การศึกษาเทคนิคการสร้างกรณีทดสอบอัตโนมัติโดยใช้แผนภาพกิจกรรม

Study of Automatic Test Case Generation Techniques using UML Activity Diagram

วลัยภรณ์ ศรเกลี้ยง[1] และ ดร.ฐิมาพร เพชรแก้ว [2]

Walaiporn Sornkliang, Thimaporn Phetkeaw

[1]นักศึกษาปริญญาเอก สาขาการจัดการเทคโนโลยีสารสนเทศ สำนักวิชาสารสนเทศศาสตร์

มหาวิทยาลัยวลัยลักษณ์ นครศรีธรรมราช

[2]ผู้ช่วยศาสตราจารย์ สำนักวิชาสารสนเทศศาสตร์ มหาวิทยาลัยวลัยลักษณ์ นครศรีธรรมราช

[1]Emails: vlaiporn@hotmail.com, [2]pthimapo@wu.ac.th

ABSTRACT – There are many techniques to generate test case from UML activity diagram, this brings about the difficulty in selecting the appropriate technique for different types of program. In this paper, we compared the eight algorithms of the automatic test case generation: Round-Robin, Condition-Classification Tree, Activity Tree, Intermediate Testable Model, Activity Dependency Table and Activity Dependency Graph, Test Path Reduction, Activity Convert Grammar, and Activity Flow Graph. The results showed that (1) in the aspect of covering all paths, almost all algorithm could generate all possible paths while the Round-Robin could generate too many paths, (2) in the aspect of covering basis paths, all eight algorithms could generate all basis paths, (3) in the aspect of the complex model activity diagram that we constructed, the Activity Convert Grammar could generate too many paths, it has 44.23% of paths that exceed the number of target path, the Test Path Reduction could generate too few paths, it has 65.38% of paths lower than the number of target paths, and the Activity Flow Graph could generate the amount of paths as same as the expected paths, it has only 1.92% of paths that exceed the number of target paths.

KEY WORDS -- Software Testing; Test Cases; Test Cases Generation; UML Activity Diagram; Coverage Criteria

บทคัดย่อ -- การสร้างกรณีทดสอบจากแผนภาพกิจกรรมมีหลายวิธีและมีความยากในการตัดสินใจเลือกใช้ให้เหมาะสมกับลักษณะของโปรแกรมที่แตกต่างกัน ผู้วิจัยจึงทำการศึกษาเปรียบเทียบการสร้างกรณีทดสอบอัตโนมัติจากแผนภาพกิจกรรมดังนี้ วิธี Round-Robin วิธี Condition-Classification Tree วิธี Activity Tree วิธี Intermediate Testable Model วิธี Activity Dependency Table and Activity Dependency Graph วิธี Test Path Reduction วิธี Activity Convert Grammar และวิธี Activity Flow Graph ผลการศึกษามีดังนี้ (1) การครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้ พบว่าวิธีการส่วนใหญ่ผ่านเกณฑ์ข้อนี้ แต่วิธี Round-Robin จะสร้างกรณีทดสอบได้เกินกว่าเส้นทางทั้งหมดที่เป็นไปได้เป็นจำนวนมาก (2) การครอบคลุมเส้นทางพื้นฐาน พบว่าทุกวิธีสามารถสร้างเส้นทางการทดสอบได้ครอบคลุมเส้นทางพื้นฐานและ (3) จากแผนภาพกิจกรรมจำลองความซับซ้อนที่ผู้วิจัยสร้างขึ้นเมื่อวิเคราะห์ความคลาดเคลื่อนของจำนวนกรณีทดสอบที่สร้างได้เทียบกับเส้นทางที่เป็นไปได้ทั้งหมด พบว่าวิธี Activity Convert Grammar สร้างเส้นทางทดสอบมากเกินความจำเป็นโดยมีค่าความคลาดเคลื่อน 44.23% วิธี Test Path Reduction สร้างเส้นทางทดสอบได้น้อยกว่าที่จำเป็นต้องใช้โดยมีค่าความคลาดเคลื่อน -65.38% และวิธี Activity Flow Graph สร้างเส้นทางทดสอบได้ใกล้เคียงที่สุดมีค่าความคลาดเคลื่อนต่ำที่สุด คือ 1.92%

คำสำคัญ -- การทดสอบซอฟต์แวร์; กรณีทดสอบ; การสร้างกรณีทดสอบ; แผนภาพกิจกรรม; เกณฑ์การครอบคลุม

1. บทนำ

การทดสอบซอฟต์แวร์ (Software Testing) เป็นขั้นตอนหนึ่งที่มีความสำคัญของวงจรการพัฒนาระบบ สามารถออกแบบการทดสอบได้ตั้งแต่ช่วงต้นของกระบวนการพัฒนาระบบเทียบกับข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirement Specification) หรือเอกสารการออกแบบซอฟต์แวร์ (Detailed Design Document) จะช่วยลดเวลา และค่าใช้จ่ายในการพัฒนาซอฟต์แวร์ได้

ปัจจุบันซอฟต์แวร์ส่วนใหญ่พัฒนาขึ้นโดยใช้เทคโนโลยีเชิงวัตถุ และภาษาการออกแบบเชิงโมเดล (Unified Modelling Language: UML) ในการวิเคราะห์และการออกแบบการทำงานของซอฟต์แวร์ แผนภาพกิจกรรม (Activity Diagram) เป็นแผนภาพหนึ่งของภาษาการออกแบบเชิงโมเดลที่อธิบายพฤติกรรมการทำงานที่เกิดขึ้นตามลำดับหรือเกิดขึ้นพร้อมกันตั้งแต่เริ่มต้นจนสิ้นสุดการทำงานตามเส้นทางการทำงานของกิจกรรมต่างๆ ของโปรแกรม สามารถนำแผนภาพกิจกรรมมาใช้ในการออกแบบกรณีทดสอบ (Test Cases) ได้

การทดสอบซอฟต์แวร์ต้องสร้างกรณีทดสอบมาใช้ทดสอบเพื่อหาผลลัพธ์ที่คาดหวัง จากการศึกษางานวิจัยพบว่ามีเทคนิคในการสร้างกรณีทดสอบโดยอัตโนมัติจากแผนภาพกิจกรรม โดยแบ่งเป็น 3 กลุ่ม คือ การสร้างกรณีทดสอบจากต้นไม้ (Tree Based Methods) เช่น วิธี Round-Robin [1] วิธี Condition-Classification Tree [2] และวิธี Activity Tree [3] เป็นต้น การสร้างกรณีทดสอบจากกราฟ (Graph Based Methods) เช่น วิธี Intermediate Testable Model [4] วิธี Activity Dependency Table and Activity Dependency Graph [5] วิธี Test Path Reduction [6] วิธี Activity Convert Grammar [7] และวิธี Activity Flow Graph [8] เป็นต้น และการสร้างกรณีทดสอบด้วยวิธีการทางปัญญาประดิษฐ์ (Artificial Intelligence) เช่น ขั้นตอนวิธีพันธุกรรม (Genetic Algorithm) การหาค่าเหมาะสมที่สุดแบบฝูงผึ้งประดิษฐ์ (Artificial Bee Colony Optimization) และการหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมด (Ant Colony Optimization) เป็นต้น

การทดสอบซอฟต์แวร์ให้ครอบคลุมการทำงานของโปรแกรมมากที่สุด จะต้องนำกรณีทดสอบไปประเมินการครอบคลุมเส้นทางการทำงานของซอฟต์แวร์ ซึ่งใช้เกณฑ์การครอบคลุม (Coverage Criteria) สำหรับวัดการครอบคลุมเส้นทางของโปรแกรมจากกรณีทดสอบ เช่น การครอบคลุมคำสั่ง (Statement Coverage) การครอบคลุมทางเลือก (Branch Coverage) การครอบคลุมเส้นทางพื้นฐาน (Basis Path Coverage) การครอบคลุมเส้นทางกิจกรรม (Activity Path Coverage) และการครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้ (All Path Coverage) เป็นต้น การสร้างกรณีทดสอบจึงเป็นกระบวนการที่ต้องใช้ผู้มีประสบการณ์ และเข้าใจกระบวนการทางธุรกิจในการนำซอฟต์แวร์ไปใช้งาน จึงจะสามารถสร้างกรณีทดสอบให้ครอบคลุมในแต่ละระดับได้ จากการศึกษาพบว่าการสร้างกรณีทดสอบอัตโนมัติสำหรับแผนภาพกิจกรรม ทำให้สามารถลดเวลา และค่าใช้จ่ายโดยรวมของซอฟต์แวร์ลงได้

การสร้างกรณีทดสอบจากแผนภาพกิจกรรมมีหลายวิธี แต่ละวิธีมีวิธีการที่แตกต่างกันและให้ผลลัพธ์ครอบคลุมเส้นทางการทำงานของโปรแกรมไม่เท่ากัน จึงมีความยากในการเลือกใช้ให้เหมาะสมกับลักษณะของโปรแกรม ผู้วิจัยจึงทำการศึกษาเปรียบเทียบขั้นตอนวิธีการสร้างกรณีทดสอบอัตโนมัติโดยใช้แผนภาพกิจกรรมที่ผู้วิจัยสร้างขึ้น เพื่อจำลองความซับซ้อนของโปรแกรมและนำไปใช้ในการเปรียบเทียบ

เนื้อหาของบทความนี้ประกอบไปด้วย หัวข้อที่ 2 กล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง หัวข้อที่ 3 กล่าวถึงวิธีการดำเนินการวิจัย หัวข้อที่ 4 กล่าวถึงผลการทดลอง และหัวข้อที่ 5 เป็นบทสรุปและแนวทางการวิจัยในอนาคต

2. ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง

2.1 แผนภาพกิจกรรม

แผนภาพกิจกรรม (UML Activity Diagram) เป็นแผนภาพที่ใช้อธิบายขั้นตอนการทำงานที่เกิดขึ้นในลักษณะสายงานของการทำงาน (Workflow) แผนภาพกิจกรรมประกอบด้วย กิจกรรม (Activities) และเส้นทางของกิจกรรม (Transition) โดยกิจกรรมแทนด้วยสัญลักษณ์รูปสี่เหลี่ยมผืนผ้าโค้งมน ▢ จุดเริ่มต้นแทนด้วยสัญลักษณ์วงกลมทึบ ● จุดสุดท้ายแทนด้วยสัญลักษณ์วงกลมและมีวงกลมทึบอยู่ภายใน ◉ เส้นทางของกิจกรรมแสดงการควบคุมสายงานของการทำงานแทนด้วยสัญลักษณ์ลูกศรกำกับ → การตัดสินใจ (Decision) แสดงทางเลือกกิจกรรมแทนด้วยสัญลักษณ์รูปหลามตัด ◇ การรวมกัน (Merge) แสดงการรวมกิจกรรมมากกว่า 1 กิจกรรมไปสู่กิจกรรมหนึ่งแทนด้วยสัญลักษณ์ของรูปหลามตัด ◇ การแยกการทำงาน (Fork) แสดงกิจกรรมที่เกิดขึ้นในกรณีที่มีกิจกรรมมากกว่า 1 กิจกรรมและกิจกรรมเหล่านั้นสามารถ

ดำเนินไปพร้อมกันได้โดยไม่ต้องรออีกกิจกรรมหนึ่งเสร็จแทนด้วยสัญลักษณ์ การรวมการทำงาน (Join) แสดงการรวมกิจกรรมหลังดำเนินไปพร้อมกันแทนด้วยสัญลักษณ์ และการแบ่งหน้าที่การทำงาน (Swimlane) แสดงการแบ่งกลุ่มของกิจกรรมที่เกิดขึ้นแทนด้วยสัญลักษณ์

2.2 เกณฑ์การครอบคลุมในการทดสอบ

เกณฑ์การครอบคลุม (Coverage Criteria) ใช้สำหรับวัดการทดสอบว่าทำการทดสอบได้ครอบคลุมในระดับใดหรือใช้วัดว่ากรณีทดสอบสามารถครอบคลุมการทำงานของโปรแกรมในระดับใด ซึ่งระดับของเกณฑ์การครอบคลุม มีดังนี้

2.2.1 การครอบคลุมคำสั่ง คือ การทดสอบที่ต้องการให้ทุกคำสั่งได้รับการทดสอบอย่างน้อยหนึ่งครั้ง

2.2.2 การครอบคลุมทางเลือก คือ การทดสอบที่ต้องการให้ทุกเส้นทางที่เป็นผลลัพธ์จากการตัดสินใจได้รับการทดสอบอย่างน้อยหนึ่งครั้ง

2.2.3 การครอบคลุมเส้นทางพื้นฐาน คือ การทดสอบที่ต้องการให้มีจำนวนการทดสอบเป็นไปตามค่าที่ได้จากการคำนวณจำนวนเส้นทางการทดสอบกรณีทดสอบต่ำสุดที่เหมาะสมเมื่อเทียบกับความซับซ้อนของเส้นทางการทำงานของโปรแกรม

2.2.4 การครอบคลุมเส้นทางกิจกรรม คือ การทดสอบที่ต้องการให้ทุกกิจกรรมได้รับการทดสอบอย่างน้อยหนึ่งครั้ง และจะทำการทดสอบทุกเส้นทางที่เป็นไปได้จากกิจกรรมแรกถึงกิจกรรมสุดท้าย

2.2.5 การครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้ คือ การทดสอบทุกเส้นทางที่โปรแกรมสามารถทำงานได้

สำหรับงานวิจัยนี้ใช้ระดับของเกณฑ์การครอบคลุมทั้ง 5 ประเภทดังที่กล่าวไว้ข้างต้นนี้

2.3 การหาค่าจำนวนเส้นทางที่ครอบคลุมเส้นทางพื้นฐาน

การหาค่าจำนวนเส้นทางที่ครอบคลุมเส้นทางพื้นฐาน สามารถคำนวณจากความซับซ้อนไซโคลมาติก (Cyclomatic Complexity) [9] โดยกำหนดให้ V(G) คือ ค่าความซับซ้อนของกราฟสายงานควบคุม (Control Flow Graph) สามารถคำนวณได้จากวิธีการใดวิธีการหนึ่งตามสมการ (1), (2) หรือสมการ (3) ดังนี้

V(G) = จำนวนเส้นเชื่อมต่อระหว่างโนด – จำนวนโนด + 2 (1)

V(G) = จำนวนโนดการตัดสินใจ + 1 (2)

V(G) = จำนวนของพื้นที่แบบปิดของผังงานโปรแกรม +1 (3)

2.4 การสร้างกรณีทดสอบจากต้นไม้

ต้นไม้ คือ กราฟอย่างง่ายที่เชื่อมต่อกันเป็นลำดับชั้น มีเส้นเชื่อมเพียงเส้นเดียวระหว่างสองโนดใด ๆ เริ่มต้นจากโนดราก (Root Node) และสิ้นสุดที่โนดใบ (Leaf Node) งานวิจัยที่นำต้นไม้มาใช้เป็นเครื่องมือช่วยในการสร้างกรณีทดสอบมีดังนี้

วิธี Round-Robin [1] เป็นวิธีการสร้างกรณีทดสอบจากแผนภาพกิจกรรมย่อยก่อนแล้วจึงนำมารวมกับกรณีทดสอบที่สร้างจากแผนภาพกิจกรรมหลักตามลำดับชั้น โดยเริ่มจากนำแผนภาพกิจกรรมหลักมาสร้างต้นไม้ส่วนประกอบ (Composition Tree) และนำแผนภาพกิจกรรมย่อยมาสร้างต้นไม้ส่วนประกอบ จากนั้นหาเส้นทางที่เป็นไปได้จากต้นไม้ส่วนประกอบของแผนภาพกิจกรรมหลัก และหาเส้นทางที่เป็นไปได้จากต้นไม้ส่วนประกอบของแผนภาพกิจกรรมย่อย จากนั้นสร้างกรณีทดสอบโดยใช้เส้นทางที่เป็นไปได้จากต้นไม้ส่วนประกอบของแผนภาพกิจกรรมย่อยและเส้นทางที่เป็นไปได้จากต้นไม้ส่วนประกอบของแผนภาพกิจกรรมหลัก มารวมกันด้วยวิธี Round-Robin โดยเริ่มจากโนดรากของต้นไม้ส่วนประกอบของแผนภาพกิจกรรมหลัก แล้วใช้ขั้นตอนวิธีการค้นหาแนวกว้าง (Width First Searching Algorithm) ไปยังโนดระดับล่างของต้นไม้ จนถึงกิจกรรมที่อยู่ในต้นไม้ส่วนประกอบของแผนภาพกิจกรรมย่อย แล้วนำมารวมกันเป็นกรณีทดสอบ งานวิจัยนี้กรณีทดสอบที่สร้างได้ครอบคลุมตามเกณฑ์การครอบคลุมเส้นทางกิจกรรม และเกณฑ์การครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้

วิธี Condition-Classification Tree [2] ใช้แบบจำลองต้นไม้การจำแนกแบบมีเงื่อนไขสำหรับสร้างกรณีทดสอบจากแผนภาพกิจกรรม โดยพิจารณาส่วนการตัดสินใจของแผนภาพกิจกรรมมาสร้างต้นไม้การจำแนก ต้นไม้แต่ละต้นจะถูกสร้างโดยพิจารณาแต่ละตำแหน่งของการตัดสินใจจากเส้นทางก่อนเข้าถึงส่วนการตัดสินใจว่าได้ผ่านเงื่อนไขการตัดสินใจอื่นๆ ส่วนใดมาบ้าง กรณีที่พบให้นำเงื่อนไขการตัดสินใจนั้นระบุไว้ในต้นไม้การจำแนก โดยเรียงลำดับตามการตัดสินใจ จากนั้นสร้างกรณีทดสอบที่เกิดขึ้นตามเส้นทางจากโนดรากไปยังโนด

ใบ งานวิจัยนี้กรณีทดสอบที่สร้างได้ครอบคลุมตามเกณฑ์การครอบคลุมทางเลือก

วิธี Activity Tree [3] เป็นวิธีการสร้างกรณีทดสอบจากแผนภาพกิจกรรม ซึ่งจะพิจารณาในส่วนของการนำเสนอการสร้างกรณีทดสอบด้วยวิธีการสร้างต้นไม้กิจกรรม (Activity Tree) จากแผนภาพกิจกรรม โดยเริ่มจากนำแผนภาพกิจกรรมแปลงเป็นต้นไม้สายงานทดสอบ (Test Flow Tree) จากกิจกรรมเริ่มต้นไปถึงกิจกรรมสุดท้าย ในกรณีที่มีการวนซ้ำ (Loop) กิจกรรมก่อนเข้าสู่การวนซ้ำจะถูกนำมาใช้เป็นกิจกรรมสุดท้ายเมื่อสิ้นสุดการวนซ้ำ จากนั้นทำการค้นหาเส้นทางการทดสอบที่เป็นไปได้ทั้งหมด โดยใช้วิธีการค้นหาเส้นทางการทดสอบแบบแนวลึก (Depth First Search) ในกรณีเส้นทางทดสอบที่มีการวนซ้ำ จะทำการค้นหาเส้นทางทดสอบจากโนดเริ่มต้นไปยังโนดสุดท้ายในต้นไม้สายงานทดสอบได้มากสุดเพียงครั้งเดียว งานวิจัยนี้กรณีทดสอบที่สร้างได้ครอบคลุมตามเกณฑ์การครอบคลุมเส้นทางกิจกรรม

2.5 การสร้างกรณีทดสอบจากกราฟ

กราฟ คือ คู่ลำดับ G = (V, E) โดยเซตของ V (Vertices หรือ Nodes) เป็นโนด และ E (Edges) เป็นเส้นเชื่อมระหว่างจุด งานวิจัยที่นำกราฟมาใช้เป็นเครื่องมือช่วยในการสร้างกรณีทดสอบมีดังนี้

วิธี Intermediate Testable Model [4] ได้ศึกษาวิธีการสังเคราะห์สถานการณ์การทดสอบจากแผนภาพกิจกรรม เริ่มจากนำแผนภาพกิจกรรมมาสร้างเป็นกราฟสายงานควบคุม (Control Flow Graph) จากนั้นนำกราฟสายงานควบคุมมาจัดเป็นกลุ่มๆ คือ โครงสร้างแบบทางเลือก (Selection Constructs) โครงสร้างแบบวนซ้ำ (Loop Constructs) และโครงสร้างการแยกกิจกรรมที่ทำงานขนานกัน (Fork Constructs) โดยยุบรวมโนดที่อยู่ในกลุ่มเดียวกันเหลือเพียงโนดเดียว ในการสร้างกรณีทดสอบผู้ทดสอบสามารถเลือกโครงสร้างในส่วนที่สนใจมาทำการแตกเส้นทางการทดสอบย่อยได้ อย่างไรก็ตามหากนำทุกโครงสร้างมาพิจารณาในการสร้างเส้นทางการทดสอบก็จะได้เส้นทางทั้งหมดเท่ากับเกณฑ์การครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้ งานวิจัยนี้กรณีทดสอบที่สร้างได้ครอบคลุมเกณฑ์การครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้

วิธี Activity Dependency Table and Activity Dependency Graph [5] นำเสนอการสร้างกรณีทดสอบจากแผนภาพกิจกรรม โดยสร้างตารางการขึ้นต่อกันของกิจกรรม (Activity Dependency Table) แล้วนำมาสร้างกราฟการขึ้นต่อกันของกิจกรรม (Activity Dependency Graph) โดยครอบคลุมทุกกิจกรรม จากกราฟการขึ้นต่อกันของกิจรรมใช้วิธีการค้นหาเส้นทางการทดสอบแบบแนวลึก (Depth First Search) ในการหาเส้นทางการทดสอบที่เป็นไปได้ทั้งหมด โดยเทียบกับเกณฑ์การครอบคลุมเส้นทางพื้นฐาน ในกรณีที่กิจกรรมมีโครงสร้างแบบวนซ้ำ จะกำหนดให้ทำการวนซ้ำเพียงครั้งเดียว งานวิจัยนี้กรณีทดสอบที่สร้างได้ครอบคลุมตามเกณฑ์ การครอบคลุมเส้นทางพื้นฐาน และการครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้

วิธี Test Path Reduction [6] ได้ปรับปรุงเทคนิคการสร้างกรณีทดสอบบนแผนภาพกิจกรรม โดยนำแผนภาพกิจกรรมมาสร้างตารางการขึ้นต่อกันของกิจกรรม แล้วสร้างกราฟการขึ้นต่อกันของกิจกรรม เหมือนกับวิธี Activity Dependency Table and Activity Dependency Graph จากนั้นทำการปรับปรุงกราฟการขึ้นต่อกันของกิจกรรมด้วยการเอาออก (Remove) เช่น กิจกรรมที่มีชื่อซ้ำกัน สัญลักษณ์การตัดสินใจ (Decision) สัญลักษณ์ทางแยกการทำงาน (Fork) สัญลักษณ์การรวมการทำงาน (Join) และสัญลักษณ์การรวม (Merge) เมื่อปรับปรุงแล้วจะได้กราฟการขึ้นต่อกันของกิจกรรมหลังการปรับปรุง จากนั้นนำกราฟการขึ้นต่อกันของกิจกรรมหลังการปรับปรุงนี้ไปสร้างเส้นทางทดสอบ จากเส้นทางทดสอบนำไปสร้างกรณีทดสอบ งานวิจัยนี้กรณีทดสอบที่สร้างได้ครอบคลุมเกณฑ์การครอบคลุมเส้นทางพื้นฐาน และเกณฑ์การครอบคลุมทางเลือก

วิธี Activity Convert Grammar [7] นำเสนอการสร้างกรณีทดสอบจากแผนภาพกิจกรรมบนพื้นฐานของไวยากรณ์การแปลงกิจกรรม (Activity Convert Grammar) โดยนำแผนภาพกิจกรรมมาสร้างตารางการขึ้นต่อกันของกิจกรรม และตารางการขึ้นต่อกันการตัดสินใจ (Decision Dependency Table) จากชุดข้อมูลทดสอบ จากนั้นนำตารางการขึ้นต่อกันของกิจกรรมและตารางการขึ้นต่อกันการตัดสินใจมาพิจารณาร่วมกันในการสร้างเส้นทางทดสอบตามไวยากรณ์ โดยไวยากรณ์จะแบ่งเป็น 2 ส่วน คือ กิจกรรมในส่วนซ้าย (Left Hand Side: LHS) และกิจกรรมในส่วนขวา (Right Hand Side: RHS) ซึ่งกิจกรรมในส่วนซ้ายจะพิจารณากิจกรรมที่ขึ้นต่อกัน กิจกรรมในส่วนขวาจะทำการทดสอบกิจกรรม (Activity) กิจกรรมในทางเลือก

(Branch) กิจกรรมที่ทำงานขนานกัน (Fork) และสร้างกรณีทดสอบจากไวยากรณ์ โดยเริ่มจากกิจกรรมเริ่มต้น เรียงไปตามกิจกรรมตามลำดับ หากพบกิจกรรมแบบทางเลือกจะพิจารณาตารางการขึ้นต่อกันการตัดสินใจมาตรวจสอบตามเงื่อนไข และเรียงตามกิจกรรมในส่วนซ้ายไปกิจกรรมในส่วนขวา ส่วนในการแยกกิจกรรมที่ทำงานขนานกัน (Fork) จะทำการทดสอบทั้งแบบเรียงกิจกรรมจากซ้ายไปขวา และเรียงกิจกรรมจากขวาไปซ้าย งานวิจัยนี้กรณีทดสอบที่สร้างได้ครอบคลุมตามเกณฑ์การครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้

วิธี Activity Flow Graph [8] เป็นการสร้างกรณีทดสอบจากแผนภาพกิจกรรม โดยนำแผนภาพกิจกรรมไปสร้างตารางกิจกรรมสายงานควบคุม (Control Flow Activity Table) พร้อมกับให้ค่าของเงื่อนไขที่เกิดขึ้นในแต่ละกิจกรรม และนำไปสร้างกราฟสายงานกิจกรรม (Activity Flow Graph) เรียงตามลำดับของสายงานควบคุมจากกิจกรรมเริ่มต้นไปกิจกรรมสุดท้าย ซึ่งมีกิจกรรมทางเลือก กิจกรรมเงื่อนไข กิจกรรมการวนซ้ำ และการแยกกิจกรรมที่ทำงานขนานกัน แล้วหาเส้นทางทดสอบจากกราฟโดยวิธีการค้นหาเส้นทางการทดสอบแบบแนวลึก ในกรณีที่โนดมีโครงสร้างแบบวนซ้ำ การหาเส้นทางให้ทำการวนซ้ำเพียงครั้งเดียว จากนั้นนำแต่ละเส้นทางไปสร้างกรณีทดสอบ งานวิจัยนี้กรณีทดสอบที่สร้างได้ครอบคลุมตามเกณฑ์การครอบคลุมเส้นทางกิจกรรม

3. วิธีการดำเนินการวิจัย

ผู้วิจัยได้ดำเนินการวิจัยโดยศึกษาเปรียบเทียบการสร้างกรณีทดสอบจากแผนภาพกิจกรรม มีวิธีการดำเนินการวิจัย คือ (1) ศึกษาขั้นตอนวิธีการสร้างกรณีทดสอบอัตโนมัติโดยใช้แผนภาพกิจกรรมจำนวน 8 วิธี ดังที่กล่าวไว้ข้างต้น (2) กำหนดแผนภาพกิจกรรมที่ใช้ในการเปรียบเทียบโดยใช้แผนภาพกิจกรรมทั่วไปที่นิยมใช้จำนวน 3 แผนภาพ และสร้างแผนภาพกิจกรรมจำลองความซับซ้อนขึ้นเอง จำนวน 1 แผนภาพ โดยในแต่ละแผนภาพประกอบด้วย แผนภาพกิจกรรมหลัก และแผนภาพกิจกรรมย่อยของกิจกรรม (3) ทำการทดลองสร้างกรณีทดสอบ และ (4) เปรียบเทียบผลการสร้างกรณีทดสอบในประเด็นของการครอบคลุมเส้นทางและจำนวนกรณีทดสอบที่สร้างได้

แผนภาพกิจกรรมทั่วไปจำนวน 3 แผนภาพ ได้แก่ แผนภาพกิจกรรมการถอนเงินด้วยเอทีเอ็ม (ATM Withdraw Activity Diagram) แผนภาพกิจกรรมการจัดการธุรกิจ (The Business Management Activity Diagram) และแผนภาพกิจกรรมการสั่งซื้อออนไลน์ (Online Handle Order Activity Diagram) วิธีการคัดเลือกแผนภาพกิจกรรม คือ คัดเลือกแผนภาพกิจกรรมที่งานวิจัยด้านนี้ส่วนใหญ่นิยมใช้ และเป็นแผนภาพกิจกรรมที่เป็นระบบงานที่ใช้ในชีวิตประจำวัน เข้าใจง่าย นอกจากนี้คัดเลือกโดยพิจารณาจากโครงสร้างควบคุม (Control Structure) เช่น แผนภาพกิจกรรมการถอนเงินด้วยเอทีเอ็ม และแผนภาพกิจกรรมการสั่งซื้อออนไลน์ จะมีโครงสร้างควบคุมเป็นแบบโครงสร้างตามลำดับ (Sequence Structure) โครงสร้างแบบทางเลือก (Selection Structure) โครงสร้างแบบวนซ้ำ (Iteration Structure) และโครงสร้างการแยกกิจกรรมที่ทำงานขนานกัน (Fork Structure) ส่วนแผนภาพกิจกรรมการจัดการธุรกิจจะไม่มีโครงสร้างแบบวนซ้ำ ตัวแปรควบคุมของแต่ละแผนภาพกิจกรรม คือ จำนวนเส้นทางที่เป็นไปได้ทั้งหมด และจำนวนเส้นทางพื้นฐาน

แผนภาพกิจกรรมจำลองความซับซ้อน (Complex Model Activity Diagram) ที่ผู้วิจัยสร้างขึ้น ออกแบบโดยมีวัตถุประสงค์เพื่อเปรียบเทียบผลการสร้างกรณีทดสอบจากขั้นตอนวิธีการสร้างกรณีทดสอบอัตโนมัติจากแผนภาพกิจกรรมที่มีผู้นำเสนอไว้ว่ามีผลลัพธ์ของการสร้างกรณีทดสอบเป็นอย่างไร ผ่านเกณฑ์การครอบคลุมในระดับใด ซึ่งแผนภาพกิจกรรมจำลองความซับซ้อนมีโครงสร้างควบคุม เช่น โครงสร้างตามลำดับ โครงสร้างแบบทางเลือก ซึ่งในโครงสร้างแบบทางเลือกมีทั้งกรณี if-then และกรณี if-then-else โครงสร้างแบบวนซ้ำ และโครงสร้างการแยกกิจกรรมที่ทำงานขนานกัน ซึ่งในโครงสร้างการแยกกิจกรรมที่ทำงานขนานกันมีทั้งโครงสร้างตามลำดับ โครงสร้างแบบทางเลือก และโครงสร้างการแยกกิจกรรมที่ทำงานขนานกันอยู่ภายใน

นอกจากนี้ภายในแผนภาพกิจกรรมจำนวน 4 แผนภาพ จะมีกิจกรรมที่มีแผนภาพกิจกรรมย่อยด้วย เนื่องจากวิธี Round-Robin [1] มีการสร้างกรณีทดสอบจากแผนภาพกิจกรรมย่อยก่อนแล้วจึงนำมารวมกับกรณีทดสอบที่สร้างจากแผนภาพกิจกรรมหลักตามลำดับชั้น

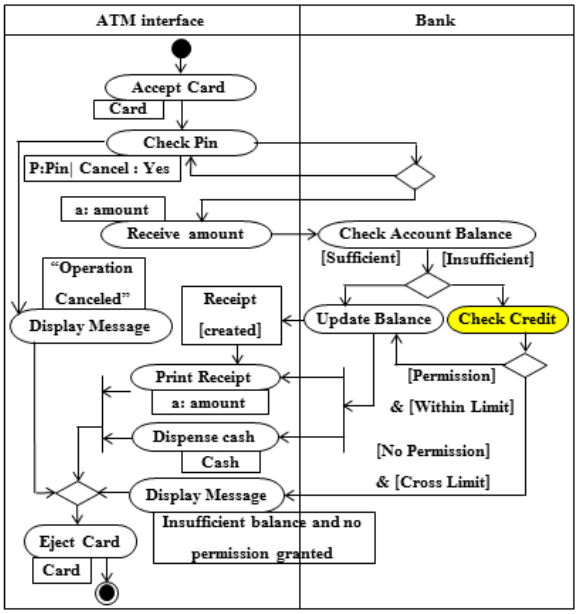
แผนภาพกิจกรรมจำนวน 4 แผนภาพที่ใช้ในการทดลองมีดังนี้ แผนภาพกิจกรรมการถอนเงินด้วยเอทีเอ็ม ดังรูปที่ 1 เริ่มจากเครื่องเอทีเอ็มรับบัตรทำการตรวจสอบรหัส ถ้าไม่ถูกต้องจะยกเลิก ถ้าถูกต้องจะรับข้อมูลจำนวนเงินแล้วตรวจสอบ

ยอดเงินในบัญชี ถ้ายอดเงินไม่เพียงพอ ระบบจะตรวจสอบสิทธิ์ในการเบิกเงิน ถ้าไม่อนุญาตจะแสดงข้อความและคืนบัตร กรณีที่ยอดเงินเพียงพอ ระบบปรับปรุงยอดเงิน พิมพ์ใบเสร็จรับเงินถอนเงิน สุดท้ายคืนบัตร และแผนภาพกิจกรรมย่อยของกิจกรรม “Check Credit” จากรูปที่ 1 แสดงดังรูปที่ 2 ระบบตรวจสอบสิทธิ์ในการเบิกเงินที่ต้องการด้วยค่าสูงสุดที่อนุญาต ถ้ามีการอนุญาต ระบบจะตรวจสอบจำนวนเงินที่ต้องการ หากพบว่าจำนวนเงินอยู่ภายในวงเงินที่กำหนด จะอนุญาตให้เบิกเงินเกินบัญชี

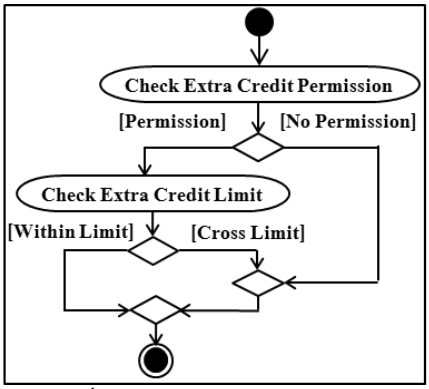
แผนภาพกิจกรรมการจัดการธุรกิจ ดังรูปที่ 3 เริ่มจากตรวจสอบบัญชีผู้ใช้ หากมีบัญชีผู้ใช้จะตัดสินใจว่าจะระงับบัญชีผู้ใช้หรือไม่ ถ้าไม่ระงับจะทำการโอนบัญชีนี้ให้กับผู้อื่นและทำลายบัญชีผู้ใช้นี้ และสิ้นสุดการทำงาน แต่ถ้าใช่จะระงับบัตรไอซี จากนั้นตรวจสอบว่าบัตรไอซีสูญหายหรือไม่ ถ้าสูญหายเปลี่ยนบัตรไอซีใหม่ ถ้าไม่สูญหายจะต่ออายุบัตรไอซี และสิ้นสุดการทำงาน ในกรณีตรวจสอบบัญชีผู้ใช้พบว่าไม่มี ทำการสร้างบัญชีผู้ใช้ แล้วซื้อบัตรไอซี และซื้อกล่องสัญญาณ จากนั้นสั่งซื้อสินค้า สิ้นสุดการทำงาน และแผนภาพกิจกรรมย่อยของกิจกรรม “Create an user account” จากรูปที่ 3 แสดงดังรูปที่ 4 เริ่มจากเข้าสู่ระบบ แล้วตรวจสอบบัญชีผู้ใช้และรหัสผ่าน ถ้ารหัสผ่านถูกต้องทำการเพิ่มชื่อ เพศ วันเดือนปีเกิด ที่อยู่ การติดต่อ หมายเลขโทรศัพท์ พื้นที่ ผู้ติดต่อ ประเภทผู้ใช้และส่วนลด จากนั้นจ่ายค่าธรรมเนียม แต่ถ้ารหัสผ่านไม่ถูกต้องจะสิ้นสุดการทำงาน

แผนภาพกิจกรรมการสั่งซื้อออนไลน์ ดังรูปที่ 5 เริ่มจากเลือกสินค้าเสร็จแล้วเข้าสู่กระบวนการสั่งซื้อ โดยกรอกใบสั่งซื้อ ส่งใบสั่งซื้อ และคำนวณบิล ถ้าเป็นลูกค้าวีไอพีมีการคำนวณส่วนลดแล้วส่งไปเตรียมใบแจ้งหนี้ เตรียมรับประกัน จากนั้นจัดส่งสินค้าและส่งใบเอกสารตอบรับโดยแสดงเป็นข้อความ และแผนภาพกิจกรรมย่อยของกิจกรรม “Selection Product” จากรูปที่ 5 แสดงดังรูปที่ 6 ในการเลือกสินค้า ถ้าเลือกรายละเอียดจะทำการเลือกชนิดสินค้า เลือกยี่ห้อ เลือกสี และเลือกราคา จากนั้นระบบแสดงรายละเอียดของสินค้า ถ้าไม่เลือกรายละเอียดจะยกเลิกการเลือกสินค้า

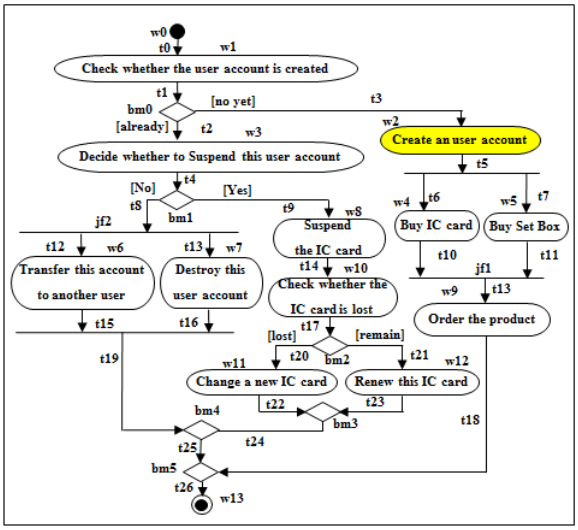
แผนภาพกิจกรรมจำลองความซับซ้อนที่สร้างขึ้น ดังรูปที่ 7 เริ่มจากกิจกรรม A ถ้า B เป็นเท็จทำกิจกรรม C ถ้า B เป็นจริงทำกิจกรรม D และกิจกรรม C แล้วตรวจสอบเงื่อนไข D ถ้าเป็นจริงทำกิจกรรม F จนกระทั่ง D เป็นเท็จ ถ้า G เป็นจริงทำกิจกรรม H แล้วแยกการทำงานทำกิจกรรม L แล้วแยกการทำงานทำกิจกรรม N และกิจกรรม O จากนั้นตัดสินใจถ้า E เป็นเท็จทำกิจกรรม R แต่ถ้า E เป็นจริงทำกิจกรรม S และกิจกรรม T แล้วรวมกันทำกิจกรรม V แล้วตัดสินใจถ้า W เป็นเท็จทำกิจกรรม X ถ้า W เป็นจริงทำกิจกรรม Y แล้วรวมกันทำกิจกรรม AA สุดท้ายรวมการแยกทำกิจกรรม AC ถ้า G เป็นเท็จทำกิจกรรม I, J, AC จบการทำงาน และแผนภาพกิจกรรมย่อยของกิจกรรม “J” จากรูปที่ 7 แสดงดังรูปที่ 8 เริ่มจากทำกิจกรรม J1 แล้วตัดสินใจ ถ้า J2 เป็นจริงทำกิจกรรม J3 แล้วตรวจสอบเงื่อนไข J4 เป็นเท็จทำกิจกรรม J3 ซ้ำ หาก J4 เป็นจริงตรวจสอบเงื่อนไข J5 ถ้าเป็นจริงทำกิจกรรม J6 แล้วตรวจสอบเงื่อนไข J7 เป็นจริงทำกิจกรรม J8 แล้วจบการทำงาน



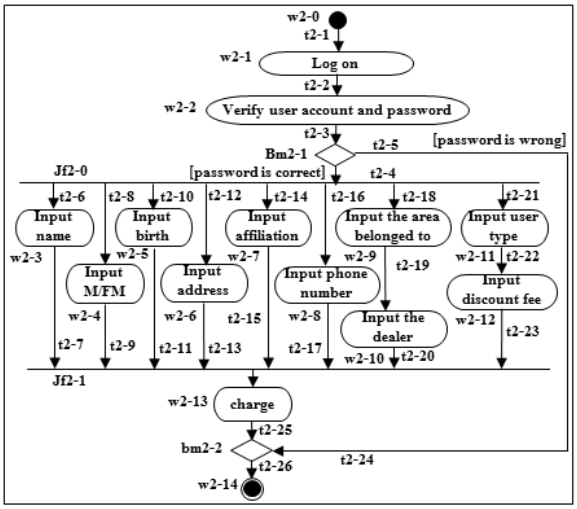
รูปที่ 1. แสดงแผนภาพกิจกรรมการถอนเงินด้วยเอทีเอ็ม [6]



รูปที่ 2. แสดงแผนภาพกิจกรรมย่อยของกิจกรรม “Check Credit”

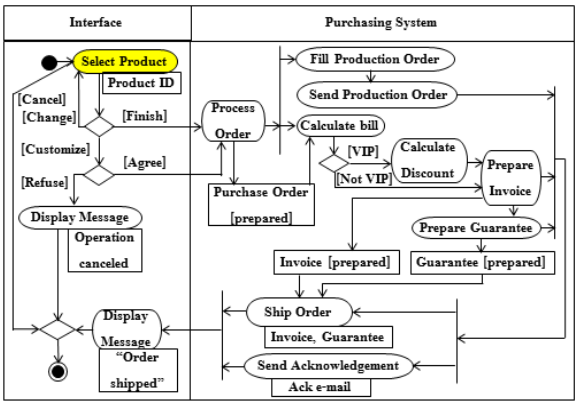


รูปที่ 3. แสดงแผนภาพกิจกรรมการจัดการธุรกิจ [1]

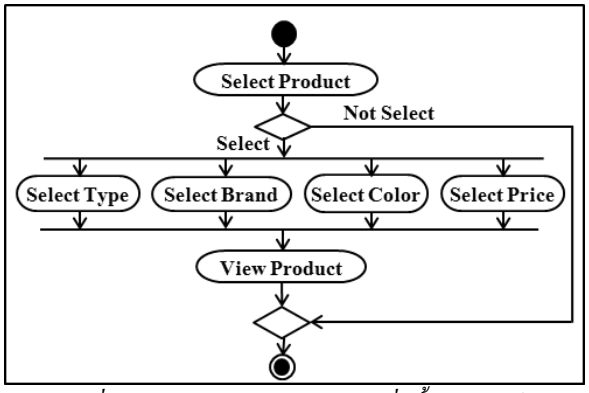


รูปที่ 4. แสดงแผนภาพกิจกรรมย่อย

ของกิจกรรม "Create an user account" [1]

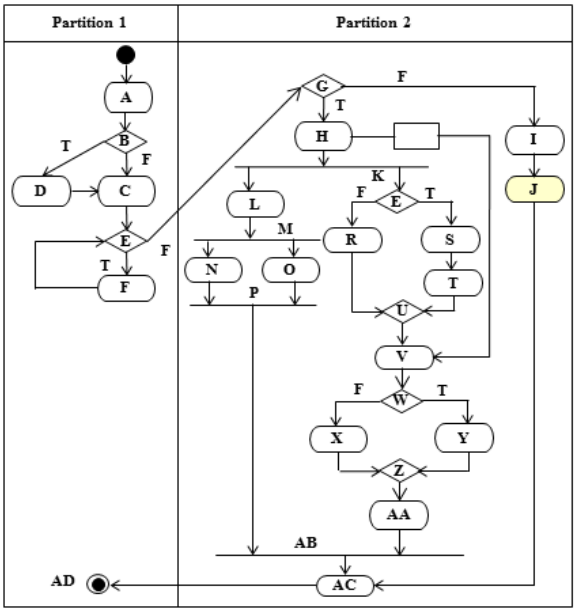


รูปที่ 5. แสดงแผนภาพกิจกรรมการสั่งซื้อออนไลน์ [10]

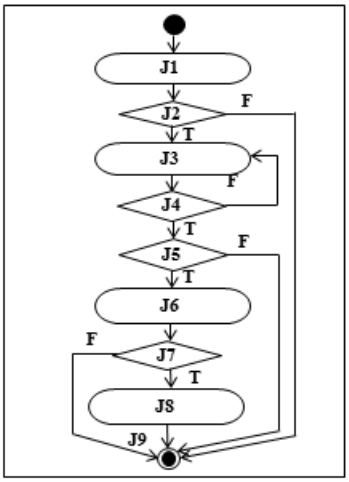


รูปที่ 6. แสดงแผนภาพกิจกรรมการสั่งซื้อออนไลน์

ของกิจกรรม "Select Product"



รูปที่ 7. แสดงแผนภาพกิจกรรมจำลองความซับซ้อน



รูปที่ 8. แสดงแผนภาพกิจกรรมจำลองความซับซ้อนย่อย

ของกิจกรรม "J"

4. ผลการทดลอง

ผลการทดลองการเปรียบเทียบขั้นตอนวิธีของการสร้างกรณีทดสอบอัตโนมัติจากแผนภาพกิจกรรม ผู้วิจัยใช้แผนภาพกิจกรรมจำนวน 4 แผนภาพ คือ แผนภาพกิจกรรมหลัก และแผนภาพกิจกรรมย่อย จากรูปที่ 1 ถึงรูปที่ 8 มาใช้ในการทดสอบกับขั้นตอนวิธีของการสร้างกรณีทดสอบอัตโนมัติจากแผนภาพกิจกรรมจากงานวิจัยที่ศึกษาจำนวน 8 วิธี ผลการสร้างกรณีทดสอบตามเทคนิคการสร้างกรณีทดสอบจากต้นไม้แสดงดังตารางที่ 1 และผลการสร้างกรณีทดสอบตามเทคนิคการสร้างกรณีทดสอบจากกราฟ แสดงดังตารางที่ 2 ในการคำนวณหาเส้นทางพื้นฐานใช้สมการ (1), (2) หรือ (3) แล้วเลือกค่าที่เป็นผลลัพธ์ส่วนใหญ่จาก 3 สมการนี้

ตารางที่ 1. แสดงผลการทดลองการสร้างกรณีทดสอบจากแผนภาพกิจกรรม โดยใช้เทคนิคการสร้างกรณีทดสอบจากต้นไม้

แผนภาพกิจกรรม	เส้นทางที่เป็นไปได้ทั้งหมด	เส้นทางพื้นฐาน	วิธี Round-Robin						วิธี Condition-Classification Tree						วิธี Activity Tree					
			กรณีทดสอบที่สร้างได้	เกณฑ์การครอบคลุม					กรณีทดสอบที่สร้างได้	เกณฑ์การครอบคลุม					กรณีทดสอบที่สร้างได้	เกณฑ์การครอบคลุม				
				คำสั่ง	ทางเลือก	เส้นทางพื้นฐาน	เส้นทางกิจ-กรรม	เส้นทางทั้งหมด		คำสั่ง	ทางเลือก	เส้นทางพื้นฐาน	เส้นทางกิจ-กรรม	เส้นทางทั้งหมด		คำสั่ง	ทางเลือก	เส้นทางพื้นฐาน	เส้นทางกิจ-กรรม	เส้นทางทั้งหมด
การถอนเงินด้วยเอทีเอ็ม	24	6	30	✓	✓	✓	✓	✓	16	✓	✓	✓	×	×	13	✓	✓	✓	×	×
การจัดการธุรกิจ	5	5	6	✓	✓	✓	✓	✓	5	✓	✓	✓	✓	✓	5	✓	✓	✓	✓	✓
การสั่งซื้อออนไลน์	40	5	108	✓	✓	✓	✓	✓	14	✓	✓	✓	×	×	19	✓	✓	✓	×	×
แผนภาพกิจกรรมจำลองความซับซ้อน	52	10	56	✓	✓	✓	✓	✓	44	✓	✓	✓	✓	×	24	✓	✓	✓	×	×

ตารางที่ 2. แสดงผลการทดลองการสร้างกรณีทดสอบจากแผนภาพกิจกรรม โดยใช้เทคนิคการสร้างกรณีทดสอบจากกราฟ

แผนภาพกิจกรรม	เส้นทางที่เป็นไปได้ทั้งหมด	เส้นทางพื้นฐาน	วิธี Intermediate Testable Model						วิธี Activity Dependency Table and Activity Dependency Graph						วิธี Test Path Reduction						วิธี Activity Convert Grammar						วิธี Activity Flow Graph					
			กรณีทดสอบที่สร้างได้	เกณฑ์การครอบคลุม					กรณีทดสอบที่สร้างได้	เกณฑ์การครอบคลุม					กรณีทดสอบที่สร้างได้	เกณฑ์การครอบคลุม					กรณีทดสอบที่สร้างได้	เกณฑ์การครอบคลุม					กรณีทดสอบที่สร้างได้	เกณฑ์การครอบคลุม				
				คำสั่ง	ทางเลือก	เส้นทางพื้นฐาน	เส้นทางกิจกรรม	เส้นทางทั้งหมด		คำสั่ง	ทางเลือก	เส้นทางพื้นฐาน	เส้นทางกิจกรรม	เส้นทางทั้งหมด		คำสั่ง	ทางเลือก	เส้นทางพื้นฐาน	เส้นทางกิจกรรม	เส้นทางทั้งหมด		คำสั่ง	ทางเลือก	เส้นทางพื้นฐาน	เส้นทางกิจกรรม	เส้นทางทั้งหมด		คำสั่ง	ทางเลือก	เส้นทางพื้นฐาน	เส้นทางกิจกรรม	เส้นทางทั้งหมด
การถอนเงินด้วยเอทีเอ็ม	24	6	26	✓	✓	✓	✓	✓	16	✓	✓	✓	×	×	6	✓	✓	✓	×	×	26	✓	✓	✓	✓	✓	25	✓	✓	✓	✓	✓
การจัดการธุรกิจ	5	5	*	✓	✓	✓	✓	✓	5	✓	✓	✓	✓	✓	5	✓	✓	✓	✓	✓	9	✓	✓	✓	✓	✓	5	✓	✓	✓	✓	✓
การสั่งซื้อออนไลน์	40	5	189	✓	✓	✓	✓	✓	20	✓	✓	✓	×	×	7	✓	✓	✓	×	×	76	✓	✓	✓	✓	✓	41	✓	✓	✓	✓	✓
แผนภาพกิจกรรมจำลองความซับซ้อน	52	10	61	✓	✓	✓	✓	✓	56	✓	✓	✓	✓	✓	18	✓	✓	✓	×	×	75	✓	✓	✓	✓	✓	53	✓	✓	✓	✓	✓

หมายเหตุ * = 907,200

จากผลการทดลองสามารถสรุปเป็น 3 ประเด็น ได้แก่ การครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้ การครอบคลุมเส้นทางพื้นฐาน และผลการสร้างกรณีทดสอบจากแผนภาพกิจกรรมจำลองความซับซ้อน

ประเด็นการครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้ จากตารางที่ 1 และตารางที่ 2 พบว่า วิธี Round-Robin วิธี Intermediate Testable Model วิธี Activity Convert Grammar และวิธี Activity Flow Graph สามารถสร้างกรณีทดสอบผ่านเกณฑ์การครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้ โดยวิธี Round-Robin จะสร้างกรณีทดสอบได้เกินกว่าเส้นทางทั้งหมดที่เป็นไปได้เยอะมาก หากแผนภาพกิจกรรมนั้นในกิจกรรมแรกๆ มีกิจกรรมย่อย เพราะเส้นทางในแผนภาพกิจกรรมย่อยจะถูกนำมากระจายเข้าไปในเส้นทางการทดสอบที่ได้จากแผนภาพกิจกรรมหลัก วิธี Intermediate Testable Model จะสร้างกรณีทดสอบได้จำนวนมากเป็นทวีคูณ หากมีกิจกรรมที่ทำงานขนานกันเป็นจำนวนมาก ส่วนวิธี Activity Convert Grammar และวิธี Activity Flow Graph นอกจากจะได้เส้นทางทดสอบครอบคลุมเส้นทางทั้งหมดแล้ว จะเห็นได้ว่าจำนวนเส้นทางมากกว่าเส้นทางที่เป็นไปได้ทั้งหมดเพียงเล็กน้อย

ประเด็นการครอบคลุมเส้นทางพื้นฐาน ในทางปฏิบัติการทดสอบทุกเส้นทางที่เป็นไปได้ทำได้ยาก เนื่องจากแผนภาพกิจกรรมที่มีขนาดใหญ่และซับซ้อนจะมีเส้นทางที่เป็นไปได้จำนวนมาก ในกรณีนี้ผู้ทดสอบอาจเลือกใช้เกณฑ์การครอบคลุมเส้นทางพื้นฐาน จากตารางที่ 1 และตารางที่ 2 พบว่าวิธีการสร้างกรณีทดสอบทั้ง 8 วิธี สามารถสร้างกรณีทดสอบที่ครอบคลุมเส้นทางพื้นฐานได้

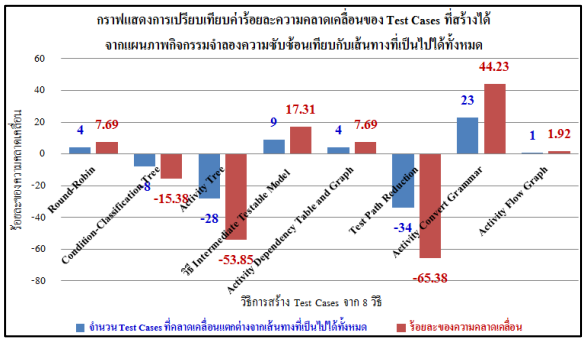
ประเด็นการสร้างกรณีทดสอบจากแผนภาพกิจกรรมจำลองความซับซ้อนที่ผู้วิจัยสร้างขึ้นซึ่งมีรูปแบบของกิจกรรมครบถ้วน จากตารางที่ 1 และตารางที่ 2 พบว่า วิธี Round-Robin วิธี Intermediate Testable Model วิธี Activity Dependency Table and Activity Dependency Graph วิธี Activity Convert Grammar และวิธี Activity Flow Graph ยังคงสามารถสร้างกรณีทดสอบได้ครอบคลุมทุกเส้นทางที่เป็นไปได้ทั้งหมด ในขณะที่วิธี Condition-Classification Tree วิธี Activity Tree และวิธี Test Path Reduction ไม่สามารถสร้างเส้นทางได้ครบทุกเส้นทางที่เป็นไปได้ และจากรูปที่ 9 ซึ่งแสดงกราฟการเปรียบเทียบค่าร้อยละความคลาดเคลื่อนของจำนวนกรณีทดสอบที่สร้างได้จากแผนภาพกิจกรรมจำลองความซับซ้อนเทียบกับเส้นทางที่เป็นไปได้ทั้งหมด

การคำนวณค่าร้อยละความคลาดเคลื่อนของจำนวนกรณีทดสอบของวิธีการใดๆ ว่ามีความคลาดเคลื่อนไปจากจำนวนเส้นทางที่เป็นไปได้ทั้งหมดของแผนภาพกิจกรรมจำลองความซับซ้อนเท่าใด สามารถคำนวณได้ดังนี้ กำหนดให้ T คือ จำนวนกรณีทดสอบของวิธีการใดๆ และ N คือ จำนวนเส้นทางที่เป็นไปได้ทั้งหมดของแผนภาพกิจกรรมจำลองความซับซ้อน สามารถคำนวณค่าความคลาดเคลื่อน Deviation (D) ได้ด้วยสมการ (4) ดังนี้

$$D = \frac{(T-N)\times 100}{N} \quad (4)$$

ตัวอย่างเช่น วิธี Round-Robin มีจำนวนกรณีทดสอบที่สร้างได้เท่ากับ 56 เส้นทาง ส่วนจำนวนเส้นทางที่เป็นไปได้ทั้งหมดของแผนภาพกิจกรรมจำลองความซับซ้อนมีเท่ากับ 52 เส้นทาง ดังนั้นจึงมีค่าร้อยละความคลาดเคลื่อนของวิธี Round-Robin เท่ากับ (56-52)x100/52 = 7.69 ค่าความคลาดเคลื่อนที่เป็นค่าบวกหมายถึงสร้างเส้นทางทดสอบได้มากเกินความจำเป็น ส่วนค่าความคลาดเคลื่อนที่เป็นค่าลบหมายถึงสร้างเส้นทางทดสอบได้น้อยเกินกว่าที่จำเป็นต้องใช้

การเปรียบเทียบค่าร้อยละความคลาดเคลื่อนของจำนวนกรณีทดสอบที่สร้างได้จากแผนภาพกิจกรรมจำลองความซับซ้อนเทียบกับเส้นทางที่เป็นไปได้ทั้งหมดพบว่า วิธี Activity Convert Grammar มีค่าความคลาดเคลื่อนสูงถึง 44.23% กล่าวคือ สร้างเส้นทางทดสอบได้มากเกินความจำเป็น วิธี Test Path Reduction มีค่าความคลาดเคลื่อนสูงถึง -65.38% กล่าวคือ สร้างเส้นทางทดสอบได้น้อยกว่าที่จำเป็นต้องใช้ และวิธี Activity Flow Graph มีค่าความคลาดเคลื่อนต่ำที่สุด คือ 1.92% กล่าวคือ สร้างเส้นทางทดสอบได้ใกล้เคียงกับเส้นทางที่เป็นไปได้ทั้งหมด



รูปที่ 9. แสดงกราฟการเปรียบเทียบค่าร้อยละความคลาดเคลื่อนของจำนวนกรณีทดสอบที่สร้างได้

ข้อเสนอแนะในการนำแผนภาพกิจกรรมมาพิจารณาถึงความเหมาะสมกับขั้นตอนวิธีการสร้างกรณีทดสอบทั้ง 8 วิธี เพื่อให้ได้จำนวนกรณีทดสอบที่ผ่านเกณฑ์การครอบคลุม และมีจำนวนกรณีทดสอบที่มีค่าร้อยละความคลาดเคลื่อนน้อยที่สุด จากผลการทดลองในตารางที่ 1 ตารางที่ 2 และค่าความคลาดเคลื่อนจากภาพที่ 9 สามารถสรุปได้ดังตารางที่ 3

ตารางที่ 3. แผนภาพกิจกรรมที่เหมาะสมกับวิธีการทดสอบ

ขั้นตอนวิธี	แผนภาพกิจกรรมที่เหมาะสม
Round-Robin	แผนภาพกิจกรรมหลักที่ไม่ซับซ้อน หากมีแผนภาพกิจกรรมย่อยควรอยู่ส่วนท้ายๆ ของแผนภาพกิจกรรมหลัก เนื่องจากเส้นทางในแผนภาพกิจกรรมย่อยจะถูกนำมากระจายเข้าไปในเส้นทางการทดสอบที่ได้จากแผนภาพกิจกรรมหลัก รวมถึงกรณีที่ไม่มีโครงสร้างควบคุมแบบวนซ้ำ
Condition-Classification Tree	แผนภาพกิจกรรมที่ไม่ซับซ้อน มีโครงสร้างควบคุมแบบตามลำดับและแบบทางเลือกเท่านั้น
Activity Tree	แผนภาพกิจกรรมที่ไม่ซับซ้อน มีโครงสร้างควบคุมแบบตามลำดับ แบบทางเลือก และอาจมีการแยกกิจกรรมที่ทำงานขนานกัน
Intermediate Testable Model	แผนภาพกิจกรรมที่ไม่ซับซ้อน มีโครงสร้างควบคุมแบบตามลำดับ แบบทางเลือก และอาจมีแบบวนซ้ำส่วนการแยกกิจกรรมที่ทำงานขนานกัน หากมีจำนวนกิจกรรมจำนวนมากอยู่ในนั้นจะทำให้เส้นทางการทดสอบเพิ่มเป็นทวีคูณตามจำนวนกิจกรรม
Activity Dependency Table and Activity Dependency Graph	แผนภาพกิจกรรมที่ไม่ซับซ้อน มีโครงสร้างควบคุมแบบตามลำดับและแบบทางเลือก
Test Path Reduction	แผนภาพกิจกรรมที่ไม่ซับซ้อน มีโครงสร้างควบคุมแบบตามลำดับและแบบทางเลือกเท่านั้น
Activity Convert Grammar	แผนภาพกิจกรรมที่ไม่ซับซ้อน ที่มีโครงสร้างควบคุมแบบตามลำดับ แบบทางเลือก ส่วนแบบวนซ้ำและแบบการแยกกิจกรรมที่ทำงานขนานกันควรเป็นอย่างง่ายที่มีเพียงหนึ่งกิจกรรมเท่านั้น
Activity Flow Graph	แผนภาพกิจกรรมที่ไม่ซับซ้อนหรือแบบซับซ้อนที่มีโครงสร้างควบคุมแบบตามลำดับ แบบทางเลือก และแบบการแยกกิจกรรมที่ทำงานขนานกัน ส่วนแบบวนซ้ำควรเป็นอย่างง่ายเพราะการทำงานจะวนซ้ำเพิ่ม 1 รอบเสมอ

5. บทสรุป

บทความนี้ศึกษาเปรียบเทียบขั้นตอนวิธีของการสร้างกรณีทดสอบอัตโนมัติจากแผนภาพกิจกรรม โดยศึกษาขั้นตอนวิธีการสร้างกรณีทดสอบจำนวน 8 วิธี คือ วิธี Round-Robin วิธี Condition-Classification Tree วิธี Activity Tree วิธี Intermediate Testable Model วิธี Activity Dependency Table and Activity Dependency Graph วิธี Test Path Reduction วิธี Activity Convert Grammar และวิธี Activity Flow Graph และใช้แผนภาพกิจกรรมจำนวน 4 แผนภาพ ผลการศึกษาเปรียบเทียบมีดังนี้ ประเด็นการครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้ วิธี Round-Robin วิธี Intermediate Testable Model วิธี Activity Convert Grammar และวิธี Activity Flow Graph สามารถสร้างกรณีทดสอบได้

ครอบคลุมเส้นทางทั้งหมดที่เป็นไปได้ แต่วิธี Round-Robin ไม่เหมาะสมกับแผนภาพกิจกรรมที่ในกิจกรรมแรกๆ มีกิจกรรมย่อย ประเด็นการครอบคลุมเส้นทางพื้นฐานพบว่าทุกวิธีสามารถสร้างกรณีทดสอบได้ครอบคลุมเส้นทางพื้นฐาน และประเด็นการสร้างกรณีทดสอบจากแผนภาพกิจกรรมจำลองความซับซ้อน พบว่าวิธี Round-Robin วิธี Intermediate Testable Model วิธี Activity Dependency Table and Activity Dependency Graph วิธี Activity Convert Grammar และวิธี Activity Flow Graph สามารถสร้างกรณีทดสอบได้ครอบคลุมเส้นทางที่เป็นไปได้ทั้งหมด แต่วิธี Condition-Classification Tree วิธี Activity Tree และวิธี Test Path Reduction ไม่สามารถสร้างเส้นทางได้ครบทุกเส้นทางที่เป็นไปได้ ผลการเปรียบเทียบความคลาดเคลื่อนของเส้นทางทดสอบที่สร้างได้เทียบกับเส้นทางทั้งหมดที่เป็นไปได้ พบว่าวิธี Activity Convert Grammar และวิธี Test Path Reduction มีค่าความคลาดเคลื่อนสูง และวิธี Activity Flow Graph มีค่าความคลาดเคลื่อนต่ำ

สำหรับงานวิจัยในอนาคตผู้วิจัยจะเลือกวิธีการสร้างกรณีทดสอบที่มีค่าความคลาดเคลื่อนต่ำไปปรับใช้ร่วมกับวิธีการทางปัญญาประดิษฐ์เพื่อคัดเลือกเส้นทางการทดสอบที่มีความเหมาะสมในประเด็นต่างๆ ที่จะเป็นประโยชน์ต่อการทดสอบ นอกจากนี้จะทำการศึกษาด้านเวลาและหน่วยความจำที่แต่ละวิธีการต้องใช้เพื่อนำไปสู่การปรับปรุงวิธีการให้มีประสิทธิภาพมากยิ่งขึ้น

เอกสารอ้างอิง

[1] F, Xin, S. Jian, L., LinLan and L. QiJun. “Test Case Generation from UML Subactivity and Activity Diagram”, *International Symposium on Electronic Commerce and Security*, *IEEE Computer Society*, Vol. 2, Oct. 2009. pp. 244-248

[2] S. Kansomkeat**,** P. Thiket and J. Offutt, “Generating test cases from UML activity diagrams using the Condition-Classification Tree Method”, *In Proceedings of 2nd International Conference on Software Technology and Engineering (ICSTE)*, Vol. 1, October. 2010. pp. 62-66

[3] R. K. Swain, V. Panthi, D. P. Mohapatra, P. K. Behera, “Prioritizing test scenarios from UML communication and activity diagrams”, Innovations in Systems and Software Engineering, Vol. 10, Issue 3, September 2014, pp. 165–180

[4] A. Nayak and D. Samanta, “Synthesis of Test Scenarios using UML Activity Diagrams”, *Software and Systems Modeling (SoSyM)*. Vol.10 Issue.1, February. 2011. pp. 63-89

[5] C. Chouhan, V. Shrivastava, P.S. Sodhi, “Test Case Generation based on Activity Diagram for Mobile Application”, *International Journal of Computer Applications*, Vol. 57, No. 23, November. 2012.

[6] P. N. Boghdady, N. L. Badr, M. Hashem and M. F. Tolba, “An Enhanced Test Case Generation Technique Based on Activity Diagrams”, *In Proceedings of the Seventh International Conference on Computer Engineering and Systems (ICCES)*. November-December 2011. pp. 289-294

[7] K. Pechtanun and S. Kansomkeat, “Generation Test Cases from UML Activity Diagram Based on AC Grammar”, *International Conference on Computer and Information Science (ICCIS)*, Vol. 2, September 2012. pp. 895-899

[8] R. K. Swain, V. Panthi, P. K. Behera, “Generation of Test Case using Activity Diagram”, *International Journal of Computer Science and Informatics*, Vol. 3, Issue. 2, November 2013. pp. 1-10

[9] P. N. Boghdady, N. L. Badr, M. Hashem and M. F.Tolba, “A Proposed Test Case Generation Technique Based on Activity Diagrams”, *International Journal of Engineering & Technology IJET/IJENS*, Vol. 11, No. 03, June 2011. pp. 37-57

[10] P. N. Boghdady, N. L. Badr, M. Hashem and M. F.Tolba, “An Enhanced Technique for Generating Hybrid Coverage Test Cases Using Activity Diagrams”, *The 8th International Conference on INFOrmatics and Systems (INFO2012)*, May. 2012. pp. 20-28