การทดสอบความน่าเชื่อถือสำหรับกระแสกระบวนการด้วยการวิเคราะห์ต้นไม้ความผิดพลาด

นางสาวพิมพ์ทิพย์ ไพบูลย์เกษมสุทธิ์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
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 University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2558
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Reliability Tests for Process Flow with Fault Tree Analysis

Miss Pimthip Paiboonkasemsut

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Software Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2015

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การทดสอบความน่าเชื่อถือสำหรับกระแสกระบวนการด้วยการวิเคราะห์ต้นไม้ความผิดพลาด
โดย	นางสาวพิมพ์ทิพย์ ไพบูลย์เกษมสุทธิ์
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์)

.....กรรมการภายนอกมหาวิทยาลัย

(อาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

พิมพ์ทิพย์ ไพบูลย์เกษมสุทธิ์ : การทดสอบความน่าเชื่อถือสำหรับกระแสกระบวนการด้วยการวิเคราะห์ต้นไม้ความผิดพลาด (Reliability Tests for Process Flow with Fault Tree Analysis) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.ญาใจ ลิ่มปิยะกรณ์, 113 หน้า.

ความน่าเชื่อถือซอฟต์แวร์เป็นมิติหนึ่งของคุณภาพซอฟต์แวร์ การทดสอบความน่าเชื่อถือมีแนวโน้มสามารถตรวจพบความล้มเหลวส่วนมากที่จะเกิดขึ้นได้แต่เนิ่นๆก่อนการดำเนินงานจริง การวิเคราะห์ต้นไม้ความผิดพลาด (เอฟทีเอ) เป็นเทคนิคหนึ่งที่ถูกใช้อย่างแพร่หลายสำหรับการประเมินความปลอดภัยและความน่าเชื่อถือ ในทางทฤษฎี การวิเคราะห์ต้นไม้ความผิดพลาดจะสร้างแผนภาพสองอย่าง คือ แผนภาพต้นไม้ความสำเร็จ (เอสทีดี) และแผนภาพต้นไม้ความผิดพลาด (เอฟทีดี) โดยทั่วไป ก่อนที่จะประยุกต์ใช้เอฟทีเอ วิธีต้นไม้การจำแนกแบบมีเงื่อนไขจะถูกใช้สำหรับระบุเส้นทางที่ถูกต้อง และไม่ถูกต้อง เพื่อใช้สร้างกรณีทดสอบการทำงานและกรณีทดสอบความน่าเชื่อถือตามลำดับ ในงานวิจัยนี้ ตัวสร้างกรณีทดสอบ *เจเอฟทีเอ* ถูกพัฒนาขึ้นด้วยหลักการออกแบบสถาปัตยกรรมแบบโมเดล วิว คอนโทรลเลอร์ (เอ็มวีซี) และพัฒนาด้วยภาษาจาวา โดยใช้ไลบรารีเจกราฟเอกซ์ เพื่อช่วยการแสดงผลลัพธ์แผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข ประกอบด้วย ต้นไม้ความสำเร็จ และต้นไม้ความผิดพลาด ทั้งนี้ ระบบที่พัฒนาขึ้นทำงานโดยรับข้อมูลนำเข้าเป็นแผนภาพกิจกรรมยูเอ็มแอลในรูปแบบเอกซ์เอ็มไอที่ได้มาจากปลั๊กอินอีคลิปส์ ปาปิรุส มาร์ส ร่วมกับการกำหนดเส้นทางที่ระบบสามารถทำงานได้อย่างถูกต้อง ซึ่งเป็นข้อมูลนำเข้าจากผู้ใช้ จากนั้น แผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข แผนภาพต้นไม้ความสำเร็จ และแผนภาพต้นไม้ความผิดพลาด จะถูกสร้างเป็นผลลัพธ์จากการประมวลผล ระบบที่พัฒนาขึ้นในงานนี้จะช่วยลดต้นทุนค่าใช้จ่ายและความผิดพลาดของมนุษย์ในการสร้างกรณีทดสอบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2558

5771004221 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: FAULT TREE ANALYSIS / TEST CASE GENERATION / RELIABILITY / ACTIVITY DIAGRAM

PIMTHIP PAIBOONKASEMSUT: Reliability Tests for Process Flow with Fault Tree Analysis. ADVISOR: ASSOC. PROF. YACHAI LIMPIYAKORN, Ph.D., 113 pp.

Software reliability is one of the dimensions of software quality. Reliability testing tends to uncover earlier those failures which are most likely in actual operations. Fault Tree Analysis (FTA) is one of the techniques widely used for safety and reliability evaluations. Theoretically, FTA method would generate two diagrams: Success Tree Diagram (STD) and Fault Tree Diagram (FTD). Prior to applying FTA, the condition classification tree method will be generally carried out to identify the valid and invalid paths that shall be used for creating functionality test cases and reliability test cases, respectively. In this research, the test case generator, *JFTA*, is developed with Model-View-Controller (MVC) architectural design. The JFTA is implemented in Java and uses JGraphX library to help rendering the output diagrams of condition classification tree, namely STD, and FTD. The implemented system works with the input of UML activity diagrams in XMI format, generated by Papyrus Mars (Eclipse Plug-in). In addition to the activity diagram, the success goals are also required to be input by the user. Afterwards, Condition-Classification Tree diagram, Success Tree Diagram and Fault Tree diagram will be generated as the results. The implementation in this work would reduce the cost and human-error in test case construction.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Department:	Computer Engineering	Student's Signature	.....
Field of Study:	Software Engineering	Advisor's Signature	.....
Academic Year:	2015		

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จสมบูรณ์ลุล่วงได้ด้วยความอนุเคราะห์อย่างดียิ่งจาก รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้กรุณาให้ความรู้ คำแนะนำ ข้อเสนอแนะความคิดเห็นต่างๆ และตลอดจนช่วยแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่ เพื่อให้การทำวิทยานิพนธ์ฉบับนี้สมบูรณ์ ผู้จัยตระหนักถึงและมีความซาบซึ้งในความกรุณาอันดีจากอาจารย์ และขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณ ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล และอาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และญาติพี่น้อง ที่คอยให้กำลังใจและการสนับสนุนช่วยเหลือในด้านต่างๆ กราบขอบพระคุณอาจารย์ทุกท่านที่ได้อบรมสั่งสอบวิชาความรู้ให้ผู้วิจัยจนสามารถทำวิทยานิพนธ์นี้ได้สำเร็จ

สุดท้ายนี้ผู้วิจัยขอขอบคุณเพื่อนๆ ทุกคนที่คอยได้ให้กำลังใจและความช่วยเหลือต่างๆ ตลอดจนผู้มีพระคุณทุกท่านที่มิได้กล่าวถึง คุณประโยชน์และคุณค่าอันเกิดจากการศึกษาวิจัยครั้งนี้ ผู้วิจัยขอมอบแด่บิดา มารดา ครู อาจารย์ และผู้มีพระคุณทุกท่านด้วยความซาบซึ้งใจอย่างยิ่ง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฎ

สารบัญรูป ..... ฎ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์การวิจัย 2

1.3 ขอบเขตการวิจัย 2

1.4 ขั้นตอนการวิจัย 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์ 3

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4

2.1 ทฤษฎีที่เกี่ยวข้อง 4

2.1.1 การวิเคราะห์ต้นไม้ความผิดพลาด (Fault Tree Analysis : FTA) 4

2.1.2 เอกซ์เอ็มแอล (eXtensible Markup Language : XML) 6

2.1.2.1 เอกซ์เอ็มไอ (XML Metadata Interchange : XMI) 6

2.1.3 อีคลิปโมเดลลิ่งเฟรมเวิร์ค (Eclipse Modeling Framework : EMF) 7

2.1.4 ปาปิรุส มาร์ส (Papyrus Mars) 7

2.1.5 เจกราฟเอกซ์ (JGraphX) 7

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

2.1.6 แผนภาพกิจกรรม (Activity Diagram) 7
2.2 งานวิจัยที่เกี่ยวข้อง 9
2.2.1 Generating test cases from UML activity diagrams using the Condition-Classification Tree Method [14] 9
2.2.2 An Approach to Generate Safety Validation Test Cases from UML Activity Diagram [15] 10
บทที่ 3 แนวคิดและวิธีวิจัย 13
3.1 ภาพรวมแนวคิดงานวิจัย 13
3.1.1 ส่วนนำเข้าข้อมูลเมทาเดตาของแผนภาพกิจกรรม 13
3.1.2 สร้างแผนภาพต้นไม้จำแนกแบบมีเงื่อนไข 18
3.1.3 สร้างตารางกรณีทดสอบ 19
3.1.4 แปลงกรณีทดสอบเป็นแผนภาพต้นไม้ความสำเร็จและแผนภาพต้นไม้ความผิดพลาด 23
บทที่ 4 การออกแบบและพัฒนาระบบ 25
4.1 ข้อกำหนดการทำงานเบื้องต้นของระบบ 25
4.2 ความต้องการเชิงหน้าที่ (Functional Requirements) 26
4.2.1 แผนภาพยูสเคส (Use Case Diagram) 27
4.2.2 คำอธิบายแผนภาพยูสเคส (Use Case Descriptions) 29
4.3 การออกแบบสถาปัตยกรรมของระบบ 35
4.3.1 การแปลงข้อมูลนำเข้า 36
4.3.2 คอนโทรลเลอร์ - โมเดล 36
4.3.2.1 ข้อมูลนำเข้า 37
4.3.2.2 ต้นไม้การจำแนกแบบมีเงื่อนไข 37
4.3.2.3 กรณีทดสอบ 37

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

4.3.2.4 แผนภาพต้นไม้ 37
4.3.3 วิว 37
4.4 การพัฒนาระบบ 38
4.4.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนาระบบ 38
4.4.2 การพัฒนาวิธีการสร้างเมทาโมเดลของแผนภาพกิจกรรม 39
4.4.3 การพัฒนาส่วนต่อประสาน 42
4.4.3.1 ส่วนของการประมวลผลนำเข้าข้อมูล 43
4.4.3.2 ส่วนแสดงผลลัพธ์ที่ได้จากการประมวลผลของข้อมูลนำเข้า 46
บทที่ 5 การทดสอบและประเมินผลระบบ 48
5.1 การทดสอบระบบ 48
5.2 การประเมินระบบ 53
5.2.1 กรณีทดสอบที่ 1 โปรแกรมที่ประกอบด้วยแผนภาพกิจกรรมอย่างง่าย 53
5.2.2 กรณีทดสอบที่ 2 โปรแกรมที่ประกอบด้วยแผนภาพกิจกรรมที่มีทางสู่เป้าหมายความสำเร็จมากกว่า 1 ทาง 65
5.2.3 กรณีทดสอบที่ 3 โปรแกรมที่ประกอบด้วย แผนภาพกิจกรรมที่มี Fork node และ Join node เป็นส่วนประกอบ 71
5.2.4 กรณีทดสอบที่ 4 กรณีทดสอบที่ 4 โปรแกรมที่ประกอบด้วยความซับซ้อนที่มี จุดตัดสินใจ 10 จุด มีทางสู่เป้าหมายความสำเร็จมากกว่า 1 ทาง และมี Fork Join เป็นส่วนประกอบ 76
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ 87
6.1 สรุปผลการวิจัย 87
6.2 ข้อจำกัด 87
6.3 แนวทางการวิจัยต่อ 87
รายการอ้างอิง 88

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

ภาคผนวก..... 90
ภาคผนวก ก. 91
ภาคผนวก ข. 96
ภาคผนวก ค. 102
ภาคผนวก ง..... 108
ประวัติผู้เขียนวิทยานิพนธ์ 113

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

ตารางที่ 1 ข้อมูลที่สกัดได้จากไฟล์เอกสารเอกซ์เอ็มไอ 15
ตารางที่ 2 ข้อมูลที่สกัดได้จากไฟล์เอกสารเอกซ์เอ็มไอ (ต่อ) 16
ตารางที่ 3 ข้อมูลที่สกัดได้จากไฟล์เอกสารเอกซ์เอ็มไอ (ต่อ) 17
ตารางที่ 4 สรุปกรณีทดสอบที่ได้จากแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข 21
ตารางที่ 5 ความต้องการเชิงหน้าที่ของระบบสร้างแผนภาพต้นไม้ความผิดพลาด 26
ตารางที่ 6 คำอธิบายยูสเคส Import Activity Diagram Metadata (XMI) file 29
ตารางที่ 7 คำอธิบายยูสเคส Extract XMI with DOM 29
ตารางที่ 8 คำอธิบายยูสเคส Convert XMI to UML node 30
ตารางที่ 9 คำอธิบายยูสเคส Build Node Relationship 30
ตารางที่ 10 คำอธิบายยูสเคส Indexing Node 31
ตารางที่ 11 คำอธิบายยูสเคส Choose Success Goal (Process Goal) 31
ตารางที่ 12 คำอธิบายยูสเคส Analyze Condition-Classification Tree Model 32
ตารางที่ 13 คำอธิบายยูสเคส Analyze Success Tree 32
ตารางที่ 14 คำอธิบายยูสเคส Analyze Fault Tree 33
ตารางที่ 15 คำอธิบายยูสเคส View Condition-Classification Tree Model Diagram 33
ตารางที่ 16 คำอธิบายยูสเคส View Success Tree Diagram 34
ตารางที่ 17 คำอธิบายยูสเคส View Fault Tree Diagram 34
ตารางที่ 18 กรณีทดสอบระบบ 49

สารบัญรูป

รูปที่ 1 บางส่วนของสัญลักษณ์ที่ใช้ในแผนภาพต้นไม้ความผิดพลาด [4] 5
รูปที่ 2 การกำหนดพฤติกรรมของระบบด้วยบัพควบคุมต่าง ๆ 8
รูปที่ 3 ตัวอย่างการพิจารณาจุดตัดสินใจบนแผนภาพกิจกรรม [14] 9
รูปที่ 4 กรอบแนวคิดในการสร้างกรณีทดสอบจากแผนภาพกิจกรรม [14] 9
รูปที่ 5 การแปลงต้นไม้ความสำเร็จให้เป็นต้นไม้ความผิดพลาด [15] 11
รูปที่ 6 ตัวอย่างการแปลงแผนภาพกิจกรรมเป็นต้นไม้ความสำเร็จด้วย OR gate [15] 12
รูปที่ 7 ตัวอย่างการแปลงเป็นต้นไม้ความสำเร็จจากการเปลี่ยนสถานะในแผนภาพกิจกรรม [15] 12
รูปที่ 8 ภาพรวมของงานวิจัย 13
รูปที่ 9 ตัวอย่างเอกซ์เอ็มไอที่ได้จากแผนภาพกิจกรรม 14
รูปที่ 10 โครงสร้างต้นไม้ที่สร้างจากจุดตัดสินใจ [14] 18
รูปที่ 11 แผนภาพกิจกรรมของแอปพลิเคชันการเริ่มต้นเกม 19
รูปที่ 12 ต้นไม้การจำแนกของแอปพลิเคชันการเริ่มต้นเกมโดยพิจารณาจากจุดตัดสินใจ 19
รูปที่ 13 ตารางกรณีทดสอบของแอปพลิเคชันการเริ่มต้นเกม 20
รูปที่ 14 การลดกรณีทดสอบที่มีความซ้ำซ้อน 21
รูปที่ 15 ตัวอย่างโค้ดที่ใช้ในการสร้างกรณีทดสอบ 22
รูปที่ 16 ต้นไม้ความสำเร็จของแอปพลิเคชันการเริ่มต้นของเกม [15] 24
รูปที่ 17 ต้นไม้ความผิดพลาดของแอปพลิเคชันการเริ่มต้นของเกม [15] 24
รูปที่ 18 แผนภาพยูสเคสของระบบสร้างแผนภาพต้นไม้ความผิดพลาด 28
รูปที่ 19 การออกแบบสถาปัตยกรรมการทำงานของระบบ 36
รูปที่ 20 เริ่มต้นการสร้างโปรเจคใหม่ 39
รูปที่ 21 การสร้างปาปิรุสโปรเจค 40
รูปที่ 22 เลือกประเภทยูเอ็มแอลคอร์ 40

รูปที่ 23 การใส่ชื่อโปรเจค 41
รูปที่ 24 เลือกประเภทแผนภาพยูเอ็มแอล 41
รูปที่ 25 หน้าจอปลั๊กอินปาปิรุส มาร์ส 42
รูปที่ 26 หน้าจอของระบบสร้างแผนภาพต้นไม้ความผิดพลาด 43
รูปที่ 27 หน้าจอในส่วนของการนำเข้าไฟล์เอกซ์เอ็มไอ 43
รูปที่ 28 หน้าจอแสดงการแจ้งเตือนกรณีการนำเข้าไฟล์ที่ผิดพลาด 44
รูปที่ 29 หน้าจอแสดงการเลือกไฟล์เอกสารเอกซ์เอ็มไอ 44
รูปที่ 30 หน้าจอแสดงการนำเข้าข้อมูล 45
รูปที่ 31 หน้าจอแสดงการเลือกเป้าหมายของกระบวนการที่ถูกต้อง 45
รูปที่ 32 หน้าจอแสดงผลกรณีทดสอบของระบบแสดงแผนภาพต้นไม้ความผิดพลาด 46
รูปที่ 33 หน้าจอแสดงผลต้นไม้การจำแนกแบบมีเงื่อนไข 46
รูปที่ 34 หน้าจอแสดงผลต้นไม้ความสำเร็จ 47
รูปที่ 35 หน้าจอแสดงผลต้นไม้ความผิดพลาด 47
รูปที่ 36 แผนภาพกิจกรรมของ Game Initialization 54
รูปที่ 37 เมทาเดตาของ Game Initialization 55
รูปที่ 38 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 1 56
รูปที่ 39 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 2 56
รูปที่ 40 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 3 57
รูปที่ 41 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 4 57
รูปที่ 42 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 5 58
รูปที่ 43 CCTM จากตัวอย่างที่ 5.2.1 ที่ทำด้วยมือ 59
รูปที่ 44 หน้าจอแสดงผลลัพธ์ CCTM 59
รูปที่ 45 CCTM จากตัวอย่างที่ 5.2.1 ที่สร้างด้วยระบบ 60
รูปที่ 46 STD จากตัวอย่างที่ 5.2.1 ที่ทำด้วยมือ 61

รูปที่ 47 หน้าจอแสดงผลลัพธ์ STD 61
รูปที่ 48 STD จากตัวอย่างที่ 5.2.1 ที่สร้างด้วยระบบ 62
รูปที่ 49 FTD จากตัวอย่างที่ 5.2.1 ที่ทำด้วยมือ 62
รูปที่ 50 หน้าจอแสดงผลลัพธ์ FTD 63
รูปที่ 51 FTD จากตัวอย่างที่ 5.2.1 ที่สร้างด้วยระบบ 64
รูปที่ 52 แผนภาพกิจกรรมการซื้อไมล์สะสมแบบง่าย 66
รูปที่ 53 เมทาเดตาของแผนภาพกิจกรรมการซื้อไมล์สะสมแบบง่าย 67
รูปที่ 54 CCTM จากตัวอย่างที่ 5.2.2 ที่ทำด้วยมือ 67
รูปที่ 55 CCTM จากตัวอย่างที่ 5.2.2 ที่สร้างด้วยระบบ 68
รูปที่ 56 STD จากตัวอย่างที่ 5.2.2 ที่ทำด้วยมือ 69
รูปที่ 57 STD จากตัวอย่างที่ 5.2.2 ที่สร้างด้วยระบบ 69
รูปที่ 58 FTD จากตัวอย่างที่ 5.2.2 ที่ทำด้วยมือ 70
รูปที่ 59 FTD จากตัวอย่างที่ 5.2.2 ที่ทำสร้างด้วยระบบ 70
รูปที่ 60 แผนภาพกิจกรรมตัวอย่างแบบมีส่วนประกอบ Fork และ Join 72
รูปที่ 61 เมทาเดตาของแผนภาพกิจกรรมตัวอย่างแบบมีส่วนประกอบ Fork และ Join 72
รูปที่ 62 CCTM จากตัวอย่างที่ 5.2.3 ที่ทำด้วยมือ 73
รูปที่ 63 CCTM จากตัวอย่างที่ 5.2.3 ที่สร้างด้วยระบบ 73
รูปที่ 64 STD จากตัวอย่างที่ 5.2.3 ที่ทำด้วยมือ 74
รูปที่ 65 STD จากตัวอย่างที่ 5.2.3 ที่สร้างด้วยระบบ 75
รูปที่ 66 FTD จากตัวอย่างที่ 5.2.3 ที่ทำด้วยมือ 75
รูปที่ 67 FTD จากตัวอย่างที่ 5.2.3 ที่สร้างด้วยระบบ 76
รูปที่ 68 แผนภาพกิจกรรมการซื้อไมล์สะสมแบบซับซ้อน 79
รูปที่ 69 เมทาเดตาของแผนภาพกิจกรรมการซื้อไมล์สะสมแบบซับซ้อน 80
รูปที่ 70 CCTM จากตัวอย่างที่ 5.2.4 ที่สร้างด้วยระบบ 81

รูปที่ 71 CCTM จากตัวอย่างที่ 5.2.4 ที่สร้างด้วยระบบ (ต่อ) 82
รูปที่ 72 Mileage Purchase STD 1 83
รูปที่ 73 Mileage Purchase STD 2 83
รูปที่ 74 Mileage Purchase STD 3 83
รูปที่ 75 Mileage Purchase STD 4 84
รูปที่ 76 Mileage Purchase FTD 1 84
รูปที่ 77 Mileage Purchase FTD 2 84
รูปที่ 78 Mileage Purchase FTD 3 85
รูปที่ 79 Mileage Purchase FTD 4 85
รูปที่ 80 Mileage Purchase FTD 5 85
รูปที่ 81 Mileage Purchase FTD 6 85
รูปที่ 82 Mileage Purchase FTD 7 86
รูปที่ 83 Mileage Purchase FTD 8 86
รูปที่ 84 โปรแกรมอีคลิปส์โมเดลลิ่งทูล เวอร์ชันมาร์ส 1 (4.5.1) 91
รูปที่ 85 หน้าจอดาวน์โหลดโปรแกรมอีคลิปส์โมเดลลิ่งทูล 92
รูปที่ 86 หน้าจอ Developer Builds 92
รูปที่ 87 หน้าจอ Select a mirror 93
รูปที่ 88 หน้าจอ Eclipse Installer 94
รูปที่ 89 หน้าจอติดตั้งอีคลิปส์โมเดลลิ่งทูล 95
รูปที่ 90 หน้าเว็บไซต์ปลั๊กอินเสริมปาปิรุส 96
รูปที่ 91 หน้าจอ Initial Welcome Page of Eclipse 97
รูปที่ 92 หน้าจอติดตั้งปลั๊กอินเสริมปาปิรุส 97
รูปที่ 93 หน้าจอดาวน์โหลดปลั๊กอินเสริมอีคลิปส์ ปาปิรุส มาร์ส 98
รูปที่ 94 หน้าจอติดตั้งซอฟต์แวร์ Available Software 99

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 95 หน้าจอรายละเอียดซอฟต์แวร์ Install Details 100
รูปที่ 96 หน้าจอ Review Licenses 100
รูปที่ 97 หน้าจอ Installing Software 101
รูปที่ 98 หน้าจอแสดงปลั๊กอินเสริมที่ถูกติดตั้ง 101
รูปที่ 99 สร้างโปรเจคแผนภาพกิจกรรมโดยปาปิรุส 102
รูปที่ 100 หน้าจอเลือก Papyrus Project 103
รูปที่ 101 หน้าจอ Diagram Language 103
รูปที่ 102 หน้าจอ Project path and the model name 104
รูปที่ 103 หน้าจอ Initialization information 105
รูปที่ 104 หน้าจอการสร้างแผนภาพกิจกรรมด้วยปาปิรุส มาร์ส 106
รูปที่ 105 สัญลักษณ์ของแผนภาพกิจกรรมบนโปรแกรมเสริมปาปิรุส 1 106
รูปที่ 106 สัญลักษณ์ของแผนภาพกิจกรรมบนโปรแกรมเสริมปาปิรุส 2 107
รูปที่ 107 เมทาเดตาที่ได้จากแผนภาพกิจกรรม 107
รูปที่ 108 หน้าจอการทำงานของระบบสร้างแผนภาพต้นไม้ความผิดพลาด 108
รูปที่ 109 หน้าจอนำเข้าเมทาเดตาของแผนภาพกิจกรรม 108
รูปที่ 110 หน้าจอแสดงการเลือกเมทาเดตาของแผนภาพกิจกรรม 109
รูปที่ 111 หน้าจอแสดงข้อมูลเมทาเดตาที่นำเข้า 109
รูปที่ 112 หน้าจอแสดงการเลือกเป้าหมายที่ได้จากเมทาเดตาของแผนภาพกิจกรรม 110
รูปที่ 113 หน้าจอแสดงผลลัพธ์ของระบบ 110
รูปที่ 114 หน้าจอแสดงผลลัพธ์ต้นไม้การจำแนกแบบมีเงื่อนไข 111
รูปที่ 115 หน้าจอแสดงผลลัพธ์ต้นไม้ความสำเร็จ 111
รูปที่ 116 หน้าจอแสดงผลลัพธ์ต้นไม้ความผิดพลาด 112

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันเครื่องมือในการสร้างกรณีทดสอบถูกใช้อย่างแพร่หลายในหลายหน่วยงานหรือองค์กร ทั้งนี้กรณีทดสอบมีหลายประเภท แต่ละประเภทมีเป้าหมายและจุดประสงค์ของการทดสอบที่แตกต่างกันออกไป ยกตัวอย่าง เช่น การทดสอบความน่าเชื่อถือ (Reliability Testing) การทดสอบการทำงานของฟังก์ชัน (Functionality Testing) การทดสอบการทำงานของฟังก์ชันเป็นการทดสอบความถูกต้องหรือเส้นทางที่ถูกต้อง (Valid paths) ส่วนการทดสอบความน่าเชื่อถือเป็นการทำสอบความผิดพลาดหรือเส้นทางที่ไม่ถูกต้อง (Invalid paths) [1, 2] ทั้งนี้จุดประสงค์ในการทดสอบความน่าเชื่อถือคือการกำหนดความน่าเชื่อถือสำหรับตัวผลิตภัณฑ์หรือซอฟต์แวร์

เมื่อกล่าวถึงการทดสอบเพื่อมุ่งเน้นความน่าเชื่อถือ การวิเคราะห์ต้นไม้ความผิดพลาด (Fault Tree Analysis) [3] ได้ถูกใช้กันอย่างกว้างขวางสำหรับการทดสอบซอฟต์แวร์ที่ต้องการความน่าเชื่อถือ ในทางเทคนิค การวิเคราะห์เหตุการณ์จากบนลงล่าง (Top-down event) เพื่อสร้างกรณีทดสอบความผิดพลาดสำหรับการทดสอบซอฟต์แวร์ในสภาวะไม่ปกติ โดยเฉพาะการสร้างต้นไม้ความผิดพลาดจะถูกสร้างมาจากกระบวนการทำงานในสภาวะปกติ ซึ่งการทำงานสภาวะปกติจะนิยมถูกนำมาใช้สำหรับการทดสอบการทำงานของฟังก์ชันโดยการทำงานนี้สามารถนำเสนอในรูปแบบของแผนภาพต้นไม้ความสำเร็จ (Success Tree Diagram) [4, 5]

อย่างไรก็ตาม กระบวนการพัฒนาซอฟต์แวร์จะไม่สมบูรณ์แบบ หากขาดการทดสอบความน่าเชื่อถือ ทั้งนี้การสร้างเครื่องมือที่สามารถสร้างกรณีทดสอบซอฟต์แวร์ทั้งด้านความน่าเชื่อถือและการทำงานของฟังก์ชันจะเป็นการขับเคลื่อนการพัฒนาซอฟต์แวร์ได้เป็นอย่างดียิ่งขึ้น

งานวิจัยนี้ได้ทำการศึกษาแนวทางการสร้างแผนภาพต้นไม้ความผิดพลาดอย่างอัตโนมัติ และได้พัฒนาเป็นระบบต้นแบบในการสร้างแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข ต้นไม้ความสำเร็จและต้นไม้ความผิดพลาด ทั้งนี้เพื่อลดเวลาและการใช้ทรัพยากรในการสร้างกรณีทดสอบต่อไป

1.2 วัตถุประสงค์การวิจัย

เพื่อนำเสนอวิธีการและพัฒนาเครื่องมือสำหรับสร้างกรณีทดสอบแบบและแผนภาพต้นไม้ความผิดพลาดแบบอัตโนมัติ จากเมทาเดตาแผนภาพกิจกรรม (Activity Diagram Metadata) เพื่อทดสอบความน่าเชื่อถือของระบบซอฟต์แวร์ ปรับปรุงกระบวนการตรวจสอบซอฟต์แวร์ การวิเคราะห์ออกแบบระบบให้มีประสิทธิภาพมากยิ่งขึ้น และลดระยะเวลาในการออกแบบการทดสอบการทำงานของระบบด้วยการวิเคราะห์ต้นไม้ความผิดพลาด

1.3 ขอบเขตการวิจัย

1. วิธีการที่นำเสนอใช้กับแผนภาพกิจกรรมที่นำเข้าในรูปแบบของเอกซ์เอ็มไอ (XMI)
2. เครื่องมือที่ใช้ประกอบในการทำวิจัย คือปาปิรุส มาร์ส (Papyrus Mars) 1.1.4 สำหรับการสร้างแผนภาพกิจกรรม
3. สนับสนุนเฉพาะรูปแบบของจุดตัดสินใจ (Decision node) ของแผนภาพกิจกรรม (Activity Diagram)
4. งานวิจัยนี้สนับสนุนเฉพาะข้อมูลนำเข้าที่เป็นภาษาอังกฤษเท่านั้น
5. ข้อความนำเข้าต้องอยู่ในรูปแบบของเอกซ์เอ็มแอล XML
6. กรณีทดสอบและผลลัพธ์จากการทดสอบจะต้องอยู่ในรูปแบบตามที่กำหนดเท่านั้น

1.4 ขั้นตอนการวิจัย

1. ศึกษาและทำความเข้าใจเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวข้อง และกำหนดขอบเขตของงานวิจัย
2. ศึกษาการใช้เครื่องมือที่ใช้สำหรับการพัฒนาโปรแกรม
3. ศึกษาวิธีการใช้เครื่องมือ Papyrus Mars 1.1.4 for UML 2
4. วิเคราะห์และกำหนดระเบียบวิธีวิจัย
5. ออกแบบ ตั้งสมมติฐาน ที่เกี่ยวข้องกับงานวิจัย
6. พัฒนาระบบสำหรับสร้างกรณีทดสอบและแผนภาพต้นไม้ความผิดพลาดจากเมทาเดตาของแผนภาพกิจกรรม
7. ทดสอบและประเมินผลงานวิจัย
8. สรุปผลงานวิจัยและข้อเสนอแนะ
9. ตีพิมพ์ผลงานทางวิชาการ
10. จัดทำวิทยานิพนธ์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้วิธีการและเครื่องมือสำหรับสร้างกรณีทดสอบเพื่อความน่าเชื่อถือด้วยการวิเคราะห์ต้นไม้ความผิดพลาด
2. ได้วิธีการและเครื่องมือสำหรับการสร้างต้นไม้การจำแนก
3. ได้เครื่องมือสนับสนุนการปรับปรุงกระบวนการซอฟต์แวร์ ซึ่งช่วยลดเวลาและการใช้ทรัพยากรในการสร้างกรณีทดสอบสำหรับระบบที่ต้องการความน่าเชื่อถือในขั้นตอนการออกแบบระบบ

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บท โดยแบ่งดังต่อไปนี้ บทที่ 1 บทนำกล่าวถึงที่มาและความสำคัญ วัตถุประสงค์ ขอบเขตการวิจัย ขั้นตอน และประโยชน์ที่คาดว่าจะได้รับจากงานวิจัยนี้ บทที่ 2 ได้กล่าวถึงทฤษฎีที่ใช้ในงานวิจัยและงานวัยจัยที่เกี่ยวข้องกับงานวิจัยนี้ บทที่ 3 กล่าวถึงแนวคิดและวิธีวิจัย ภาพรวมของงานวิจัย บทที่ 4 กล่าวถึงการออกแบบและพัฒนาระบบ ข้อกำหนดการทำงานเบื้องต้นของระบบ ความต้องการการทำงานเชิงหน้าที่ คำอธิบายระบบ สถาปัตยกรรมระบบ รวมไปถึงการพัฒนาระบบ บทที่ 5 กล่าวถึงการทดสอบและประเมินระบบ และบทที่ 6 สรุปผลการวิจัย ข้อเสนอแนะและแนวทางในการพัฒนาต่อ

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นเป็นความทางวิชาการเรื่อง “Reliability Tests for Process Flow with Fault Tree Analysis” โดย พิมพ์ทิพย์ ไพบูลย์เกษมสุทธิ์ และ ญาใจ ลิ่มปิยะกรณ์ ในรายงานสืบเนื่องการประชุมวิชาการระดับนานาชาติ The 2nd International Conference on Information Science and Security (ICISS 2015) สาธารณรัฐเกาหลี ระหว่างวันที่ 14-16 ธันวาคม 2558 หน้า 177-180

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

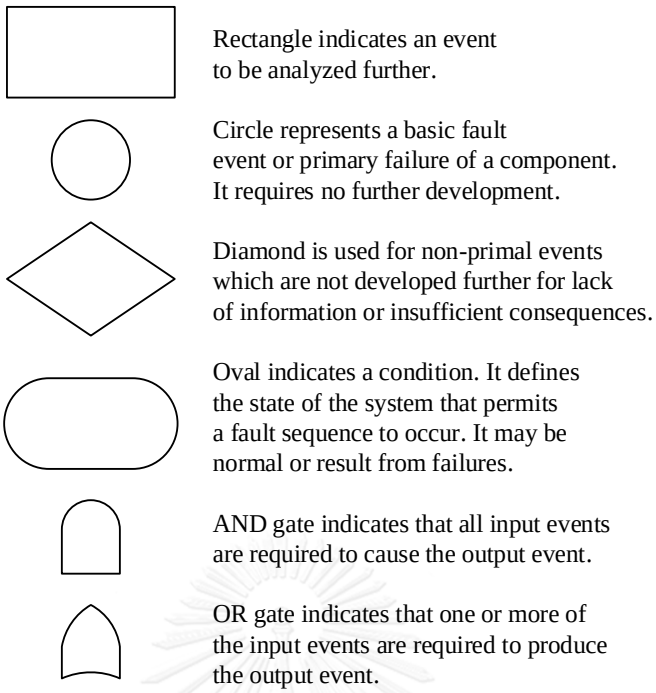
2.1.1 การวิเคราะห์ต้นไม้ความผิดพลาด (Fault Tree Analysis : FTA)

การวิเคราะห์ต้นไม้ความผิดพลาด [3] คือเครื่องมือสำหรับการวิเคราะห์ จำลองแผนภาพ และประเมินเส้นทางของความผิดพลาดในระบบ โดยการวิเคราะห์ด้วยวิธีนี้จะมีกระบวนการที่มีประสิทธิภาพในการประเมินความเสี่ยงของระบบ ในปัจจุบันนี้มีหลายองค์กรและหลายบุคคลที่มีความคุ้นเคยเครื่องนี้ชนิดนี้ ซึ่งเครื่องมือนี้มักถูกนำไปใช้เพื่อประเมินความปลอดภัยและความน่าเชื่อถือ อีกทั้งในหลายส่วนงานได้ระบุเครื่องนี้เป็นที่ต้องการสำหรับการรองรับการผลิต

การวิเคราะห์ต้นไม้ความผิดพลาดกลายเป็นเครื่องที่ถูกใช้กันอย่างแพร่หลาย เครื่องมือการเคราะห์นี้ได้ถูกนำไปใช้ในการปรับปรุงระบบต่าง ๆ ตั้งแต่ปี ค.ศ. 1961 และมีการร่วมมือจากหลากหลายบุคลากร เครื่องมือวิเคราะห์ต้นไม้ความผิดพลาดได้กลายเป็นเครื่องมือชิ้นสำคัญในการออกแบบระบบและพัฒนา

มโนทัศน์พื้นฐานของการวิเคราะห์ต้นไม้ความผิดพลาด คือ แนวคิดที่ต้องการแปลงความผิดพลาดในระบบใช้งานจริงออกมาอยู่ในรูปแบบของแผนภาพและแบบจำลองเชิงตรรกะซึ่งจะอำนวยความสะดวกในการหาความสัมพันธ์และสาเหตุของปัญหา ในแต่ละส่วนเชิงตรรกะของแบบจำลองมีความสามารถในการประเมินเชิงคุณภาพและปริมาณ การวิเคราะห์ต้นไม้ความผิดพลาดตั้งอยู่บนพื้นฐานของทฤษฎีความน่าเชื่อถือโดยการใช้ พีชคณิตเชิงตรรกะและทฤษฎีความน่าจะเป็น การตั้งชุดของกฎแบบง่ายๆจะสามารถนำไปใช้สำหรับการวิเคราะห์ระบบที่มีความซับซ้อน และระบบมีความสัมพันธ์ที่ซับซ้อนระหว่างอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ได้

หลักการทำงานเบื้องต้นของต้นไม้ความผิดพลาดคือการแปลงพฤติกรรมการทำงานที่ผิดพลาดของระบบในสถานการณ์จริงมาอยู่ในรูปแบบของแผนภาพจำลอง และแบบจำลองเชิงตรรกะ โดยที่แผนภาพที่แสดงสามารถอธิบายปัญหาของข้อผิดพลาดด้วยสัญลักษณ์ที่แสดงในรูปที่ 1



รูปที่ 1 บางส่วนของสัญลักษณ์ที่ใช้ในแผนภาพต้นไม้ความผิดพลาด [4]

ต้นไม้ความผิดพลาดมีลักษณะที่อธิบายเหตุการณ์การทำงานที่ไม่ถูกต้อง ซึ่งเหตุการณ์นี้จะถูกแสดงอยู่ในส่วนบนสุดโดยมีต้นเหตุเป็นสิ่งที่ทำให้เกิดเหตุการณ์นี้หรือกล่าวได้ว่า รากของต้นไม้ความผิดพลาดจะแสดงถึงต้นเหตุของเหตุการณ์ที่ทำให้การทำงานไม่ถูกต้องหรือไม่สำเร็จนั้น ๆ

การวิเคราะห์ต้นไม้ความผิดพลาดเป็นประยุกต์เงื่อนไขต่างๆ เพื่อที่จะนำไปสู่เหตุการณ์ที่แสดงอยู่บนสุดหรือเหตุการณ์ที่โปรแกรมไม่ทำงานได้อย่างสำเร็จ ซึ่งในเราสามารถประยุกต์ใช้การวิเคราะห์นี้เพื่อหาการทำงานในสภาวะที่ไม่ปกติในระบบซอฟต์แวร์ได้

การสร้างต้นไม้ความผิดพลาด และทำการประเมินเป็นกระบวนการที่ค่อนข้างง่ายและตรงไปตรงมาอย่างไรก็ตามเมื่อต้นไม้มีขนาดใหญ่ขึ้นและซับซ้อนขึ้น มันจะเป็นการยากที่จะแก้ได้ ความสามารถในการประเมินต้นไม้ความผิดพลาดนั้นแปรผันกับขนาด ความซับซ้อน และความสามารถของคอมพิวเตอร์

การวิเคราะห์ต้นไม้ความผิดพลาดเริ่มต้นที่ปี ค.ศ. 1961 และกล่าวได้ว่า H.A. Watson ได้เป็นบิดาของการวิเคราะห์ต้นไม้ความผิดพลาด และ Dave Hassel เป็นเจ้าพ่อของการวิเคราะห์ต้นไม้ความผิดพลาด Watson of Bell Labs ได้ประดิษฐ์การวิเคราะห์ต้นไม้ความผิดพลาด ใน

ขณะเดียวกันบริษัท Boeing ได้เล็งเห็นถึงประโยชน์และได้นำมาใช้อย่างแพร่หลาย รวมถึงการให้คำปรึกษาโครงงานต่าง ๆ ในการใช้การวิเคราะห์ต้นไม้ความผิดพลาด

อัลกอริทึมและซอฟต์แวร์ได้ถูกนำมาวิจัยกันอย่างมากจนถึงปัจจุบันนี้ โดยส่วนใหญ่จะเป็นการพัฒนาอัลกอริทึมเพื่อให้สามารถใช้งานการวิเคราะห์ต้นไม้ความผิดพลาดในคอมพิวเตอร์ส่วนบุคคลได้ในราคาถูก

การวิเคราะห์ต้นไม้ความผิดพลาดได้กลายมาเป็นเครื่องที่มีประโยชน์สำหรับงานด้านความปลอดภัย การประเมินความเสี่ยง การวิเคราะห์เหตุการณ์ที่ระบบทำงานผิดพลาด ความน่าเชื่อถือและอื่นๆ การวิเคราะห์ต้นไม้ความผิดพลาดจึงได้เป็นที่รู้จักและถูกมานำใช้ในระดับนานาชาติ [3-5]

2.1.2 เอกซ์เอ็มแอล (eXtensible Markup Language : XML)

เทคโนโลยีสารสนเทศเป็นสิ่งที่เข้ามาใช้ในชีวิตประจำวันและเป็นส่วนหนึ่งของการทำงานเอกซ์เอ็มแอล (XML) [6] เป็นภาษาที่ใช้ในการแสดงผลข้อมูล พัฒนาโดย W3C (World Wide Web Consortium) [7] มีบทบาทสำคัญในการแลกเปลี่ยนความหลากหลายของข้อมูลบนเว็บไซต์และในที่อื่นๆ เปลี่ยนแปลงข้อมูลโดยได้ใช้ฟอร์มตามมาตรฐานเอชทีเอ็มแอล (Hyper Text Markup Language : HTML) โดยเอกซ์เอ็มแอลจะทำงานได้ดีร่วมกับเอชทีเอ็มแอล

เอกซ์เอ็มแอลเป็นส่วนหนึ่งของ SGML (Standard Generalised Markup Language) ซึ่งเป็นข้อกำหนดในการสร้างเอกสารในรูปแบบอิเล็กทรอนิกส์ที่ถูกกำหนดโดย W3C และจะถูกใช้ในบราวเซอร์ตั้งแต่ IE 4.0 ขึ้นไป

เป้าหมายการออกแบบเอกซ์เอ็มแอลจะเน้นความเรียบง่าย ทั่วไป และการใช้งานผ่านอินเตอร์เน็ต เป็นส่วนขยายของเอกสารเอชทีเอ็มแอลในหลายๆระบบสามารถรองรับข้อมูลเอกซ์เอ็มแอล (XML Data) ได้ ซึ่งจะอยู่ในรูปแบบเอพีไอ-ส่วนต่อประสานโปรแกรมประยุกต์ (Application Program Interface : API) สามารถนำมาพัฒนาเพื่อการประมวลผลข้อมูลเอกซ์เอ็มแอลได้

2.1.2.1 เอกซ์เอ็มไอ (XML Metadata Interchange : XMI)

งานวิจัยนี้จะใช้ข้อมูลนำเข้าเป็นเอกซ์เอ็มไอ [8] โดยจะอยู่ในรูปแบบของเอกซ์เอ็มแอลและมีรูปแบบมาตรฐานรองรับเพื่อสนับสนุนการแลกเปลี่ยนข้อมูลของแอปพลิเคชันต่างๆ โดยลักษณะของรูปแบบดังกล่าวนี้จะสามารถอธิบายได้ถึงส่วนประกอบ คุณสมบัติ และสามารถเชื่อมโยงถึงกันในรูปแบบของเอกซ์เอ็มแอล

2.1.3 อีคลิปโมเดลลิ่งเฟรมเวิร์ค (Eclipse Modeling Framework : EMF)

หรือเรียกโดยย่อว่า อีเอ็มเอฟ อีเอ็มเอฟเป็นโอเพนซอร์สซึ่งถูกพัฒนาภายใต้อีคลิปโมเดลลิงค์โพรเจกต์ (Eclipse Modeling Project : EMP) [9] เป็นรูปแบบโครงสร้างเชิงโมเดลที่สนับสนุบสนุนการพัฒนาระบบ การพัฒนาซอฟต์แวร์ด้วยระบบโมเดลซึ่งมีคุณสมบัติในการสร้างโค้ด สำหรับสร้างเครื่องมือที่ช่วยพัฒนาโปรแกรมและแอปพลิเคชันอื่นๆ บนพื้นฐานของโมเดลโครงสร้าง (Structured Data Model) จากรูปแบบมาตราฐานที่ได้ถูกอธิบายในรูปแบบของไฟล์เอ็กซ์เอ็มไอ (XMI) อีเอ็มเอฟได้สร้างเครื่องมือเพื่อสนับสนุนการสร้าง กลุ่มของจาวาคลาส สำหรับรูปแบบนั้นๆ อีกทั้งยังสนับสนุนคลาสการแปลงซึ่งสามารถดูและสั่งการได้อีกด้วย

2.1.4 ปาปิรุส มาร์ส (Papyrus Mars)

ปาปิรุส มาร์ส [10] คือเครื่องมือโอเพนส์ซอต์ทางวิศวกรรมศาสตร์ที่ใช้โมเดลเป็นพื้นฐาน เป็นโปรแกรมเสริมสำหรับอีคลิป Eclipse เครื่องมือกราฟฟิกสำหรับยูเอ็มแอล (UML 2) ซึ่งถูกนิยามโดยโอเอ็มจี (Object Management Group : OMG) [11] เป็นเครื่องมือยูเอ็มแอลที่ช่วงอำนวยความสะดวกในการปรับแต่งเครื่องมือเพื่อสร้างแผนภาพ

ปาปิรุส มาร์ส สามารถนำมาประยุกต์ใช้ได้ทุกแผนภาพยูเอ็มแอล (UML Diagram) [12] เช่น แผนภาพคลาส (Class Diagram) แผนภาพวัตถุ (Object Diagram) แผนภาพยูสเคส (Use case Diagram) แผนภาพแสดงสถานะ (State machine Diagram) แผนภาพลำดับ (Sequence Diagram) และแผนภาพกิจกรรม (Activity Diagram) เป็นต้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

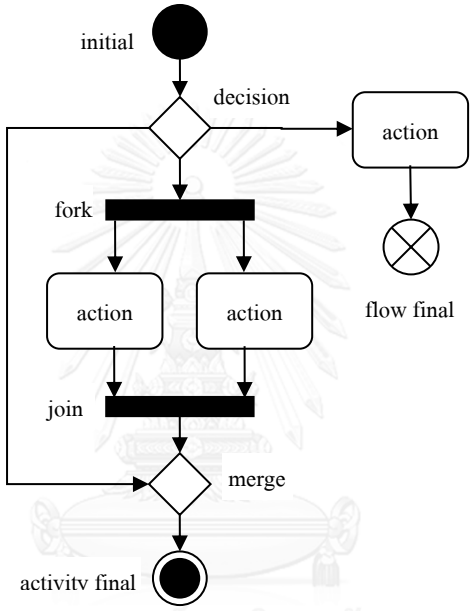
2.1.5 เจกราฟเอกซ์ (JGraphX)

เจกราฟเอกซ์ คือ จาวาสวิงไลบรารี (Java Swing Library) เป็นเครื่องมือกราฟฟิกอินเทอร์เฟสสำหรับการวาดกราฟ ภายใต้ใบอนุญาตของบีเอสดี (BSD license) เจกราฟเอกซ์มีฟังก์ชันในการแสดงผลและสร้างความสัมพันธ์ระหว่างโหนด (Node) และเส้นกราฟ (Edge) โดยใช้ภาษาจาวาในการควบคุม และแสดงผลส่วนติดต่อกับผู้ใช้งานซึ่งสวิงเป็นส่วนประกอบหนึ่งในภาษาจาวาที่ใช้สำหรับสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface) โดยสวิงจะเป็นส่วนย่อยต่างๆ ในการสร้างส่วนต่อประสาน ไม่ว่าจะเป็น ตัวอักษร เส้น ปุ่ม หรือรูปภาพ เป็นต้น [13]

2.1.6 แผนภาพกิจกรรม (Activity Diagram)

แผนภาพกิจกรรม จะมีลักษณะแบบเดียวกันกับผังงาน (Flowchart) เป็นแผนภาพที่ใช้ในการอธิบายกิจกรรมต่างๆ ที่เกิดขึ้น ในลักษณะของกระแสงาน (Workflow) ถูกนำมาใช้อย่างแพร่หลายเพื่ออธิบายพฤติกรรมของระบบในโดเมนต่างๆ การแสดงขั้นตอนการทำงานของระบบ

โดยเฉพาะอย่างยิ่งในโดเมนของวิศวกรรมซอฟต์แวร์ ซึ่งมักนิยมใช้แผนภาพกิจกรรมสำหรับออกแบบกระบวนการสำหรับองค์กรและการพัฒนาซอฟต์แวร์ โดยใช้ในการอธิบายตรรกะกระบวนการ กระบวนการทางธุรกิจ (Business process) และกระแสงาน (Workflow) ต่างๆ ทั้งกระบวนการที่ทำงานเป็นลำดับและทำงานแบบขนาน [11, 12] แผนภาพกิจกรรมหนึ่ง ๆ จะประกอบด้วยลำดับและเงื่อนไขต่างๆ ที่ใช้ในการควบคุมพฤติกรรมของระบบ โดยมีสายงานควบคุม (Control flow) และสายงานวัตถุ (Object flow) อธิบายลำดับของการกระทำ (Action) และมีบัพควบคุม (Control node) อธิบายพฤติกรรมของระบบ ดังแสดงในรูปที่ 2



รูปที่ 2 การกำหนดพฤติกรรมของระบบด้วยบัพควบคุมต่าง ๆ

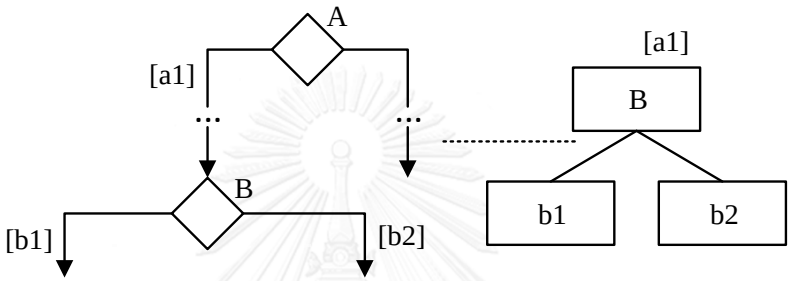
ส่วนประกอบหลักของแผนภาพกิจกรรม ประกอบด้วย

- Action : กิจกรรม
- Control flow : สายงานควบคุม
- Initial node : โหนดแสดงจุดเริ่มต้นการทำงาน
- Final-flow node : โหนดแสดงจุดสิ้นสุดการทำงาน
- Decision node : โหนดแสดงการตัดสินใจ
- Fork node และ Join node : แสดงการทำงานพร้อมกันของหลายกระบวนการที่เกิดขึ้นและจบพร้อมกัน

2.2 งานวิจัยที่เกี่ยวข้อง

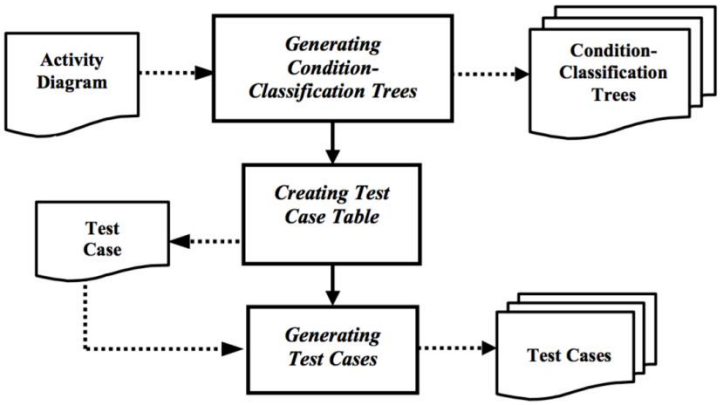
2.2.1 Generating test cases from UML activity diagrams using the Condition-Classification Tree Method [14]

บทความงานวิจัยนี้ได้นำเสนอเทคนิคการใช้กระบวนการสร้างกรณีทดสอบแผนภาพกิจกรรม โดยใช้วิธีการสร้างแบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไข CCTM ซึ่งย่อมาจาก Condition-Classification Tree Model วิธีดังกล่าวจะสร้างแผนภาพต้นไม้จำแนกจากจุดตัดสินใจของซอฟต์แวร์ที่ได้จากแผนภาพกิจกรรมยูเอ็มแอล (UML Activity Diagram)



รูปที่ 3 ตัวอย่างการพิจารณาจุดตัดสินใจบนแผนภาพกิจกรรม [14]

ตัวอย่างการพิจารณาจุดตัดสินใจบนแผนภาพกิจกรรมดังแสดงในรูปที่ 3 แสดงให้เห็นว่าแผนภาพกิจกรรมมีจุดตัดสินใจคือ จุดบี (B) เมื่อนำมาสร้างเป็นแบบจำลองต้นไม้จำแนกหรือซีซีทีเอ็มจะสามารถเกิดขึ้นได้สองกรณีคือ กรณี b1 และ กรณี b2 ซึ่งจะถูกนำมาใช้ในการสร้างต้นไม้จำแนก (Condition-Classification Tree) ซึ่งประกอบด้วยขั้นตอนหลักๆ 3 ขั้นตอน (รูปที่ 4) ดังนี้



รูปที่ 4 กรอบแนวคิดในการสร้างกรณีทดสอบจากแผนภาพกิจกรรม [14]

1. วิเคราะห์และสร้างต้นไม้การจำแนกแบบมีเงื่อนไข ขั้นตอนนี้จะทำการวิเคราะห์ แผนภาพกิจกรรมที่อยู่ในรูปแบบยูเอ็มแอลเพื่อจะสร้างต้นไม้การจำแนก
2. สร้างตารางกรณีทดสอบ โดยการนำต้นไม้การจำแนกที่ได้จากข้อ 1 มาวางเรียงกันเพื่อสร้างตารางทดสอบ จากนั้นทำการระบุเงื่อนไขแต่ของละต้นและสร้างเส้นแถวของตารางเพื่อใช้ในการกำหนดกรณีทดสอบ
3. สร้างกรณีทดสอบ โดยขั้นตอนนี้จะกำหนดกรณีทดสอบจากแถวของตาราง

แบบจำลองนี้เป็นวิธีหนึ่งที่ใช้เทคนิคการทดสอบแบบกล่องดำ (Black Box Testing) จะช่วยให้กรณีทดสอบที่สร้างขึ้นมีประสิทธิภาพและลดเวลาในการทำการวิเคราะห์เอกสารความต้องการ

แนวทางงานวิจัยนี้ได้ประยุกต์งานวิจัยข้างต้นในการนำแบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไขมาสกัดพิจารณาจุดตัดสินใจ เพื่อนำจุดตัดสินใจที่ได้ไปสร้างแผนภาพ

2.2.2 An Approach to Generate Safety Validation Test Cases from UML Activity Diagram [15]

บทความงานวิจัยนี้ได้นำเสนอเกี่ยวกับการตรวจสอบโมเดลเกี่ยวข้องกับการสร้างโมเดลของระบบซอฟต์แวร์ และการวิเคราะห์โมเดลเพื่อการค้นหาและลบข้อผิดพลาด ซึ่งในที่นี่ วิธีการวิเคราะห์ต้นไม้ความผิดพลาดได้ถูกพิจารณานำมาใช้ในการตรวจสอบโมเดลได้อย่างประสิทธิภาพ ซึ่งวิธีนี้จะทำการแปลงข้อผิดพลาดของระบบให้อยู่ในรูปแบบของโมเดลจำลองทั้งนี้เพื่อเป็นการระบุเหตุการณ์ความเป็นไปได้

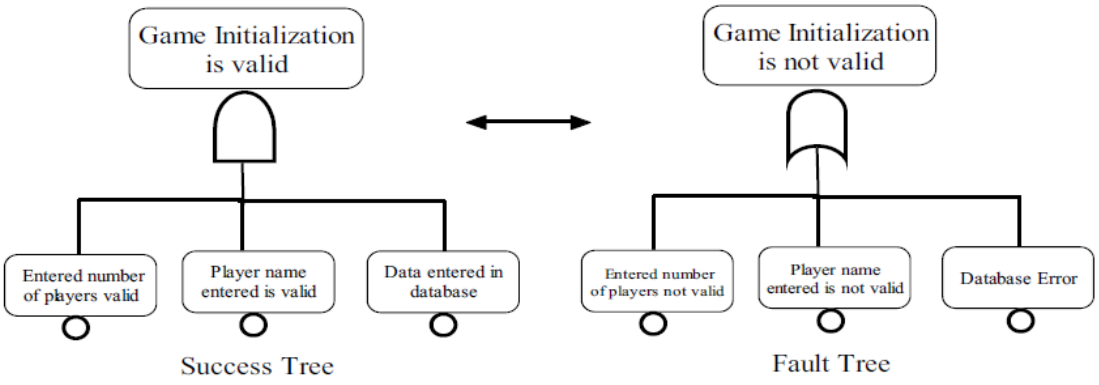
ทั้งหมดที่จะทำให้เกิดข้อผิดพลาดได้ วิธีนี้สามารถถูกนำมาใช้อย่างมีประสิทธิภาพในการวิเคราะห์ จำลองรูปแบบ และประเมินข้อผิดพลาดในระบบ

วิธีการวิเคราะห์ต้นไม้ความผิดพลาดเป็นการวิเคราะห์โดยอิงจากเหตุการณ์ข้างบนสุดเป็นหลัก โดยที่เหตุการณ์ที่ถูกวางไว้ข้างบนนั้นจะเป็นเหตุการณ์ที่ไม่ผิดพลาดซึ่งเกิดจากการรวมกันของหลายๆ เหตุการณ์ในโหนดลูก พื้นฐาน ในทางทฤษฎีของวิธีการวิเคราะห์ต้นไม้ความผิดพลาดจะระบุเหตุการณ์พื้นฐานที่สามารถนำไปสู่เหตุการณ์ที่ทำให้เกิดความผิดพลาดในชั้นบนสุดได้

การวิเคราะห์ต้นไม้ความผิดพลาดของระบบซอฟต์แวร์ สามารถนำมาเป็นเครื่องมือในการวิเคราะห์เหตุการณ์ที่สามารถนำไปสู่เหตุการณ์ขั้นสูงสุดหรือเหตุการณ์ที่ทำให้เกิดความผิดพลาดนั้นเอง โดยวิธีนี้จะถูกนำมาใช้ในรูปแบบของต้นไม้ ซึ่งจะถูกเรียกว่า ต้นไม้ความผิดพลาด ซึ่งแผนภาพนี้จะประกอบไปด้วยสัญลักษณ์ที่ระบุถึงประเภทสามประเภท นั้นคือ เหตุการณ์พื้นฐาน เหตุการณ์ขั้นกลาง และลอจิกเกต (logic gate) โดยที่เหตุการณ์พื้นฐาน และเหตุการณ์ขั้นกลางนั้น แสดงถึงต้นเหตุ

ของเหตุการณ์นั้นบนสุดหรือก็คือเหตุการณ์ที่ไม่ปกติ ในขณะเดียวกันลอจิกเกตมีหน้าที่แสดงถึงความสัมพันธ์ที่จำเป็นที่จะให้เกิดเหตุการ์ขั้นสูงต่อไป โดยที่ลอจิกเกตพื้นฐานที่ถูกนำมาใช้จะมีด้วยกันสองแบบคือ ออร์เกต (OR gate) และ แอนด์เกต (AND gate)

ในขณะเดียวกัน ต้นไม้ความสำเร็จได้ถูกสร้างขึ้นโดยมีลักษณ์เหมือนต้นไม้แห่งความผิดพลาด ซึ่งมี เหตุการณ์พื้นฐาน เหตุการณ์ขั้นกลาง และลอจิกเกต (logic gate) แต่มีข้อแตกต่างกันนั่นคือ เหตุการณ์ขั้นสูงสุดของต้นไม้ความสำเร็จจะเป็นเหตุการณ์ที่ทำให้ระบบทำงานได้อย่างถูกต้อง ซึ่งเรานิยมนำมาใช้การตรวจสอบความถูกต้องของการทำงานในระบบซอฟต์แวร์ ดังนั้น แผนภาพทั้งสองจะถูกนำมาใช้การวิเคราะห์ข้อผิดพลาดและความถูกต้องของระบบ ตัวอย่างของแผนภาพทั้งสองจะถูกแสดงในรูปที่ 5



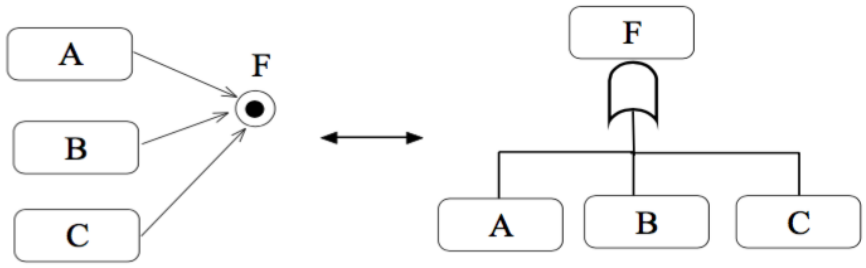
รูปที่ 5 การแปลงต้นไม้ความสำเร็จให้เป็นต้นไม้ความผิดพลาด [15]

Chulalongkorn University

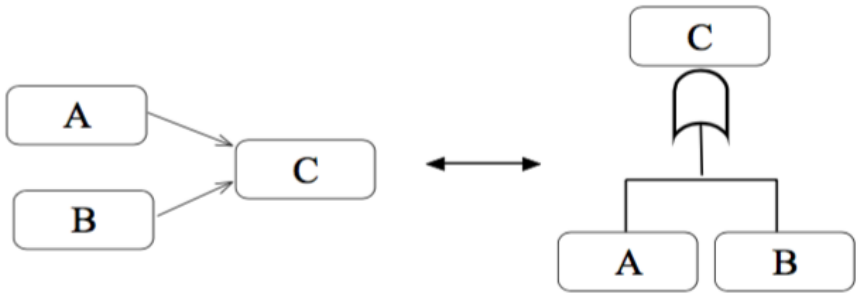
วิธีการที่นำเสนอของงานวิจัยนี้ เป็นการแปลงแผนภาพกิจกรรมให้กลายเป็นต้นไม้ความสำเร็จ เมื่อได้ต้นไม้แห่งความสำเร็จแล้ว จะถูกนำไปแปลงเป็นต้นไม้ความผิดพลาดและจะถูกนำไปทดสอบต่อไป

ตัวอย่างอย่างการแปลงแผนภาพกิจกรรมเป็นต้นไม้ความสำเร็จ

1. รูปที่ 6 อธิบายการแปลงแผนภาพกิจกรรมเป็นต้นไม้ความสำเร็จ โดยมีลักษณะเป็นออร์เกต โดยที่ F จะเป็นจริงก็ต่อเมื่อ A หรือ B หรือ C เป็นจริง
2. รูปที่ 7 อธิบายการเปลี่ยนสถานะแผนภาพกิจกรรมจะถูกแปลงเป็นข้อมูลนำเข้าของออร์เกตในแผนภาพต้นไม้



รูปที่ 6 ตัวอย่างการแปลงแผนภาพกิจกรรมเป็นต้นไม้ความสำเร็จด้วย OR gate [15]



รูปที่ 7 ตัวอย่างการแปลงเป็นต้นไม้ความสำเร็จจากการเปลี่ยนสถานะในแผนภาพกิจกรรม [15]

ในวิธีการวิจัยนี้จะแสดงให้เห็นว่า กฎการสร้างต้นไม้ได้ครอบคลุมหลายกรณีที่สามารถเกิดขึ้นได้ในแผนภาพกิจกรรมเมื่อเทียบกับงานวิจัยอื่นแล้วจะพบว่าวิธีนี้สามารถนำมาใช้เป็นเครื่องมือในการสร้างกรณีทดสอบได้อย่างมีประสิทธิภาพ อย่างไรก็ตามผู้วิจัยยังให้ความเห็นถึงข้อด้อยกับงานวิจัยนี้ว่าการสร้างแผนภาพต้นไม้ความผิดพลาดนั้นยังจำเป็นต้องใช้ทรัพยากรมนุษย์ซึ่งนั่นหมายถึงการระดมสมองและเพิ่มค่าใช้จ่ายในการพัฒนาระบบซอฟต์แวร์ เนื่องจากการพิจารณาในการแผนภาพกิจกรรมต้องใช้ความละเอียดเป็นอย่างมากมิเช่นนั้นแล้วอาจจะเกิดข้อผิดพลาดได้

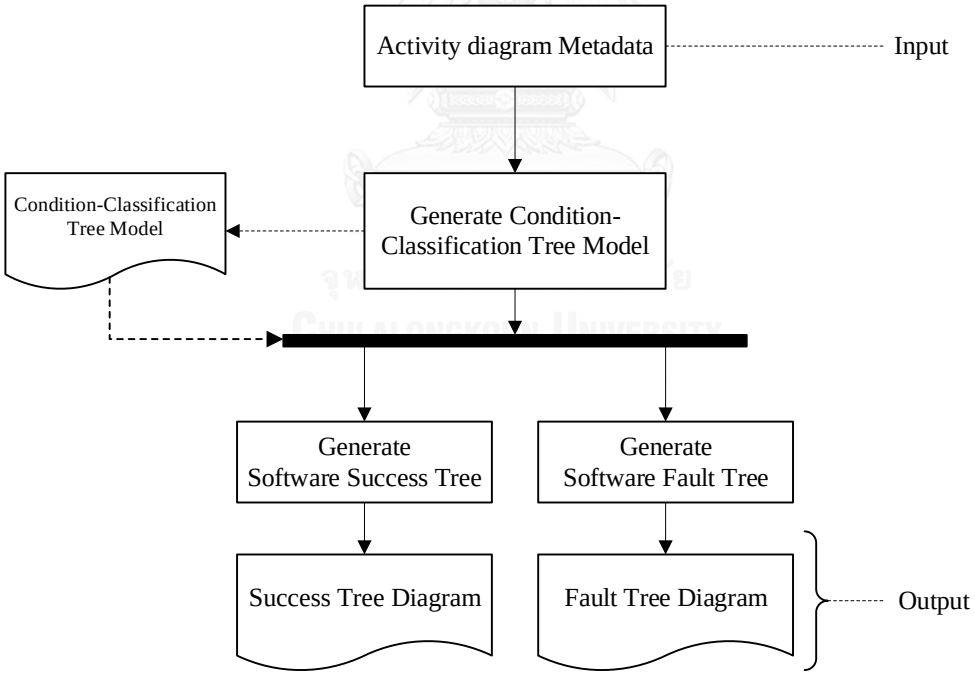
การทดสอบความคงทนจะทดสอบระบบด้วยกรณีทดสอบแบบปกติและแบบพิเศษ ถึงกระนั้นการสร้างกรณีทดสอบในการสร้างข้อผิดพลาดนั้นสามารถทำได้ยาก ต้นไม้ความผิดพลาดจึงถูกมาใช้เพื่อการสร้างกรณีทดสอบและระบุถึงความสัมพันธ์ของข้อมูลนำเข้าและเงื่อนไข ซึ่งการทดสอบในรูปแบบนี้สามารถนำไปเพิ่มความน่าเชื่อถือของระบบซอฟต์แวร์ได้อย่างมีประสิทธิภาพ ข้อดีอีกประการของวิธีการนี้คือการระบุความผิดพลาดของระบบซอฟต์แวร์ได้ในกระบวนการขั้นต้นของการพัฒนาซอฟต์แวร์ซึ่งทำให้การพัฒนาสามารถการแก้ไขหรือป้องกันได้ เนื่องจากถ้าความผิดพลาดพื้นฐานไม่เกิดขึ้นความผิดพลาดของระบบย่อมไม่เกิดขึ้นเช่นเดียวกัน

บทที่ 3
แนวคิดและวิธีวิจัย

3.1 ภาพรวมแนวคิดงานวิจัย

งานวิจัยนี้นำเสนอแนวทางและเครื่องมือสำหรับการสร้างกรณีทดสอบแบบอัตโนมัติจากแผนภาพกิจกรรมซึ่งอยู่ในรูปแบบเมทาเดตา โดยจะแบ่งการทำงานหลักๆ ออกเป็น 4 ส่วน คือ 1) ส่วนนำเข้าข้อมูลเมทาเดตาของแผนภาพกิจกรรม (Activity Diagram Metadata) 2) สร้างแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข 3) การสร้างกรณีทดสอบด้วยแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข 4) แปลงกรณีทดสอบที่ได้จากแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไขให้เป็นแผนภาพต้นไม้ความสำเร็จและแผนภาพต้นไม้ความผิดพลาด

ภาพรวมแนวคิดงานวิจัยนี้แสดงดังรูปที่ 8 รายละเอียดการสร้างกรณีทดสอบอัตโนมัติจากแผนภาพกิจกรรมในรูปแบบของยูเอ็มแอลเมทาเดตา

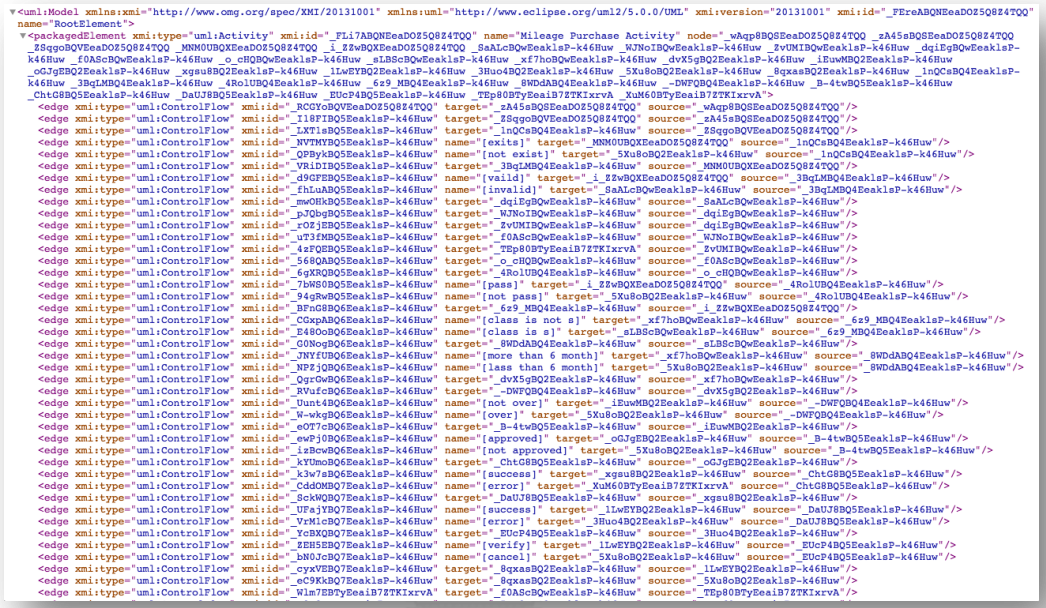


รูปที่ 8 ภาพรวมของงานวิจัย

3.1.1 ส่วนนำเข้าข้อมูลเมทาเดตาของแผนภาพกิจกรรม

ข้อมูลนำเข้าของงานวิจัยนี้เป็นเมทาเดตาที่สกัดได้จากแผนภาพกิจกรรม ซึ่งเริ่มแรกจะทำการ

สร้างแผนภาพกิจกรรมโดยใช้โปรแกรมโมเดลลิ่งเฟรมเวิร์ค เช่น ปาปิรุส จากนั้นนำแผนภาพกิจกรรมยูเอ็มแอลที่ได้มาแปลงให้เป็นภาษาตรฐานสำหรับยูเอ็มแอลซึ่งอยู่ในรูปแบบเอกซ์เอ็มไอ ตัวอย่างเอกซ์เอ็มไอที่ได้จากแผนภาพกิจกรรมยูเอ็มแอลดังรูปที่ 9 เพื่อทำการวิเคราะห์และสร้างต้นไม้การจำแนกแบบมีเงื่อนไข



```
<uml:Model xmlns:xmi="http://www.omg.org/spec/XMI/20131001" xmlns:uml="http://www.eclipse.org/uml2/5.0.0/UML" xmi:version="20131001" xmi:id="_FEreABQNEeaDOZ5Q8Z4TQQ"
 name="RootElement">
  <packagedElement xmi:type="uml:Activity" xmi:id="_FLi7ABQNEeaDOZ5Q8Z4TQQ" name="Mileage Purchase Activity" node="_wAqp8BQSEeaDOZ5Q8Z4TQQ _zA45sBQSEeaDOZ5Q8Z4TQQ
   _ZSqqoBQVEeaDOZ5Q8Z4TQQ _MNM0UBQXEeaDOZ5Q8Z4TQQ _i_ZZwBQXEeaDOZ5Q8Z4TQQ _SaALcBQwEeaklsP-k46Huw _WJNoIBQwEeaklsP-k46Huw _ZvUMIBQwEeaklsP-k46Huw _dqiEgBQwEeaklsP-
   k46Huw _f0AScBQwEeaklsP-k46Huw _o_cHQBQwEeaklsP-k46Huw _sLBScBQwEeaklsP-k46Huw _xf7hoBQwEeaklsP-k46Huw _dvX5gBQ2EeaklsP-k46Huw _iEuwMBQ2EeaklsP-k46Huw
   _oGJgEBQ2EeaklsP-k46Huw _xgsu8BQ2EeaklsP-k46Huw _1LwEYBQ2EeaklsP-k46Huw _3Huo4BQ2EeaklsP-k46Huw _5Xu8oBQ2EeaklsP-k46Huw _8qxasBQ2EeaklsP-k46Huw _1nQCsBQ4EeaklsP-
   k46Huw _3BqLMBQ4EeaklsP-k46Huw _4RolUBQ4EeaklsP-k46Huw _6z9_MBQ4EeaklsP-k46Huw _8WDdABQ4EeaklsP-k46Huw _-DWFQBQ4EeaklsP-k46Huw _B-4twBQ5EeaklsP-k46Huw
   _ChtG8BQ5EeaklsP-k46Huw _DaUJ8BQ5EeaklsP-k46Huw _EUcP4BQ5EeaklsP-k46Huw _TEp80BTyEeaiB7ZTKIxrvA _XuM60BTyEeaiB7ZTKIxrvA">
    <edge xmi:type="uml:ControlFlow" xmi:id="_RCGYoBQVEeaDOZ5Q8Z4TQQ" target="_zA45sBQSEeaDOZ5Q8Z4TQQ" source="_wAqp8BQSEeaDOZ5Q8Z4TQQ"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_I18FIBQ5EeaklsP-k46Huw" target="_ZSqqoBQVEeaDOZ5Q8Z4TQQ" source="_zA45sBQSEeaDOZ5Q8Z4TQQ"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_LXT1sBQ5EeaklsP-k46Huw" target="_1nQCsBQ4EeaklsP-k46Huw" source="_ZSqqoBQVEeaDOZ5Q8Z4TQQ"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_NVTMYBQ5EeaklsP-k46Huw" name="[exits]" target="_MNM0UBQXEeaDOZ5Q8Z4TQQ" source="_1nQCsBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_QPBykBQ5EeaklsP-k46Huw" name="[not exist]" target="_5Xu8oBQ2EeaklsP-k46Huw" source="_1nQCsBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_VRiDIBQ5EeaklsP-k46Huw" target="_3BqLMBQ4EeaklsP-k46Huw" source="_MNM0UBQXEeaDOZ5Q8Z4TQQ"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_d9GFEBQ5EeaklsP-k46Huw" name="[vaild]" target="_i_ZZwBQXEeaDOZ5Q8Z4TQQ" source="_3BqLMBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_fhLuABQ5EeaklsP-k46Huw" name="[invalid]" target="_SaALcBQwEeaklsP-k46Huw" source="_3BqLMBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_mwOHkBQ5EeaklsP-k46Huw" target="_dqiEgBQwEeaklsP-k46Huw" source="_SaALcBQwEeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_pJQbgBQ5EeaklsP-k46Huw" target="_WJNoIBQwEeaklsP-k46Huw" source="_dqiEgBQwEeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_rOZjEBQ5EeaklsP-k46Huw" target="_ZvUMIBQwEeaklsP-k46Huw" source="_dqiEgBQwEeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_uT3fMBQ5EeaklsP-k46Huw" target="_f0AScBQwEeaklsP-k46Huw" source="_WJNoIBQwEeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_4zFQEBQ5EeaklsP-k46Huw" target="_TEp80BTyEeaiB7ZTKIxrvA" source="_ZvUMIBQwEeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_56BQABQ5EeaklsP-k46Huw" target="_o_cHQBQwEeaklsP-k46Huw" source="_f0AScBQwEeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_6gXRQBQ5EeaklsP-k46Huw" target="_4RolUBQ4EeaklsP-k46Huw" source="_o_cHQBQwEeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_7bWS0BQ5EeaklsP-k46Huw" name="[pass]" target="_i_ZZwBQXEeaDOZ5Q8Z4TQQ" source="_4RolUBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_94gRwBQ5EeaklsP-k46Huw" name="[not pass]" target="_5Xu8oBQ2EeaklsP-k46Huw" source="_4RolUBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_BFnG8BQ6EeaklsP-k46Huw" target="_6z9_MBQ4EeaklsP-k46Huw" source="_i_ZZwBQXEeaDOZ5Q8Z4TQQ"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_CGxpABQ6EeaklsP-k46Huw" name="[class is not s]" target="_xf7hoBQwEeaklsP-k46Huw" source="_6z9_MBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_E48OoBQ6EeaklsP-k46Huw" name="[class is s]" target="_sLBScBQwEeaklsP-k46Huw" source="_6z9_MBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_G0NogBQ6EeaklsP-k46Huw" target="_8WDdABQ4EeaklsP-k46Huw" source="_sLBScBQwEeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_JNYfUBQ6EeaklsP-k46Huw" name="[more than 6 month]" target="_xf7hoBQwEeaklsP-k46Huw" source="_8WDdABQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_NPZjQBQ6EeaklsP-k46Huw" name="[lass than 6 month]" target="_5Xu8oBQ2EeaklsP-k46Huw" source="_8WDdABQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_QgrGwBQ6EeaklsP-k46Huw" target="_dvX5gBQ2EeaklsP-k46Huw" source="_xf7hoBQwEeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_RVufcBQ6EeaklsP-k46Huw" target="_-DWFQBQ4EeaklsP-k46Huw" source="_dvX5gBQ2EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_Uunt4BQ6EeaklsP-k46Huw" name="[not over]" target="_iEuwMBQ2EeaklsP-k46Huw" source="_-DWFQBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_W-wkgBQ6EeaklsP-k46Huw" name="[over]" target="_5Xu8oBQ2EeaklsP-k46Huw" source="_-DWFQBQ4EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_eOT7cBQ6EeaklsP-k46Huw" target="_B-4twBQ5EeaklsP-k46Huw" source="_iEuwMBQ2EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_ewPj0BQ6EeaklsP-k46Huw" name="[approved]" target="_oGJgEBQ2EeaklsP-k46Huw" source="_B-4twBQ5EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_izBcwBQ6EeaklsP-k46Huw" name="[not approved]" target="_5Xu8oBQ2EeaklsP-k46Huw" source="_B-4twBQ5EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_kYUmoBQ6EeaklsP-k46Huw" target="_ChtG8BQ5EeaklsP-k46Huw" source="_oGJgEBQ2EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_k3w7sBQ6EeaklsP-k46Huw" name="[success]" target="_xgsu8BQ2EeaklsP-k46Huw" source="_ChtG8BQ5EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_CddOMBQ7EeaklsP-k46Huw" name="[error]" target="_XuM60BTyEeaiB7ZTKIxrvA" source="_ChtG8BQ5EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_SckWQBQ7EeaklsP-k46Huw" target="_DaUJ8BQ5EeaklsP-k46Huw" source="_xgsu8BQ2EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_UFajYBQ7EeaklsP-k46Huw" name="[success]" target="_1LwEYBQ2EeaklsP-k46Huw" source="_DaUJ8BQ5EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_VrMlcBQ7EeaklsP-k46Huw" name="[error]" target="_3Huo4BQ2EeaklsP-k46Huw" source="_DaUJ8BQ5EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_YcBXQBQ7EeaklsP-k46Huw" target="_EUcP4BQ5EeaklsP-k46Huw" source="_3Huo4BQ2EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_ZEH5EBQ7EeaklsP-k46Huw" name="[verify]" target="_1LwEYBQ2EeaklsP-k46Huw" source="_EUcP4BQ5EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_bN0JcBQ7EeaklsP-k46Huw" name="[cancel]" target="_5Xu8oBQ2EeaklsP-k46Huw" source="_EUcP4BQ5EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_cyxVEBQ7EeaklsP-k46Huw" target="_8qxasBQ2EeaklsP-k46Huw" source="_1LwEYBQ2EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_eC9KkBQ7EeaklsP-k46Huw" target="_8qxasBQ2EeaklsP-k46Huw" source="_5Xu8oBQ2EeaklsP-k46Huw"/>
    <edge xmi:type="uml:ControlFlow" xmi:id="_Wlm7EBTyEeaiB7ZTKIxrvA" target="_f0AScBQwEeaklsP-k46Huw" source="_TEp80BTyEeaiB7ZTKIxrvA"/>
```


รูปที่ 9 ตัวอย่างเอกซ์เอ็มไอที่ได้จากแผนภาพกิจกรรม

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 1 ข้อมูลที่สกัดได้จากไฟล์เอกสารเอกซ์เอ็มไอ

XMI Tag Name	Example	XMI Tag Attribute	XMI Tag Value	คำอธิบาย
packagedElement	<packagedElement xmi:type="uml:Activity" xmi:id="_FLi7ABQNEeaDOZ5Q8Z4TQQ" name="Mileage Purchase Activity"... >	xmi:type	uml:Activity	คุณลักษณะที่บ่งบอกว่า packagedElement ชนิดนี้เป็น Activity Diagram
		name	Activity Diagram name	ชื่อของ Activity Diagram เช่น “Mileage Purchase Activity”
		xmi:id	Activity Diagram ID	ระบุ ID ของ Activity Diagram นั้นๆ
		node	Node ID	ID ของโหนดในแผนภาพกิจกรรม
edge	<edge> ... </edge>	-	-	Tag ภายใน packagedElement ที่เป็น Root Element ของ edge ทั้งหมด
edge	<edge xmi:type="uml:ControlFlow" ..>	xmi:type	uml:ControlFlow	คุณลักษณะที่บ่งบอกว่า edge เป็นชนิด Control Flow
edge	<edge xmi:type ..... xmi:id="_RCGYoBQVEeaDOZ5Q8Z4TQQ"...	xmi:id	value	เก็บข้อมูล edge id
edge	<edge xmi:type="uml:ControlFlow" xmi:id="_7bWS0BQ5EeaklsP-k46Huw" name="[pass]">	name	value	เก็บข้อมูลชื่อของ edge

ตารางที่ 2 ข้อมูลที่สกัดได้จากไฟล์เอกสารเอกซ์เอ็มไอ (ต่อ)

XMI Tag Name	Example	XMI Tag Attribute	XMI Tag Value	คำอธิบาย
edge	<edge xmi:type="uml:ControlFlow" ... target="_i_ZZwBQXEeaDOZ5Q8Z4TQQ"/>	target	value	เก็บแหล่งปลายทางของ edge
edge	<edge xmi:type="uml:ControlFlow" .... source="_4RolUBQ4EeaklsP-k46Huw" />	source	value	เก็บแหล่งต้นทางของ edge
node	<node> ... </node>	-	-	Tag ภายใน packagedElement ที่เป็น Root Element ของ node ทั้งหมด
node	<node xmi:type="uml:OpaqueAction" xmi:id="_zA45sBQSEeaDOZ5Q8Z4TQQ" name="Login" incoming="_RCGYoBQVEeaDOZ5Q8Z4TQQ" outgoing="_I18FIBQ5EeaklsP-k46Huw" />	xmi:type	InitialNode	คุณลักษณะที่บ่งบอกว่า node เป็นชนิด Initial Node หรือโหนดเริ่มต้น
			OpaqueAction	คุณลักษณะที่บ่งบอกว่า node เป็นชนิด Action หรือโหนดกิจกรรม
			ForkNode	คุณลักษณะที่บ่งบอกว่า node เป็นชนิด Fork Node หรือแสดงการทำงานพร้อมกัน
			JoinNode	คุณลักษณะที่บ่งบอกว่า node เป็นชนิด Join Node หรือแสดงการทำงานพร้อมกัน

ตารางที่ 3 ข้อมูลที่สกัดได้จากไฟล์เอกสารเอกซ์เอ็มไอ (ต่อ)

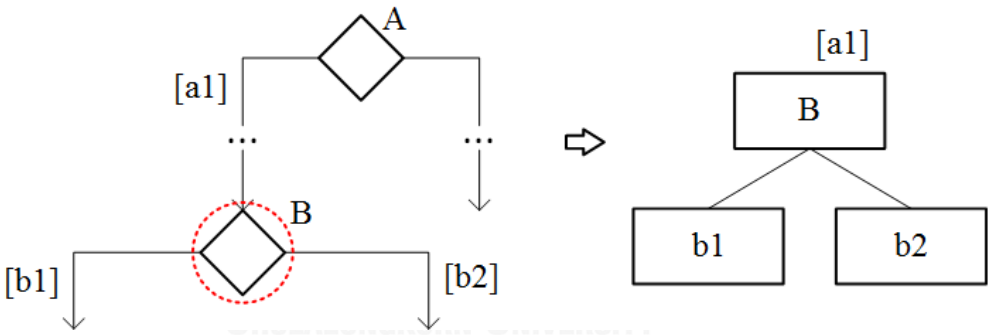
XMI Tag Name	Example	XMI Tag Attribute	XMI Tag Value	คำอธิบาย
node		xmi:type	DecisionNode	คุณลักษณะที่บ่งบอกว่า node เป็นชนิด Decision Node หรือโหนดแสดงจุดตัดสินใจ
			ActivityFinalNode	คุณลักษณะที่บ่งบอกว่า node เป็นชนิด Final Node หรือโหนดสิ้นสุด
node	<node xmi:type="uml:OpaqueAction" xmi:id="_zA45sBQSEeaDOZ5Q8Z4TQQ" >	xmi:id	value	เก็บข้อมูล node id
node	<node xmi:type="uml:OpaqueAction" ... name="Login" ..>	name	value	เก็บข้อมูลชื่อของ node
node	<node xmi:type="uml:OpaqueAction" ... incoming="_RCGYoBQVEeaDOZ5Q8Z4TQ	incoming	value	เก็บข้อมูลต้นทางขาเข้าของ node
node	<node xmi:type="uml:OpaqueAction" ... outgoing="_I18FIBQ5EeaklsP-k46Huw" />	outgoing	value	เก็บข้อมูลปลายทางขาออกของ node

3.1.2 สร้างแผนภาพต้นไม้จำแนกแบบมีเงื่อนไข

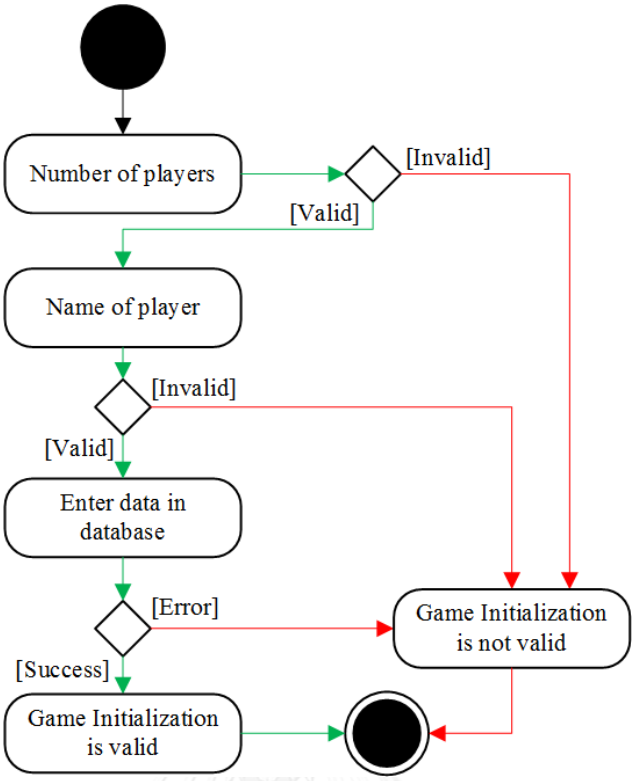
จากขั้นตอน 3.1.1 ที่ได้กล่าวมานั้นจะเห็นได้ว่า สามารถจำแนกจุดตัดสินใจได้จากแท็กเอกซ์เอ็มไอ (XMI tag) ประเภท xmi:type=”uml:DecisionNode” โดยที่เราจะนำจุดตัดสินใจนี้มาสร้างต้นไม้การจำแนกแบบมีเงื่อนไข

การสร้างต้นไม้จำแนกแบบมีเงื่อนไข วิเคราะห์เมทาเดตาของแผนภาพกิจกรรม โดยจะพิจารณาจากจุดตัดสินใจ (Decision Node) ที่มีเงื่อนไข (Condition) เพื่อกำหนดทางเลือกทำกิจกรรมถัดไป

1. สร้างโหนดพ่อแม่ (Parents Node) โดยการพิจารณาจากจุดตัดสินใจที่แสดงดังรูปที่ 10 คือ จุดบี (B) จากนั้นจึงสร้างโหนดตั้งต้นคือ จุดบี (B) ขึ้นมา
2. สร้างโหนดลูก (Child Node) โดยการพิจารณาจากเงื่อนไขหรือเส้นทางออกจากจุดตัดสินใจดังรูปที่ 10 จากจุดตัดสินใจคือจุดบี (B) เส้นทางที่เป็นเงื่อนไขออกมาจากจุดบี (B) ประกอบด้วย 2 เส้นทาง คือ จุดบี 1 (b1) และ จุดบี 2 (b2) จึงสร้างเป็นโหนดลูก

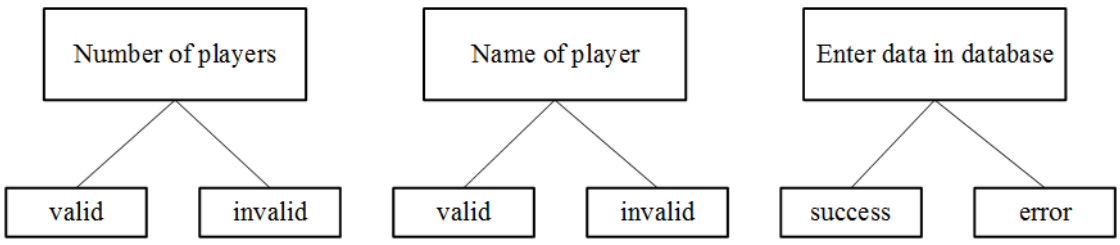


รูปที่ 10 โครงสร้างต้นไม้ที่สร้างจากจุดตัดสินใจ [14]



รูปที่ 11 แผนภาพกิจกรรมของแอปพลิเคชันการเริ่มต้นเกม

ขั้นตอนนี้จะทำการสร้างต้นไม้การจำแนกแบบมีเงื่อนไขจากการพิจารณาจุดตัดสินใจแต่ละจุด โดยวิธีการสร้างได้ถูกกล่าวไว้ข้างต้น จากตัวอย่างในรูปที่ 11 จะสามารถสร้างต้นไม้การจำแนกได้ดังรูปที่ 12



รูปที่ 12 ต้นไม้การจำแนกของแอปพลิเคชันการเริ่มต้นเกมโดยพิจารณาจากจุดตัดสินใจ

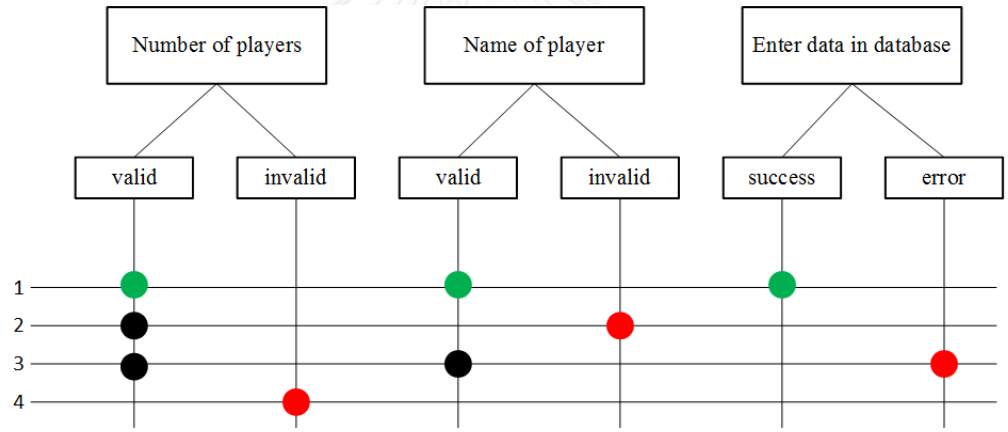
3.1.3 สร้างตารางกรณีทดสอบ

ขั้นตอนนี้จะรวบรวมข้อมูลจากทุกจุดตัดสินใจมาวางเรียงกันตามลำดับเพื่อสร้างเป็นตารางกรณีทดสอบ โดยลากเส้นจากโหนดลูกของต้นไม้ และพิจารณาเงื่อนไขของโหนดลูกจากต้นไม้การ

จำแนกว่าสามารถเลือกจับกับโหนดลูกต้นใดได้จากนั้นจึงทำจุดลงบนเส้นที่ลากจากโหนดลูก โดยเส้นแถวแนวนอนคือหมายเลขของกรณีทดสอบ จะลากผ่านเส้นของโหนดลูกทุกๆ เส้น เพื่อให้ทราบว่าจุดนั้นอยู่ในกรณีทดสอบหมายเลขอะไร

ในขั้นตอนนี้ต้นไม้การจำแนกควรจะถูกสร้างจากจุดตัดสินใจทุกๆจุด ในแผนภาพกิจกรรม ดังนั้น ขั้นตอนต่อไปคือการสร้างกรณีทดสอบซึ่งสามารถแบ่งได้เป็น 2 ขั้นตอน

1. ระบุกรณีทดสอบแต่ละกรณี โดยมีการระบุหมายเลขหรือชื่อในแต่ละแถวของตารางซึ่งจะพิจารณาจากจุดตัดสินใจที่ได้ถูกแปลงเป็นต้นไม้การจำแนก ในที่นี้จะเริ่มจากซ้ายไปขวา การพิจารณาเงื่อนไขจะถูกใช้เพื่อระบุว่าต้นไม้การจำแนกสามารถเชื่อมต่อกับกรณีทดสอบ โดยวิเคราะห์จากแผนภาพกิจกรรม จากตัวอย่างในรูปที่ 11 และ 12 จะสามารถสร้างตารางกรณีทดสอบได้ดังรูปที่ 13 จะเห็นได้ว่ากรณีทดสอบแรกนั้นมีการเชื่อมต่อแต่ละโหนด โดยโหนดแรกซึ่งเริ่มจากทางด้านซ้ายจะเชื่อมต่อกับโหนดลูกของจำนวนผู้เล่น (Number of players)



รูปที่ 13 ตารางกรณีทดสอบของแอปพลิเคชันการเริ่มต้นเกม

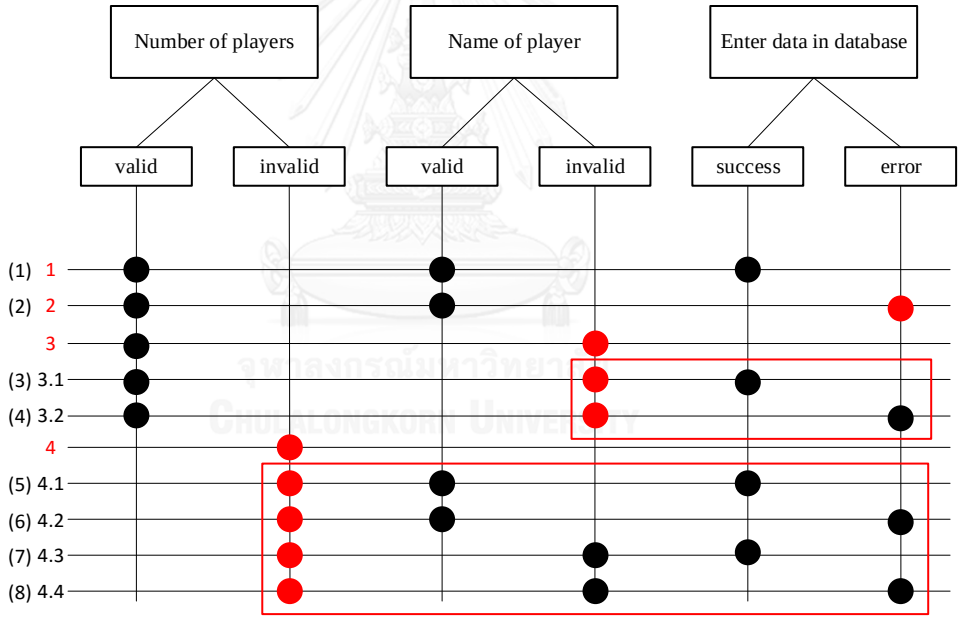
2. กรณีที่มีโหนดลูกในต้นไม้การจำแนกที่ไม่ถูกเลือกหลังจากทำการพิจารณาครบทุกต้นไม้การจำแนกแล้ว สามารถกำหนดให้เป็นโหนดเดียวในกรณีทดสอบนั้นได้ เช่น กรณีทดสอบที่ 4 จากรูปที่ 13

จากรูปที่ 13 สามารถสรุปกรณีทดสอบได้เป็น 4 กรณี โดยมี 1 กรณีที่โปรแกรมสามารถทำงานได้อย่างถูกต้อง และ 3 กรณีที่โปรแกรมไม่สามารถทำงานได้ โดยสามารถสรุปตารางกรณีทดสอบดังตารางที่ 4

ตารางที่ 4 สรุปกรณีทดสอบที่ได้จากแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข

Case	#players	Name of player	Enter data in database	Result: Game Initialization
1	Valid	Valid	Success	Valid
2	Valid	Valid	Error	Invalid
3	Valid	Invalid	–	Invalid
4	Invalid	–	–	Invalid

จากตารางจะเห็นได้ว่าจำนวนของกรณีทดสอบจะมีแค่ 4 กรณีทดสอบ จากกรณีทดสอบทั้งหมด 8 กรณี ซึ่งในงานวิจัยนี้ได้ลดกรณีที่ซ้ำซ้อนออก



รูปที่ 14 การลดกรณีทดสอบที่มีความซ้ำซ้อน

จากตัวอย่างรูปที่ 14 จะเห็นได้ว่าหากกรณีทดสอบที่ 3 ไม่ได้ลดความซ้ำซ้อนจะมีกรณีทดสอบเพิ่มขึ้นมาอีกสองกรณี คือ กรณีทดสอบที่ 3.1 และ 3.2 ซึ่งเป็นกรณีทดสอบที่มีความซ้ำซ้อนและไม่จำเป็นในการใช้เพื่อทดสอบ เนื่องจากเป็นกรณีที่ไม่สามารถเกิดขึ้นได้ ดังนั้นงานวิจัยและระบบที่พัฒนาในงานวิจัยนี้จึงได้ทำการลดกรณีทดสอบที่ซ้ำซ้อนออก ทั้งนี้เพื่อให้ได้กรณีทดสอบที่กระชับครอบคลุม ลดกรณีทดสอบที่ไม่จำเป็น

เช่นเดียวกับกรณีทดสอบที่ 4 จะมีกรณีทดสอบทั้งหมดที่สามารถเกิดขึ้นได้คือ 4.1 – 4.4 ซึ่งเป็นกรณีทดสอบที่ไม่ควรจะที่สร้างขึ้น เมื่อลดกรณีที่ซ้ำซ้อนออกจะทำให้เหลือกรณีที่ 4 เพียงกรณีเดียว เพื่อทำให้การทดสอบง่ายขึ้น

```
/**
 * Method Name : generateTestCaseTable
 * Parameter   : UmlNodes - list of UmlNodes from Input class
 *             : finalNode - final UmlNode
 * Description : recursive walk thru every possible paths.
 *               Also, create CCTM and connect to the matched condition
 * Output      : populated testcases
 */
private void generateTestCaseTable(ArrayList<UmlNode> nodes, UmlNode finalNode){
    //Declared Array
    ArrayList<AdjacentNode> sortedArray = new ArrayList<AdjacentNode>();
    for(AdjacentNode goal : finalNode.getSources()){
        UmlNode startNode = nodes.get(goal.getIndex());
        if(startNode.isValidPath()){
            sortedArray.add(0,goal);
        }else {sortedArray.add(goal);}
    }
    //Loop through nodes that connect to final node
    for (AdjacentNode goal : sortedArray){
        //Get start node
        UmlNode startNode = nodes.get(goal.getIndex());
        //Set valid flag
        temp.setValid(startNode.isValidPath());
        //Call recursive function to get test case
        getDraftTestCase_recursive(startNode);

    }
    removeDup(); //Remove duplicate test case
}
```

รูปที่ 15 ตัวอย่างโค้ดที่ใช้ในการสร้างกรณีทดสอบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตัวอย่างโค้ดดังรูป 15 ระบบที่ถูกพัฒนาในงานวิจัยนี้ทำการสร้างกรณีทดสอบโดยเริ่มจากการค้นหาโหนดสิ้นสุดวนไปจนถึงโหนดเริ่มต้น เพื่อทำการค้นหาเส้นทางที่เป็นไปได้เพื่อสร้างกรณีทดสอบ โดยจะมีการกำหนดเส้นทางที่ถูกต้องและทำการลดกรณีทดสอบที่ซ้ำซ้อนออกไป

จากตัวอย่างโปรแกรมที่ได้ยกมาแล้วนั้นเมื่อสร้างตารางความจริงเพื่อกำหนดกรณีทดสอบ จะได้ทั้งหมด 8 กรณีเนื่องจากมี 3 จุดตัดสินใจ อย่างไรก็ตาม ในงานวิจัยนี้ เราใช้วางข้อมูลของโหนดต่างๆ ในรูปของ โคตรสร้างต้นไม้ (tree structure) และใช้การวนซ้ำแบบรีเคอร์ซีฟ (recursive) เพื่อสร้างกรณีทดสอบ ซึ่งจะส่งผลให้กรณีทดสอบลดความซ้ำซ้อนและไม่จำเป็น

จากตัวอย่างที่ได้ยกมานั้น เราได้ทำการวาง ข้อมูลของโหนด name of จะถูกเก็บในรูปแบบของ tree structure โดยที่มีสองกิ่ง คือ enter data in database และ game initialize is invalid ซึ่งเป็น โหนดธรรมดาและโหนดสิ้นสุดตามลำดับ เมื่อโปรแกรมทำการวนซ้ำแบบ recursive จะพบว่า

เส้นทางที่พบคือ กรณีทดสอบที่ 3 ในขั้นต้น ต่อมาทำประมวลผลโหนด enter data in database เมื่อพบว่า โหนดนี้มีกิ่งเป็น final node ดังนั้น กรณีทดสอบ 3.1 และ 3.2 จะไม่เกิดขึ้น เช่นเดียวกับกรณีทดสอบที่ 4.1 – 4.4

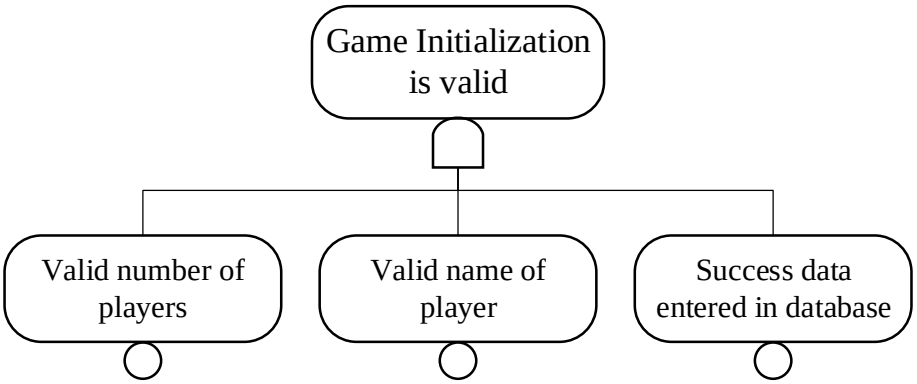
3.1.4 แปลงกรณีทดสอบเป็นแผนภาพต้นไม้ความสำเร็จและแผนภาพต้นไม้ความผิดพลาด

ตารางกรณีทดสอบที่ถูกสร้างขึ้นจะสามารถนำมาใช้วิเคราะห์และสร้างแผนภาพ 2 ชนิด คือ ต้นไม้ความสำเร็จและต้นไม้ความผิดพลาดซึ่งทั้งสองแผนภาพนี้จะถูกแสดงออกมาในรูปของลอจิกเกต (Logic gate) โดยจุดประสงค์ของสองแผนภาพนี้จะแตกต่างกัน กล่าวคือ แผนภาพต้นไม้ทั้งสองจะใช้สำหรับทดสอบความถูกต้องของการทำงานและความน่าเชื่อถือของซอฟต์แวร์ตามลำดับ

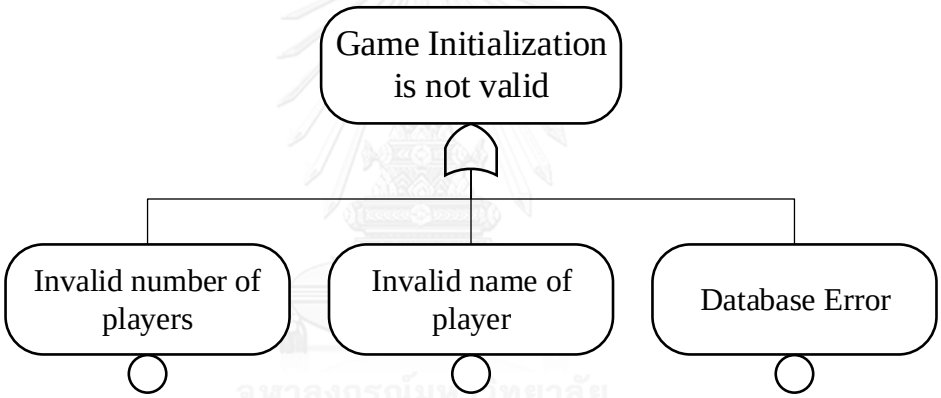
การสร้างต้นไม้ความสำเร็จและต้นไม้ความผิดพลาด มีขั้นตอนดังนี้

1. หาเส้นทางที่ถูกต้อง (Valid path) จากแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไขและกรณีทดสอบ ซึ่งในตัวอย่างที่ยกมานี้จะเป็นกรณีทดสอบที่ 1 ของรูปที่ 13
2. สร้างเป้าหมายของแผนภาพต้นไม้ความสำเร็จซึ่งอยู่ด้านบนสุดของแผนภาพด้วยเป้าหมายจากขั้นตอนที่ 1
3. สร้างแผนภาพต้นไม้ความสำเร็จโดยใช้แอนด์เกต (AND gate) ที่เชื่อมต่อกับโหนดในกรณีทดสอบที่ 1 ซึ่งในกรณีนี้ข้อมูลนำเข้าประกอบไปด้วย 3 อย่างด้วยกัน คือ จำนวนผู้เล่นที่ถูกต้อง ชื่อผู้เล่นที่ถูกต้อง และการบันทึกข้อมูลลงฐานข้อมูลที่สำเร็จ
4. ในกรณีที่เส้นทางที่ถูกต้องมีมากกว่าหนึ่งหรือกรณีทดสอบที่ไปสู่เป้าหมายที่สำเร็จมีมากกว่าหนึ่ง จะต้องนำออเกต (OR gate) มาใช้เพิ่มเติม
5. สร้างแผนภาพต้นไม้ความผิดพลาดโดยใช้ออเกต (OR gate) ซึ่งในกรณีนี้คือกรณีทดสอบที่ 2, 3 และ 4

เมื่อนำห้าขั้นตอนดังกล่าวมาใช้กับตัวอย่างที่ได้ยกมานั้นคือโปรแกรมการเริ่มต้นเกม สามารถสร้างแผนภาพต้นไม้ความสำเร็จและแผนภาพต้นไม้ความผิดพลาดได้ดังรูปที่ 16 และ 17



รูปที่ 16 ต้นไม้ความสำเร็จของแอปพลิเคชันการเริ่มต้นของเกม [15]



จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 17 ต้นไม้ความผิดพลาดของแอปพลิเคชันการเริ่มต้นของเกม [15]

บทที่ 4
การออกแบบและพัฒนาระบบ

รายละเอียดในบทนี้จะอธิบายหลักการออกแบบและพัฒนาระบบต้นแบบเพื่อสนับสนุนการสร้างแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข แผนภาพต้นไม้ความสำเร็จ และแผนภาพต้นไม้ความผิดพลาดเพื่อสร้างกรณีทดสอบต่อไป โดยเนื้อหาในบทนี้จะประกอบด้วย 1) ข้อกำหนดการทำงานเบื้องต้นของระบบ 2) ความต้องเชิงหน้าที่ 3) การออกแบบสถาปัตยกรรมของระบบ และ 4) การพัฒนาระบบ

4.1 ข้อกำหนดการทำงานเบื้องต้นของระบบ

- ผู้ใช้งาน (User)

  ผู้ที่มีความรู้เกี่ยวกับภาพรวมของระบบ (Software Designer) นักวิเคราะห์และออกแบบระบบ (System Analyst) หรือผู้ทดสอบระบบ (Tester)

- ข้อมูลนำเข้า (Input)

  แฟ้มเอกสารนำเข้ารูปแบบเอกซ์เอ็มไอที่ได้มาจากการแปลงแผนภาพกิจกรรมด้วยเครื่องมือวาดแผนภาพยูเอ็มแอล ปาปิรุส มาร์ส เวอร์ชัน 1.1.4

- ข้อมูลนำออก (Output)
  1. แผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข
  2. แผนภาพต้นไม้ความสำเร็จ
  3. แผนภาพต้นไม้ความผิดพลาด

- ข้อจำกัดของระบบ (Constraint)
  1. ข้อมูลนำเข้าจะต้องอยู่ในรูปแบบไฟล์เอกซ์เอ็มไอที่มาจากโปรแกรมเสริมอีคลิปส์ ปาปิรุส มาร์ส เวอร์ชัน1.1.4 เท่านั้น
  2. ข้อมูลนำเข้าสนับสนุนภาษาอังกฤษเท่านั้น
  3. สนับสนุนเฉพาะรูปแบบของจุดตัดสินใจของแผนภาพกิจกรรม
  4. ความสมบูรณ์ถูกต้องของข้อมูลนำออกขึ้นอยู่กับการออกแบบของเมทาเดตาข้อมูลนำเข้าที่เหมาะสม และการเลือกเป้าหมายเริ่มต้นของโปรแกรมที่ถูกต้อง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.2 ความต้องการเชิงหน้าที่ (Functional Requirements)

ระบบสร้างแผนภาพต้นไม้ความผิดพลาดที่ใช้ในงานวิจัยฉบับนี้มีความต้องการของระบบในเชิงหน้าที่ จะถูกอธิบายในตารางที่ 5

ตารางที่ 5 ความต้องการเชิงหน้าที่ของระบบสร้างแผนภาพต้นไม้ความผิดพลาด

รหัสอ้างอิง	ชื่อ	คำอธิบายความต้องการเชิงหน้าที่
FREQ-01	นำเข้าแผนภาพกิจกรรมในรูปแบบไฟล์เอกซ์เอ็มไอ	ระบบสามารถนำเข้าแผนภาพกิจกรรมในรูปแบบไฟล์เอกสารเอกซ์เอ็มไอได้
FREQ-02	สกัดเมทาเดตาแผนภาพกิจกรรมได้	ระบบสามารถสกัดคำอธิบายแผนภาพกิจกรรมได้
FREQ-03	ประมวลผลเพื่อแปลงจากเอกซ์เอ็มไอเป็นยูเอ็มแอลโหนดได้	ระบบสามารถประมวลผลเพื่อแปลงจากเอกซ์เอ็มไอเป็นยูเอ็มแอลโหนดได้
FREQ-04	ประมวลหาความสัมพันธ์ของโหนด	ระบบสามารถประมวลหาความสัมพันธ์ของโหนดต่างๆ ได้
FREQ-05	ประมวลผลโหนดเพื่อทำดรรชนีในการค้นหา	ระบบสามารถประมวลผลโหนดเพื่อทำดรรชนีในการค้นหาได้
FREQ-06	ประมวลผลเป้าหมายที่เป็นไปได้และสามารถให้ผู้ใช้งานเลือกเป้าหมายที่ถูกต้องได้	ระบบสามารถให้ผู้ใช้งานเลือกเป้าหมายที่ถูกต้องได้
FREQ-07	ประมวลผลต้นไม้การจำแนกแบบมีเงื่อนไขได้	ระบบสามารถประมวลผลต้นไม้การจำแนกแบบมีเงื่อนไขได้
FREQ-08	ประมวลผลต้นไม้ความสำเร็จได้	ระบบสามารถประมวลผลต้นไม้ความสำเร็จได้
FREQ-09	ประมวลผลต้นไม้ความผิดพลาดได้	ระบบสามารถประมวลผลต้นไม้ความผิดพลาดได้
FREQ-10	แสดงแผนภาพต้นไม้จำแนกแบบมีเงื่อนไขได้	ระบบสามารถแสดงแผนภาพต้นไม้จำแนกแบบมีเงื่อนไขได้
FREQ-11	แสดงแผนภาพต้นไม้ความสำเร็จได้	ระบบสามารถแสดงแผนภาพต้นไม้

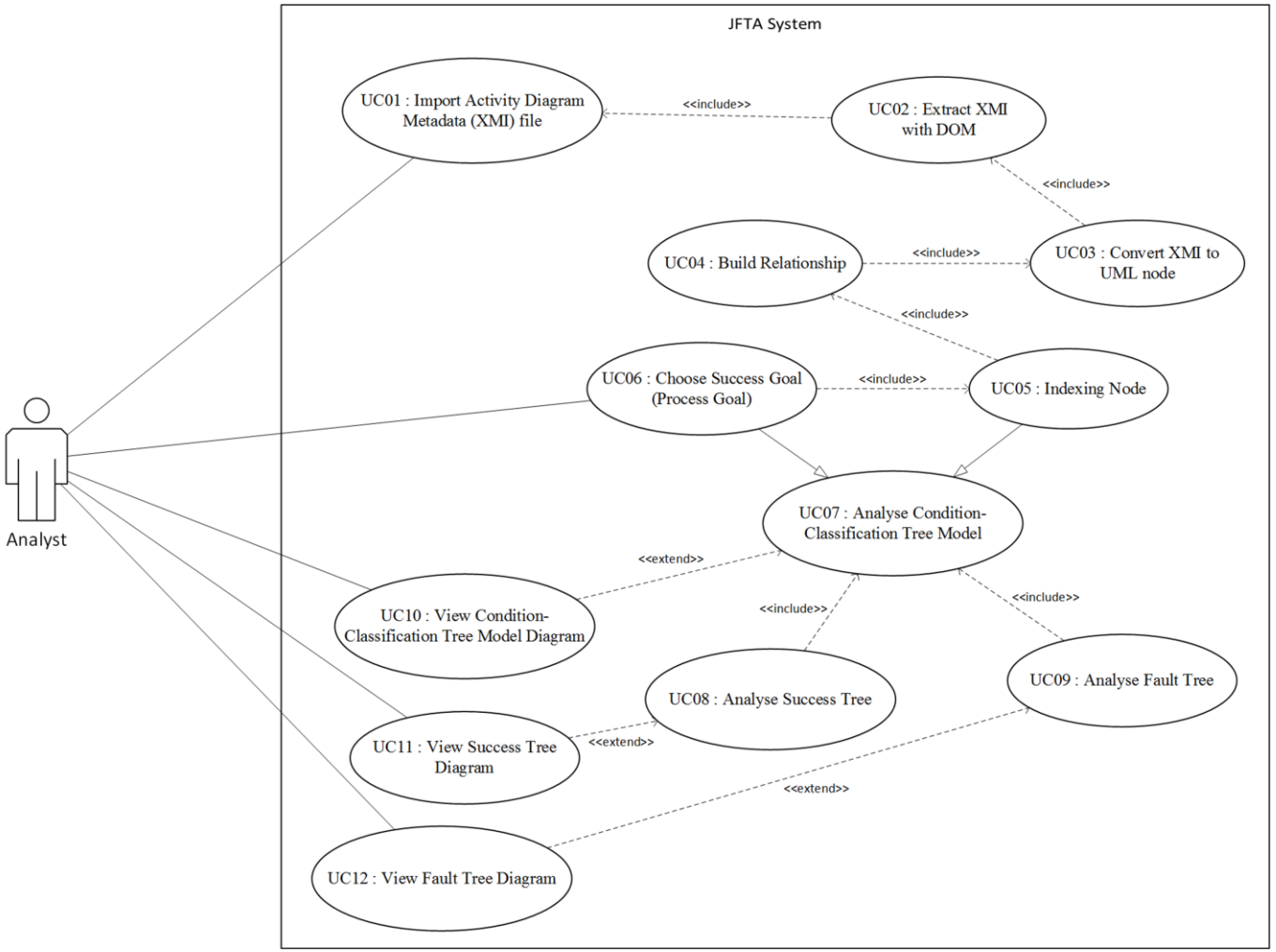
รหัสอ้างอิง	ชื่อ	คำอธิบายความต้องการเชิงหน้าที่
		ความสำเร็จได้
FREQ-12	แสดงแผนภาพต้นไม้ความผิดพลาดได้	ระบบสามารถแสดงแผนภาพต้นไม้ความผิดพลาดได้

จากความต้องการเชิงหน้าที่ในข้างต้นสามารถออกแบบระบบและแสดงเป็นแผนที่แสดงการทำงานของผู้ใช้ระบบ แผนภาพยูสเคส (Use Case) และคำอธิบายแผนภาพยูสเคส (Use Case Description)

4.2.1 แผนภาพยูสเคส (Use Case Diagram)

ระบบสร้างแผนภาพต้นไม้ความผิดพลาดสามารถแสดงการทำงานของผู้ใช้ระบบและระบบย่อยโดยใช้แผนภาพยูสเคสในการแสดงดังรูปที่ 18

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 18 แผนภาพยูสเคสของระบบสร้างแผนภาพต้นไม้ความผิดพลาด

4.2.2 คำอธิบายแผนภาพยูสเคส (Use Case Descriptions)

คำอธิบายแผนภาพยูสเคสจะอธิบายฟังก์ชันการทำงานในระบบที่ปรากฏอยู่ในแผนภาพยูสเคส ซึ่งแสดงการทำงานของผู้ใช้งานระบบและระบบย่อย และระบบภายในมีการทำงานอะไรบ้าง โดยคำอธิบายยูสเคสของระบบสร้างแผนภาพต้นไม้ความผิดพลาด จะแสดงดังตารางที่ 6 ถึง ตารางที่ 17 ดังต่อไปนี้

ตารางที่ 6 คำอธิบายยูสเคส Import Activity Diagram Metadata (XMI) file

หมายเลขยูสเคส:	UC01	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้ที่เกี่ยวข้อง:	ผู้ออกแบบระบบ ผู้วิเคราะห์ระบบ และผู้ทดสอบระบบ		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการนำเข้าแผนภาพกิจกรรมในรูปแบบไฟล์เอกสารเอกซ์เอ็มไอ		
เงื่อนไขขั้นต้น:	แผนภาพกิจกรรมที่นำเข้าต้องทำการแปลงแผนภาพให้อยู่ในรูปแบบไฟล์เอกสารเอกซ์เอ็มไอ		
การทำงานโดยปกติ:	1. ผู้ใช้งานคลิกปุ่ม ‘Choose File’ 2. เลือกไฟล์เอกสารเอกซ์เอ็มไอที่จะนำเข้า 3. ระบบตรวจสอบไฟล์ 4. ระบบประมวลผลไฟล์		
ทางเลือกเพิ่มเติมในการทำงาน:	3.1 นำเข้าไฟล์ไม่ถูกต้อง 3.2 ระบบแจ้งเตือนการนำเข้าไฟล์ไม่ถูกต้อง 3.3 เลือกกลับไปทำงานในขั้นตอนที่ 2 หรือออกจากระบบ		

ตารางที่ 7 คำอธิบายยูสเคส Extract XMI with DOM

หมายเลขยูสเคส:	UC02	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้ที่เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการสกัดเมทาเดตาเป็นอ๊อบเจกต์จากแผนภาพกิจกรรม		
เงื่อนไขขั้นต้น:	นำเข้าแผนภาพกิจกรรมในรูปแบบไฟล์เอกสารเอกซ์เอ็มไอ		
การทำงานโดยปกติ:	1. ระบบอ่านไฟล์เอกสารเอกซ์เอ็มไอของแผนภาพกิจกรรมแล้วสกัดเมทาเดตา และวิเคราะห์แท็กระบุ (Tag) ที่ต้องการ		

	2. ระบบนำอ๊อบเจกต์ (object) ที่ได้จากการประมวลผล ไปใช้ในการทำงานในขั้นตอนถัดไป
ทางเลือกเพิ่มเติมในการทำงาน:	-

ตารางที่ 8 คำอธิบายยูสเคส Convert XMI to UML node

หมายเลขยูสเคส:	UC03	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้ที่เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการแปลงอ๊อบเจกต์ของเอกซ์เอ็มไอเป็นยูเอ็มแอลโหนด		
เงื่อนไขขั้นต้น:	ระบบต้องสกัดเมทาเดตาและอ๊อบเจกต์จากแผนภาพกิจกรรมเรียบร้อยแล้ว		
การทำงานโดยปกติ:	1. ระบบทำการประมวลผลอ๊อบเจกต์ที่ได้ให้เป็นยูเอ็มแอลโหนดที่มีลักษณะเสมือนส่วนประกอบในแผนภาพกิจกรรม 2. ระบบนำยูเอ็มแอลโหนดที่ได้จากการประมวลไปใช้ในการทำงานในขั้นตอนถัดไป		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

ตารางที่ 9 คำอธิบายยูสเคส Build Node Relationship

หมายเลขยูสเคส:	UC04	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้ที่เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการหาความสัมพันธ์ของโหนด		
เงื่อนไขขั้นต้น:	ระบบต้องทำการประมวลผลเอกซ์เอ็มไอเป็นยูเอ็มแอลโหนดเรียบร้อยแล้ว		
การทำงานโดยปกติ:	1. ระบบทำการประมวลผลวิเคราะห์โหนดทั้งหมดในระบบเพื่อหาและจับคู่ความสัมพันธ์ 2. ระบบทำระบุความสัมพันธ์ของแต่ละโหนดโดยเพิ่มข้อมูลลงในยูเอ็มแอลโหนด 3. ระบบนำยูเอ็มแอลโหนดแบบมีความสัมพันธ์ที่ได้จากการประมวลผล		

	วิเคราะห์ไปใช้ในการทำงานในขั้นตอนถัดไป
ทางเลือกเพิ่มเติมในการทำงาน:	-

ตารางที่ 10 คำอธิบายยูสเคส Indexing Node

หมายเลขยูสเคส:	UC05	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้ที่เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการประมวลผลยูเอ็มแอลโหนดเพื่อทำดรรชนีค้นหา		
เงื่อนไขขั้นต้น:	ระบบต้องทำการประมวลผลหาความสัมพันธ์ของโหนดเรียบร้อยแล้ว		
การทำงานโดยปกติ:	1. ระบบสามารถทำดรรชนีเพื่อเข้าถึงโหนดได้ 2. ระบบสามารถเรียงลำดับโหนดได้ 3. ระบบสามารถแบ่งแยกโหนดเริ่มต้น (Initial Node) และโหนดสิ้นสุด (Final Node) ได้		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

ตารางที่ 11 คำอธิบายยูสเคส Choose Success Goal (Process Goal)

หมายเลขยูสเคส:	UC06	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้ที่เกี่ยวข้อง:	ผู้ออกแบบระบบ ผู้วิเคราะห์ระบบ และผู้ทดสอบระบบ		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการประมวลผลเป้าหมายที่เป็นไปได้และสามารถให้ผู้ใช้งานเลือกเป้าหมายที่ถูกต้องตามต้องการ		
เงื่อนไขขั้นต้น:	ระบบต้องทำการประมวลผลยูเอ็มแอลโหนดเพื่อทำดรรชนีค้นหาเรียบร้อยแล้ว		
การทำงานโดยปกติ:	1. ผู้ใช้งานทำการเลือกเป้าหมายที่ถูกต้อง 2. ผู้ใช้งานคลิกที่ปุ่ม ‘Choose Goal’		
ทางเลือกเพิ่มเติมในการทำงาน:	2.1 ระบบจะแสดงผลไม่ถูกต้องตามความต้องการหากทำการเลือกเป้าหมายผิด		

ตารางที่ 12 คำอธิบายยูสเคส Analyze Condition-Classification Tree Model

หมายเลขยูสเคส:	UC07	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้ที่เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการประมวลผลต้นไม้การจำแนกแบบมีเงื่อนไข		
เงื่อนไขขั้นต้น:	ระบบต้องทำการประมวลผลยูเอ็มแอลโหนดเพื่อทำดรรชนีค้นหาเรียบร้อยแล้วและผู้ใช้งานทำการเลือกเป้าหมายที่ถูกต้อง		
การทำงานโดยปกติ:	1. ระบบสามารถประมวลผลต้นไม้การจำแนกแบบมีเงื่อนไขได้ 2. ระบบสามารถสร้างอ๊อบเจกต์ในรูปแบบของแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไขได้		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

ตารางที่ 13 คำอธิบายยูสเคส Analyze Success Tree

หมายเลขยูสเคส:	UC08	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้ที่เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการประมวลผลต้นไม้ความสำเร็จ		
เงื่อนไขขั้นต้น:	ระบบต้องทำการประมวลผลต้นไม้การจำแนกแบบมีเงื่อนไขเรียบร้อยแล้ว		
การทำงานโดยปกติ:	1. ระบบสามารถประมวลผลต้นไม้ความสำเร็จได้ 2. ระบบสามารถสร้างอ๊อบเจกต์ในรูปแบบของแผนภาพต้นไม้ความสำเร็จได้		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 14 คำอธิบายยูสเคส Analyze Fault Tree

หมายเลขยูสเคส:	UC09	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้ที่เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการประมวลผลต้นไม้ความผิดพลาด		
เงื่อนไขขั้นต้น:	ระบบต้องทำการประมวลผลต้นไม้การจำแนกแบบมีเงื่อนไขเรียบร้อยแล้ว		
การทำงานโดยปกติ:	1. ระบบสามารถประมวลผลต้นไม้ความผิดพลาดได้ 2. ระบบสามารถสร้างอ๊อบเจกต์ในรูปแบบของแผนภาพต้นไม้ความผิดพลาดได้		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

ตารางที่ 15 คำอธิบายยูสเคส View Condition-Classification Tree Model Diagram

หมายเลขยูสเคส:	UC10	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (Moderate)		
ผู้ที่เกี่ยวข้อง:	ผู้ออกแบบระบบ ผู้วิเคราะห์ระบบ และผู้ทดสอบระบบ		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการแสดงผลของแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข		
เงื่อนไขขั้นต้น:	ระบบต้องทำการประมวลผลต้นไม้การจำแนกแบบมีเงื่อนไขเรียบร้อยแล้ว		
การทำงานโดยปกติ:	1. ผู้ใช้งานคลิกปุ่ม ‘View CCTM’ 2. ระบบสามารถประมวลผลอ๊อบเจกต์ในรูปแบบแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไขได้ 3. ระบบสามารถแสดงแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไขได้		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

ตารางที่ 16 คำอธิบายยูสเคส View Success Tree Diagram

หมายเลขยูสเคส:	UC11	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (Moderate)		
ผู้ที่เกี่ยวข้อง:	ผู้ออกแบบระบบ ผู้วิเคราะห์ระบบ และผู้ทดสอบระบบ		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการแสดงผลของแผนภาพต้นไม้ความสำเร็จ		
เงื่อนไขขั้นต้น:	ระบบต้องทำการประมวลผลต้นไม้ความสำเร็จเรียบร้อยแล้ว		
การทำงานโดยปกติ:	1. ผู้ใช้งานคลิปปุ่ม ‘View STD’ 2. ระบบสามารถประมวลผลอ๊อบเจกต์ในรูปแบบแผนภาพต้นไม้ความสำเร็จได้ 3. ระบบสามารถแสดงแผนภาพต้นไม้ความสำเร็จได้		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

ตารางที่ 17 คำอธิบายยูสเคส View Fault Tree Diagram

หมายเลขยูสเคส:	UC12	**เวอร์ชัน:**	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (Moderate)		
ผู้ที่เกี่ยวข้อง:	ผู้ออกแบบระบบ ผู้วิเคราะห์ระบบ และผู้ทดสอบระบบ		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการแสดงผลของแผนภาพต้นไม้ความผิดพลาด		
เงื่อนไขขั้นต้น:	ระบบต้องทำการประมวลผลต้นไม้ความผิดพลาดเรียบร้อยแล้ว		
การทำงานโดยปกติ:	1. ผู้ใช้งานคลิปปุ่ม ‘View FTD’ 2. ระบบสามารถประมวลผลอ๊อบเจกต์ในรูปแบบแผนภาพต้นไม้ความผิดพลาดได้ 3. ระบบสามารถแสดงแผนภาพต้นไม้ความผิดพลาดได้		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

4.3 การออกแบบสถาปัตยกรรมของระบบ

สถาปัตยกรรม เอ็มวีซี (Model-View-Controller : MCV) [16, 17] ได้ถูกนำมาใช้ในงานวิจัยนี้ โดยทั่วไปแล้วสถาปัตยกรรม เอ็มวีซีนิยมนำมาสร้างเว็บแอปพลิเคชัน โดยที่โครงสร้างของสถาปัตยกรรมนี้ประกอบไปด้วย 3 ส่วนประกอบเชิงตรรกะคือ 1) ส่วนชั้นธุรกิจ ซึ่งประกอบไปด้วยโมเดลเชิงตรรกะ 2) ส่วนการแสดงผล ซึ่งประกอบไปด้วย วิวเชิงตรรกะ 3) ส่วนควบคุมข้อมูลนำเข้าหรือส่วนควบคุมเชิงตรรกะ การแบ่งส่วนประกอบเป็น 3 ส่วนดังที่ได้กล่าวมานั้น มีส่วนช่วยให้นักพัฒนาสามารถทำงานได้อย่างมีประสิทธิภาพเนื่องจากมีการแบ่งสัดส่วนที่ชัดเจน

โมเดล หรือ เอ็ม ในเอ็มวีซี แสดงถึง แกนหลักของแอพพลิเคชั่นที่รองรับข้อมูลเชิงตรรกะ ตัวอย่างเช่น ข้อมูลการซื้อ การคำนวณวันเกิดของสมาชิก หรือ ผลรวมของภาษี เป็นต้น

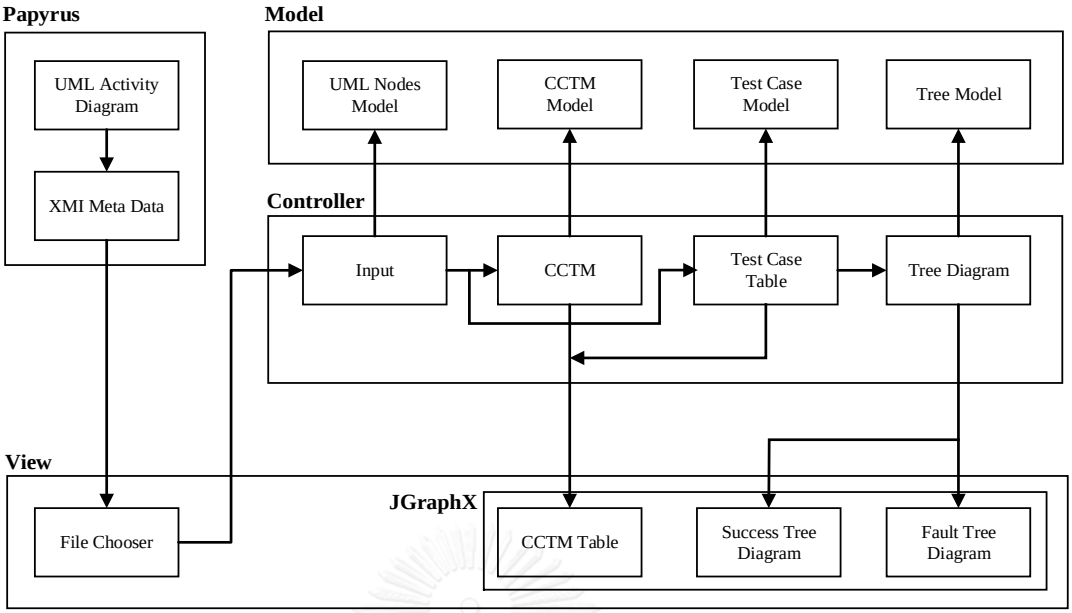
วิว หรือ วี ในเอ็มวีซี เป็นส่วนจัดการเรื่องการแสดงข้อมูล โดยทั่วไปแล้ว วิว จะถูกสร้างจากข้อมูลในโมเดล ซึ่งข้อมูลในหนึ่งโมเดลสดงออกมาได้หลายวิวเนื่องจากวัตถุประสงค์ในการแสดงผล วิวมีหน้าที่เป็นส่วนติดต่อกับผู้ใช้งานโดยการแสดงผล

คอนโทรลเลอร์ หรือ ซี ในเอ็มวีซี รับหน้าที่ประมวลข้อมูลนำเข้าจากผู้ใช้งานและใช้งานอัลกอริทึมต่าง ๆ ด้วยวัตุประสงค์ที่ต่างกันไป โดยแต่ละข้อมูลในโมเดลมีหน้าที่สนับสนุนอัลกอริทึมและคอนโทรลเลอร์สามารถส่งผลตอบสนองให้กับวิวเพื่อแสดงผลให้กับผู้ใช้งานได้อีกด้วย

การพัฒนาด้วย สถาปัตยกรรม เอ็มวีซี โดยทั่วไปจะถูกนำมาใช้สำหรับ อินเตอร์เน็ตแอพพลิเคชั่น วิวจะแสดงถึงการแสดงผลในเว็บไซต์หรือ โปรแกรมภาษาเอชทีเอ็มแอล ในขณะเดียวกันคอนโทรลเลอร์ยังรับหรือส่งจากนั้นประมวลผลด้วยข้อมูลในโมเดลเพื่อแสดงผลต่อไป โดยที่โมเดลประกอบไปด้วยกฎของธุรกิจสำหรับการประมวลผลและการแสดงผล

การสร้างกรณีทดสอบด้วยแผนภาพต้นไม้ความผิดพลาดในงานวิจัยนี้ มีข้อสมมติฐานและข้อจำกัดในการพัฒนาระบบดังนี้

- โปรแกรม ปาปิรุส มาร์ส จะถูกใช้ เพื่อแปลงแผนภาพกิจกรรมให้เป็นเมทาเดตาเอกซ์เอ็มไอ
- เจกราฟเอกซ์ (JGraphX) ได้ถูกนำมาใช้เป็นไลบรารี่สำหรับวิวเพื่อการสร้างแผนภาพ
- ข้อมูลนำเข้าของแผนภาพกิจกรรมจำเป็นต้องมีเป้าหมายที่สำเร็จหนึ่งเดียวและต้องได้รับการระบุมาจากผู้ใช้งาน
- แผนภาพต้นไม้ความผิดพลาดจะถูกสร้างจากจุดตัดสินใจในแผนภาพกิจกรรม



รูปที่ 19 การออกแบบสถาปัตยกรรมการทำงานของระบบ

4.3.1 การแปลงข้อมูลนำเข้า

การเริ่มต้นของงานวิจัยชิ้นนี้ จะเริ่มจากการแปลงแผนภาพกิจกรรมกลายเป็นเอ็กซ์เอ็มไอเมทาเดตา ซึ่งในที่นี่จะใช้โปรแกรม ปาปิรุส ซึ่งเอ็กซ์เอ็มไอจะประกอบไปด้วยข้อมูลของแผนภาพกิจกรรมในรูปแบบของเอ็กซ์เอ็มแอล ซึ่งแต่ละ เอ็กซ์เอ็มแอลประกอบไปด้วยคุณลักษณะ ของชื่อ รหัสเป้าหมาย และต้นกำเนิด โดยเอ็กซ์เอ็มแอลจะมีสองประเภทด้วยกันคือ แบบโหนด และ เส้นการควบคุม

4.3.2 คอนโทรลเลอร์ - โมเดล

ลำดับต่อมาเมื่อผู้ใช้งานมี เอ็กซ์เอ็มไอเมทาเดตาแล้ว โปรแกรมจะมีหน้าที่แปลงข้อมูลที่ได้มานั้นไปสู่ โมเดลของแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข ดังที่แสดงในรูป x ในส่วนนี้จะถูกแบ่งเป้น 4 ส่วนด้วยกันคือ

1. ข้อมูลนำเข้า
2. ต้นไม้การจำแนกแบบมีเงื่อนไข
3. กรณีทดสอบ
4. แผนภาพต้นไม้

ซึ่งโปรแกรมจะทำการประมวลผลวิเคราะห์เป็นลำดับขั้นดังรูป โดยที่แต่ละส่วนจะแสดงถึงคอนโทรลเลอร์คลาสและโมเดลของแต่ละตัว

4.3.2.1 ข้อมูลนำเข้า

วัตถุประสงค์ของส่วนนี้คือการแปลงเอ็กซ์เอ็มไอไฟล์กลายเป็นอ๊อบเจกต์ ซึ่งมีโครงสร้างคล้ายคลึงกับ ยูเอ็มแอลโหนด โดยที่โครงสร้างของยูเอ็มแอลโหนดนั้นสามารถสนับสนุนโหลดได้หลายประเภท เช่น โหนดการตัดสินใจ โหนดเริ่มต้น โหนดกิจกรรม และโหนดสิ้นสุด ซึ่งแต่ละโหนดนั้นควรจะประกอบไปด้วยคำอธิบายที่แสดงถึง ชื่อ เป้าหมาย ต้นทาง และคุณลักษณะ โดยทั่วไปแล้วข้อมูลนำเข้าควรจะถูกเก็บอยู่ในรูปแบบของกลุ่มของตัวแปร

4.3.2.2 ต้นไม้การจำแนกแบบมีเงื่อนไข

ในส่วนนี้จะรับหน้าที่ในการสร้างกรณีทดสอบด้วยวิธีแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข ซึ่งจุดตัดสินใจในระบบ จะถูกแปลงเป็นต้นไม้การจำแนกแบบมีเงื่อนไข ซึ่งรากของแต่ละต้นไม้การจำแนกเกิดจากจุดตัดสินใจที่มีความสัมพันธ์ของโหนดตัดสินใจนั้นๆ และจุดตัดสินใจทั้งหมดควรจะถูกประมวลผล

4.3.2.3 กรณีทดสอบ

จากทฤษฎี การสร้างกรณีทดสอบของแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไขนั้นเป็นฐานสำคัญในการสร้างแผนภาพต้นไม้ความผิดพลาด ซึ่งในงานวิจัยนี้ส่วนนี้จะมี ลักษณะการทำงานหรือการคำนวณที่ให้ผลลัพธ์แบบย้อนกลับหรือทำซ้ำขั้นตอนเดิม โดยที่ผลลัพธ์จะถูกเก็บในรูปแบบของตัวแปรอ๊อบเจกต์

4.3.2.4 แผนภาพต้นไม้

ขั้นต่อมา ผลลัพธ์จากส่วนที่กล่าวมานั้น งานวิจัยนี้จะนำผลลัพธ์นี้ไปใช้การสร้างแผนภาพต้นไม้ความสำเร็จและแผนภาพต้นไม้ความผิดพลาด ซึ่งเป็นผลลัพธ์ของงานวิจัยฉบับนี้ โดยที่ทฤษฎีได้กล่าวในบทขั้นต้น

งานวิจัยของแอปพลิเคชันในงานวิจัยนี้ จะใช้อ๊อบเจกต์ที่มีโครงสร้างของแผนภาพต้นไม้

4.3.3 วิว

ทำหน้าที่ติดต่อและแสดงผลแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข แผนภาพต้นไม้ความสำเร็จ และ แผนภาพต้นไม้ความผิดพลาด ซึ่งในงานวิจัยนี้จะใช้โปรแกรม JGraphX เป็นตัวช่วยในการสร้างแผนภาพต้นไม้ต่างๆ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

โปรแกรม JGraphX เป็นโปรแกรมที่ไม่คิดค่าใช้จ่ายใด ๆ และถูกเลือกมาใช้ในงานวิจัยนี้เนื่องด้วยโปรแกรมนี้ มีความยืดหยุ่นของแผนภาพต้นไม้ การสร้างแผนภาพ รวมถึงการปรับปรุงและสามารถสร้างสัญลักษณ์ตามที่ต้องการ โดยฉพาะการสร้างออร์เกตและแอนด์เกต

4.4 การพัฒนาระบบ

4.4.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนาระบบ

สภาพแวดล้อมที่ใช้ในการพัฒนาซอฟต์แวร์ระบบมีสภาพแวดล้อมดังนี้

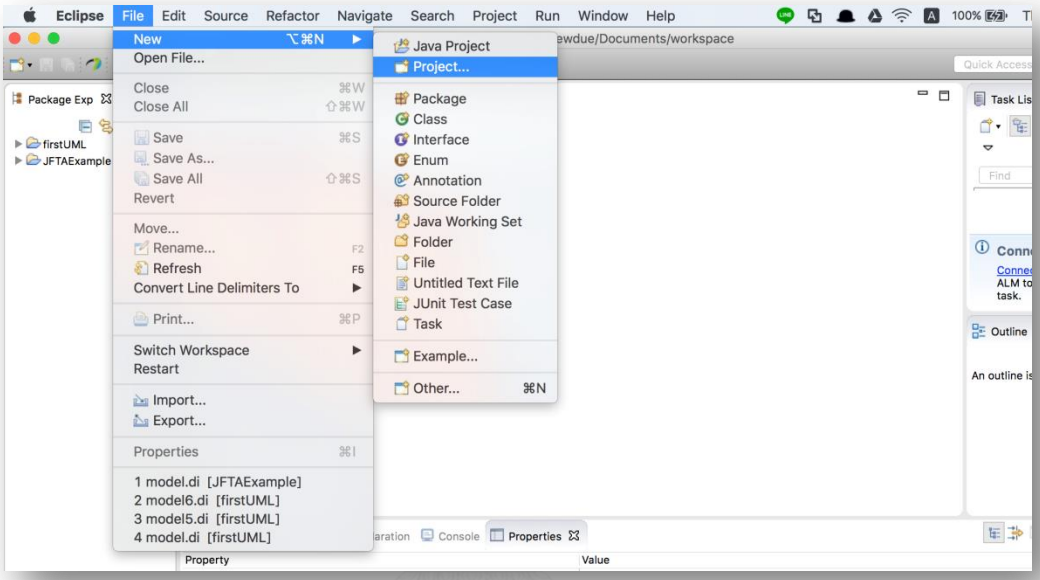
- ฮาร์ดแวร์
    - หน่วยประมวลผลกลาง (CPU) ดูโอคอร์ อินเทล คอร์ไอ 5 2.7 กิกะเฮิรตซ์ (2.7 GHz CPU dual-core Inter Core i5)
    - หน่วยความจำ (RAM) 8 กิกะไบต์ (RAM 8 GB)
    - จานบันทึกข้อมูล (Flash Storage) 256 กิกะไบต์ (256GB PCle-based flash storage)

- ซอฟต์แวร์

1. ระบบปฏิบัติการแมคโอเอสเอกซ์ เอล กับปิตัน 64 บิต เวอร์ชัน 10.11.3 (OSX El Capitan 64 bit version 10.11.3)
2. โปรแกรมอีคลิปส์ เวอร์ชัน 4.5.1 (Eclipse Modeling Tools)
3. โปรแกรมเสริมสำหรับวาดแผนภาพยูเอ็มแอล ปาปิรุส มาร์ส 1.1.4 (Papyrus Mars 1.1.4)
4. โปรแกรมเน็ตบีน เวอร์ชัน 8.0.2 (NetBeans IDE 8.0.2)
5. จาวา เจดีเค 8 (Java JDK 8)
6. เจกราฟเอกซ์ (JGraphX) เป็นจาวาสวิงสำหรับใช้ในการสร้างแผนภาพ

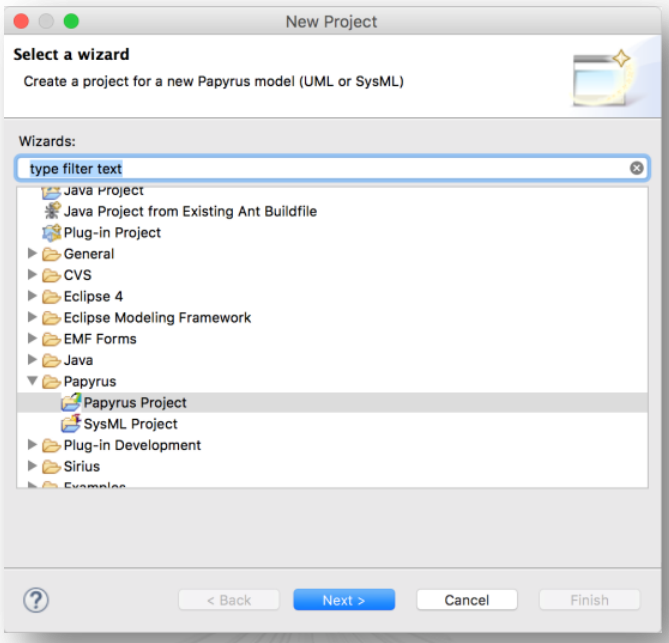
4.4.2 การพัฒนาวิธีการสร้างเมทาโมเดลของแผนภาพกิจกรรม

การพัฒนาระบบจะทำการสร้างแผนภาพกิจกรรมยูเอ็มแอลหรือเมทาโมเดลนั้น งานวิจัยนี้จะใช้อีคลิปส์โมเดลลิ่งทูนในการสร้าง ซึ่งก่อนที่จะเริ่มสร้างแผนภาพกิจกรรมยูเอ็มแอล จำเป็นต้องติดตั้งปลั๊กอินเสริมเพื่อให้อีคลิปส์สามารถสร้างแผนภาพยูเอ็มแอลได้ ซึ่งวิธีการติดตั้งและรายละเอียดถูกเขียนเพิ่มเติมไว้ที่ภาคผนวก ข. ในการพัฒนาเราจะเริ่มต้นโดยทำการสร้างโครงการ (Project) ใหม่



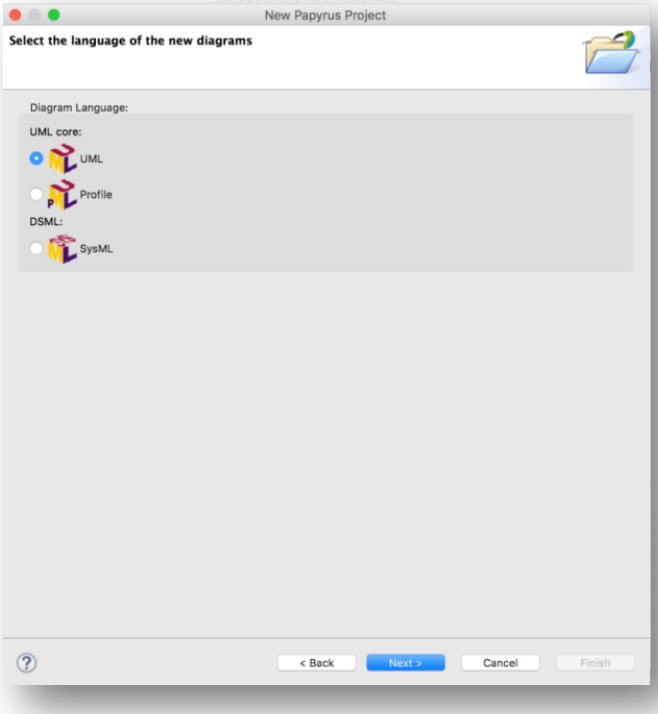
รูปที่ 20 เริ่มต้นการสร้างโปรเจคใหม่

เมื่อทำการสร้างโปรเจคเรียบร้อยแล้ว ขั้นต่อมาภายใต้ Wizards ให้ทำการเลือก ปาปิรุส (Papyrus) และ ปาปิรุส โปรเจค (Papyrus Project) ตามลำดับ



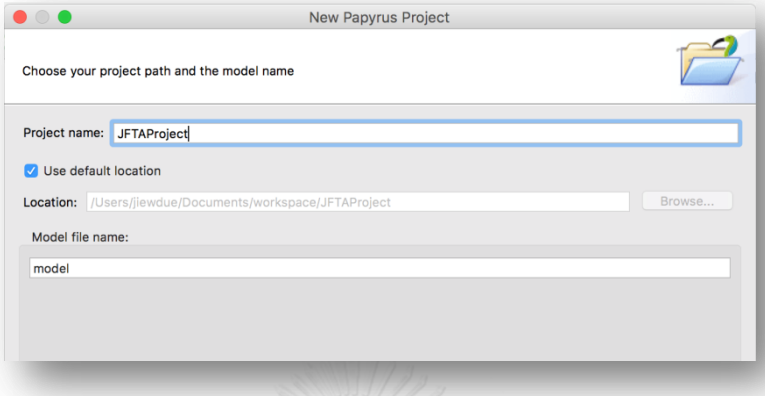
รูปที่ 21 การสร้างปาปิรุสโปรเจค

จากนั้นทำการเลือกยูเอ็มแอลคอร์ (UML core) เป็น ยูเอ็มแอล (UML) คลิก next ดังรูปที่ 22

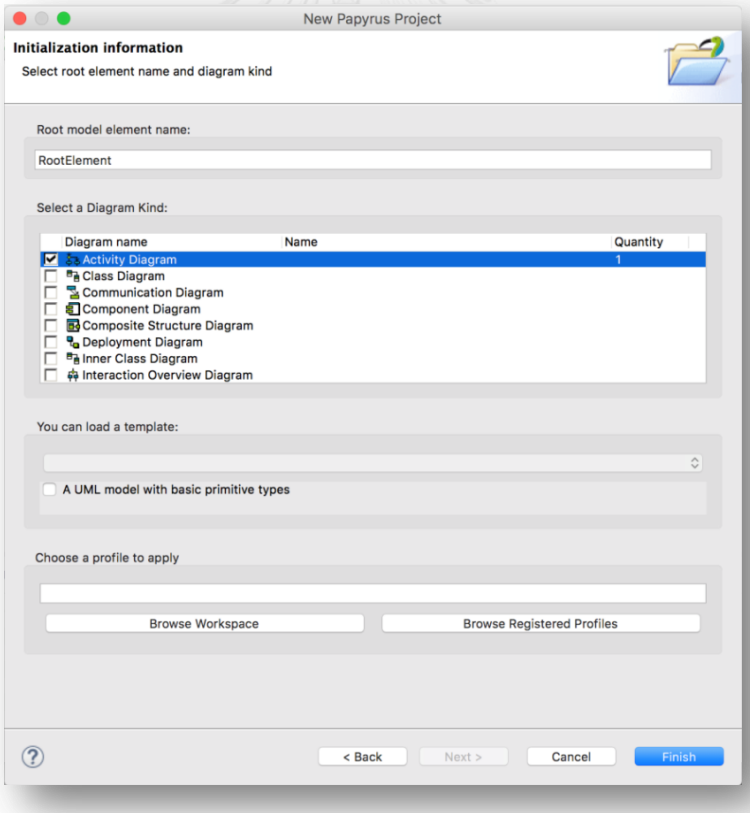


รูปที่ 22 เลือกประเภทยูเอ็มแอลคอร์

หลังจากนั้นทำการใส่ชื่อโปรเจคในช่อง Project Name แล้วทำการเลือกประเภทของแผนภาพยูเอ็มแอล (Select a Diagram Kind) เป็นแผนภาพกิจกรรม (Activity Diagram)



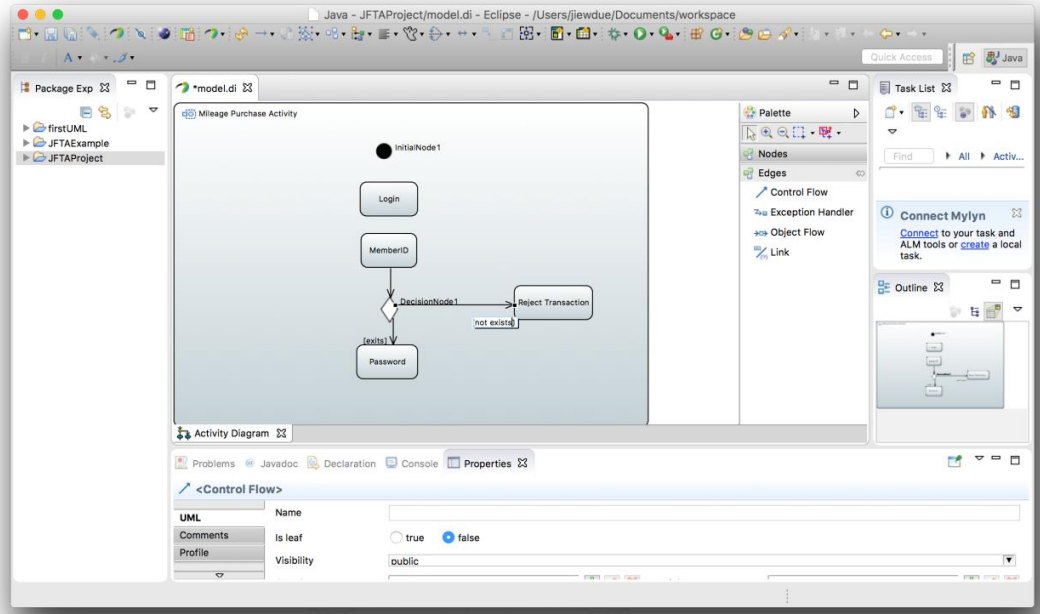
รูปที่ 23 การใส่ชื่อโปรเจค



รูปที่ 24 เลือกประเภทแผนภาพยูเอ็มแอล

หลังจากนั้นทำการสร้างแผนภาพกิจกรรมเพื่อที่จะทำเป็นเมทาเดตาของระบบต่อไป ลักษณะของโปรแกรมจะคล้ายโปรแกรมวาดแผนภาพอื่นๆ โดยที่ผู้ใช้งานสามารถลากสัญลักษณ์มาวางไว้ในกรอบของกิจกรรมที่ถูกสร้างไว้ได้ ดังรูปที่ 25

โดยในโปรแกรมสัญลักษณ์หลักที่ใช้ในแผนภาพกิจกรรมจะมีโหนด (node) และเส้น (edge)

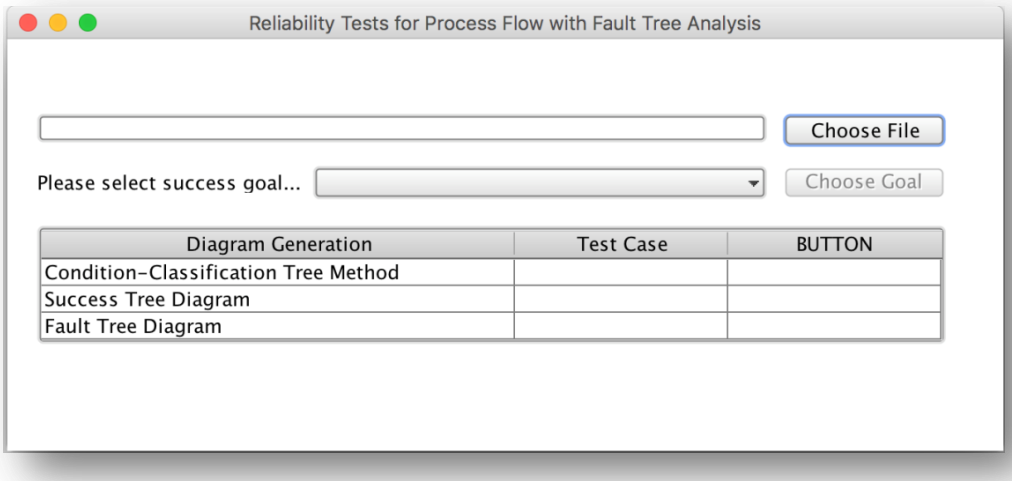


รูปที่ 25 หน้าจอปลั๊กอินปาปิรุส มาร์ส

4.4.3 การพัฒนาส่วนต่อประสาน

การพัฒส่วนต่อประสานสำหรับผู้ใช้งานให้สอดคล้องกับขอบเขตของระบบในบทที่ 1 และมีฟังก์ชันการทำงานตรงกันกับความต้องการด้านหน้าที่ (Function Requirements) ในบทที่ 4 หัวข้อย่อย 4.2 ในการพัฒนาระบบส่วนต่อประสาน งานวิจัยฉบับนี้จะใช้ไลบรารีของเจกราฟเอกซ์ (JGraphX) โดยใช้โปรแกรมเน็ตบีน (NetBeans 8.0.2) ในการพัฒนา

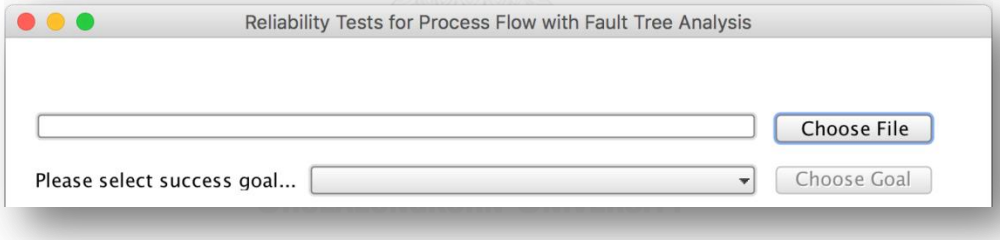
เมื่อเปิดระบบสร้างแผนภาพต้นไม้ความผิดพลาดขึ้นมา ระบบจะแสดงหน้าจอ ดังรูปที่ 26 ซึ่งระบบจะแบ่งหน้าจอหลักในการทำงานออกเป็น 2 ส่วน คือ ส่วนของการประมวลผลนำเข้าข้อมูล และส่วนแสดงผลลัพธ์ที่ได้จากการประมวลผลของข้อมูลนำเข้า



รูปที่ 26 หน้าจอของระบบสร้างแผนภาพต้นไม้ความผิดพลาด

4.4.3.1 ส่วนของการประมวลผลนำเข้าข้อมูล

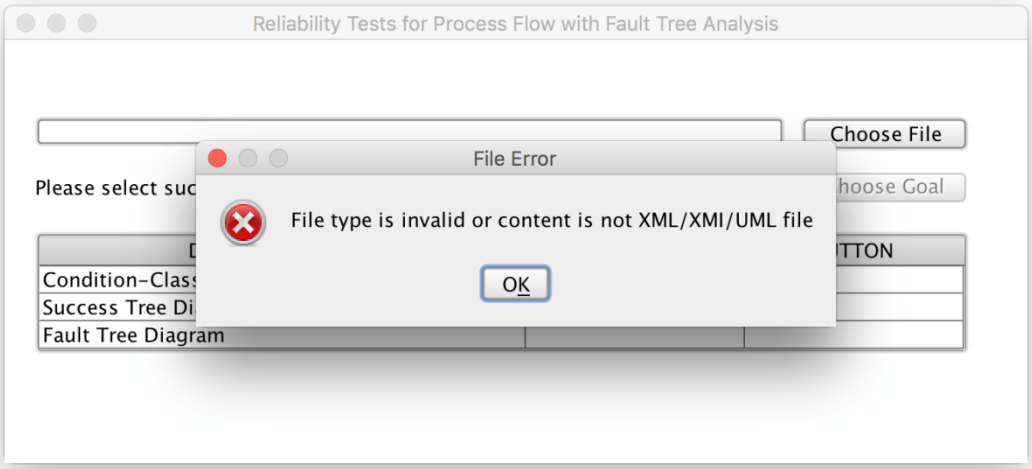
ในส่วนนี้เป็นส่วนของการนำเข้าข้อมูล โดยผู้ใช้จะต้องเลือกไฟล์นำเข้าข้อมูลเมตาเดตาของแผนภาพกิจกรรมซึ่งอยู่ในรูปแบบเอกซ์เอ็มไอ ดังรูปที่ 27



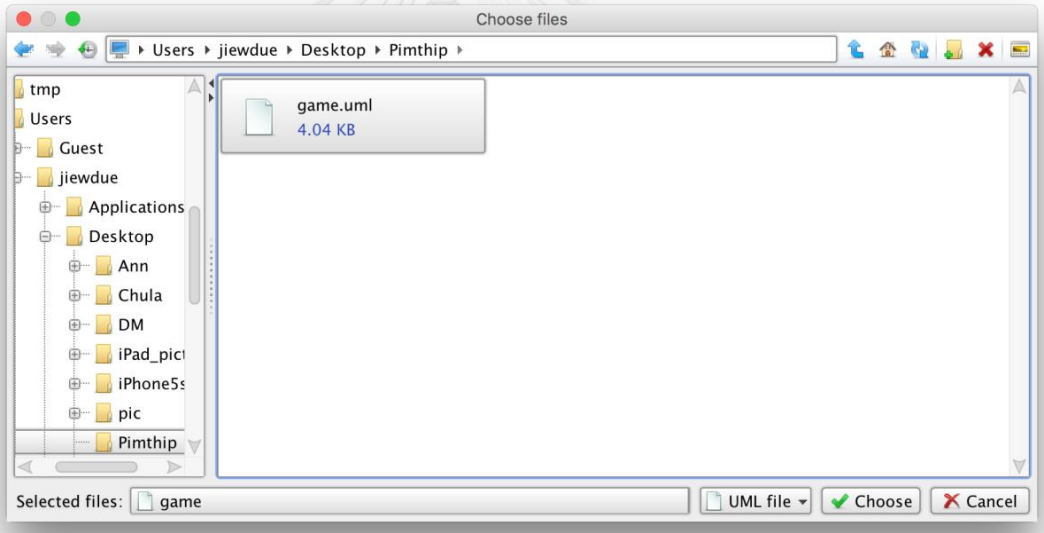
รูปที่ 27 หน้าจอในส่วนของการนำเข้าไฟล์เอกซ์เอ็มไอ

กรณีที่ผู้ใช้งานทำการเลือกไฟล์เอกสารนอกเหนือจากข้อกำหนดของระบบที่ได้กำหนดไว้ ระบบจะทำการแจ้งเตือนโดยขึ้นกล่องข้อความ (Popup message) ดังรูปที่ 28

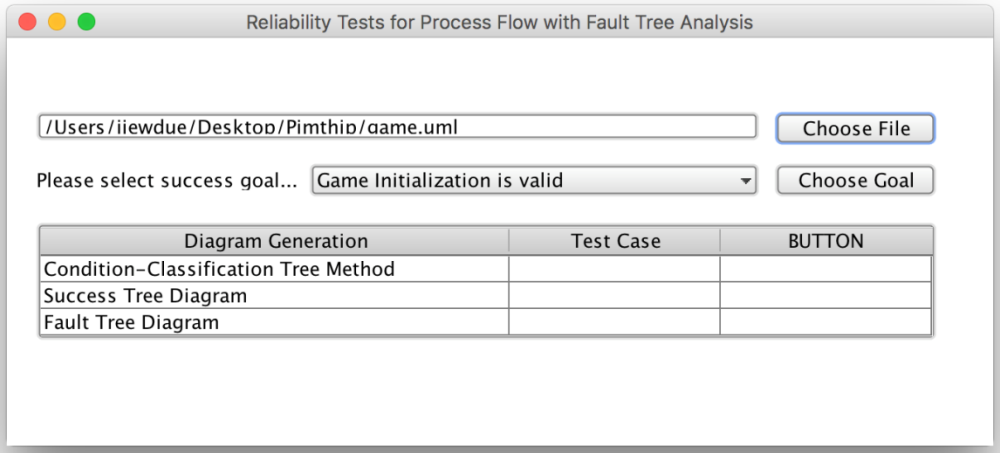
กรณีที่ผู้ใช้งานเลือกไฟล์ได้ถูกต้อง ระบบจะแสดงชื่อไฟล์ที่ผู้ใช้งานาเลือกในช่องข้อความ (Text box) ดังรูปที่ 29 และ 30



รูปที่ 28 หน้าจอแสดงการแจ้งเตือนกรณีการนำเข้าไฟล์ที่ผิดพลาด

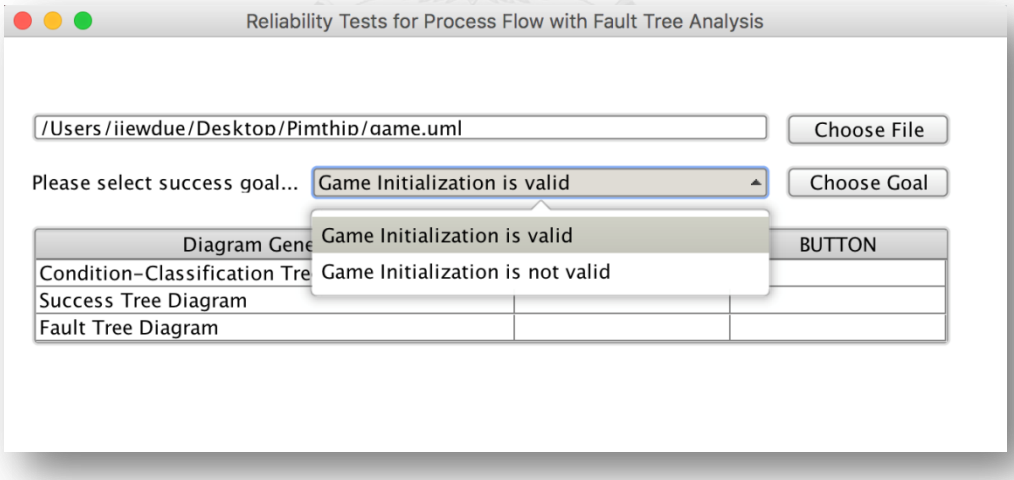


รูปที่ 29 หน้าจอแสดงการเลือกไฟล์เอกสารเอกซ์เอ็มไอ



รูปที่ 30 หน้าจอแสดงการนำเข้าข้อมูล

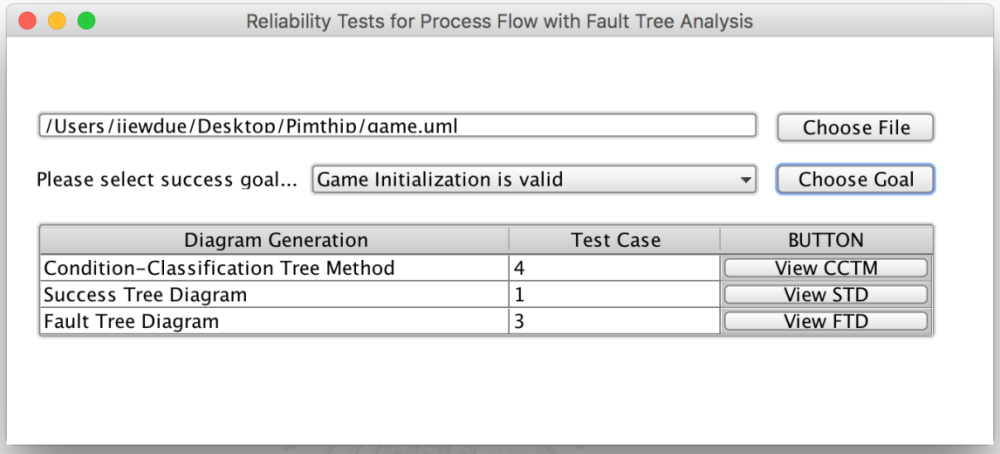
เมื่อผู้ใช้เลือกไฟล์นำเข้าที่ตรงตามข้อกำหนดระบบเรียบร้อยแล้ว ระบบจะแสดงเป้าหมายที่เป็นไปได้จากแผนภาพกิจกรรมให้ผู้ใช้เลือก ซึ่งผู้ใช้งานจะต้องเลือกเป้าหมายที่เป็นเป้าหมายที่ถูกต้อง ซึ่งสามารถทำให้กระบวนการทำงานในแผนภาพกิจกรรมสำเร็จได้



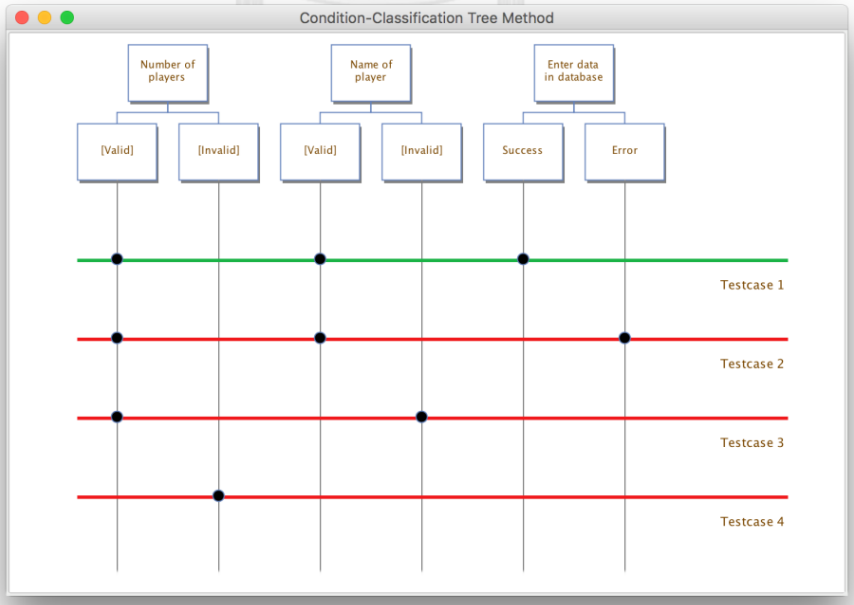
รูปที่ 31 หน้าจอแสดงการเลือกเป้าหมายของกระบวนการที่ถูกต้อง

4.4.3.2 ส่วนแสดงผลลัพธ์ที่ได้จากการประมวลผลของข้อมูลนำเข้า

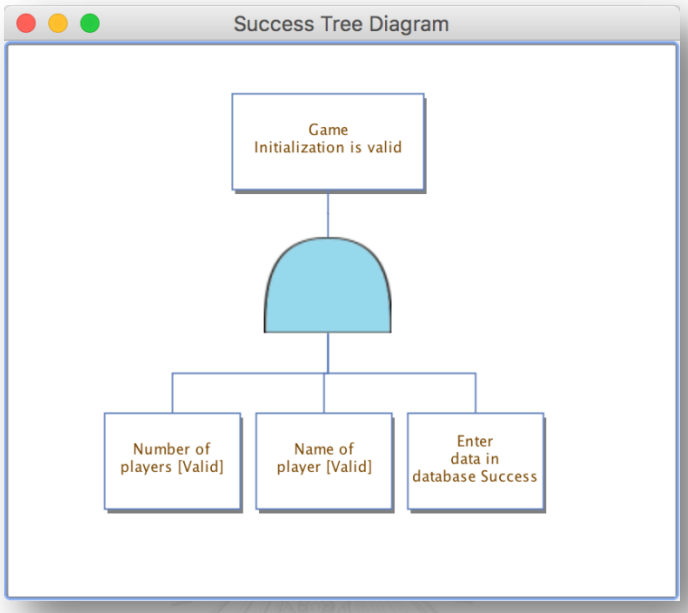
เมื่อผู้ใช้งานเลือกเป้าหมายที่ต้องการเรียบร้อยแล้ว ระบบจะแสดงจำนวนกรณีทดสอบที่เป็นไปได้ของแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข ต้นไม้ความสำเร็จ และต้นไม้ความผิดพลาด ดังรูปที่ 32 โดยผู้ใช้งานสามารถกดดูแผนภาพได้ตามลำดับ ดังรูปที่ 33, 34 และ 35



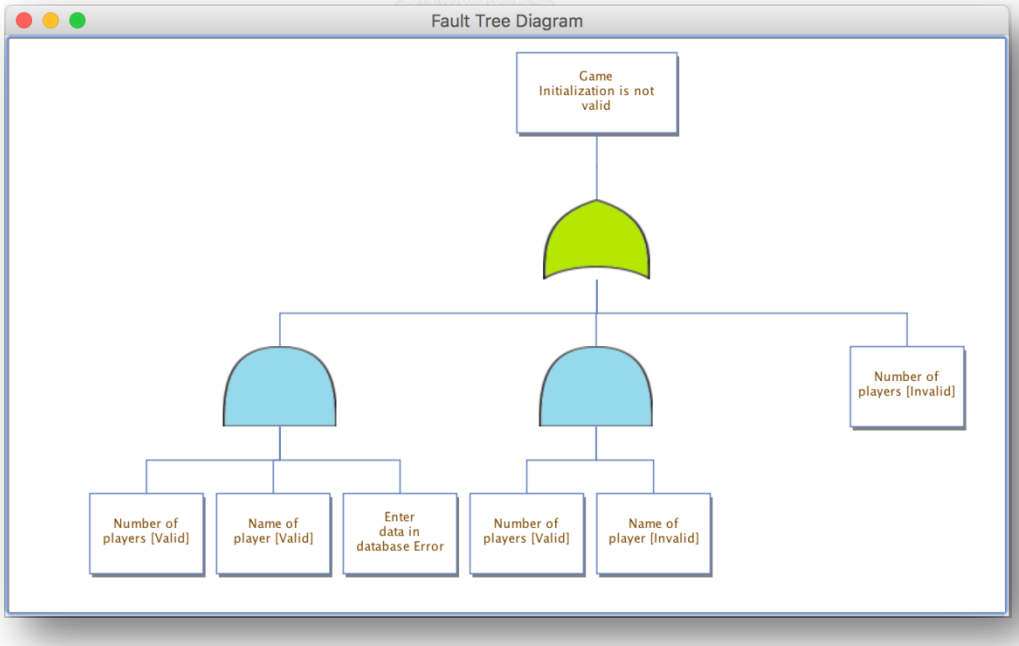
รูปที่ 32 หน้าจอแสดงผลกรณีทดสอบของระบบแสดงแผนภาพต้นไม้ความผิดพลาด



รูปที่ 33 หน้าจอแสดงผลต้นไม้การจำแนกแบบมีเงื่อนไข



รูปที่ 34 หน้าจอแสดงผลต้นไม้ความสำเร็จ



รูปที่ 35 หน้าจอแสดงผลต้นไม้ความผิดพลาด

บทที่ 5
การทดสอบและประเมินผลระบบ

5.1 การทดสอบระบบ

จากการทำงานของระบบที่ได้กล่าวไว้ในบทที่ 4 เราจำเป็นต้องมีการทดสอบการทำงานที่ได้ระบุไว้ ซึ่งการทดสอบในบทนี้เราจะใช้วิธีการทดสอบหน้าที่การทำงาน (Black Box Testing) ตามกรณีทดสอบดังต่อไปนี้ ซึ่งได้ถูกออกแบบตามการใช้งานของผู้ใช้

1) การทดสอบการนำเข้าไฟล์ยูเอ็มแอลเมทาเดตา (UML Metadata)
    1.1. ทดสอบรูปแบบการนำเข้าของไฟล์ยูเอ็มแอลเมทาเดตาในส่วนติดต่อกับผู้ใช้
2) ทดสอบการวิเคราะห์เป้าหมายที่เป็นไปได้ของยูเอ็มแอลเมทาเดตา
    2.1. ทดสอบการวิเคราะห์เป้าหมาย ในกรณีที่มีจุดตัดสินใจ 2 จุด
    2.2. ทดสอบการวิเคราะห์เป้าหมาย ในกรณีที่มีจุดตัดสินใจ 6 จุด
    2.3. ทดสอบการวิเคราะห์เป้าหมาย ในกรณีที่มีจุดตัดสินใจ 6 จุด แบบมีเป็นไปได้มากกว่า 1 ทางในการไปถึง เป้าหมายความสำเร็จ
    2.4. ทดสอบการวิเคราะห์เป้าหมาย ในกรณีที่มี จุดตัดสินใจ 2 จุด แบบมี Fork node และ Join node เป็นส่วนประกอบ
3) ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้าง แบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไข แผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ
    3.1. ทดสอบจำนวนของกรณีทดสอบที่ได้จากการวิเคราะห์แบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไข
    3.2. ทดสอบการแสดงผลของแบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไข
    3.3. ทดสอบจุดเชื่อมต่อของกรณีทดสอบในแบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไข
    3.4. ทดสอบจำนวนของกรณีทดสอบที่ได้จากการวิเคราะห์แผนภาพต้นไม้ความสำเร็จ
    3.5. ทดสอบการแสดงผลของแผนภาพต้นไม้ความสำเร็จ
    3.6. ทดสอบข้อมูลของส่วนประกอบด้านขาเข้า ในแผนภาพต้นไม้ความสำเร็จ
    3.7. ทดสอบข้อมูลของส่วนประกอบด้านขาออก ในแผนภาพต้นไม้ความสำเร็จ
    3.8. ทดสอบข้อมูลเหตุการณ์บนสุดของแผนภาพต้นไม้ความสำเร็จ
    3.9. ทดสอบจำนวนของกรณทดสอบที่จากการวิเคราะห์แผนภาพต้นไม้ความผิดพลาด

3.10. ทดสอบการแสดงผลของแผนภาพต้นไม้ความผิดพลาด

3.11. ทดสอบข้อมูลของส่วนประกอบด้านขาเข้า ในแผนภาพต้นไม้ผิดพลาด

3.12. ทดสอบข้อมูลของส่วนประกอบด้านขาออก ในแผนภาพต้นไม้ผิดพลาด

3.13. ทดสอบข้อมูลเหตุการณ์บนสุดของแผนภาพต้นไม้ผิดพลาด

ตารางที่ 18 กรณีทดสอบระบบ

# กรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการทดสอบ
1.1	การทดสอบการนำเข้าไฟล์ ยูเอ็มแอลเมทาเดตา	ทดสอบรูปแบบการนำเข้าของไฟล์ยูเอ็มแอลเมทาเดตาในส่วนติดต่อกับผู้ใช้	แสดงรายชื่อไฟล์ที่มีนามสกุล XML	ถูกต้อง
			แสดงชื่อไฟล์ที่ถูกเลือก	ถูกต้อง
			แจ้งเตือนกรณีไฟล์ไม่ได้อยู่ในรูปแบบ XML	ถูกต้อง
2.1	ทดสอบการวิเคราะห์เป้าหมายที่เป็นไปได้ของยูเอ็มแอลเมทาเดตา	ทดสอบการวิเคราะห์เป้าหมายในกรณีที่มีจุดตัดสินใจ 2 จุด	แสดงรายการเป้าหมายที่เป็นไปได้อย่างถูกต้อง	ถูกต้อง
2.2	ทดสอบการวิเคราะห์เป้าหมายที่เป็นไปได้ของยูเอ็มแอลเมทาเดตา	ทดสอบการวิเคราะห์เป้าหมายในกรณีที่มีจุดตัดสินใจ 6 จุด	แสดงรายการเป้าหมายที่เป็นไปได้อย่างถูกต้อง	ถูกต้อง
2.3	ทดสอบการวิเคราะห์เป้าหมายที่เป็นไปได้ของยูเอ็มแอลเมทาเดตา	ทดสอบการวิเคราะห์เป้าหมายในกรณีที่มีจุดตัดสินใจ 6 จุด แบบมีเป็นไปได้มากกว่า 1 ทาง ในการไปถึงเป้าหมาย	แสดงรายการเป้าหมายที่เป็นไปได้อย่างถูกต้อง	ถูกต้อง

# กรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการทดสอบ
		ความสำเร็จ		
2.4	ทดสอบการวิเคราะห์เป้าหมายที่เป็นไปได้ของยูเอ็มแอลเมทาเดตา	ทดสอบการวิเคราะห์เป้าหมายในกรณีที่มีจุดตัดสินใจ 2 จุด แบบมี Fork node และ Join node เป็นส่วนประกอบ	แสดงรายการเป้าหมายที่เป็นไปได้อย่างถูกต้อง	ถูกต้อง
3.1	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไข แผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบจำนวนของกรณีทดสอบที่ได้จากการวิเคราะห์แบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไข	ต้นไม้การจำแนกแบบมีเงื่อนไข ต้องมีจำนวนเท่ากับจุดตัดสินใจในยูเอ็มแอลเมทาเดตา	ถูกต้อง
3.2	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบการแสดงผลของแบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไข	ต้นไม้การจำแนกต้องมีชื่อตรงกับจุดตัดสินใจ	ถูกต้อง
			เงื่อนไขของต้นไม้การจำแนกต้องมีชื่อต้องกับ ผลลัพธ์ที่เป็นไปได้จากจุดตัดสินใจนั้น ๆ	ถูกต้อง
3.3	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบจุดเชื่อมต่อของกรณีทดสอบในแบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไข	แต่ละเงื่อนไขของต้นไม้การจำแนกจำเป็นต้องเชื่อมต่อกับเส้นกรณีทดสอบอย่างถูกต้อง	ถูกต้อง

# กรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการทดสอบ
3.4	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบจำนวนของกรณีทดสอบที่ได้จากการวิเคราะห์แผนภาพต้นไม้ความสำเร็จ	จำนวนข้อมูลขาเข้าที่นำไปสู่เหตุการณ์ขั้นบนสุดต้องมีจำนวนเท่ากับ กรณีทดสอบที่ถูกต้องในแผนภาพต้นไม้การจำแนก	ถูกต้อง
3.5	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบการแสดงผลของแผนภาพต้นไม้ความสำเร็จ	แบบจำลองเชิงตรรกะที่ถูกนำมาใช้ต้องถูกว่าอย่างถูกต้อง	ถูกต้อง
3.6	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบข้อมูลของส่วนประกอบด้านขาเข้าในแผนภาพต้นไม้ความสำเร็จ	ข้อมูลขาเข้าของแบบจำลองเชิงตรรกะต้องเป็นชื่อเดียวกับเงื่อนไขของกรณีทดสอบที่ได้จากแผนภาพต้นไม้การจำแนก	ถูกต้อง
3.7	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบข้อมูลของส่วนประกอบด้านขาออก ในแผนภาพต้นไม้ความสำเร็จ	ข้อมูลขาออกของแบบจำลองเชิงตรรกะต้องเป็นชื่อเดียวกับต้นไม้การจำแนก หรือเป็นชื่อเดียวกับเป้าหมายความสำเร็จที่ผู้ใช้ได้เลือกไว้	ถูกต้อง
3.8	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้าง	ทดสอบข้อมูลเหตุการณ์บนสุด	เหตุการณ์บนสุดของแผนภาพต้น	ถูกต้อง

# กรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการทดสอบ
	แผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ของแผนภาพต้นไม้ความสำเร็จ	ความสำเร็จจะต้องถูกแสดงถึงเป้าหมายความสำเร็จที่ถูกเลือกโดยผู้ใช้	
3.9	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบจำนวนของกรณีทดสอบที่ได้จากการวิเคราะห์แผนภาพต้นไม้ความผิดพลาด	จำนวนข้อมูลขาเข้าที่นำไปสู่เหตุการณ์ขั้นบนสุดต้องมีจำนวนเท่ากับ กรณีทดสอบที่ถูกต้องในแผนภาพต้นไม้การจำแนก	ถูกต้อง
3.10	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบการแสดงผลของแผนภาพต้นไม้ความผิดพลาด	แบบจำลองเชิงตรรกะที่ถูกนำมาใช้ต้องถูกว่าอย่างถูกต้อง	ถูกต้อง
3.11	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบข้อมูลของส่วนประกอบด้านขาเข้า ในแผนภาพต้นไม้ความผิดพลาด	ข้อมูลขาเข้าของแบบจำลองเชิงตรรกะต้องเป็นชื่อเดียวกับเงื่อนไขของกรณีทดสอบที่ได้จากแผนภาพต้นไม้การจำแนก	ถูกต้อง
3.12	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพ	ทดสอบข้อมูลของส่วนประกอบด้านขาออก ในแผนภาพต้นไม้ความผิดพลาด	ข้อมูลขาออกของแบบจำลองเชิงตรรกะต้องเป็นชื่อเดียวกับต้นไม้การจำแนก หรือเป็นชื่อเดียวกับ	ถูกต้อง

# กรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการทดสอบ
	ต้นไม้ความสำเร็จ		เป้าหมายที่ผู้ใช้ไม่ได้เลือกให้เป็นเป้าหมายความสำเร็จ	
3.13	ทดสอบผลลัพธ์ที่ได้จากการทำงานของการสร้างแผนภาพจำแนกแผนภาพต้นไม้ความผิดพลาด และแผนภาพต้นไม้ความสำเร็จ	ทดสอบข้อมูลเหตุการณ์บนสุดของแผนภาพต้นไม้ความผิดพลาด	เหตุการณ์บนสุดของแผนภาพต้นความผิดพลาดจะต้องถูกแสดงถึงเป้าหมายความสำเร็จที่ไม่ถูกเลือกโดยผู้ใช้	ถูกต้อง

5.2 การประเมินระบบ

การประเมินผลระบบจะประเมินโดย จะนำโปรแกรมต่าง ๆ ในรูปแบบของแผนภาพกิจกรรมยูเอ็มแอล โดยที่แต่ละโปรแกรมจะมีส่วนประกอบที่แตกต่างกันดังจะได้กล่าวต่อไปนี้

1) กรณีทดสอบที่ 1 โปรแกรมที่ประกอบด้วยแผนภาพกิจกรรมอย่างง่ายโดยที่มีจุดตัดสินใจไม่เกิน 3 จุด
2) กรณีทดสอบที่ 2 โปรแกรมที่ประกอบด้วยแผนภาพกิจกรรมที่มีทางสู่เป้าหมายความสำเร็จมากกว่า 1 ทาง
3) กรณีทดสอบที่ 3 โปรแกรมที่ประกอบด้วยแผนภาพกิจกรรมที่มี Fork node และ Join node เป็นส่วนประกอบ
4) กรณีทดสอบที่ 4 โปรแกรมที่ประกอบด้วยความซับซ้อนที่มี จุดตัดสินใจ 10 จุด มีทางสู่เป้าหมายความสำเร็จมากกว่า 1 ทาง และมี Fork Join เป็น ส่วนประกอบ

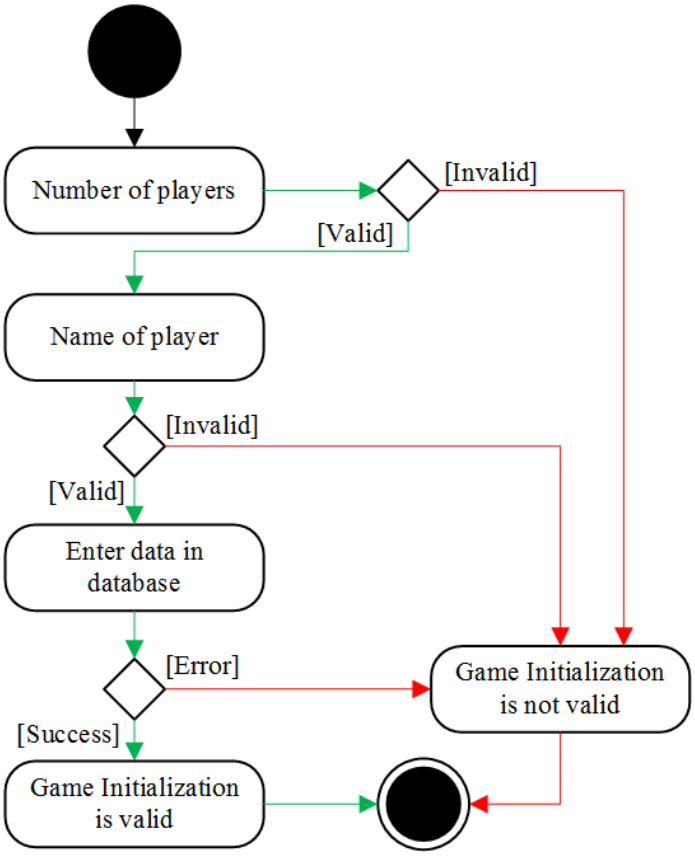
5.2.1 กรณีทดสอบที่ 1 โปรแกรมที่ประกอบด้วยแผนภาพกิจกรรมอย่างง่าย

สำหรับในกรณีนี้เราจะยกตัวอย่างอย่างง่ายของโปรแกรมการเริ่มต้นของเกมซึ่งประกอบไปด้วยจุดตัดสินใจ 3 จุดด้วยกันนั้นคือ การตรวจสอบจำนวนของผู้เล่น การตรวจสอบชื่อของผู้เล่น และการบันทึกข้อมูลลงฐานข้อมูล ซึ่งเป้าหมายความสำเร็จของโปรแกรมนี้คือ การเริ่มต้นเกมอย่างถูกต้องโดย

ที่เงื่อนไขทั้งสามนั้นต้องเป็นไปตามที่กำหนดเท่านั้น นั้นคือ จำนวนผู้เล่นที่ถูกต้อง ชื่อของผู้เล่นที่ถูกต้อง และ บันทึกข้อมูลลงฐานข้อมูลสำเร็จ ตามลำดับ

- แผนภาพกิจกรรมที่ต้องการทดสอบ

จากการทำงานของโปรแกรมการเริ่มต้นของเกมที่ได้กล่าวไปแล้วนั้น เราสามารถนำมาสร้างเป็นแผนภาพกิจกรรมยูเอ็มแอลได้ดังนี้



รูปที่ 36 แผนภาพกิจกรรมของ Game Initialization

- ข้อมูลนำเข้า

อย่างไรก็ตาม ข้อมูลแผนภาพกิจกรรมดังกล่าวจำเป็นต้องแปลงให้เป็น ยูเอ็มแอลเมทาเดตา ซึ่งจะอยู่ในรูปแบบของยูเอ็มแอล ซึ่งทางผู้วิจัยจะใช้โปรแกรมปาปิรุสในการแปลง โดยที่โปรแกรมนี้จะสร้าง เอกซ์เอมไอเมทาเดตาซึ่งตามมาตรฐานของโอเอ็มจี

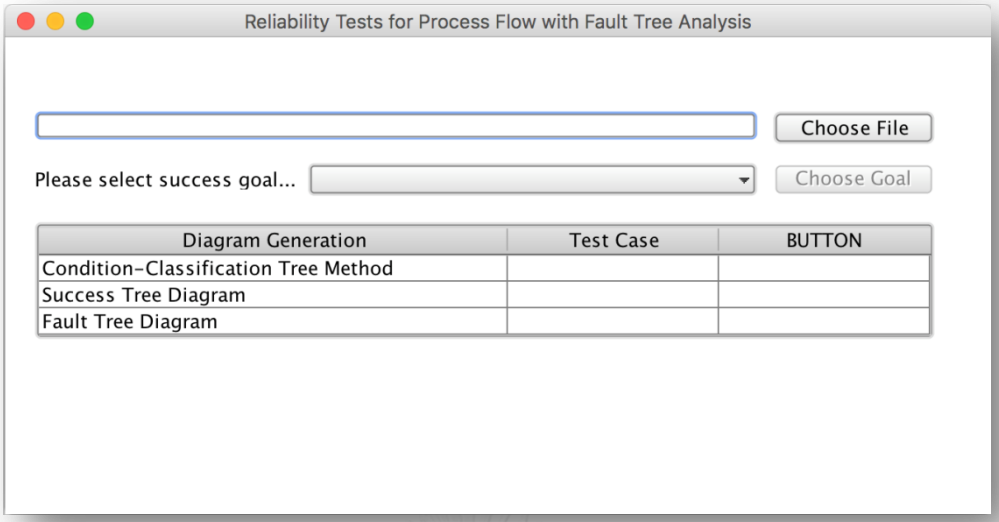
รูปที่ 37 เมทาเดตาของ Game Initialization

จากรูปแผนภาพกิจกรรมด้านบนจะสามารถแปลงเป็น ยูเอ็มแอลเมทาเดตาในรูปแบบยูเอ็มแอล ได้ดังนี้

จากยูเอ็มแอลไฟล์ที่ได้มานั้นจะสังเกตได้ว่าผลลัพธ์นั้นอยู่ในรูปแบบของเอกซ์เอ็มแอล (XML) ซึ่งส่วนประกอบแต่ละตัวจะถูกอธิบายอยู่ในเอกซ์เอ็มแอล เส้นและโหนดโดยที่จะมีคุณสมบัติเป็นการอธิบายในเชิงลึก ซึ่งประกอบไปด้วยประเภท หมายเลขประจำตัว ชื่อ และความสัมพันธ์ระหว่างส่วนประกอบ ในที่นี่จะเห็นได้ว่ามีจุดตัดสินใจทั้งหมด 3 จุด ตามที่ได้กล่าวไว้ในข้างต้น

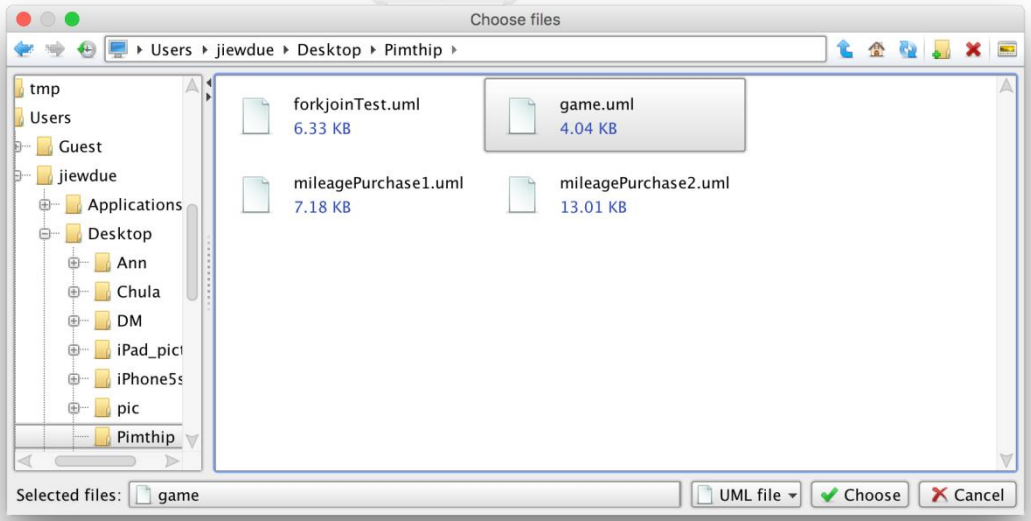
- ส่วนที่ติดต่อกับผู้ใช้

ไฟล์ที่ได้รับมานั้นสามารถนำมาเป็นข้อมูลนำเข้าของโปรแกรมสำหรับงานวิจัยนี้ได้ โดยที่โปรแกรมจะมีรูปแบบในส่วนติดต่อกับผู้ใช้ดังนี้



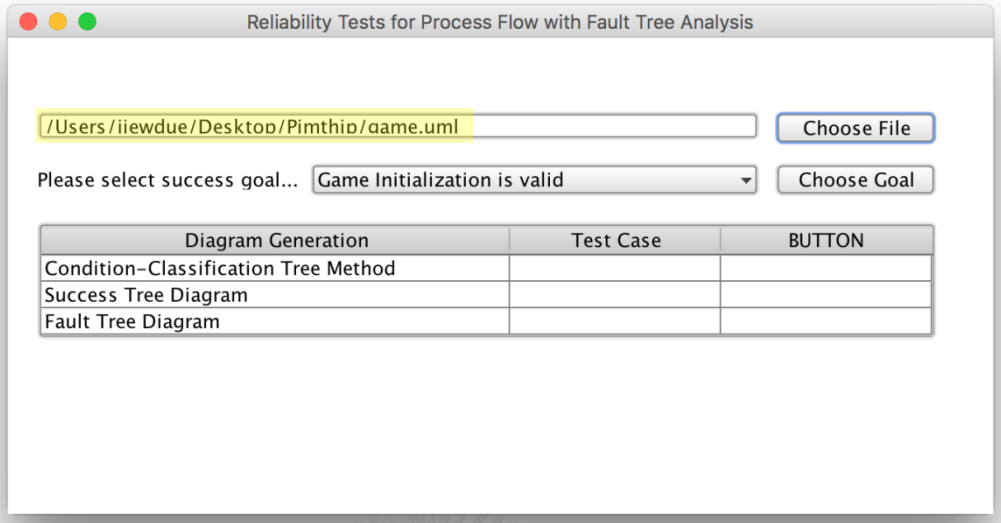
รูปที่ 38 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 1

ในขั้นต้นผู้ใช้จะมีเพียงทางเลือกเดียวคือการเลือกไฟล์ที่ปุ่ม “Choose File” จากนั้นจะมีหน้าต่างอีกหน้าต่างขึ้นมาเพื่อให้ผู้ใช้ทำการเลือกไฟล์เอกซ์เอ็มไอที่ต้องการ



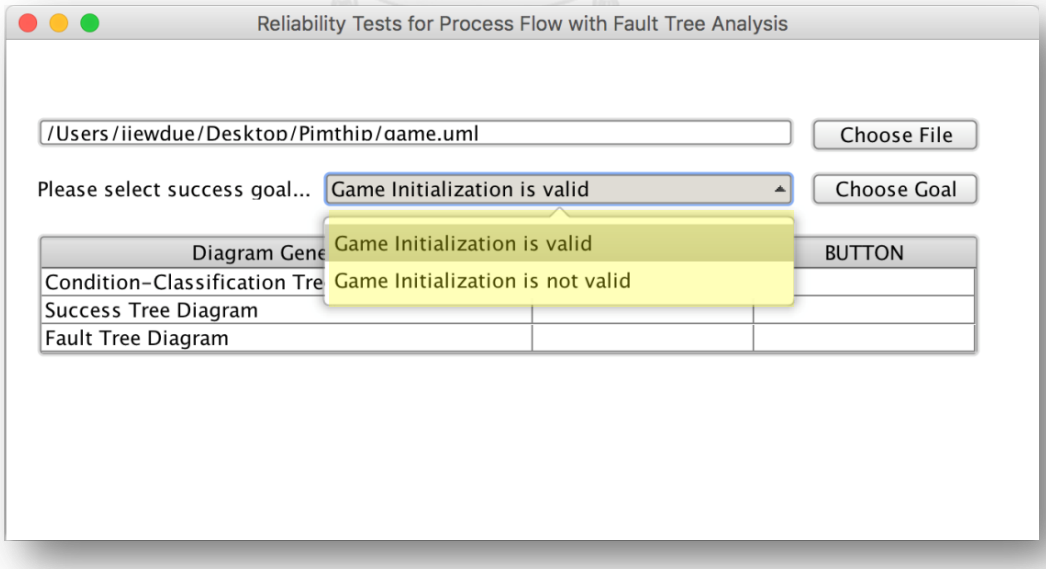
รูปที่ 39 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 2

เมื่อผู้ใช้ทำการเลือกไฟล์แล้ว โปรแกรมจะทำการแสดงชื่อไฟล์ที่ได้ถูกเลือกมาในกล่องข้อความ



รูปที่ 40 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 3

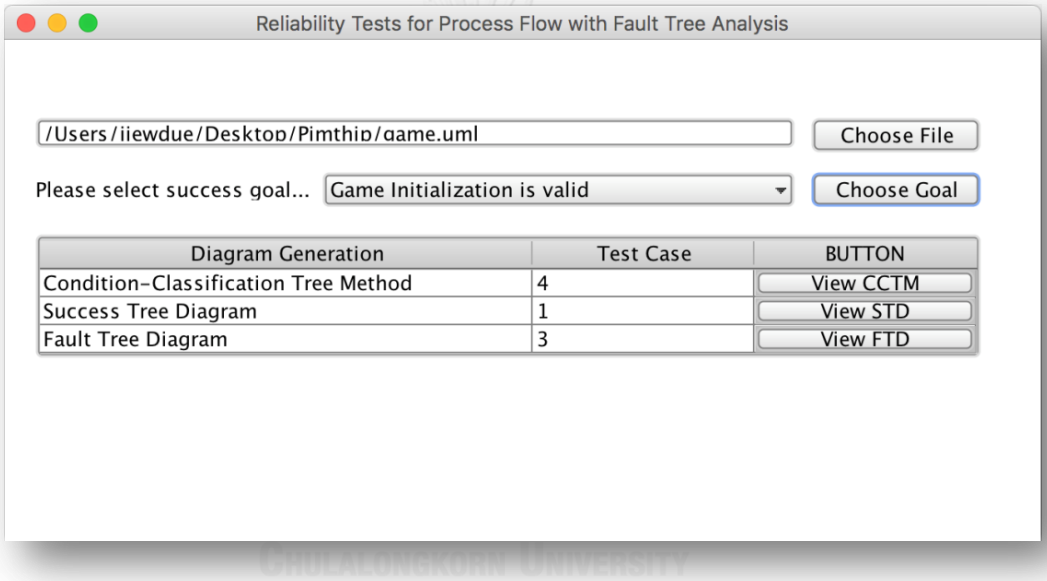
จากนั้นโปรแกรมต้องการให้ผู้ใช้เลือกเป้าหมายความสำเร็จหรือเป้าหมายที่ถูกต้อง โดยที่จะมีปุ่ม “Choose Goal ”ที่สามารถทำการคลิกได้แสดงขึ้นมา รวมถึงรายการเลือกแบบดึงลงแสดงให้เห็นถึง รายการเป้าหมายที่โปรแกรมได้ทำการวิเคราะห์เป้าหมายที่เป็นไปได้ออกมา



รูปที่ 41 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 4

จะเห็นได้ว่าโปรแกรมได้ทำการวิเคราะห์ออกได้อย่างถูกต้องโดยเป้าหมายที่เป็นไปได้คือ การเริ่มต้นเกมที่ถูกต้อง (Game Initialization is valid) และการเริ่มต้นเกมที่ไม่ถูกต้อง (Game Initialization is not valid)

เมื่อได้ทำการเลือก เป้าหมายความสำเร็จที่ต้องการแล้ว ทำการกดปุ่ม “Choose Goal” โปรแกรมจะทำการ วิเคราะห์และเริ่มการทำงานดังที่กล่าวในบทที่ 4 โดยที่โปรแกรมจะแสดงผลหลังจากทำงานเสร็จด้วยการแสดงถึงจำนวนกรณีทดสอบได้วิเคราะห์ออกมาในสามรูปแบบคือ แผนภาพจำลองต้นไม้จำแนกแบบมีเงื่อนไข แผนภาพต้นไม้ความสำเร็จ และแผนภาพต้นไม้ความผิดพลาด ดังรูปที่ 42

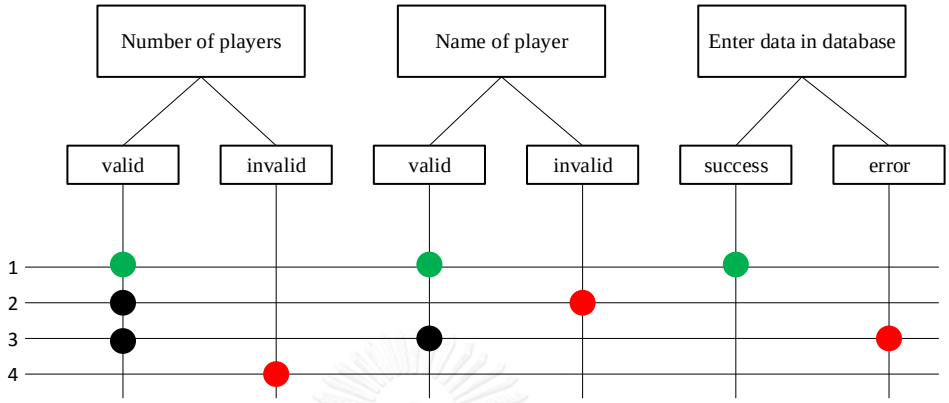


รูปที่ 42 หน้าจอระบบส่วนติดต่อกับผู้ใช้งาน 5

จากรูปจะเห็นได้ว่า แผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข (Condition-Classification Tree Method) ได้กรณีทดสอบทั้งหมด 4 กรณีทดสอบ สำหรับแผนภาพต้นไม้ความสำเร็จ มี เพียงกรณีทดสอบเพียง 1 กรณีเท่านั้น และ 3 กรณีทดสอบสำหรับแผนภาพต้นไม้ความผิดพลาด

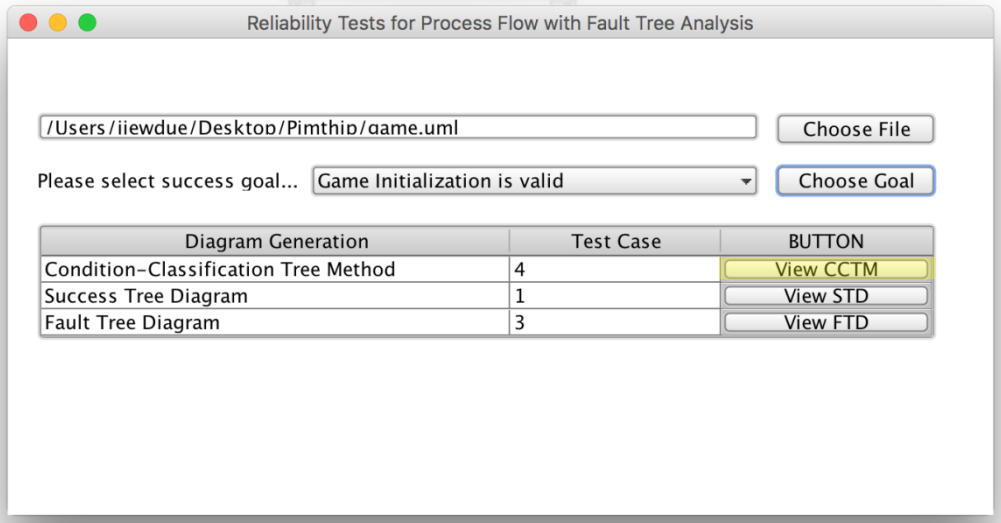
- การประเมินผลลัพธ์ แผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข

เนื่องจากบทนี้เป็นการประเมินการทำงานของระบบดังนั้นผู้วิจัยจำเป็นต้องทำการวิเคราะห์แผนภาพกิจกรรมยูเอ็มแอลที่ได้กล่าวมาด้วยวิธีการ การวิเคราะห์แบบแผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข ซึ่งได้ออกมาดังนี้



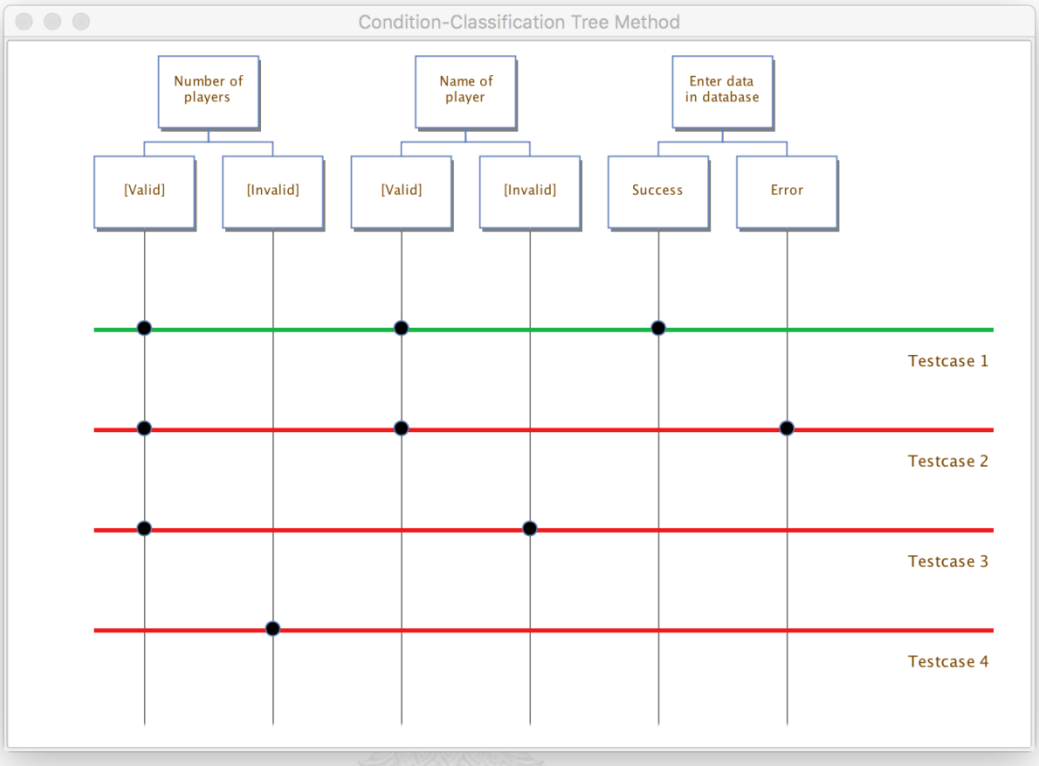
รูปที่ 43 CCTM จากตัวอย่างที่ 5.2.1 ที่ทำด้วยมือ

จากรูปจะเห็นได้ว่ามีกรณีทดสอบทั้งหมด 4 กรณีทดสอบ ซึ่งตรงกับผลลัพธ์ที่ได้จากการวิเคราะห์ด้วยโปรแกรมในงานวิจัยนี้ จากนั้นเราสามารถดูผลลัพธ์การวิเคราะห์ของโปรแกรมได้ด้วยการกดปุ่ม “View CCTM” เพื่อทำการแสดงผลในรูปแบบของแผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข



รูปที่ 44 หน้าจอแสดงผลลัพธ์ CCTM

จากนั้นโปรแกรมจะแสดงหน้าต่างใหม่ขึ้นมาและมีแผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข แสดงอยู่ดังรูปที่ 45

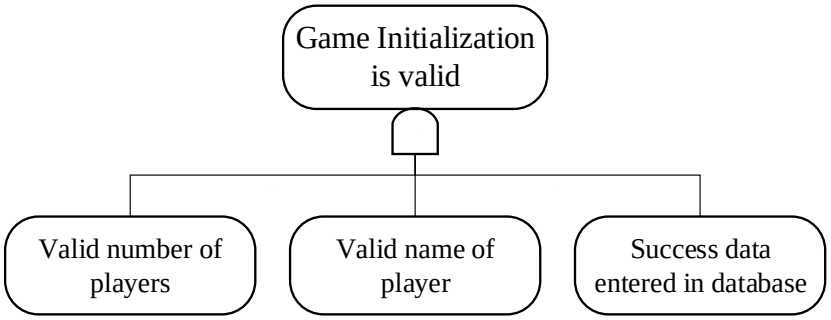


รูปที่ 45 CCTM จากตัวอย่างที่ 5.2.1 ที่สร้างด้วยระบบ

จากรูปจะเห็นได้ว่าโปรแกรมได้มีจำนวน กรณีทดสอบเท่านั้น คือ กรณีทดสอบและ มี 4 เงื่อนไขที่ตรงกันกับการวิเคราะห์ด้วยมือจากที่กล่าวมาในเบื้องต้น

- การประเมินผลลัพธ์ แผนภาพต้นไม้ความสำเร็จ

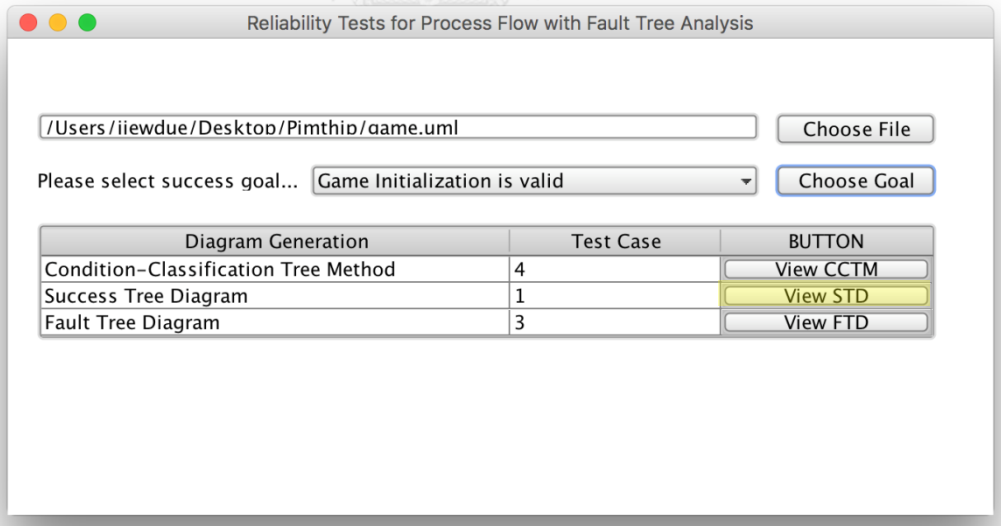
ลำดับต่อมาเป็นการประเมินผลลัพธ์ การวิเคราะห์แบบแผนภาพต้นไม้ความสำเร็จ ในกรณีนี้ทางผู้วิจัยได้ทำการวิเคราะห์แผนภาพกิจกรรมออกมาได้ดังนี้



รูปที่ 46 STD จากตัวอย่างที่ 5.2.1 ที่ทำด้วยมือ

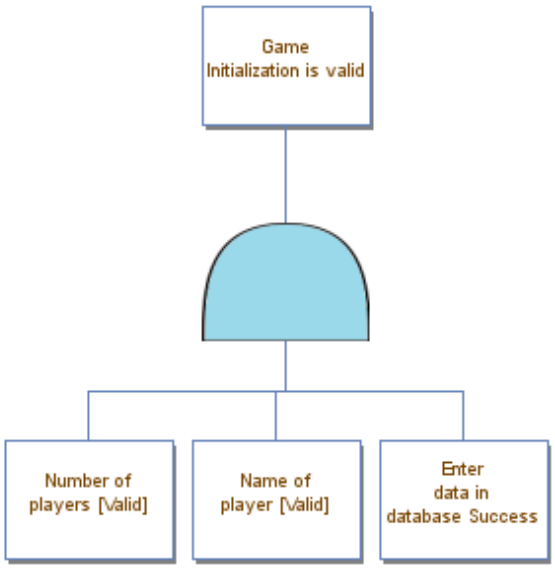
จากรูปที่ 46 จะเห็นได้ว่า เหตุการณ์บนสุดจะแสดงถึงเป้าหมายที่เราได้ทำการเลือกไว้นั้นคือ Game initialization is valid และมีแบบจำลองเชิงตรรกะ แอนด์เกตโดยมีข้อมูลขาเข้าสามเงื่อนไขนั้นคือ จำนวนผู้เล่นถูกต้อง ชื่อผู้เล่นถูกต้อง และการบันทึกลงฐานข้อมูลสำเร็จ

ในด้านของโปรแกรมของงานวิจัยนี้ ผู้ใช้สามารถกดปุ่ม “View STD” เพื่อทำการแสดงผลลัพธ์ในรูปแบบของแผนภาพต้นไม้ความสำเร็จได้



รูปที่ 47 หน้าจอแสดงผลลัพธ์ STD

เมื่อทำการกดปุ่มจะมีหน้าต่างแสดงออกมาดังรูปที่ 48

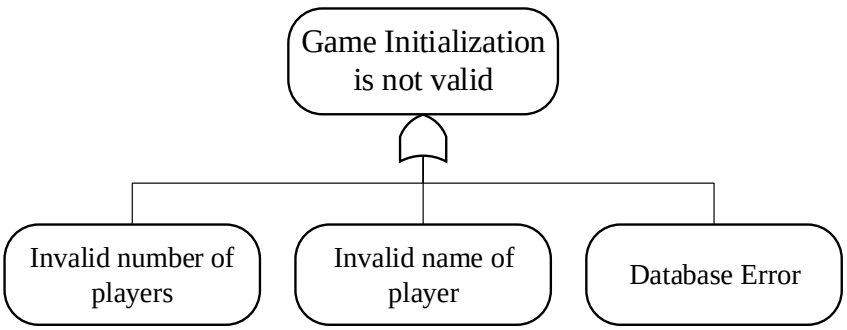


รูปที่ 48 STD จากตัวอย่างที่ 5.2.1 ที่สร้างด้วยระบบ

จะเห็นได้ว่ามีเป้าหมายความสำเร็จคือเป้าหมายที่ผู้ใช้ได้ทำการเลือกนั้นคือ Game Initialization is valid และมีแบบจำลองเชิงตรรกะ แอนด์เกตโดยมีข้อมูลขาเข้าสามเงื่อนไข นั้นคือ จำนวนผู้เล่นถูกต้อง ชื่อผู้เล่นถูกต้อง และการบันทึกลงฐานข้อมูลสำเร็จ ซึ่งจากการลัพธ์ที่ได้แสดงออกมานั้น การประเมินนี้จึงกล่าวได้ว่าโปรแกรมมีการทำงานที่ถูกต้อง

- การประเมินผลลัพธ์ แผนภาพต้นไม้ความผิดพลาด

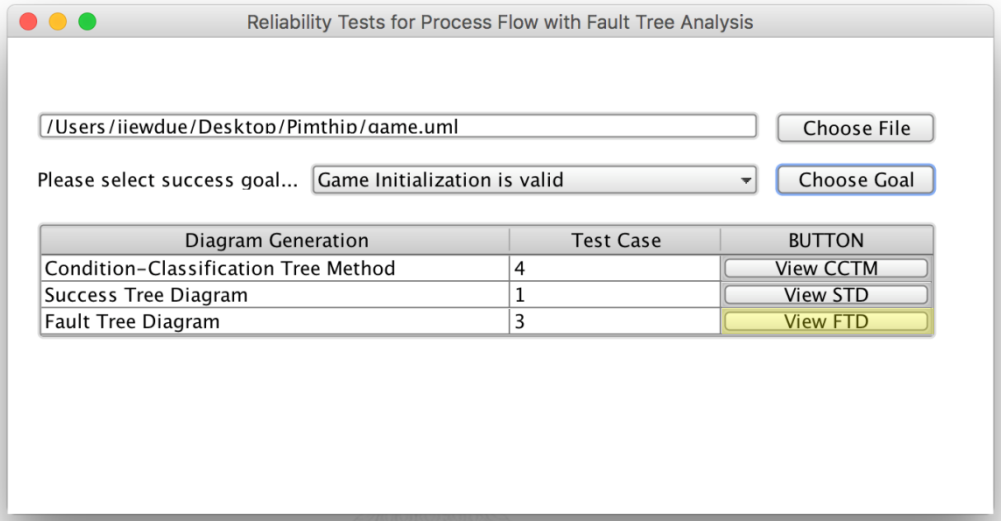
ลำดับต่อมาเป็นการประเมินผลลัพธ์ การวิเคราะห์แบบแผนภาพต้นไม้ความผิดพลาด ในกรณีนี้ทางผู้วิจัยได้ทำการวิเคราะห์แผนภาพกิจกรรมออกมาได้ดังนี้



รูปที่ 49 FTD จากตัวอย่างที่ 5.2.1 ที่ทำด้วยมือ

จากรูปที่ 49 จะเห็นได้ว่า เหตุการณ์บนสุดจะแสดงถึงเป้าหมายที่ผู้ใช้ไม่ได้ทำการเลือกไว้นั้น คือ Game initialization is not valid และมีแบบจำลองเชิงตรรกะ ออร์เกตโดยมีข้อมูลขาเข้าสามเงื่อนไข นั้นคือ จำนวนผู้เล่นไม่ถูกต้อง ชื่อผู้เล่นไม่ถูกต้อง และการบันทึกลงฐานข้อมูลผิดพลาด

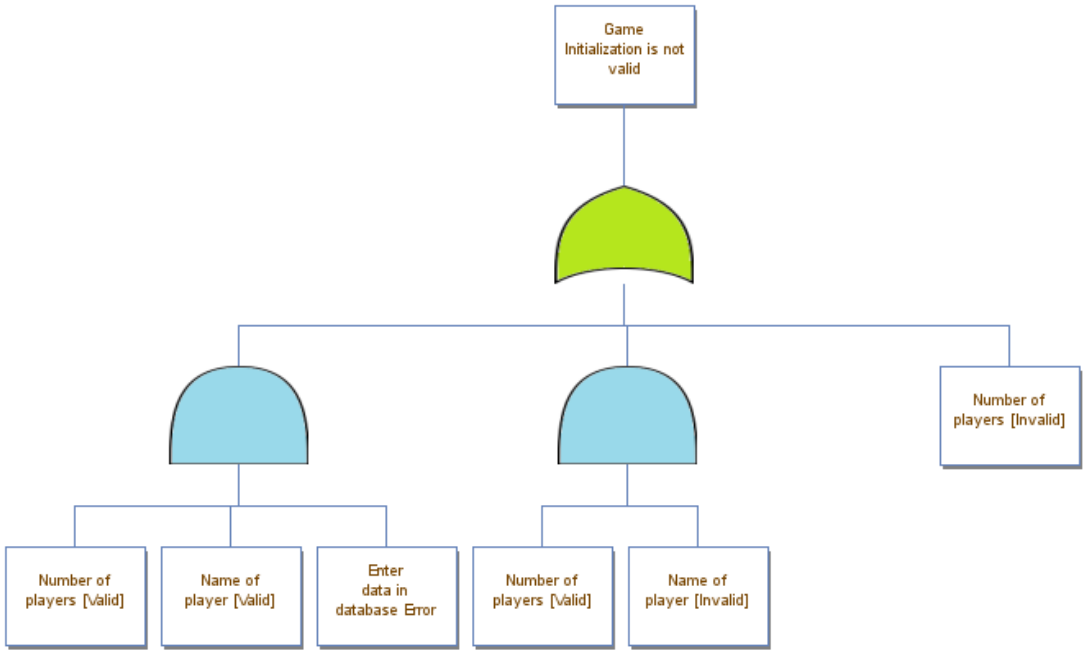
ในด้านของโปรแกรมของงานวิจัยนี้ ผู้ใช้สามารถกดปุ่ม “View FTD” เพื่อทำการแสดงผลลัพธ์ในรูปแบบของแผนภาพต้นไม้ความผิดพลาดได้



รูปที่ 50 หน้าจอแสดงผลลัพธ์ FTD

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เมื่อทำการกดปุ่มจะมีหน้าต่างแสดงออกมาดังรูปที่ 51



รูปที่ 51 FTD จากตัวอย่างที่ 5.2.1 ที่สร้างด้วยระบบ

จะเห็นได้ว่ามีเป้าหมายความผิดพลาดคือเป้าหมายที่ผู้ใช้ไม่ได้ทำการเลือกนั้นคือ Game Initialization is not valid และมีแบบจำลองเชิงตรรกะ ออร์เกต แต่ในผลลัพธ์นี้จะมีข้อแตกต่างกันอย่างนึงนั้นคือทางโปรแกรมจะวิเคราะห์ถึงการทดสอบที่จะเกิดขึ้นได้จริงนั้นกล่าวคือในข้อมูลขาเข้าของ ออร์เกต จากการทำด้วยมือนั้นประกอบไปด้วย จำนวนผู้เล่นไม่ถูกต้อง ชื่อผู้เล่นไม่ถูกต้อง และการบันทึกลงฐานข้อมูลผิดพลาด แต่ในการทดสอบจริงนั้นการที่ผู้ทดสอบจะสามารถสร้างกรณีชื่อผู้เล่นที่ไม่ถูกต้องได้ จำเป็นต้องบรรลุเงื่อนไขจำนวนผู้เล่นที่ถูกต้องได้เสียก่อน เช่นเดียวกันกับกรณีบันทึกข้อมูลลงฐานข้อมูลผิดพลาด ผู้ทดสอบโปรแกรมจำเป็นต้องผ่านบรรลุสองเงื่อนไขก่อนนั้นคือจำนวนผู้เล่นถูกต้อง และชื่อผู้เล่นถูกต้องจึงสามารถสร้างกรณีทดสอบของการบันทึกฐานข้อมูลผิดพลาดได้ ดังนั้นผู้วิจัยจึงทำการสรุป

การประเมินแผนภาพต้นไม้ความสำเร็จนั้น ทางโปรแกรมได้วิเคราะห์และแสดงผลลัพธ์ออกมาในลักษณะคล้ายกัน โดยที่ทางโปรแกรมจะมีข้อมูลในรายละเอียดที่ลึกกว่า

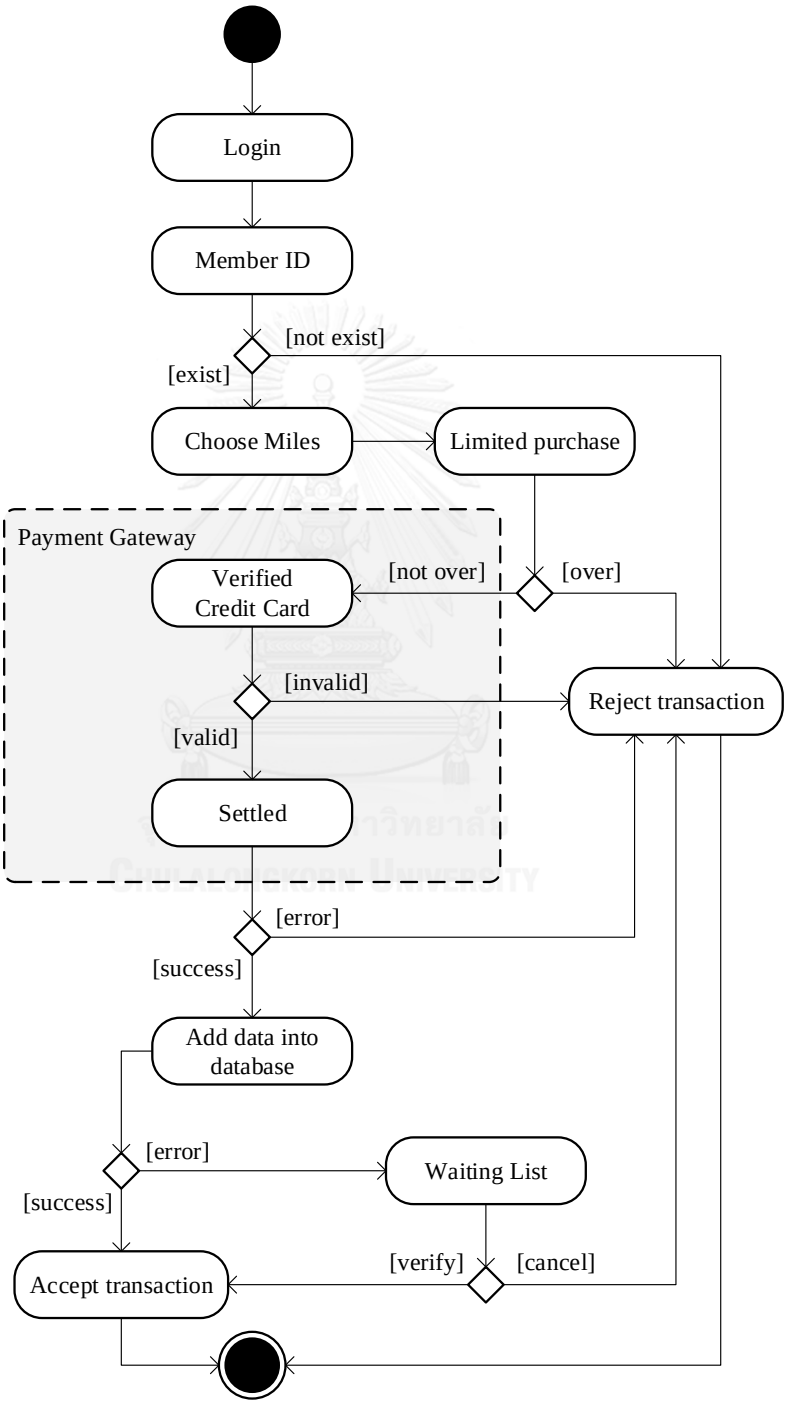
5.2.2 กรณีทดสอบที่ 2 โปรแกรมที่ประกอบด้วยแผนภาพกิจกรรมที่มีทางสู่เป้าหมายความสำเร็จมากกว่า 1 ทาง

สำหรับในกรณีนี้เราจะยกตัวอย่างของโปรแกรมการซื้อไมล์สะสมซึ่งประกอบไปด้วยจุดตัดสินใจ 6 จุดด้วยกันนั้นคือ การทำงานของโปรแกรมจะมีลักษณะดังนี้

1) ผู้ใช้จำเป็นต้องเข้าสู่ระบบสมาชิก
2) โปรแกรมจะทำการตรวจสอบสมาชิก
    a. หากสมาชิกมีอยู่ในระบบ จะดำเนินการขั้นต่อไป
    b. หากสมาชิกไม่มีอยู่ในระบบ โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์
3) ตรวจสอบข้อจำกัดในการสั่งซื้อ
    a. หากการซื้อไม่เกินข้อจำกัด โปรแกรมจะเดินการขั้นต่อไป
    b. หากการซื้อเกินข้อจำกัด โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์
4) ระบบจะเข้าสู่ชำระเงิน
    a. หากบัตรเครดิตผ่านการทวนสอบ โปรแกรมจะเดินการขั้นต่อไป
    b. หากบัตรเครดิตไม่ผ่านการทวนสอบ โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์
5) ระบบชำระเงินจะทำการยืนยันการชำระเงิน
    a. หากการชำระเงินสำเร็จ โปรแกรมจะเดินการขั้นต่อไป
    b. หากการชำระเงินไม่สำเร็จ โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์
6) การบันทึกข้อมูลลงฐานข้อมูล
    a. หากการบันทึกสำเร็จ โปรแกรมจบการทำงานด้วยการยอมรับการสั่งซื้อไมล์
    b. หากการบันทึกไม่สำเร็จ โปรแกรมจะเดินการขั้นต่อไป
7) โปรแกรมจะใส่ข้อมูลอยู่ในระบบรอการอนุมัติ
    a. หากการได้รับการอนุมัติ โปรแกรมจบการทำงานด้วยการยอมรับการสั่งซื้อไมล์
    b. หากการไม่ได้รับการอนุมัติ โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์

- แผนภาพกิจกรรมที่ต้องการทดสอบ

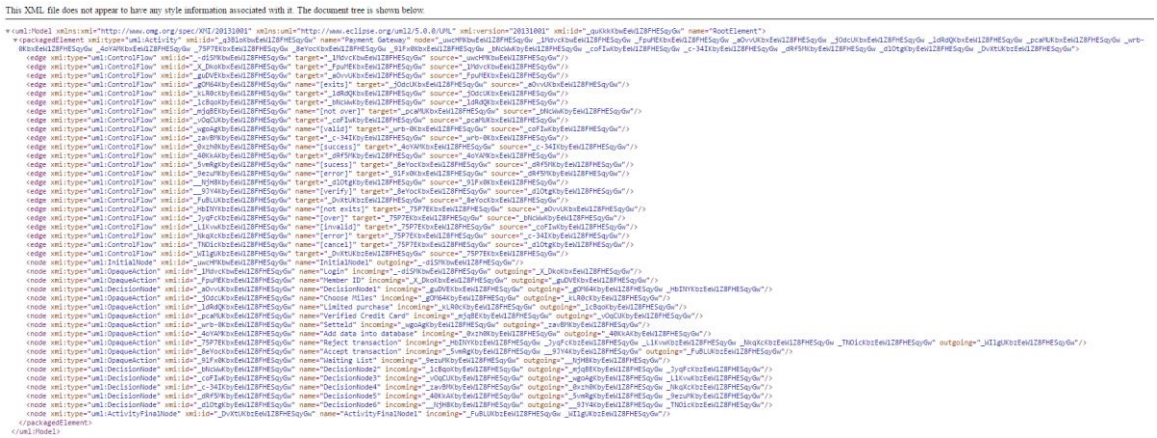
จากการทำงานของโปรแกรมการสั่งซื้อไมล์ที่ได้กล่าวไปแล้วนั้น เราสามารถนำมาสร้างเป็นแผนภาพกิจกรรมยูเอ็มแอลได้ดังนี้



รูปที่ 52 แผนภาพกิจกรรมการซื้อไมล์สะสมแบบง่าย

- ข้อมูลนำเข้า

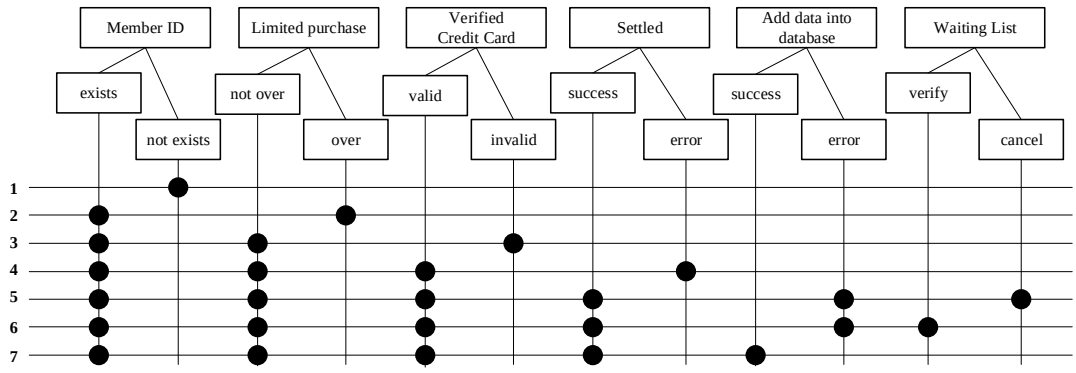
ข้อมูลแผนภาพกิจกรรมดังกล่าวจำเป็นต้องแปลงให้เป็นยูเอ็มแอลเมทาเดตาซึ่งจะอยู่ในรูปแบบของยูเอ็มแอล (UML) ซึ่งทางผู้วิจัยจะใช้โปรแกรมปาปิรุสในการแปลง โดยที่โปรแกรมนี้จะสร้าง เอ็กซ์เอมไอ เมทาเดตา ซึ่งตามมาตรฐานของโอเอ็มจีจะได้ผลลัพธ์ออกมาดังรูป



รูปที่ 53 เมทาเดตาของแผนภาพกิจกรรมการซื้อไมล์สะสมแบบง่าย

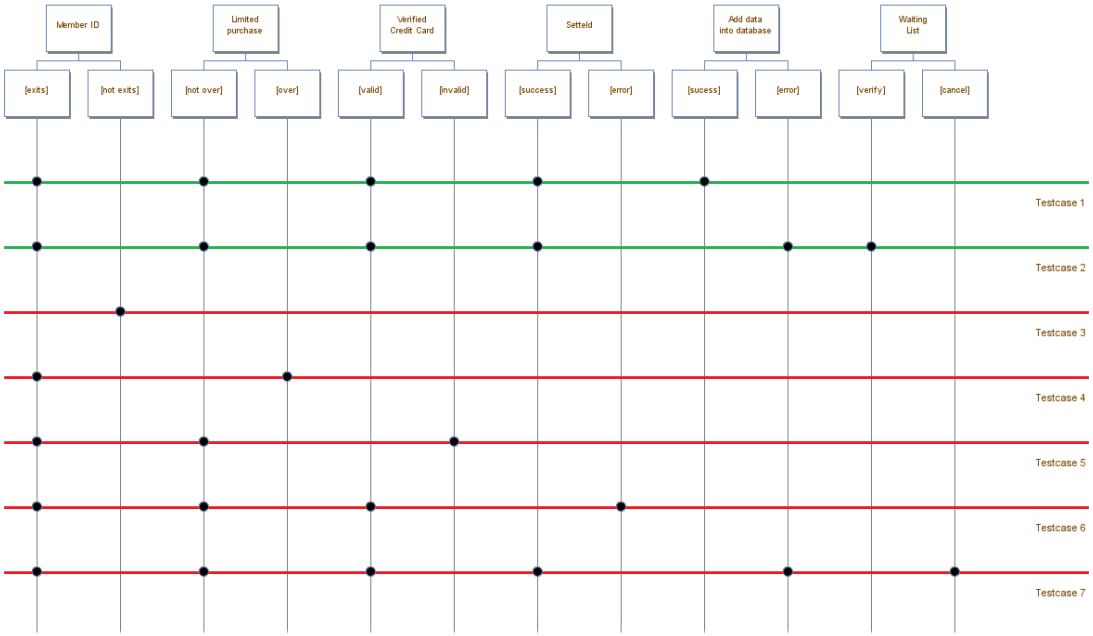
- การประเมินผลลัพธ์ แผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข

เช่นเดียวกับกรณีที่ได้กล่าวมานั้น ผู้วิจัยได้ทำการวิเคราะห์แผนภาพกิจกรรมยูเอ็มแอลที่ได้กล่าวมาด้วยวิธีการ การวิเคราะห์แบบแผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข ซึ่งได้ออกมาดังนี้



รูปที่ 54 CCTM จากตัวอย่างที่ 5.2.2 ที่ทำด้วยมือ

เมื่อได้ทำการใช้โปรแกรมในงานวิจัยนี้ วิเคราะห์ด้วยแผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข จะได้ผลลัพธ์ดังนี้

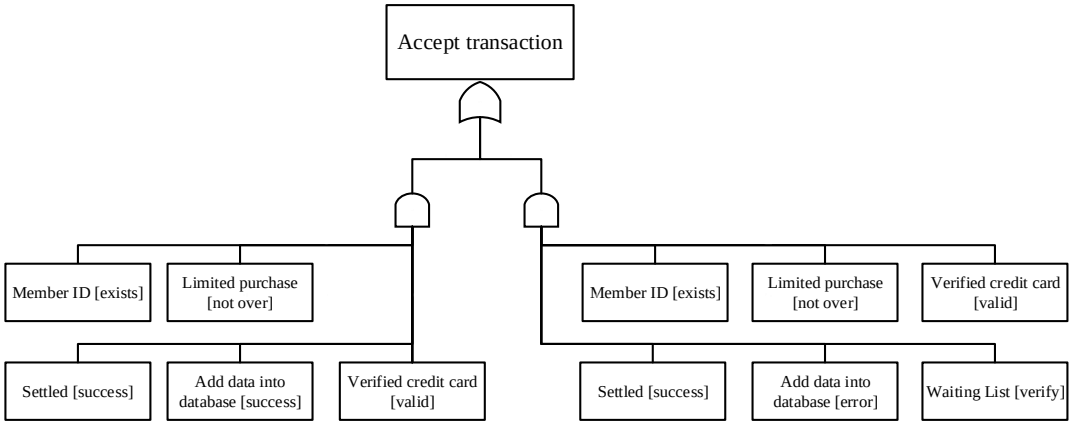


รูปที่ 55 CCTM จากตัวอย่างที่ 5.2.2 ที่สร้างด้วยระบบ

จากการเปรียบเทียบจะพบว่า การวิเคราะห์ด้วยโปรแกรมและการวิเคราะห์ด้วยมือได้ผลลัพธ์ที่เหมือนกัน นั้นคือ ได้กรณีทดสอบทั้งหมด 7 กรณีทดสอบและเงื่อนไขตรงกันทุกประการ

- การประเมินผลลัพธ์ แผนภาพต้นไม้ความสำเร็จ

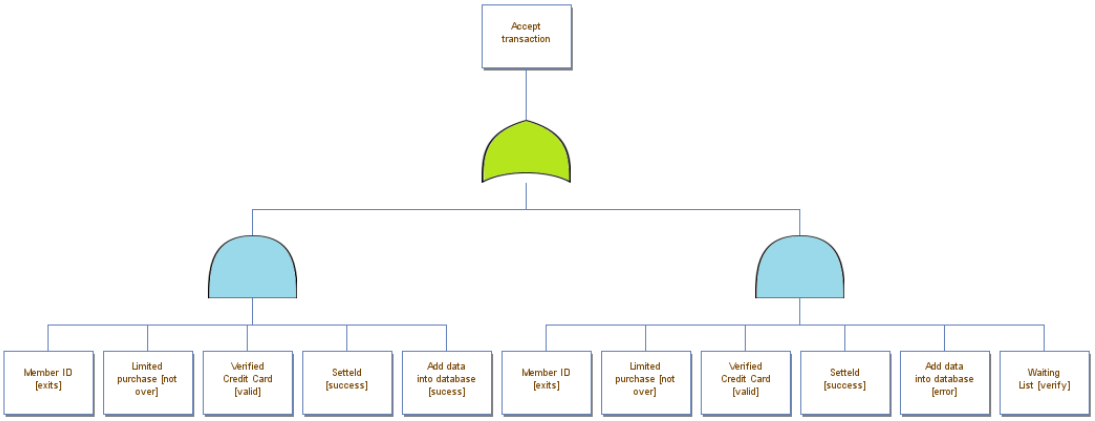
ลำดับต่อมาเป็นการประเมินผลลัพธ์ การวิเคราะห์แบบแผนภาพต้นไม้ความสำเร็จ ในกรณีนี้ทางผู้วิจัยได้ทำการวิเคราะห์แผนภาพกิจกรรมออกมาได้ดังนี้



รูปที่ 56 STD จากตัวอย่างที่ 5.2.2 ที่ทำด้วยมือ

ในกรณีจะสังเกตได้ว่า เงื่อนไขในการไปถึงเป้าหมายความสำเร็จที่ได้ตั้งไว้นั้น มีเส้นทางที่สามารถไปถึงได้ 2 ทาง จากทฤษฎีแล้วเราควรจะใช้แบบจำลองเชิงตรรกะออร์เกต เข้ามาช่วยซึ่งในกรณีนี้เราจะนำออร์เกตมาวางอยู่ด้านบนสุด เพื่อเป็นเงื่อนไขก่อนไปถึงเป้าหมายความสำเร็จที่ได้ตั้งไว้ ในระดับต่อมา เราจะใช้แบบจำลองเชิงตรรกะแอนด์เกตตามปกติ สำหรับแผนภาพต้นไม้ความสำเร็จ

เมื่อได้ทำการใช้โปรแกรมในงานวิจัยนี้ วิเคราะห์ด้วยแผนภาพจำลองต้นไม้ความสำเร็จ จะได้ผลลัพธ์ดังนี้

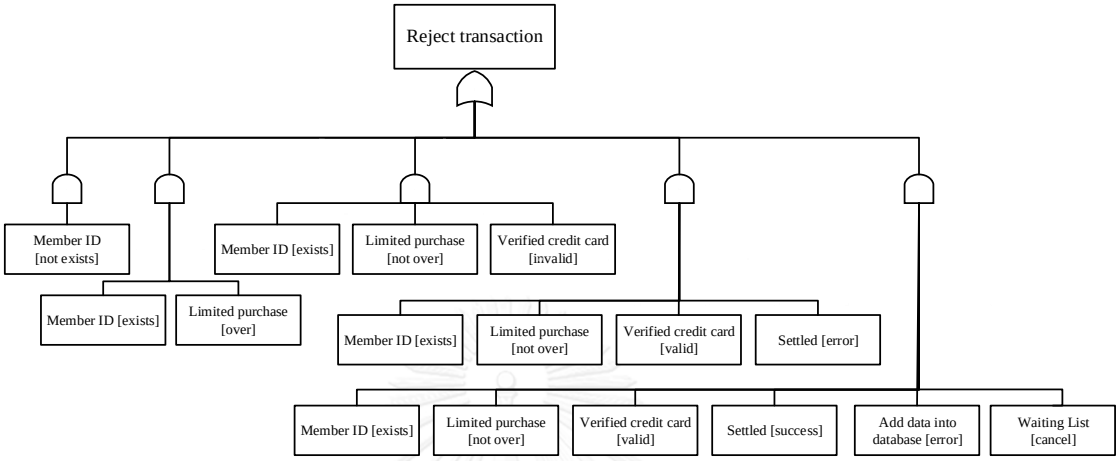


รูปที่ 57 STD จากตัวอย่างที่ 5.2.2 ที่สร้างด้วยระบบ

จากการเปรียบเทียบจะพบว่า การวิเคราะห์ด้วยโปรแกรมและการวิเคราะห์ด้วยมือได้ผลลัพธ์ที่เหมือนกัน นั้นคือเงื่อนไขและแบบจำลองเชิงตรรกะเหมือนกันทุกประการ

- การประเมินผลลัพธ์ แผนภาพต้นไม้ความผิดพลาด

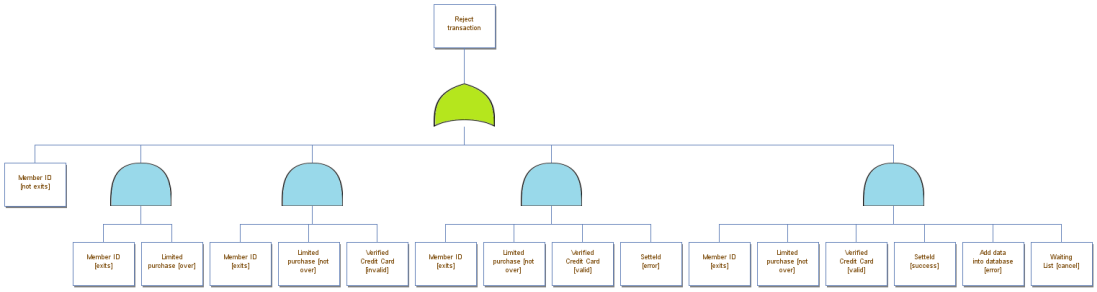
ลำดับต่อมาเป็นการประเมินผลลัพธ์ การวิเคราะห์แบบแผนภาพต้นไม้ความผิดพลาด ในกรณีนี้ทางผู้วิจัยได้ทำการวิเคราะห์แผนภาพกิจกรรมออกมาได้ดังนี้



รูปที่ 58 FTD จากตัวอย่างที่ 5.2.2 ที่ทำด้วยมือ

จากรูปจะสังเกตได้ว่าผู้วิจัยได้นำ แบบจำลองเชิงตรรกะ แอนด์เกต มาใช้เพื่อให้ได้ผลลัพธ์ที่ใกล้เคียงการทดสอบจริง ดังที่ได้กล่าวในกรณีทดสอบแรกของหัวข้อย่อยนี้

เมื่อได้ทำการใช้โปรแกรมในงานวิจัยนี้ วิเคราะห์ด้วยแผนภาพจำลองต้นไม้ความผิดพลาด จะได้ผลลัพธ์ดังนี้



รูปที่ 59 FTD จากตัวอย่างที่ 5.2.2 ที่ทำสร้างด้วยระบบ

จากการเปรียบเทียบจะพบว่า การวิเคราะห์ด้วยโปรแกรมและการวิเคราะห์ด้วยมือได้ผลลัพธ์ที่เหมือนกัน นั้นคือเงื่อนไขและแบบจำลองเชิงตรรกะเหมือนกันทุกประการ

กล่าวโดยสรุปจากกรณีทดสอบที่ 2 โปรแกรมที่ประกอบด้วย แผนภาพกิจกรรมที่มีทางสู่เป้าหมายความสำเร็จมากกว่า 1 ทาง ซึ่งในที่นี่ใช้โปรแกรมการสั่งซื้อไมล์ โปรแกรมในการวิจัยนี้สามารถทำงานได้ผลลัพธ์ตรงกับการทำงานด้วยมือทุกประการ

5.2.3 กรณีทดสอบที่ 3 โปรแกรมที่ประกอบด้วย แผนภาพกิจกรรมที่มี Fork node และ Join node เป็นส่วนประกอบ

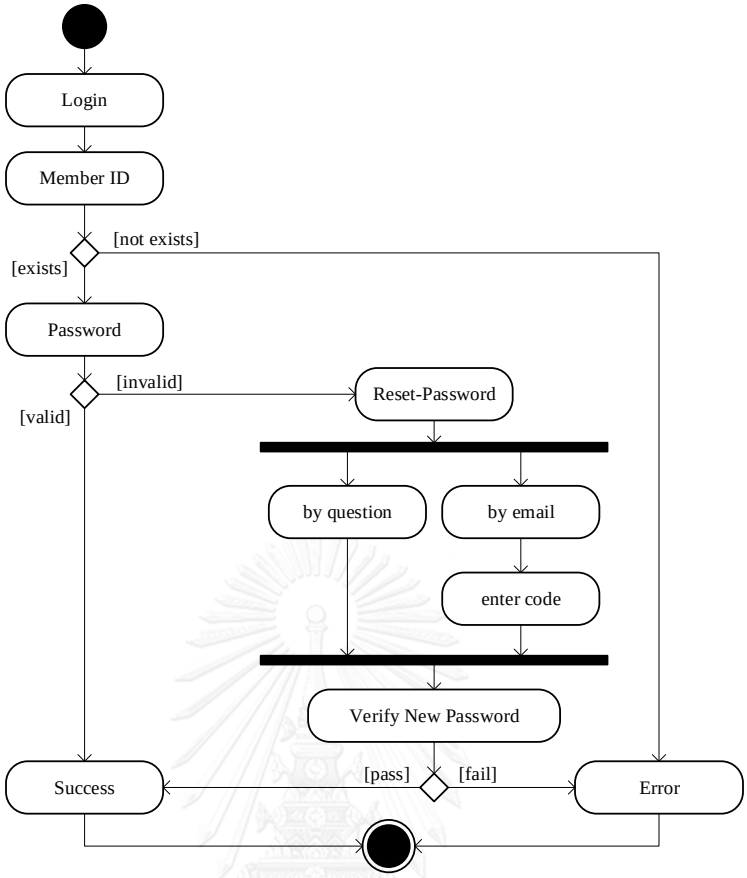
สำหรับในกรณีนี้เราจะยกตัวอย่างของโปรแกรมการตรวจสอบการเข้าสู่ระบบ ซึ่งประกอบไปด้วยจุดตัดสินใจ 3 จุด และมี Fork node กับ Join node เป็นส่วนประกอบด้วย นั้นคือ การทำงานของโปรแกรมจะมีลักษณะดังนี้

1) จำเป็นต้องเข้าสู่ระบบสมาชิก
2) โปรแกรมจะทำการตรวจสอบสมาชิก
    a. หากสมาชิกมีอยู่ในระบบ จะดำเนินการขั้นต่อไป
    b. หากสมาชิกไม่มีอยู่ในระบบ โปรแกรมจะจบการทำงานด้วยการแจ้งเตือนความผิดพลาด
3) โปรแกรมจะทำการตรวจสอบรหัสผ่าน
    a. หากรหัสผ่านถูกต้อง โปรแกรมจะจบการทำงานด้วยการเข้าสู่ระบบที่สำเร็จ
    b. หากรหัสผ่านไม่ถูกต้อง โปรแกรมจะดำเนินการขั้นต่อไป
4) ระบบจะเข้าสู่ระบบการตั้งรหัสผ่านใหม่ ซึ่งการตั้งรหัสผ่านใหม่จะสามารถแบ่งได้สองทางคือ
    a. การตั้งรหัสผ่านใหม่ด้วยคำถาม
    b. การตั้งรหัสผ่านใหม่ด้วยอีเมล
5) โปรแกรมจะทำการตรวจสอบความถูกต้องของการตั้งรหัสผ่านใหม่
    a. หากการตั้งรหัสผ่านสำเร็จ โปรแกรมจะจบการทำงานด้วยการเข้าสู่ระบบที่สำเร็จ
    b. หากการตั้งรหัสผ่านไม่สำเร็จ โปรแกรมจะจบการทำงานด้วยการแจ้งเตือนความผิดพลาด

- แผนภาพกิจกรรมที่ต้องการทดสอบ

จากการทำงานของโปรแกรมการเข้าสู่ระบบที่ได้กล่าวไปแล้วนั้น เราสามารถนำมาสร้างเป็นแผนภาพกิจกรรมยูเอ็มแอลได้ดังนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 60 แผนภาพกิจกรรมตัวอย่างแบบมีส่วนประกอบ Fork และ Join

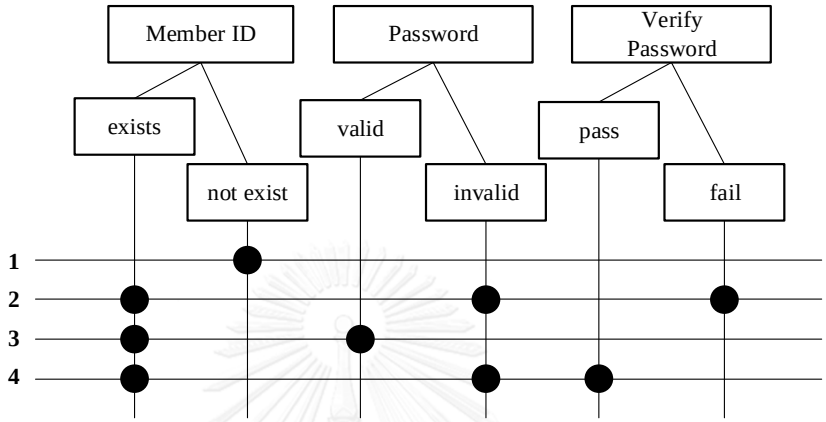
- ข้อมูลนำเข้า

ข้อมูลแผนภาพกิจกรรมดังกล่าวจำเป็นต้องแปลงให้เป็นยูเอ็มแอลเมทาเดตา ซึ่งจะอยู่ในรูปแบบของยูเอ็มแอล (UML) ซึ่งทางผู้วิจัยจะใช้โปรแกรมปาปิรุสในการแปลง โดยที่โปรแกรมนี้จะสร้าง เอ็กซ์เอมไอ เมทาเดตา ซึ่งตามมาตรฐานของโอเอ็มจี จะได้ผลลัพธ์ออกมาดังรูป

รูปที่ 61 เมทาเดตาของแผนภาพกิจกรรมตัวอย่างแบบมีส่วนประกอบ Fork และ Join

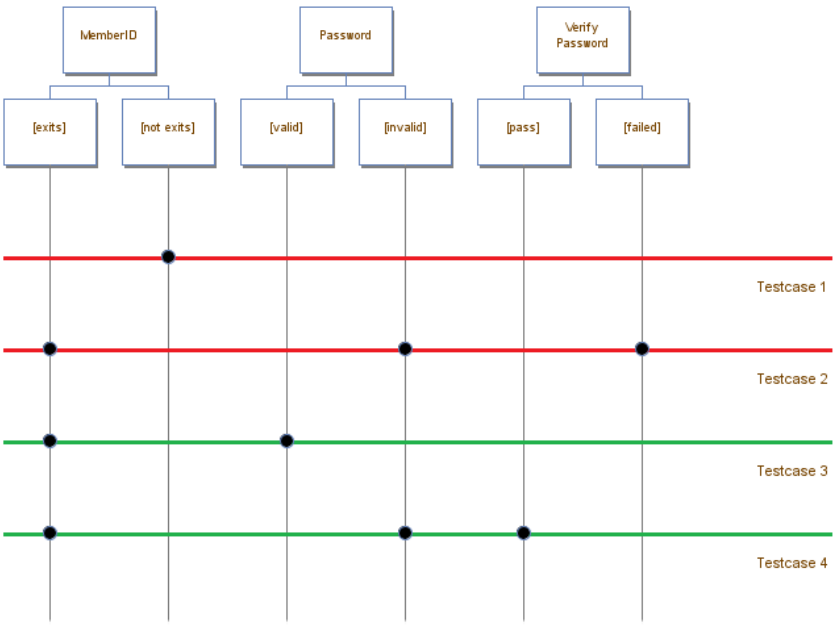
- การประเมินผลลัพธ์ แผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข

เช่นเดียวกับกรณีที่ได้กล่าวมานั้น ผู้วิจัยได้ทำการวิเคราะห์แผนภาพกิจกรรมยูเอ็มแอลที่ได้กล่าวมาด้วยวิธีการ การวิเคราะห์แบบแผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข ผลลัพธ์ดังรูปที่ 62



รูปที่ 62 CCTM จากตัวอย่างที่ 5.2.3 ที่ทำด้วยมือ

เมื่อได้ทำการใช้โปรแกรมในงานวิจัยนี้ วิเคราะห์ด้วยแผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข จะได้ผลลัพธ์ดังรูปที่ 63



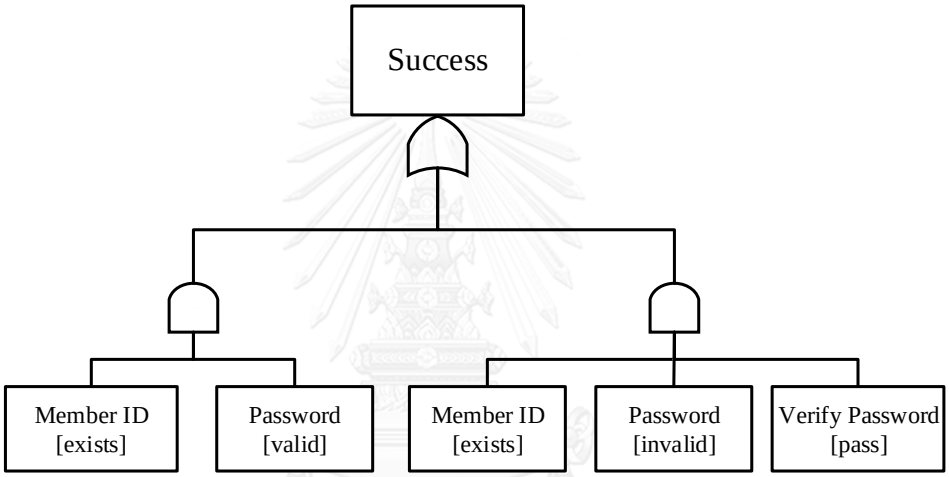
รูปที่ 63 CCTM จากตัวอย่างที่ 5.2.3 ที่สร้างด้วยระบบ

จากผลลัพธ์ที่ได้ออกมานั้นจะเห็นได้ว่า เราได้กรณีทดสอบทั้งหมด 4 กรณี โดยมี สองกรณีเป็นเส้นทางความสำเร็จ และอีกสองกรณีเป็นเส้นทางสู่ความผิดพลาด

จากการเปรียบเทียบจะพบว่า การวิเคราะห์ด้วยโปรแกรมและการวิเคราะห์ด้วยมือได้ผลลัพธ์ที่เหมือนกัน นั้นคือ ได้กรณีทดสอบและเงื่อนไขตรงกันทุกประการ

- การประเมินผลลัพธ์ แผนภาพต้นไม้ความสำเร็จ

ลำดับต่อมาเป็นการประเมินผลลัพธ์ การวิเคราะห์แบบแผนภาพต้นไม้ความสำเร็จ ในกรณีนี้ทางผู้วิจัยได้ทำการวิเคราะห์แผนภาพกิจกรรมออกมาได้ดังรูป 64

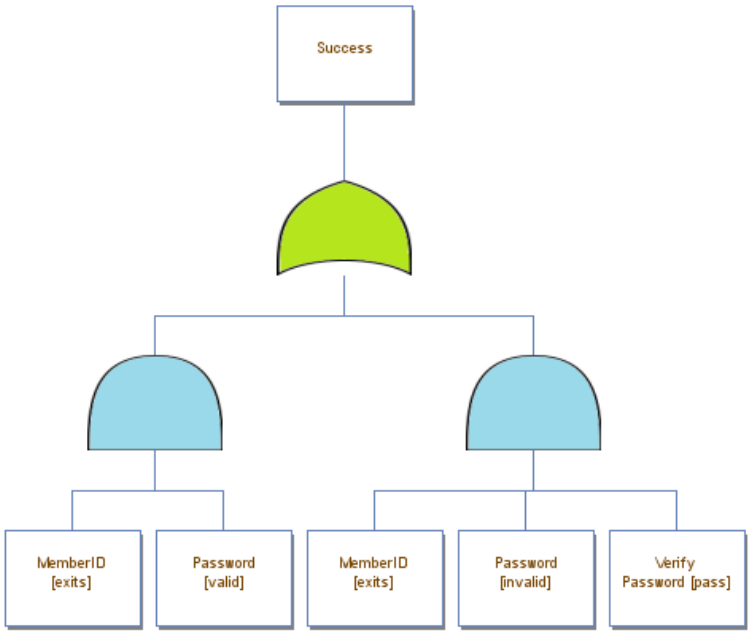


รูปที่ 64 STD จากตัวอย่างที่ 5.2.3 ที่ทำด้วยมือ

Chulalongkorn University

ในกรณีนี้จะสังเกตได้ว่า เงื่อนไขในการไปถึงเป้าหมายความสำเร็จที่ได้ตั้งไว้นั้น มีเส้นทางที่สามารถไปถึงได้ 2 ทาง จากทฤษฎีแล้วเราควรจะใช้ แบบจำลองเชิงตรรกะ ออร์เกต เข้ามาช่วยซึ่งในกรณีนี้เราจะนำออร์เกตมาวางอยู่ด้านบนสุด เพื่อเป็นเงื่อนไขก่อนไปถึงเป้าหมายความสำเร็จที่ได้ตั้งไว้ ในระดับต่อมา เราจะใช้แบบจำลองเชิงตรรกะ แอนด์เกตตามปกติ สำหรับแผนภาพต้นไม้ความสำเร็จ

เมื่อได้ทำการใช้โปรแกรมในงานวิจัยนี้ วิเคราะห์ด้วยแผนภาพจำลองต้นไม้ความสำเร็จ จะได้ผลลัพธ์ดังรูปที่ 65

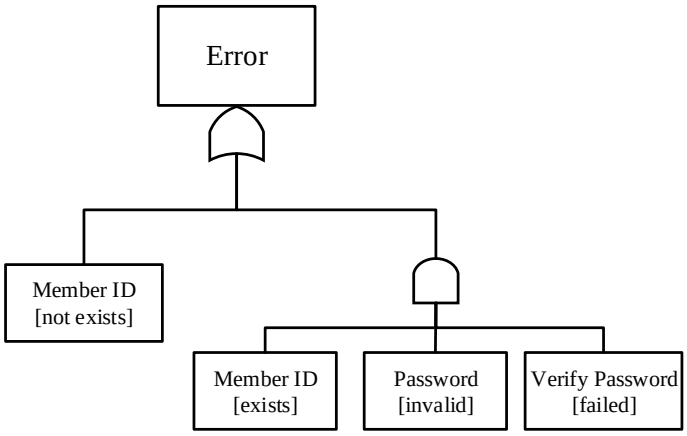


รูปที่ 65 STD จากตัวอย่างที่ 5.2.3 ที่สร้างด้วยระบบ

จากการเปรียบเทียบจะพบว่า การวิเคราะห์ด้วยโปรแกรมและการวิเคราะห์ด้วยมือได้ผลลัพธ์ที่เหมือนกัน นั้นคือเงื่อนไขและแบบจำลองเชิงตรรกะเหมือนกันทุกประการ

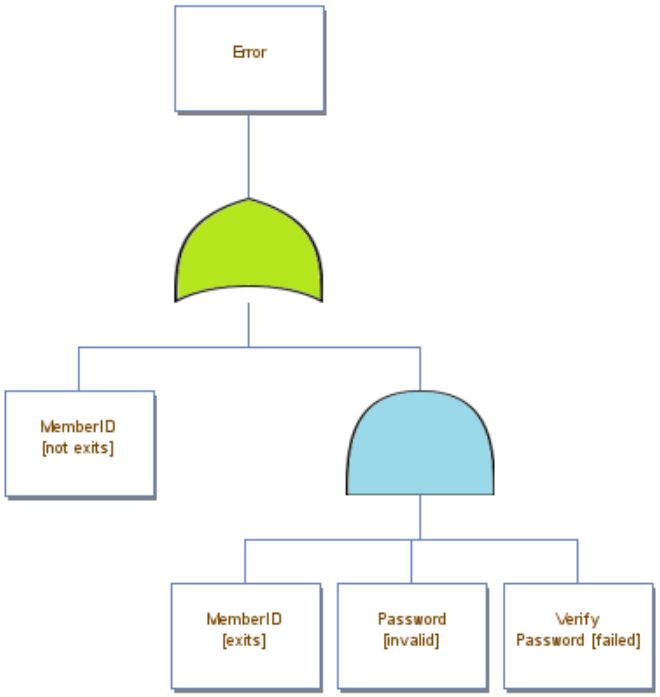
- การประเมินผลลัพธ์ แผนภาพต้นไม้ความผิดพลาด

ลำดับต่อมาเป็นการประเมินผลลัพธ์ การวิเคราะห์แบบแผนภาพต้นไม้ความผิดพลาด ในกรณีนี้ทางผู้วิจัยได้ทำการวิเคราะห์แผนภาพกิจกรรมออกมาได้ดังรูป 66



รูปที่ 66 FTD จากตัวอย่างที่ 5.2.3 ที่ทำด้วยมือ

เมื่อได้ทำการใช้โปรแกรมในงานวิจัยนี้ วิเคราะห์ด้วยแผนภาพจำลองต้นไม้ความผิดพลาด จะได้ผลลัพธ์ดังรูป 67



รูปที่ 67 FTD จากตัวอย่างที่ 5.2.3 ที่สร้างด้วยระบบ

จากการเปรียบเทียบจะพบว่า การวิเคราะห์ด้วยโปรแกรมและการวิเคราะห์ด้วยมือได้ผลลัพธ์ที่เหมือนกัน นั้นคือเงื่อนไขและแบบจำลองเชิงตรรกะเหมือนกันทุกประการ

กล่าวโดยสรุปจากกรณีทดสอบที่ 3 กรณีทดสอบที่ 3 โปรแกรมที่ประกอบด้วย แผนภาพกิจกรรมที่มี Fork node และ Join node เป็นส่วนประกอบ ซึ่งในที่นี่ใช้โปรแกรมการเข้าสู่ระบบโปรแกรมในการวิจัยนี้สามารถทำงานได้ผลลัพธ์ตรงกับการทำงานด้วยมือทุกประการ

5.2.4 กรณีทดสอบที่ 4 กรณีทดสอบที่ 4 โปรแกรมที่ประกอบด้วยความซับซ้อนที่มี จุดตัดสินใจ 10 จุด มีทางสู่เป้าหมายความสำเร็จมากกว่า 1 ทาง และมี Fork Join เป็นส่วนประกอบ

สำหรับในกรณีนี้เราจะยกตัวอย่างของโปรแกรมการตรวจสอบการเข้าสู่ระบบและการสั่งซื้อไมล์สำหรับสมาชิกแต่ละระดับ ซึ่งประกอบไปด้วยจุดตัดสินใจ 10 จุด และมี Fork กับ Join เป็นส่วนประกอบด้วย นั้นคือ การทำงานของโปรแกรมจะมีลักษณะดังนี้

1) ผู้ใช้จำเป็นต้องเข้าสู่ระบบสมาชิก
2) โปรแกรมจะทำการตรวจสอบสมาชิก
    a. หากสมาชิกมีอยู่ในระบบ จะดำเนินการขั้นต่อไป
    b. หากสมาชิกไม่มีอยู่ในระบบ โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์
3) โปรแกรมจะทำการตรวจสอบรหัสผ่าน
    a. หากรหัสผ่านถูกต้อง โปรแกรมจะดำเนินการขั้นตอนที่ 6)
    b. หากรหัสผ่านไม่ถูกต้อง โปรแกรมจะดำเนินการขั้นต่อไป
4) ระบบจะเข้าสู่ระบบการตั้งรหัสผ่านใหม่ ซึ่งการตั้งรหัสผ่านใหม่จะสามารถเป็นไปได้สองทางคือ
    a. การตั้งรหัสผ่านใหม่ด้วยคำถาม
    b. การตั้งรหัสผ่านใหม่ด้วยอีเมล
5) โปรแกรมจะทำการตรวจสอบความถูกต้องของการตั้งรหัสผ่านใหม่
    a. หากการตั้งรหัสผ่านสำเร็จ โปรแกรมจะดำเนินการขั้นต่อไป
    b. หากการตั้งรหัสผ่านไม่สำเร็จ โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์
6) โปรแกรมจะทำการตรวจสอบระดับของสมาชิก
    a. หากสมาชิกเป็นสมาชิกระดับพื้นฐาน โปรแกรมจะดำเนินการขั้นต่อไป
    b. หากสมาชิกไม่เป็นสมาชิกระดับพื้นฐาน โปรแกรมจะดำเนินการขั้นตอนที่ 8)
7) โปรแกรมจะทำการตรวจสอบอายุของสมาชิก
    a. หากสมาชิกมีอายุการเป็นสมาชิกมากกว่า เดือน โปรแกรมจะดำเนินการขั้น 6 ต่อไป
    b. หากสมาชิกมีอายุการเป็นสมาชิกน้อยกว่า เดือน โปรแกรมจะจบการทำงาน 6 ด้วยการปฏิเสธการสั่งซื้อไมล์
8) โปรแกรมจะเข้าสู่ระบบการเลือกจำนวนไมล์ที่ต้องการสั่งซื้อ
    a. หากการซื้อไม่เกินข้อจำกัด โปรแกรมจะเดินการขั้นต่อไป
    b. หากการซื้อเกินข้อจำกัด โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์
9) ระบบจะเข้าสู่ชำระเงิน
    a. หากบัตรเครดิตผ่านการทวนสอบ โปรแกรมจะเดินการขั้นต่อไป

b. หากบัตรเครดิตไม่ผ่านการทวนสอบ โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์

10) ระบบชำระเงินจะทำการยืนยันการชำระเงิน

a. หากการชำระเงินสำเร็จ โปรแกรมจะเดินการขั้นต่อไป

b. หากการชำระเงินไม่สำเร็จ โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์

11) การบันทึกข้อมูลลงฐานข้อมูล

a. หากการบันทึกสำเร็จ โปรแกรมจบการทำงานด้วยการยอมรับการสั่งซื้อไมล์

b. หากการบันทึกไม่สำเร็จ โปรแกรมจะเดินการขั้นต่อไป

12) โปรแกรมจะใส่ข้อมูลอยู่ในระบบรอการอนุมัติ

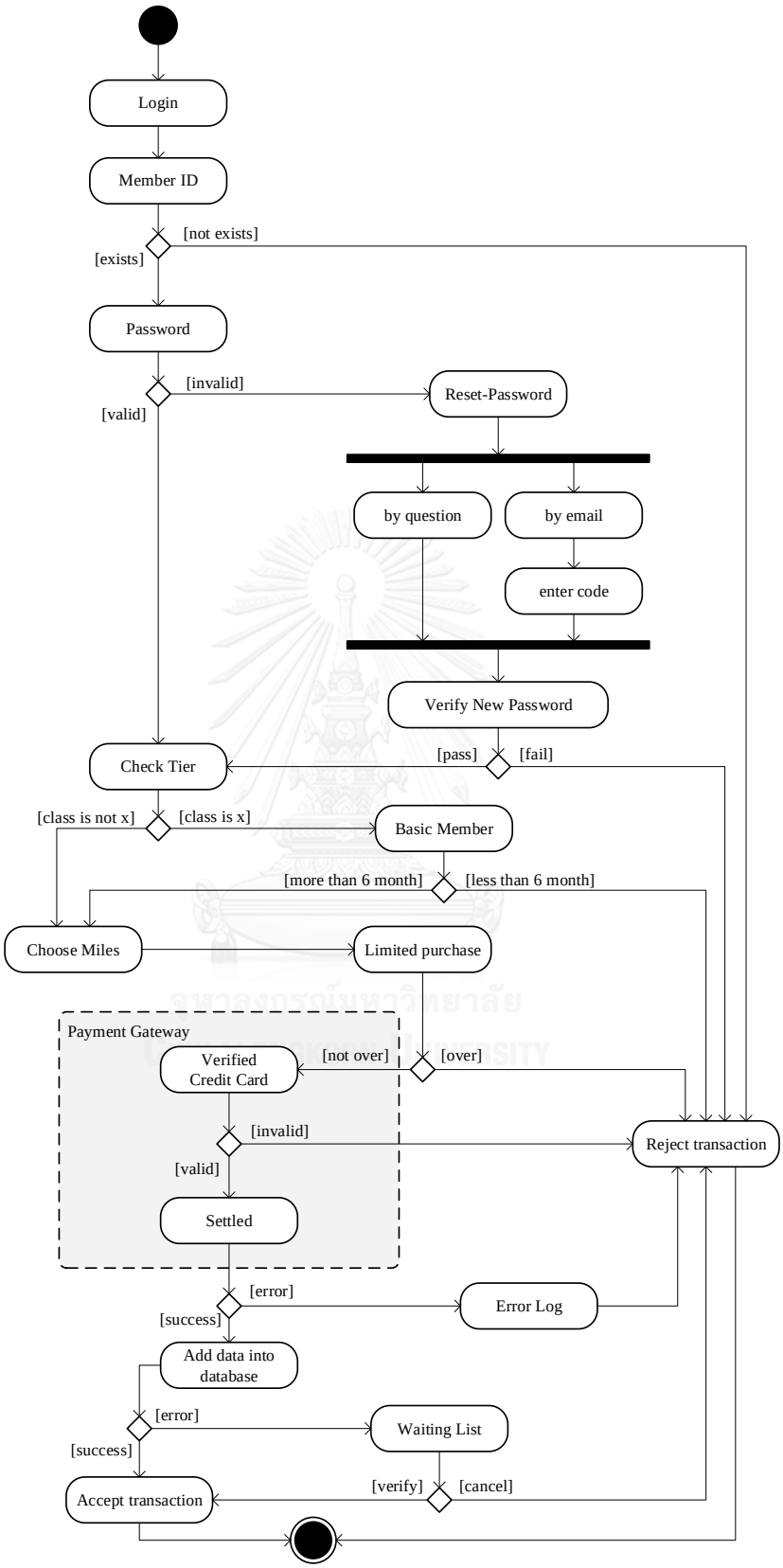
a. หากการได้รับการอนุมัติ โปรแกรมจบการทำงานด้วยการยอมรับการสั่งซื้อไมล์

b. หากการไม่ได้รับการอนุมัติ โปรแกรมจะจบการทำงานด้วยการปฏิเสธการสั่งซื้อไมล์

- แผนภาพกิจกรรมที่ต้องการทดสอบ

จากการทำงานของโปรแกรมการเข้าสู่ระบบที่ได้กล่าวไปแล้วนั้น เราสามารถนำมาสร้างเป็นแผนภาพกิจกรรมยูเอ็มแอลได้ดังนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 68 แผนภาพกิจกรรมการซื้อไมล์สะสมแบบซับซ้อน

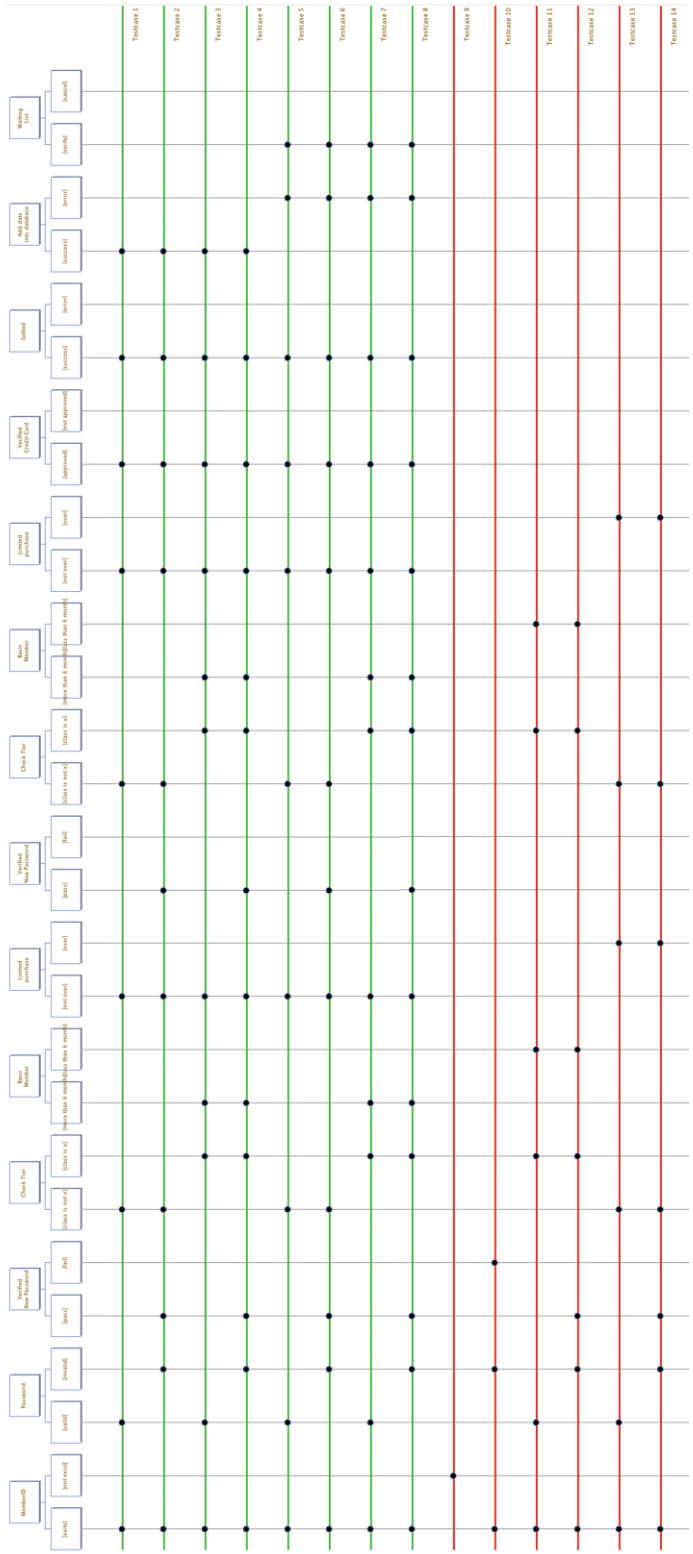
- ข้อมูลนำเข้า

ข้อมูลแผนภาพกิจกรรมดังกล่าวจำเป็นต้องแปลงให้เป็น ยูเอ็มแอลเมทาเดตา ซึ่งจะอยู่ในรูปแบบของ ยูเอ็มแอล (UML) ซึ่งทางผู้วิจัยจะใช้โปรแกรมปาปิรุสในการแปลง โดยที่โปรแกรมนี้จะสร้าง เอ็กซ์เอมไอ เมทาเดตา ซึ่งตามมาตรฐานของโอเอ็มจี จะได้ผลลัพธ์ออกมาดังรูป

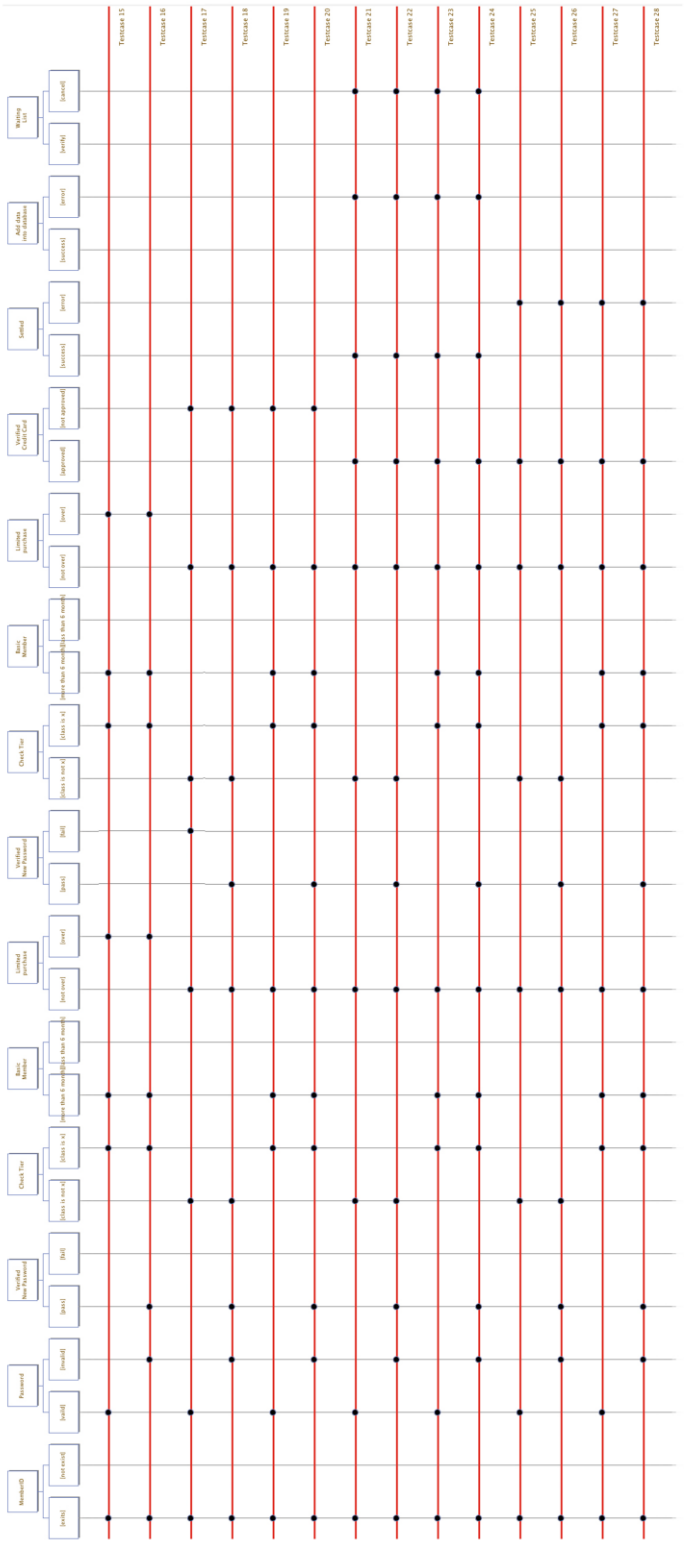
รูปที่ 69 เมทาเดตาของแผนภาพกิจกรรมการซื้อไมล์สะสมแบบซับซ้อน

- การประเมินผลลัพธ์ แผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข

เมื่อได้ทำการใช้โปรแกรมในงานวิจัยนี้ วิเคราะห์ด้วยแผนภาพจำลองต้นไม้การจำแนกแบบมีเงื่อนไข จะได้ผลลัพธ์รูปที่ 70 และ 71



รูปที่ 70 CCTM จากตัวอย่างที่ 5.2.4 ที่สร้างด้วยระบบ



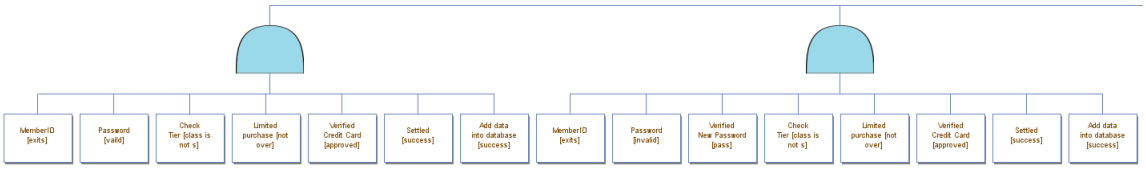
รูปที่ 71 CCTM จากตัวอย่างที่ 5.2.4 ที่สร้างด้วยระบบ (ต่อ)

จากผลลัพธ์ที่ได้ออกมานั้นจะเห็นได้ว่า เราได้กรณีทดสอบทั้งหมด 28 กรณี โดยมี 8 กรณี เป็นเส้นทางความสำเร็จ และอีก 20 กรณีเป็นเส้นทางสู่ความผิดพลาด

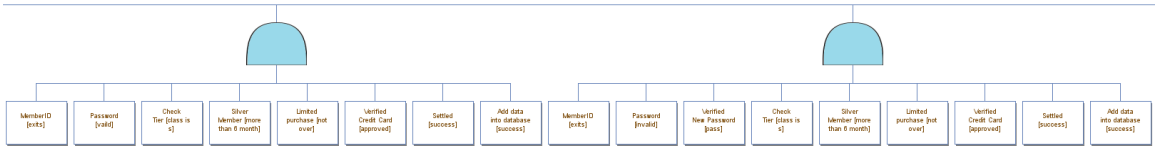
จากการเปรียบเทียบจะพบว่า การวิเคราะห์ด้วยโปรแกรมและการวิเคราะห์ด้วยมือได้ผลลัพธ์ที่เหมือนกัน นั้นคือ ได้กรณีทดสอบและเงื่อนไขตรงกันทุกประการ

- การประเมินผลลัพธ์ แผนภาพต้นไม้ความสำเร็จ

ลำดับต่อมาเป็นการประเมินผลลัพธ์ การวิเคราะห์แบบแผนภาพต้นไม้ความสำเร็จ ในกรณีนี้ทางผู้วิจัยได้ใช้โปรแกรมในงานวิจัยนี้ทำการสร้างแผนภาพต้นไม้ความสำเร็จ และเนื่องจากกรณีทดสอบและเงื่อนไขมีจำนวนมาก ทางผู้วิจัยจึงขอนำเสนอผลลัพธ์ในรูปที่มีลักษณะเชื่อมต่อถึงกันดังรูปที่ 72 – 75

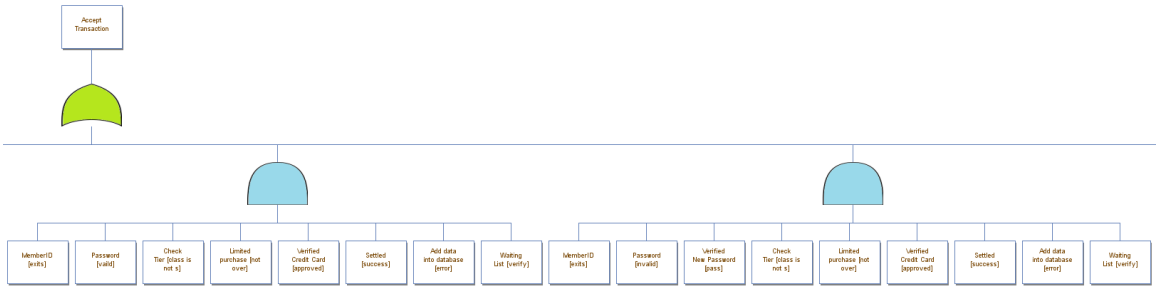


รูปที่ 72 Mileage Purchase STD 1

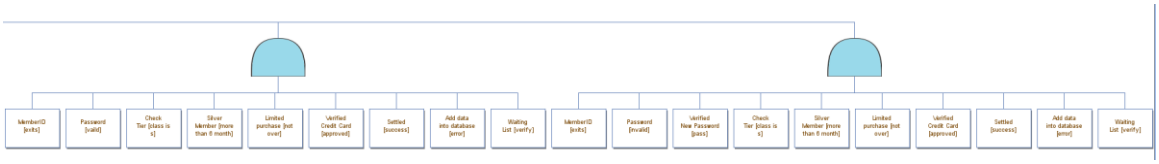


Chulalongkorn University

รูปที่ 73 Mileage Purchase STD 2



รูปที่ 74 Mileage Purchase STD 3

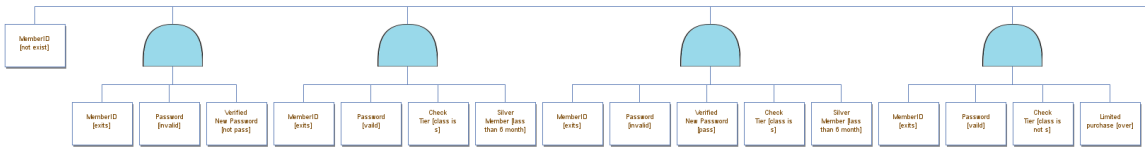


รูปที่ 75 Mileage Purchase STD 4

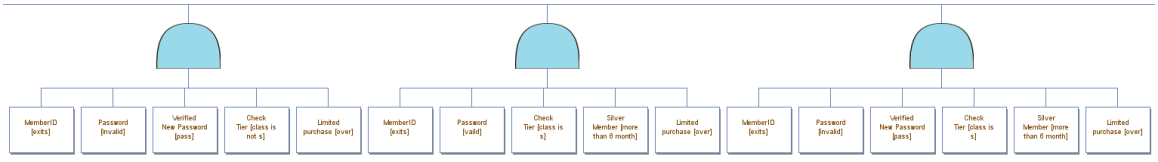
จากผลลัพธ์ที่ได้ จะเห็นว่าได้กรณีทดสอบสำหรับแผนภาพต้นไม้ความสำเร็จทั้งหมด 8 กรณี และมีเงื่อนไขตรงตามที่ได้จาก แผนภาพต้นไม้การจำแนก

- การประเมินผลลัพธ์ แผนภาพต้นไม้ความผิดพลาด

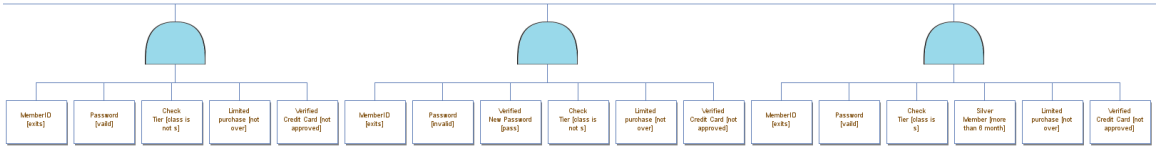
ลำดับต่อมาเป็นการประเมินผลลัพธ์ การวิเคราะห์แบบแผนภาพต้นไม้ความผิดพลาด ในกรณีนี้ทางผู้วิจัยได้ใช้โปรแกรมในงานวิจัยนี้ทำการสร้างแผนภาพต้นไม้ความผิดพลาด และเนื่องจาก กรณีทดสอบและเงื่อนไขมีจำนวนมาก ทางผู้วิจัยจึงขอนำเสนอผลลัพธ์ในรูปที่มีลักษะณะ เชื่อมต่อถึงกันดังรูปที่ 76 – 83



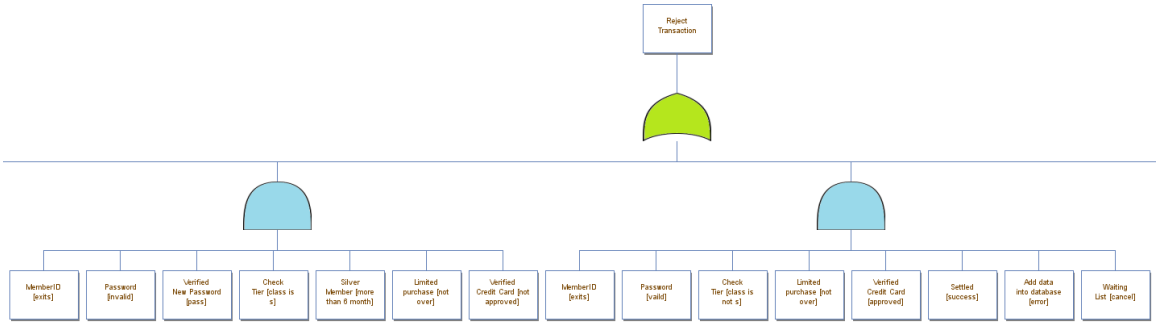
รูปที่ 76 Mileage Purchase FTD 1



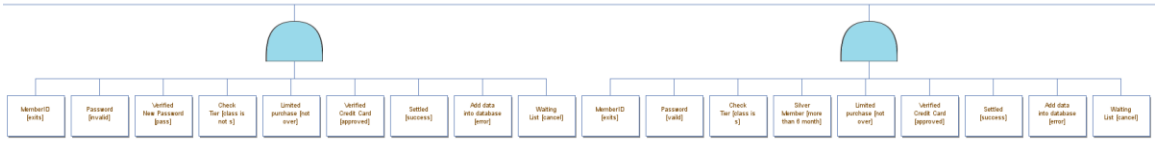
รูปที่ 77 Mileage Purchase FTD 2



รูปที่ 78 Mileage Purchase FTD 3

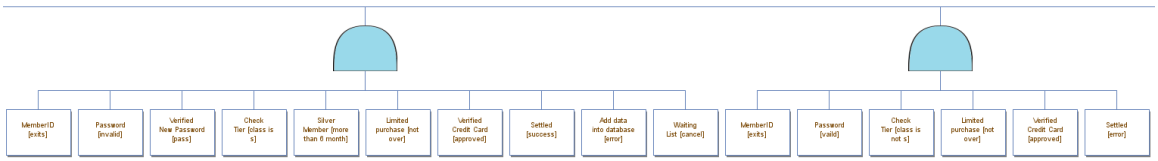


รูปที่ 79 Mileage Purchase FTD 4

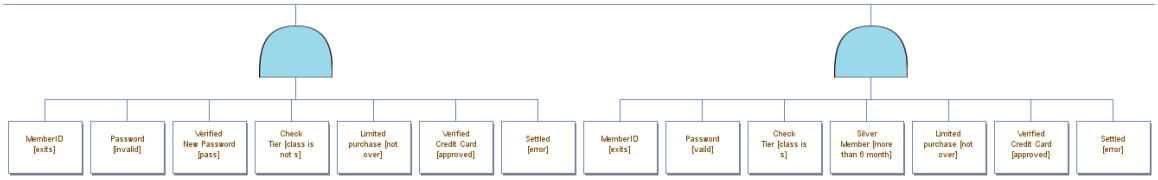


จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

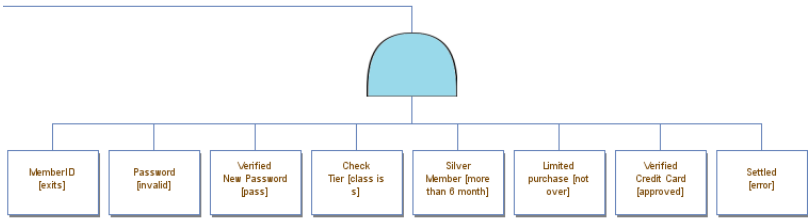
รูปที่ 80 Mileage Purchase FTD 5



รูปที่ 81 Mileage Purchase FTD 6



รูปที่ 82 Mileage Purchase FTD 7



รูปที่ 83 Mileage Purchase FTD 8

จากผลลัพธ์ที่ได้ จะเห็นว่าได้กรณีทดสอบสำหรับแผนภาพต้นไม้ความผิดพลาดทั้งหมด 20 กรณี และมีเงื่อนไขตรงตามที่ได้จากแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

ผลลัพธ์โดยสรุปที่ได้จากงานวิจัยฉบับนี้ มีดังนี้

1. ได้แนวคิดและวิธีการสร้างแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข แผนภาพต้นไม้ความสำเร็จ และแผนภาพต้นไม้ความผิดพลาด
2. ได้แอปพลิเคชันการสร้างแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข แผนภาพต้นไม้ความสำเร็จ และแผนภาพต้นไม้ความผิดพลาด
3. ได้เครื่องมือสนับสนุนกรณีทดสอบกระบวนการทำงานของซอฟต์แวร์ ซึ่งช่วยลดการใช้ทรัพยากรและเวลาในการออกแบบและพัฒนาซอฟต์แวร์
4. ได้เครื่องมือสนับสนุนกรณีทดสอบกระบวนการทำงานของซอฟต์แวร์ ซึ่งช่วยลดข้อผิดพลาดในขั้นตอนการออกแบบและพัฒนาซอฟต์แวร์

6.2 ข้อจำกัด

- ไฟล์เอกสารเอกซ์เอ็มไอที่นำเข้าในงานวิจัยฉบับนี้รองรับเฉพาะที่สร้างขึ้นมาจากโปรแกรมเสริมอีคลิปโมเดลลิ่งเฟรมเวิร์คค ปาปิรุส มาร์ส เท่านั้น
- ข้อมูลนำเข้าในงานวิจัยฉบับนี้สนับสนุนรองรับเฉพาะภาษาอังกฤษเท่านั้น
- ข้อมูลนำเข้าจะสนับสนุนเฉพาะรูปแบบของจุดตัดสินใจของแผนภาพกิจกรรม
- ผู้ใช้ระบบจำเป็นต้องเลือกเป้าหมายที่ถูกต้องเพื่อผลลัพธ์การทำงานได้ถูกต้องตามที่ต้องการ

6.3 แนวทางการวิจัยต่อ

- รองรับส่วนประกอบแผนภาพกิจกรรมให้ครบถ้วนทุกประเภทและมีความสมบูรณ์มากขึ้น
- ปรับปรุงส่วนแสดงผลแผนภาพต้นไม้ให้รองรับกรณีทดสอบจำนวนมาก
- พัฒนาต่อยอดการแปลงผลลัพธ์ที่ได้จากงานวิจัยนี้ให้อยู่ในรูปแบบเอกซ์เอ็มแอล เพื่อนำไปใช้ในแอปพลิเคชันอื่นๆ ต่อไปได้

รายการอ้างอิง

[1] Istqbexamcertification.com. (2016). *What is reliability testing in software?* Available: http://istqbexamcertification.com/what-is-reliability-testing-in-software/

[2] M. Grochtmann and K. Grimm, "Classification trees for partition testing," *Software Testing, Verification and Reliability,* vol. 3, pp. 63-82, 1993.

[3] C. Ericson, "Fault Tree Analysis - A History," in *The Proceedings of The 17th International System Safety Conference*, 1999, pp. 1-9.

[4] G. Helmer, J. Wong, M. Slagell, V. Honavar, L. Miller, and R. Lutz, "A Software Fault Tree Approach to Requirements Analysis of an Intrusion Detection System," *Requirements Engineering,* vol. 7, pp. 207-220, 2002.

[5] K. M. Hansen, A. P. Ravn, and V. Stavridou, "From safety analysis to software requirements," *IEEE Transactions on Software Engineering,* vol. 24, pp. 573-584, 1998.

[6] W3schools.com. (2016). *XML Introduction*. Available: http://www.w3schools.com/xml/xml_whatis.asp

[7] W3C. (2016). *World Wide Web Consortium (W3C)*. Available: https://www.w3.org/

[8] M. Rouse. (2005). *What is XMI (XML Metadata Interchange)?* Available: http://searchsoa.techtarget.com/definition/XMI

[9] D. Steinberg and e. al., *EMF Eclipse Modeling Framework*: Addison-Wesley Professional, 2008.

[10] Eclipse.org. (2016). *Papyrus*. Available: http://www.eclipse.org/papyrus/

[11] "OMG Unified Modeling Language (OMG UML)," in *Superstructure Version 2.3*, ed: Object Management Group
2011.

[12] D. Tegarden, A. Dennis, and B. H. Wixom, *Systems Analysis and Design with UML*. Hoboken, N.J.: Wiley, 2013.

[13] GitHub. (2016). *JGraphX*. Available: https://jgraph.github.io/mxgraph/docs/manual_javavis.html

[14] S. Kansomkeat, P. Thiket, and J. Offutt, "Generating test cases from UML activity diagrams using the Condition-Classification Tree Method," in *Software Technology and Engineering (ICSTE), 2010 2nd International Conference on*, 2010, pp. V1-62-V1-66.

[15] S. Tiwari and A. Gupta, "An Approach to Generate Safety Validation Test Cases from UML Activity Diagram," in *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, 2013, pp. 189-198.

[16] J. Atwood. (2008). *Understanding Model-View-Controller*. Available: https://blog.codinghorror.com/understanding-model-view-controller/

[17] tutorialspoint.com. (2016). *MVC Framework - Introduction*. Available: http://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ก.

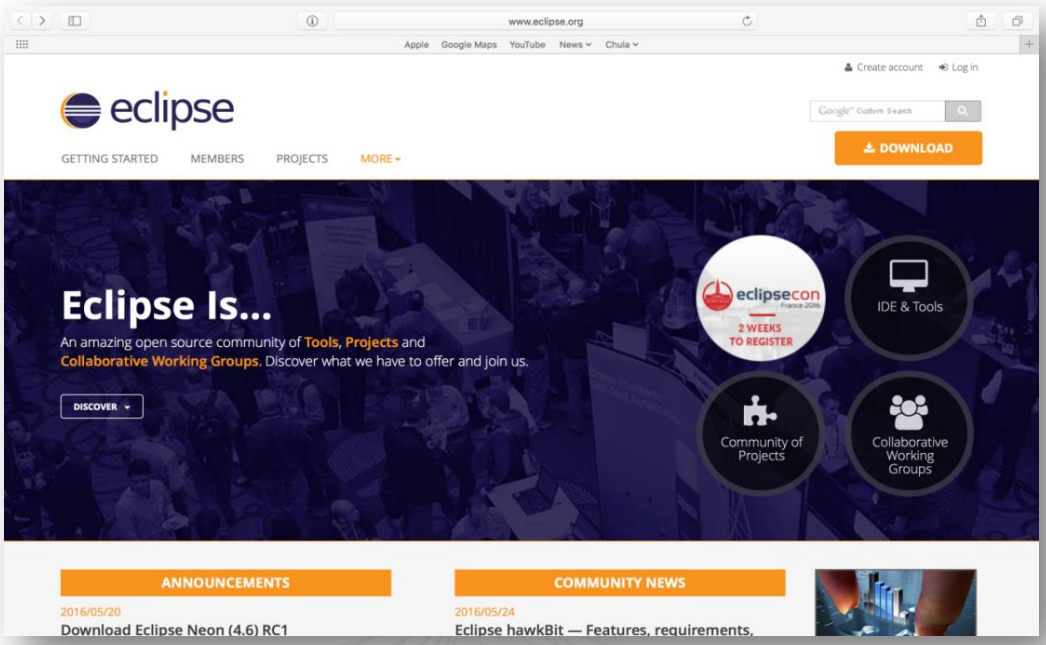
ติดตั้งเครื่องมืออีคลิปส์โมเดลลิ่งทูล

รูปที่ 84 โปรแกรมอีคลิปส์โมเดลลิ่งทูล เวอร์ชันมาร์ส 1 (4.5.1)

โปรแกรมอีคลิปส์โมเดลลิ่งทูลเวอร์ชัน 1 รุ่น 4.5.1 ดังรูปที่ 81 เป็นเครื่องมือที่ใช้สำหรับติดตั้งปลั๊กอินหรือโปรแกรมเสริมสำหรับการสร้างแผนภาพยูเอ็มแอล

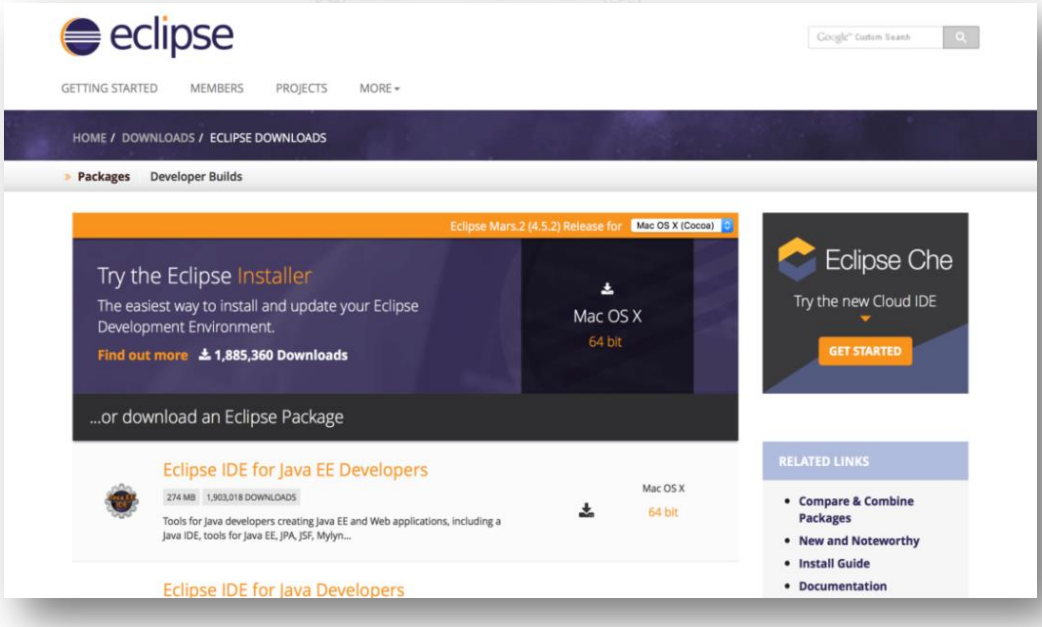
ขั้นตอนการติดตั้งโปรแกรมอีคลิปส์โมเดลลิ่งทูล มีรายละเอียดดังนี้

1. เริ่มขั้นตอนการดาวน์โหลดโปรแกรมอีคลิปส์โมเดลลิ่งทูล โดยสามารถเข้าไปดาวน์โหลดได้ที่ https://eclipse.org/ ดังรูปที่ 85 แล้วคลิกที่ “Download”



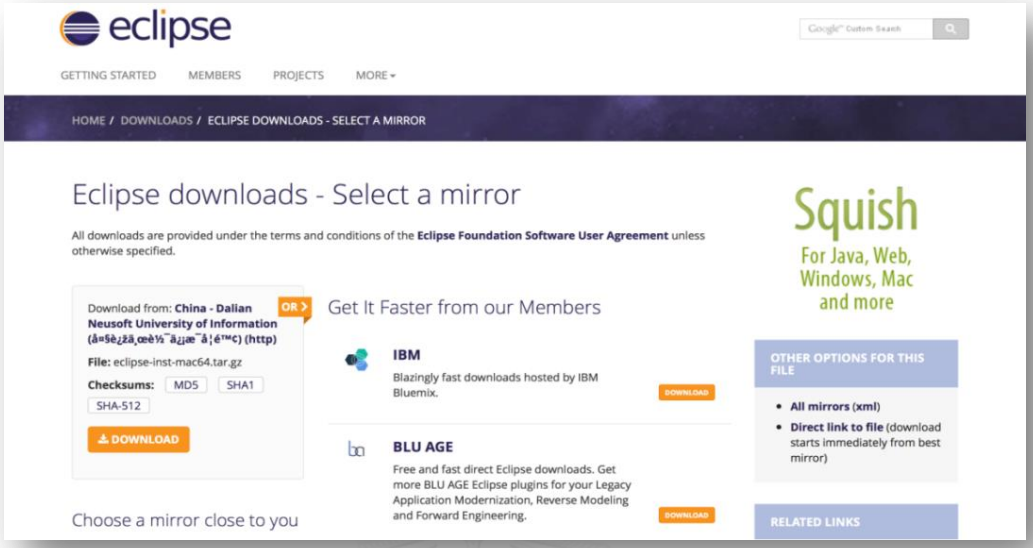
รูปที่ 85 หน้าจอดาวน์โหลดโปรแกรมอีคลิปส์โมเดลลิ่งทูล

2. เลือกระบบปฏิบัติการที่ใช้งาน ในกรณีนี้เลือก “Mac OS X” ดังรูปที่ 86



รูปที่ 86 หน้าจอ Developer Builds

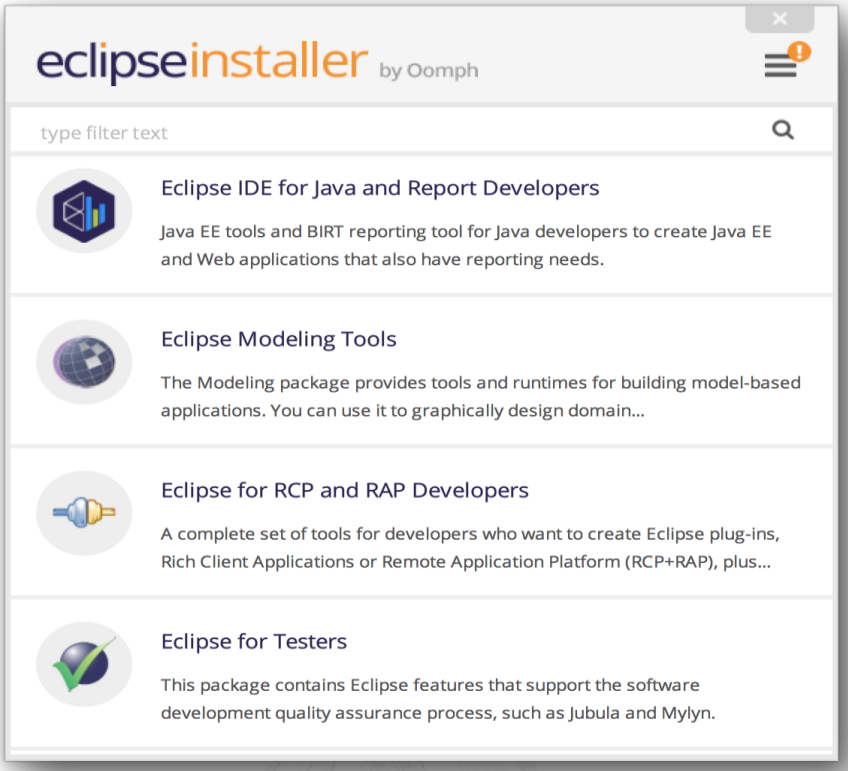
3. เลือกแหล่ง Select a mirror ที่จะทำการดาวน์โหลด ดังรูปที่ 87



รูปที่ 87 หน้าจอ Select a mirror

4. เมื่อดาวน์โหลดแล้วจะได้ไฟล์ “eclipse-inst-mac64.tar” และเมื่อแตกไฟล์ออกมาจะได้ Eclipse Installer ดังรูปที่ 88 แล้วเลือกคลิกที่ “Eclipse Modeling Tools”

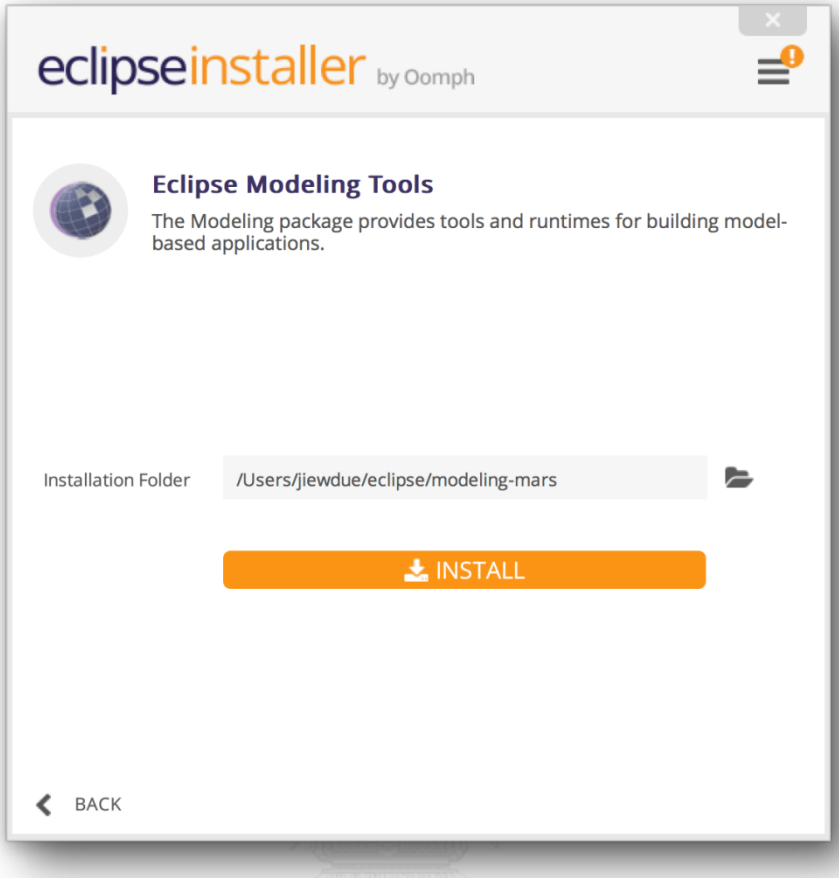
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 88 หน้าจอ Eclipse Installer

5. กด Install ดังรูปที่ 89 จึงเป็นอันเสร็จสิ้นการติดตั้งโปรแกรมอีคลิปส์โมเดลลิ่งทูล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

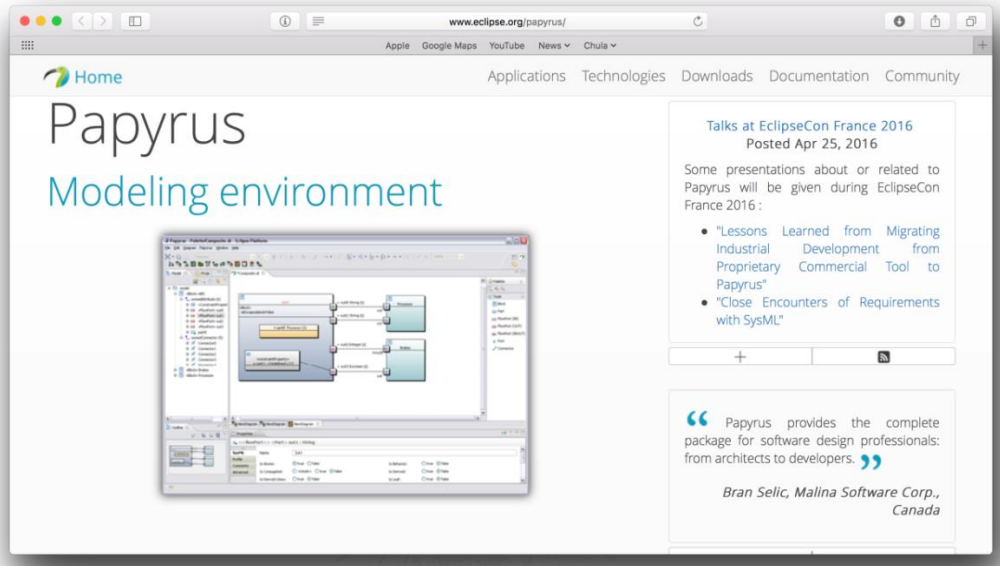


รูปที่ 89 หน้าจอติดตั้งอีคลิปส์โมเดลลิ่งทูล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ข.

ติดตั้งเครื่องมือสนับสนุนการสร้างแผนภาพยูเอ็มแอลบนอีคลิปส์



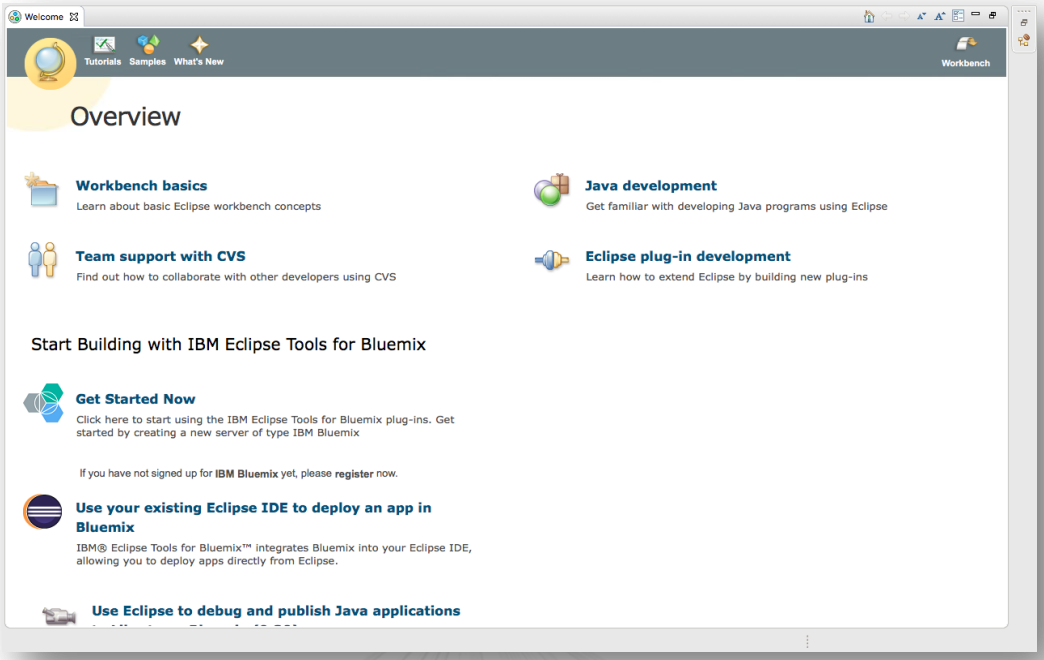
รูปที่ 90 หน้าเว็บไซต์ปลั๊กอินเสริมปาปิรุส

เครื่องมือโอเพนซอร์สสำหรับสร้างเมทาโมเดลหรือแผนภาพยูเอ็มแอลเป็นปลั๊กอินหรือโปรแกรมเสริมอีคลิปส์ พัฒนาระบบภายใต้แนวคิดอีคลิปส์โมเดลลิ่งเฟรมเวิร์ค (Eclipse Modeling Framework : EMF) รองรับมาตรฐานยูเอ็มแอล 2 (UML 2) ซึ่งถูกนิยามโดย OMG โดยคุณลักษณะจะประกอบด้วย

1. สามารถสร้างแผนภาพยูเอ็มแอลหรือ UML metamodel เพื่อสนันสนุนรูปแบบการพัฒนาระบบโดยใช้โมเดลเป็นพื้นฐาน
2. ใช้ XMI ในการเป็นมาตรฐานกลางในการสื่อสารแลกเปลี่ยนข้อมูลโมเดล
3. สร้างแผนภาพโมเดลตามมาตรฐานที่ได้กำหนดไว้

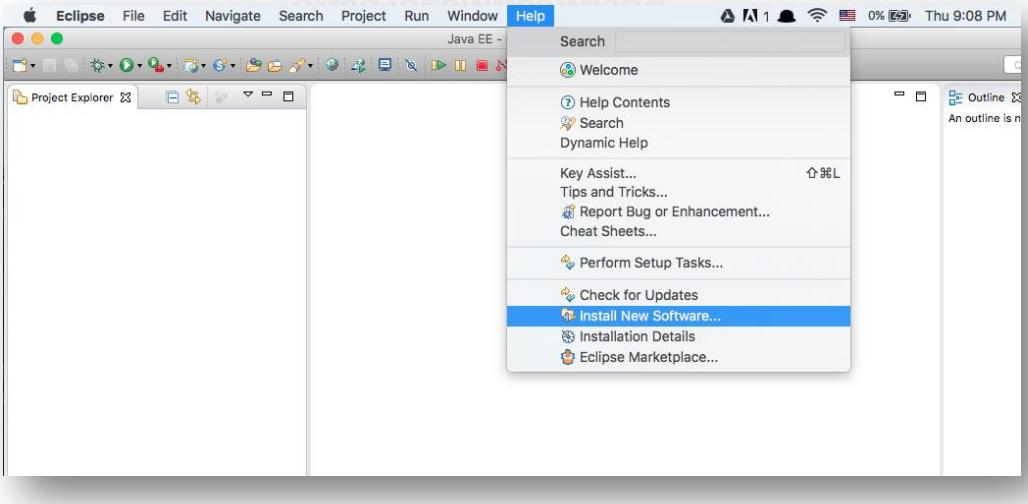
ขั้นตอนการติดตั้ง Papyrus Mars Plugin สำหรับ Eclipse UML Modeling Tools 4.5.1

1. เปิดโปรแกรมอีคลิปส์โมเดลลิ่งทูลที่ได้ทำการติดตั้งเรียบร้อยแล้ว (ภาคผนวก ก.) ดังรูปที่ 91



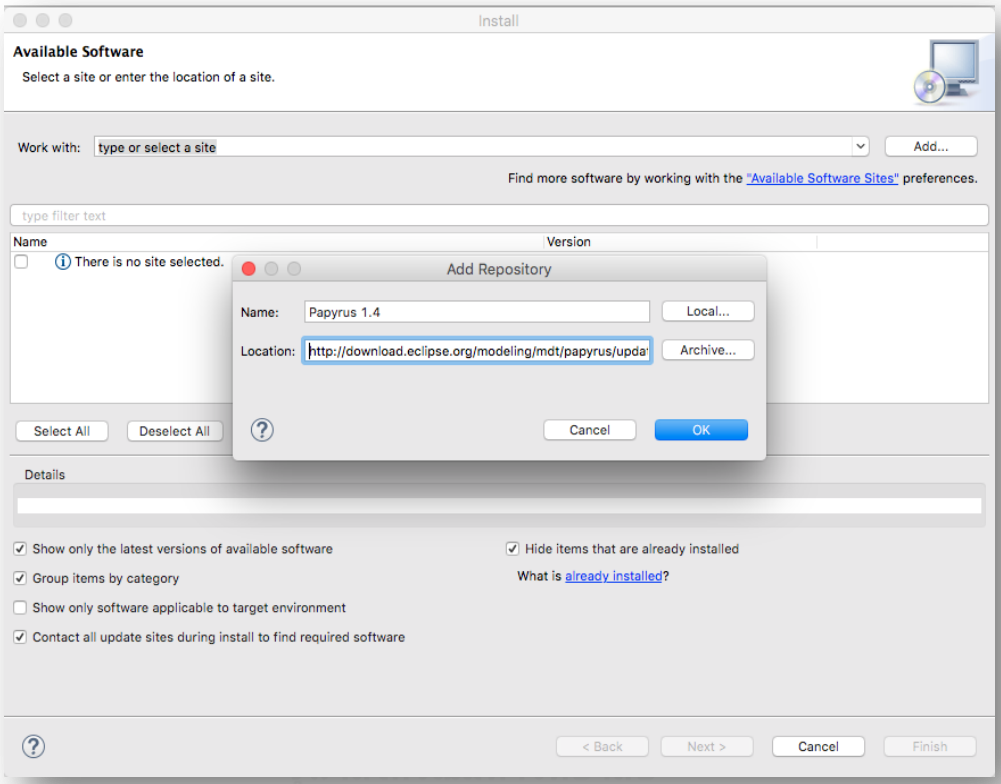
รูปที่ 91 หน้าจอ Initial Welcome Page of Eclipse

2. เริ่มขั้นตอนการติดตั้งปาปิรุส มาร์ส หลังจากเปิดโปรแกรม ไปที่เมนูบาร์ “Help” แล้วเลือก “Install New Software”



รูปที่ 92 หน้าจอติดตั้งปลั๊กอินเสริมปาปิรุส

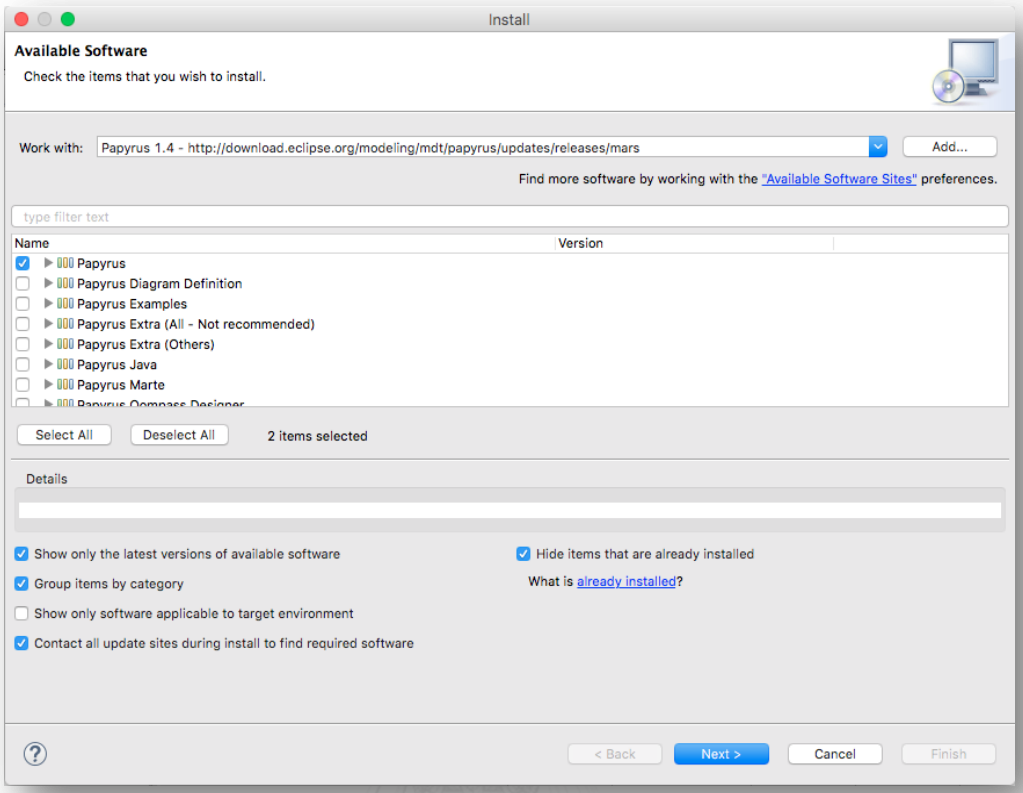
3. คลิกที่ปุ่ม “Add...” แล้วใส่เว็บไซต์ปาปิรุส มาร์ส Papyrus 1.1.x update site: ดังรูปที่ 93 ใส่ชื่อสำหรับปลั๊กอินที่ใช้ในงานวิจัย ส่วนข้อมูลในช่อง Location ใช้ใส่ไดเรคทอรี่ลิงค์สำหรับโหลดไฟล์ปลั๊กอินเสริม โดยใส่ URL: http://download.eclipse.org/modeling/mdt/papyrus/updates/releases/mars/



CHULALONGKORN UNIVERSITY

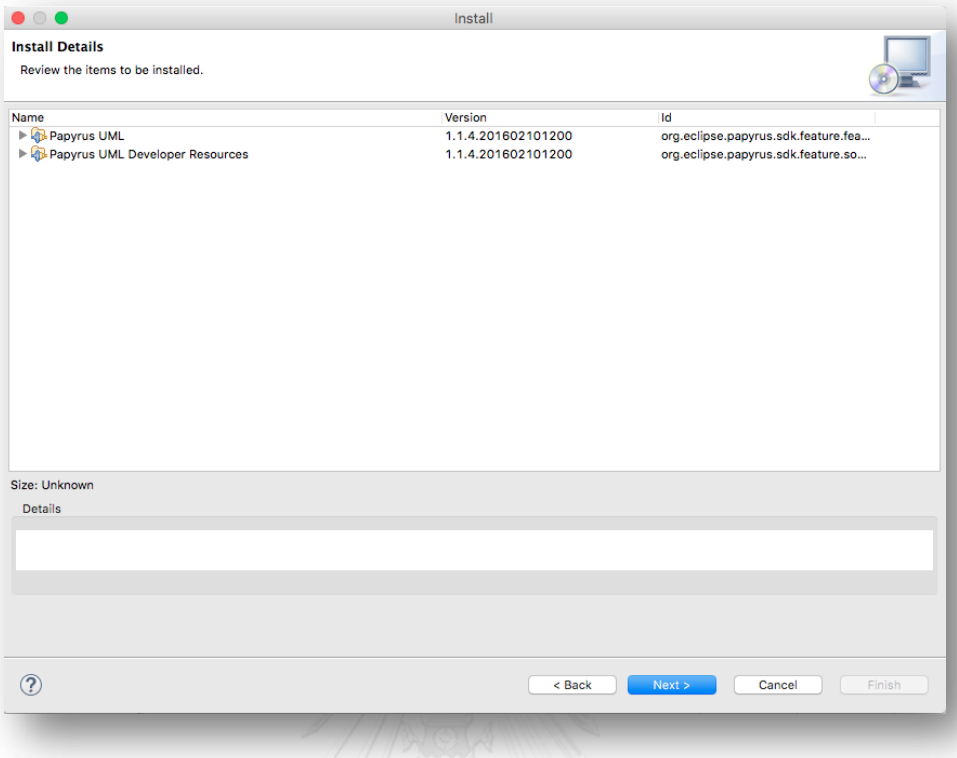
รูปที่ 93 หน้าจอดาวน์โหลดปลั๊กอินเสริมอีคลิปส์ ปาปิรุส มาร์ส

4. เลือกซอฟต์แวร์ที่จะทำการติดตั้ง จากนั้นคลิก “Next” ดังรูปที่ 94



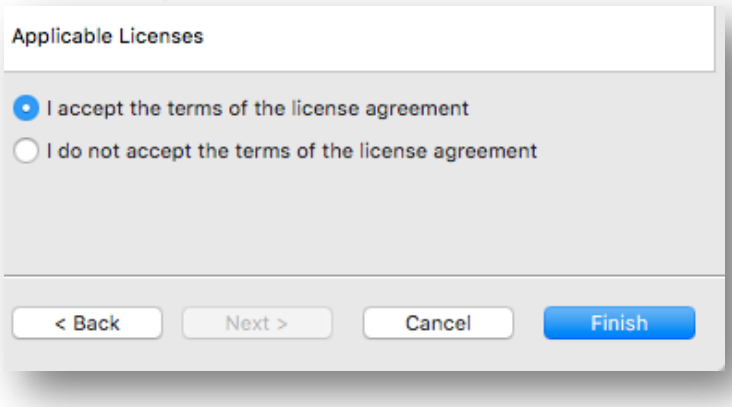
รูปที่ 94 หน้าจอติดตั้งซอฟต์แวร์ Available Software

5. ในหน้าจอ Install Details ดังรูปที่ 95 จะเป็นรายละเอียดของเวอร์ชันที่จะทำการติดตั้ง เมื่อทำการตรวจสอบข้อมูลเรียบร้อยแล้วคลิก “Next”



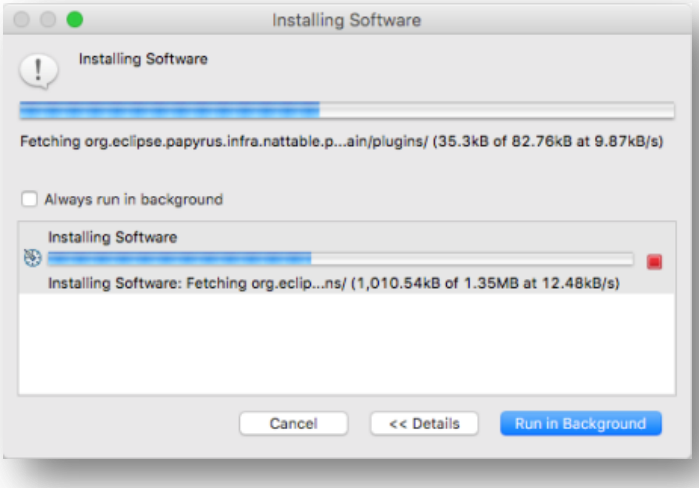
รูปที่ 95 หน้าจอรายละเอียดซอฟต์แวร์ Install Details

6. หน้ายอมรับการเงื่อนไขการใช้งาน โดยเลือกที่ “I accept the terms of the license agreement” และคลิก “Finish”



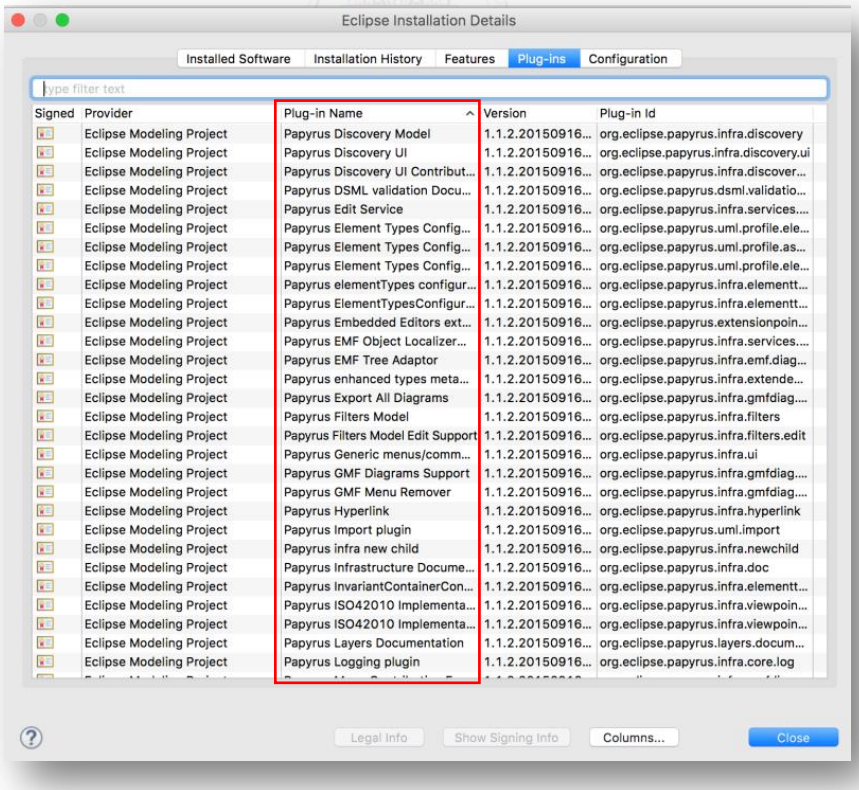
รูปที่ 96 หน้าจอ Review Licenses

7. จากนั้นโปรแกรมจะทำการติดตั้งซอฟต์แวร์ดังรูปที่ 97 เมื่อรอติดตั้งเสร็จจึงเป็นอันเสร็จสิ้นการติดตั้งปลั๊กอินเสริมปาปิรุส



รูปที่ 97 หน้าจอ Installing Software

8. เมื่อทำการติดตั้งปลั๊กอินเสริมเสร็จเรียบร้อยจะสามารถสร้างเมทาโมเดลแผนภาพกิจกรรมได้ โดยผลการติดตั้งปลั๊กอิน Papyrus mars จะแสดงดังรูปที่ 98

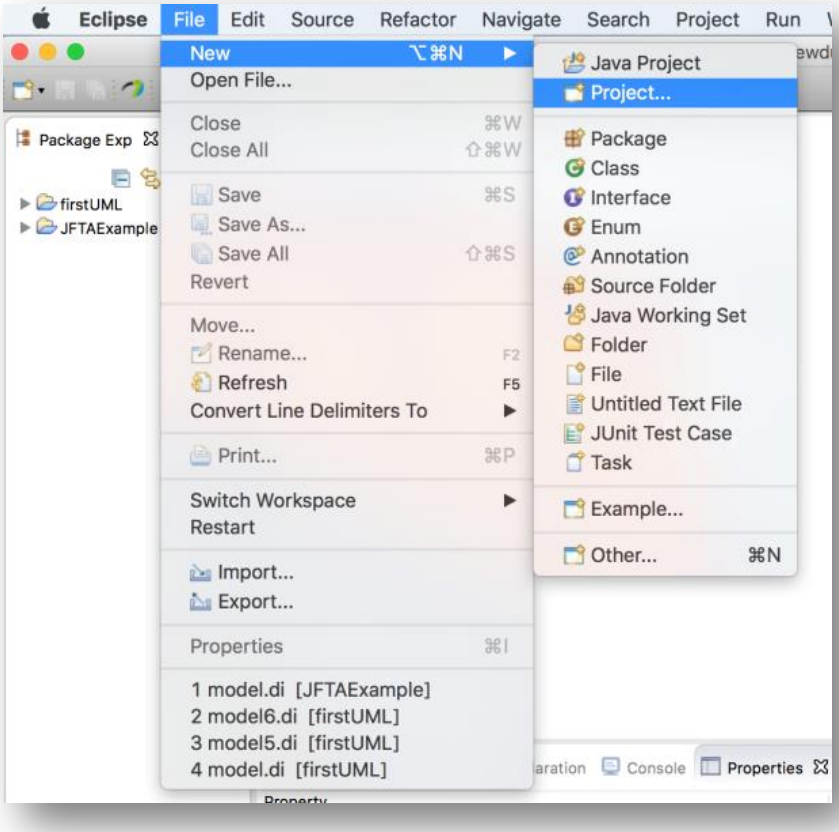


รูปที่ 98 หน้าจอแสดงปลั๊กอินเสริมที่ถูกติดตั้ง

ภาคผนวก ค.

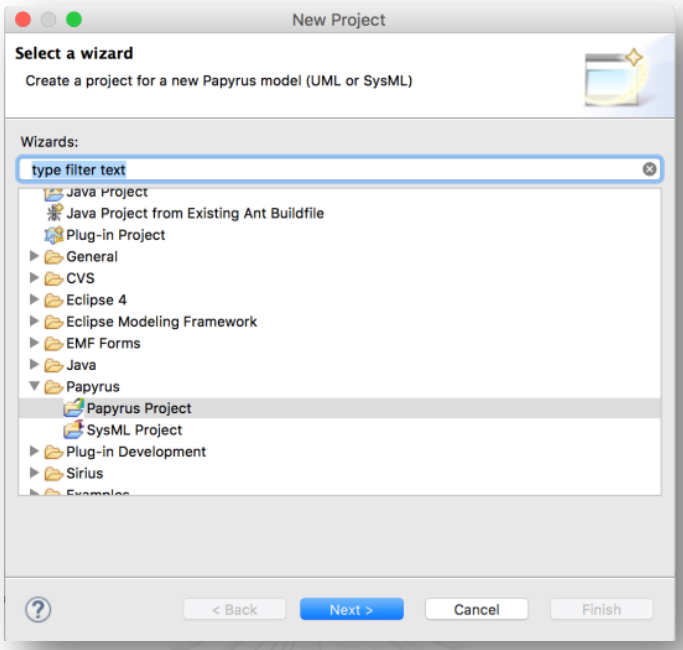
การใช้งานปลั๊กอินเสริมปาปิรุส มาร์ส

1. เมื่อทำงานติดตั้งโปรแกรมเสริม (รายละเอียด ภาคผนวก ข.) เรียบร้อยแล้ว ขั้นตอนต่อมาจะเริ่มทำการสร้างแผนภาพกิจกรรมยูเอ็มแอลหรือเมทาโมเดล โดยเลือกที่เมนู “File” เลือกที่ “New” และเลือก “Project...”



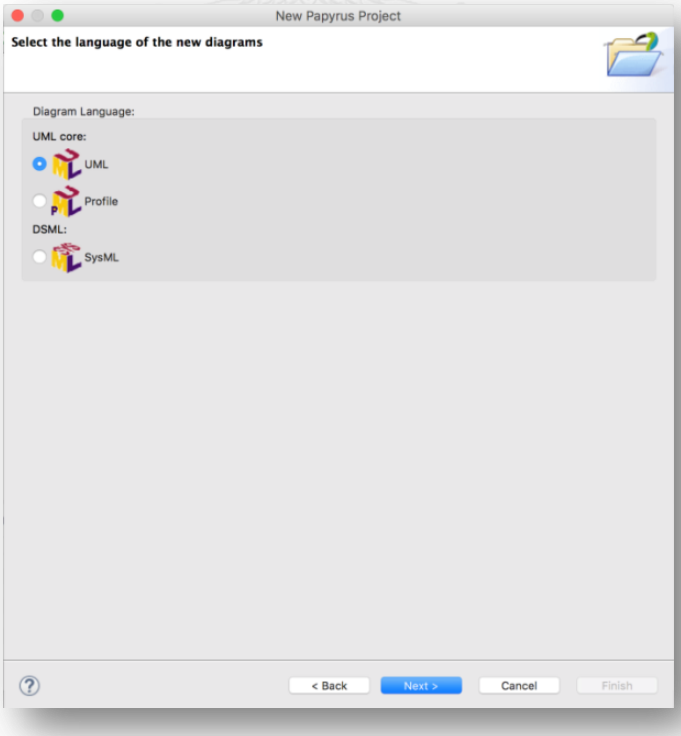
รูปที่ 99 สร้างโปรเจคแผนภาพกิจกรรมโดยปาปิรุส

2. เลือก “Papyrus Project” จากนั้นคลิกที่ปุ่ม “Next”



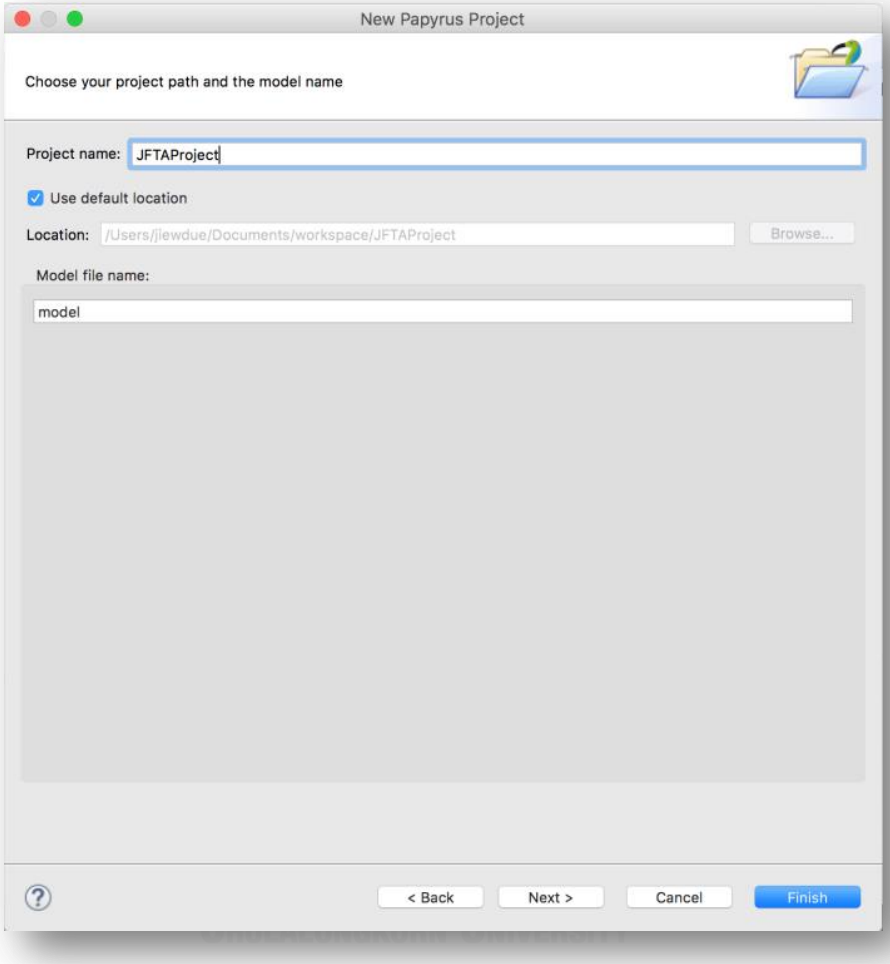
รูปที่ 100 หน้าจอเลือก Papyrus Project

3. เลือก “UML” จากนั้นคลิกที่ปุ่ม “Next”



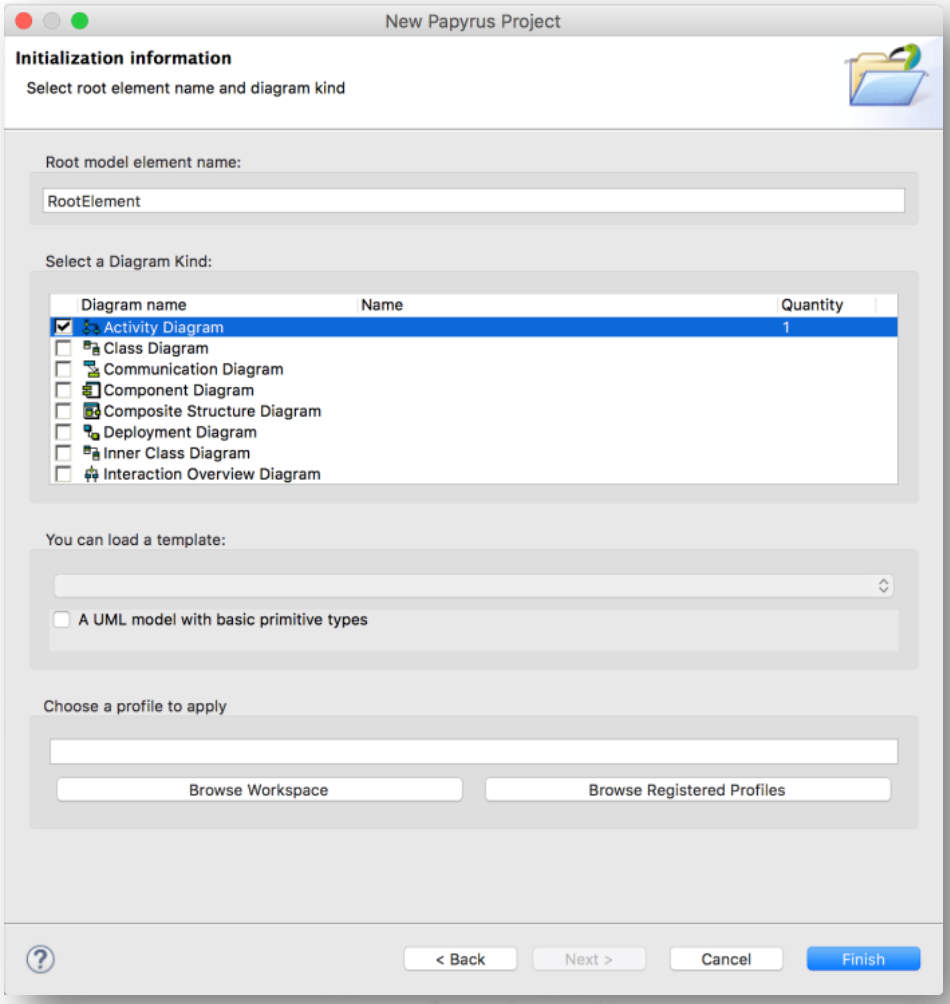
รูปที่ 101 หน้าจอ Diagram Language

4. ใส่ชื่อโปรเจคลงในช่อง “Project name” และใส่ชื่อโมเดลแผนภาพที่จะสร้างลงในช่อง “Model file name” จากนั้นคลิกที่ปุ่ม “Next”



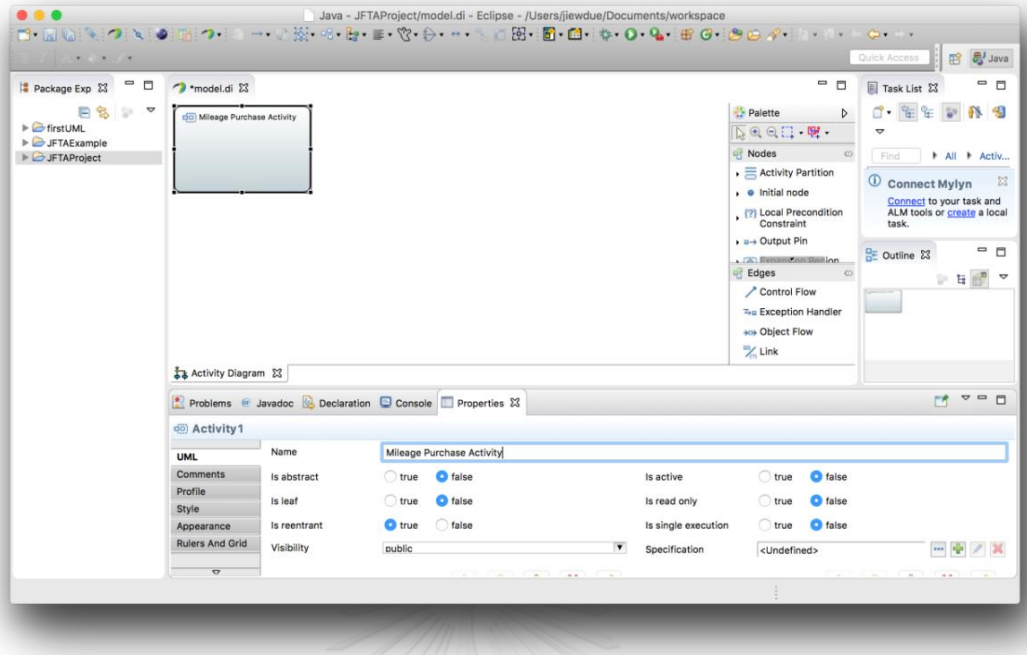
รูปที่ 102 หน้าจอ Project path and the model name

5. ใส่ชื่อ Root element ในช่อง “Root model element name” และประเภทของแผนภาพในส่วนของ “Select a Diagram Kind” ให้เลือกเป็น “Activity Diagram” จากนั้นคลิกที่ปุ่ม “Finish”



รูปที่ 103 หน้าจอ Initialization information

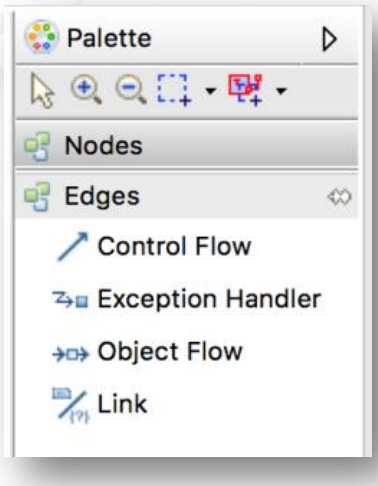
6. เมื่อสร้างโปรเจคเสร็จจะได้ไฟล์โมเดลเป็น .di ผู้ใช้งานสามารถเปลี่ยนชื่อแผนภาพในส่วนของ “Properties” ตรงช่อง “Name” จากนั้นผู้ใช้งานสามารถกดลากสัญลักษณ์จากทางด้านขวามือเข้ามาในกล่องของไฟล์ .di ได้ดังรูปที่ 104



รูปที่ 104 หน้าจอการสร้างแผนภาพกิจกรรมด้วยปาปิรุส มาร์ส

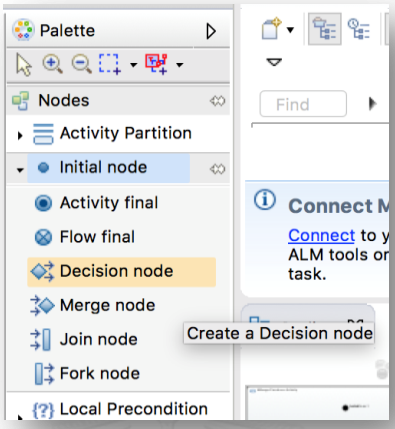
โดยสัญลักษณ์ของแผนภาพกิจกรรมบนโปรแกรมเสริมปาปิรุส มาร์ส ประกอบด้วย 2 ส่วนหลักคือ

- เส้น (Edges) - เส้นเชื่อมระหว่างโหนด โดยในงานวิจัยนี้เราจะใช้ตัว Control Flow เพื่ออธิบายถึงทางเดินของกิจกรรมหรือโหนด ดังรูปที่ 105



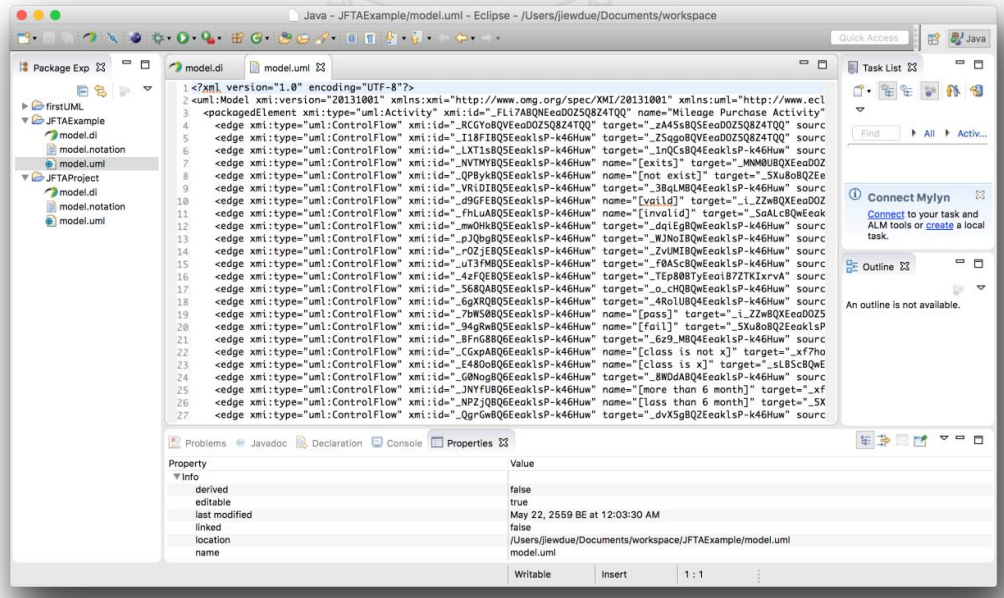
รูปที่ 105 สัญลักษณ์ของแผนภาพกิจกรรมบนโปรแกรมเสริมปาปิรุส 1

- โหนด (Nodes) - โหนดของกิจกรรม โดยสัญลักษณ์อื่นๆ นอกจากเส้นระหว่างโหนด จะอยู่ในหมวดของโหนดซึ่งได้แก่ จุดเริ่มต้นการทำงาน จุดสิ้นสุด จุดตัดสินใจ เป็นต้น ดังรูปที่ 106



รูปที่ 106 สัญลักษณ์ของแผนภาพกิจกรรมบนโปรแกรมเสริมปาปิรุส 2

7. เมื่อสร้างแผนภาพหรือเมทาโมเดลแล้วโปรแกรมจะสร้างไฟล์เอกซ์เอ็มไอ อยู่ในรูปแบบนามสกุลไฟล์ .uml ดังรูป 107 ซึ่งจะถูกใช้เป็นเมทาเดตาสำหรับเป็นข้อมูลนำเข้าในงานวิจัยนี้ โดยเราสามารถดึง .uml ตัวนี้มาใช้งานได้ในระบบของงานวิจัย

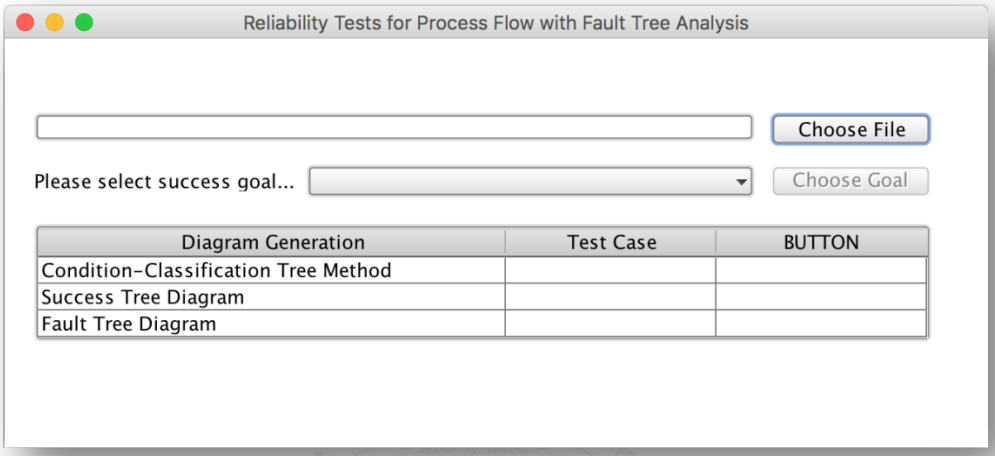


รูปที่ 107 เมทาเดตาที่ได้จากแผนภาพกิจกรรม

ภาคผนวก ง.

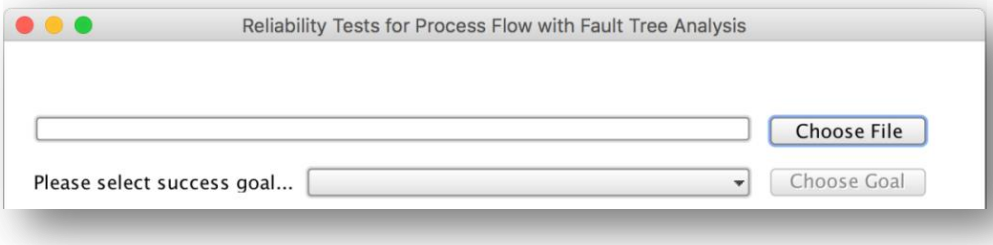
การใช้งานระบบ

ระบบสร้างแผนภาพต้นไม้ความผิดพลาด จะทำการสร้างแผนภาพต้นไม้การจำแนก ต้นไม้ความสำเร็จ และต้นไม้ความผิดพลาดจากเมทาเดตาของแผนภาพกิจกรรมซึ่งได้มาจากโปรแกรมเสริมปาปิรุส มาร์ส โดยเราจะนำผลลัพธ์ (output) ที่ได้จากปาปิรุส มาร์ส มาเป็นข้อมูลนำเข้า (input) ของระบบ



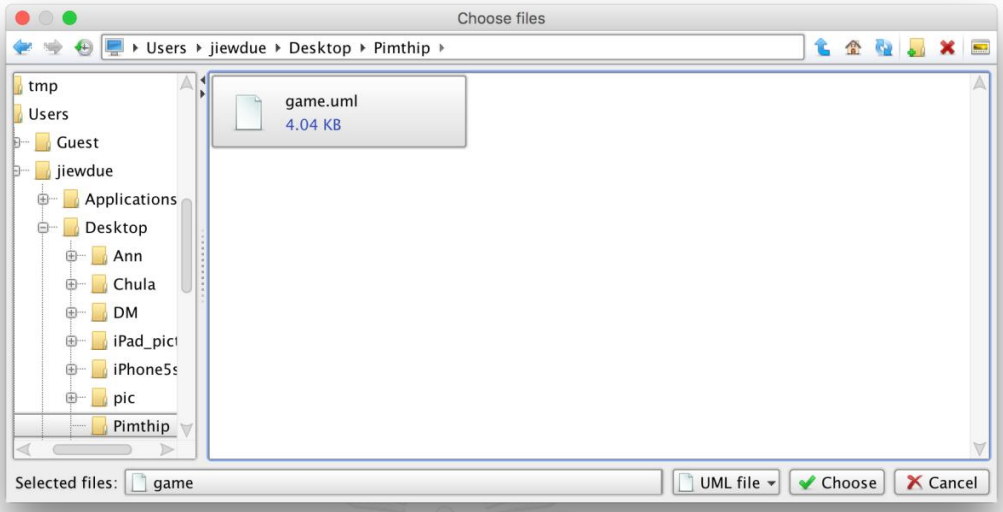
รูปที่ 108 หน้าจอการทำงานของระบบสร้างแผนภาพต้นไม้ความผิดพลาด

1. ในส่วนของข้อมูลนำเข้า ให้ผู้ใช้งานเลือกเมทาเดตาที่ได้จากปาปิรุส มาร์ส โดยกดที่ปุ่ม “Choose File” ดังรูปที่ 109



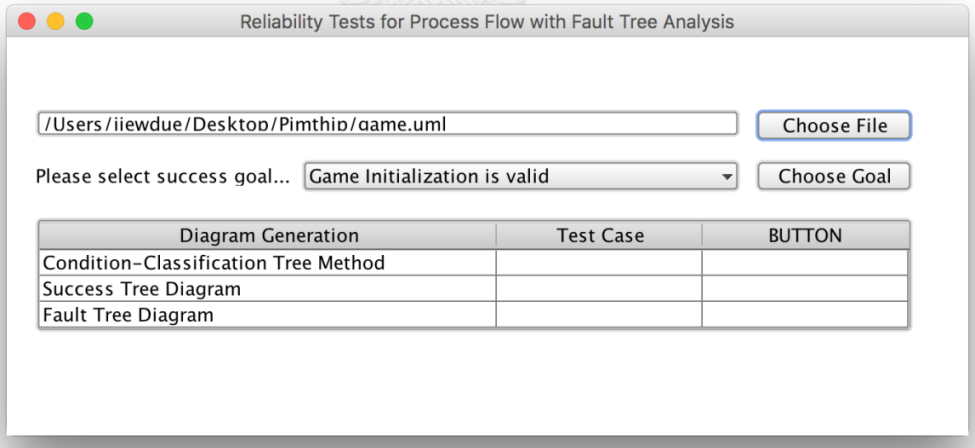
รูปที่ 109 หน้าจอนำเข้าเมทาเดตาของแผนภาพกิจกรรม

2. เลือกไฟล์เอกซ์เอ็มไอที่ต้องการนำเข้า จากนั้นกด “Choose”



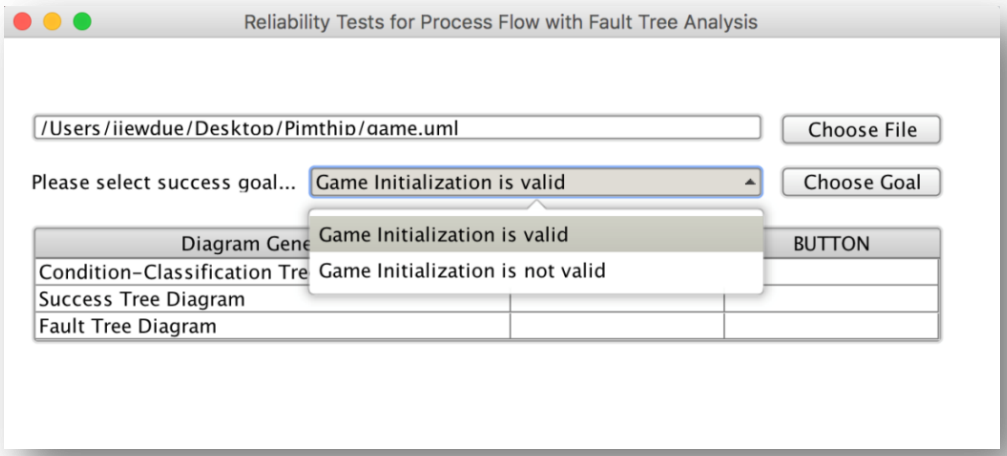
รูปที่ 110 หน้าจอแสดงการเลือกเมทาเดตาของแผนภาพกิจกรรม

3. เมื่อทำการเลือกไฟล์ได้แล้ว ระบบจะแสดงชื่อไฟล์ในกล่องข้อความ ดังรูปที่ 111



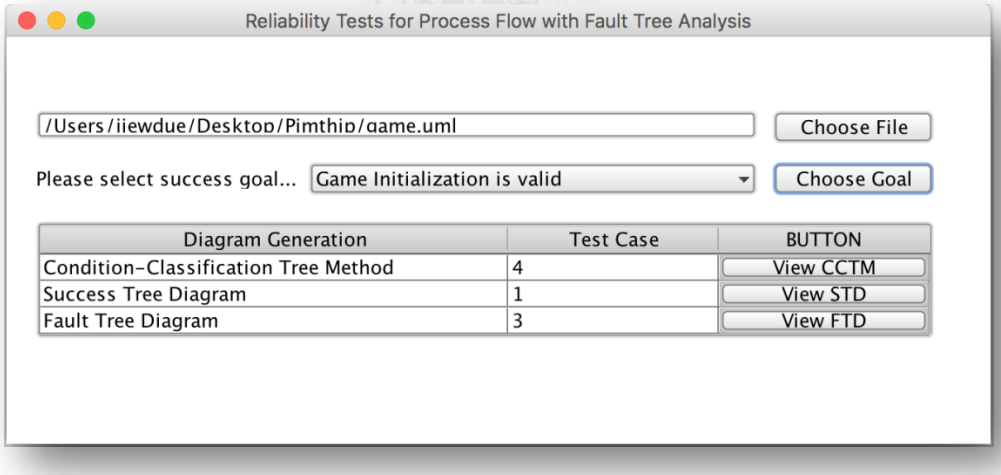
รูปที่ 111 หน้าจอแสดงข้อมูลเมทาเดตาที่นำเข้า

4. ระบบจะทำการสร้าง Success goal จากเมทาเดตาของแผนภาพกิจกรรม โดยจะต้องทำการเลือกเป้าหมายที่ถูกต้องซึ่งสามารถทำให้โปรแกรมทำงานได้อย่างสำเร็จได้ โดยเลือกที่ “Please select success goal...” จากนั้นกดที่ปุ่ม “Choose Goal”



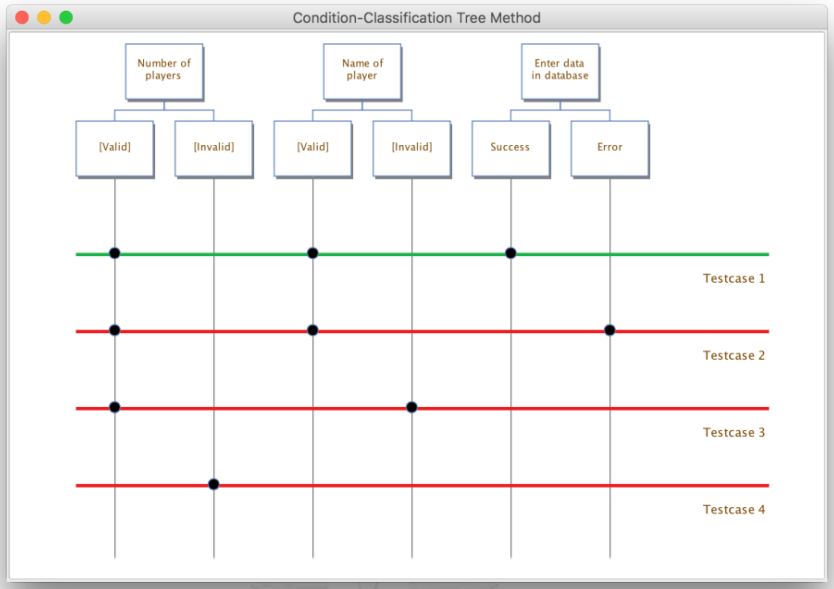
รูปที่ 112 หน้าจอแสดงการเลือกเป้าหมายที่ได้จากเมทาเดตาของแผนภาพกิจกรรม

5. ระบบแสดงจำนวนกรณีทดสอบที่เป็นไปได้และแผนภาพต้นไม้การจำแนก แผนภาพต้นไม้ความสำเร็จ และแผนภาพต้นไม้ความผิดพลาดตามลำดับ



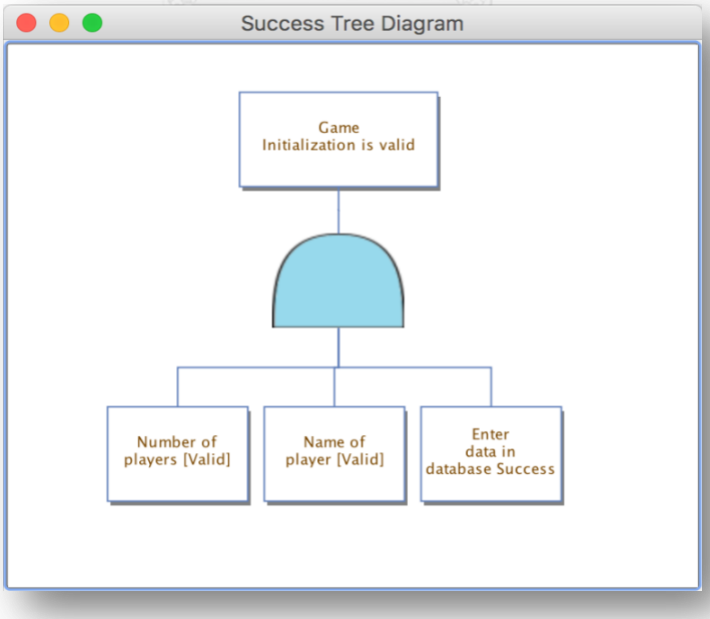
รูปที่ 113 หน้าจอแสดงผลลัพธ์ของระบบ

- ดูหน้าจอแผนภาพต้นไม้การจำแนกแบบมีเงื่อนไข โดยคลิกที่ปุ่ม “View CCTM”



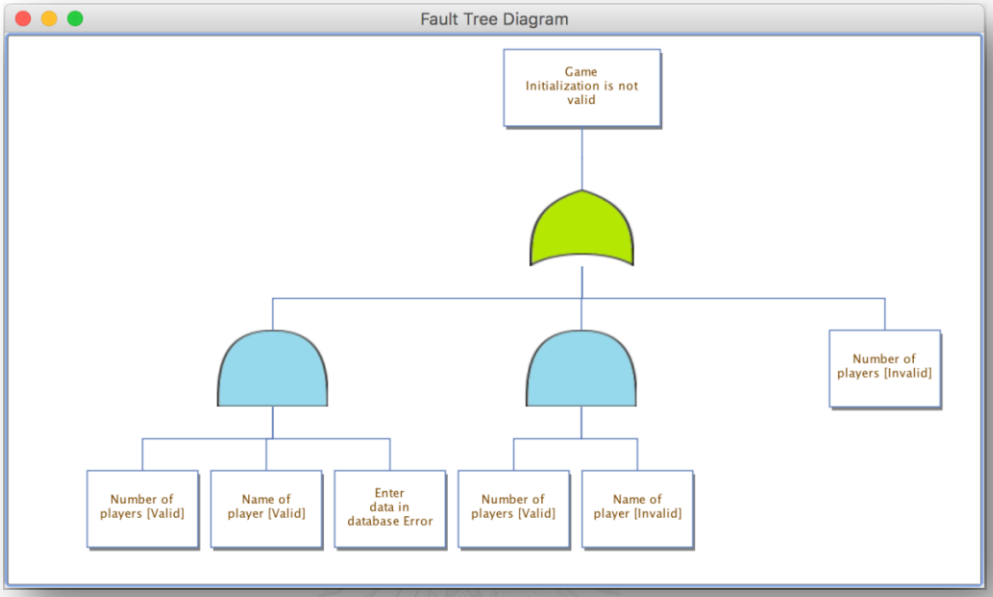
รูปที่ 114 หน้าจอแสดงผลลัพธ์ต้นไม้การจำแนกแบบมีเงื่อนไข

- ดูหน้าจอแผนภาพต้นไม้ความสำเร็จ โดยคลิกที่ปุ่ม “View STD”



รูปที่ 115 หน้าจอแสดงผลลัพธ์ต้นไม้ความสำเร็จ

- ดูหน้าจอแผนภาพต้นไม้ความผิดพลาด โดยคลิกที่ปุ่ม “View FTD”



รูปที่ 116 หน้าจอแสดงผลลัพธ์ต้นไม้ความผิดพลาด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวพิมพ์ทิพย์ ไพบูลย์เกษมสุทธิ์ เกิดเมื่อวันที่ 27 เมษายน พ.ศ. 2532 ที่จังหวัดสงขลา สำเร็จการศึกษาเกียรตินิยมอันดับหนึ่ง หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะเทคโนโลยีและสิ่งแวดล้อม ในปีการศึกษา 2555 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2557

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University