

ตัวแปลงแผนภาพกิจกรรมสำหรับรูปแบบมาตรฐานเอ็กซ์เอ็มไอล

นายธนกฤต เจริญรัตน์นากร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2556

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.

ACTIVITY DIAGRAM CONVERTER FOR STANDARDIZATION OF XMI FORMATS

Mr. Tanakrit Jaroenratanakorn

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2013

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	ตัวแปลงแผนภาพกิจกรรมสำหรับรูปแบบมาตรฐาน เอ็กซ์เอ็มไอ
โดย	นายธนภุต เจริญรัตนากร
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ญาใจ ลิ้มปิยะภรณ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.ญาใจ ลิ้มปิยะภรณ์)

..... กรรมการภายนอกมหาวิทยาลัย
(อาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต)

ธนภฤต เจริญรัตนากร: ตัวแปลงแผนภาพกิจกรรมสำหรับรูปแบบมาตรฐานเอ็กซ์เอ็มไอ. (ACTIVITY DIAGRAM CONVERTER FOR STANDARDIZATION OF XMI FORMATS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.ญาใจ ลิ้มปิยะภรณ์, 66 หน้า.

แผนภาพกิจกรรมยูเอ็มแอลถูกนำมาใช้อย่างแพร่หลายเพื่ออธิบายกระบวนการทำงานด้วยสัญลักษณ์ ซึ่งใช้แทนกิจกรรมและแสดงความสัมพันธ์ระหว่างกิจกรรมให้สามารถทำความเข้าใจได้ง่าย ปัจจุบัน มีเครื่องมือสำหรับช่วยการสร้างแผนภาพกิจกรรมอยู่มากมาย อย่างไรก็ตามรูปแบบของไฟล์เอ็กซ์เอ็มไอที่ได้จากเครื่องมือยูเอ็มแอลที่ต่างกันนั้น จะมีคุณลักษณะที่แตกต่างกันขึ้นอยู่กับเครื่องมือที่เลือกใช้ ส่งผลให้เกิดข้อจำกัดในการแลกเปลี่ยนข้อมูลระหว่างแอปพลิเคชันดังกล่าว งานวิจัยนี้จึงได้นำเสนอแนวทางและพัฒนาตัวแปลงรูปแบบเอ็กซ์เอ็มไอของแผนภาพกิจกรรมที่แตกต่างกันให้อยู่ในรูปแบบเดียวกันที่กำหนดโดยนิยามเค้าร่างเอ็กซ์เอ็มแอลหรือเอ็กซ์เอสดี ซึ่งเฉพาะเจาะจงสำหรับแอปพลิเคชันหนึ่งๆ ตัวแปลงรูปแบบเอ็กซ์เอ็มไอสำหรับแผนภาพกิจกรรมยูเอ็มแอลที่พัฒนาขึ้นในงานวิจัยนี้ สามารถใช้แปลงไฟล์เอ็กซ์เอ็มไอของแผนภาพกิจกรรมที่สร้างขึ้นจากเครื่องมือที่ต่างกัน 5 ชนิด ให้เป็นรูปแบบเอ็กซ์เอ็มไอเดียวกัน เพื่อนำเข้าระบบตรวจสอบความถูกต้องของแผนภาพกิจกรรมตามมาตรฐานที่กำหนดโดยโอเอ็มจี แนวทางที่นำเสนอสามารถประยุกต์ใช้กับแอปพลิเคชันอื่นๆให้สามารถรองรับการนำเข้าข้อมูลในรูปแบบเอ็กซ์เอ็มไอที่แตกต่างกันได้

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก

ปีการศึกษา 2556

5471055021 : MAJOR COMPUTER SCIENCE

KEYWORDS : UML ACTIVITY DIAGRAM / XMI / STANDARDIZATION / DATA EXCHANGE

TANAKRIT JAROENRATANAKORN: ACTIVITY DIAGRAM CONVERTER FOR STANDARDIZATION OF XMI FORMATS: ASSOC. PROF. YACHAI LIMPIYAKORN, Ph.D. 66 pp.

UML Activity diagrams are widely used for describing processes with notations, which represent activities and their relationships, so as for easy understanding. Nowadays, there are many tools to facilitate the construction of Activity diagrams. However, the formats of XMI files obtained from different UML tools will differ from each other, resulting in the restriction of data exchange among applications. Therefore, this research presents an approach and develops a converter to transform various XMI formats of Activity diagrams to the standard format established by the XML Schema Definition (XSD), particular to a certain application. The XMI converter for UML Activity diagrams developed in this work can be used for transforming various formats of XMI files created from five different tools to the standardized format, as an input into a system for verifying the conformance of Activity diagrams to OMG standard. The presented approach can be applied to other applications that require the capability of working with input in multiple XMI formats.

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

Field of Study : Computer Science..... Advisor's Signature

Academic Year : 2013.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีจากความช่วยเหลือและสนับสนุนของบุคคลหลายท่าน ประกอบด้วย รองศาสตราจารย์ ดร.ญาใจ ลิ้มปิยะภรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งเป็นผู้เสียสละเวลาในการแนะนำแนวทางพัฒนา ซึ่งให้เห็นถึงปัญหา และคอยแก้ไขตรวจสอบความเรียบร้อยของงานมาโดยตลอด และคณะกรรมการสอบวิทยานิพนธ์ ประกอบด้วย ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล และอาจารย์ ดร.ภาสกร อภิรักษ์วรพิณิต ซึ่งเป็นผู้ให้คำแนะนำและชี้จุดบกพร่องที่ควรแก้ไข ข้าพเจ้าจึงขอกราบขอบพระคุณเป็นอย่างยิ่งในความกรุณาของท่านไว้ ณ ที่นี้

ท้ายที่สุด ข้าพเจ้าขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ภรรยาและลูกสำหรับกำลังใจที่มีค่ายิ่ง รวมถึงขอขอบพระคุณนิสิตรุ่นพี่สาขาวิศวกรรมซอฟต์แวร์ และมิตรสหายที่ให้กำลังใจ ให้การสนับสนุนและความช่วยเหลือในด้านต่างๆ และท่านอื่นๆ ที่มีได้กล่าวชื่อไว้ 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.....	1
1.3 ขอบเขตของการวิจัย.....	1
1.4 ข้อตกลงเบื้องต้น.....	2
1.5 ข้อจำกัดของการวิจัย.....	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.7 วิธีดำเนินงานวิจัย.....	2
1.8 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์.....	3
1.9 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ทฤษฎีที่เกี่ยวข้อง.....	4
2.2 งานวิจัยที่เกี่ยวข้อง.....	13
บทที่ 3 วิธีดำเนินการวิจัย.....	18
3.1 แนวคิดวิธีดำเนินการวิจัย.....	18
3.2 การตรวจสอบความถูกต้องของไฟล์อิเล็กทรอนิกส์.....	19
3.3 การแจ้งส่วนอิเล็กทรอนิกส์.....	20
3.4 การสร้างรูปแบบมาตรฐานที่ผู้ใช้งานกำหนด.....	27
บทที่ 4 การออกแบบและพัฒนาระบบ.....	31
4.1 สถาปัตยกรรมระบบ.....	31

หน้า

4.2 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา	31
4.3 การพัฒนาระบบ	32
บทที่ 5 การประเมินและการวัดผล.....	42
5.1 แนวทางการประเมินผลงานวิจัย	42
5.2 การทดสอบตัวอย่างที่ 1.....	43
5.3 การทดสอบตัวอย่างที่ 2.....	46
5.4 การทดสอบตัวอย่างที่ 3.....	49
5.5 การทดสอบตัวอย่างที่ 4.....	52
5.6 การทดสอบตัวอย่างที่ 5.....	55
บทที่ 6 สรุปผลการวิจัย และข้อเสนอแนะ.....	59
6.1 สรุปผลการวิจัย.....	59
6.2 ข้อจำกัด.....	59
6.3 แนวทางการวิจัยต่อ	59
รายการอ้างอิง.....	60
ภาคผนวก.....	61
ประวัติผู้เขียนวิทยานิพนธ์	66

สารบัญตาราง

หน้า

ตารางที่ 1 ประเภทและสัญลักษณ์ที่ใช้ในแผนภาพกิจกรรม [4].....	6
ตารางที่ 2 ผลการแลกเปลี่ยนข้อมูลอิเล็กทรอนิกส์ไอแผนภาพคลาสรระหว่างเครื่องมือ ที่เลือกใช้ในงานวิจัย [7].....	15
ตารางที่ 3 ผลการแลกเปลี่ยนข้อมูลอิเล็กทรอนิกส์ไอระหว่างเครื่องมือสร้างแผนภาพ กิจกรรมยูเอ็มแอลที่ใช้ในการวิจัย.....	42

สารบัญภาพ

หน้า

ภาพที่ 1 ตัวอย่างแผนภาพกิจกรรมแบบ Intermediate.....	6
ภาพที่ 2 ตัวอย่างลักษณะโครงสร้างลำดับชั้นของเอกสาร XML.....	9
ภาพที่ 3 ตัวอย่าง XML Schema ชื่อ "shiporder.xsd"	10
ภาพที่ 4 ตัวอย่าง XML DOM	12
ภาพที่ 5 ตัวอย่าง XML DOM Node Tree	13
ภาพที่ 6 ระบบทบทวนแผนภาพกิจกรรม	14
ภาพที่ 7 รูปแบบการสูญหายจากการแลกเปลี่ยนข้อมูลเอ็กซ์เอ็มไอระหว่างต่างเครื่องมือ [7]	15
ภาพที่ 8 ภาพรวมการทำงานของระบบที่ได้จากงานวิจัย [4].....	16
ภาพที่ 9 ภาพรวมระบบตัวแปลงรูปแบบมาตรฐานเอ็กซ์เอ็มไอสำหรับแผนภาพกิจกรรม.....	19
ภาพที่ 10 ตัวอย่างโน้ตกราฟที่มีแอมพลิฟายเออร์สำหรับเก็บเอ็กซ์เอ็มไอเวอร์ชัน.....	20
ภาพที่ 11 ตัวอย่าง UML:Model.....	21
ภาพที่ 12 ตัวอย่าง UML:CompositeState.subvertex.....	22
ภาพที่ 13 ตัวอย่าง UML:StateMachine.transitions.....	22
ภาพที่ 14 ตัวอย่าง UML:StateMachine.transitions.....	25
ภาพที่ 15 ตัวอย่างแท็ก node ในเอ็กซ์เอ็มไอไฟล์จากเครื่องมือทดสอบ.....	26
ภาพที่ 16 ตัวอย่างแท็ก edge ในเอ็กซ์เอ็มไอไฟล์จากเครื่องมือทดสอบ.....	26
ภาพที่ 17 ตัวอย่างเอ็กซ์เอ็มแอลสคีม่าที่ใช้กำหนดรูปแบบการนำออกไฟล์เอ็กซ์เอ็มไอ.....	28
ภาพที่ 18 ตัวอย่างคลาสที่มีความสัมพันธ์ตามรูปแบบของเอ็กซ์เอ็มแอลสคีม่าที่กำหนดใน ชื่อ 1	29
ภาพที่ 19 ตัวอย่างผลลัพธ์ไฟล์เอ็กซ์เอ็มไอ จากเครื่องมือตัวแปลงแผนภาพกิจกรรม.....	30
ภาพที่ 20 การสร้างปลั๊กอิน-โปรเจกต์.....	33
ภาพที่ 21 กำหนดรายละเอียดและการตั้งค่าสำหรับโปรเจกต์ใหม่.....	34
ภาพที่ 22 กำหนดรายละเอียดและการตั้ง ค่าสำหรับโปรเจกต์ใหม่.....	35
ภาพที่ 23 การกำหนดรายละเอียดเทมเพลต.....	36
ภาพที่ 24 กำหนดรายละเอียดหน้า Dependencies ของปลั๊กอิน.....	37

ภาพที่ 25 กำหนดรายละเอียด Extensions สำหรับปลั๊กอิน..... 37

ภาพที่ 26 ภาพการทำงานการตรวจสอบความถูกต้องของไฟล์เอ็กซ์เอ็มไอ..... 38

ภาพที่ 27 การสร้างคลาสเพื่อใช้ในการแจ้งส่วนเอ็กซ์เอ็มไอ..... 39

ภาพที่ 28 ภาพการทำงานการแจ้งส่วนเอ็กซ์เอ็มไอ..... 40

ภาพที่ 29 ภาพการทำงานการสร้างรูปแบบมาตรฐานที่ผู้ใช้งานกำหนด..... 41

ภาพที่ 30 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ..... 43

ภาพที่ 31 เอ็กซ์เอ็มไอไฟล์ที่ได้จากเครื่องมือทดสอบ..... 44

ภาพที่ 32 ผลลัพธ์เอ็กซ์เอ็มไอของตัวแปลงแผนภาพกิจกรรม..... 45

ภาพที่ 33 ผลลัพธ์จากเครื่องมือทวนสอบแผนภาพกิจกรรมในงานวิจัย[2] 46

ภาพที่ 34 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ..... 46

ภาพที่ 35 เอ็กซ์เอ็มไอไฟล์ที่ได้จากเครื่องมือทดสอบ..... 47

ภาพที่ 36 ผลลัพธ์เอ็กซ์เอ็มไอของตัวแปลงแผนภาพกิจกรรม 48

ภาพที่ 37 ผลลัพธ์จากเครื่องมือทวนสอบแผนภาพกิจกรรมในงานวิจัย [2] 49

ภาพที่ 38 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ..... 49

ภาพที่ 39 เอ็กซ์เอ็มไอไฟล์ที่ได้จากเครื่องมือทดสอบ..... 50

ภาพที่ 40 ผลลัพธ์เอ็กซ์เอ็มไอของตัวแปลงแผนภาพกิจกรรม 51

ภาพที่ 41 ผลลัพธ์จากเครื่องมือทวนสอบแผนภาพกิจกรรมในงานวิจัย [2] 52

ภาพที่ 42 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ..... 52

ภาพที่ 43 เอ็กซ์เอ็มไอไฟล์ที่ได้จากเครื่องมือทดสอบ..... 53

ภาพที่ 44 ผลลัพธ์เอ็กซ์เอ็มไอของตัวแปลงแผนภาพกิจกรรม 54

ภาพที่ 45 ผลลัพธ์จากเครื่องมือทวนสอบแผนภาพกิจกรรมในงานวิจัย [2] 55

ภาพที่ 46 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ..... 55

ภาพที่ 47 เอ็กซ์เอ็มไอไฟล์ที่ได้จากเครื่องมือทดสอบ..... 56

ภาพที่ 48 ผลลัพธ์เอ็กซ์เอ็มไอของตัวแปลงแผนภาพกิจกรรม 57

ภาพที่ 49 ผลลัพธ์จากเครื่องมือทวนสอบแผนภาพกิจกรรมในงานวิจัย [2] 58

ภาพที่ 50 ภาพแสดงตัวอย่าง plugin.xml ในงานวิจัย..... 63

ภาพที่ 51 ภาพไฟล์เอ็กซ์เอ็มแอลก่อนนำเข้ากระบวนการสร้างไฟล์เอ็กซ์เอ็มไอ.....64

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

แผนภาพยูเอ็มแอล (UML Diagrams) เป็นเอกสารการออกแบบเชิงวัตถุที่มีการใช้งานอย่างหลากหลายในหลายโดเมน แผนภาพกิจกรรม (Activity Diagram) เป็นชนิดหนึ่งของแผนภาพยูเอ็มแอล ถูกพัฒนาขึ้นเพื่อใช้จำลองแบบพฤติกรรมหรือกระบวนการทำงานด้วยสัญกรณ์ (Notation) ที่กำหนดเป็นมาตรฐานโดยโอเอ็มจี (OMG: Object Management Group) [1]

แผนภาพกิจกรรมนิยมนำมาใช้เพื่ออธิบายลำดับขั้นตอนของการทำงานด้วยสัญกรณ์ ซึ่งใช้แทนกิจกรรมและความสัมพันธ์ระหว่างกิจกรรมที่ซับซ้อนให้สามารถทำความเข้าใจได้ง่าย ปัจจุบัน มีเครื่องมือสำหรับช่วยออกแบบแผนภาพกิจกรรมมากมาย อย่างไรก็ตาม รูปแบบของไฟล์เอกซ์เอ็มไอ (XMI: XML Metadata Interchange) ที่ได้จากเครื่องมือที่ต่างกันนั้น จะมีคุณลักษณะที่แตกต่างกันขึ้นอยู่กับเครื่องมือที่เลือกใช้

เอกซ์เอ็มไอเป็นมาตรฐานที่กำหนดขึ้นโดยโอเอ็มจี เพื่อใช้ในการแลกเปลี่ยนข้อมูลบนพื้นฐานของเอกสารเอกซ์เอ็มแอล โดยการใช้งานจะต้องมีการกำหนดชื่อของแท็ก (Tag) และแอทริบิวต์ (Attribute) ของแท็กขึ้นมาให้เป็นมาตรฐานกลาง ทุกแผนภาพยูเอ็มแอลสามารถแปลงเป็นไฟล์เอกซ์เอ็มไอได้ ซึ่งมีข้อดีคือ ทำให้สามารถแลกเปลี่ยนข้อมูลระหว่างแผนภาพหรือแอปพลิเคชันได้

งานวิจัยนี้จึงได้ศึกษาข้อแตกต่างของลักษณะเฉพาะของไฟล์เอกซ์เอ็มไอที่ได้จากเครื่องมือที่ต่างกัน เพื่อแปลงผลลัพธ์ที่ได้ให้อยู่ในรูปแบบเดียวกัน โดยรูปแบบผลลัพธ์ถูกกำหนดขึ้นเป็นมาตรฐานเดียวกันตามข้อกำหนดของแต่ละองค์กรหรือการใช้งานของแต่ละแอปพลิเคชัน

1.2 วัตถุประสงค์ของการวิจัย

1. นำเสนอวิธีการและพัฒนาตัวแปลงเอกสารเอกซ์เอ็มไอเพื่อเพิ่มความยืดหยุ่นในการแลกเปลี่ยนข้อมูลแผนภาพกิจกรรมระหว่างเครื่องมือต่าง ๆ ให้เหมาะสมในการนำไปใช้งานกับระบบงานธุรกิจ

1.3 ขอบเขตของการวิจัย

1. งานวิจัยรองรับการตรวจทานแผนภาพกิจกรรมตามมาตรฐาน UML version 2.3
2. สามารถรองรับเวอร์ชันใหม่ของเอกสารเอกซ์เอ็มไอ โดยการระบุแม่แบบเอกสารได้

3. การประเมินงานวิจัย จะทดสอบแปลงเอกสารจากแผนภาพกิจกรรมยูเอ็มแอลที่มีรูปแบบเอ็กซ์เอ็มไอแตกต่างกันไม่น้อยกว่า 5 รูปแบบจากเครื่องมือที่ต่างกัน โดยผลลัพธ์ไฟล์เอ็กซ์เอ็มไอที่ได้จากการแปลงจะต้องนำเข้าระบบที่พัฒนาขึ้นในงานวิจัย [2] ได้
4. งานวิจัยรองรับการแปลงแผนภาพกิจกรรมสำหรับข้อมูลที่ขาดหายไประหว่างการแลกเปลี่ยนข้อมูล

1.4 ข้อตกลงเบื้องต้น

1. ไฟล์แผนภาพกิจกรรมที่นำเข้าระบบจะต้องอยู่ในรูปแบบเอ็กซ์เอ็มไอ
2. ไฟล์นำเข้าเอ็กซ์เอ็มไอจะต้องมีโครงสร้างตามมาตรฐานที่โอเอ็มจีกำหนด
3. มาตรฐานรูปแบบผลลัพธ์จะต้องอยู่ในรูปแบบของเอ็กซ์เอ็มแอลสคีม่า นามสกุลจะต้องเป็น .xsd เท่านั้น

1.5 ข้อกำหนดของการวิจัย

1. ไฟล์เอ็กซ์เอ็มไอตามรูปแบบที่ผู้ใช้กำหนดเป็นมาตรฐาน
2. รูปแบบมาตรฐานเอ็กซ์เอ็มไอจะมีความเฉพาะเจาะจงและแตกต่างกันขึ้นอยู่กับ XSD ที่ผู้ใช้กำหนดขึ้น
3. งานวิจัยรองรับรูปแบบมาตรฐานที่กำหนดขึ้นเพื่อใช้ในงานวิจัย [2]

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เครื่องมือในการแปลงแผนภาพกิจกรรมยูเอ็มแอลให้อยู่ในรูปแบบเอ็กซ์เอ็มไอเดียวกัน
2. ได้เครื่องมือเพื่อใช้ในการปรับรุ่นของเอกสารเอ็กซ์เอ็มไอเพื่อนำไปใช้กับเครื่องมืออื่น ๆ

1.7 วิธีดำเนินงานวิจัย

1. ศึกษาข้อมูลมาตรฐานเอ็กซ์เอ็มไอที่มีประกาศใช้อยู่ตั้งแต่เริ่มต้นจนถึงปัจจุบัน
2. ศึกษาแนวทางการแปลงค่าเอ็กซ์เอ็มไอไฟล์ที่ได้ให้อยู่ในมาตรฐานเดียวกัน
3. ศึกษาวิธีการทบทวนแผนภาพกิจกรรมด้วยภาษาอธิบายการกระทำหรือเอดีแอลสำหรับแผนภาพกิจกรรมที่มีอยู่แล้ว
4. สร้างซอฟต์แวร์สำหรับการแปลงค่าเอ็กซ์เอ็มไอไฟล์ที่ได้ให้อยู่ในรูปแบบทั่วไปของเอกสาร

เอ็กซ์เอ็มไอที่กำหนด

5. นำค่าที่ได้ไปเข้าระบบทวนสอบแผนภาพกิจกรรมเพื่อแสดงแผนภาพที่เกิดขึ้นจากการทวนสอบ
6. เปรียบเทียบแผนภาพตั้งต้นกับแผนภาพที่ได้จากระบบทวนสอบแผนภาพกิจกรรม
7. ประเมินและสรุปผล
8. ตีพิมพ์ผลงานวิชาการ
9. จัดทำวิทยานิพนธ์

1.8 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บท ดังนี้ บทที่ 1 บทนำ กล่าวถึงที่มาและความสำคัญของปัญหา วัตถุประสงค์ของการวิจัย ขอบเขตของงานวิจัย ประโยชน์ที่คาดว่าจะได้รับและผลงานตีพิมพ์ บทที่ 2 กล่าวถึง ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับงานวิจัย บทที่ 3 กล่าวถึงวิธีดำเนินการวิจัย บทที่ 4 กล่าวถึง การออกแบบและพัฒนาระบบตามแนวทางการวิจัย บทที่ 5 กล่าวถึงวิธีการประเมินและวัดผลการทดลอง และบทที่ 6 สรุปผลการวิจัย ข้อเสนอแนะ และแนวทางสำหรับการวิจัยต่อ

1.9 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความวิจัยในหัวข้อเรื่อง “ตัวแปลงแผนภาพกิจกรรมสำหรับรูปแบบมาตรฐานเอ็กซ์เอ็มไอ” โดย ธนกฤต เจริญรัตนากร และ ญาใจ ลิ้มปิยะกรณ์ ในวารสารรามคำแหง ฉบับวิศวกรรมศาสตร์ (Ramkhamhaeng Journal of Engineering) ปีที่ 8 ฉบับที่ 1 เดือนพฤษภาคม 2557

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ยูเอ็มแอล (UML: Unified Modeling Language)

ปัจจุบัน ยูเอ็มแอลเป็นเครื่องมือที่ได้รับการยอมรับอย่างแพร่หลาย และเริ่มประยุกต์ใช้กับระบบงานมากขึ้น เพราะเป็นเครื่องมือที่มีความหลากหลายในการจำลองแบบซอฟต์แวร์ รวมทั้งเป็นโมเดลมาตรฐานที่ใช้หลักการออกแบบโปรแกรมเชิงวัตถุ (Object Oriented Programming) รูปแบบของภาษามีสัญกรณ์ (Notation) เป็นสัญลักษณ์สำหรับสื่อความหมาย มีกฎระเบียบที่มีความหมายต่อการเขียนโปรแกรม ดังนั้น การใช้ยูเอ็มแอลจะต้องทราบความหมายของสัญกรณ์ เช่น generalization, association, dependency, class, package เป็นต้น สิ่งเหล่านี้มีความจำเป็นต่อการตีความการออกแบบ ก่อนนำไปอิมพลีเมนต์ระบบงานจริง

โครงสร้างพื้นฐานของยูเอ็มแอลประกอบด้วยองค์ประกอบย่อย 3 ส่วน คือ

1. สัญลักษณ์ทั่วไป (Things) คือ สัญลักษณ์พื้นฐานที่ถูกใช้งานในการสร้างแผนภาพต่างๆ ของภาษายูเอ็มแอล ถือว่าเป็นรูปแบบที่เล็กที่สุดของแบบจำลอง (model)
2. ความสัมพันธ์ (Relationships) เป็นสิ่งที่ใช้แสดงความสัมพันธ์ระหว่างสัญลักษณ์ทั่วไปมี 4 ชนิด คือ
 - 2.1. ความสัมพันธ์แบบขึ้นต่อกัน (Dependency) คุณสมบัติของสิ่งหนึ่งขึ้นอยู่กับคุณสมบัตินี้ของอีกสิ่งหนึ่ง
 - 2.2. ความสัมพันธ์แบบเกี่ยวข้องกัน (Association) สิ่งสองสิ่งที่มีความสัมพันธ์เชื่อมโยงกัน
 - 2.3. ความสัมพันธ์แบบทั่วไป (Generalization) คุณสมบัติของสิ่งหนึ่งเป็นคุณสมบัตินี้พื้นฐานของอีกสิ่งหนึ่งซึ่งอาจจะมีคุณสมบัตินี้มากกว่าคุณสมบัตินี้พื้นฐานนั้น
 - 2.4. ความสัมพันธ์แบบต้นแบบ (Realization) สิ่งหนึ่งถูกสร้างให้มีคุณสมบัตินี้ของอีกสิ่งหนึ่ง
3. แผนภาพ (Diagrams) คือ การนำเสนอเชิงกราฟิกที่เกิดจากรวมสัญลักษณ์ทั่วไปมาสัมพันธ์กันจนได้แผนภาพชนิดต่างๆที่มีความหมายตามต้องการ ใน UML 2.0 มีแผนภาพทั้งหมด 13 ประเภท สามารถจัดกลุ่มได้ดังนี้

แผนภาพโครงสร้าง (structure diagrams)

3.1. Class diagram

3.1.1. Component diagram

3.1.2. Composite structure diagram

3.1.3. Deployment diagram

3.1.4. Object diagram

3.1.5. Package diagram

3.2. แผนภาพพฤติกรรม (behavior diagrams)

3.2.6. Activity diagram

3.2.7. State Machine diagram

3.2.8. Use case diagram

3.3. แผนภาพการโต้ตอบ (interaction diagrams)

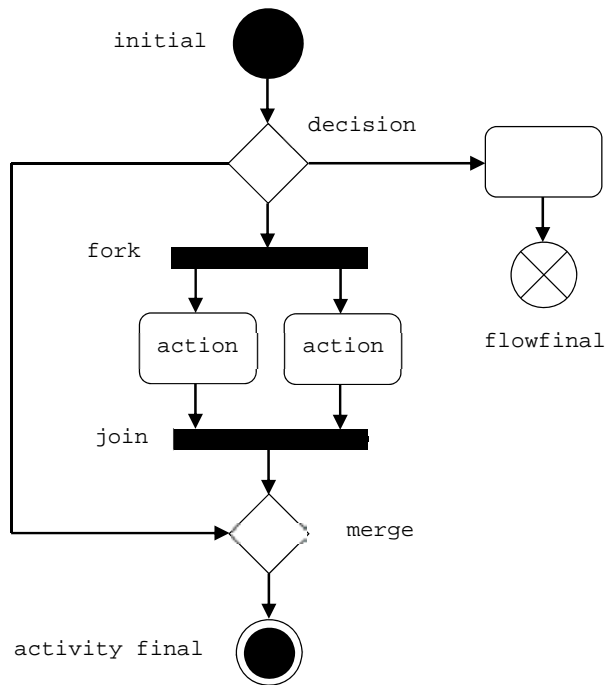
3.3.9. Communication diagram

3.3.10. Interaction overview diagram (UML 2.0)

3.3.11. Sequence diagram

3.3.12. UML Timing diagram




แผนภาพ UML ที่งานวิจัยนี้ให้ความสนใจคือ แผนภาพกิจกรรมยูเอ็มแอลถูกนำมาใช้ อย่างแพร่หลายเพื่ออธิบายกระบวนการทำงานด้วยสัญญาณ ซึ่งใช้แทนกิจกรรมและแสดงความสัมพันธ์ระหว่างกิจกรรมให้สามารถทำความเข้าใจถึงกระแสดงาน (workflow) ต่าง ๆ ประกอบด้วย โหนดและเส้นเชื่อม โดยโหนดก็จะแบ่งออกเป็นหลายประเภท คือ โหนดของกิจกรรม (Action node) โหนดของการควบคุมกิจกรรม (Control node) และโหนดของออบเจค (Object node) ส่วนเส้นเชื่อมแบ่งออกเป็นสองรูปแบบ คือ เส้นเชื่อมสายงานควบคุม (Control flow) และเส้นเชื่อมสายงานวัตถุ (Object flow) แผนภาพกิจกรรมควรเริ่มต้นจาก โหนดเริ่มต้น (Initial node) ก่อนเข้าแสดงานและจบกระแสดงานโดย โหนดสิ้นสุด (Final node) เพื่อบอกจุดสิ้นสุดของกระแสดงาน ตัวอย่างดังแสดง ในภาพที่ 1

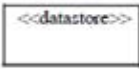


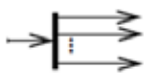

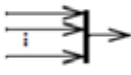

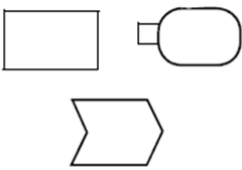

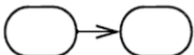
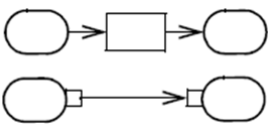


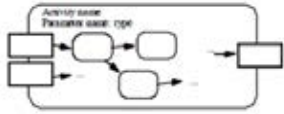

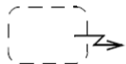
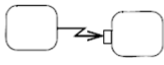
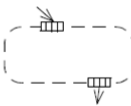
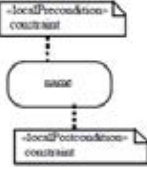

ภาพที่ 1 ตัวอย่างแผนภาพกิจกรรมแบบ Intermediate

รูปสัญลักษณ์ต่างๆที่ใช้ในการออกแบบ ประเภทของสัญลักษณ์ และความสัมพันธ์ระหว่างสัญลักษณ์กับระดับของกรอบงานแผนภาพกิจกรรมตัวอย่าง ในงานวิจัย [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

ประเภท	สัญลักษณ์	Fundamental	Basic	Intermediate	Complete	Structured	CompleteStructured	ExtraStructured
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			
SendSignalAction					X			X
PATH								
ControlFlow			X	X	X	X	X	X
ObjectFlow			X	X	X			
OTHER ELEMENTS								

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

2.1.2 เอ็กซ์เอ็มแอล (XML : eXtensible Markup Language) [5]

เป็นภาษาที่ออกแบบให้สามารถขยายความสามารถในการจัดการกับข้อมูลนอกเหนือจากการเก็บข้อมูลเพียงอย่างเดียว พัฒนาโดย W3C โดยพื้นฐานมาจากภาษาเอสจีเอ็มแอล (SGML: Standard Generalized Markup Language) และทำการลดทอนไวยากรณ์ที่ซับซ้อนของเอสจีเอ็มแอล เพื่อให้ง่ายต่อการเขียน และสามารถประมวลผลได้ทั้งจากโปรแกรมประยุกต์และบุคคลทั่วไป เอกสารเอ็กซ์เอ็มแอลจะมีการอธิบายข้อมูลในส่วนรูปแบบที่เป็นลำดับขั้น (Hierarchy)

เหมือนต้นไม้ (Tree) ทำให้โปรแกรมประยุกต์สามารถเข้าถึงเพื่อทำการประมวลผลเอกสารและเชื่อมต่อกับเอกสารเอชทีเอ็มแอล (HTML) ในการแสดงผลบนเว็บเบราว์เซอร์ได้อย่างอิสระ

เอ็กซ์เอ็มแอลประกอบด้วยสองส่วนหลัก คือ

- ข้อมูลบนเอกสารเอ็กซ์เอ็มแอลคือ ข้อมูลรวมกับแท็กเอ็กซ์เอ็มแอลที่อธิบายความหมายและโครงสร้างของข้อมูล รวมถึงคุณลักษณะต่างๆ ของข้อมูลที่บรรจุอยู่ในเอกสารในลักษณะโครงสร้างลำดับชั้น ดังแสดงใน ภาพที่ 2
- ดีทีดี (DTD: Document Type Declaration) และเอ็กซ์เอ็มแอลสคีมา (XML Schema) เป็นตัวกำหนดกฎว่า สิ่งใดที่สามารถและไม่สามารถรวมอยู่ในเอกสารเอ็กซ์เอ็มแอล รวมถึงรายละเอียดของเนื้อหาเอกสาร เป็นไวยากรณ์ของเอกสารเอ็กซ์เอ็มแอลที่มีรูปแบบถูกต้อง (Well-formed document) ดังแสดงในภาพที่ 3

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<shiporder orderid="889923"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="shiporder.xsd">
  <orderperson>John Smith</orderperson>
  <shipto>
    <name>Ola Nordmann</name>
    <address>Langgt 23</address>
    <city>4000 Stavanger</city>
    <country>Norway</country>
  </shipto>
  <item>
    <title>Empire Burlesque</title>
    <note>Special Edition</note>
    <quantity>1</quantity>
    <price>10.90</price>
  </item>
  <item>
    <title>Hide your heart</title>
    <quantity>1</quantity>
    <price>9.90</price>
  </item>
</shiporder>
```

ภาพที่ 2 ตัวอย่างลักษณะโครงสร้างลำดับชั้นของเอกสาร XML

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="shiporder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="orderperson" type="xs:string"/>
      <xs:element name="shipto">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="address" type="xs:string"/>
            <xs:element name="city" type="xs:string"/>
            <xs:element name="country" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="item" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="title" type="xs:string"/>
            <xs:element name="note" type="xs:string" minOccurs="0"/>
            <xs:element name="quantity" type="xs:positiveInteger"/>
            <xs:element name="price" type="xs:decimal"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="orderid" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

</xs:schema>

```

ภาพที่ 3 ตัวอย่าง XML Schema ชื่อ "shiporder.xsd" [5]

กฎไวยากรณ์ของเอ็กซ์เอ็มแอล

- แต่ละเอกสารเอ็กซ์เอ็มแอลสามารถมีโนดราก (Root Element) ได้เพียงโนดเดียวเท่านั้น
- ชื่อแท็กปิดจะเหมือนกันกับแท็กเปิดเพียงแต่แท็กปิดจะมีเครื่องหมาย "/" นำหน้า
- ลำดับของแท็กปิดจะสะท้อนให้เห็นลำดับของแท็กเปิด จะไม่สามารถมีการซ้อนเหลื่อมกันของแท็ก (Overlap) ได้
- สำหรับบางแท็กไม่ต้องการแท็กปิดเนื่องจากแท็กดังกล่าวได้รวมเครื่องหมาย "/" ในแท็กนั้น ๆ แล้วเช่น

- แท็กว่าง สามารถแสดงได้จาก 2 รูปแบบ คือ แบบ 1 บรรทัดคือ <emptyNode /> และอีกรูปแบบคือ <emptyNode></emptyNode> ซึ่งไม่นิยมใช้
- เมื่อมีแท็กเปิด จำเป็นต้องมีแท็กปิดเสมอ

- ชื่อแท็กมีคุณสมบัติ case-sensitive คือ ตัวพิมพ์เล็ก-ตัวพิมพ์ใหญ่ ถือว่าเป็นคนละแท็กกัน
- แอทริบิวต์จะต้องกำหนดค่าของข้อมูล อยู่ในเครื่องหมาย Double Quote ("") หรือ Single Quote (') เท่านั้น
- อักขระที่สงวนไว้ 5 ตัว คือ <, >, &, ", ' ไม่สามารถใช้ในการระบุข้อมูลได้ หากต้องการใช้อักขระสงวน จะต้องมีการเขียนตามรูปแบบที่เอ็กซ์เอ็มแอลกำหนด
- ห้ามใช้ตัวอักษร xml นำหน้าในการตั้งชื่อแท็กที่กำหนดขึ้นเอง

2.1.3 DOM (Document Object Model) [6]

DOM เป็นมาตรฐานของ W3C กำหนดขึ้นเพื่อใช้ในการเข้าถึงเอกสาร แบ่งออกเป็น 3 ส่วนที่แตกต่างกัน ดังนี้

1. Core DOM เป็นแบบจำลองมาตรฐานสำหรับทุกๆ โครงสร้างเอกสาร
2. XML DOM เป็นแบบจำลองมาตรฐานสำหรับเอกสารเอ็กซ์เอ็มแอล
3. HTML DOM เป็นแบบจำลองมาตรฐานสำหรับเอกสารเอ็ชทีเอ็มแอล

โครงสร้างของ DOM ประกอบด้วย

- Document node คือ โหนดสำหรับเอกสารทั้งหมด
- Element node คือ โหนดสำหรับทุก ๆ XML elements
- Text node คือ ค่าที่เก็บอยู่ใน XML elements
- Attribute node คือ โหนดสำหรับทุก ๆ แอทริบิวต์
- Comment nodes คือ โหนดสำหรับ comments

งานวิจัยนี้ใช้ XML DOM ซึ่งถูกสร้างขึ้นเพื่อใช้อธิบายการเข้าถึงและจัดการกับ XML Document ดังภาพที่ 4 โดยจะมองเอกสารเอ็กซ์เอ็มแอลในรูปแบบของโครงสร้างต้นไม้ เรียกว่า โหนดทรี (node-tree) ทุกโหนดสามารถเข้าถึงภายใต้โครงสร้างต้นไม้เพื่อเข้าไปจัดการกับข้อมูลภายใน โดยโหนดทรีจะแสดงกลุ่มของโหนดรวมถึงการติดต่อกันของแต่ละโหนดด้วย เริ่มต้นจากโหนดราก (root node) คือ bookstore และแตกกิ่งก้านไปจนถึงแท็กซีโนด (text node) ซึ่งเป็นโหนดล่างสุดของโครงสร้างต้นไม้ ดังแสดงในภาพที่ 5 เงื่อนไขของโครงสร้างต้นไม้มีทั้งหมด 5 ข้อ ดังนี้

1. โหนดที่อยู่บนสุดเรียกว่า ราก (root)
2. ทุกโหนดจะต้องมีโหนดพ่อแม่ (Parent Node) ยกเว้นโหนดราก

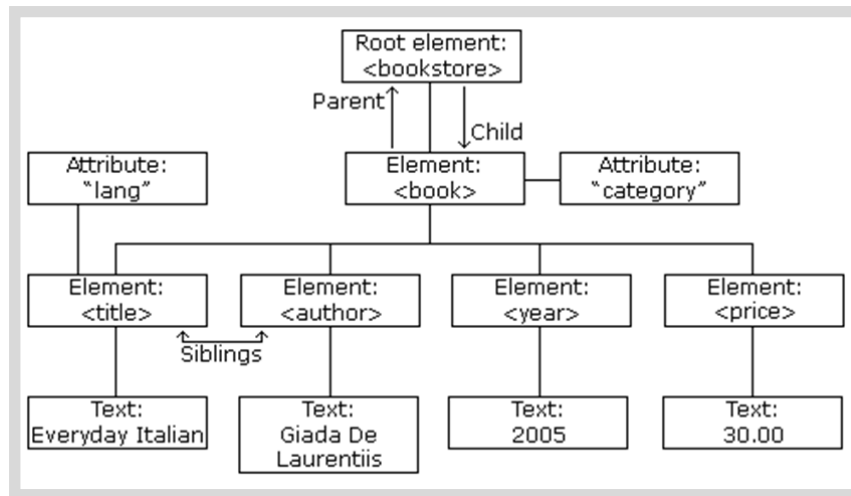
3. หนึ่งโหนดใดๆ จะสามารถมีลูกที่โหนดก็ได้
4. ใบ (Leaf) จะต้องไม่มีโหนดลูก
5. ซิบลิงส์ (Siblings) จะต้องเป็นโหนดที่อยู่ในระดับเดียวกัน

XML DOM มีคุณลักษณะสำคัญ 4 ประการ ประกอบด้วย

1. มาตรฐานแบบจำลองวัตถุของ XML
2. มาตรฐานสำหรับโปรแกรมมิ่งอินเตอร์เฟซของ XML
3. ไม่ขึ้นอยู่กับแพลตฟอร์ม
4. เป็นมาตรฐานของ W3C

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J. K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

ภาพที่ 4 ตัวอย่าง XML DOM



ภาพที่ 5 ตัวอย่าง XML DOM Node Tree

2.1.4 เอ็กซ์เอ็มไอ (XML Metadata Interchange–XMI)) [3]

เป็นมาตรฐานที่กำหนดขึ้นโดยโอเอ็มจี เพื่อใช้ในการแลกเปลี่ยนข้อมูลเมตาเดตาบนพื้นฐานของภาษาเอ็กซ์เอ็มแอล หน้าหลักคือใช้เพื่อสร้างมาตรฐานในการแลกเปลี่ยนข้อมูลในรูปแบบที่แตกต่างกันของแผนภาพยูเอ็มแอล เอ็กซ์เอ็มไอได้นิยามมุมมองที่เกี่ยวข้องกับวัตถุในเอ็กซ์เอ็มแอล ดังนี้

1. ในแง่ของเอกสารเอ็กซ์เอ็มแอล ประกอบด้วยแท็กของโนดที่เป็นตัวแทนของวัตถุและแอทริบิวต์ของโนดเป็นพื้นฐาน
2. เอกสารเอ็กซ์เอ็มแอลสามารถอ้างอิงเอ็กซ์เอ็มไอกับเอกสารเอ็กซ์เอ็มแอลอื่นได้โดยอาศัย Xlink technology
3. สามารถตรวจสอบความถูกต้องของเอ็กซ์เอ็มแอล โดยใช้เอ็กซ์เอ็มแอลสคีมา

EBNF product rule ถูกกำหนดขึ้นในแต่ละเวอร์ชันของเอกสารเอ็กซ์เอ็มไอ เพื่อใช้ในการสร้างเอกสารเอ็กซ์เอ็มแอล และสคีมา เพื่อเพิ่มความสามารถในการแลกเปลี่ยนวัตถุได้อย่างมีประสิทธิภาพ

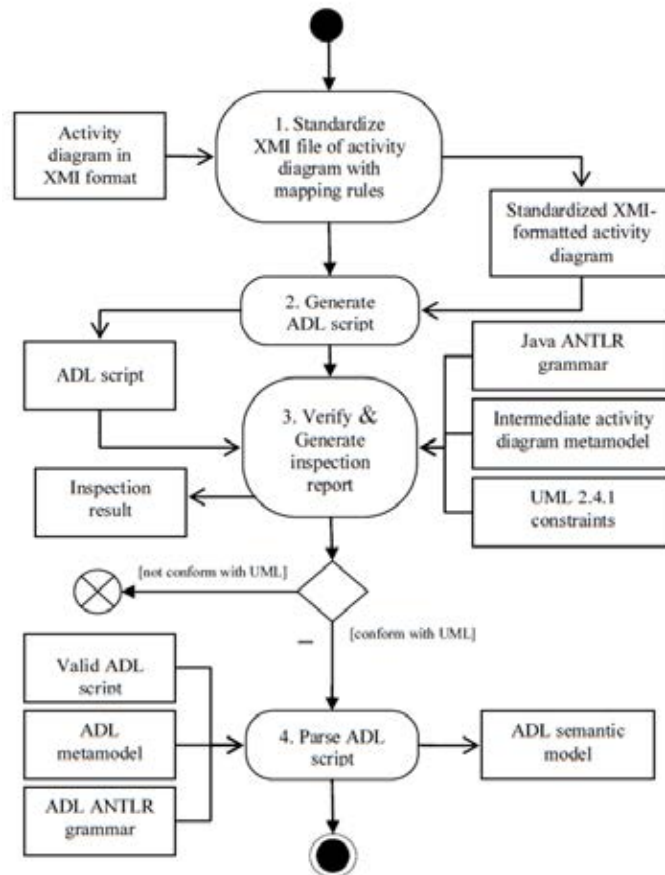
2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Enhancement of Action Description Language for UML Activity Diagram Review [2]

งานวิจัย [2] ได้นำเสนอวิธีการขยายความสามารถของภาษาอธิบายการกระทำหรือเอดี

แอล (Action Description Language - ADL) ที่พัฒนาขึ้นในงานวิจัย [4] เพื่อใช้ตรวจทานแผนภาพกิจกรรมที่สร้างโดยเครื่องมือยูเอ็มแอลเสร็จเรียบร้อยแล้ว กระบวนการตรวจทานแบ่งออกเป็น 4 ขั้นตอนหลัก ดังแสดงในภาพที่ 6 : 1) ขั้นตอนการใช้กฎการเทียบ (mapping rules) เพื่อแปลงไฟล์นำเข้าเอ็กซ์เอ็มไอของแผนภาพกิจกรรมที่สร้างขึ้นจากเครื่องมือ 2 ชนิด คือ AgroUML และ Modelio ให้อยู่ในรูปแบบเอ็กซ์เอ็มไอที่กำหนดขึ้นเพื่อประมวลผลในงานวิจัยนี้ 2) ขั้นตอนการสร้างสคริปต์เอดีแอลจากผลลัพธ์ที่ได้จากขั้นตอนแรก 3) ขั้นตอนการทวนสอบแผนภาพกิจกรรมและรายงานผลการตรวจวิเคราะห์ 4) ขั้นตอนการสร้างโมเดลความหมายเอดีแอล (ADL semantic model) ด้วยวิธีการแจงส่วน (Parse) โดยมีสิ่งนำเข้าประกอบด้วย สคริปต์เอดีแอลที่ถูกต้อง เอดีแอลเมตาโมเดล และ ไวยากรณ์ตัวสร้างตัวแจงส่วนคือ ADL ANTLR grammar

อย่างไรก็ตาม งานวิจัยดังกล่าวยังมีข้อจำกัดในขั้นตอนที่ 1 กล่าวคือ สามารถตรวจทานความถูกต้องของแผนภาพกิจกรรมที่สร้างขึ้นจากเครื่องมือ 2 ชนิดเท่านั้น



ภาพที่ 6 ภาพรวมระบบทบทวนแผนภาพกิจกรรม [2]

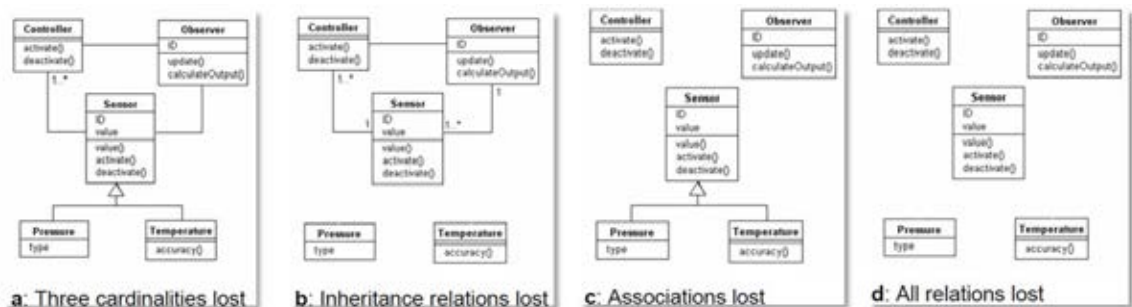
2.2.2 OSS tools in a heterogeneous environment for embedded systems modelling: an analysis of adoptions of XMI [7]

งานวิจัย [7] ได้ทำการศึกษาการเข้ากันได้ของเครื่องมือสำหรับการสร้างแผนภาพคลาส ยูเอ็มแอลที่สนับสนุนการนำออกข้อมูลเป็นไฟล์เอ็กซ์เอ็มไอ โดยได้ทำการเลือกเครื่องมือต่างๆ เพื่อใช้ศึกษาพร้อมทั้งรวบรวมรายละเอียดของการนำเข้า-ออกไฟล์เอ็กซ์เอ็มไอ และยูเอ็มแอลเวอร์ชันของแต่ละเครื่องมือนั้นๆ

ตารางที่ 2 สรุปผลการแลกเปลี่ยนข้อมูลระหว่างเครื่องมือ เช่น Argo สามารถนำออกข้อมูลเอ็กซ์เอ็มไอและนำไปเข้า Poseidon ได้อย่างสมบูรณ์ ขณะที่ข้อมูลเอ็กซ์เอ็มไอที่นำออกจาก Poseidon ไม่สามารถนำเข้าไปใช้งานใน Argo ได้ เป็นต้น

ตารางที่ 2 ผลการแลกเปลี่ยนข้อมูลเอ็กซ์เอ็มไอแผนภาพคลาสระหว่างเครื่องมือที่เลือกใช้ในงานวิจัย [7]

Import → Export ↓	Argo	Fujaba	Umbrello	Artisan	Poseidon	Rhapsody	Rose	TAU	Visio
Argo	Y	N	N	N	Y	N,c	N,a	Y	---
Fujaba	N	Y	N	N,d ¹	N	N	N	Y	---
Umbrello	N	N	Y	N	N	N	N	N	---
Artisan	N	N	N,d	N,b	N	N	N,c	N,b	---
Poseidon	N	N	N,d	N,b ¹	Y	N	N,c	Y	---
Rhapsody	Y	N	N	N	N	Y	N	Y	---
Rose 1.0/1.1	Y N	N N	N N	N N,b ¹	Y N,c	N,c N	Y	Y Y	---
TAU	---	---	---	---	---	---	---	---	---
Visio	Y	N	N	N	N	N	N	Y	---



ภาพที่ 7 รูปแบบการสูญหายจากการแลกเปลี่ยนข้อมูลเอ็กซ์เอ็มไอระหว่างต่างเครื่องมือ [7]

N คือ ไม่สามารถแลกเปลี่ยนข้อมูลระหว่างเครื่องมือได้ ตัวอักษร a, b, c, และ d แทนรูปแบบของการสูญหาย ดังแสดงในรูปที่ 2 กล่าวคือ

N,a: Three cardinalities lost

N,b: Inheritance relations lost

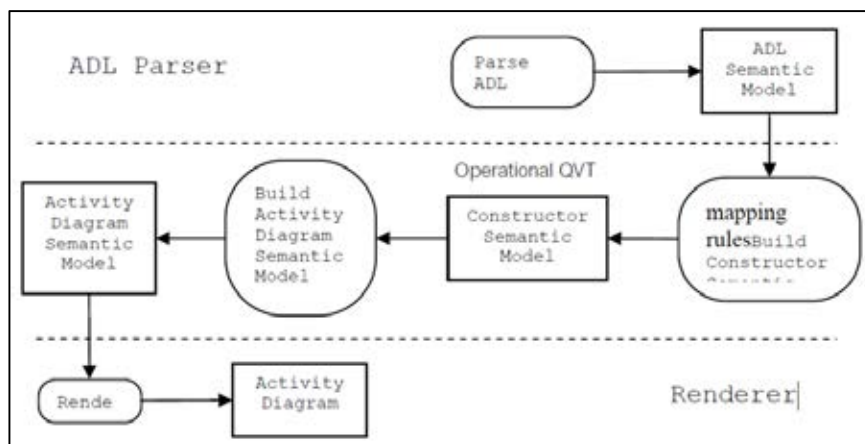
N,c: Associations lost

N,d: All relations lost

ผลลัพธ์ที่ได้จากการศึกษาพบว่าเครื่องมือที่ใช้ในการศึกษาที่แตกต่างกันนั้น โดยส่วนใหญ่ไม่สามารถแลกเปลี่ยนข้อมูลด้วยรูปแบบเอ็กซ์เอ็มไอระหว่างกันได้อย่างไรก็ตาม งานวิจัยนี้ไม่ได้นำเสนอในส่วนของวิธีการที่สามารถทำให้รูปแบบเอ็กซ์เอ็มไอเข้ากันได้ เพื่อใช้ในการแลกเปลี่ยนข้อมูลระหว่างเครื่องมือที่แตกต่างกัน

2.2.3 Rendering UML Activity Diagrams as a Domain Specific Language - ADL [4]

ได้เสนอข้อกำหนดครุภัณฑ์เพื่อทวนสอบแผนภาพกิจกรรมของกระแสระบวนการ โดยใช้ภาษาอธิบายการกระทำหรือเอดีแอล จัดอยู่ในประเภทหนึ่งของภาษาจำเพาะโดเมนหรือดีเอสแอล (Domain Specific Language – DSL) งานวิจัยได้นำเสนอวิธีการทวนสอบสำหรับการสร้างแผนภาพกิจกรรมจากบทคำสั่งเอดีแอลที่มีการระบุความสัมพันธ์ระหว่างวัตถุในการกำหนดบังคับควบคุมต่างๆ อีกทั้งยังมีการตรวจสอบความสมเหตุสมผลของข้อมูลและพฤติกรรมในแผนภาพกิจกรรม ผลลัพธ์ที่ได้คือแผนภาพยูเอ็มแอลที่ผ่านกระบวนการของเอดีแอล แปลงเป็นไฟล์เอ็กซ์เอ็มไอ และส่งออกเป็นผลลัพธ์คือ แผนภาพยูเอ็มแอลที่มีการตรวจสอบความสัมพันธ์แล้ว



ภาพที่ 8 ภาพรวมการทำงานของระบบที่ได้จากงานวิจัย [4]

ภาพที่ 8 แสดงภาพรวมของระบบที่พัฒนาขึ้นในงานวิจัย [4] แบ่งการทำงานออกเป็น 3 ขั้นตอน คือ 1) กระบวนการแจกส่วนเอดีแอล ทำหน้าที่แปลคำสั่งที่ถูกเขียนอยู่ในรูปแบบภาษาเอดีแอลสคริปต์ ให้เป็นภาษาเอดีแอลที่มีความหมายในโมเดล 2) ขั้นตอนการแปลงแบบจำลอง หลังจากได้เอดีแอลที่มีความหมายทางโมเดลแล้ว ส่วนนี้จะทำหน้าที่แปลภาษาเอดีแอลให้มีความหมายกับการทำแผนภาพกิจกรรม ผ่านกฎความสัมพันธ์ที่กำหนดขึ้นได้ผลลัพธ์เป็นแบบจำลองแผนภาพกิจกรรมที่มีความหมาย และ 3) ขั้นตอนการแสดงผลแผนภาพกิจกรรม จะนำผลลัพธ์ที่ได้จากข้อ 2 แสดงผลลัพธ์ในรูปแบบกราฟ

บทที่ 3

วิธีดำเนินการวิจัย

3.1 แนวคิดวิธีดำเนินการวิจัย

งานวิจัยนี้เป็นงานวิจัยต่อยอดจากงานวิจัย [2] ซึ่งนำเสนอวิธีการขยายความสามารถของภาษาเอดีแอล ในการตรวจทานแผนภาพกิจกรรม ดังภาพที่ 6 โดยงานวิจัยนี้มุ่งเน้นที่จะลดข้อจำกัดในขั้นตอนที่ 1 ของงานวิจัย [2] ซึ่งสามารถรับข้อมูลเข้าแผนภาพกิจกรรมที่สร้างจากเครื่องมือ ArgoUML และ Modelio เท่านั้น วิธีการที่นำเสนอในงานวิจัยนี้คือ พัฒนาตัวแปลงแผนภาพกิจกรรมยูเอ็มแอลที่อยู่ในรูปแบบเอ็กซ์เอ็มไอที่ต่างเวอร์ชันกัน หรือสร้างจากเครื่องมือที่ต่างกันให้อยู่ในรูปแบบเอ็กซ์เอ็มไอเดียวกัน ซึ่งเป็นรูปแบบของข้อมูลนำเข้าของระบบที่พัฒนาขึ้นในงานวิจัย [2] ภาพที่ 9 แสดงภาพรวมการทำงานของระบบ ประกอบด้วยขั้นตอนการทำงานหลัก 3 ขั้นตอน ดังนี้

- 1) ขั้นตอนตรวจสอบความถูกต้องของไฟล์เอ็กซ์เอ็มไอ กล่าวคือ ตรวจสอบว่าไฟล์นำเข้าอยู่ในรูปแบบมาตรฐานเอ็กซ์เอ็มไอที่กำหนดโดยโอเอ็มจีหรือไม่ รวมทั้งทำการตรวจสอบเวอร์ชันของไฟล์เอ็กซ์เอ็มไอจากเมตาเดตา

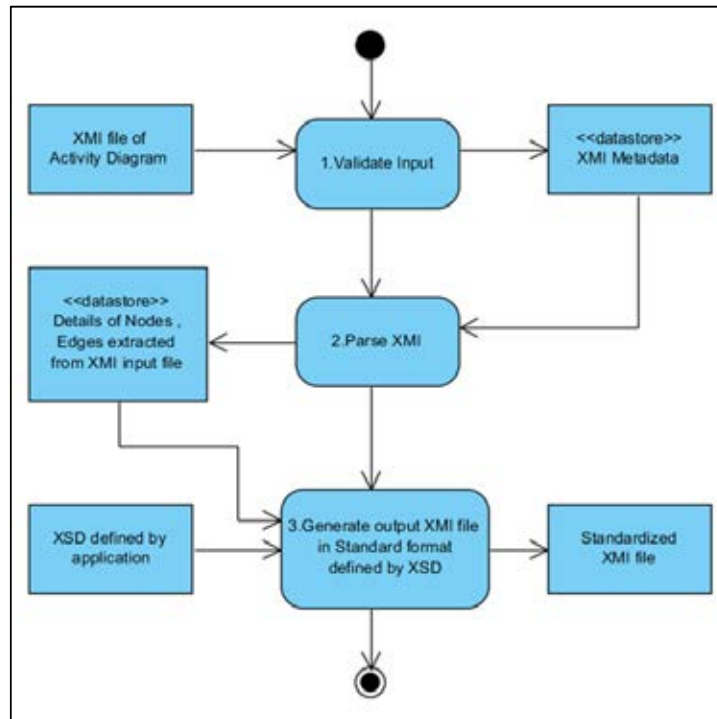
ในงานวิจัยนี้รองรับการตรวจหารูปแบบไฟล์เอ็กซ์เอ็มไอแบ่งเป็น 2 ประเภทตามเวอร์ชัน กล่าวคือ 1) รูปแบบเอ็กซ์เอ็มไอที่เวอร์ชัน 1.4 และ 2) รูปแบบเอ็กซ์เอ็มไอที่เวอร์ชันมากกว่า 1.4

- 2) ขั้นตอนแจงส่วนเอ็กซ์เอ็มไอ จะทำการสกัดข้อมูลจากไฟล์เอ็กซ์เอ็มไอโดยดูจากเวอร์ชันของไฟล์เอ็กซ์เอ็มไอที่ได้จากเมตาเดตาในขั้นตอนแรก เพื่อกำหนดรูปแบบการเข้าถึงข้อมูล

เนื่องจากมาตรฐานเอ็กซ์เอ็มไอที่ถูกกำหนดขึ้นโดยโอเอ็มจีมีการปรับปรุงมาตรฐานอย่างต่อเนื่อง แต่การเปลี่ยนแปลงครั้งสำคัญสำหรับมาตรฐานของเอ็กซ์เอ็มไอคือ การปรับเวอร์ชันหลังจาก 1.4 เป็นต้นไป กล่าวคือโครงสร้างของเอ็กซ์เอ็มไอที่กำหนดไว้เพื่อเข้าถึงชุดข้อมูลโนดและชุดข้อมูลเส้นเชื่อมสำหรับเวอร์ชันที่ 1.4 จะถูกกำหนดโดยชื่อของแท็ก เช่น `<UML:ActionState />` แทนการเข้าถึงชุดข้อมูลโนด และ `<UML:Transition />` แทนการเข้าถึงชุดข้อมูลเส้นเชื่อมสำหรับโครงสร้างของเอ็กซ์เอ็มไอที่เวอร์ชันมากกว่า 1.4 การเข้าถึงชุดข้อมูลโนดและชุดข้อมูลเส้นเชื่อมจะเปลี่ยนเป็นแท็กที่ชื่อว่า `<node />` และ `<edge />` แทน

- 3) ขั้นตอนการสร้างไฟล์นำออกเอ็กซ์เอ็มไอตามรูปแบบมาตรฐานที่ผู้ใช้งานกำหนด โดยผู้วิจัยได้กำหนดเอ็กซ์เอ็มแอลสคีม่า ดังรูปที่ 5 เพื่อเป็นรูปแบบมาตรฐานสำหรับสร้างไฟล์เอ็กซ์เอ็มไอที่สามารถใช้งานได้กับงานวิจัย [2] และทำการเชื่อมต่อชุดข้อมูลโนดและชุดข้อมูลเส้นเชื่อมกับรูปแบบมาตรฐานก่อนนำออกเป็นไฟล์เอ็กซ์เอ็มไอ

ทั้งนี้ หากมีเวอร์ชันใหม่ของรูปแบบเอ็กซ์เอ็มไอ วิธีการที่ผู้วิจัยนำเสนอจะต้องสามารถทำการปรับปรุงข้อมูลในเอกสารเอ็กซ์เอ็มไอ จากเวอร์ชันที่ต่ำกว่าให้เป็นเอกสารเอ็กซ์เอ็มไอตามมาตรฐานที่เกิดขึ้นใหม่ได้ โดยใช้แม่แบบเอกสาร ในการระบุรูปแบบของเอกสารใหม่ ซึ่งในการพัฒนาระบบ ผู้วิจัยเลือกพัฒนาซอฟต์แวร์ ปลั๊กอินอีคลิปส์ เนื่องจากมีความเหมาะสมในการใช้งานและง่ายต่อการพัฒนาต่อยอด รวมทั้งสะดวกในการให้นักวิจัยท่านอื่นสามารถเลือกติดตั้งใช้งานผ่านเว็บไซต์ได้



ภาพที่ 9 ภาพรวมระบบตัวแปลงรูปแบบมาตรฐานเอ็กซ์เอ็มไอสำหรับแผนภาพกิจกรรม

3.2 การตรวจสอบความถูกต้องของไฟล์เอ็กซ์เอ็มไอ

จากภาพที่ 9 ระบบจะทำการตรวจสอบว่าไฟล์นำเข้าอยู่ในรูปแบบมาตรฐานเอ็กซ์เอ็มไอที่กำหนดโดยโอเอ็มจีหรือไม่ หลังจากนั้นระบบจะทำการตรวจสอบจากเมตาเดตาเพื่อหาเวอร์ชันของไฟล์เอ็กซ์เอ็มไอจากโนดรากเพื่อส่งให้ตัวแปลงเอกสารทำงานต่อไป

```

1 <?xml version = '1.0' encoding = 'UTF-8' ?>
2 <XMI xmi.version = '1.4' xmlns:UML = 'org.omg.xmi.namespace.UML'
  timestamp = 'Sat Oct 27 23:34:14 ICT 2012'>
3 <XMI.header><XMI.documentation>
4   <XMI.exporter>ArgoUML (using Netbeans XMI Writer version
5     1.0)</XMI.exporter>
6   <XMI.exporterVersion>0.34(6) revised on $Date: 2010-01-11
7     22:20:14 +0100 (Mon, 11 Jan 2010) $ </XMI.exporterVersion>
8 </XMI.documentation>
9 </XMI.header >
10

```

ภาพที่ 10 ตัวอย่างโน้ตกรากที่มีแอททริบิวต์สำหรับเก็บเอ็กซ์เอ็มไอเวอร์ชัน

ในงานวิจัยนี้รองรับการตรวจหารูปแบบไฟล์เอ็กซ์เอ็มไอแบ่งเป็น 2 ประเภทตามเวอร์ชัน กล่าวคือ 1) รูปแบบเอ็กซ์เอ็มไอที่เวอร์ชัน 1.4 และ 2) รูปแบบเอ็กซ์เอ็มไอที่เวอร์ชันมากกว่า 1.4 เนื่องจากมาตรฐานเอ็กซ์เอ็มไอที่ถูกกำหนดขึ้นโดยโอเอ็มจีมีการปรับปรุงมาตรฐานอย่างต่อเนื่อง แต่การเปลี่ยนแปลงครั้งสำคัญสำหรับมาตรฐานของเอ็กซ์เอ็มไอคือ การปรับเวอร์ชันหลังจาก 1.4 เป็นต้นไป กล่าวคือโครงสร้างของเอ็กซ์เอ็มไอที่กำหนดไว้เพื่อเข้าถึงชุดข้อมูลโน้ดและชุดข้อมูลเส้นเชื่อมสำหรับเวอร์ชันที่ 1.4 จะถูกกำหนดโดยชื่อของแท็ก เช่น `<UML:ActionState />` แทนการเข้าถึงชุดข้อมูลโน้ด และ `<UML:Transition. />` แทนการเข้าถึงชุดข้อมูลเส้นเชื่อม สำหรับโครงสร้างของเอ็กซ์เอ็มไอที่เวอร์ชันมากกว่า 1.4 การเข้าถึงชุดข้อมูลโน้ดและชุดข้อมูลเส้นเชื่อมจะเปลี่ยนเป็นแท็กที่ชื่อว่า `<node />` และ `<edge />` แทน

3.3 การแจ่งส่วนเอ็กซ์เอ็มไอ

จะทำการสกัดข้อมูลจากไฟล์เอ็กซ์เอ็มไอโดยดูจากเวอร์ชันของไฟล์เอ็กซ์เอ็มไอที่ได้จากเมตาเดตาในขั้นต้นแรก เพื่อกำหนดรูปแบบการเข้าถึงข้อมูลโดยแบ่งเป็น 2 ส่วนดังนี้

3.3.1 ประเภทของไฟล์เอ็กซ์เอ็มไอที่เวอร์ชัน1.4

จากการทดสอบด้วยเครื่องมือที่นำมาใช้ในงานวิจัยสำหรับไฟล์เอ็กซ์เอ็มไอที่เวอร์ชัน 1.4 คือ โอโกลยูเอ็มแอลเป็นซอฟต์แวร์ที่ใช้สำหรับออกแบบแผนภาพยูเอ็มแอล พัฒนาด้วยภาษาจาวา และเป็นซอฟต์แวร์โอเพนซอร์ส [8] รองรับเอกสารเอ็กซ์เอ็มไอตามมาตรฐานเอ็กซ์เอ็มไอที่เวอร์ชัน 1.4 และรองรับมาตรฐานยูเอ็มแอลเวอร์ชัน 1.4

ประกอบด้วยแท็กที่สำคัญ 3 ส่วนดังนี้

1. UML:Model ให้รายละเอียดชื่อของแผนภาพกิจกรรม รหัสของแผนภาพกิจกรรม และคุณสมบัติต่างๆ ตัวอย่างแสดงได้ดังภาพที่ 11


```

1 <UML:Model xmi.id =
  '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000865'
2   name = 'ex1' isSpecification = 'false' isRoot = 'false'
   isLeaf = 'false'
3   isAbstract = 'false'>

```

ภาพที่ 11 ภาพตัวอย่าง UML:Model

2. UML:CompositeState.subvertex ให้รายละเอียดเกี่ยวกับโนดประเภทต่าง ๆ ทั้งหมดที่อยู่ในแผนภาพกิจกรรมได้แก่ โหนดของกิจกรรม โหนดของออฟเจค และโนดควบคุม ของกิจกรรมต่างๆ ได้แก่ โหนดแยก โหนดรวม โหนดตัดสินใจ โหนดเริ่มต้น โหนดสิ้นสุด ตัวอย่างแสดงได้ดังภาพที่ 12

ประกอบด้วยแท็กที่สำคัญแบ่งออกเป็น 4 ประเภท ดังนี้

- 2.1. UML:ActionState ให้รายละเอียดข้อมูลของโนดกิจกรรม
- 2.2. UML:ObjectFlowState ให้รายละเอียดข้อมูลของโนดออฟเจค
- 2.3. UML:Pseudostate ให้รายละเอียดข้อมูลของโนดควบคุมต่างๆ ได้แก่ โหนดเริ่มต้น โหนดแยก โหนดรวม โหนดตัดสินใจ และโนดผสาน
- 2.4. UML:FinalState ให้รายละเอียดข้อมูลของโนดหยุดสายงาน

ภายในแท็กของโนดประเภทต่าง ๆ นั้น จะประกอบด้วยแท็กที่ใช้ในการอ้างอิงเส้นเชื่อม อีก 2 แท็ก คือ

- 2.5. 1) UML:StateVertex.outgoing เป็นแท็กสำหรับข้อมูลเส้นเชื่อมออก
- 2.6. 2) UML:StateVertex.incoming เป็นแท็กสำหรับข้อมูลเส้นเชื่อมเข้า โดยเส้นเชื่อมแต่ละเส้นอ้างอิงค่าจากแท็ก UML:Transition ภายใต้แอททริบิว

xmi.idref

3. UML:StateMachine.transitions ให้รายละเอียดเกี่ยวกับเส้นเชื่อมทั้งหมดที่อยู่ในแผนภาพกิจกรรม ตัวอย่างแสดงได้ดังภาพที่ 13 ประกอบด้วยแท็กที่สำคัญ 2 แท็ก ดังนี้

- 3.1. UML:Transition.source เป็นแท็กให้ข้อมูลต้นทางของเส้นเชื่อม
- 3.2. UML:Transition.target เป็นแท็กให้ข้อมูลปลายทางของเส้นเชื่อม

```

<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--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000874' />
  <UML:Transition xmi.idref =
'-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000875' />
  <UML:Transition xmi.idref =
'-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000876' />
  </UML:StateVertex.outgoing>
  <UML:StateVertex.incoming>
  <UML:Transition xmi.idref =
'-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000873' />
  </UML:StateVertex.incoming>
  <UML:State.entry>
  <UML:UninterpretedAction xmi.id =
'-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000871'
isSpecification = 'false' isAsynchronous = 'false'>
  <UML:Action.script>
  <UML:ActionExpression xmi.id =
'-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000877'
language = '' body = 'Unmarshall Order' />
  </UML:Action.script>
  </UML:UninterpretedAction>
  </UML:State.entry>
</UML:ActionState>
<UML:ObjectFlowState xmi.id =
'-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000869'
</UML:ObjectFlowState>
</UML:CompositeState.subvertex>

```

ภาพที่ 12 ภาพตัวอย่าง UML:CompositeState.subvertex

```

1 <UML:StateMachine.transitions>
2   <UML:Transition xmi.id =
3     '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000873'
4     isSpecification = 'false'>
5     <UML:Transition.source>
6       <UML:ObjectFlowState xmi.idref =
7         '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000869' />
8       </UML:Transition.source>
9       <UML:Transition.target>
10        <UML:ActionState xmi.idref =
11          '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000868' />
12        </UML:Transition.target>
13      </UML:Transition>
14    </UML:StateMachine.transitions>

```

ภาพที่ 13 ภาพตัวอย่าง UML:StateMachine.transitions

3.3.1.1. กฎในการแปลงกลุ่มของไฟล์เอ็กซ์เอ็มไอที่เวอร์ชัน 1.4

กฎในการแปลงประกอบด้วยกัน 18 ข้อ โดยทำการตรวจสอบจากชื่อของแท็ก ดังนี้

1. แท็กชื่อ UML:ActivityGraph ดำเนินการเก็บชื่อของแผนภาพกิจกรรม
2. แท็กชื่อ UML:StateMachine.transitions กำหนดสถานะพร้อมสร้างเส้นเชื่อม

3. แท็กชื่อ UML:Transition และสถานะเท่ากับพร้อมสร้างเส้นเชื่อม ดำเนินการสร้างเส้นเชื่อมขึ้นและกำหนดรหัสของเส้นเชื่อม และค่าอ้างอิงที่อ่านได้จากแอททริบิวต์ xmi.id หากเป็นเส้นเชื่อมจากโนดตัดสินใจจะมีชื่อให้เก็บจากแอททริบิวต์ชื่อ name
4. แท็กชื่อ UML:Transition.source กำหนดสถานะพร้อมจะบันทึกโนดต้นทาง
5. แท็กเป็นประเภทโนด และแท็กชื่อ UML:ActionState หรือ UML:ObjectFlowState หรือ UML:Pseudostate และสถานะเป็นพร้อมจะบันทึกโนดต้นทาง ให้ดำเนินการสร้างโนดต้นทางกับเส้นเชื่อมที่ได้จากข้อ 3
6. แท็กชื่อ UML:Transition.target กำหนดสถานะพร้อมจะบันทึกโนดปลายทาง
7. แท็กเป็นประเภทโนด และแท็กชื่อ UML:ActionState หรือ UML:ObjectFlowState หรือ UML:Pseudostate และสถานะเป็นพร้อมจะบันทึกโนดปลายทาง ให้ดำเนินการสร้างโนดปลายทางกับเส้นเชื่อมที่ได้จากข้อ 3
8. แท็กชื่อ UML:CompositeState.subvertex กำหนดสถานะพร้อมสร้างโนด
9. แท็กชื่อ UML:ActionState และสถานะเท่ากับพร้อมสร้างโนด ดำเนินการสร้างโนดขึ้น กำหนดรหัสของโนด และค่าอ้างอิงที่อ่านได้จากแอททริบิวต์ xmi.id กำหนดชื่อจากแอททริบิวต์ชื่อ name และกำหนดประเภทของโนดเท่ากับแอ็กชัน
10. แท็กชื่อ UML: ObjectFlowState และสถานะเท่ากับพร้อมสร้างโนด ดำเนินการสร้างโนดชั่วคราวขึ้นกำหนดรหัสของโนด และค่าอ้างอิงที่อ่านได้จากแอททริบิวต์ xmi.id กำหนดชื่อจาก แอททริบิวต์ชื่อ name และกำหนดประเภทของโนดเท่ากับออกพเจ็ค กำหนดสถานะพร้อมสร้างโนด
11. แท็กชื่อ UML: Pseudostate และประเภทของโนดเท่ากับ initial และสถานะเท่ากับพร้อมสร้างโนด ดำเนินการสร้างโนดชั่วคราวขึ้นกำหนดรหัสของโนด และค่าอ้างอิงที่อ่านได้จากแอททริบิวต์ xmi.id กำหนดชื่อจาก กำหนดประเภทของโนดเท่ากับโนดเริ่มต้น กำหนดสถานะพร้อมสร้างโนด
12. แท็กชื่อ UML: Pseudostate และสถานะเท่ากับพร้อมสร้างโนด ดำเนินการสร้างโนดชั่วคราวขึ้นกำหนดรหัสของโนด และค่าอ้างอิงที่อ่านได้จากแอททริบิวต์ xmi.id กำหนดชื่อจาก แอททริบิวต์ชื่อ name และกำหนดประเภทของโนดเท่ากับออกพเจ็ค กำหนดสถานะพร้อมสร้างโนด
13. แท็กชื่อ UML: FinalState และสถานะเท่ากับพร้อมสร้างโนด ดำเนินการสร้างโนดชั่วคราวขึ้นกำหนดรหัสของโนด และค่าอ้างอิงที่อ่านได้จากแอททริบิวต์ xmi.id กำหนด

ประเภทของโนดเป็นโนดสิ้นสุดกิจกรรม กำหนดสถานะพร้อมสร้างโนด

14. แท็กชื่อ UML:StateVertex.incoming และสถานะเท่ากับพร้อมสร้างโนด ดำเนินการ
สร้างเส้นเชื่อมเข้าชั่วคราว
15. แท็กชื่อ UML:Transition และ ข้อ 14 มีการสร้างเส้นเชื่อม ให้โนดชั่วคราวเพิ่มข้อมูล
เส้นเชื่อมเข้าด้วยค่าอ้างอิงของแท็ก xmi.idref
16. แท็กชื่อ UML:StateVertex.outgoing และสถานะเท่ากับพร้อมสร้างโนด ดำเนินการ
สร้างเส้นเชื่อมออกชั่วคราว
17. แท็กชื่อ UML:Transition และ ข้อ 16 มีการสร้างเส้นเชื่อม ให้โนดชั่วคราวเพิ่มข้อมูล
เส้นเชื่อมออกด้วยค่าอ้างอิงของแท็ก xmi.idref
18. แท็กปิดเป็นแท็กเป็นประเภทโนด และแท็กชื่อ UML:ActionState หรือ
UML:ObjectFlowState หรือ UML:Pseudostate ให้ทำการบันทึกโนดชั่วคราว เข้าใน
ชุดข้อมูลแผนภาพกิจกรรม

3.3.2 ไฟล์เอ็กซ์เอ็มไอทีเวอร์ชันมากกว่า 1.4

จากการทดสอบด้วยเครื่องมือที่นำมาใช้ในงานวิจัยสำหรับไฟล์เอ็กซ์เอ็มไอทีเวอร์ชันที่
มากกว่า 1.4 คือ

- 1) Visual Paradigm for UML เป็นเครื่องมือสำหรับออกแบบแผนภาพยูเอ็มแอล พัฒนา
โดยบริษัท Visual Paradigm International ในเวอร์ชัน 10.0 รองรับมาตรฐานยูเอ็มแอล 2.3 และ
รองรับเอ็กซ์เอ็มไอทีเวอร์ชัน 2.1.
- 2) โมเดลลิโอเป็นเครื่องมือสำหรับออกแบบแผนภาพยูเอ็มแอล พัฒนาโดยบริษัท
ModelioSoft ปัจจุบันเป็นซอฟต์แวร์โอเพนซอร์สอยู่ภายใต้สัญญา GPLv3 และ Key APIs อยู่
ภายใต้สัญญา ApacheLicense 2.0 ในเวอร์ชันปัจจุบัน 2.2.1 รองรับมาตรฐานยูเอ็มแอล 2.1.1
และ รองรับเอ็กซ์เอ็มไอทีเวอร์ชัน 2.1.
- 3) Rhapsody เป็นเครื่องมือสำหรับออกแบบแผนภาพยูเอ็มแอล โดยบริษัท IBM เป็น
เจ้าของซอฟต์แวร์เวอร์ชันที่นำมาทดสอบ Rhapsody Version 8.03 รองรับมาตรฐานยูเอ็มแอล
2.0 และรองรับเอ็กซ์เอ็มไอทีเวอร์ชัน 2.1.
- 4) Umodel Enterprise เป็นเครื่องมือสำหรับออกแบบแผนภาพยูเอ็มแอล โดยบริษัท
Altova เป็นเจ้าของซอฟต์แวร์ เวอร์ชันที่นำมาทดสอบ Umodel Enterprise 2013 rel. 2 sp1
รองรับมาตรฐานยูเอ็มแอล 2.4 และรองรับเอ็กซ์เอ็มไอทีเวอร์ชัน 2.4

จากการวิเคราะห์กลุ่มของซอฟต์แวร์ที่เอ็กซ์เอ็มไอเวอร์ชันมากกว่า 1.4 แท็กที่สำคัญของไฟล์เอ็กซ์เอ็มไอประกอบด้วยกัน 3 อย่างคือ

1. แท็กที่ให้รายละเอียดของแผนภาพกิจกรรม จากการทดสอบในงานวิจัยพบว่าต้องสร้างกฎความสัมพันธ์สำหรับแท็กเพื่อให้เหมาะสมกับเครื่องมือที่นำมาทดสอบ ดังแสดงในภาพที่ 14 ดังนี้
 - 1.1. แท็กชื่อ packagedElement หรือ uml:Diagram ทำหน้าที่เก็บรายละเอียดของแผนภาพกิจกรรมได้แก่ รหัสประจำตัวของแผนภาพ ชื่อของแผนภาพ และประเภทของแผนภาพกิจกรรม

```

1 <packagedElement xmi:type="uml:Activity"
  xmi:id="_wWnY4i2DEeKSKePaylnoVA" name="ex2">
2
3 <!-- modelio - Rhapsody are using <packagedElement -->
4
5 <uml:Diagram diagramType="ActivityDiagram"
  documentation="" name="Ex1" toolName="Visual Paradigm for
  UML" xmi:id="Xy4ecAKAUIDIowMD">
6
7 <!-- Visual Paradigm is using <uml:Diagram -->
8

```

ภาพที่ 14 ภาพตัวอย่าง UML:StateMachine.transitions

2. แท็ก node ให้ข้อมูลรายละเอียดของโนดประเภทต่างๆ ของแผนภาพกิจกรรม ได้แก่ รหัสประจำตัวของโนด ชื่อของโนด ประเภทของโนด และข้อมูลเส้นเชื่อม ตัวอย่างแสดงได้ดังภาพที่ 15 สำหรับข้อมูลเส้นเชื่อมสามารถกำหนดกฎความสัมพันธ์ดังนี้
 - 2.1. ถ้ามีแอททริบิวต์เส้นเชื่อมเข้า ภายในแอททริบิวต์จะเก็บข้อมูลรหัสของเส้นเชื่อมเข้า กรณีมีมากกว่า 1 เส้นจะแบ่งด้วยช่องว่าง 1 ช่อง
 - 2.2. ถ้ามีแอททริบิวต์เส้นเชื่อมออก ภายในแอททริบิวต์จะเก็บข้อมูลรหัสของเส้นเชื่อมออก กรณีมีมากกว่า 1 เส้นจะแบ่งด้วยช่องว่าง 1 ช่อง
 - 2.3. กรณีไม่เจอเส้นเชื่อมเข้า หรือเส้นเชื่อมออก ให้อ่านแท็กชื่อ incoming และ outgoing ตามลำดับ ดำเนินการเก็บข้อมูลรหัสของเส้นเชื่อมเข้าและออกตามลำดับ

```

1 <node name="Order" ordering="FIFO"
  xmi:id="I8cvgzKGAqACIwOo"
  xmi:type="uml:CentralBufferNode">
2   <xmi:Extension extender="Visual Paradigm for UML">
3     <model value="ObjectNode"/>
4     <mustIsolate xmi:value="false"/>
5     <leaf xmi:value="false"/>
6     <qualityScore value="-1"/>
7   </xmi:Extension>
8 </node>
9 <node name="Name" xmi:id="BAKvzgKGAqACIwQx"
  xmi:type="uml:CallBehaviorAction">
10  <xmi:Extension extender="Visual Paradigm for UML">
11    <mustIsolate xmi:value="false"/>
12    <leaf xmi:value="false"/>
13    <qualityScore value="-1"/>
14  </xmi:Extension>
15 </node>

```

ภาพที่ 15 ตัวอย่างแท็ก node ในเอกซ์เอ็มไอไฟล์จากเครื่องมือทดสอบ

3. แท็ก edge ให้ข้อมูลรายละเอียดของเส้นเชื่อมทั้งหมดของแผนภาพกิจกรรมได้แก่ รหัสเส้นเชื่อม ชื่อของเส้นเชื่อม ประเภทของเส้นเชื่อม โหนดต้นทางของเส้นเชื่อมและ โหนดปลายทางของเส้นเชื่อม ตัวอย่างแสดงได้ดังภาพที่ 16

```

1 <edge isLeaf="false" source="5k2ecAKAUIDIowMO"
  target="Y0avzqKGAqACIwRA" xmi:id="K0avzqKGAqACIwRC"
  xmi:type="uml:ControlFlow">
2   <xmi:Extension extender="Visual Paradigm for UML">
3     <qualityScore value="-1"/>
4   </xmi:Extension>
5 </edge>
6 <edge isLeaf="false" source="5k2ecAKAUIDIowMO"
  target="RN0vzqKGAqACIwRR" xmi:id="bNOvzqKGAqACIwRT"
  xmi:type="uml:ControlFlow">
7   <xmi:Extension extender="Visual Paradigm for UML">
8     <qualityScore value="-1"/>
9   </xmi:Extension>
10 </edge>

```

ภาพที่ 16 ตัวอย่างแท็ก edge ในเอกซ์เอ็มไอไฟล์จากเครื่องมือทดสอบ

3.3.2.1. กฎในการแปลงไฟล์เอกซ์เอ็มไอที่เวอร์ชันมากกว่า 1.4

กฎในการแปลงประกอบด้วยกัน 3 ข้อ ดังนี้

1. แท็กชื่อ packagedElement ให้ดำเนินการเก็บข้อมูลชื่อของแผนภาพกิจกรรม
2. แท็กชื่อ uml:Diagram ให้ดำเนินการเก็บข้อมูลชื่อของแผนภาพกิจกรรม
3. ถ้าแท็กเป็น node กำหนดให้สร้างโนดชั่วคราว เก็บรหัสของโนด และ เก็บข้อมูลประเภทของโนด และให้ดำเนินการต่อไปนี้
 - ถ้ามีแอททริบิวต์เส้นเชื่อมเข้า ภายในแอททริบิวต์จะเก็บข้อมูลรหัสของเส้นเชื่อมเข้า กรณีมีมากกว่า 1 เส้นจะแบ่งด้วยช่องว่าง 1 ช่อง

- ถ้ามีแอดทริบิวต์เส้นเชื่อมออก ภายในแอดทริบิวต์จะเก็บข้อมูลรหัสของเส้นเชื่อมออก กรณีมีมากกว่า 1 เส้นจะแบ่งด้วยช่องว่าง 1 ช่อง
 - กรณีไม่เจอเส้นเชื่อมเข้า หรือเส้นเชื่อมออก ให้อ่านแท็กชื่อ incoming และ outgoing ตามลำดับ ดำเนินการเก็บข้อมูลรหัสของเส้นเชื่อมเข้าและออกตามลำดับเข้าในดักข้อความ
 - ดำเนินการเพิ่มโนดข้อความ เข้าไปเก็บในรายการโนดของแผนภาพกิจกรรม
4. แท็กชื่อ edge ให้ดำเนินสร้างเส้นเชื่อมข้อความ เก็บรหัสของเส้นเชื่อม และ เก็บข้อมูลประเภทของเส้นเชื่อม และให้ดำเนินการต่อไปนี้
- กำหนดชื่อของเส้นเชื่อม (ถ้ามี)
 - กำหนดโนดต้นทาง
 - กำหนดโนดปลายทาง

ผลลัพธ์จากการใช้กฎความสัมพันธ์เพื่อสกัดชุดข้อมูล จะได้ชุดข้อมูลประกอบด้วย 1) รายละเอียดของแผนภาพกิจกรรม 2) ชุดข้อมูลโนด 3) ชุดข้อมูลเส้นเชื่อม จากนั้นจะนำข้อมูลดังกล่าวไปรูปแบบมาตรฐานที่ผู้ใช้กำหนดเพื่อนำออกเป็นไฟล์เอ็กซ์เอ็มไอตามรูปแบบมาตรฐานที่ผู้ใช้กำหนด

3.4 การสร้างรูปแบบมาตรฐานที่ผู้ใช้งานกำหนด

ขั้นตอนการสร้างไฟล์นำออกเอ็กซ์เอ็มไอตามรูปแบบมาตรฐานที่ผู้ใช้งานกำหนด โดยผู้วิจัยได้กำหนดเอ็กซ์เอ็มแอลสคีมา เพื่อเป็นรูปแบบมาตรฐานสำหรับสร้างไฟล์เอ็กซ์เอ็มไอที่สามารถใช้งานได้กับงานวิจัย [2] และทำการเชื่อมต่อชุดข้อมูลโนดและชุดข้อมูลเส้นเชื่อมกับรูปแบบมาตรฐานก่อนนำออกเป็นไฟล์เอ็กซ์เอ็มไอ โดยกระบวนการทำงานในขั้นตอนนี้ แบ่งออกเป็น 3 ขั้นตอน ดังนี้

1. การเตรียมข้อมูลแม่แบบรูปแบบมาตรฐานสำหรับสร้างไฟล์เอ็กซ์เอ็มไอ โดยจะต้องทำการเตรียมแม่แบบให้อยู่ในรูปของเอ็กซ์เอ็มแอลสคีมา ดังแสดงในภาพที่ 17

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
   attributeFormDefault="unqualified">
3   <!-- XML Schema Generated from XML Document on Wed Sep 4 13:05:41 UTC+0700 2013 -->
4   <!-- with XmlGrid.net Free Online Service http://xmlgrid.net -->
5   <xs:element name="umlModel">
6     <xs:complexType>
7       <xs:sequence>
8         <xs:element name="packagedElement" maxOccurs="unbounded">
9           <xs:complexType>
10            <xs:sequence>
11              <xs:element name="ownedParameter" maxOccurs="unbounded">
12                <xs:complexType base="umlModelParameter" />
13              </xs:element>
14              <xs:element name="node" maxOccurs="unbounded">
15                <xs:complexType base="umlModelNode" />
16              </xs:element>
17              <xs:element name="edge" maxOccurs="unbounded">
18                <xs:complexType base="umlModelEdge" />
19              </xs:element>
20            </xs:sequence>
21            <xs:attribute name="xmitype" type="xs:string"/>
22            <xs:attribute name="xmIID" type="xs:string"/>
23            <xs:attribute name="name" type="xs:string"/>
24          </xs:complexType>
25        </xs:element>
26      </xs:sequence>
27      <xs:attribute name="xmINameuml" type="xs:string"/>
28      <xs:attribute name="xmINamexmi" type="xs:string"/>
29      <xs:attribute name="xmIVersion" type="xs:double"/>
30      <xs:attribute name="xmIID" type="xs:string"/>
31      <xs:attribute name="name" type="xs:string"/>
32    </xs:complexType>
33  </xs:element>
34 </xs:schema>

```

ภาพที่ 17 ตัวอย่างเอ็กซ์เอ็มแอลสคีมาที่ใช้กำหนดรูปแบบการนำออกไฟล์เอ็กซ์เอ็มไอ

2. การแปลงเอ็กซ์เอ็มแอลสคีมาให้อยู่ในรูปแบบของ จาวาคลาส (Java Class)

ผลลัพธ์ที่ได้จากขั้นตอนนี้ คือ

- 2.1. คลาสแพคทอรี เพื่อใช้สำหรับการสร้างและจัดการกับ คลาสยูเอ็มแอลโมเดล
- 2.2. คลาสยูเอ็มแอลโมเดลที่มีความสัมพันธ์ตามรูปแบบของเอ็กซ์เอ็มแอลสคีมา ดังแสดงในภาพที่18

ผลลัพธ์จากการผ่านตัวแปลงแผนภาพกิจกรรมจะได้ไฟล์เอ็กซ์เอ็มไอตามรูปแบบมาตรฐานที่ผู้ใช้กำหนด ดังแสดงในภาพที่ 19


```

package th.ac.chula.eng.cp.xmitransforms.handlers;

import java.util.ArrayList;

/* <p>
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "", propOrder = { "packagedElement" })
@XmlRootElement(name = "uml:Model")
public class UmlModel {

    private static final PackagedElement packagedElement = null;
    @XmlElement(required = true)
    protected List<UmlModel.PackagedElement> packagedElement;
    @XmlAttribute(name = "xmlns:uml")
    protected String xmlnsuml;
    @XmlAttribute(name = "xmlns:xmi")
    protected String xmlnsxmi;
    @XmlAttribute(name = "xmi:version")
    protected String xmiVersion;
    @XmlAttribute(name = "xmi:id")
    protected String xmiId;
    @XmlAttribute(name = "name")
    protected String name;

    public UmlModel() {

    }

    * Gets the value of the packagedElement property.
    public List<UmlModel.PackagedElement> getPackagedElement() {
        if (packagedElement == null) {
            packagedElement = new ArrayList<UmlModel.PackagedElement>();
            // getPackagedElement().add(null);
        }
        return this.packagedElement;
    }

    <p>
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "", propOrder = { "ownedParameter", "node", "edge" })
public static class PackagedElement {

    @XmlElement(required = true)
    protected List<UmlModel.PackagedElement.OwnedParameter> ownedParameter;
    @XmlElement(required = true)
    protected List<UmlModel.PackagedElement.Node> node;
    @XmlElement(required = true)
    protected List<UmlModel.PackagedElement.Edge> edge;
    @XmlAttribute(name = "xmi:type")
    protected String xmiType;
    @XmlAttribute(name = "xmi:id")
    protected String xmiId;
    @XmlAttribute(name = "name")
    protected String name;

    * Gets the value of the ownedParameter property.
    public List<UmlModel.PackagedElement.OwnedParameter> getOwnedParameter() {
        if (ownedParameter == null) {
            ownedParameter = new ArrayList<UmlModel.PackagedElement.OwnedParameter>();
            // writing to be here
            // ownedParameter = new ArrayList<XMIData.XMINode>();
        }
        return this.ownedParameter;
    }

    * Gets the value of the node property.
    public List<UmlModel.PackagedElement.Node> getNode() {
        if (node == null) {
            node = new ArrayList<UmlModel.PackagedElement.Node>();
        }
        return this.node;
    }

    * Gets the value of the edge property.
    public List<UmlModel.PackagedElement.Edge> getEdge() {
        if (edge == null) {
            edge = new ArrayList<UmlModel.PackagedElement.Edge>();
        }
        return this.edge;
    }
}
}

```

ภาพที่ 18 ตัวอย่างคลาสที่มีความสัมพันธ์ตามรูปแบบของเอ็กซ์เอ็มแอลสคีมาที่กำหนดใน

ข้อ 1

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <uml:Model xmlns:uml="http://www.omg.org/spec/UML/20090901"
  xmlns:xmi="http://schema.omg.org/spec/XMI/2.1" name="Activity Diagrams"
  xmi:id="_wNqo0qFzEd6Eo8cPzRliQg" xmi:version="2.1">
3 <packagedElement name="ex2" xmi:id="_wNqo6aFzEd6Eo8cPzRliQg" xmi:type="uml:Activity">
4 <node name="Fill Order" xmi:id="_wNnY5y2DEeKSKePaylnoVA"
  xmi:type="uml:OpaqueAction">
5 <result name="Call10.output0" outgoing="_wNnY7y2DEeKSKePaylnoVA"
  xmi:id="_wNnY7y2DEeKSKePaylnoVA" xmi:type="uml:OutputPin">
6 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
7 </result>
8 </node>
9 <node name="Ship Order" xmi:id="_wNnY6C2DEeKSKePaylnoVA"
  xmi:type="uml:OpaqueAction">
10 <argument incoming="_wNnY8S2DEeKSKePaylnoVA" name="Call11.input0"
  xmi:id="_wNnY8S2DEeKSKePaylnoVA" xmi:type="uml:InputPin">
11 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
12 </argument>
13 </node>
14 <node name="Send Invoice" xmi:id="_wNnY6S2DEeKSKePaylnoVA"
  xmi:type="uml:OpaqueAction">
15 <argument incoming="_wNnY8y2DEeKSKePaylnoVA" name="Call12.input0"
  xmi:id="_wNnY8y2DEeKSKePaylnoVA" xmi:type="uml:InputPin">
16 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
17 </argument>
18 </node>
19 <node name="Fork" xmi:id="_wNnY6i2DEeKSKePaylnoVA" xmi:type="uml:ForkNode">
20 <argument incoming="_wNnY7y2DEeKSKePaylnoVA" name="Call13.input0"
  xmi:id="_wNnY7y2DEeKSKePaylnoVA" xmi:type="uml:InputPin">
21 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
22 </argument>
23 <result name="Call13.output0" outgoing="_wNnY8S2DEeKSKePaylnoVA"
  xmi:id="_wNnY8S2DEeKSKePaylnoVA" xmi:type="uml:OutputPin">
24 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
25 </result>
26 <result name="Call13.output1" outgoing="_wNnY8y2DEeKSKePaylnoVA"
  xmi:id="_wNnY8y2DEeKSKePaylnoVA" xmi:type="uml:OutputPin">
27 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
28 </result>
29 </node>
30 <edge source="_wNnY5y2DEeKSKePaylnoVA" target="_wNnY6i2DEeKSKePaylnoVA"
  xmi:id="_wNnY7y2DEeKSKePaylnoVA" xmi:type="uml:ControlFlow">
31 <weight value="1" xmi:id="_wNqpc34zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
32 </edge>
33 <edge source="_wNnY6i2DEeKSKePaylnoVA" target="_wNnY6C2DEeKSKePaylnoVA"
  xmi:id="_wNnY8S2DEeKSKePaylnoVA" xmi:type="uml:ControlFlow">
34 <weight value="1" xmi:id="_wNqpc92zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
35 </edge>
36 <edge source="_wNnY6i2DEeKSKePaylnoVA" target="_wNnY6S2DEeKSKePaylnoVA"
  xmi:id="_wNnY8y2DEeKSKePaylnoVA" xmi:type="uml:ControlFlow">
37 <weight value="1" xmi:id="_wNqpc3zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
38 </edge>
39 </packagedElement>
40 </uml:Model>

```

ภาพที่ 19 ตัวอย่างผลลัพธ์ไฟล์เอ็กซ์เอ็มไอ จากเครื่องมือตัวแปลงแผนภาพกิจกรรม

บทที่ 4

การออกแบบและพัฒนาระบบ

4.1 สถาปัตยกรรมระบบ

ระบบที่พัฒนาขึ้น ประกอบด้วยขั้นตอนการทำงานหลัก 3 ขั้นตอน ดังนี้

1) ขั้นตอนตรวจสอบความถูกต้องของไฟล์เอ็กซ์เอ็มไอ กล่าวคือ ตรวจสอบว่าไฟล์นำเข้า อยู่ในรูปแบบมาตรฐานเอ็กซ์เอ็มไอที่กำหนดโดยโอเอ็มจีหรือไม่ รวมทั้งทำการตรวจสอบเวอร์ชันของไฟล์เอ็กซ์เอ็มไอจากเมตาเดตา

ในงานวิจัยนี้รองรับการตรวจหารูปแบบไฟล์เอ็กซ์เอ็มไอแบ่งเป็น 2 ประเภทตามเวอร์ชัน กล่าวคือ 1.รูปแบบเอ็กซ์เอ็มไอที่เวอร์ชัน 1.4 และ 2.รูปแบบเอ็กซ์เอ็มไอที่เวอร์ชันมากกว่า 1.4

2) ขั้นตอนแจ่งส่วนเอ็กซ์เอ็มไอ จะทำการสกัดข้อมูลจากไฟล์เอ็กซ์เอ็มไอโดยดูจากเวอร์ชันของไฟล์เอ็กซ์เอ็มไอที่ได้จากเมตาเดตาในขั้นตอนแรก เพื่อกำหนดรูปแบบการเข้าถึงข้อมูล

เนื่องจากมาตรฐานเอ็กซ์เอ็มไอที่ถูกกำหนดขึ้นโดยโอเอ็มจีมีการปรับปรุงมาตรฐานอย่างต่อเนื่อง แต่การเปลี่ยนแปลงครั้งสำคัญสำหรับมาตรฐานของเอ็กซ์เอ็มไอคือ การปรับเวอร์ชันหลังจาก 1.4 เป็นต้นไป กล่าวคือโครงสร้างของเอ็กซ์เอ็มไอที่กำหนดไว้เพื่อเข้าถึงชุดข้อมูลโนดและชุดข้อมูลเส้นเชื่อมสำหรับเวอร์ชันที่ 1.4 จะถูกกำหนดโดยชื่อของแท็ก เช่น <UML:ActionState /> แทนการเข้าถึงชุดข้อมูลโนด และ <UML:Transition /> แทนการเข้าถึงชุดข้อมูลเส้นเชื่อมสำหรับโครงสร้างของเอ็กซ์เอ็มไอที่เวอร์ชันมากกว่า 1.4 การเข้าถึงชุดข้อมูลโนดและชุดข้อมูลเส้นเชื่อมจะเปลี่ยนเป็นแท็กที่ชื่อว่า <node /> และ <edge /> แทน

3) ขั้นตอนการสร้างไฟล์นำออกเอ็กซ์เอ็มไอตามรูปแบบมาตรฐานที่ผู้ใช้งานกำหนด โดยผู้วิจัยได้กำหนดเอ็กซ์เอ็มแอลสคีม่า ดังรูปที่ 5 เพื่อเป็นรูปแบบมาตรฐานสำหรับสร้างไฟล์เอ็กซ์เอ็มไอที่สามารถใช้งานได้กับงานวิจัย [2] และทำการเชื่อมต่อชุดข้อมูลโนดและชุดข้อมูลเส้นเชื่อมกับรูปแบบมาตรฐานก่อนนำออกเป็นไฟล์เอ็กซ์เอ็มไอกลุ่มวัตถุข้อความหรือกลุ่มคำ โดยมีคุณสมบัติ ดังนี้

4.2 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา

สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนาระบบประกอบด้วยรายการฮาร์ดแวร์และซอฟต์แวร์ดังต่อไปนี้

4.2.1 สภาพแวดล้อม

1. คอมพิวเตอร์เอชพี รุ่น ProBook 6470b ปี 2013

2. หน่วยประมวลผลอินเทล คอร์ไอไฟว์
3. หน่วยความจำขนาด 6 กิกะไบต์
4. ฮาร์ดดิสก์ไดรฟ์ขนาดความจุ 500 กิกะไบต์
5. ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 7 แบบ 64 บิต

4.2.2 เครื่องมือที่ใช้ในการพัฒนา

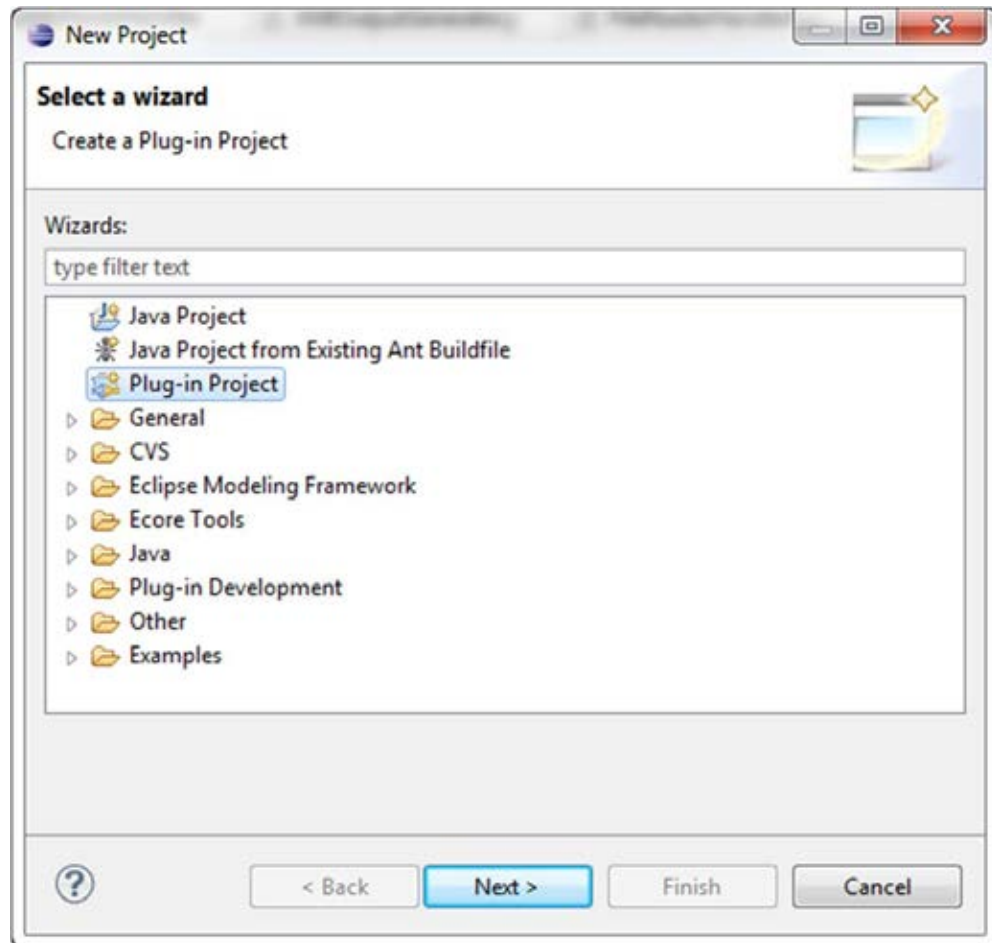
1. ArgoUML Version 0.34
2. Visual paradigm Version 10.0
3. Modelio Version 2.2.1
4. Altova Umodel Enterprise Ed. 2013 rel. 2 sp1
5. Rhapsody Version 8.03
6. อีคลิปส์ 3.7.2
7. ชุดเครื่องมือพัฒนาจาวา 7
8. อีคลิปส์โมเดลลิงทูล 1.4.2

4.3 การพัฒนาระบบ

การพัฒนาระบบในงานวิจัยจะถูกพัฒนาโดยซอฟต์แวร์อีคลิปส์โมเดลลิงทูลเป็นเครื่องมือหลัก ระบบที่ได้จะอยู่ในรูปแบบของปลั๊กอินอีคลิปส์ ที่แต่ละส่วนของการทำงาน (Workspace) จะแยกออกจากกันอย่างชัดเจนตามแนวคิดของส่วนประกอบร่วม

4.3.1 การสร้างโปรเจกต์ปลั๊กอินอีคลิปส์

การสร้างโปรเจกต์เริ่มจากเราทำการเลือกสร้างโปรเจกต์ใหม่ แล้วเลือกสร้างปลั๊กอินโปรเจกต์ ดังภาพที่ 20



ภาพที่ 20 การสร้างปลั๊กอิน-โปรเจกต์

ขั้นตอนต่อไปเป็นการกำหนดรายละเอียดปลั๊กอินโปรเจกต์ที่เราสร้างขึ้น ได้แก่ กำหนดชื่อของโปรเจกต์และรายละเอียดสนับสนุนของอีคลิปส์ กำหนดคุณสมบัติของปลั๊กอินและกำหนดเทมเพลตของปลั๊กอิน ดังแสดงในภาพที่ 21, 22 และ 23 ตามลำดับ

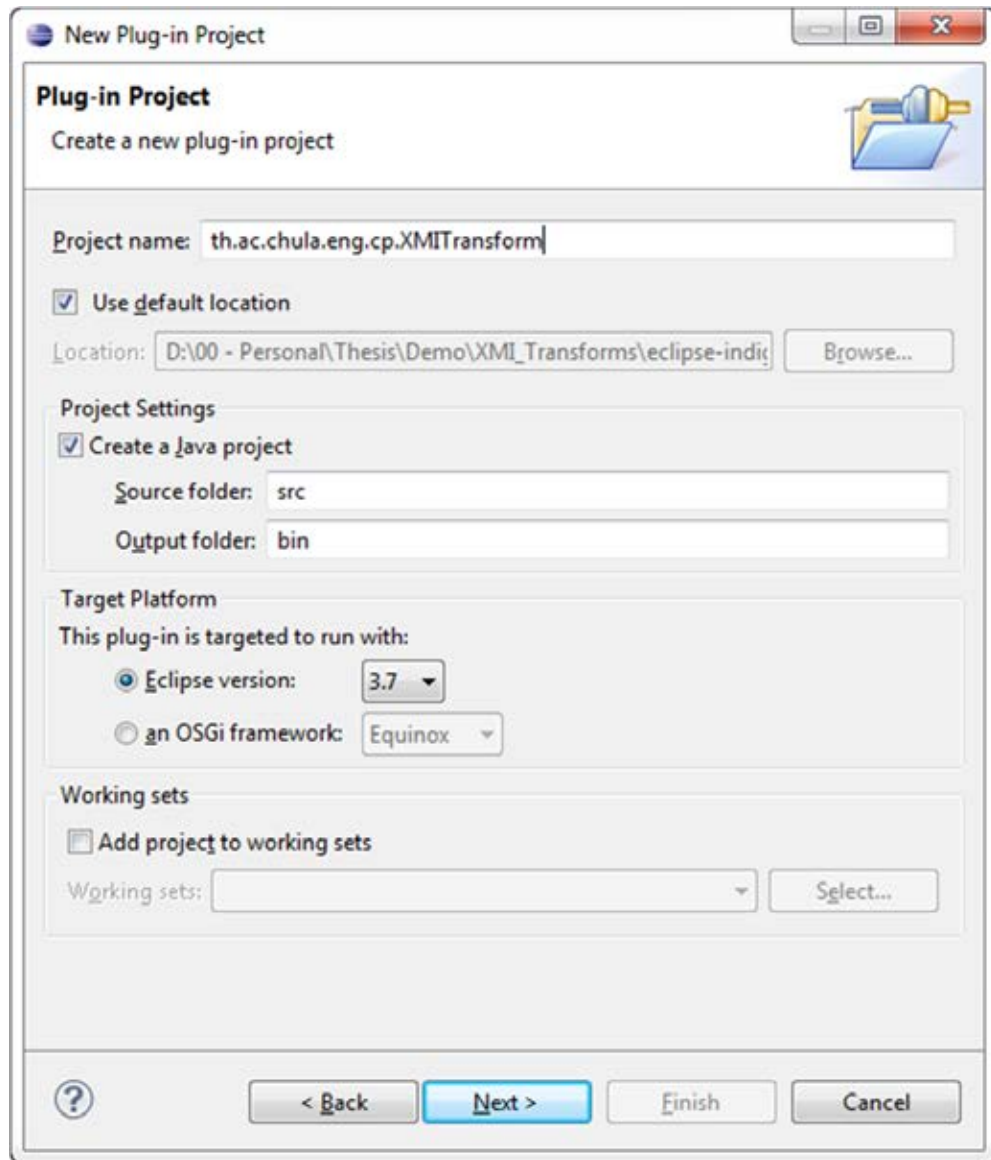
การกำหนดชื่อของโปรเจกต์และรายละเอียดการสนับสนุนของอีคลิปส์ ประกอบไปด้วยรายละเอียดที่สำคัญคือ กำหนดสถานที่สำหรับจัดเก็บไฟล์ต้นทาง กำหนดสถานที่สำหรับจัดเก็บไฟล์ผลลัพธ์ ระบุเวอร์ชันที่ต้องการให้ปลั๊กอินรองรับที่เวอร์ชัน 3.7 ซึ่งเป็นเวอร์ชันที่ผู้วิจัยใช้ในการพัฒนา

การกำหนดคุณสมบัติของปลั๊กอิน ประกอบไปด้วยรายละเอียดที่สำคัญคือ การกำหนดค่าต่าง ๆ ซึ่งได้แก่ รหัส เวอร์ชัน ชื่อของปลั๊กอิน ชื่อผู้พัฒนาปลั๊กอินและเวอร์ชันของจาวาที่ต้องการใช้ในการทำงานกับปลั๊กอิน

การกำหนดเทมเพลตของปลั๊กอิน เป็นการเลือกแม่แบบในการสร้างปลั๊กอินสำเร็จรูป แต่

ผู้วิจัยไม่ได้พัฒนางานวิจัยนี้ด้วยรูปแบบสำเร็จรูป

เสร็จจากขั้นตอนการสร้างโปรเจกต์ปลั๊กอินแล้ว หลังจากนั้นทำการกำหนดรายละเอียดไฟล์เมนิเฟสของปลั๊กอิน และสร้างไฟล์คลาสต่าง ๆ ที่เกี่ยวข้องด้วยภาษาจาวา การสร้างไอคอนสำหรับการรันผ่านเมนู



ภาพที่ 21 กำหนดรายละเอียดและการตั้งค่าสำหรับโปรเจกต์ใหม่

New Plug-in Project

Content
Enter the data required to generate the plug-in.

Properties

ID:

Version:

Name:

Provider:

Execution Environment:

Options

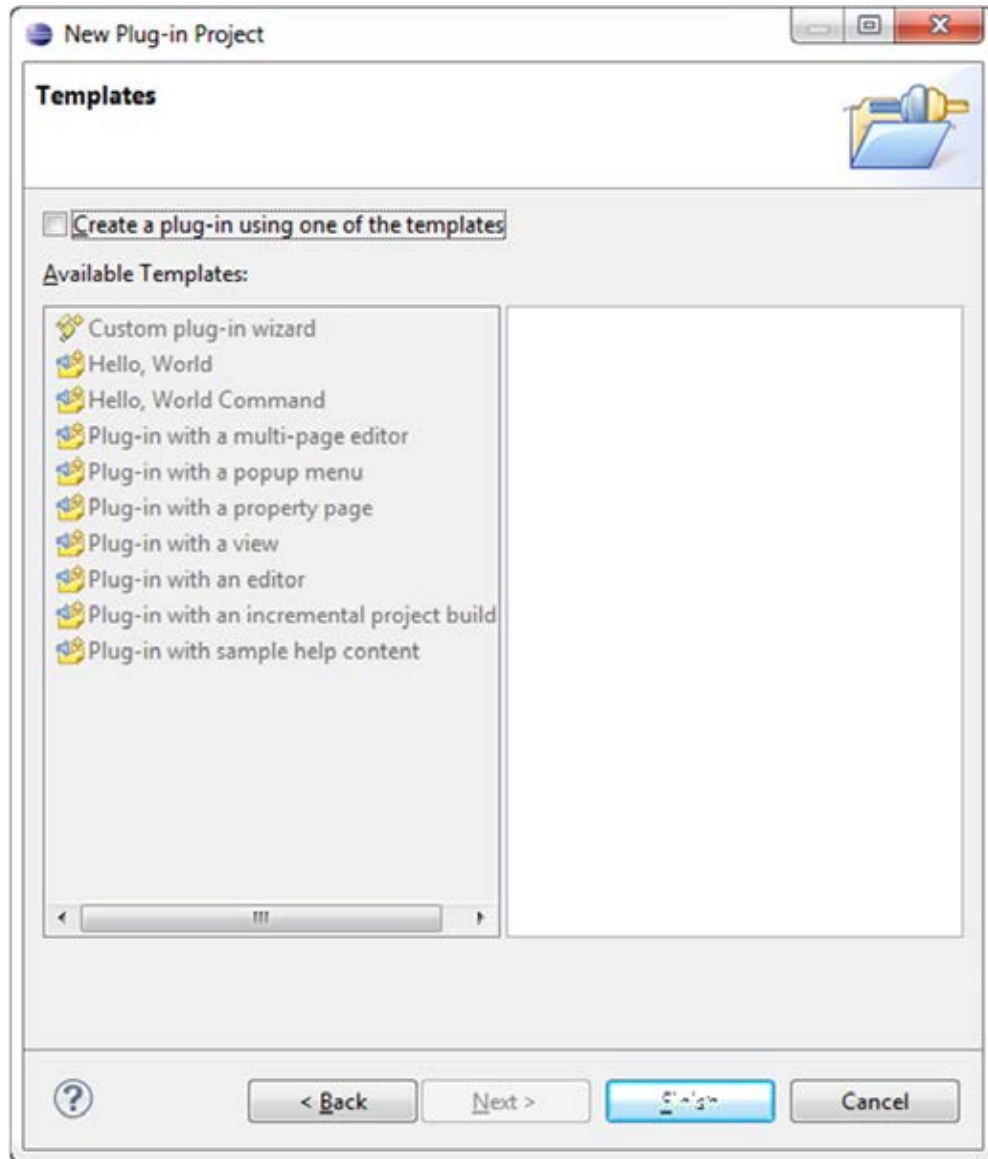
Generate an activator, a Java class that controls the plug-in's life cycle
Activator:

This plug-in will make contributions to the UI

Enable API Analysis

Rich Client Application
Would you like to create a rich client application? Yes No

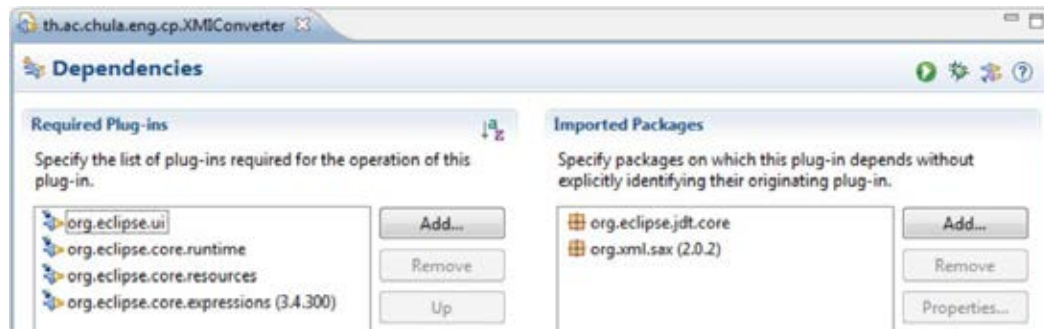
ภาพที่ 22 กำหนดรายละเอียดปลั๊กอิน



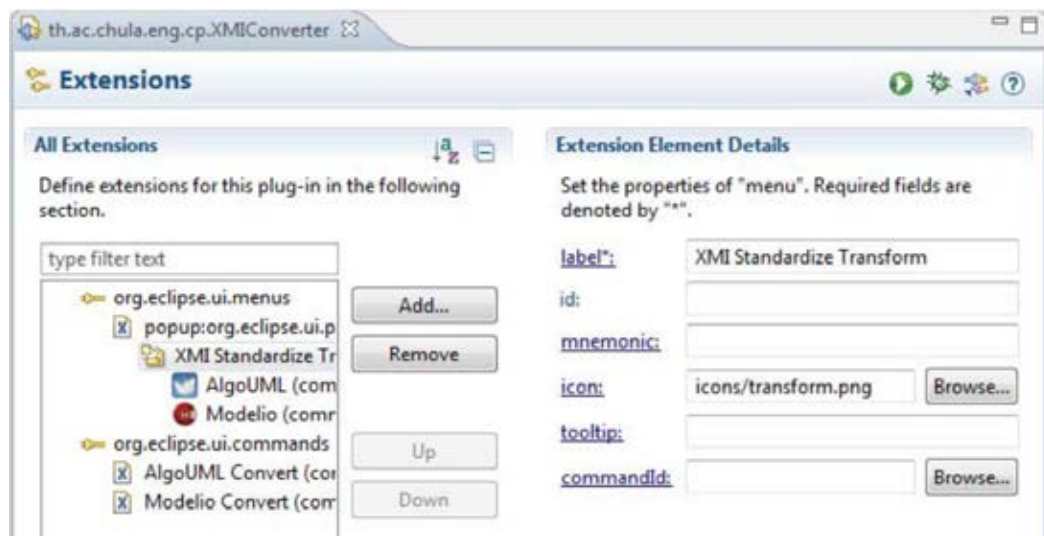
ภาพที่ 23 การกำหนดรายละเอียดเทมเพลต

4.3.2 การตรวจสอบความถูกต้องของไฟล์เอ็กซ์เอ็มไอ

การพัฒนาส่วนการตรวจสอบความถูกต้องของไฟล์เอ็กซ์เอ็มไอ เริ่มจากสร้างโปรเจกต์ ปลั๊กอินอีคลิปส์ขึ้นมาใหม่ กำหนดรายละเอียดตามข้อ 4.3.1 จากนั้นทำการกำหนดรายละเอียด ไฟล์เมนิเฟส ใน 2 ส่วน ได้แก่ ส่วน Dependencies และส่วนExtensions เพื่อสร้างเมนูสำหรับให้ ผู้ใช้งานเรียกใช้งานตัวแปลงแผนภาพกิจกรรม ดังแสดงในรูปที่ 24 และ 25 ตามลำดับ อ่าน รายละเอียดเพิ่มเติม ที่ภาคผนวก ก



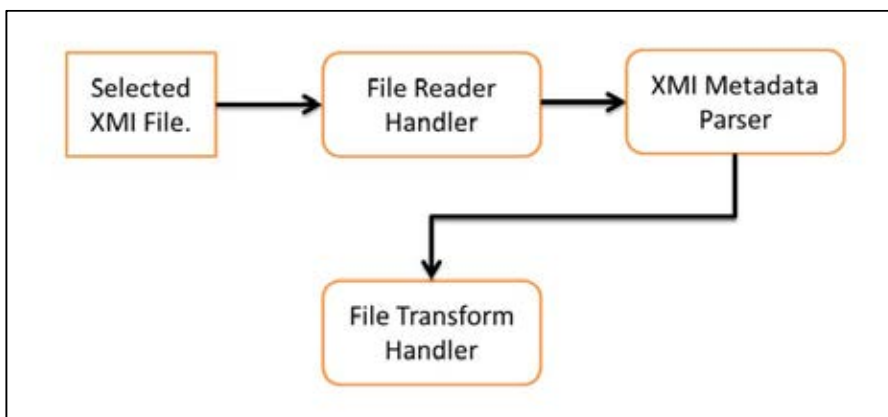
ภาพที่ 24 กำหนดรายละเอียดหน้า Dependencies ของปลั๊กอิน



ภาพที่ 25 กำหนดรายละเอียด Extensions สำหรับปลั๊กอิน

หลังจากกำหนดค่าเบื้องต้นต่างๆ ในไฟล์เมนิเฟสเรียบร้อยแล้ว ขั้นตอนต่อไปจะต้องสร้างคลาสไฟล์เพื่อจัดการกับไฟล์เอ็กซ์เอ็มไอนำเข้า โดยแต่ละคลาสจะมีหน้าที่ในการทำงานแตกต่างกัน และความสัมพันธ์ของแต่ละคลาสแสดงได้ดังภาพที่ 26 ซึ่งแต่ละคลาสจะมีหน้าที่ดังนี้

- FileReaderHandlerForXML2Output เป็นคลาสเริ่มต้นเมื่อมีคำสั่งให้ตรวจสอบความถูกต้องของไฟล์เอ็กซ์เอ็มไอที่ถูกเลือกและเรียกคลาสการตรวจสอบเมตาเดตาขึ้นมาทำงาน
- XMIMetadataParser ทำหน้าที่ในการตรวจสอบเมตาเดตาของไฟล์เอ็กซ์เอ็มไอและจัดเก็บเวอร์ชันของไฟล์เอ็กซ์เอ็มไอและเรียกคลาสการแจงส่วนเอกสารเอ็กซ์เอ็มไอขึ้นมาทำงาน

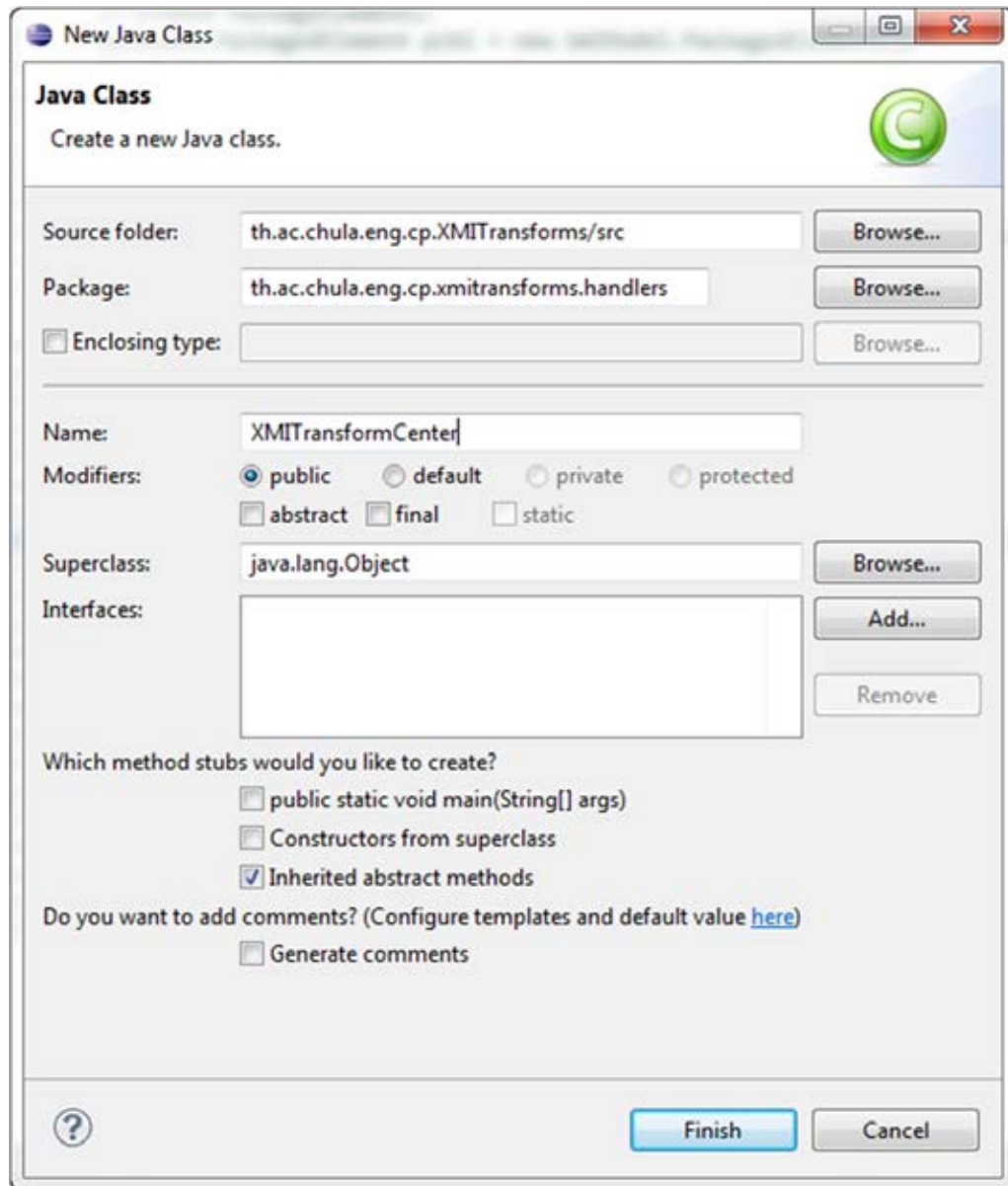


ภาพที่ 26 ภาพการทำงานการตรวจสอบความถูกต้องของไฟล์เอ็กซ์เอ็มไอ

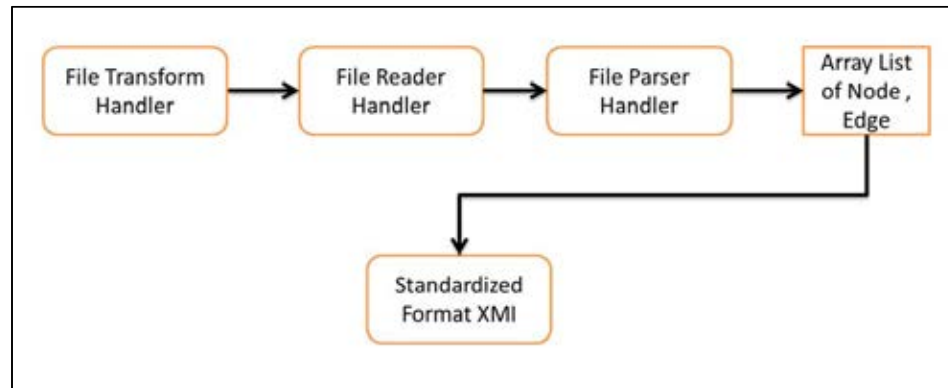
4.3.3 การแจงส่วนเอ็กซ์เอ็มไอ

การพัฒนาส่วนการแจงส่วนเอ็กซ์เอ็มไอ ทำโดยสร้างคลาสไฟล์ต่าง ๆ เพื่อใช้แจงส่วนไฟล์เอ็กซ์เอ็มไอที่ได้จากขั้นตอนแรก สำหรับคลาสที่สำคัญและมีหน้าที่ในการแจงส่วนไฟล์เอ็กซ์เอ็มไอ มีดังนี้

- XMITransformsCenter เป็นคลาสที่ใช้ในการจัดการการทำงานตามรูปแบบของเอ็กซ์เอ็มไอเวอร์ชันที่ได้จากขั้นตอนแรก และเรียกคลาสที่เกี่ยวข้องเพื่อแจงส่วนข้อมูลขึ้นมาทำงาน
- XMIDataParser หน้าที่หลักคือทำการแจงส่วนไฟล์เอ็กซ์เอ็มไอที่เลือกโดยดูจากเวอร์ชันของไฟล์เอ็กซ์เอ็มไอในข้อ 4.3.1 เพื่อเลือกวิธีเข้าถึงข้อมูล หลังจากนั้นทำการเก็บข้อมูลที่แจงส่วนเรียบร้อยไว้ในอาเรย์ลิสของออกพเจค และมีหน้าที่เรียกและส่งผ่านข้อมูลให้กับ XMIOutputGenerator และนำผลลัพธ์ที่ได้ที่จาก XMIOutputGenerator ส่งไปให้คลาสการสร้างรูปแบบมาตรฐานที่ผู้ใช้งานกำหนดเพื่อทำงาน



ภาพที่ 27 การสร้างคลาสเพื่อใช้ในการแจงส่วนเอ็กซ์เอ็มไอ

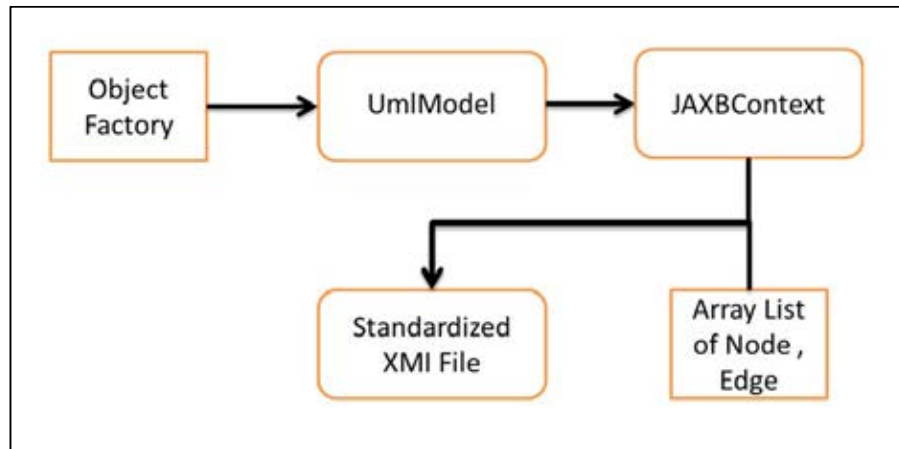


ภาพที่ 28 ภาพการทำงานการแจงส่วนเอ็กซ์เอ็มไอ

4.3.4 การสร้างรูปแบบมาตรฐานที่ผู้ใช้งานกำหนด

การพัฒนาการสร้างรูปแบบมาตรฐาน ทำโดยการนำเอ็กซ์เอ็มแอลสคีมามาที่ผู้ใช้งานกำหนด ขึ้น ทำการสร้างคลาสจากเอ็กซ์เอ็มแอลสคีม่า และทำการเชื่อมต่อชุดข้อมูลโนดและชุดข้อมูลเส้น เชื่อมกับรูปแบบมาตรฐานก่อนนำออกเป็นไฟล์เอ็กซ์เอ็มไอ โดยกระบวนการทำงานในขั้นตอนนี้จะเป็นการเขียนคลาสไฟล์ต่าง ๆ ซึ่งแต่ละคลาสจะมีหน้าที่ต่างกันดังนี้

- ทำหน้าที่ในการสร้างออปเจครูปแบบมาตรฐานและกำหนดค่ารายละเอียดแผนภาพ ชุดข้อมูลโนด และชุดข้อมูลเส้นเชื่อมให้กับ UmlModel
- UmlModel ทำหน้าที่เป็นแม่แบบของไฟล์เอ็กซ์เอ็มไอที่จะนำออกจากระบบ โดยคลาส UmlModel จะประกอบไปด้วยคลาสย่อยเพื่อใช้ในการเก็บข้อมูลที่ได้จากการแจงส่วนไฟล์เอ็กซ์เอ็มไอ
- JAXBContext ทำหน้าที่ในการแปลงออปเจค UmlModel หลังจากทำการเชื่อมต่อชุดข้อมูลโนดและชุดข้อมูลเส้นเชื่อม เพื่อนำออกเป็นไฟล์เอ็กซ์เอ็มไอ



ภาพที่ 29 ภาพการทำงานการสร้างรูปแบบมาตรฐานที่ผู้ใช้งานกำหนด

บทที่ 5 การประเมินและการวัดผล

5.1 แนวทางการประเมินผลงานวิจัย

การประเมินผลของงานวิจัยนี้ ได้ทำการทดลองและวิเคราะห์การแลกเปลี่ยนข้อมูลในตารางที่ 3 พบว่ามีบางเครื่องมือที่สามารถแลกเปลี่ยนข้อมูลกันได้ หากแต่เป็นส่วนน้อยเท่านั้น โดยส่วนใหญ่จะยังคงไม่สามารถแลกเปลี่ยนข้อมูลกันได้ ถึงแม้ข้อมูลจะอยู่ในรูปแบบเอ็กซ์เอ็มไอ

รูปแบบของการสูญหายเกิดขึ้นจากข้อมูลของเอ็กซ์เอ็มไอที่แตกต่างกันตามแต่ละเครื่องมือทดสอบ ซึ่งเมื่อนำไฟล์เอ็กซ์เอ็มไอที่ได้จากเครื่องมือทดสอบแต่ละเครื่องมือมาทำการแปลงผ่าน ตัวแปลงแผนภาพกิจกรรมสำหรับรูปแบบมาตรฐานเอ็กซ์เอ็มไอ ทำให้ได้ผลลัพธ์คือไฟล์เอ็กซ์เอ็มไอตามรูปแบบที่กำหนดและสามารถแลกเปลี่ยนข้อมูลกันได้ทั้งหมดระหว่างเครื่องมือทดสอบทั้ง 5 ประเภท

ตารางที่ 3 ผลการแลกเปลี่ยนข้อมูลเอ็กซ์เอ็มไอระหว่างเครื่องมือสร้างแผนภาพกิจกรรมยูเอ็มแอลที่ใช้ในการวิจัย

Import \ Export	Agro UML	Visual paradigm	Modelio	Umodel	Rhapsody
AgroUML	Y	N	N	N1	N1
Visual paradigm	N	Y	N	Y	N1
Modelio	N	N	Y	N3	Y
Umodel	N	N2	N	Y	N3
Rhapsody	N	N	N1	N4	Y

จากตารางที่ 3 สามารถแยกประเภทของผลลัพธ์จากการแลกเปลี่ยนข้อมูลได้เป็น 6 ประเภท ดังนี้

Y คือ ข้อมูลสามารถแลกเปลี่ยนกันได้โดยสมบูรณ์

N1 คือ ไม่สามารถระบุประเภทของเอกสารได้

N2 คือ ข้อมูลบางส่วนของโนดและเส้นเชื่อมสูญหายระหว่างแลกเปลี่ยนข้อมูล

N3 คือ ข้อมูลบางส่วนของเส้นเชื่อมสูญหายระหว่างแลกเปลี่ยนข้อมูล

N4 คือ ข้อมูลเส้นเชื่อมสูญหายทั้งหมดระหว่างแลกเปลี่ยนข้อมูล

N คือ ไม่สามารถแลกเปลี่ยนข้อมูลใดๆ ได้

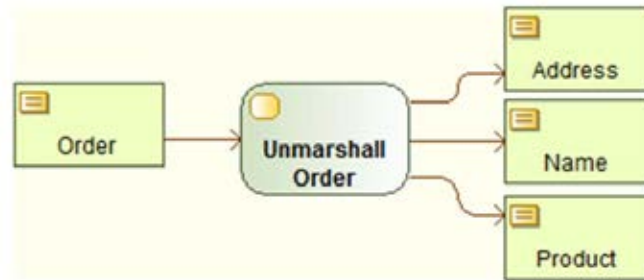
ด้วยหลักการและวิธีการที่นำเสนอในงานวิจัยนี้ ตัวแปลงแผนภาพกิจกรรมสำหรับรูปแบบมาตรฐานเอ็กซ์เอ็มไอที่พัฒนาขึ้น สามารถรองรับการแปลงไฟล์เอ็กซ์เอ็มไอที่สร้างจากเครื่องมือยูเอ็มแอลชนิดอื่น ๆ ได้นอกจากเครื่องมือที่นำมาทดสอบ แต่อยู่ภายใต้เงื่อนไขของไฟล์เอ็กซ์เอ็มไอที่นำเข้าระบบจะต้องเป็นไปตามมาตรฐานที่ไอเอ็มจีกำหนด [3]

การสรุปผลนั้นจะพิจารณาจากการเปรียบเทียบจากไฟล์เอ็กซ์เอ็มไอที่ได้จากเครื่องมือทดสอบ 5 ชนิด แปลงผ่านตัวแปลงแผนภาพกิจกรรม นำไฟล์เอ็กซ์เอ็มไอที่เป็นผลลัพธ์จากตัวแปลงแผนภาพกิจกรรมสำหรับรูปแบบมาตรฐานเอ็กซ์เอ็มไอไปเป็นข้อมูลนำเข้ากับเครื่องมือตรวจทานแผนภาพกิจกรรมในงานวิจัย[2] ผลลัพธ์ที่ได้จากตัวแปลงแผนภาพกิจกรรมจะต้องผ่านขั้นตอนการตรวจทานด้วยเครื่องมือในงานวิจัย [2] ได้

5.2 การทดสอบตัวอย่างที่ 1

5.2.1 ข้อมูลนำเข้าจากซอฟต์แวร์ Modelio Version 2.2.1

ในตัวอย่างทดสอบที่ 3 นั้น ใช้ชุดแผนภาพกิจกรรมทดสอบจากเครื่องมือ Modelio Version 2.2.1 แสดงให้เห็นถึงแอ็กชันอินพุตและแอ็กชันเอาต์พุต ตามรูปแบบพื้นฐานแผนภาพกิจกรรมทั่วไป ดังแสดงในภาพที่ 30



ภาพที่ 30 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ

หลังจากสร้างแผนภาพกิจกรรมบนเครื่องมือ Modelio Version 2.2.1 แล้ว ให้นำออกข้อมูลในรูปแบบเอกซ์เอ็มไอไฟล์เพื่อเป็นอินพุตสำหรับเครื่องมือตัวแปลงแผนภาพกิจกรรม ตัวอย่างแสดงในภาพที่ 31

```

1 <?xml version="1.0" encoding="UTF-8"?><uml:Model
  xmlns:uml="http://www.omg.org/spec/UML/20090901"
  xmlns:xmi="http://schema.omg.org/spec/XMI/2.1" xmi:version="2.1"
  xmi:id="_mhxQICICEeKSKePaylnoVA" name="ex1">
2 <eAnnotations xmi:id="_mhyeQCICEeKSKePaylnoVA" source="Objing">
10 </eAnnotations>
11 <packagedElement xmi:type="uml:Activity" xmi:id="_mhyeRiICEeKSKePaylnoVA" name="ex1">
12 <nestedClassifier xmi:type="uml:Collaboration" xmi:id="_mhyeRyICEeKSKePaylnoVA"
  name="locals">
24 </nestedClassifier>
25 <node xmi:type="uml:OpaqueAction" xmi:id="_mhZFWSICEeKSKePaylnoVA" name="Unmarshall Order"
  outgoing="_mhZFbiICEeKSKePaylnoVA" _mhZsYyICEeKSKePaylnoVA" _mhZsaCICEeKSKePaylnoVA"
  incoming="_mhZsbSICEeKSKePaylnoVA">
26 <body/>
27 </node>
28 <node xmi:type="uml:CentralBufferNode" xmi:id="_mhZFWiICEeKSKePaylnoVA" name="Order"
  outgoing="_mhZsbSICEeKSKePaylnoVA" isControlType="true">
29 <eAnnotations xmi:id="_mhZFWyICEeKSKePaylnoVA" source="Objing">
33 </eAnnotations>
34 <upperBound xmi:type="uml:LiteralString" xmi:id="_mhZFXiICEeKSKePaylnoVA"
  name="UpperBound" value="1"/>
35 </node>
36 <node xmi:type="uml:CentralBufferNode" xmi:id="_mhZFXyICEeKSKePaylnoVA" name="Name"
  incoming="_mhZFbiICEeKSKePaylnoVA">
37 <eAnnotations xmi:id="_mhZFYICEeKSKePaylnoVA" source="Objing">
41 </eAnnotations>
42 <upperBound xmi:type="uml:LiteralString" xmi:id="_mhZFYyICEeKSKePaylnoVA"
  name="UpperBound" value="1"/>
43 </node>
44 <node xmi:type="uml:CentralBufferNode" xmi:id="_mhZFZICEeKSKePaylnoVA" name="Address"
  incoming="_mhZsYyICEeKSKePaylnoVA">
45 <eAnnotations xmi:id="_mhZFISICEeKSKePaylnoVA" source="Objing">
49 </eAnnotations>
50 <upperBound xmi:type="uml:LiteralString" xmi:id="_mhZFaCICEeKSKePaylnoVA"
  name="UpperBound" value="1"/>
51 </node>
52 <node xmi:type="uml:CentralBufferNode" xmi:id="_mhZFaSICEeKSKePaylnoVA" name="Product"
  incoming="_mhZsaCICEeKSKePaylnoVA">
53 <eAnnotations xmi:id="_mhZFaiICEeKSKePaylnoVA" source="Objing">
57 </eAnnotations>
58 <upperBound xmi:type="uml:LiteralString" xmi:id="_mhZFsSICEeKSKePaylnoVA"
  name="UpperBound" value="1"/>
59 </node>
60 <edge xmi:type="uml:ObjectFlow" xmi:id="_mhZFbiICEeKSKePaylnoVA" name="ObjectFlow"
  source="_mhZFWSICEeKSKePaylnoVA" target="_mhZFXyICEeKSKePaylnoVA">
61 <eAnnotations xmi:id="_mhZFbyICEeKSKePaylnoVA" source="Objing">
65 </eAnnotations>
66 <weight xmi:type="uml:LiteralInteger" xmi:id="_mhZsYiICEeKSKePaylnoVA" value="1"/>
67 </edge>
68 <edge xmi:type="uml:ObjectFlow" xmi:id="_mhZsYyICEeKSKePaylnoVA" name="ObjectFlow"
  source="_mhZFWSICEeKSKePaylnoVA" target="_mhZFZICEeKSKePaylnoVA">
69 <eAnnotations xmi:id="_mhZsZICEeKSKePaylnoVA" source="Objing">
73 </eAnnotations>
74 <weight xmi:type="uml:LiteralInteger" xmi:id="_mhZsYzICEeKSKePaylnoVA" value="1"/>
75 </edge>
76 <edge xmi:type="uml:ObjectFlow" xmi:id="_mhZsaCICEeKSKePaylnoVA" name="ObjectFlow"
  source="_mhZFWSICEeKSKePaylnoVA" target="_mhZFaSICEeKSKePaylnoVA">
77 <eAnnotations xmi:id="_mhZsaSICEeKSKePaylnoVA" source="Objing">
81 </eAnnotations>
82 <weight xmi:type="uml:LiteralInteger" xmi:id="_mhZsbCICEeKSKePaylnoVA" value="1"/>
83 </edge>
84 <edge xmi:type="uml:ObjectFlow" xmi:id="_mhZsbSICEeKSKePaylnoVA" name="ObjectFlow"
  source="_mhZFWiICEeKSKePaylnoVA" target="_mhZFWSICEeKSKePaylnoVA">
85 <eAnnotations xmi:id="_mhZsbiICEeKSKePaylnoVA" source="Objing">
89 </eAnnotations>
90 <weight xmi:type="uml:LiteralInteger" xmi:id="_mhZscSICEeKSKePaylnoVA" value="1"/>
91 </edge>
92 </packagedElement>
93 </uml:Model>
94

```

ภาพที่ 31 เอกซ์เอ็มไอไฟล์ที่ได้จากเครื่องมือทดสอบ

เอกสารเอ็มไอไฟล์ที่ถูกแปลงจากตัวแปลงแผนภาพกิจกรรม ให้ข้อมูลผลลัพธ์ ดังแสดงในภาพ
ที่ 32

```

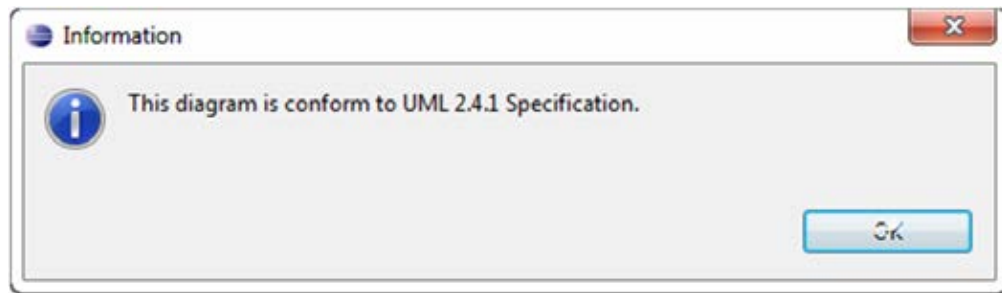
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <uml:Model xmlns:uml="http://www.omg.org/spec/UML/20090901"
  xmlns:xmi="http://schema.omg.org/spec/XMI/2.1" name="Activity Diagrams"
  xmi:id="_wNqoQgFzEd6Eo8cPzRliQg" xmi:version="2.1">
3 <packagedElement name="ex1" xmi:id="_wNqo6aFzEd6Eo8cPzRliQg" xmi:type="uml:Activity">
4 <node name="Unmarshall Order" xmi:id="_mhzFWSZCEeKSKePaylnoVA"
  xmi:type="uml:OpaqueAction">
5 <argument incoming="_mhzsbSZCEeKSKePaylnoVA" name="Call0.input0"
  xmi:id="_mhzsbSZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
6 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
7 </argument>
8 <result name="Call0.output0" outgoing="_mhzFbiZCEeKSKePaylnoVA"
  xmi:id="_mhzFbiZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
9 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
10 </result>
11 <result name="Call0.output1" outgoing="_mhzsYyZCEeKSKePaylnoVA"
  xmi:id="_mhzsYyZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
12 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
13 </result>
14 <result name="Call0.output2" outgoing="_mhzsaCZCEeKSKePaylnoVA"
  xmi:id="_mhzsaCZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
15 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
16 </result>
17 </node>
18 <node name="Order" xmi:id="_mhzFWiZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
19 <result name="Call1.output0" outgoing="_mhzsbSZCEeKSKePaylnoVA"
  xmi:id="_mhzsbSZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
20 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
21 </result>
22 </node>
23 <node name="Name" xmi:id="_mhzFXyZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
24 <argument incoming="_mhzFbiZCEeKSKePaylnoVA" name="Call2.input0"
  xmi:id="_mhzFbiZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
25 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
26 </argument>
27 </node>
28 <node name="Address" xmi:id="_mhzFZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
29 <argument incoming="_mhzsYyZCEeKSKePaylnoVA" name="Call3.input0"
  xmi:id="_mhzsYyZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
30 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
31 </argument>
32 </node>
33 <node name="Product" xmi:id="_mhzFaSZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
34 <argument incoming="_mhzsaCZCEeKSKePaylnoVA" name="Call4.input0"
  xmi:id="_mhzsaCZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
35 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
36 </argument>
37 </node>
38 <edge source="_mhzFWSZCEeKSKePaylnoVA" target="_mhzFXyZCEeKSKePaylnoVA"
  xmi:id="_mhzFbiZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
39 <weight value="1" xmi:id="_wNqpC3izEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
40 </edge>
41 <edge source="_mhzFWSZCEeKSKePaylnoVA" target="_mhzFZCEeKSKePaylnoVA"
  xmi:id="_mhzsYyZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
42 <weight value="1" xmi:id="_wNqpC28zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
43 </edge>
44 <edge source="_mhzFWSZCEeKSKePaylnoVA" target="_mhzFaSZCEeKSKePaylnoVA"
  xmi:id="_mhzsaCZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
45 <weight value="1" xmi:id="_wNqpC98zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
46 </edge>
47 <edge source="_mhzFWiZCEeKSKePaylnoVA" target="_mhzFWSZCEeKSKePaylnoVA"
  xmi:id="_mhzsbSZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
48 <weight value="1" xmi:id="_wNqpC80zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
49 </edge>
50 </packagedElement>
51 </uml:Model>

```

ภาพที่ 32 ผลลัพธ์เอกสารเอ็มไอของตัวแปลงแผนภาพกิจกรรม

5.2.2 สรุปผล

ผลลัพธ์ที่ได้ เมื่อนำไปทดสอบกับเครื่องมือที่พัฒนาในงานวิจัยที่ [2] เพื่อทวนสอบแผนภาพกิจกรรม ผลปรากฏว่า เครื่องมือทวนสอบแผนภาพกิจกรรมสามารถอ่านไฟล์เอ็กซ์เอ็มไอ และทำการทวนสอบได้ผลลัพธ์ ดังแสดงในภาพที่ 33

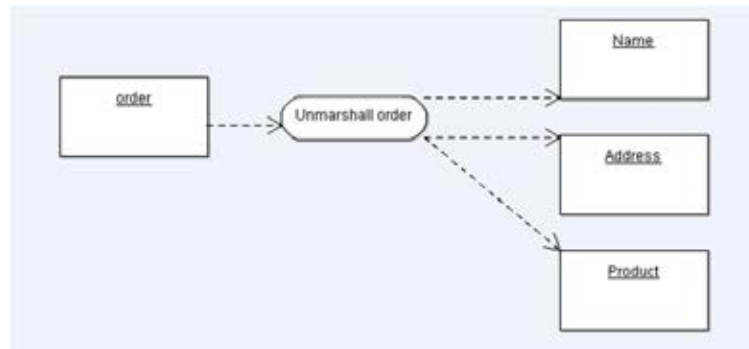


ภาพที่ 33 ผลลัพธ์จากเครื่องมือทวนสอบแผนภาพกิจกรรมในงานวิจัย [2]

5.3 การทดสอบตัวอย่างที่ 2

5.3.1 ข้อมูลนำเข้าจากซอฟต์แวร์ ArgoUML Version 0.34

ในตัวอย่างทดสอบที่ 3 นั้น ใช้ชุดแผนภาพกิจกรรมทดสอบจากเครื่องมือ ArgoUML Version 0.34 แสดงให้เห็นถึงแอ็กชันอินพุตและแอ็กชันเอาต์พุต ตามรูปแบบพื้นฐานแผนภาพกิจกรรมทั่วไป ดังแสดงในภาพที่ 34



ภาพที่ 34 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ

หลังจากสร้างแผนภาพกิจกรรมบนเครื่องมือ ArgoUML Version 0.34 แล้ว ให้นำออกข้อมูลในรูปแบบเอ็กซ์เอ็มไอไฟล์เพื่อเป็นอินพุตสำหรับเครื่องมือตัวแปลงแผนภาพกิจกรรม ตัวอย่างแสดงในภาพที่ 35

```

1 <?xml version = '1.0' encoding = 'UTF-8' ?>
2 <XMI xmi.version = '1.2' xmlns:UML = 'org.omg.xmi.namespace.UML' timestamp = 'Sat Oct 27 23:34:14
ICT 2012'>
3 <XMI.header> <XMI.documentation>
4 <XMI.exporter>ArgoUML (using Netbeans XMI Writer version 1.0)</XMI.exporter>
5 <XMI.exporterVersion>0.34(6) revised on $Date: 2010-01-11 22:20:14 +0100 (Mon, 11 Jan 2010)
$ </XMI.exporterVersion>
6 </XMI.documentation>
7 <XMI.metamodel xmi.name="UML" xmi.version="1.4"/></XMI.header>
8 <XMI.content>
9 <UML:Model xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000865'
10 name = 'ex1' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'
11 isAbstract = 'false'>
12 <UML:Namespace.ownedElement>
13 <UML:ActivityGraph xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000866'
14 name = 'ex1' isSpecification = 'false'>
15 <UML:StateMachine.context>
16 <UML:Model xmi.idref = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000865' />
17 </UML:StateMachine.context>
18 <UML:StateMachine.top>
19 <UML:CompositeState xmi.id =
20 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000867'
21 name = 'top' isSpecification = 'false' isConcurrent = 'false'>
22 <UML:CompositeState.subvertex>
23 <UML:ActionState xmi.id =
41 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000868'
42 </UML:ActionState>
43 <UML:ObjectFlowState xmi.id =
44 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000869'
45 </UML:ObjectFlowState>
46 <UML:ObjectFlowState xmi.id =
47 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086A'
48 name = 'Name' isSpecification = 'false' isSynch = 'false'>
49 <UML:StateVertex.incoming>
50 </UML:StateVertex.incoming>
51 <UML:ObjectFlowState.type>
52 <UML:Class xmi.idref =
53 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086E' />
54 </UML:ObjectFlowState.type>
55 </UML:ObjectFlowState>
56 <UML:ObjectFlowState xmi.id =
57 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086B'
58 </UML:ObjectFlowState>
59 <UML:ObjectFlowState xmi.id =
60 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086C'
61 name = 'Product' isSpecification = 'false' isSynch = 'false'>
62 <UML:StateVertex.incoming>
63 <UML:Transition xmi.idref =
64 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000876' />
65 </UML:StateVertex.incoming>
66 <UML:ObjectFlowState.type>
67 <UML:Class xmi.idref =
68 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000870' />
69 </UML:ObjectFlowState.type>
70 </UML:ObjectFlowState>
71 </UML:CompositeState.subvertex>
72 </UML:CompositeState>
73 </UML:StateMachine.top>
74 <UML:StateMachine.transitions>
75 </UML:StateMachine.transitions>
76 </UML:ActivityGraph>
77 <UML:Class xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086D'
78 </UML:Class>
79 <UML:Class xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086E'
80 </UML:Class>
81 <UML:Class xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086F'
82 </UML:Class>
83 <UML:Class xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000870'
84 </UML:Class>
85 </UML:Namespace.ownedElement>
86 </UML:Model>
87 </XMI.content>
88 </XMI>
89

```

ภาพที่ 35 เอ็กซ์เอ็มไอไฟล์ที่ได้จากเครื่องมือทดสอบ

เอกสารเอ็มไอไฟล์ที่ถูกแปลงจากตัวแปลงแผนภาพกิจกรรม ให้ข้อมูลผลลัพธ์ ดังแสดงในภาพ

ที่ 36

```

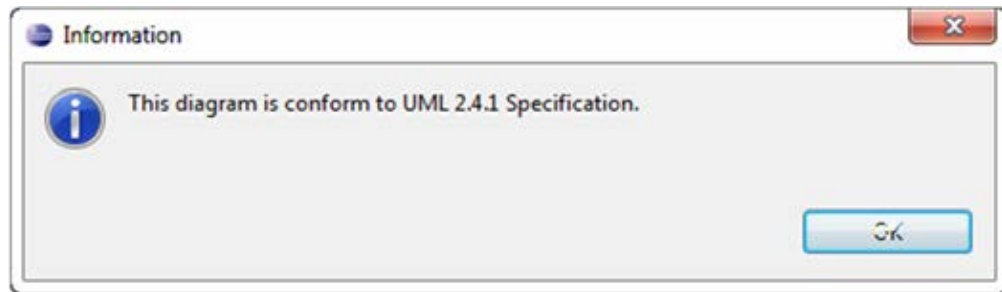
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <uml:Model xmlns:uml="http://www.omg.org/spec/UML/20090901"
  xmlns:xmi="http://schema.omg.org/spec/XMI/2.1" name="Activity Diagrams"
  xmi:id="wNqo0qFzEd6Eo8cPzRliQg" xmi:version="2.1">
3 <packagedElement name="xmi" xmi:id="wNqo6aFzEd6Eo8cPzRliQg" xmi:type="uml:Activity">
4 <node name="Unmarshall Order" xmi:id="_mhZFWSZCEeKSKePaylnoVA"
  xmi:type="uml:OpaqueAction">
5 <argument incoming="_mhZsbSZCEeKSKePaylnoVA" name="Call0.input0"
  xmi:id="_mhZsbSZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
6 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
7 </argument>
8 <result name="Call0.output0" outgoing="_mhZFbIZCEeKSKePaylnoVA"
  xmi:id="_mhZFbIZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
9 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
10 </result>
11 <result name="Call0.output1" outgoing="_mhZsYyZCEeKSKePaylnoVA"
  xmi:id="_mhZsYyZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
12 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
13 </result>
14 <result name="Call0.output2" outgoing="_mhZsaCZCEeKSKePaylnoVA"
  xmi:id="_mhZsaCZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
15 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
16 </result>
17 </node>
18 <node name="Order" xmi:id="_mhZFwIZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
19 <result name="Call1.output0" outgoing="_mhZsbSZCEeKSKePaylnoVA"
  xmi:id="_mhZsbSZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
20 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
21 </result>
22 </node>
23 <node name="Name" xmi:id="_mhZFXyZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
24 <argument incoming="_mhZFbIZCEeKSKePaylnoVA" name="Call2.input0"
  xmi:id="_mhZFbIZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
25 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
26 </argument>
27 </node>
28 <node name="Address" xmi:id="_mhZFZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
29 <argument incoming="_mhZsYyZCEeKSKePaylnoVA" name="Call3.input0"
  xmi:id="_mhZsYyZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
30 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
31 </argument>
32 </node>
33 <node name="Product" xmi:id="_mhZFaSZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
34 <argument incoming="_mhZsaCZCEeKSKePaylnoVA" name="Call4.input0"
  xmi:id="_mhZsaCZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
35 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
36 </argument>
37 </node>
38 <edge source="_mhZFWSZCEeKSKePaylnoVA" target="_mhZFXyZCEeKSKePaylnoVA"
  xmi:id="_mhZFbIZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
39 <weight value="1" xmi:id="wNqoC3izEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
40 </edge>
41 <edge source="_mhZFWSZCEeKSKePaylnoVA" target="_mhZFZCEeKSKePaylnoVA"
  xmi:id="_mhZsYyZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
42 <weight value="1" xmi:id="wNqoC28zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
43 </edge>
44 <edge source="_mhZFWSZCEeKSKePaylnoVA" target="_mhZFaSZCEeKSKePaylnoVA"
  xmi:id="_mhZsaCZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
45 <weight value="1" xmi:id="wNqoC98zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
46 </edge>
47 <edge source="_mhZFwIZCEeKSKePaylnoVA" target="_mhZFWSZCEeKSKePaylnoVA"
  xmi:id="_mhZsbSZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
48 <weight value="1" xmi:id="wNqoC80zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
49 </edge>
50 </packagedElement>
51 </uml:Model>

```

ภาพที่ 36 ผลลัพธ์เอกสารเอ็มไอของตัวแปลงแผนภาพกิจกรรม

5.3.2 สรุปผล

ผลลัพธ์ที่ได้ เมื่อนำไปทดสอบกับเครื่องมือที่พัฒนาในงานวิจัยที่ [2] เพื่อทวนสอบแผนภาพกิจกรรม ผลปรากฏว่า เครื่องมือทวนสอบแผนภาพกิจกรรมสามารถอ่านไฟล์เอ็กซ์เอ็มไอ และทำการทวนสอบได้ผลลัพธ์ ดังแสดงในภาพที่ 37

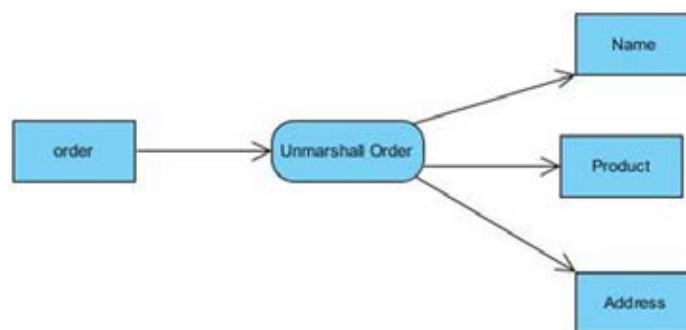


ภาพที่ 37 ผลลัพธ์จากเครื่องมือทวนสอบแผนภาพกิจกรรมในงานวิจัย [2]

5.4 การทดสอบตัวอย่างที่ 3

5.4.1 ข้อมูลนำเข้าจากซอฟต์แวร์ Visual paradigm Version 10.0

ในตัวอย่างทดสอบที่ 3 นั้น ใช้ชุดแผนภาพกิจกรรมทดสอบจากเครื่องมือ Visual paradigm Version 10.0 แสดงให้เห็นถึงแอ็กชันอินพุตและแอ็กชันเอาต์พุต ตามรูปแบบพื้นฐานแผนภาพกิจกรรมทั่วไป ดังแสดงในภาพที่ 38



ภาพที่ 38 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ

หลังจากสร้างแผนภาพกิจกรรมบนเครื่องมือ Visual paradigm Version 10.0 แล้ว ให้นำออกข้อมูลในรูปแบบเอ็กซ์เอ็มไอฟाइล์เพื่อเป็นอินพุตสำหรับเครื่องมือตัวแปลงแผนภาพกิจกรรม ตัวอย่างแสดงในภาพที่ 39

```

1 <?xml version = '1.0' encoding = 'UTF-8' ?>
2 <XMI xmi.version = '1.2' xmlns:UML = 'org.omg.xmi.namespace.UML' timestamp = 'Sat Oct 27 23:34:14
ICT 2012'>
3 <XMI.header> <XMI.documentation>
4 <XMI.exporter>ArgoUML (using Netbeans XMI Writer version 1.0)</XMI.exporter>
5 <XMI.exporterVersion>0.34(6) revised on $Date: 2010-01-11 22:20:14 +0100 (Mon, 11 Jan 2010)
$ </XMI.exporterVersion>
6 </XMI.documentation>
7 <XMI.metamodel xmi.name="UML" xmi.version="1.4"/></XMI.header>
8 <XMI.content>
9 <UML:Model xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000865'
10 name = 'ex1' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'
11 isAbstract = 'false'>
12 <UML:Namespace.ownedElement>
13 <UML:ActivityGraph xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000866'
14 name = 'ex1' isSpecification = 'false'>
15 <UML:StateMachine.context>
16 <UML:Model xmi.idref = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000865' />
17 </UML:StateMachine.context>
18 <UML:StateMachine.top>
19 <UML:CompositeState xmi.id =
20 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000867'
21 name = 'top' isSpecification = 'false' isConcurrent = 'false'>
22 <UML:CompositeState.subvertex>
23 <UML:ActionState xmi.id =
41 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000868'
42 </UML:ActionState>
43 <UML:ObjectFlowState xmi.id =
44 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000869'
45 </UML:ObjectFlowState>
46 <UML:ObjectFlowState xmi.id =
47 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086A'
48 name = 'Name' isSpecification = 'false' isSynch = 'false'>
49 <UML:StateVertex.incoming>
50 </UML:StateVertex.incoming>
51 <UML:ObjectFlowState.type>
52 <UML:Class xmi.idref =
53 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086E' />
54 </UML:ObjectFlowState.type>
55 </UML:ObjectFlowState>
56 <UML:ObjectFlowState xmi.id =
57 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086B'
58 </UML:ObjectFlowState>
59 <UML:ObjectFlowState xmi.id =
60 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086C'
61 name = 'Product' isSpecification = 'false' isSynch = 'false'>
62 <UML:StateVertex.incoming>
63 <UML:Transition xmi.idref =
64 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000876' />
65 </UML:StateVertex.incoming>
66 <UML:ObjectFlowState.type>
67 <UML:Class xmi.idref =
68 '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000870' />
69 </UML:ObjectFlowState.type>
70 </UML:ObjectFlowState>
71 </UML:CompositeState.subvertex>
72 </UML:CompositeState>
73 </UML:StateMachine.top>
74 <UML:StateMachine.transitions>
75 </UML:StateMachine.transitions>
76 </UML:ActivityGraph>
77 <UML:Class xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086D'
78 </UML:Class>
79 <UML:Class xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086E'
80 </UML:Class>
81 <UML:Class xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:000000000000086F'
82 </UML:Class>
83 <UML:Class xmi.id = '-64--88-1-5-1cf81117:13aa30d6c5e:-8000:0000000000000870'
84 </UML:Class>
85 </UML:Namespace.ownedElement>
86 </UML:Model>
87 </XMI.content>
88 </XMI>
89

```

ภาพที่ 39 เอ็กซ์เอ็มไอฟाइล์ที่ได้จากเครื่องมือทดสอบ

เอกสารเอ็มไอไฟล์ที่ถูกแปลงจากตัวแปลงแผนภาพกิจกรรม ให้ข้อมูลผลลัพธ์ ดังแสดงในภาพที่ 40

```

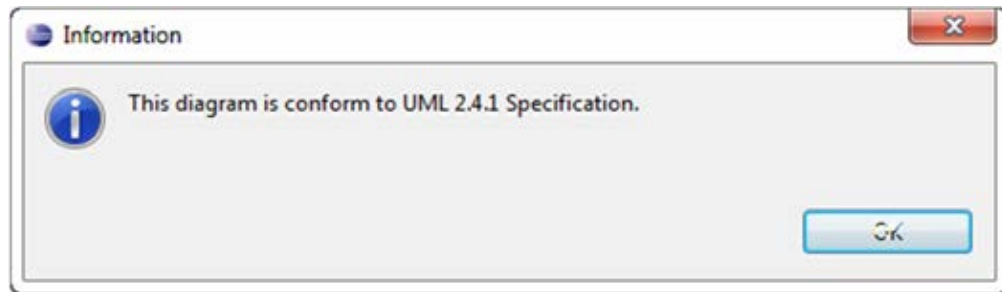
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <uml:Model xmlns:uml="http://www.omg.org/spec/UML/20090901"
  xmlns:xmi="http://schema.omg.org/spec/XMI/2.1" name="Activity Diagrams"
  xmi:id="_wNqoQgFzEd6Eo8cPzRliQg" xmi:version="2.1">
3 <packageElement name="ex1" xmi:id="_wNqo6aFzEd6Eo8cPzRliQg" xmi:type="uml:Activity">
4 <node name="Unmarshal Order" xmi:id="_mhZfWSZCEeKSKePaylnoVA"
  xmi:type="uml:OpaqueAction">
5 <argument incoming="_mhZsbS2CEeKSKePaylnoVA" name="Call10.input0"
  xmi:id="_mhZsbS2CEeKSKePaylnoVA" xmi:type="uml:InputPin">
6 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
7 </argument>
8 <result name="Call10.output0" outgoing="_mhZfbiZCEeKSKePaylnoVA"
  xmi:id="_mhZfbiZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
9 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
10 </result>
11 <result name="Call10.output1" outgoing="_mhZsYyZCEeKSKePaylnoVA"
  xmi:id="_mhZsYyZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
12 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
13 </result>
14 <result name="Call10.output2" outgoing="_mhZsaC2CEeKSKePaylnoVA"
  xmi:id="_mhZsaC2CEeKSKePaylnoVA" xmi:type="uml:OutputPin">
15 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
16 </result>
17 </node>
18 <node name="Order" xmi:id="_mhZfWiZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
19 <result name="Call11.output0" outgoing="_mhZsbS2CEeKSKePaylnoVA"
  xmi:id="_mhZsbS2CEeKSKePaylnoVA" xmi:type="uml:OutputPin">
20 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
21 </result>
22 </node>
23 <node name="Name" xmi:id="_mhZfXyZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
24 <argument incoming="_mhZfbiZCEeKSKePaylnoVA" name="Call12.input0"
  xmi:id="_mhZfbiZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
25 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
26 </argument>
27 </node>
28 <node name="Address" xmi:id="_mhZfZ2CEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
29 <argument incoming="_mhZsYyZCEeKSKePaylnoVA" name="Call13.input0"
  xmi:id="_mhZsYyZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
30 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
31 </argument>
32 </node>
33 <node name="Product" xmi:id="_mhZfaS2CEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
34 <argument incoming="_mhZsaC2CEeKSKePaylnoVA" name="Call14.input0"
  xmi:id="_mhZsaC2CEeKSKePaylnoVA" xmi:type="uml:InputPin">
35 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
36 </argument>
37 </node>
38 <edge source="_mhZfWSZCEeKSKePaylnoVA" target="_mhZfXyZCEeKSKePaylnoVA"
  xmi:id="_mhZfbiZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
39 <weight value="1" xmi:id="_wNqpc31zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
40 </edge>
41 <edge source="_mhZfWSZCEeKSKePaylnoVA" target="_mhZfZ2CEeKSKePaylnoVA"
  xmi:id="_mhZsYyZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
42 <weight value="1" xmi:id="_wNqpc28zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
43 </edge>
44 <edge source="_mhZfWSZCEeKSKePaylnoVA" target="_mhZfaS2CEeKSKePaylnoVA"
  xmi:id="_mhZsaC2CEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
45 <weight value="1" xmi:id="_wNqpc98zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
46 </edge>
47 <edge source="_mhZfWiZCEeKSKePaylnoVA" target="_mhZfWSZCEeKSKePaylnoVA"
  xmi:id="_mhZsbS2CEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
48 <weight value="1" xmi:id="_wNqpc80zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
49 </edge>
50 </packageElement>
51 </uml:Model>

```

ภาพที่ 40 ผลลัพธ์เอกสารเอ็มไอของตัวแปลงแผนภาพกิจกรรม

5.4.2 สรุปผล

ผลลัพธ์ที่ได้ เมื่อนำไปทดสอบกับเครื่องมือที่พัฒนาในงานวิจัยที่ [2] เพื่อทวนสอบแผนภาพกิจกรรม ผลปรากฏว่า เครื่องมือทวนสอบแผนภาพกิจกรรมสามารถอ่านไฟล์เอ็กซ์เอ็มไอ และทำการทวนสอบได้ผลลัพธ์ ดังแสดงในภาพที่ 41

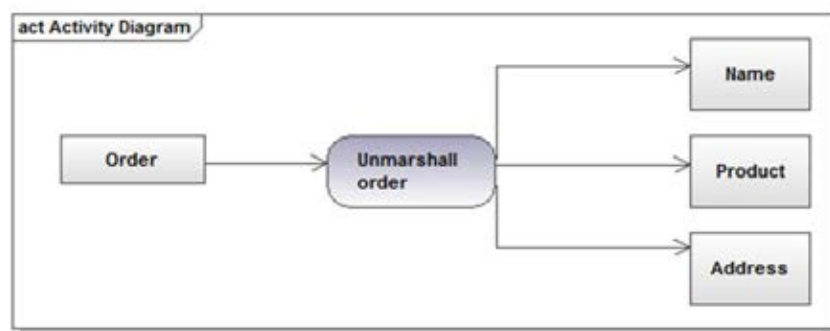


ภาพที่ 41 ผลลัพธ์จากเครื่องมือทวนสอบแผนภาพกิจกรรมในงานวิจัย [2]

5.5 การทดสอบตัวอย่างที่ 4

5.5.1 ข้อมูลนำเข้าจากซอฟต์แวร์ Altova Umodel Enterprise Ed. 2013 rel. 2 sp1

ในตัวอย่างทดสอบที่ 4 นั้น ใช้ชุดแผนภาพกิจกรรมทดสอบจาก Altova Umodel Enterprise Ed. 2013 rel. 2 sp1 แสดงให้เห็นถึงแอ็กชันอินพุตและแอ็กชันเอาต์พุต ตามรูปแบบพื้นฐานแผนภาพกิจกรรมทั่วไป ดังแสดงในภาพที่ 42



ภาพที่ 42 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ

หลังจากสร้างแผนภาพกิจกรรมบนเครื่องมือ Altova Umodel Enterprise Ed. 2013 R.2 แล้วให้นำออกข้อมูลในรูปแบบเอ็กซ์เอ็มไอไฟล์เพื่อเป็นอินพุตสำหรับเครื่องมือตัวแปลงแผนภาพกิจกรรม ตัวอย่างแสดงในภาพที่ 43

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xml:XMI xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"
  xmlns:uml="http://schema.omg.org/spec/UML/2.1.2" xmi:version="2.1">
3   <xmi:Documentation contact="www.altova.com" exporter="Altova UModel2013r2sp1"
  exporterVersion="11"/>
4   <xmi:Extension extender="UModel">
5     <OpenDiagrams activeDiagram="f3f8fb66-547f-40c3-99f6-aa04e78d6ea6">
6     </OpenDiagrams>
7     <Diagrams>
8     </Diagrams>
9   </xmi:Extension>
10  <uml:Model xmi:id="U00000001-7510-11d9-86f2-000476a22f44"
  xmi:uuid="00000001-7510-11d9-86f2-000476a22f44" name="Root">
11    <packageElement xmi:type="uml:Package" xmi:id="U00000003-7510-11d9-86f2-000476a22f44"
  xmi:uuid="00000003-7510-11d9-86f2-000476a22f44" name="Component View"/>
12    <packageElement xmi:type="uml:Activity" xmi:id="U13eab02a-c011-4eee-b332-16c3f9160ed3"
  xmi:uuid="13eab02a-c011-4eee-b332-16c3f9160ed3" name="Activity1" isReentrant="true">
13      <node xmi:type="uml:OpaqueAction" xmi:id="U0250464b-286d-4102-979c-5f411d12d7f9"
  xmi:uuid="0250464b-286d-4102-979c-5f411d12d7f9" name="Hire Employee">
14        <outgoing xmi:idref="Ueb6d747e-4603-4f55-8772-3aad69d08cc8"/>
15      </node>
16      <node xmi:type="uml:DataStoreNode" xmi:id="Uf00a26e3-2d4d-4a9e-bdd1-bb9ba2af5c7a"
  xmi:uuid="f00a26e3-2d4d-4a9e-bdd1-bb9ba2af5c7a" name="Personal Database">
17        <incoming xmi:idref="Ueb6d747e-4603-4f55-8772-3aad69d08cc8"/>
18        <outgoing xmi:idref="Ub9846c01-f22e-44de-8877-0ce7ac440b4b"/>
19        <outgoing xmi:idref="U4daadf57-6755-4f3d-b6e2-5418563ed363"/>
20      </node>
21      <node xmi:type="uml:CallOperationAction"
  xmi:id="U59c1148a-5e45-4a46-b7ce-7e7099b51b22"
  xmi:uuid="59c1148a-5e45-4a46-b7ce-7e7099b51b22" name="Assign Employee">
22        <incoming xmi:idref="Ub9846c01-f22e-44de-8877-0ce7ac440b4b"/>
23      </node>
24      <node xmi:type="uml:JoinNode" xmi:id="U65bcc87f-c033-462f-8a3b-c57e78738caf"
  xmi:uuid="65bcc87f-c033-462f-8a3b-c57e78738caf" name="JoinNode">
25        <incoming xmi:idref="U4daadf57-6755-4f3d-b6e2-5418563ed363"/>
26        <incoming xmi:idref="U3dd765ae-4ef1-43c6-8f68-6f3b576b5b15"/>
27        <outgoing xmi:idref="U790170e2-27ec-4666-9c48-b6a9bdb4f2b7"/>
28      </node>
29      <node xmi:type="uml:AcceptEventAction" xmi:id="Uf6c4122b-a407-4186-8c78-914f5f7cd171"
  xmi:uuid="f6c4122b-a407-4186-8c78-914f5f7cd171" name="Once a year">
30        <trigger xmi:type="uml:Trigger" xmi:id="Udf729a4a-3b97-4b8e-bbd2-81a523df52a3"
  xmi:uuid="df729a4a-3b97-4b8e-bbd2-81a523df52a3" name="Trigger"
  event="Ua0855c58-8609-4ce3-816b-c1ea0d643f88"/>
31        <outgoing xmi:idref="U3dd765ae-4ef1-43c6-8f68-6f3b576b5b15"/>
32      </node>
33      <node xmi:type="uml:OpaqueAction" xmi:id="U6e21bac4-7fb1-473c-8702-6a6c5ed0c201"
  xmi:uuid="6e21bac4-7fb1-473c-8702-6a6c5ed0c201" name="Review Employee">
34        <incoming xmi:idref="U790170e2-27ec-4666-9c48-b6a9bdb4f2b7"/>
35      </node>
36      <edge xmi:type="uml:ControlFlow" xmi:id="Ueb6d747e-4603-4f55-8772-3aad69d08cc8"
  xmi:uuid="eb6d747e-4603-4f55-8772-3aad69d08cc8"
  source="U0250464b-286d-4102-979c-5f411d12d7f9"
  target="Uf00a26e3-2d4d-4a9e-bdd1-bb9ba2af5c7a"/>
37      <edge xmi:type="uml:ControlFlow" xmi:id="Ub9846c01-f22e-44de-8877-0ce7ac440b4b"
  xmi:uuid="b9846c01-f22e-44de-8877-0ce7ac440b4b"
  source="Uf00a26e3-2d4d-4a9e-bdd1-bb9ba2af5c7a"
  target="U59c1148a-5e45-4a46-b7ce-7e7099b51b22"/>
38      <edge xmi:type="uml:ControlFlow" xmi:id="U4daadf57-6755-4f3d-b6e2-5418563ed363"
  xmi:uuid="4daadf57-6755-4f3d-b6e2-5418563ed363"
  source="Uf00a26e3-2d4d-4a9e-bdd1-bb9ba2af5c7a"
  target="U65bcc87f-c033-462f-8a3b-c57e78738caf"/>
39      <edge xmi:type="uml:ControlFlow" xmi:id="U3dd765ae-4ef1-43c6-8f68-6f3b576b5b15"
  xmi:uuid="3dd765ae-4ef1-43c6-8f68-6f3b576b5b15"
  source="Uf6c4122b-a407-4186-8c78-914f5f7cd171"
  target="U65bcc87f-c033-462f-8a3b-c57e78738caf"/>
40      <edge xmi:type="uml:ControlFlow" xmi:id="U790170e2-27ec-4666-9c48-b6a9bdb4f2b7"
  xmi:uuid="790170e2-27ec-4666-9c48-b6a9bdb4f2b7"
  source="U65bcc87f-c033-462f-8a3b-c57e78738caf"
  target="U6e21bac4-7fb1-473c-8702-6a6c5ed0c201"/>
41    </packageElement>
42    <packageElement xmi:type="uml:TimeEvent" xmi:id="Ua0855c58-8609-4ce3-816b-c1ea0d643f88"
  xmi:uuid="a0855c58-8609-4ce3-816b-c1ea0d643f88"/>
43  </uml:Model>
44 </xml:XMI>

```

ภาพที่ 43 เอ็กซ์เอ็มไอไฟล์ที่ได้จากเครื่องมือทดสอบ

เอกสารเอ็มไอไฟล์ที่ถูกแปลงจากตัวแปลงแผนภาพกิจกรรม ให้ข้อมูลผลลัพธ์ ดังแสดงในภาพ
ที่ 44

```

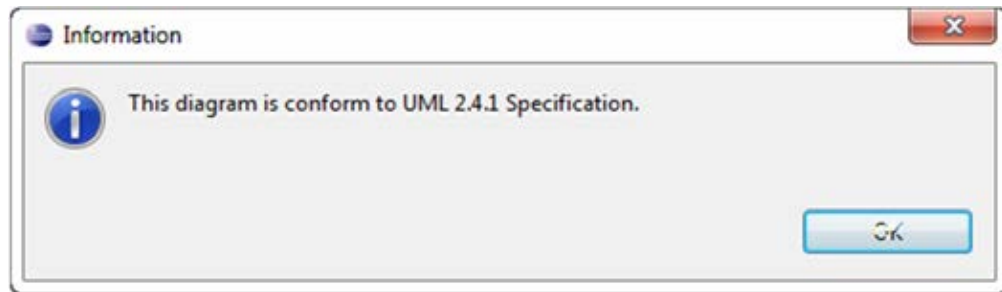
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <uml:Model xmlns:uml="http://www.omg.org/spec/UML/20090901"
  xmlns:xmi="http://schema.omg.org/spec/XMI/2.1" name="Activity Diagrams"
  xmi:id="_wNqoQqFzEd6Eo8cPzRliQg" xmi:version="2.1">
3 <packagedElement name="ex1" xmi:id="_wNqo6aFzEd6Eo8cPzRliQg" xmi:type="uml:Activity">
4 <node name="Unmarshall Order" xmi:id="_mhZFWSZCEeKSKePaylnoVA"
  xmi:type="uml:OpaqueAction">
5 <argument incoming="_mhZsbSZCEeKSKePaylnoVA" name="Call10.input0"
  xmi:id="_mhZsbSZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
6 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
7 </argument>
8 <result name="Call10.output0" outgoing="_mhZFbiZCEeKSKePaylnoVA"
  xmi:id="_mhZFbiZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
9 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
10 </result>
11 <result name="Call10.output1" outgoing="_mhZsaYyZCEeKSKePaylnoVA"
  xmi:id="_mhZsaYyZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
12 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
13 </result>
14 <result name="Call10.output2" outgoing="_mhZsaCZCEeKSKePaylnoVA"
  xmi:id="_mhZsaCZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
15 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
16 </result>
17 </node>
18 <node name="Order" xmi:id="_mhZFwiZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
19 <result name="Call11.output0" outgoing="_mhZsbSZCEeKSKePaylnoVA"
  xmi:id="_mhZsbSZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
20 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
21 </result>
22 </node>
23 <node name="Name" xmi:id="_mhZFKyZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
24 <argument incoming="_mhZFbiZCEeKSKePaylnoVA" name="Call12.input0"
  xmi:id="_mhZFbiZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
25 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
26 </argument>
27 </node>
28 <node name="Address" xmi:id="_mhZF2CZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
29 <argument incoming="_mhZsaYyZCEeKSKePaylnoVA" name="Call13.input0"
  xmi:id="_mhZsaYyZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
30 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
31 </argument>
32 </node>
33 <node name="Product" xmi:id="_mhZFaSZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
34 <argument incoming="_mhZsaCZCEeKSKePaylnoVA" name="Call14.input0"
  xmi:id="_mhZsaCZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
35 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
36 </argument>
37 </node>
38 <edge source="_mhZFWSZCEeKSKePaylnoVA" target="_mhZFKyZCEeKSKePaylnoVA"
  xmi:id="_mhZFbiZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
39 <weight value="1" xmi:id="_wNqpC31zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
40 </edge>
41 <edge source="_mhZFWSZCEeKSKePaylnoVA" target="_mhZF2CZCEeKSKePaylnoVA"
  xmi:id="_mhZsaYyZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
42 <weight value="1" xmi:id="_wNqpC28zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
43 </edge>
44 <edge source="_mhZFWSZCEeKSKePaylnoVA" target="_mhZFaSZCEeKSKePaylnoVA"
  xmi:id="_mhZsaCZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
45 <weight value="1" xmi:id="_wNqpC98zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
46 </edge>
47 <edge source="_mhZFwiZCEeKSKePaylnoVA" target="_mhZFWSZCEeKSKePaylnoVA"
  xmi:id="_mhZsbSZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
48 <weight value="1" xmi:id="_wNqpC80zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
49 </edge>
50 </packagedElement>
51 </uml:Model>

```

ภาพที่ 44 ผลลัพธ์เอกสารเอ็มไอของตัวแปลงแผนภาพกิจกรรม

5.5.2 สรุปผล

ผลลัพธ์ที่ได้ เมื่อนำไปทดสอบกับเครื่องมือที่พัฒนาในงานวิจัยที่ [2] เพื่อทวนสอบแผนภาพกิจกรรม ผลปรากฏว่า เครื่องมือทวนสอบแผนภาพกิจกรรมสามารถอ่านไฟล์เอ็กซ์เอ็มไอ และทำการทวนสอบได้ผลลัพธ์ ดังแสดงในภาพที่ 45

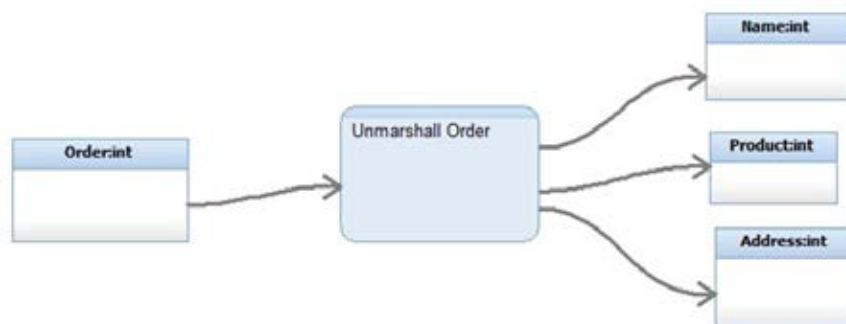


ภาพที่ 45 ผลลัพธ์จากเครื่องมือทวนสอบแผนภาพกิจกรรมในงานวิจัย [2]

5.6 การทดสอบตัวอย่างที่ 5

5.6.1 เข้าจากซอฟต์แวร์ Rhapsody Version 8.03

ในตัวอย่างทดสอบที่ 5 นั้น ใช้ชุดแผนภาพกิจกรรมทดสอบจาก Rhapsody Version 8.03 แสดงให้เห็นถึงแอ็กชันอินพุตและแอ็กชันเอาต์พุต ตามรูปแบบพื้นฐานแผนภาพกิจกรรมทั่วไป ดังแสดงในภาพที่ 46



ภาพที่ 46 ตัวอย่างแผนภาพกิจกรรมจากเครื่องมือทดสอบ

หลังจากสร้างแผนภาพกิจกรรมบนเครื่องมือ Rhapsody Version 8.03 แล้ว ให้นำออกข้อมูลในรูปแบบเอ็กซ์เอ็มไอไฟล์เพื่อเป็นอินพุตสำหรับเครื่องมือตัวแปลงแผนภาพกิจกรรม ตัวอย่างแสดงในภาพที่ 47

```

1 <?xml version="1.0" encoding="UTF-8"?><xml:XMI
xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:PredefinedTypes_profile="http://RhapsodyStandardModel.PredefinedTypes/schemas/PredefinedTypes_profile/_RAwYNSLEeKuAoLszjM2Jw/0"
xmlns:RHP="http://RhapsodyStandardModel.RhpProperties/schemas/RHP/_RAm_YNSLEeKuAoLszjM2Jw/0"
xmlns:RhapsodyProfile="http://RhapsodyStandardModel/schemas/RhapsodyProfile/_RAmYUN8LEeKuAoLszjM2Jw/0"
xmlns:TestCase4_TAGS="http://Project.TestCase4/schemas/TestCase4_TAGS/_RAnmcN8LEeKuAoLszjM2Jw/0" xmlns:UmlActions="http://Project/schemas/UmlActions/_RAoNgN8LEeKuAoLszjM2Jw/0"
xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore"
xmlns:uml="http://www.eclipse.org/uml2/3.0.0/UML" xmi:version="2.1"
xsi:schemaLocation="http://RhapsodyStandardModel.PredefinedTypes/schemas/PredefinedTypes_profile/_RAwYNSLEeKuAoLszjM2Jw/0 #_RLn9It8LEeKuAoLszjM2Jw
http://RhapsodyStandardModel.RhpProperties/schemas/RHP/_RAm_YNSLEeKuAoLszjM2Jw/0 #_RLn8-d8LEeKuAoLszjM2Jw
http://RhapsodyStandardModel/schemas/RhapsodyProfile/_RAmYUN8LEeKuAoLszjM2Jw/0 #_RLnVgd8LEeKuAoLszjM2Jw
http://Project.TestCase4/schemas/TestCase4_TAGS/_RAnmcN8LEeKuAoLszjM2Jw/0 #_RLgn198LEeKuAoLszjM2Jw http://Project/schemas/UmlActions/_RAoNgN8LEeKuAoLszjM2Jw/0
UmlActions.sbs.xmi#_RLh18N8LEeKuAoLszjM2Jw http://www.omg.org/spec/UML/20090901
http://www.eclipse.org/uml2/2.0.0/UML">
2 <uml:Model xmi:id="GUID+82df70aa-1377-422c-926e-7adfadce01f2" name="Project">
3   <elementImport xmi:type="uml:ElementImport" xmi:id="_RLgn0d8LEeKuAoLszjM2Jw"
importedElement="_RLmucd8LEeKuAoLszjM2Jw"
importingNamespace="GUID+82df70aa-1377-422c-926e-7adfadce01f2"/>
4   <elementImport xmi:type="uml:ElementImport" xmi:id="_RLgn0t8LEeKuAoLszjM2Jw"
importedElement="_RLn8998LEeKuAoLszjM2Jw"
importingNamespace="GUID+82df70aa-1377-422c-926e-7adfadce01f2"/>
5   <elementImport xmi:type="uml:ElementImport" xmi:id="_RLgn098LEeKuAoLszjM2Jw"
importedElement="_RLn9IN8LEeKuAoLszjM2Jw"
importingNamespace="GUID+82df70aa-1377-422c-926e-7adfadce01f2"/>
6   <packagedElement xmi:type="uml:Package"
href="Default.sbs.xmi#GUID+bbea9db7-4373-4fd2-9e28-4a5aa25cbea5"/>
7   <packagedElement xmi:type="uml:Package"
xmi:id="GUID+echf3cdf-be84-483a-b99f-41c44e15f6c8" name="TestCase4">
8     <packagedElement xmi:type="uml:Profile" xmi:id="_RLgn1d8LEeKuAoLszjM2Jw"
name="TestCase4_TAGS" metaclassReference="_RLh07N8LEeKuAoLszjM2Jw">
9     </packagedElement>
10    <packagedElement xmi:type="uml:Activity" xmi:id="_wNqolaFzEd6Eo8cPzRliQg"
name="Activity1">
11      <ownedParameter xmi:type="uml:Parameter" xmi:id="_RLhPad8LEeKuAoLszjM2Jw"
name="output1" direction="out"/>
12      <ownedParameter xmi:type="uml:Parameter" xmi:id="_RLhPat8LEeKuAoLszjM2Jw"
name="input"/>
13      <ownedParameter xmi:type="uml:Parameter" xmi:id="_RLhPA98LEeKuAoLszjM2Jw"
name="output2" direction="out"/>
14      <node xmi:type="uml:ForkNode" xmi:id="_wNqo4KFzEd6Eo8cPzRliQg" name="Fork"
activity="_wNqolaFzEd6Eo8cPzRliQg"/>
15      <node xmi:type="uml:ActivityParameterNode" xmi:id="_wNqo2aFzEd6Eo8cPzRliQg"
name="output1" activity="_wNqolaFzEd6Eo8cPzRliQg"
parameter="_RLhPad8LEeKuAoLszjM2Jw">
16        <upperBound xmi:type="uml:LiteralInteger" xmi:id="_RLhPBt8LEeKuAoLszjM2Jw"
value="1"/>
17      </node>
18      <node xmi:type="uml:ActivityParameterNode" xmi:id="_wNqo36FzEd6Eo8cPzRliQg"
name="input" activity="_wNqolaFzEd6Eo8cPzRliQg"
parameter="_RLhPat8LEeKuAoLszjM2Jw">
19        <upperBound xmi:type="uml:LiteralInteger" xmi:id="_RLhPCN8LEeKuAoLszjM2Jw"
value="1"/>
20      </node>
21      <node xmi:type="uml:ActivityParameterNode" xmi:id="_wNqo3KFzEd6Eo8cPzRliQg"
name="output2" activity="_wNqolaFzEd6Eo8cPzRliQg"
parameter="_RLhPA98LEeKuAoLszjM2Jw">
22        <upperBound xmi:type="uml:LiteralInteger" xmi:id="_RLhPct8LEeKuAoLszjM2Jw"
value="1"/>
23      </node>
24    </packagedElement>
25  </packagedElement>
26  <packagedElement xmi:type="uml:Profile"
href="UmlActions.sbs.xmi#GUID+8854472e-ce9a-473a-93d5-26db77eaa009"/>
27 </uml:Model>
28 </xml:XMI>

```

ภาพที่ 47 เอ็กซ์เอ็มไอไฟล์ที่ได้จากเครื่องมือทดสอบ

เอกสารเอ็มไอไฟล์ที่ถูกแปลงจากตัวแปลงแผนภาพกิจกรรม ให้ข้อมูลผลลัพธ์ ดังแสดงในภาพ

ที่ 48

```

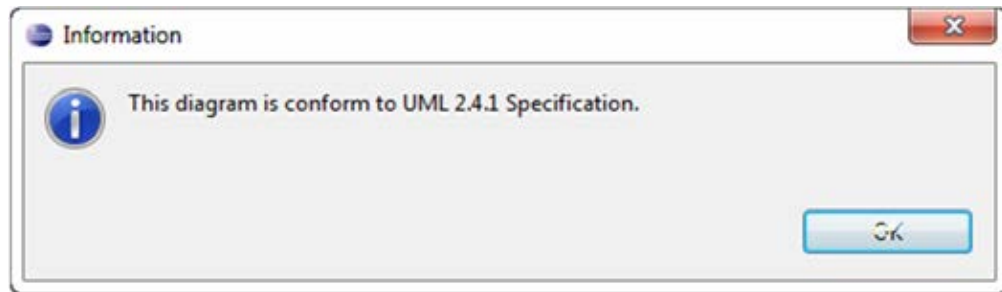
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <uml:Model xmlns:uml="http://www.omg.org/spec/UML/20090901"
  xmlns:xmi="http://schema.omg.org/spec/XMI/2.1" name="Activity Diagrams"
  xmi:id="_wNqoQgFzEd6Eo8cPzRliQg" xmi:version="2.1">
3 <node name="Unmarshall Order" xmi:id="_mhzFWSZCEeKSKePaylnoVA"
  xmi:type="uml:OpaqueAction">
4 <argument incoming="_mhzsbSZCEeKSKePaylnoVA" name="Call0.input0"
  xmi:id="_mhzsbSZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
5 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
6 </argument>
7 <result name="Call0.output0" outgoing="_mhzFbiZCEeKSKePaylnoVA"
  xmi:id="_mhzFbiZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
8 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
9 </result>
10 <result name="Call0.output1" outgoing="_mhzsYyZCEeKSKePaylnoVA"
  xmi:id="_mhzsYyZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
11 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
12 </result>
13 <result name="Call0.output2" outgoing="_mhzsaCZCEeKSKePaylnoVA"
  xmi:id="_mhzsaCZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
14 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
15 </result>
16 </node>
17 <node name="Order" xmi:id="_mhzFWiZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
18 <result name="Call1.output0" outgoing="_mhzsbSZCEeKSKePaylnoVA"
  xmi:id="_mhzsbSZCEeKSKePaylnoVA" xmi:type="uml:OutputPin">
19 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
20 </result>
21 </node>
22 <node name="Name" xmi:id="_mhzFXyZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
23 <argument incoming="_mhzFbiZCEeKSKePaylnoVA" name="Call2.input0"
  xmi:id="_mhzFbiZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
24 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
25 </argument>
26 </node>
27 <node name="Address" xmi:id="_mhzFZCZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
28 <argument incoming="_mhzsYyZCEeKSKePaylnoVA" name="Call3.input0"
  xmi:id="_mhzsYyZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
29 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
30 </argument>
31 </node>
32 <node name="Product" xmi:id="_mhzFaSZCEeKSKePaylnoVA" xmi:type="uml:CentralBufferNode">
33 <argument incoming="_mhzsaCZCEeKSKePaylnoVA" name="Call4.input0"
  xmi:id="_mhzsaCZCEeKSKePaylnoVA" xmi:type="uml:InputPin">
34 <type href="http://www.omg.org/spec/UML/20090901/UML.xmi#Integer"
  xmi:type="uml:PrimitiveType"/>
35 </argument>
36 </node>
37 <edge source="_mhzFWSZCEeKSKePaylnoVA" target="_mhzFXyZCEeKSKePaylnoVA"
  xmi:id="_mhzFbiZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
38 <weight value="1" xmi:id="_wNqpC3IzEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
39 </edge>
40 <edge source="_mhzFWSZCEeKSKePaylnoVA" target="_mhzFZCZCEeKSKePaylnoVA"
  xmi:id="_mhzsYyZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
41 <weight value="1" xmi:id="_wNqpC28zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
42 </edge>
43 <edge source="_mhzFWSZCEeKSKePaylnoVA" target="_mhzFaSZCEeKSKePaylnoVA"
  xmi:id="_mhzsaCZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
44 <weight value="1" xmi:id="_wNqpC98zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
45 </edge>
46 <edge source="_mhzFWiZCEeKSKePaylnoVA" target="_mhzFWSZCEeKSKePaylnoVA"
  xmi:id="_mhzsbSZCEeKSKePaylnoVA" xmi:type="uml:ObjectFlow">
47 <weight value="1" xmi:id="_wNqpC80zEd6Eo8cPzRliQg"
  xmi:type="uml:LiteralUnlimitedNatural"/>
48 </edge>
49 </packagedElement>
50 </uml:Model>
51

```

ภาพที่ 48 ผลลัพธ์เอกสารเอ็มไอของตัวแปลงแผนภาพกิจกรรม

5.6.2 สรุปผล

ผลลัพธ์ที่ได้ เมื่อนำไปทดสอบกับเครื่องมือที่พัฒนาในงานวิจัยที่ [2] เพื่อทดสอบแผนภาพกิจกรรม ผลปรากฏว่า เครื่องมือทดสอบแผนภาพกิจกรรมสามารถอ่านไฟล์เอ็กซ์เอ็มไอ และทำการทดสอบได้ผลลัพธ์ ดังแสดงในภาพที่ 49



ภาพที่ 49 ผลลัพธ์จากเครื่องมือทดสอบแผนภาพกิจกรรมในงานวิจัย [2]

บทที่ 6

สรุปผลการวิจัย และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอแนวทางและได้พัฒนา ตัวแปลงแผนภาพกิจกรรมสำหรับรูปแบบมาตรฐานเอ็กซ์เอ็มไอ โดยมีระเบียบวิธีการหลักๆ ประกอบด้วย 1) นำเข้าไฟล์เอ็กซ์เอ็มไอของแผนภาพกิจกรรม 2) ระบุความสัมพันธ์จากไฟล์นำเข้าและสกัดจัดเก็บข้อมูลโนดกับเส้นเชื่อมของแผนภาพ และ 3) นำออกไฟล์เอ็กซ์เอ็มไอซึ่งอยู่ในรูปแบบเดียวกันตามมาตรฐานที่กำหนดโดยนิยามเค้าร่างเอ็กซ์เอ็มแอลหรือเอ็กซ์เอสดีที่เฉพาะเจาะจงกับแต่ละแอปพลิเคชัน ระบบต้นแบบที่พัฒนาขึ้นจะช่วยให้แอปพลิเคชันสามารถรองรับการนำเข้าข้อมูลในรูปแบบเอ็กซ์เอ็มไอที่แตกต่างกันได้ ทำให้เพิ่มความสามารถในการแลกเปลี่ยนข้อมูลรูปแบบเอ็กซ์เอ็มไอ

การประเมินผลของงานวิจัยนี้ได้ทดลองเปรียบเทียบไฟล์เอ็กซ์เอ็มไอที่ได้จากเครื่องมือทดสอบ 5 ชนิด กับไฟล์เอ็กซ์เอ็มไอที่เป็นผลลัพธ์จาก ตัวแปลงแผนภาพกิจกรรมสำหรับรูปแบบมาตรฐานเอ็กซ์เอ็มไอ ผลปรากฏว่าสามารถป้องกันความสูญหายของข้อมูลระหว่างการแลกเปลี่ยนข้อมูลได้ทั้งหมด

6.2 ข้อจำกัด

1. งานวิจัยชิ้นนี้ยังไม่สามารถแปลงแผนภาพกิจกรรมที่มากกว่า 1 แผนภาพต่อ 1 ไฟล์เอ็กซ์เอ็มไอได้
2. ไฟล์แผนภาพกิจกรรมที่อยู่ในรูปแบบเอ็กซ์เอ็มไอ
3. ไฟล์นำเข้าเอ็กซ์เอ็มไอจะต้องมีโครงสร้างตามมาตรฐานที่โอเอ็มจีกำหนด
4. มาตรฐานรูปแบบผลลัพธ์จะต้องอยู่ในรูปแบบของเอ็กซ์เอ็มแอลสคีมา นามสกุลจะต้องเป็น .xsd เท่านั้น
5. ไฟล์เอ็กซ์เอ็มไอตามรูปแบบที่ผู้ใช้กำหนดเป็นมาตรฐาน
6. รูปแบบมาตรฐานเอ็กซ์เอ็มไอจะมีความเฉพาะเจาะจงและแตกต่างกันขึ้นอยู่กับ XSD ที่ผู้ใช้กำหนดขึ้น

6.3 แนวทางการวิจัยต่อ

- การแปลงแผนภาพกิจกรรมยังมีข้อจำกัดอยู่ คือ ไม่รองรับการแปลงไฟล์เอ็กซ์เอ็มไอที่มีข้อมูลแผนภาพกิจกรรมมากกว่า 1 แผนภาพกิจกรรมได้

รายการอ้างอิง

- [1] Object Management Group: OMG Unified Modeling Language (OMG UML) : Superstructure Version 2.3. 2010. Available from : <http://www.omg.org>
- [2] C. Kaewchinporn and Y. Limpiyakorn, "Enhancement of Action Description Language for UML Activity Diagram Review", International Journal of Software Engineering and Its Application,7, 2 (March, 2013).
- [3] Object Management Group: MOF 2 XMI Mapping Specification : Superstructure version 2.4. 2010. Available from : <http://www.omg.org>
- [4] C. Narkngam and Y. Limpiyakorn, "Rendering UML Activity Diagrams as a Domain Specific Language - ADL", Proceedings of 24th International Conference on Software Engineering and Knowledge Engineering, San Francisco Bay, USA, (Jul 2012): 724–729.
- [5] World Wide Web Consortium (W3C). XML Schema [Online] 2005, Available from : <http://www.3c.org/TR/xmlschema-0> [2011, Feb 4]
- [6] World Wide Web Consortium (W3C) : DOM and XML DOM, Available from : <http://www.3c.org>
- [7] A. Persson, H. Gustavsson, B. Lings, Bj, r. Lundell, A. Mattsson, U., and rlig, "OSS tools in a heterogeneous environment for embedded systems modelling: an analysis of adoptions of XMI" presented at the Proceedings of the fifth workshop on Open source software engineering, St. Louis, Missouri, 2005.
- [8] Jason E. R. ArgoUML [Online]. 2001. Available From: <http://argouml.tigris.org/> [2003, Jan 28]

ภาคผนวก

ภาคผนวก ก.

รายละเอียดเพิ่มเติมของข้อมูลที่ใช้ นำเข้าระบบ

ก.1. ส่วนการตรวจสอบความถูกต้องของเอ็กซ์เอ็มไอไฟล์ที่ออกแบบด้วยเครื่องมือ

ทดสอบ

การกำหนดรายละเอียดไฟล์แมนนิเฟสต์ต่างๆ ที่สำคัญของส่วนการแปลงแผนภาพกิจกรรมที่ออกแบบด้วยโอเพนซอร์สให้อยู่ในรูปแบบมาตรฐานมีดังนี้

ที่หน้า Dependencies เพิ่มปลั๊กอินที่ต้องการและแพ็กเกจนำเข้าเพิ่มเติมดังภาพที่ 24 ปลั๊กอินที่ต้องการ

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

แพ็กเกจนำเข้าเพิ่มเติม

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

จากภาพที่ 25 แสดงตัวอย่างสำหรับการเพิ่มเอ็กซ์ทีเอ็นชัน 2 ชนิด โดยมีรายละเอียดดังนี้

org.eclipse.ui.menus คือ เป็นส่วนขยายทำหน้าที่ในการจัดการกับเมนูต่าง ๆ ซึ่งตรงส่วนนี้จะทำให้เราสามารถเพิ่มเมนูการใช้งานของการแปลงแผนภาพกิจกรรม โดยการคลิกเมาส์ขวาและสั่งผ่านเมนูที่เราสร้างขึ้นได้ การเพิ่มเมนูของอีคลิป์ส์สามารถทำได้ 3 รูปแบบคือ เมนูด้านบน เมนูไอคอนด้านบน และเมนูที่เกิดขึ้น เมื่อคลิกขวาที่รายการไฟล์ใน Package Explorer ในส่วนนี้เอง งานวิจัยได้ทำการเพิ่มเมนูหนึ่งเข้าไป ชื่อว่า “XMI Transforms” คือเมนูที่เกิดขึ้น เมื่อคลิกที่ไฟล์นามสกุลเอ็กซ์เอ็มไอ เพื่อเลือกให้ทำการแปลงไฟล์เอกสารเอ็กซ์เอ็มไอที่เลือก และมีเมนูย่อยอีก 1 รายการคือ ” XMI 2.X Standard” สำหรับแปลงตามแม่แบบที่กำหนดเพื่อใช้กับงานวิจัย [2]

org.eclipse.ui.commands คือ ตัวเชื่อมระหว่างเมนูและไฟล์คลาส ในการพัฒนานั้น ในระบบที่พัฒนาขึ้นมีการแปลงรูปแบบรูปแบบเดียว ดังนั้น ใน org.eclipse.ui.commands จะมีคำสั่งและมีไฟล์คลาสในการเชื่อมต่อคือ FileReaderHandlerForXMI2Output.java

ผลลัพธ์ในการเพิ่มข้อมูลในไฟล์แมนนิเฟสต์นั้น สามารถตรวจสอบทั้งหมดได้ผ่านหน้า

plugin.xml โดยผลลัพธ์ที่ได้ของซอฟต์แวร์แสดงได้ดังภาพที่ 50

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <?eclipse version="3.4"?>
3  <plugin>
4  <extension
5      point="org.eclipse.ui.popupMenus">
6  <objectContribution
7      objectClass="org.eclipse.core.resources.IFile"
8      id="th.ac.chula.eng.cp.XMITransforms.contribution1">
9  <menu
10         id="th.ac.chula.eng.cp.XMITransforms.menu"
11         label="XMI Transforms">
12  <separator
13         name="separator">
14  </separator>
15  </menu>
16  <action
22  </action>
23  <action
24         class="th.ac.chula.eng.cp.xmitransforms.handlers.File
25         ReaderHandlerForXMI2Output"
26         enablesFor="1"
27         id="th.ac.chula.eng.cp.XMITransforms.FileReaderXMI2Po
28         pupMenu"
29         label="XMI 2.X Standard"
30         menubarPath="th.ac.chula.eng.cp.xmitransforms.menu/se
31         parator">
32  </action>
33  </objectContribution>
34  </extension>
35  </plugin>

```

ภาพที่ 50 ภาพแสดงตัวอย่าง plugin.xml ในงานวิจัย

ก.2. ไฟล์เอ็กซ์เอ็มแอลที่ใช้ในการนำเข้าไปสร้างเอกสารเอ็กซ์เอ็มไอ

ในส่วนการสร้างไฟล์เอกสารเอ็กซ์เอ็มไอที่มีข้อมูลนำเข้าในส่วนนี้เป็นไฟล์เอ็กซ์เอ็มแอล โดยจะมีโครงสร้างที่ได้มาจากออปเจกกราฟ ดังภาพที่ 51 ก่อนที่จะถูกนำเข้าสู่กระบวนการในการสร้างไฟล์เอ็กซ์เอ็มไอ

```

<?xml version="1.0" encoding="UTF-8" ?>
<ActivityDiagram>
  <node>
    <propertyid></propertyid>
    <propertynodetype></propertynodetype>
    <propertytext></propertytext>
    <edge>
      <propertyid></propertyid>
      <propertytarget></propertytarget>
      <propertysource></propertysource>
    </edge>
  </node>
</ActivityDiagram>

```

ภาพที่ 51 ภาพไฟล์เอ็กซ์เอ็มแอลก่อนนำเข้ากระบวนการสร้างไฟล์เอ็กซ์เอ็มไอ

ส่วนประกอบแท็กต่างๆภายในไฟล์นั้นใช้อธิบายโครงสร้างของกราฟที่ได้จากแผนภาพกิจกรรม โดยมีแท็กดังต่อไปนี้

- ActivityDiagram แท็กที่บอกรายละเอียดขององค์ประกอบภายในแผนภาพ
- Node แท็กที่บอกรายละเอียดขององค์ประกอบของโหนด ซึ่งใช้สื่อถึงสัญญาณต่างๆในแผนภาพกิจกรรมที่ไม่ใช่เส้นกระแส โดยจะมีแท็ก propertynodetype ที่ใช้ในการบ่งบอกถึงรายละเอียดประเภทของสัญญาณนั้น
- Propertyid คุณสมบัตินี้ใช้ตั้งขึ้นเพื่อใช้เป็นชื่อแทนในการเชื่อมความสัมพันธ์กัน
- Propertynodetype คุณสมบัตินี้บอกรายละเอียดประเภทของโหนดในกิจกรรม action node, fork, join, decision, merge, initial และ final
- Propertytext คุณสมบัตินี้ใช้ข้อความ
- Edge แท็กที่บอกรายละเอียดขององค์ประกอบของเส้นกระแส ซึ่งใช้สื่อถึงเส้นกระแสทั้งหมดภายในแผนภาพกิจกรรม
- Propertytarget คุณสมบัตินี้ของแท็ก edge ที่ใช้ในการบอกรายละเอียดของเส้นกระแส เป็นชื่อหรือรหัสของสิ่งที่อยู่ปลายทาง

- PropertySource คุณสมบัติของแท็ก edge ที่ใช้ในการบอกต้นทางของเส้น กระแส เป็นชื่อหรือรหัสของสิ่งที่อยู่ต้นทาง

ประวัติผู้เขียนวิทยานิพนธ์

นายธนกฤต เจริญรัตนกร เกิดเมื่อวันที่ 25 กันยายน พ.ศ. 2527 ที่จังหวัดสงขลา สำเร็จ การศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) สาขาวิทยาการคอมพิวเตอร์ คณะ วิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ในปีการศึกษา 2548 และเข้าศึกษาต่อในหลักสูตร วิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2554