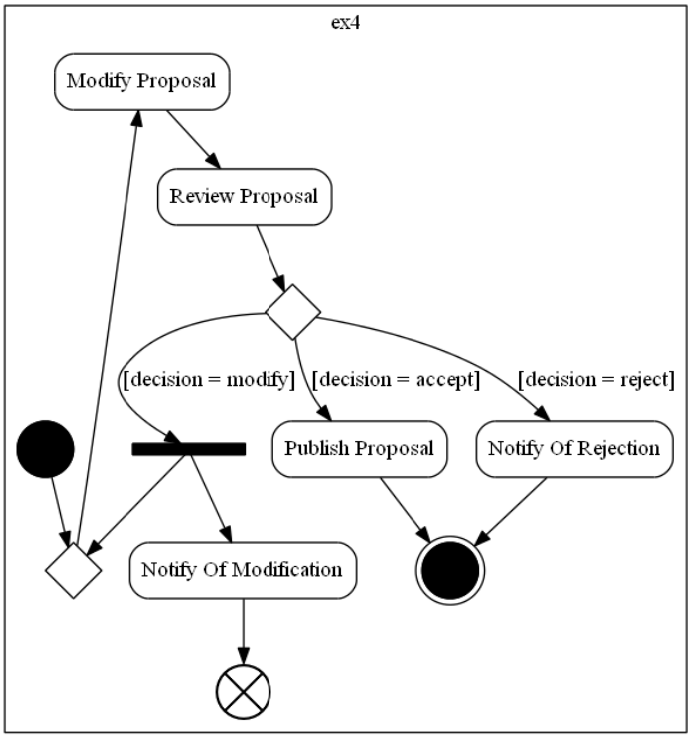
5.5.4 ผลลัพธ์เอดีแอลของระบบ

ผลลัพธ์ที่ได้จากการแปลงแผนภาพกิจกรรมตัวอย่างที่ 4 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังนี้

```
diagram 'ex4'
      action Modify Proposal end
      action Review Proposal end
      action Notify Of Modification end
      action Publish Proposal end
      action Notify Of Rejection end

      decision from Review Proposal
            if 'decision = modify' then Modify Proposal and
                  Notify Of Modification
            else
                  if 'decision = accept' then Publish Proposal
                  else
                        if 'decision = reject' then Notify Of
                              Rejection
                        endif
                  endif
            endif
      end
      Modify Proposal->Review Proposal
      Notify Of Modification->break
end
```

5.5.5 ผลลัพธ์แผนภาพกิจกรรมของระบบ

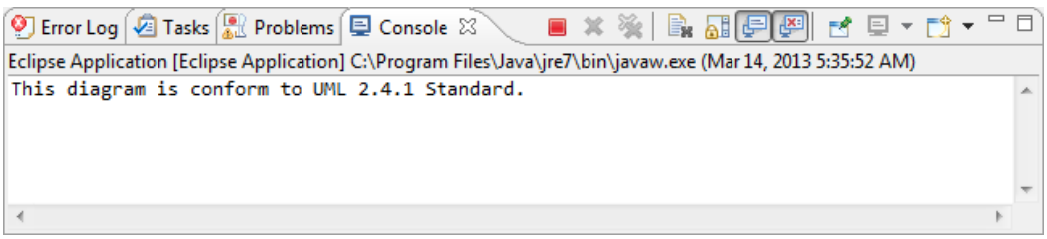
ผลลัพธ์ที่ได้จากการนำแบบจำลองความหมายของเอดีแอลตัวอย่างที่ 4 ไปสร้างแผนภาพกิจกรรมในรูปแบบกราฟวิซเพื่อแปลงเป็นรูปภาพ [4] แสดงได้ดังภาพที่ 66



ภาพที่ 66 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 4

5.5.6 ผลลัพธ์การทวนสอบแผนภาพกิจกรรม

ผลลัพธ์ที่ได้จากการทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 4 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังภาพที่ 67



ภาพที่ 67 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 4

5.5.7 ข้อสังเกต

เอดีแอลที่ได้จากระบบ เมื่อเปรียบเทียบตัวอย่างข้อมูล พบว่ามีการนิยามแอ็กชันเพิ่มเติมขึ้นมาสำหรับแอ็กชันที่ไม่มีการเชื่อมโยงเข้าหรือออกกับบัฟวัตถุ สำหรับลำดับของแอ็กชันและการตัดสินใจเหมือนกันทุกประการ และเมื่อเปรียบเทียบแผนภาพ เนื่องจากแผนภาพกิจกรรมตัวอย่างไม่มีการกำหนดการทำงานหลังจากแอ็กชัน“Notify of Modification” ทำให้สามารถอนุมานได้ว่า

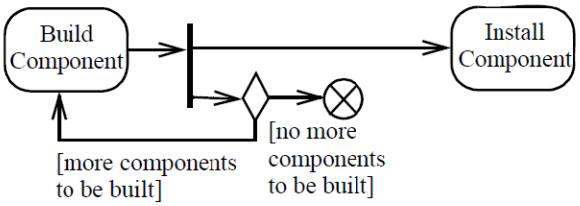
จะเกิดการหยุดสายงานขึ้นหลังจากดำเนินงานแอ็กชันดังกล่าว ดังนั้นความหมายของแผนภาพกิจกรรมตัวอย่างและแผนภาพกิจกรรมที่ได้จากระบบจึงมีความหมายเหมือนกัน

ผลลัพธ์การทวนสอบแผนภาพกิจกรรม แสดงให้เห็นว่าแผนภาพกิจกรรมสอดคล้องตามมาตรฐานยูเอ็มแอล 2.4.1

5.6 ผลการเปรียบเทียบตัวอย่างที่ 5

5.6.1 คำอธิบายแผนภาพกิจกรรม

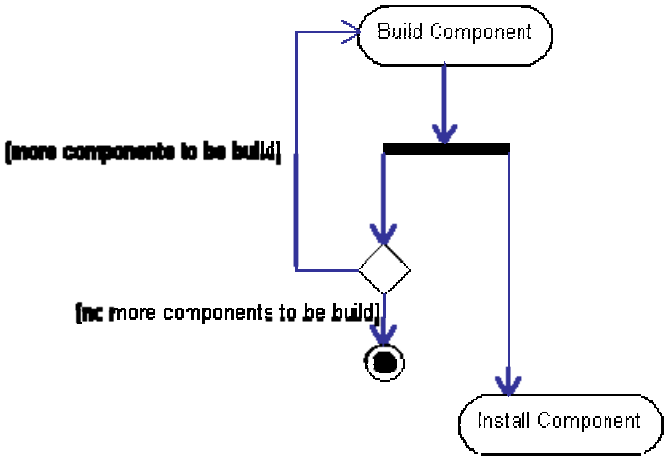
ตัวอย่างที่ 5 เป็นตัวอย่างของกลุ่มตัวอย่างที่หนึ่งซึ่งเป็นข้อมูลแผนภาพกิจกรรมจากเอกสารยูเอ็มแอลของโอเอ็มจี แสดงให้เห็นถึงจุดเริ่มต้นการทำงาน ซึ่งสามารถอนุมานได้จากแอ็กชันที่ไม่เป็นแอ็กชันปลายทางของแอ็กชันใด ๆ (สามารถเป็นแอ็กชันปลายทางของแอ็กชันตนเองได้)



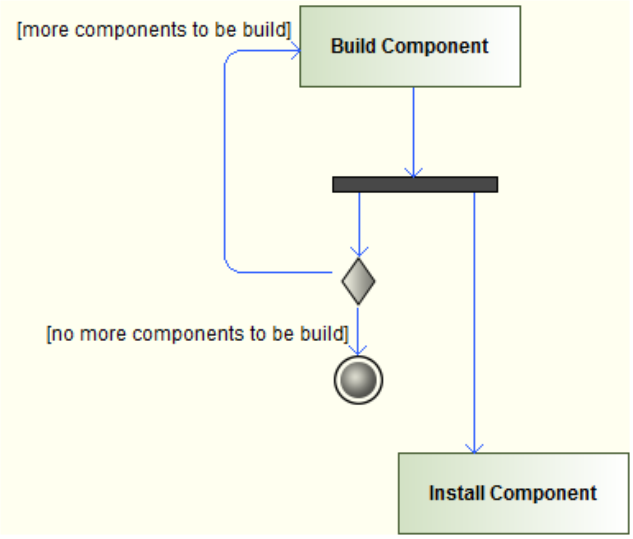
ภาพที่ 68 แผนภาพกิจกรรมตัวอย่างที่ 5 [3]

5.6.2 ข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส

ตัวอย่างที่ 5 สามารถพัฒนาด้วยซอฟต์แวร์โอเพนซอร์ส โดยผลลัพธ์จากซอฟต์แวร์อัลโก้ยูเอ็มแอลแสดงได้ดังภาพที่ 69 และผลลัพธ์จากซอฟต์แวร์โมเดลลิโอแสดงได้ดังภาพที่ 70



ภาพที่ 69 แผนภาพกิจกรรมตัวอย่างที่ 5 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล



ภาพที่ 70 แผนภาพกิจกรรมตัวอย่างที่ 5 จากซอฟต์แวร์โมเดลลิโอ

5.6.3 ข้อมูลอธิบายแผนภาพกิจกรรมด้วยเอดีแอลของงานวิจัย [4]

แผนภาพกิจกรรมตัวอย่างที่ 5 สามารถเขียนให้อยู่ในรูปบทคำสั่งเอดีแอลได้ดังต่อไปนี้

```
diagram 'ex5'
      buildComponent->installComponent
      decision from buildComponent
            if 'no more components to be built' then break
            else
                  if 'more components to be built'
                  then buildComponent
                  endif
            endif
      end
end
```

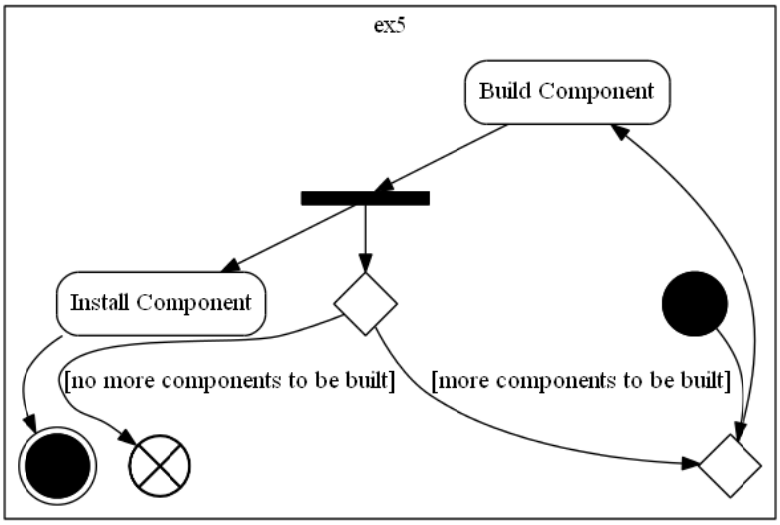
5.6.4 ผลลัพธ์เอดีแอลของระบบ

ผลลัพธ์ที่ได้จากการแปลงแผนภาพกิจกรรมตัวอย่างที่ 5 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังนี้

```
diagram 'ex5'
      action Build Component end
      action Install Component end
      decision from Build Component
            if 'no more components to be built' then break
            else
                  if 'more components to be built' then
                        Build Component
                  endif
            endif
      end
      Build Component->Install Component
end
```

5.6.5 ผลลัพธ์แผนภาพกิจกรรมของระบบ

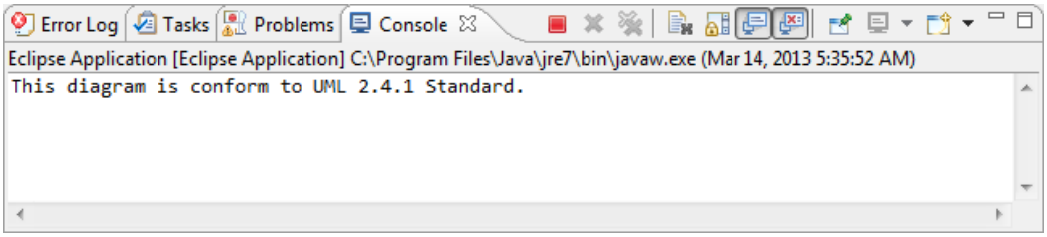
ผลลัพธ์ที่ได้จากการนำแบบจำลองความหมายของเอดีแอลตัวอย่างที่ 5 ไปสร้างแผนภาพกิจกรรมในรูปแบบกราฟวิซเพื่อแปลงเป็นรูปภาพ [4] แสดงได้ดังภาพที่ 71



ภาพที่ 71 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 5

5.6.6 ผลลัพธ์การทวนสอบแผนภาพกิจกรรม

ผลลัพธ์ที่ได้จากการทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 5 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังภาพที่ 72



ภาพที่ 72 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 5

5.6.7 ข้อสังเกต

เอดีแอลที่ได้จากระบบ เมื่อเปรียบเทียบตัวอย่างข้อมูล พบว่ามีการนิยามแอ็กชันเพิ่มเติมขึ้นมาสำหรับแอ็กชันที่ไม่มีการเชื่อมโยงเข้าหรือออกกับบัฟวัตถุ สำหรับลำดับของแอ็กชันและการตัดสินใจเหมือนกันทุกประการ และเมื่อเปรียบเทียบแผนภาพ แสดงให้เห็นว่าแผนภาพกิจกรรมที่ได้จากระบบจะมีการเพิ่มบัพผสานเข้ามาก่อนการทำงานแอ็กชัน “Build Component” ซึ่งเป็นการบ่งบอกว่าเมื่อมีโทเค็นให้สามารถดำเนินการผ่านได้ทันที อย่างไรก็ตาม ความหมายของแผนภาพ

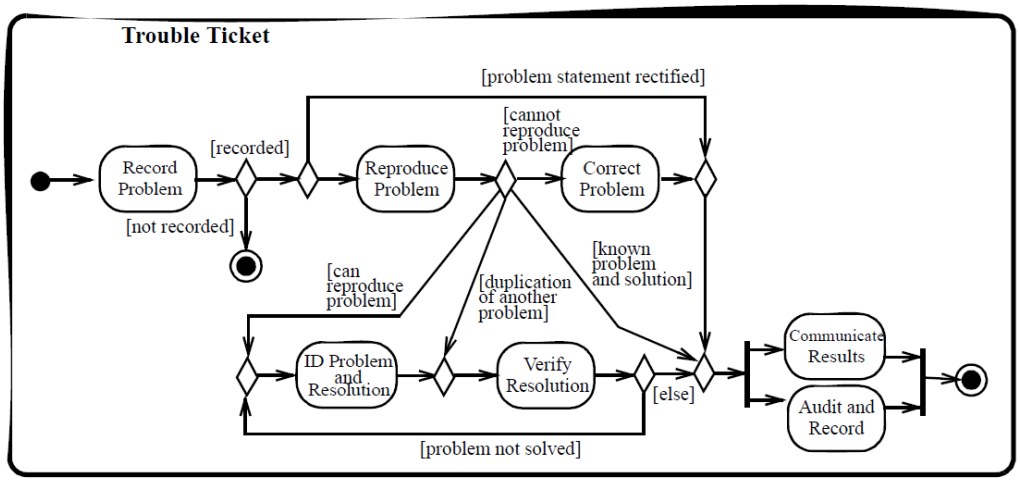
กิจกรรมตัวอย่างและแผนภาพกิจกรรมที่ได้จากระบบยังคงมีความหมายเหมือนกัน

ผลลัพธ์การทวนสอบแผนภาพกิจกรรม แสดงให้เห็นว่าแผนภาพกิจกรรมสอดคล้องตามมาตรฐานยูเอ็มแอล 2.4.1

5.7 ผลการเปรียบเทียบตัวอย่างที่ 6

5.7.1 คำอธิบายแผนภาพกิจกรรม

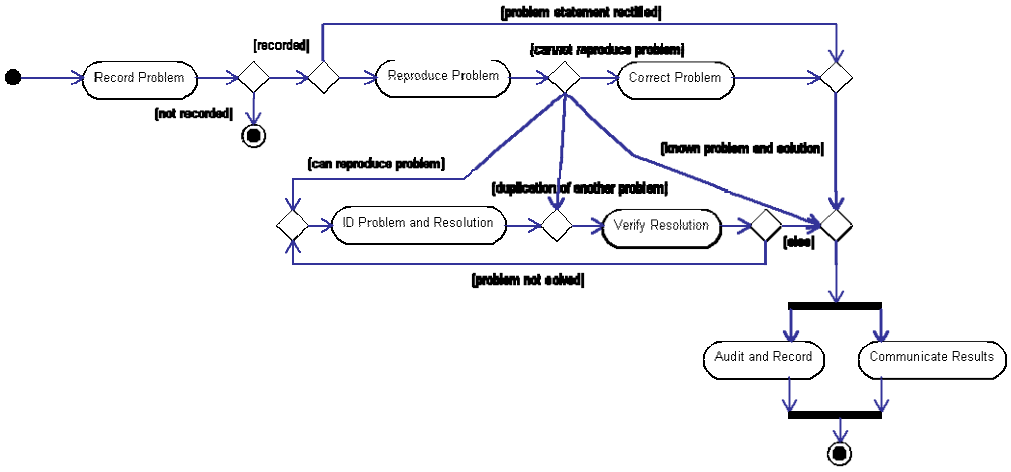
ตัวอย่างที่ 6 เป็นตัวอย่างของกลุ่มตัวอย่างที่หนึ่งซึ่งเป็นข้อมูลแผนภาพกิจกรรมจากเอกสารยูเอ็มแอลของโอเอ็มจี แสดงให้เห็นถึงกิจกรรมที่มีการระบุเงื่อนไขการตัดสินใจเป็นลำดับชั้น ซึ่งจากตัวอย่างภาพที่ 73 แสดงให้เห็นถึงความยากในการออกแบบแผนภาพกิจกรรมให้มีความถูกต้องตามต้องการ



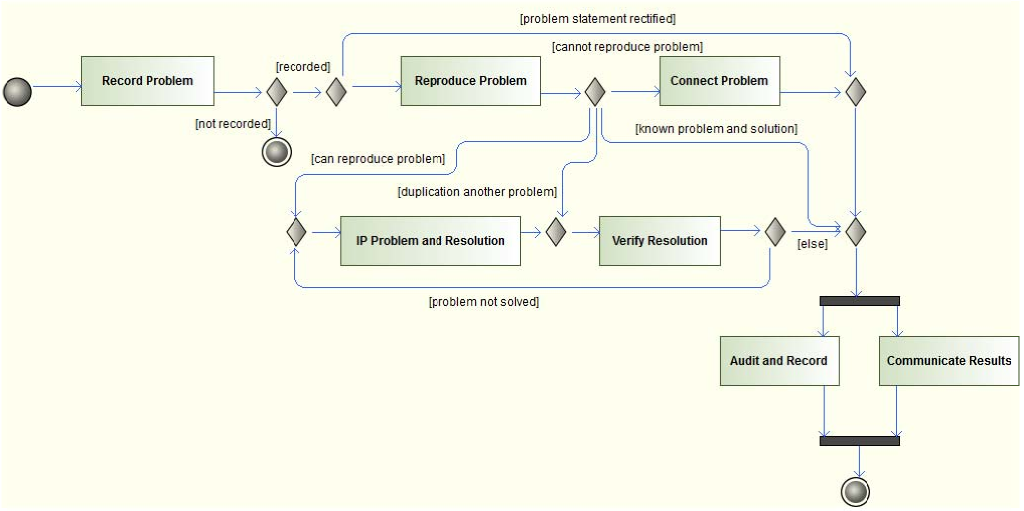
ภาพที่ 73 แผนภาพกิจกรรมตัวอย่างที่ 6 [3]

5.7.2 ข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส

ตัวอย่างที่ 6 สามารถพัฒนาด้วยซอฟต์แวร์โอเพนซอร์ส โดยผลลัพธ์จากซอฟต์แวร์อัลโก้ยูเอ็มแอลแสดงได้ดังภาพที่ 74 และผลลัพธ์จากซอฟต์แวร์โมเดลลิโอแสดงได้ดังภาพที่ 75



ภาพที่ 74 แผนภาพกิจกรรมตัวอย่างที่ 6 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล



ภาพที่ 75 แผนภาพกิจกรรมตัวอย่างที่ 6 จากซอฟต์แวร์โมเดลลิโอ

5.7.3 ข้อมูลอธิบายแผนภาพกิจกรรมด้วยเอดีแอลของงานวิจัย [4]

แผนภาพกิจกรรมตัวอย่างที่ 6 สามารถเขียนให้อยู่ในรูปบทคำสั่งเอดีแอลได้ดังต่อไปนี้

```
diagram 'ticket trouble scenario'
      action recordProblem end
      action reproduceProblem end
      action correctProblem end
      action idProblemAndResolution end
      action verifyResolution end
      action auditAndRecord end
      action communicateResult end

      decision from recordProblem
            if 'recorded' then
            reproduceProblem
```

```
                  if 'problem statement rectified'
                  then auditAndRecord and communicateResult
                  endif
            else
                  if 'not recorded' then terminate endif
            endif
      end

      decision from reproduceProblem
            if 'cannot reproduce problem' then correctProblem
            else
                  if 'can reproduce problem'
                  then idProblemAndResolution
                  endif
            else
                  if 'duplication of another problem'
                  then verifyResolution
                  endif
            else
                  if 'known problem and solution'
                  then auditAndRecord and communicateResult
                  endif
            endif
      end

      idProblemAndResolution -> verifyResolution

      correctProblem -> auditAndRecord
      correctProblem -> communicateResult

      decision from verifyResolution
            if 'problem not solved' then idProblemAndResolution
            else
                  auditAndRecord, communicateResult
            endif
      end
end
```

5.7.4 ผลลัพธ์เอดีแอลของระบบ

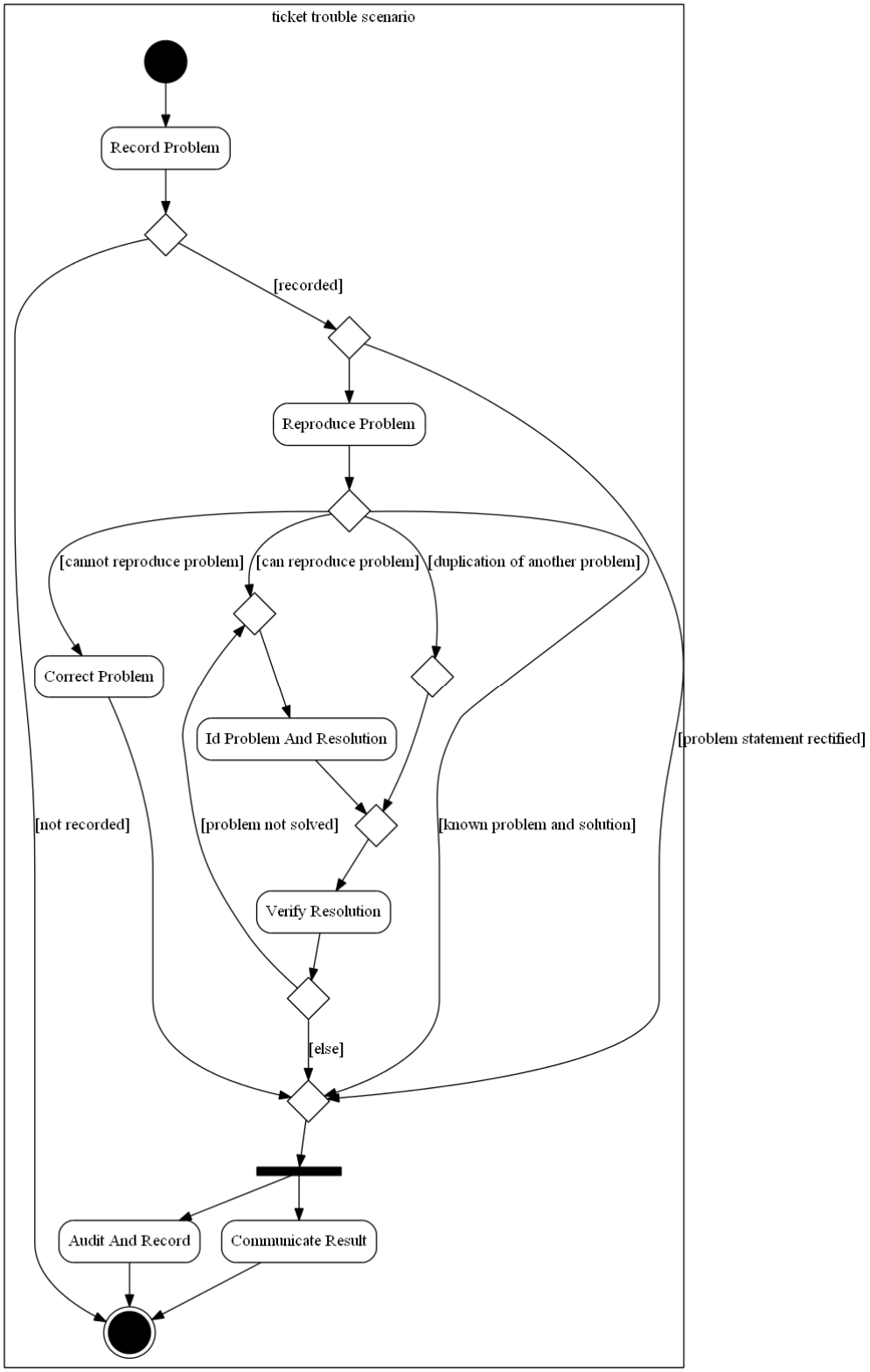
ผลลัพธ์ที่ได้จากการแปลงแผนภาพกิจกรรมตัวอย่างที่ 6 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังนี้

```
diagram 'ticket trouble scenario'
      action Record Problem end
      action Reproduce Problem end
      action Correct Problem end
      action Id Problem And Resolution end
      action Verify Resolution end
      action Audit And Record end
      action Communicate Result end
```

```
        decision from Record Problem
              if 'recorded' then
              if 'problem statement rectified' then Audit And
                    Record and Communicate Result
              else
                    if 'else' then Reproduce Problem
                    endif
              endif
              else
                    if 'not recorded' then break
                    endif
              endif
        end

        decision from Reproduce Problem
              if 'cannot reproduce problem' then Correct Problem
              else

                    if 'can reproduce problem' then Id Problem
                          And Resolution
                    else
                          if 'duplication of another problem'
                                then Verify Resolution
                          else
                                if 'known problem and solution'
                                then Audit And Record and
                                      Communicate Result
                                endif
                          endif
                    endif
              endif
        end

        decision from Verify Resolution
              if 'problem not solved' then Id Problem And
                    Resolution
              else
                    if 'else' then Audit And Record and
                          Communicate Result
                    endif
              endif
        end
        Id Problem And Resolution->Verify Resolution
        Correct Problem->Audit And Record and Communicate Result
  end
```

5.7.5 ผลลัพธ์แผนภาพกิจกรรมของระบบ

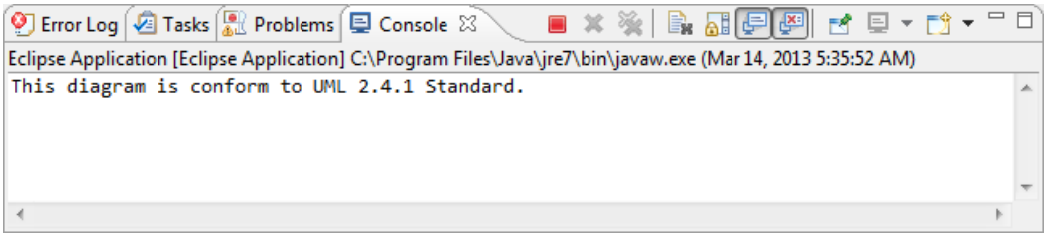
ผลลัพธ์ที่ได้จากการนำแบบจำลองความหมายของเอดีแอลตัวอย่างที่ 6 ไปสร้างแผนภาพกิจกรรมในรูปแบบกราฟวิซเพื่อแปลงเป็นรูปภาพ [4] แสดงได้ดังภาพที่ 76



ภาพที่ 76 แผนภาพกิจกรรมจากผลลัพธ์ของซอฟต์แวร์ตัวอย่างที่ 6

5.7.6 ผลลัพธ์การทวนสอบแผนภาพกิจกรรม

ผลลัพธ์ที่ได้จากการทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 6 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังภาพที่ 77



ภาพที่ 77 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 6

5.7.7 ข้อสังเกต

เอดีแอลที่ได้จากระบบ เมื่อเปรียบเทียบตัวอย่างข้อมูล พบว่ามีการนิยามแอ็กชันของทั้งคู่เหมือนกัน มีจุดที่แตกต่างกัน 2 จุดคือ บัพตัดสินใจที่มีการเชื่อมต่อกับบัพตัดสินใจอีกครั้งหนึ่ง เอดีแอลที่ได้จากระบบไม่สามารถเขียนแบบลดรูปคำสั่งหรือใช้คียเวิร์ด “Else” แทนกรณีที่ผิดเงื่อนไข “If” ได้อีกจุดหนึ่งคือลำดับของแอ็กชันของบัพแยก ระบบจะมองบัพแยกเป็นคำสั่งเดียวไม่แยกผลออกเป็น 2 คำสั่งเหมือนตัวอย่างข้อมูล เหตุผลที่เป็นเช่นนี้ถ้าเปลี่ยนให้เป็น 2 คำสั่งจะขัดกับแผนภาพกิจกรรมตัวอย่างก่อนหน้านี้ และเมื่อเปรียบเทียบแผนภาพ แสดงให้เห็นว่าแผนภาพกิจกรรมที่ได้จากระบบและแผนภาพกิจกรรมตัวอย่างนั้นมีองค์ประกอบของแผนภาพกิจกรรมที่ใช้ในการสร้างเงื่อนไขการตัดสินใจเหมือนกัน

ผลลัพธ์การทวนสอบแผนภาพกิจกรรม แสดงให้เห็นว่าแผนภาพกิจกรรมสอดคล้องตามมาตรฐานยูเอ็มแอล 2.4.1

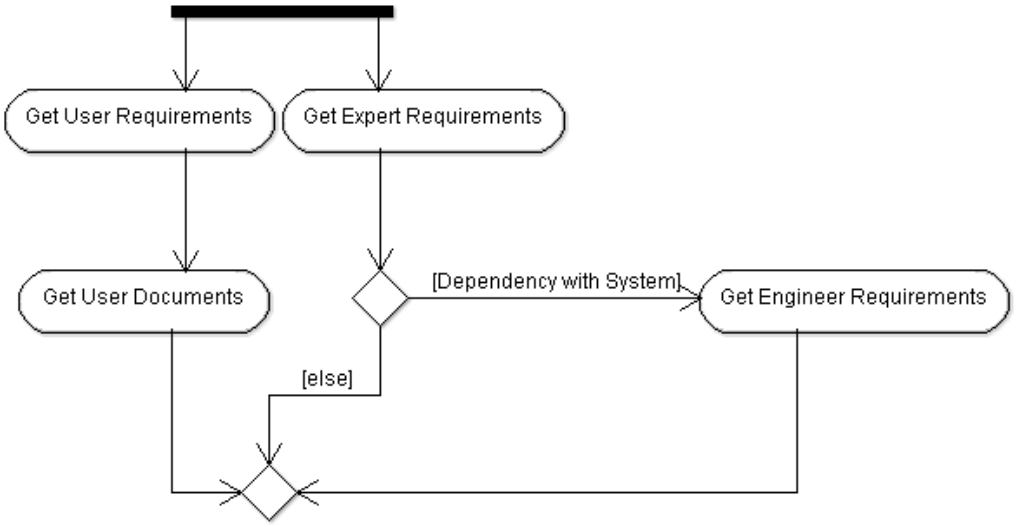
5.8 ผลการเปรียบเทียบตัวอย่างที่ 7

5.8.1 คำอธิบายแผนภาพกิจกรรม

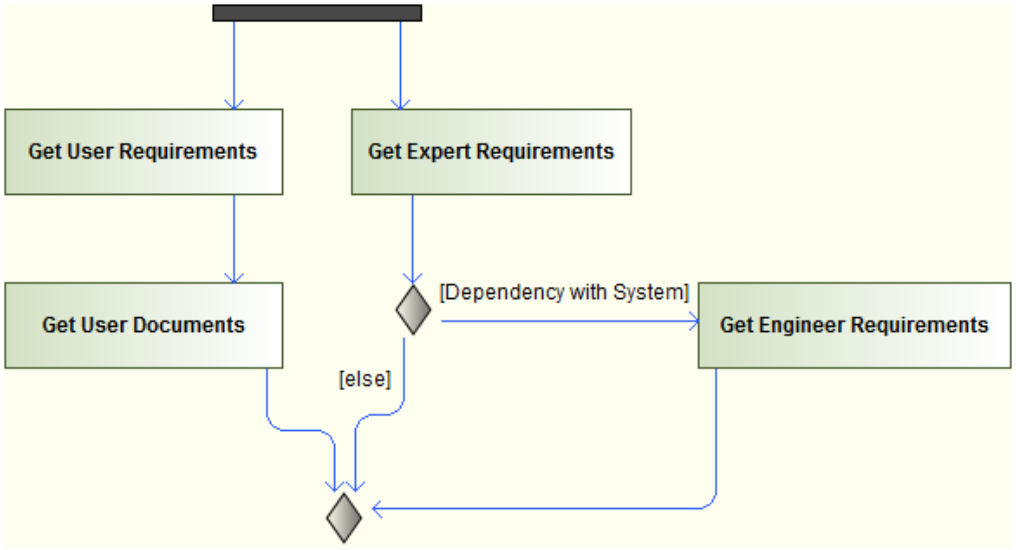
ตัวอย่างที่ 7 เป็นตัวอย่างของกลุ่มตัวอย่างที่สองซึ่งเป็นข้อมูลแผนภาพกิจกรรมที่ผิดตามเงื่อนไขข้อบังคับตามมาตรฐานยูเอ็มแอลของโอเอ็มจี 2.4.1 แสดงให้เห็นจุดผิดที่สำคัญสองอย่าง คือ บัพแยกไม่มีเส้นเชื่อมเข้าและบัพผสานไม่มีเส้นเชื่อมออก

5.8.2 ข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส

ตัวอย่างที่ 7 สามารถพัฒนาด้วยซอฟต์แวร์โอเพนซอร์ส โดยผลลัพธ์จากซอฟต์แวร์อัลโก้ยูเอ็มแอลแสดงได้ดังภาพที่ 78 และผลลัพธ์จากซอฟต์แวร์โมเดลลิโอแสดงได้ดังภาพที่ 79



ภาพที่ 78 แผนภาพกิจกรรมตัวอย่างที่ 7 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล



ภาพที่ 79 แผนภาพกิจกรรมตัวอย่างที่ 7 จากซอฟต์แวร์โมเดลลิโอ

5.8.3 ข้อมูลอธิบายแผนภาพกิจกรรมด้วยเอดีแอลของงานวิจัย [4]

ไม่มี

5.8.4 ผลลัพธ์เอดีแอลของระบบ

ผลลัพธ์ที่ได้จากการแปลงแผนภาพกิจกรรมตัวอย่างที่ 7 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล และโมเดลลิโอแสดงได้ดังนี้

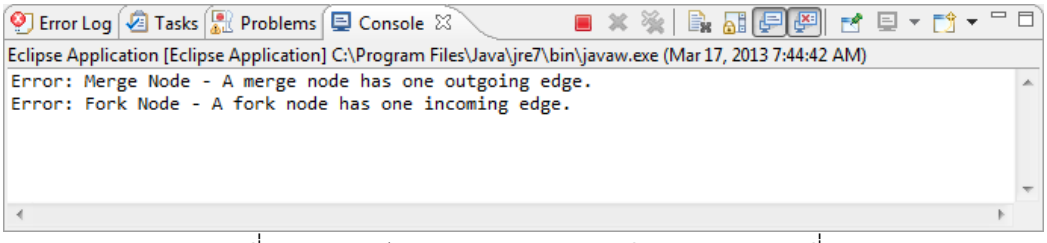
```
diagram 'ex7'
      action Get User Requirements end
      action Get Expert Requirements end
      action Get User Documents end
      action Get Engineer Requirements end
      Fork Node->Get User Requirements and Get Expert
            Requirements
      Get User Requirements->Get User Documents
      Get User Documents->Merge Node
      Get Engineer Requirements-> Merge Node

      decision from Get Expert Requirements
            if 'Dependency with System' then Get Engineer
                  Requirement
            else  Merge Node
            endif
      end
end
```

5.8.5 ผลลัพธ์แผนภาพกิจกรรมของระบบ

ไม่สามารถสร้างแผนภาพกิจกรรมได้

5.8.6 ผลลัพธ์การทวนสอบแผนภาพกิจกรรม

ผลลัพธ์ที่ได้จากการทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 7 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังภาพที่ 80



ภาพที่ 80 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 7

5.8.7 ข้อสังเกต

ผลลัพธ์การทวนสอบแผนภาพกิจกรรม เนื่องจากบัพแยกไม่มีเส้นเชื่อมเข้า และบัพผสานไม่มีเส้นเชื่อมออก ผลการทวนสอบจึงแสดงให้เห็นว่าแผนภาพกิจกรรมนั้นขาดเงื่อนไขบังคับทั้ง 2 ข้อนี้

5.9 ผลการเปรียบเทียบตัวอย่างที่ 8

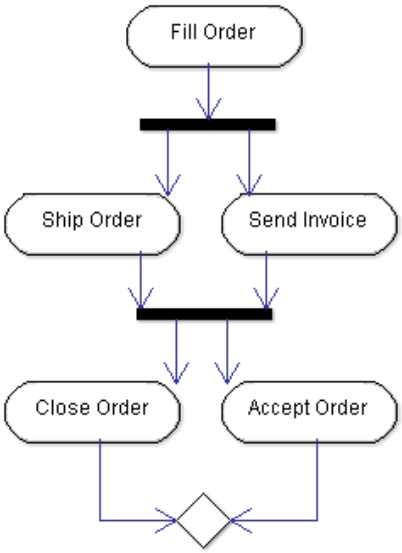
5.9.1 คำอธิบายแผนภาพกิจกรรม

ตัวอย่างที่ 8 เป็นตัวอย่างของกลุ่มตัวอย่างที่สองซึ่งเป็นข้อมูลแผนภาพกิจกรรมที่ผิดตามเงื่อนไขข้อบังคับตามมาตรฐานยูเอ็มแอลของโอเอ็มจี 2.4.1 แสดงให้เห็นจุดผิดที่สำคัญสองอย่าง

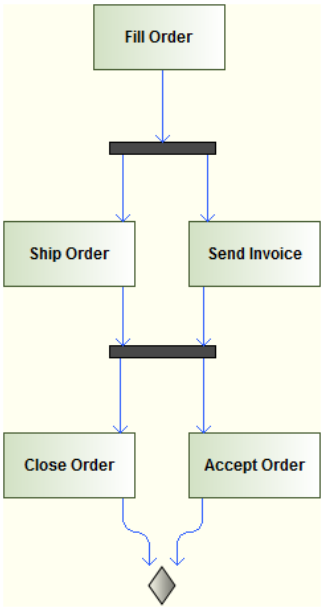
คือบัพรวมมีเส้นเชื่อมออกมากกว่า 1 เส้น และบัพผสานไม่มีเส้นเชื่อมออก

5.9.2 ข้อมูลนำเข้าจากซอฟต์แวร์โอเพนซอร์ส

ตัวอย่างที่ 8 สามารถพัฒนาด้วยซอฟต์แวร์โอเพนซอร์ส โดยผลลัพธ์จากซอฟต์แวร์อัลโก้ยูเอ็มแอลแสดงได้ดังภาพที่ 81 และผลลัพธ์จากซอฟต์แวร์โมเดลลิโอแสดงได้ดังภาพที่ 82



ภาพที่ 81 แผนภาพกิจกรรมตัวอย่างที่ 8 จากซอฟต์แวร์อัลโก้ยูเอ็มแอล



ภาพที่ 82 แผนภาพกิจกรรมตัวอย่างที่ 8 จากซอฟต์แวร์โมเดลลิโอ

5.9.3 ข้อมูลอธิบายแผนภาพกิจกรรมด้วยเอดีแอลของงานวิจัย [4]

ไม่มี

5.9.4 ผลลัพธ์เอดีแอลของระบบ

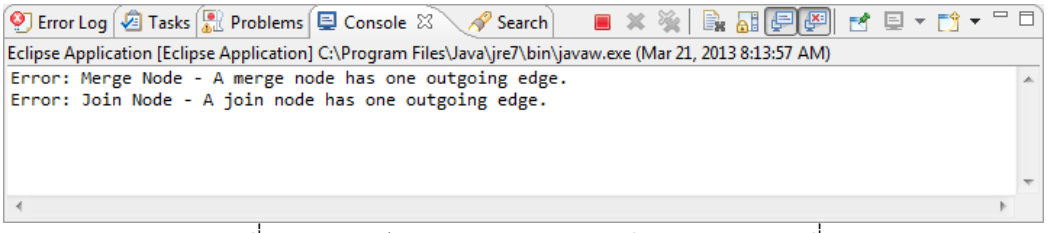
```
diagram 'ex8'
      action Fill Order end
      action Ship Order end
      action Send Invoice end
      action Close Order end
      action Accept Invoice end

      Fill Order->Ship Order and Send Invoice
      Ship Order and Send Invoice->Close Order
      Ship Order and Send Invoice->Accept Order
      Close Order->Merge Node
      Accept Order->Merge Node
end
```

5.9.5 ผลลัพธ์แผนภาพกิจกรรมของระบบ

ไม่สามารถสร้างแผนภาพกิจกรรมได้

5.9.6 ผลลัพธ์การทวนสอบแผนภาพกิจกรรม

ผลลัพธ์ที่ได้จากการทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 8 จากซอฟต์แวร์อัลโก้ยูเอ็มแอลและโมเดลลิโอแสดงได้ดังภาพที่ 83



ภาพที่ 83 ผลลัพธ์การทวนสอบแผนภาพกิจกรรมตัวอย่างที่ 8

5.9.7 ข้อสังเกต

ผลลัพธ์การทวนสอบแผนภาพกิจกรรม เนื่องจากบัพรวมมีเส้นเชื่อมออกมากกว่า 1 เส้น และบัพผสานไม่มีเส้นเชื่อมออก ผลการทวนสอบจึงแสดงให้เห็นว่าแผนภาพกิจกรรมนั้นขาดเงื่อนไขบังคับทั้ง 2 ข้อนี้

บทที่ 6
สรุปผลการวิจัย และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้เสนอการปรับปรุงในการขยายความสามารถของเอดีแอลสำหรับการตรวจทานแผนภาพกิจกรรม การเปิดช่องทางใช้งานร่วมกับซอฟต์แวร์ออกแบบยูเอ็มแอลโอเพนซอร์สอื่นๆ ซึ่งได้แก่ อัลโก้ยูเอ็มแอล และโมเดลลิโอ ผ่านมาตรฐานเอ็กซ์เอ็มไอ ผลลัพธ์ที่ได้คือปลั๊กอินอีคลิปส์ที่ช่วยการทำงานที่กล่าวมา และแบบแผนในการพัฒนาต่อยอดของเอดีแอล

ซึ่งจากการทดลองใช้งานปลั๊กอินอีคลิปส์กับแผนภาพกิจกรรมที่พัฒนาด้วยอัลโก้ยูเอ็มแอล และโมเดลลิโอในบทที่ 5 แสดงให้เห็นว่าปลั๊กอินสามารถแปลงแผนภาพกิจกรรมเป็นเอดีแอลได้อย่างถูกต้อง ถึงแม้ว่าเอกสารเอดีแอลที่ได้นั้นมีรูปแบบไม่เหมือนกับคำสั่งต้นฉบับทั้งหมด แต่จากการทดสอบสร้างแผนภาพกิจกรรมก็ให้ผลลัพธ์ที่เหมือนกับต้นฉบับในทุกตัวอย่าง

สำหรับการตรวจทานกิจกรรมจะเห็นได้ว่า ซอฟต์แวร์สามารถตรวจสอบพบจุดที่แผนภาพกิจกรรมชุดตัวอย่างไม่สอดคล้องตามเงื่อนไขบังคับได้อย่างมีประสิทธิภาพและมีความถูกต้องตามมาตรฐานยูเอ็มแอลของโอเอ็มจี

6.2 ข้อจำกัด

1. การแปลงเอกสารเอ็กซ์เอ็มไอนั้นไม่รองรับความสามารถในการระบุชื่อที่แตกต่างจากข้อมูลชื่อของตัวระบุได้
2. งานวิจัยชิ้นนี้ยังไม่สามารถร้อยแผนภาพกิจกรรมที่มีมากกว่าหนึ่งกิจกรรมได้

6.3 แนวทางการวิจัยต่อ

- เอดีแอลเป็นภาษาใหม่ มีหลายปัญหาที่ต้องปรับปรุงแก้ไข ตัวอย่างเช่น ไวยากรณ์บางอย่างที่สำคัญและขาดหายไปเช่น คอนเม้นต์ ฟังก์ชัน หรือ คลาส หรือเรื่องที่เอดีแอลนั้นสนับสนุนมาตรฐานยูเอ็มแอลและเอ็กซ์เอ็มไอจำเพาะเจาะจงบางเวอร์ชัน ทำให้ยากในการพัฒนาร่วมด้วย ดังนั้นการปรับปรุงเอดีแอลเพื่อให้รองรับกับความสามารถดังกล่าวจะช่วยให้เกิดการนำกลับมาใช้ใหม่ได้ เพิ่มความสะดวกรวดเร็วในการพัฒนาซอฟต์แวร์ และช่วยให้นักออกแบบเกิดความสนใจเพื่อที่จะสร้างการประยุกต์ใหม่ๆ ขึ้นมาได้

- การแปลงเอกสารเอ็กซ์เอ็มไอเพื่อให้เป็นรูปแบบมาตรฐานนั้นยังรองรับซอฟต์แวร์ได้แค่ 2 ซอฟต์แวร์เท่านั้น ยังมีซอฟต์แวร์ในการออกแบบแผนภาพยูเอ็มแอลอีกมาก ดังนั้นการปรับปรุงส่วนแปลงเอกสารเอ็กซ์เอ็มไอจะช่วยให้เกิดความน่าสนใจในเอดีแอลได้เป็นอย่างดี
- การแปลงเอกสารเอ็กซ์เอ็มไอเพื่อให้เป็นรูปแบบมาตรฐานไม่รองรับข้อมูลนำเข้าแบบกระดาษ ทั้งที่มีข้อมูลลักษณะนี้อยู่เป็นจำนวนมาก ดังนั้นการปรับปรุงส่วนการแปลงเอกสารเอ็มซ์เอ็มไอนี้จะช่วยให้เกิดคุณภาพในการวัดผลแผนภาพกิจกรรมที่มี และช่วยในการนำแผนภาพที่มีนั้นกลับมาใช้ใหม่ได้ง่ายในรูปแบบของเอดีแอล
- การพัฒนาเอดีแอลยังไม่มีเครื่องมือที่สามารถช่วยอำนวยความสะดวกในการพัฒนาได้อย่างมีประสิทธิภาพ เช่น การทำสีเพื่อแบ่งแยกส่วนของคำสั่งและเนื้อหา การทำออโต้คอมพลีสในการเขียนโปรแกรม การสร้างแทมเพลสในการพัฒนาขึ้น หรือการดูผลลัพธ์โปรแกรมแบบทันทีทันใด ดังนั้นการปรับปรุงสร้างเครื่องมือขึ้นมาจะช่วยให้ลดความผิดพลาด แต่สร้างเอกสารเอดีแอลได้อย่างมีคุณภาพ

รายการอ้างอิง

[1] Object Management Group. Unified Modeling LanguageTM (OMG UML): Superstructure Version 2.3. Object Management Group, Inc., 2010.

[2] Object Management Group. Unified Modeling LanguageTM (OMG UML): Infrastructure Version 2.4.1. Object Management Group, Inc., 2011.

[3] Object Management Group. Unified Modeling LanguageTM (OMG UML): Superstructure Version 2.4.1. Object Management Group, Inc., 2011.

[4] เจริญศักดิ์ นาคงาม. ข้อกำหนดรูปนัยเพื่อทวนสอบแผนภาพกิจกรรมของกระแสกระบวนการ.วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิชาวิศวกรรมซอฟต์แวร์. คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2554.

[5] Jason E. R. ArgoUML [Online]. 2001. Available From: http://argouml.tigris.org/ [2003, Jan 28]

[6] Modeliosoft. Modelio [Online]. 2011. Available From: http://www.modelio.org/ [2003, Jan 28]

[7] Object Management group. XML Meta Data Interchange (XMI): version 2.4.1. Object Management Group, Inc., 2010.

[8] Kotb Y., Gondow K. and Katayama T. Web Mining: Applications and Techniques. Idea Group Inc. Publications, 2005.

[9] Clayberg E. and Rubel D. Eclipse Plug-ins (3rd Edition). Eclipse Series, 2008.

[10] Kim et al. (Eds.). Computer Application for Software Engineering, Disaster Recovery, and Business Continuity. Springer, 2012.

[11] Shen W., Compton K. and Huggins J. A Toolset for Supporting UML Static and Dynamic Model Checking. Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment, pp. 147-152. USA, 2002.

[12] Kotb Y. and Katayama T. Consistency Checking of UML Model Diagrams Using the XML Semantics Approach.Special interest tracks and posters of the 14th international conference on World Wide Web, pp. 902-903. USA, 2005.

ภาคผนวก

ภาคผนวก

การกำหนดรายละเอียดต่าง ๆ ของระบบ

1. ส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน

การกำหนดรายละเอียดไฟล์แมนนิเฟล็กต่างๆ ที่สำคัญของส่วนการแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐานมีดังนี้

ที่หน้า Dependencies เพิ่มปลั๊กอินที่ต้องการและแพ็กเพจนำเข้าเพิ่มเติมดังภาพที่ 28

ปลั๊กอินที่ต้องการ

- org.eclipse.ui
- org.eclipse.core.runtime
- org.eclipse.core.resources
- org.eclipse.core.expressions

แพ็กเพจนำเข้าเพิ่มเติม

- org.eclipse.jdt.core
- org.xml.sax

ที่หน้า Extensions เพิ่มเอ็กซ์เท็นชั่น 2 ชนิดได้แก่ org.eclipse.ui.menus และ org.eclipse.ui.commands ดังภาพที่ 29 โดยแต่ละชนิดมีรายละเอียดดังนี้

org.eclipse.ui.menus คือ เอ็กซ์เท็นชั่นของเมนูภายในอีคลิปส์ ซึ่งจะช่วยให้เราสามารถเพิ่มเมนูต่างๆ ในเมนูของอีคลิปส์ได้ โดยการเพิ่มนั้นสามารถเพิ่มได้ 3 ตำแหน่งคือเมนูด้านบน เมนูไอคอนด้านบน และเมนูที่เกิดขึ้นเมื่อคลิกขวาที่รายการไฟล์ใน Package Explorer ในส่วนนี้เราจะเพิ่มเมนูหนึ่งเข้าไป ชื่อว่า “XMI Standardize Transform” คือเมนูที่เกิดขึ้นเมื่อคลิกที่ไฟล์นามสกุลเอ็กซ์เอ็มไอ เพื่อเลือกให้ทำการแปลงไฟล์เอกสารเอ็กซ์เอ็มไอที่เลือก และมีเมนูย่อยด้วยกัน 2 รายการคือ ” AlgoUML Convert” สำหรับเลือกว่าต้องการแปลงด้วยวิธีการแปลงของอัลโก้ยูเอ็มแอล และ “Modelio Convert” สำหรับเลือกว่าต้องการแปลงด้วยวิธีการแปลงของโมเดลลิโอ

org.eclipse.ui.commands คือ ตัวเชื่อมระหว่างเมนูและไฟล์คลาส ในการพัฒนานั้นมีการแปลง 2 รูปแบบ ดังนั้น ใน org.eclipse.ui.commands จะมีคำสั่ง 2 รายการ และมีไฟล์คลาสในการเชื่อมต่อ 2 ไฟล์ คือ AlgoUMLReaderHandler.java และ ModelioReaderHandler.java

ผลลัพธ์ในการเพิ่มข้อมูลในไฟล์แมนนิเฟล็กนั้น สามารถตรวจสอบทั้งหมดได้ผ่านหน้า plugin.xml โดยผลลัพธ์ที่ได้ของซอฟต์แวร์แสดงได้ดังภาพที่ 84

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
<extensionpoint="org.eclipse.ui.menus">
<menuContributionlocationURI="popup:org.eclipse.ui.popup.any">
<menuicon="icons/transform.png"label="XMI Standardize
            Transform">
<commandcommandId="th.ac.chula.eng.cp.xmiconverter.algo
               uml.convert"icon="icons/algouml.png"label="AlgoUML"
               style="push">
</command>
<command commandId="th.ac.chula.eng.cp.xmiconverter.model
               io.convert"icon="icons/modelio.png"label="Modelio"
               style="push">
</command>
</menu>
</menuContribution>
</extension>
<extensionpoint="org.eclipse.ui.commands">
<commanddefaultHandler="th.ac.chula.eng.cp.xmiconverter.
         handlers.AlgoUMLReaderHandler"id="th.ac.chula.eng.cp.xmi
         converter.algouml.convert"name="AlgoUML Convert">
</command>
<commanddefaultHandler="th.ac.chula.eng.cp.xmiconverter.
         handlers.ModelioReaderHandler"id="th.ac.chula.eng.cp.xmi
         converter.modelio.convert"name="Modelio Convert">
</command>
</extension>
</plugin>
```

ภาพที่ 84 หน้า plugin.xml ของปลั๊กอินส่วนแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐาน

2. การสร้างส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล

การกำหนดรายละเอียดไฟล์แมนนิเฟล็กต่างๆ ที่สำคัญของส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอลมีดังนี้

ที่หน้า Dependencies เพิ่มปลั๊กอินที่ต้องการและแพ็กเพจนำเข้าเพิ่มเติมดังภาพที่ 31

ปลั๊กอินที่ต้องการ

- org.eclipse.ui
- org.eclipse.core.runtime
- org.eclipse.core.resources
- org.eclipse.core.expressions

แพ็กเพจนำเข้าเพิ่มเติม

- org.eclipse.jdt.core
- org.xml.sax

ที่หน้า Extensions เพิ่มเอ็กซ์เท็นชั่น 2 ชนิด ได้แก่ org.eclipse.ui.menus และ org.eclipse.ui.commands ดังภาพที่ 32 โดยเมนูที่เพิ่มเข้าไป ชื่อว่า “ADL Script Generate” คือ เมนูที่เกิดขึ้นเมื่อคลิกที่ไฟล์นามสกุลเอ็กซ์เอ็มไอ เพื่อเลือกให้ทำการแปลงไฟล์เอกสารเอ็กซ์เอ็มไอที่เลือก ในที่นี้เราสร้างเมนูขึ้นมาแต่เลือกเป็นชนิดคอมมานต์แทน เนื่องจากคำสั่งนั้นไม่มีเมนูย่อย และทำงานเมื่อเราเลือกคำสั่งนั้นโดยตรง

org.eclipse.ui.commands คือ ตัวเชื่อมระหว่างเมนูและไฟล์คลาส ในการแปลงนี้จะมีคำสั่ง 1 รายการ สำหรับคำสั่ง “ADL Script Generate” โดยที่มีไฟล์คลาสในการเชื่อมต่อ คือ FileReaderHandler.java

ผลลัพธ์ในการเพิ่มข้อมูลในไฟล์แมนนิเฟล็กนั้น เราสามารถตรวจสอบทั้งหมดได้ผ่านหน้า plugin.xml โดยผลลัพธ์ที่ได้ของซอฟต์แวร์แสดงได้ดังภาพที่ 85

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
<extensionpoint="org.eclipse.ui.menus">
<menuContribution allPopups="false"locationURI="popup:org.
        eclipse.ui.popup.any">
<commandcommandId="th.ac.chula.eng.cp.adlgenerator
           .generate"icon="icons/transform.png"label="ADL Script
           Generate"style="push">
</command>
</menuContribution>
</extension>
<extensionpoint="org.eclipse.ui.commands">
<commanddefaultHandler="th.ac.chula.eng.cp.adlgenerator.
        handlers.FileReaderHandler"id="th.ac.chula.eng.cp.adl
        generator.generate"name="ADL Scripts Generate">
</command>
</extension>
</plugin>
```

ภาพที่ 85 หน้า plugin.xml ของปลั๊กอินส่วนแปลงเอกสารเอ็กซ์เอ็มไอรูปแบบมาตรฐานให้อยู่ในรูปแบบเอดีแอล

3. การสร้างส่วนสร้างแบบจำลองความหมายของเอดีแอลและตรวจทานแผนภาพกิจกรรม

การกำหนดรายละเอียดไฟล์แมนนิเฟล็กต่างๆ ที่สำคัญของส่วนสร้างแบบจำลองความหมายของเอดีแอลและตรวจทานแผนภาพกิจกรรมมีดังนี้

ที่หน้า Dependencies เพิ่มปลั๊กอินที่ต้องการและแพ็กเพจนำเข้าเพิ่มเติมดังภาพที่ 34

ปลั๊กอินที่ต้องการ

- org.eclipse.ui
- org.eclipse.core.runtime
- org.eclipse.core.resources
- org.eclipse.core.expressions

ที่หน้า Extensions เพิ่มเอ็กซ์เท็นชั่น 2 ชนิด ได้แก่ org.eclipse.ui.menus และ org.eclipse.ui.commands ดังภาพที่ 35 โดยที่เมนูที่เพิ่มเข้าไป ชื่อว่า “ADL Validate and

Generate ADL Semantic Model” คือ เมนูที่เกิดขึ้นเมื่อคลิกที่ไฟล์นามสกุลเอดีแอล เพื่อเลือกให้ทำการตรวจทานแผนภาพกิจกรรม และสร้างแบบจำลองความหมายของเอดีแอลขึ้นมา ในที่นี้เราสร้างเมนูขึ้นมาแต่เลือกเป็นชนิดคอมมานต์แทน เนื่องจากคำสั่งนั้นไม่มีเมนูย่อย และทำงานเมื่อเราเลือกคำสั่งนั้นโดยตรง

org.eclipse.ui.commands คือ ตัวเชื่อมระหว่างเมนูและไฟล์คลาส ในการแปลงนี้จะมีคำสั่ง 1 รายการ สำหรับคำสั่ง “ADL Validate and Generate ADL Semantic Model” โดยที่มีไฟล์คลาสในการเชื่อมต่อ คือ FileReaderHandler.java

ผลลัพธ์ในการเพิ่มข้อมูลในไฟล์แมนนิเฟล็กนั้น เราสามารถตรวจสอบทั้งหมดได้ผ่านหน้า plugin.xml โดยผลลัพธ์ที่ได้ของซอฟต์แวร์แสดงได้ดังภาพที่ 86

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
<extensionpoint="org.eclipse.ui.menus">
<menuContributionlocationURI="popup:org.eclipse.ui.popup.any">
<commandcommandId="th.ac.chula.eng.cp.adlvalidation.validate"
              icon="icons/transform.png"
              label="ADL Validate and Generate ADL Semantic Model"
              style="push">
</command>
</menuContribution>
</extension>
<extensionpoint="org.eclipse.ui.commands">
<commanddefaultHandler="th.ac.chula.eng.cp.adlvalidation.validate.
      FileReaderHandler" id="th.ac.chula.eng.cp.adlvalidation.
      validate "name="ADL Validate and Generate ADL Semantic Model">
</command>
</extension>
</plugin>
```

ภาพที่ 86 หน้า plugin.xml ของปลั๊กอินส่วนสร้างแบบจำลองความหมายของเอดีแอลและตรวจทานแผนภาพกิจกรรมมีดังนี้

ประวัติผู้เขียนวิทยานิพนธ์

นายชินพัฒน์ แก้วชินพร เกิดเมื่อวันที่ 13 พฤศจิกายน พ.ศ. 2531 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีเจ้าคุณทหารพระจอมเกล้าลาดกระบัง ในปีการศึกษา 2554 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2554